

# Designing Teams of Unattended Ground Sensors Using Genetic Algorithms

Ayse S. Yilmaz<sup>1</sup> and Brian N. McQuay<sup>1</sup> and Annie S. Wu<sup>1</sup> and John C. Sciortino, Jr.<sup>2</sup>

<sup>1</sup>School of Electrical Engineering and Computer Science,  
University of Central Florida, Orlando, FL 32816, USA;

<sup>2</sup>Naval Research Laboratory, Washington, DC 20375, USA

## ABSTRACT

Improvements in sensor capabilities have driven the need for automated sensor allocation and management systems. Such systems provide a penalty-free test environment and valuable input to human operators by offering candidate solutions. These abilities lead, in turn, to savings in manpower and time. Determining an optimal team of cooperating sensors for military operations is a challenging task. There is a tradeoff between the desire to decrease the cost and the need to increase the sensing capabilities of a sensor suite. This work focuses on unattended ground sensor networks consisting of teams of small, inexpensive sensors. Given a possible configuration of enemy radar, our goal is to generate sensor suites that monitor as many enemy radar as possible while minimizing cost. In previous work, we have shown that genetic algorithms (GAs) can be used to evolve successful teams of sensors for this problem. This work extends our previous work in two ways: we use an improved simulator containing a more accurate model of radar and sensor capabilities for our fitness evaluations and we introduce two new genetic operators, insertion and deletion, that are expected to improve the GA's fine tuning abilities. Empirical results show that our GA approach produces near optimal results under a variety of enemy radar configurations using sensors with varying capabilities. Detection percentage remains stable regardless of changes in the enemy radar placements.

**Keywords:** Radar detection, sensor suites, team development, evolving sensors, design.

## 1. INTRODUCTION

The problem of determining an optimal team of cooperating sensors for military operations is a challenging design problem. Tactical improvements along with increased sensor types and abilities have driven the need for automated sensor allocation and management systems. Such systems can provide valuable input to human operators in terms of exploring and offering candidate solutions, providing a testbed on which to evaluate candidate solutions, and providing a penalty-free environment on which to test for severe failure conditions.

Given a possible configuration of enemy radars, our goal is to design a team of sensors, a *sensor suite*, that can sense as many enemy radars as possible. Sensors are available in a wide range of capabilities and costs and the number of sensors in a team can vary. The mission goals may restrict the types and numbers of available sensors as well as the maximum cost. There is a tradeoff between the need to maximize the sensing capability of a sensor suite and the desire to minimize cost. The open-endedness of this problem makes it an interesting design problem. The number of potential components (sensor type, characteristics, and placement) that make up a solution is extremely large. There are very few restrictions as well as very little guidance as to how many and what types of these components would make up a good solution.

We use a genetic algorithm (GA) to tackle this problem of designing optimal sensor suites. In previous work, we developed a GA that successfully generates sensor suites for monitoring a variety of enemy radar

---

Further author information:

A.S.Y.: E-mail: selen@cs.ucf.edu

B.N.M.: E-mail: bmcquay@cs.ucf.edu

A.S.W.: E-mail: selen@cs.ucf.edu

J.C.S.: E-mail: john.sciortino@nrl.navy.mil

configurations. Candidate solutions are evaluated using an abstract simulation of radar and antenna sensing capabilities.<sup>1</sup> In this paper, we extend our work by improving the simulation model and making it more realistic. We also introduce problem specific genetic operators which are expected to improve the GA performance. We describe experiments which explore the performance of our GA in different types of enemy environments using different types of sensor resources. Our experimental results indicate that such a GA approach is able to design reasonable and effective sensor suites.

## 2. RELATED WORK

Evolutionary Algorithms have been used to generate optimal designs effectively. Peter Bentley's<sup>2</sup> pioneering work in evolutionary design incorporated evolutionary computation techniques into the evolution of design components that would build wide range of solid objects such as tables, prisms and cars. The system takes the input specifications for the object to be designed and evolves the shape of the design that performs the required function. Funes *et al.*<sup>3</sup> apply evolutionary techniques to the design of structures in a simulation environment that are then buildable using Lego parts. Instead of using expert engineering knowledge, they use a fitness function to evaluate the feasibility and functionality of evolved structures. Husbands *et al.*<sup>4</sup> report the results of a comparative study of ten different optimization techniques in structural design optimization. The distributed Genetic Algorithm (DGA) and various hybrid methods (DGA with gradient descent, Simulated Annealing with gradient descent) appear to have significant advantages over the other techniques tested. Lee *et al.*<sup>5</sup> develop a hybrid approach to evolve both controllers and robot bodies for performing specific tasks. They evolve physical body parameters such as the number and location of sensors and show that the design of robot body considerably can affect the behavior of the system.

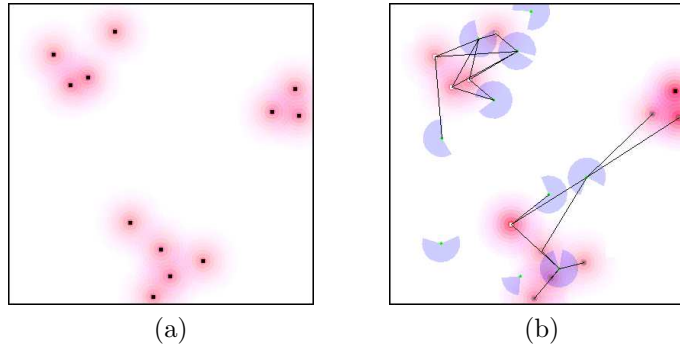
Bugajska *et al.*<sup>6</sup> investigate the co-evolution of form and function for autonomous agents. The form component in<sup>6</sup> focuses on the characteristics of a sensor suite. All sensors within a sensor suite are assumed to have the same characteristics. Characteristics that can be evolved are the number of the sensors in a sensor suite and the sensor coverage area of each sensor. The maximum number of sensors is limited to 32 and the detection angle varies from 5 to 30 degrees. The location of the sensors are fixed and the coverage degradation by the distance is not taken into consideration. In,<sup>7</sup> Bugajska *et al.* added the detection angle and the placement of the sensor to the list of evolved characteristics. They assume that power efficiency degrades with increased sensor coverage area. They evolve the number of the sensors implicitly by allowing a zero detection angle to indicate the existence of no sensor. The maximum number of sensors is limited to nine. The sensor detection angle ranges from 0 to 45 degrees. By contrast, in our system, detection angle can vary from 0 to 359 degrees and the placement of the sensors is not limited to fixed locations. The evolvable characteristics of our sensors are the location, detection coverage, power threshold, and diameter of each sensor.

The sensor management problem has been investigated by researchers using non-evolutionary techniques as well. Gaskell *et al.*<sup>8</sup> use a decision theoretic approach to develop a sensor management system for mobile robots. Popoli<sup>9</sup> proposes a sensor management scheme that uses fuzzy set theory, knowledge-based reasoning and expert systems. Schmaedeke *et al.*<sup>10</sup> uses an information theoretic approach for sensor to target assignment optimization. We believe the evolutionary technique and the advances in our work will have an advantage in terms of being robust and easily transferable between different problem scenarios with little or no modification to the system.

## 3. PROBLEM ENVIRONMENT

Before looking at the details of our GA implementation, it is necessary to understand the problem to which we apply the GA. The problem environment consists of a collection of stationary enemy radars located in a two-dimensional plane. Figure 1(a) shows an example environment consisting of twelve randomly placed enemy radars. Radars are represented as points surrounded by gradually fading circles. The environment is restricted to two dimensions in our current work but can be extended to three dimensions in the future.

Each radar location is identified by an  $x$  and  $y$  coordinate value. Radar strength is given by the Effective Radiated Power ( $ERP$ ) which is the product of the radar power ( $P_T$ ) and radar gain ( $G_T$ ). In our simulation,  $ERP$  ranges between  $10^6$  milliwatts (60 dBm) and  $10^{10}$  milliwatts (100 dBm). Radars transmit signals that



**Figure 1.** (a) Example problem environment; (b) Problem environment with candidate solution.

have radio frequency ( $f$ ) ranging between 300 MHz and 12000 MHz. The location,  $ERP$ , and frequencies of the enemy radar are configured beforehand and remain static throughout a run.

A radar must be detected by at least three difference sensors to be *fully detected*. (Three measurements are necessary for accurate triangulation of position.) Radars that are detected by two sensors are *partially detected* and radars that are detected by one sensor are *minimally detected*. Partially and minimally detected radars contribute less to the fitness evaluation than fully detected radars. In the experiments described here, we set all radars to operate on a single frequency in order to test the GA’s ability to evolve minimal cost teams.

Figure 1(b) shows the same environment along with a candidate solution. The pie shaped elements indicate sensors and their direction and detection range. Lines indicate detection of a radar by a sensor.

## 4. GA DESIGN

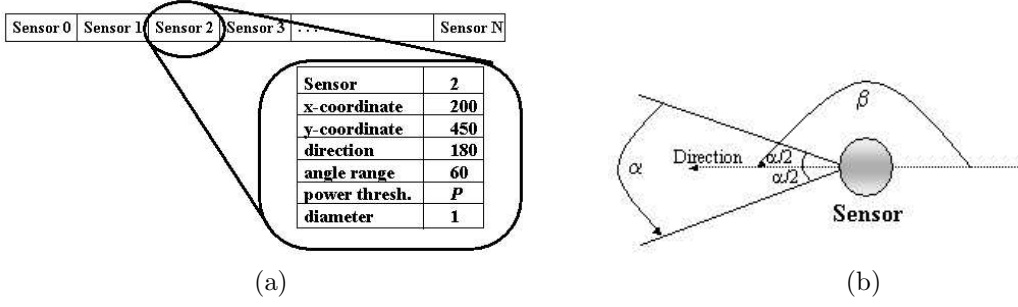
### 4.1. Problem Representation

Each individual in a GA population specifies the composition and arrangement of a team of sensors encoded as a vector of genes. Each gene encodes the evolvable characteristics for a single sensor. Figure 2(a) shows an example individual which represents a team of  $N$  sensors. Example parameter values for Sensor 2 are shown in detail. As the optimal number of sensors may not be known in advance, we allow the GA to evolve variable length individuals. Initially, each individual contains 20 randomly configured sensors. The maximum possible length of an individual is 150, indicating a maximum team size of 150 sensors.

The location of each sensor is encoded as  $x$  and  $y$  coordinate values. Sensors can be active or inactive. As we have no restriction on representation, multiple sensors in an individual may have the same location in the environment. If that occurs, only the sensor that appears first in the individual (the leftmost sensor) is active. The others are inactive and are unable to detect any radars; however, they are still included in the cost component of the fitness function.

Figure 2(b) illustrates the direction and the detection range of a sensor. The direction angle indicates the “front” of a sensor and is a counter-clockwise angle relative to the 3 o’clock position. A sensor’s detection range is centered around its direction and indicates the angle within which a sensor can detect signals. Suppose that we have a square environment with a radar in each corner and a sensor located in the center of the square. A sensor that has a direction of 45 degrees and a detection range of 90 degrees is capable of detecting only radars located in the upper right quadrant of the environment, as long as other requirements such as frequency and power are satisfied.

The minimum power threshold for each sensor indicates the minimum power value that a sensor must receive from an enemy radar in order for detection to occur. A sensor is initially randomly configured with a minimum threshold power for radar detection. The received signal power must be greater than the minimum power threshold for detection to occur; otherwise, the sensor will fail to detect that radar.



**Figure 2.** (a) Problem representation for a team of sensors; (b) Sensor characteristics:  $\alpha$  = detection range and  $\beta$ = direction

The base signal power generated by a radar is given by its effective radiated power ( $ERP$ ). Larger ERP values indicate stronger signals. Signal frequency affects the power received: power received decreases as frequency increases. Sensor gain amplifies the power received. The following formula summarizes the effect of all components on the signal power received by the sensor antenna. We use deciBel(dB) notation in the formulas.

$$S_{DB} = 10 \log ERP + G_{DBS} - 32.45 - 20 \log f - 20 \log R_E, \quad (1)$$

where  $S_{DB}$  is signal power received,  $ERP$  is the effective radiated power of the radar,  $f$  is the frequency of the signal transmitted,  $R_E$  is the distance between the radar and the sensor, and  $G_{DBS}$  is the gain of the sensor antenna.  $G_{DBS}$  is calculated using the following formula:

$$G_{DBS} = 20 \log(D \times f/c) + 7.34dB, \quad (2)$$

where  $D$  is the diameter of the sensor antenna and  $c$  is the speed of light,  $3.10^8$  m/sec. The value of the diameter,  $D$ , ranges from 0.43m to 1m.

## 4.2. Fitness Evaluation

The fitness function consists of two parts, the detection capability and the total cost of a solution. The fitness function is:

$$fitness = \rho/\sigma, \quad (3)$$

where  $\rho$  is the detection fitness that indicates the detection capability, and  $\sigma$  is the total cost of a solution. To calculate  $\rho$ , the detection fitness, we count the number of radars that are fully, partially, and minimally detected. The detection fitness is:

$$\rho = \frac{3 * F + 2 * P + M}{R}, \quad (4)$$

where  $R$  is the number of enemy radars and  $F, P, M$  are the numbers of fully, partially and minimally detected radars, respectively.

The fitness is inversely proportional to the total solution cost. The total cost of a solution is its basic cost plus the total cost of all of the sensors:

$$\sigma = \mathbf{b} + \sum_{i=1}^n ((D_i - p_i)/100), \quad (5)$$

where  $b$  is the fixed basic cost of the deployment,  $n$  is the total number of sensors,  $D_i$  is the antenna diameter of the sensor  $i$ ,  $p_i$  is the minimum power threshold of sensor  $i$ .

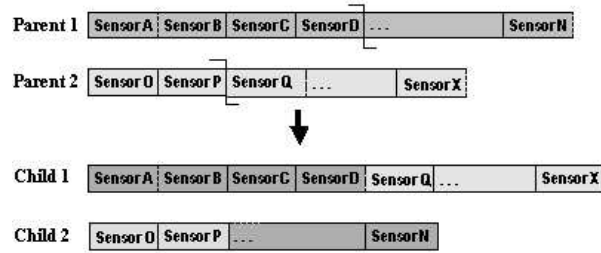


Figure 3. Inter-gene level crossover operation

### 4.3. Selection and Genetic Operators

The selection method we use is deterministic tournament selection with size two. One-point crossover operator is applied to variable length individuals where the crossover point on each parent is chosen independently. The length of an offspring may be different from its parents. Crossover points always fall in between the genes as shown in Figure 3.

We use a mutation scheme that is different from traditional mutation. Mutation is done at the intra-gene level. Each characteristic of each gene is subject to the mutation rate. A Poisson distribution function is used to generate a value as an offset from the original value. That is, the new value is not randomly selected from a specified range, but rather follows a distribution probability that favors smaller changes. As a result, mutation is more likely to generate values that are similar to the original value instead of simply mutating randomly to any new value. We expect this mutation scheme to encourage accurate adjustment of the location, direction angle, detection angle range, diameter and power threshold of the sensors.

In addition, we use two problem specific operators called insertion and deletion mutation. Insertion mutation inserts a new sensor with randomly initialized random characteristics into the sensor suite. The probability that a new sensor will be inserted into an individual is given by the ‘insertion mutation rate’ parameter. Deletion mutation randomly selects a sensor to be removed from the sensor suite. The ‘deletion mutation rate’ parameter gives the probability that each sensor will be deleted from its team.

## 5. EXPERIMENTS

The goal of our experiments is to demonstrate that a GA is capable of designing teams of sensors for the detection of the enemy radars in a reasonable amount of time. We test our GA on a variety of environmental scenarios and sensor configurations to study the robustness of the system with respect to the inputs for the environmental conditions.

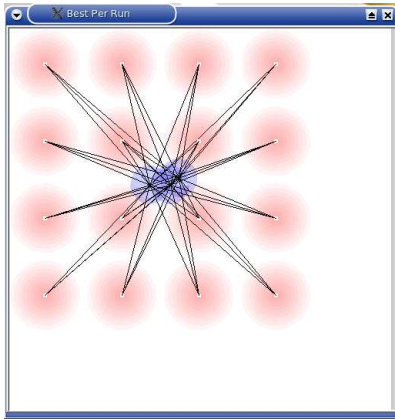
### 5.1. Test Sets

The two environmental scenarios that we test are :

- Radars organized in a  $4 \times 4$  grid pattern totaling up to 16 radars. We expect a GA to evolve a relatively distributed placement of sensors in this environment. Placing sensors within the area covered by radars is expected to be most efficient; however, the rightmost column and the bottom row of the environment are left empty to see if the GA places sensors outside the perimeter of the radars.
- Radars clustered at random locations: 15 radars in four clusters. We expect sensors to be clustered near radar clusters. The locations that have no enemy radars are expected to have no sensors.

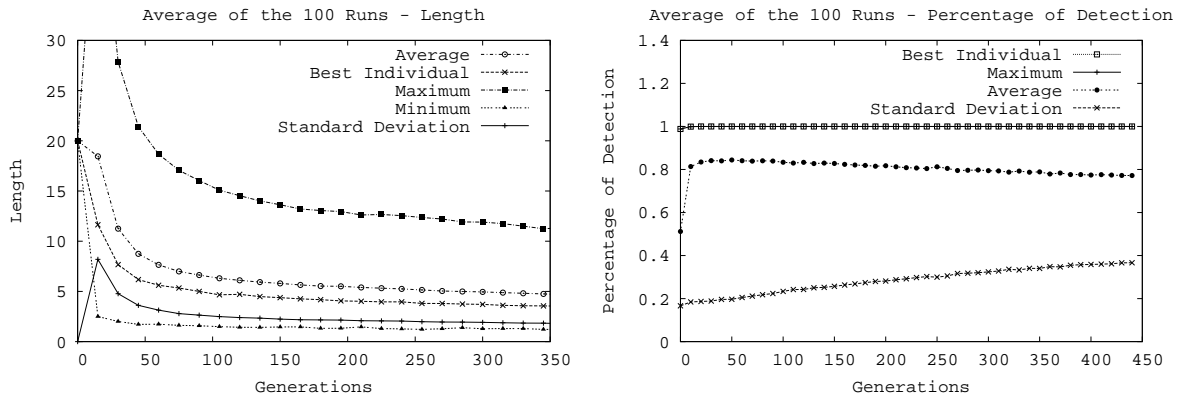
The two general types of sensors that we test are:

- Long range sensors that are able to cover the entire environment. As any sensor should be able to detect all radars, we expect the GA to evolve solutions that are close to three sensors, the minimum number of sensors required to fully detect a radar.



Per Run Statistics	
Generation:	417
Best Individual's Length:	3
Average Length of Population:	4.705
Avg Length Standard Deviation:	1.722781181694297
Best Individual's number of Active Sensors	3
Best individual's Fitness	0.057100962363086774
Average Fitness	0.04980492011802114
Avg Fitness Standard Deviation	0.009693892541355791
Best individual's Cost	53.76521422810485
Average Cost	52.53851906950287
Avg Cost Standard Deviation	1.3899960668378737
Number of Enemy Radars	16
Number of Enemy Radar detected by Best Individual	16

**Figure 4.** Long range sensors with different number of enemy radars located in a grid for a single run



**Figure 5.** Length (Number of Sensors) and percentage of detection averaged over 100 runs for long range sensors and  $4 \times 4$  grid placement scheme for enemy radars

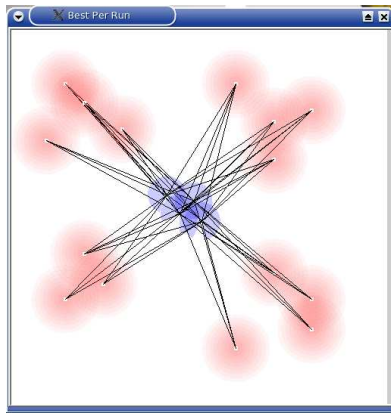
- Short range sensors that cover less than a quarter of the environment. We expect solutions to consist of more sensors as short range sensors cannot sense the entire environment.

Using the test cases explained above, we test four different scenarios in our experiments:

1. Long range sensors with radars placed evenly in a grid pattern.
2. Long range sensors with radars placed randomly in three clusters
3. Short range sensors with radars placed evenly in a grid pattern.
4. Short range sensors with radars placed randomly in three clusters

All enemy radars are set to operate on same frequency. We selected the following GA parameter settings based on the performance of previous experiments:

Population size : 200, initialized randomly  
 Initial length : 20  
 Parent Selection : Tournament, size:2  
 Crossover type : one-point  
 Crossover rate : 0.7



Per Run Statistics	
Generation:	370
Best Individual's Length:	4
Average Length of Population:	5.53
Avg Length Standard Deviation:	2.3149730020023984
Best Individual's number of Active Sensors:	4
Best individual's Fitness:	0.055665169916556866
Average Fitness:	0.049036600231444266
Avg Fitness Standard Deviation:	0.00801202739899364
Best individual's Cost:	55.0969336121477
Average Cost:	53.89366464697864
Avg Cost Standard Deviation:	2.0787249369921517
Number of Enemy Radars:	15
Number of Enemy Radar detected by Best Individ...	15

Figure 6. Long range sensors with enemy radars located randomly in 4 clusters for a single run

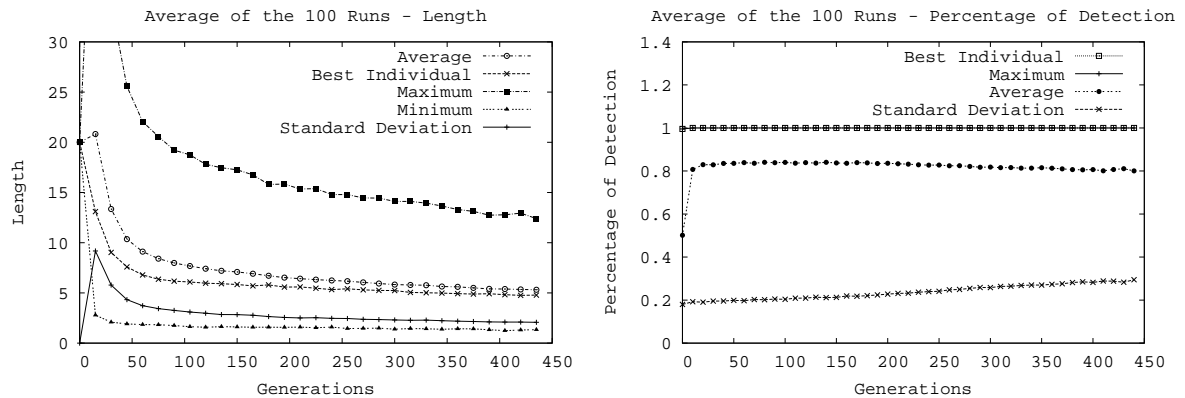


Figure 7. Length (Number of sensors) and percentage of detection averaged over 100 runs for long range sensors and cluster placement scheme for enemy radars

Mutation rate : 0.01 (per gene)  
 Deletion Mutation rate : 0.05 (per gene)  
 Insertion Mutation rate : 0.1 (per individual)  
 Max number of generations : 450  
 Number of runs : 100

## 5.2. Test Results

Figure 4 shows an example solution from our first scenario. In this experiment, sensors are able to extend their coverage area to include the whole environment. We observe that the GA is able to evolve a team of three sensors the minimum requirement to fully detect all enemy radars. There are no undetected, minimally nor partially detected radars. All of the 16 enemy radars are *fully* detected. The results we obtain over 100 runs are consistent with the example solution. Figure 5 shows the evolution of individual length, i.e. number of sensors evolved, and percentage of radar detection averaged over 100 runs. The number of sensors evolved in the best individuals levels off around four, and the percentage of radar detection for the best individual is typically 100%.

The second scenario uses long range sensors in a clustered environment. One of the best solutions evolved teams of four sensors as shown in Figure 6. All 15 radars are fully detected. The average behavior of the GA over 100 runs, as shown in Figure 7, is similar to that shown in Figure 5, suggesting that the behavior of this GA, in terms of the number of evolved sensors and the percentage of detection, is independent of the placement of the enemy radars. This result is expected as long range sensors should have the same capability regardless of



Figure 8. Short range sensors with enemy radars located in a grid for a single run

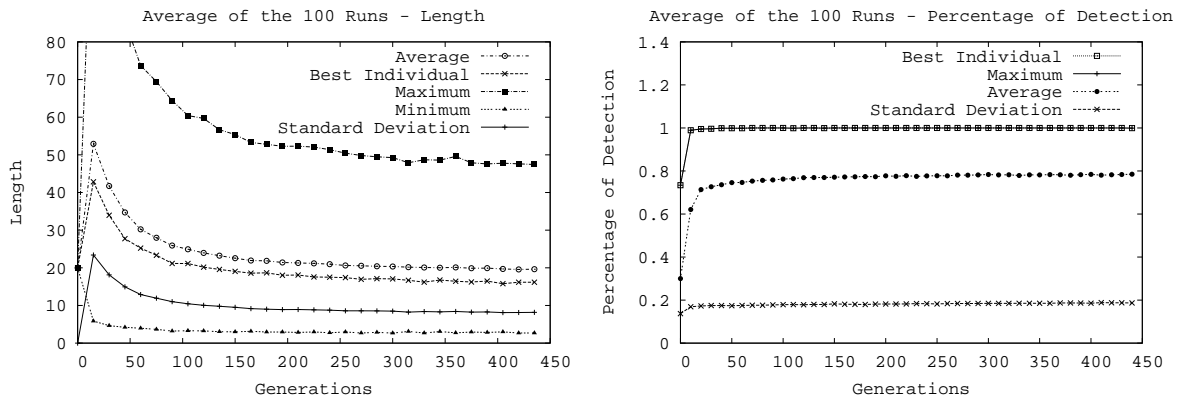


Figure 9. Length (Number of sensors) and percentage of detection averaged over 100 runs for short range sensors and grid placement scheme for enemy radars

the enemy radar configuration. In general, sensors tend to cluster around the center of the environment in order to detect all radars most efficiently.

The third scenario employs sensors that can sense a maximum of one quarter of the environment. Given the distributed radars and assuming the need for at least three sensors for each quarter of the environment, we anticipate solutions to contain a minimum of twelve sensors. The GA is able to evolve a team of fourteen sensors, as shown in the example solution in Figure 8. All enemy radars are fully detected except for one partially detected radar. The fourth scenario employs short range sensors in a clustered environment. We expect each cluster to require at least three sensors for full detection. Figure 10 shows an example solution that evolves thirteen sensors. All the radars are fully detected. Because of their decreased range capacity, most of the sensors can only detect radars within the one of the clusters.

The results we obtain over 100 runs are also consistent with the single run results. The individual length, i.e. the number of sensors evolved and the percentage of detection averaged over 100 runs are reported in Figure 9 and Figure 11, for grid and the clustered placement of the enemy radars respectively. The GA achieves a detection percentage of 100% for both radar placement schemes. In addition, the number of sensors evolved in the best individuals levels off to around sixteen for both grid placement and the clustered placement. Due to limited capability of the sensors, we observe a considerable increase in the number of sensors evolved compared to the scenarios with long range sensors.





Figure 10. Short range sensors with enemy radars located randomly in 4 clusters

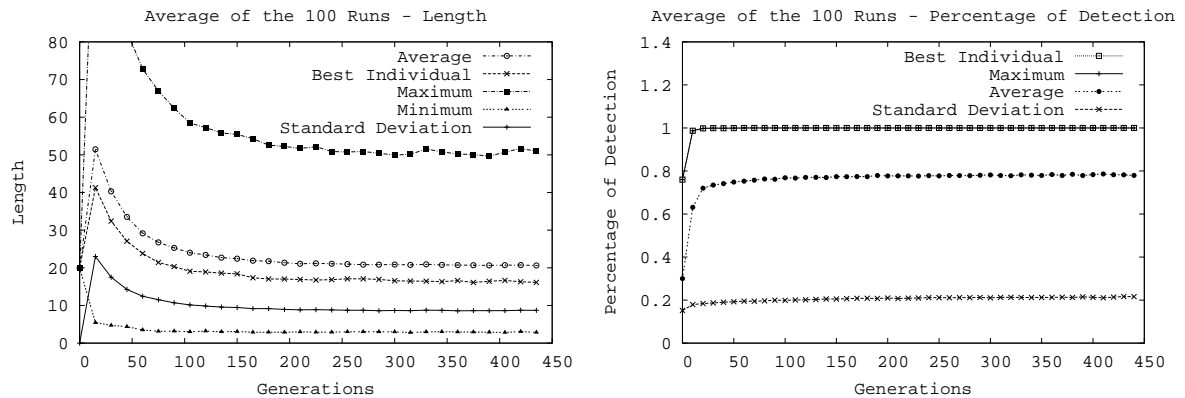


Figure 11. Length and percentage of detection averaged over 100 runs for short range sensors and cluster placement scheme for enemy radars

## 6. CONCLUSION

In this study, we examine the problem of designing teams of sensors to detect enemy radars located in different configurations. We extend previous work<sup>1</sup> to include a more accurate description of the sensors and a more realistic detection criteria. In addition, we modify the GA to use problem specific operators such as insertion and deletion. Based on the detection criteria we have, the results we obtain are promising. Our GA is able to evolve close to the optimal number of sensors based on the capabilities of the sensors. As compared to our previous work, the current GA evolves solutions that have more fully detected radars. In addition, the detection percentage has increased to 100%. The problem specific operators introduced appear to increase the GA performance. The results obtained are robust to changes in the sensing capabilities of the sensors and changes to the enemy radar configurations.

Future work includes investigation of additional problem specific improvements to the GA and an evaluation of the GA's ability to identify building blocks that form effective subteams. In addition, we would like to evaluate GA performance on more complex problems, for example, in environments which contain obstacles or mobile radar or sensors.

## Acknowledgements

The study we reported in this paper is sponsored by the Naval Research Laboratory, ITT Industries Inc., and partially supported by NSF.

## REFERENCES

1. A. S. Yilmaz, B. N. McQuay, H. Yu, A. S. Wu, and J. John C. Sciortino, "Evolving sensor suites for enemy radar detection," in *Genetic and Evolutionary Computation Conference - GECCO 2003*, Springer Verlag, Berlin.
2. P. J. Bentley, "Generic evolutionary design of solid objects using a genetic algorithm," in *Ph.D. thesis, Division of Computing and Control Systems, School of Engineering, University of Huddersfield*, 1996.
3. P. Funes and J. Pollack, "Computer evolution of buildable objects," in *Procs. of fourth European Conference on AI*, 1997.
4. P. Husbands, G. Jermy, M. McIlhagga, and R. Ives, "Two applications of genetic algorithms to component design," in *Workshop on EC*, 1996.
5. W. Lee, J. Hallam, and H. Lund, "A hybrid gp/ga approach for co-evolving controllers and robot bodies to achieve fitness-specified tasks," in *Procs of IEEE third International Conference on Evolutionary Computation*, 1996.
6. M. Bugajska and A. Schultz, "Co-evolution of form and function in the design of autonomous agents: Micro air vehicle project," in *GECCO-2000 Workshop on Evolution of Sensors in Nature, Hardware and Simulation*, (Las Vegas, NV), 2000.
7. M. Bugajska and A. Schultz, "Co-evolution of form and function in the design of micro air vehicles," in *NASA/DoD Conference on Evolvable HW*, 2002.
8. A. Gaskell and P. Probert, "Sensor models and a framework for sensor management," in *Sensor Fusion VI. Proceedings of SPIE*, **2059**, 1993.
9. R. Popoli, "The sensor management imperative," in *Chapter in Multitarget-Multisensor Tracking: Applications and Advances*, **II**, 1992.
10. W. Schmaedeke and K. Kastella, "Information based sensor management and immkf," in *Signal and data processing of small targets 1998: proceedings of the SPIE*, **3373**, (Orlando, FL), 1998.