Evolving Sensor Suites For Enemy Radar Detection

Ayse S. Yilmaz[1], Brian N. McQuay[1], Han Yu[1], Annie S. Wu[1], and John C. Sciortino, Jr.[2]

[1] School of Electrical Engineering and Computer Science,
University of Central Florida,
Orlando, FL 32816, USA
{selen, bmcquay, hyu, aswu}@cs.ucf.edu
[2] Naval Research Laboratory,
Washington, DC 20375, USA
john.sciortino@nrl.navy.mil

Abstract. Designing optimal teams of sensors to detect the enemy radars for military operations is a challenging design problem. Many applications require the need to manage sensor resources. There is a tradeoff between the need to decrease the cost and to increase the capabilities of a sensor suite. In this paper, we address this design problem using genetic algorithms. We attempt to evolve the characteristics, size, and arrangement of a team of sensors, focusing on minimizing the size of sensor suite while maximizing its detection capabilities. The genetic algorithm we have developed has produced promising results for different environmental configurations as well as varying sensor resources.

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 find a team of sensors that can sense as many enemy radar as possible. Sensors are available in a wide range of capabilities and costs and the number of sensors in a team can vary. The importance of a mission 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.

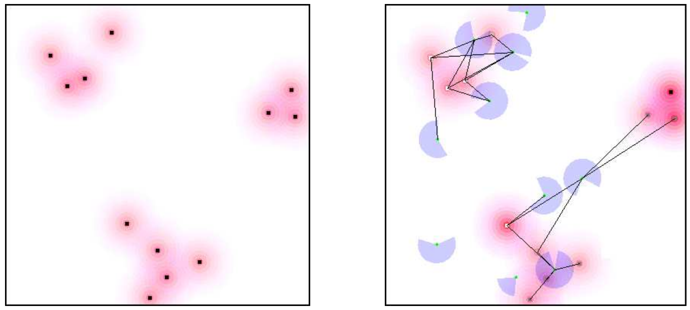
The open-endedness of this problem makes it an interesting design problem. The number of potential components (sensor type, characteristics, and placement) that make up a solution is extremely large. There are very few restrictions as well as very little guidance as to how many and what types of these components would make up a good solution.

We use a genetic algorithm (GA) to tackle this problem of designing optimal sensor suites. We have developed a GA that will generate candidate sensor suites that are evaluated using a simulator based on military specification. In this paper, 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 [1] pioneering work in evolutionary design incorporated evolutionary computation techniques to into the evolution of design components that would build wide range of solid objects such as tables, prisms, 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.* [2] apply evolutionary techniques to the design of structures in a simulation environment that are then buildable from parts 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.* [3] 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. [4] 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.* [5] investigate the co-evolution of form and function for autonomous agents. The form component in [5] focuses on the characteristics of the sensor suite. All sensors within a sensor suite are assumed to be have same the 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 [6], 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 can range from 0 to 45 degrees. In our work, 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 location, detection coverage, power and the frequency band the sensor operate at.



(a) (b)

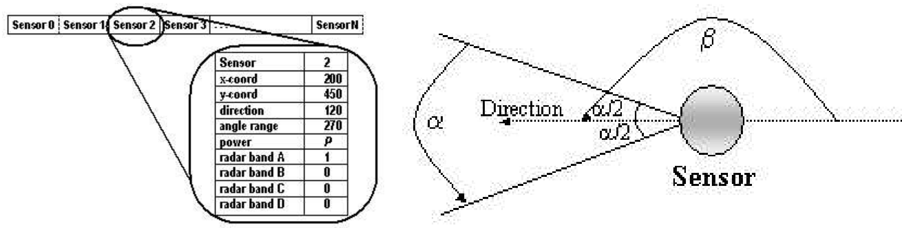
Fig. 1. (a) Example problem environment; (b) Problem environment with candidate solution.

The sensor management problem has been investigated by researchers using non-evolutionary techniques as well. Gaskell *et al.* [7] use a decision theoretic approach to develop a sensor management system for mobile robots. Popoli [8] proposes a sensor management scheme that uses fuzzy set theory, knowledge-based reasoning and expert systems. Schmaedeke *et al.* [9] 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 a 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 radar. Radar are represented as points surrounded by gradually fading circles. The environment is restricted to two dimensions in our current work but can easily be extended to three dimensions in the future.

The location, power, and frequencies of the enemy radar are configured beforehand and remain static throughout a run. Each radar can operate on one of four different frequencies: A, B, C, or D. A radar can only be detected by sensors that are configured to sense on the same frequency. 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 radar contribute less to the fitness evaluation than fully detected radar. In the experiments described here, we set all radar to operate on a single frequency in order to test the GA's ability to evolve miminal cost teams.



(a) (b)

Fig. 2. (a) Problem representation for the evolvable team of sensors; (b) Sensor characteristics, α = Detection angle range and β= Direction angle

Figure 1(b) shows the same environment along with a candidate solution. The pie shapes elements indicate sensors and their detection angle and 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 solution that is 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 100, indicating a maximum team size of 100 sensors.

The location of each sensor is encoded as x and y coordinate values. Sensors can be active or inactive. If multiple sensors in an individual have the same location in the environment, 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 angle and the detection angle range. The direction angle indicates the "front" of a sensor and is a counter-clockwise angle relative to the 3 o'clock position. The detection angle range is centered around the direction angle and indicates the range 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. If the sensor has a direction angle of 45 degrees and a detection angle range of 90 degrees. This sensor would point to the upper right corner of the square and would be able to detect only within the upper right quadrant of the environment. This particular sensor is only capable of detecting the radar located in the upper right corner 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 power a sensor receives from a radar is inversely proportional to the distance between the sensor and the radar. The received power is also proportional to the base power of each enemy radar. Thus, $p = r/d$, where p is the power received by a sensor, r is the enemy radar power, and d is the distance between the sensor and the enemy radar. Only the power received by sensor is greater than the minimum threshold for that sensor can it detect a radar. The minimum threshold value that can be evolved is configurable as a system parameter which enables us to vary the covereage capabilities of the sensors that can be generated within a GA run.

Each sensor can detect radar frequencies in zero to four bands (as represented by A, B, C, and D in section refsec.prob). Detection is represented by a four bit binary string in each gene. Each bit refers to one radar band. A 1 denotes the ability to detect the radar frequency represented by that bit and a 0 indicates inability to detect the corresponding radar frequency. Sensors are capable of detecting on more than one band. A sensor can detect a radar only if it operates on the same radar frequency as the radar.

4.2 Fitness Evaluation

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

$$f = \rho/\sigma, \tag{1}$$

where f is the raw fitness, ρ is the detection fitness that indicates the detection capability, and σ is the total cost of a solution. To calculate ρ, the detection fitness, we count the number of radars that are fully, partially, and minimally detected. The detection fitness is:

$$\rho = \frac{3 * F + 2 * P + M}{R}, \tag{2}$$

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

The raw 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:

$$\sigma = b + \sum_{i=1}^{n}(p + q), \tag{3}$$

where b is the fixed basic cost of deployment, p is the fixed basic cost of each sensor, n is the total number of sensors and q is the cost due to the band

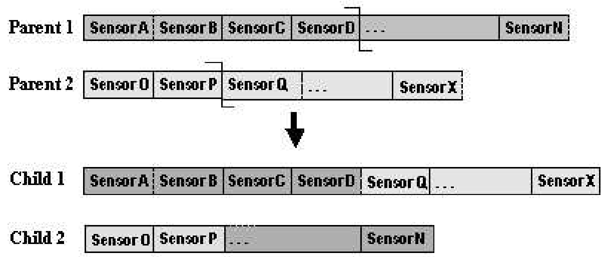


Fig. 3. Inter-gene level crossover operation

frequencies that are equipped in each sensor. The cost of radar bands favors multi-band sensors in that it is cheaper to have a sensor with four bands than to have two sensors with two bands each. The costs given to a sensor according to the number of bands it has are 1, 3, 5, 7 and 9 for zero, one, two, three and four bands respectively. Two 2-band sensors would cost 10 while one 4-band sensor would cost 9, thus, favoring multi-band sensors.

4.3 Selection and Genetic Operators

The selection method we use is 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 on the intra-gene level. Each characteristic of each gene is subject to the mutation rate. If frequency band is to be mutated, we randomly generate a value (0 or 1) for the mutated bit. If the location, direction angle, detection angle range, or minimum threshold power are to be mutated, 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. We use this mutation scheme to encourage accurate adjustment on the location, detection angle, angle range and power of the sensors as random mutation does not help to fine tune the value and improve the detection performance in a long run.

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 we test are :

- Radars organized in a 5×5 grid pattern totaling up to 25 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, 12 radars in three 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 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.
- 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 for in our experiments:

1. Long range sensors with radars placed evenly in a grid pattern.
2. Longe 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 of the enemy radars are set to operate on a single band (A). The GA parameter settings we have used throughout all the scenarios are:

Population size	: 200, initialized randomly
Initial length	: 20
Parent Selection	: Tournament, size:2
Crossover type	: one-point
Crossover rate	: 0.7
Mutation rate	: 0.01 (per gene)
Max number of generations	: 400
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.

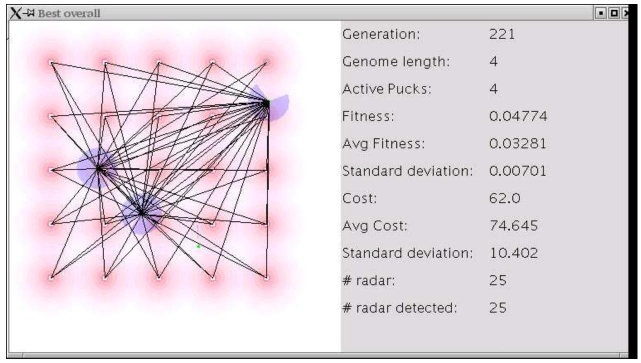


Fig. 4. Long range sensors with enemy radars located in a grid for a single run

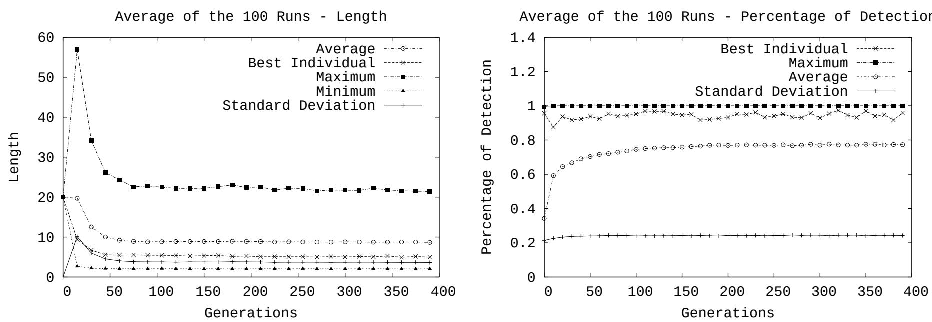


Fig. 5. Length and percentage of detection averaged over 100 runs for long range sensors and grid placement scheme for enemy radars

We observe that the GA is able to evolve a team of four sensors which is close the minimum requirement of three to fully detect all enemy radars. There are no undetected radars. 24 out of 25 enemy radars are *fully* detected . Only one enemy radar located at the top right corner is detected by two sensors. 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 in best individual is always higher than 80%. The second scenario uses long range sensors in a clustered environment. Our best solutions are again teams of four sensors as shown in Figure 6. All radar except for one are fully detected; the remaining is partially 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 the

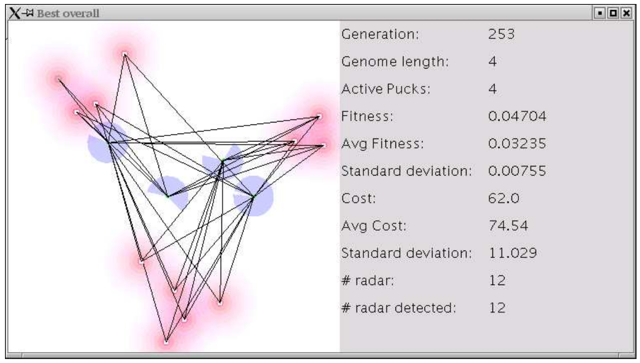


Fig. 6. Long range sensors with enemy radars located randomly in 3 clusters for a single run

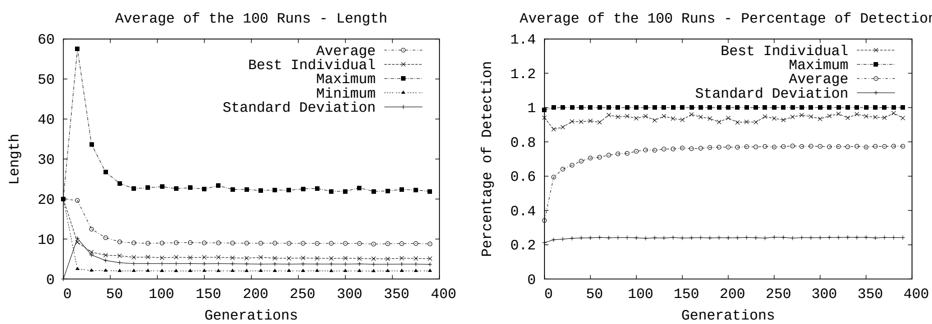


Fig. 7. Length and percentage of detection averaged over 100 runs for long range sensors and cluster placement scheme for enemy radars

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

The third scenario employs sensors that can sense a maximum of one quarter of the environment. Given the distributed radar 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 detected by at least one sensor. 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 that the GA is successful at evolving a minimum number of sensors of nine. There are no undetected radar. The number of radars detected by one, two and three sensors are one, seven and four respectively. Because of their decreased range capacity, all of the sensors can only detect radars within the one of the clusters.

We explored additional varying sensor ranges and include an example solution using medium range sensors which can potentially cover approximately 75% of

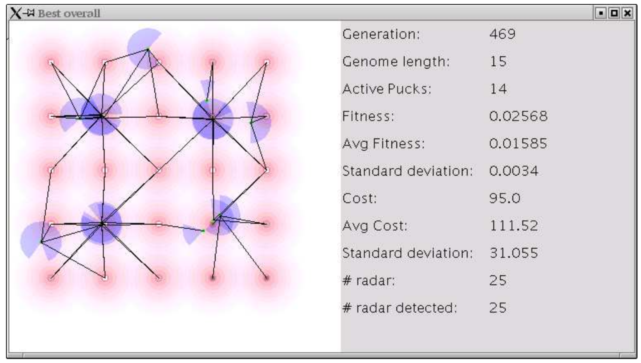


Fig. 8. Short range sensors with enemy radars located in a grid for a single run

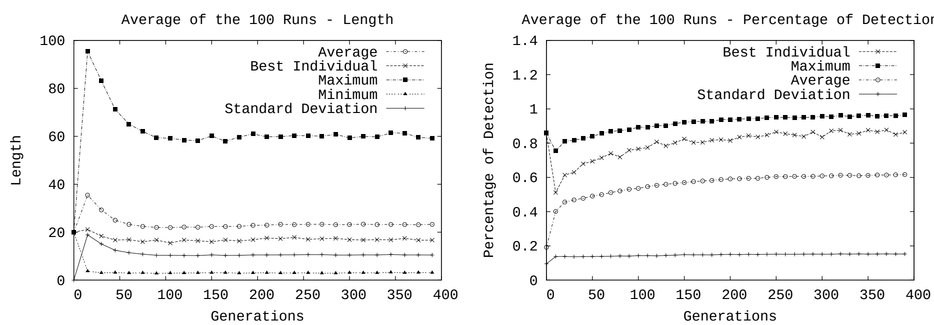


Fig. 9. Length and percentage of detection averaged over 100 runs for short range sensors and grid placement scheme for enemy radars

the working environment. This solution, shown in Figure 11, includes sensors that are mainly focused within clusters, with a few that are able to reach multiple 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 12, for grid and the clustered placement of the enemy radars respectively. GA is able to detect the enemy radars with same performance for both placement scheme with a detection percentage around 90%. However the number of sensors evolved in best individuals levels off around sixteen for grid placement while it's near fourteen for the clustered placement which makes short range sensors more sensitive to the changes in the environment than the long range sensors. 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.

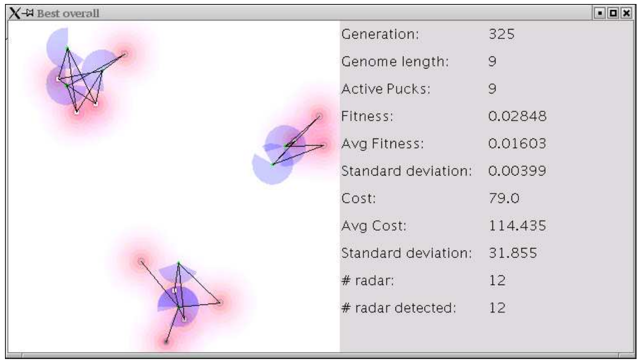


Fig. 10. Short range sensors with enemy radars located randomly in 3 clusters

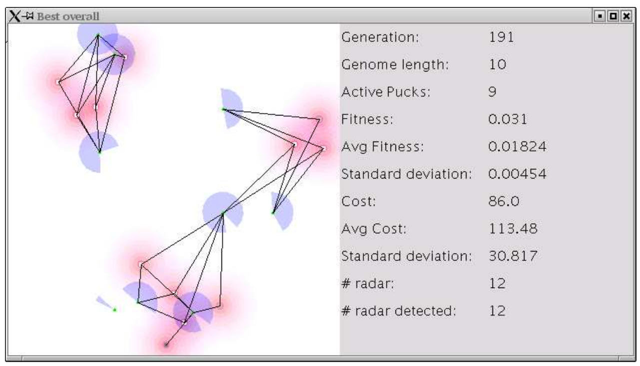
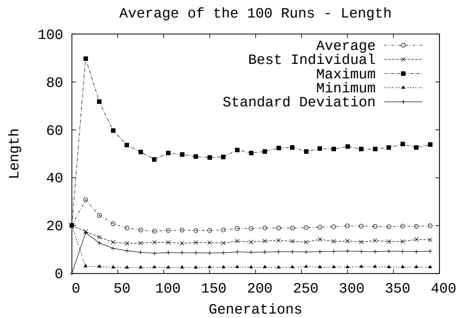


Fig. 11. Medium range sensors with enemy radars located randomly in 3 clusters

6 Conclusion

In this study, we examine the problem of designing teams of sensors to detect enemy radars located in different configurations. We extend the basic GA design to enable evolving as much sensor characteristics as possible. 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. Our system is demonstrated to be robust to the changes in the placement of the enemy radars. The detection percentage is around 90% with the limited sensing capability of the sensors and around 100% with the sensors that have full coverage capability independent from the enemy radar placement.Future goals include testing the robustness and the performance of our GA on more realistic scenarios such as ones with moving radars . In addition we plan to investigate the adaptibility of our system to changing environments, to introduce more problem specific genetic operators and to investigate the identification of the building blocks that form effective subteams.



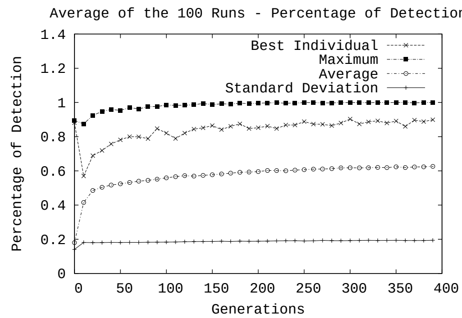


Fig. 12. Length and percentage of detection averaged over 100 runs for short range sensors and cluster placement scheme for enemy radars

References

1. Bentley, P.J.: 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)
2. Funes, P., Pollack, J.: Computer evolution of buildable objects. In: Proceedings of fourth European Conference on AI. (1997)
3. Husbands, P., Jermy, G., McIlhagga, M., Ives, R.: Two applications of genetic algorithms to component design. In: Workshop on EC. (1996)
4. Lee, W., Hallam, J., Lund, H.: A hybrid gp/ga approach for co-evolving controllers and robot bodies to achieve fitness-specified tasks. In: Proceedings of IEEE third International Conference on Evolutionary Computation. (1996)
5. Bugajska, M., Schultz, A.: 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)
6. Bugajska, M., Schultz, A.: Co-evolution of form and function in the design of micro air vehicles. In: NASA/DoD Conference on Evolvable HW. (2002)
7. Gaskell, A., Probert, P.: Sensor models and a framework for sensor management. In: Sensor Fusion VI. Proceedings of SPIE. Volume 2059. (1993)
8. Popoli, R.: The sensor management imperative. In: Chapter in Multitarget-Multisensor Tracking: Applications and Advances. Volume II., Artech House Publishers (1992)
9. Schmaedeke, W., Kastella, K.: Information based sensor management and immkf. In: Signal and data processing of small targets 1998: proceedings of the SPIE. Volume 3373., Orlando, FL (1998)