Evolution of Sensor Suites for Complex Environments

Annie S. Wu, Ayse S. Yilmaz, and John C. Sciortino, Jr.

Abstract— **We present a genetic algorithm (GA) based decision tool for the design and configuration of teams of unmanned ground sensors. The goal of the algorithm is to generate candidate solutions that meet cost and performance constraints. The GA evolves the membership, placement, and characteristics of a team of cooperating sensors. Previous work shows that this algorithm can generate successful teams in simple, obstacle free environments. This work examines the performance of our algorithm in environments that include obstacles.**

I. Introduction

The pervasiveness of technology in today's military have extended the military theater into the realm of the electromagnetic (EM) spectrum. Activity such as radio communication, laser guided control, and radar emissions all reside within the EM spectrum. Electronic Warfare (EW) refers to military actions focused on the control and use of the EM spectrum. EW is accomplished using offensive electronic attack (EA) and defensive electronic protection (EP) actions. The choice and implementation of EA and EP actions are determined by a third component of EW, electronic warfare support (ES). ES involves actions which intercept, identify, and analyze enemy radiations with a goal of detecting threat conditions and recognizing offensive opportunities.

This work addresses a general problem in ES: determining an appropriate team and organization of sensors that provides maximal detection capabilities in a given scenario. Identification and location of enemy emitters allow intelligence to be formed about the enemy order of battle, both electronic and physical. This knowledge allows for the planning of surveillance and reconnaissance. These capabilities are part of Command and Control Warfare (C2W) which is designed to prevent an enemy from exercising control over their units or at least degrading such control. Once emitter locations are known, they can be eliminated. Since emitters are associated with weapons systems, this knowledge also eliminates the weapons systems. Battle damage assessment can also be undertaken through electronic surveillance.

Previous work has shown that a genetic algorithm (GA) approach can successfully address this problem of the formation and organization of teams of unattended ground sensors [6]. This work focused on simple problem environments and investigated the GA's ability to design optimal teams of sensors for given enemy scenarios. In addition to finding good solutions in terms of the number and organization of sensors, the GA approach exhibits an added advantage of not being scenario specific, that is, the GA requires little or no reconfiguration from one problem scenario to the next. Related work in evolutionary robotics have found evolutionary algorithms to be an effective approach for designing sensor suites for autonomous agents [1], [2], [3]. These problems are more complex in that the possible sensor configurations are restricted by the physical parameters of autonomous robots.

Annie S. Wu and Ayse S. Yilmaz are with the School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816-2362, USA (email: {aswu,selen}@cs.ucf.edu).

John C. Sciortino, Jr. is with the Naval Research Laboratory, Washington, DC 20375 (email: john.sciortino@nrl.navy.mil).

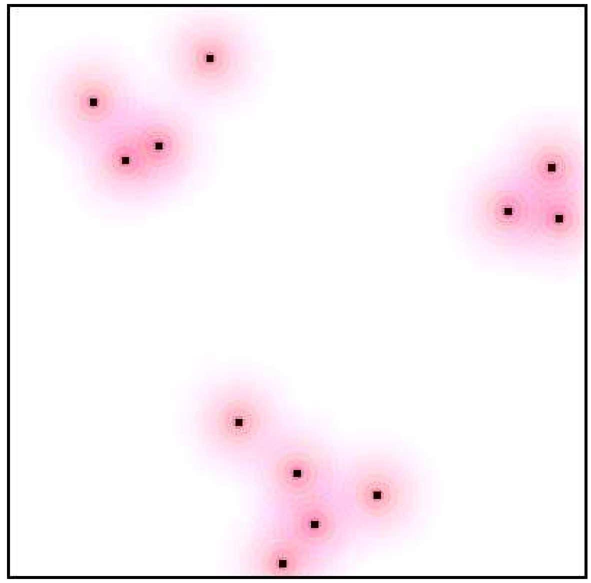


Fig. 1. Example problem environment.

In this paper, we extend our previous studies [6] to examine more complex environments that include obstacles. The addition of obstacles greatly restricts the placement and reach of sensors and complicates the problem of building and organizing effectively cooperating teams of sensors. We examine a series of test scenarios and evaluate the composition and placement of the evolved teams. Results indicate that the GA is able to intelligently design sensor placements that minimize the negative effects of obstacles in the environment.

II. Test problem

Our problem environment is an abstract simulation environment consisting of a two dimensional working area in which obstables and a collection of enemy radar are placed. Figure 1 shows an example environment consisting of twelve randomly placed enemy radar and no obstacles. Radar are represented as points surrounded by gradually fading circles. The location, power, and frequencies of the enemy radar are configured beforehand and remain static throughout a run. Radar can only be detected by sensors that are configured to sense on the same frequency. A radar must be detected by at

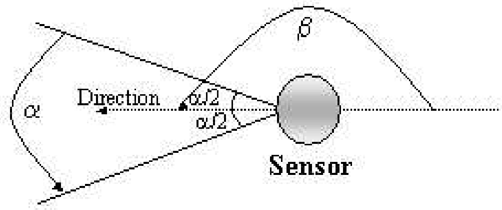


Fig. 2. Sensor characteristics: α = detection range and β= orientation.

least three sensors to be *fully detected*. (Three measurements are necessary for triangulation of position.) Radar that are detected by two sensors are *partially detected* and radar that are detected by one sensor are *minimally detected*.

The obstacles in our environment are modeled as solid rectangular objects that can vary in size. The location and size of an obstacle are predefined and remain unchanged during the course of a run. Obstacles that intersect the direct line between a sensor and a radar block that sensor's ability to detect that radar.

Sensor placement is specified as *x* and *y* coordinates and direction of orientation. As shown in Figure 2, orientation is specified as an angle, β, which runs counter clockwise with zero degrees at due east. Sensor characteristics include detection angle, power threshold, and frequency range. The detection angle, α, is centered around the direction of orientation within which a sensor can detect signals. Larger values provide greater detection capability. Both orientation and detection angle range from zero to 360 degrees. The power emitted by a radar decreases proportionally with the distance squared. Radar power must exceed the minimum power threshold of a sensor in order for that sensor to detect the radar. Frequency is represented as discrete intervals that are turned on or off. The number of available frequency intervals is a pre-defined constant.

We examine two types of sensors in our experiments. Long-range sensors have a maximum sensing range that covers the entire working area. As a result, any sensor can potentially evolve characteristics that would allow it to detect all radar in the working area. Short-range sensors have a maximum sensing range that covers at most one quarter of the environment. We expect solutions with short range sensors to consist of more sensors due to their comparatively limited capabilities.

Figure 3 shows an example of a candidate solution. The pie shaped elements indicate sensors and their detection angle and orientation. Lines indicate detection of a radar by a sensor.

III. GENETIC ALGORITHM DETAILS

The GA [4], [5] is a learning algorithm based on principles from genetics and evolutionary biology. Where nature evolves organisms that meet the requirements necessary for survival in a particular environment, GAs evolve solutions that meet the requirements necessary for solving specific problems. A typical GA works with a population of individuals, where each individual represents a potential solution to the problem to be solved. These potential solutions are evaluated and the better solutions are used to create a new population of potential solutions using genetics-inspired operators. Over multiple "generations", the quality of the evolved solutions will improve.

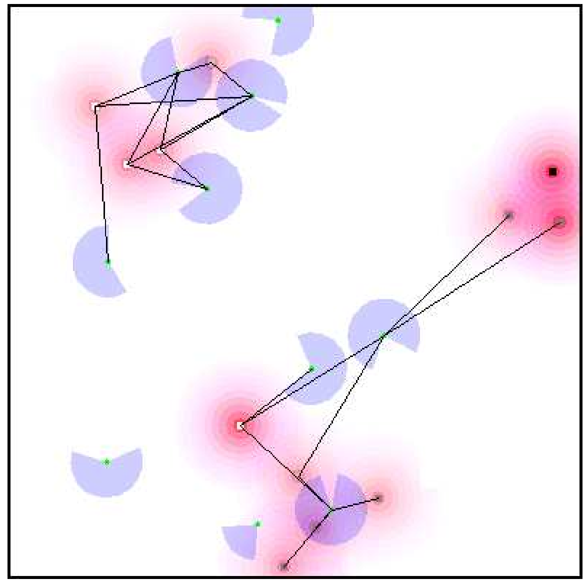


Fig. 3. Problem environment with candidate solution.

```
procedure GA
   {
   initialize population;
   while termination condition not
         satisfied do
      {
      evaluate current population;
      select parents;
      apply genetic operators to parents
         to create offspring;
      set current population equal to
         the new offspring population;
      }
   }
```

Fig. 4. Basic steps of a typical genetic algorithm.

Key features of a GA include the following. A GA works with a population of individuals where each individual represents a potential solution to the problem to be solved. Idealized genetic operators explore the search space by forming new solutions out of existing ones. Genetic operators define how encoded information is manipulated and changed by a GA. A selection function selects individuals for reproduction based on their fitness. Selection exploits useful information currently existing in a population. A fitness function evaluates the utility of each individual as a solution.

Figure 4 shows the basic steps of a GA. The initial population may be initialized randomly or with user-defined individuals. The GA then iterates thru an evaluate-select-

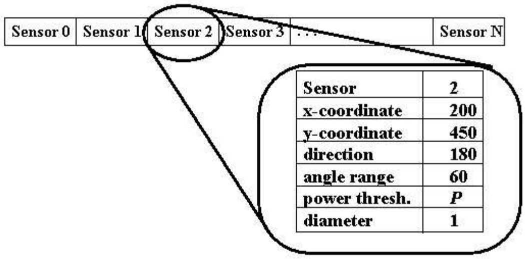


Fig. 5. Problem representation for a team of sensors.

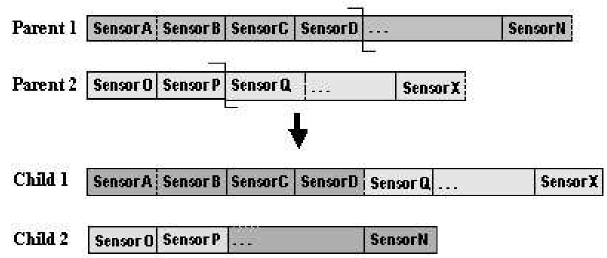


Fig. 6. Inter-gene level crossover operation.

reproduce cycle until either a user-defined stopping condition is satisfied or the maximum number of allowed generations is exceeded.

A. 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 5 shows an example individual which represents a team of N+1 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.

Multiple sensors in an individual may have the same location in the environment. When that occurs, the first (leftmost) sensor at a given location is active. The remaining are inactive and are unable to detect any radar; however, all sensors are included in the cost component of the fitness function.

B. Fitness evaluation

The fitness of each candidate solution generated by the GA is evaluated by inserting the solution (sensor team) in the test problem simulation and evaluating its performance within the simulation. Obstacles are not directly factored into the fitness evaluation. They indirectly affect fitness evaluation because an obstacle that intersects the direct line between a sensor and a radar will prevent that sensor from detecting the corresponding radar.

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

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

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

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

where R is the total number of radar and F, P, M are the numbers of fully, partially and minimally detected radar, respectively. Partially and minimally detected radar contribute less to the fitness evaluation than fully detected radar.

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 sensors:

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

where b is the fixed basic cost of the deployment, n is the total number of sensors, p_i is the basic cost of each sensor i, and q_i is the cost of the sensor frequency ranges.

C. Selection and Genetic Operators

We use deterministic tournament selection with tournament size two, one-point variable length crossover, and a problem specific mutation operator.

The crossover rate indicates the probability that two selected parents will undergo crossover. Parents that do not undergo crossover are copied unchanged into offspring. Crossover points are selected independently on each parent; consequently, the length of an offspring may be different from its parents. Crossover points always fall in between the genes as shown in Figure 6.

Mutation occurs at the intra-gene level. Each characteristic of each gene is subject to mutation at the given mutation rate. Sensor characteristics such as location, orientation, detection angle, and power threshold mutate using a Poisson distribution function which generates an offset from the original value. As a result, mutation is likely to generate values that are similar to the original value rather than simply mutating randomly to any new value. We expect this mutation scheme to encourage accurate adjustment of sensor characteristics.

We use two additional operators called insertion and deletion mutation. Insertion mutation inserts into a sensor suite a new sensor with randomly initialized random characteristics with probability given by the insertion mutation rate. Deletion mutation randomly selects a sensor to remove from a sensor suite with probability given by the deletion mutation rate.

IV. Experimental results

We test our algorithm on two radar configurations. In the *grid* configuration, enemy radar are laid out in a grid

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

TABLE I
GA PARAMETER SETTINGS USED.

and cover almost the entire working area. This configuration tests the algorithm's ability to evolve solutions that provide maximum coverage of the working area. In the *cluster* configuration, enemy radar are randomly laid out in several clusters. This configuration tests the algorithms ability to focus on specific areas of the working area.

Table I gives the GA parameter settings used in our experiments. These values were selected based on performance in previous experiments.

We begin with the simplest case of one obstacle. A single rectangular obstacle is placed vertically down the middle of the environment, dividing the environment into two regions. An intuitive solution for this problem is to treat the two regions independently, positioning three sensors in each region. Recall that a minimum of three sensors are necessary to fully detect a radar. Figure 7 shows an example solution for the grid configuration. The GA does indeed find a solution with six sensors that can fully detect all radar. Interestingly, however, the sensors do not focus solely on one region; four of the six sensors attempt to straddle both regions. Figure 8 shows the number of sensors evolved and the detection percentage averaged over 100 runs. The number of sensors levels off around seven for the best individual, which balances the minimum cost and the maximum detection. The best individual clearly achieves 100% detection.

We repeat this experiment in the cluster configuration. Figure 9 shows an example solution from the cluster experiments. The GA generates a team of six sensors that can fully detect all radar. Again, some sensors are arranged so that they straddle both halves. Figure 10 shows the average behavior over 100 runs. The number of sensors for the best individual levels off around seven and the detection percentage is 100%.

We test increasing numbers of obstacles in a variety of positions and sizes to increase the difficulty of the problem. Figure 11 shows some example results from two, three, and multiple obstacles experiments on both the grid and cluster configurations. In all cases, the GA is able to find optimal or near-optimal solutions in which all or almost all radar are detected by at least three sensors.

The most striking feature of these example results is how the GA minimizes the team size by consistently attempting to arrange sensors close to the ends of the obstacles where they are more likely to be able to sense on both sides of an obstacle. In the two obstacle scenarios, sensors are arranged to take advantage of the small gap between the obstacles. As the number of obstacles increases, sensors are still arranged at locations where most can take advantage of a near-360 degree detection range. Whereas the teams evolved in obstable free environments tended to place sensors centrally within clusters of radar, in these experiments, the GA does occasionally place sensors outside of the radar region to allow a sensors to "reach" around obstacles. In the more dense grid environment, increased team size is unavoidable as the number of obstacles increases. In the more sparse clustered environment, the GA is able to maintain team sizes close to six even in the multiple obstacle scenario.

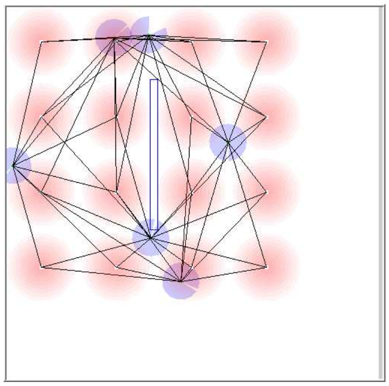
V. Conclusion

We apply a GA to the problem of designing teams of sensors that work together to detect and monitor multiple enemy radar. This problem is an important concern for electronic warfare support to aid in the detection, offensive, and assessment activities of electronic warfare. The GA evolves the count, placement, and characteristics of the sensors of a team. The goal of the GA is to design a team that maximizes the detection percentage while minimizing cost. Previous results indicate that a GA is able to successfully evolve efficient teams that can detect all or almost all radar. In this work, we test the effectiveness of the GA in more complex environments that include obstacles that can limit the detection capabilities of sensors. The sizes, locations, and the number of the obstacles affect the solutions generated by the GA. Although the detection percentage is robust to environmental changes, in terms of both the obstacle and radar configurations, the size of the evolved teams tends to increase with increasing size and number of obstacles.

Emergent strategies of how the GA arranges sensors are interesting. The current fitness function does not penalize for large detection angles. The GA takes advantage of this lack by favoring sensors with large detection angles. With no obstacles, the GA attempts to place sensors close to the center of all radar. This placement in combination with large detection angles maximizes the number of radar that a single sensor can detect. When there are obstacles in the environment, the GA either places sensors close to the center of a group of radar or at the corners of obstacles which allow the sensors to work on both sides of an obstacle. Both strategies are logical approaches to maximizing the efficiency of a sensor.

Acknowledgements

This work is sponsored by the Naval Research Laboratory, ITT Industries Incorporated, and the National Science Foundation.



Generation:	427
Best Individual's Length:	6
Average Length of Population:	8.95
Avg Length Standard Deviation:	3.652054216465029
Best Individual's number of Active Sensors	6
Best individual's Fitness	0.05440638005677908
Average Fitness	0.04585580036374229
Avg Fitness Standard Deviation	0.009073075247137113
Best individual's Cost	57.50343634455908
Average Cost	55.14059190979381
Avg Cost Standard Deviation	3.190726995773635
Number of Enemy Radars	16
Number of Enemy Radar detected by Best Indiv...	16

Fig. 7. Example solution for experiments using long range sensors for the grid configuration with one obstacle.

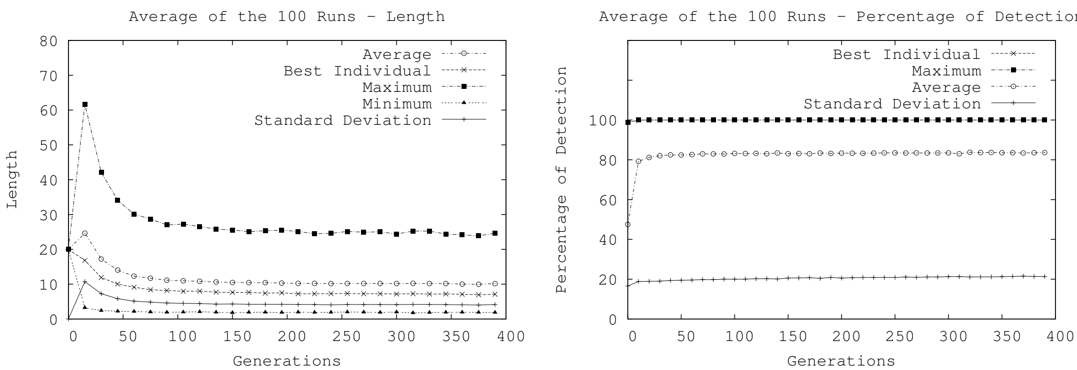
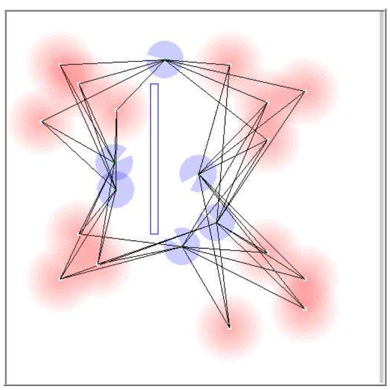


Fig. 8. Length (Number of sensors) and percentage of detection averaged over 100 runs for long range sensors in the grid configuration with one obstacle.



Generation:	413
Best Individual's Length:	6
Average Length of Population:	10.115
Avg Length Standard Deviation:	3.9334177250833675
Best Individual's number of Active Sensors	6
Best individual's Fitness	0.05422999735844566
Average Fitness	0.04622011225458004
Avg Fitness Standard Deviation	0.006178420441716242
Best individual's Cost	58.634501024279714
Average Cost	55.319936310725026
Avg Cost Standard Deviation	3.3204132476409
Number of Enemy Radars	15
Number of Enemy Radar detected by Best Indiv...	15

Fig. 9. Example solutions for experiments using long range sensors for the cluster configuration with one obstacle.

REFERENCES

[1] Karthik Balakrishnan and Vasant Honavar. On sensor evolution in robotics. In *Proc. 1996 Genetic Programming Conference*, 1996.

[2] M.D. Bugajska and A.C. 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.

[3] M.D. Bugajska and A.C. Schultz. Co-evolution of form and function in the design of micro air vehicles. In *NASA/DoD Conference on Evolvable HW*, 2002.

[4] D. E. Goldberg. *Genetic algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[5] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.

[6] Ayse S. Yilmaz, Brian N. McQuay, Han Yu, Annie S. Wu, and John C. Sciortino, Jr. Evolving sensor suites for enemy radar detection. In *Genetic and Evolutionary Computation Conference - GECCO 2003*, volume 2724, pages 2384 – 2395. Springer Verlag, Berlin, 2003.

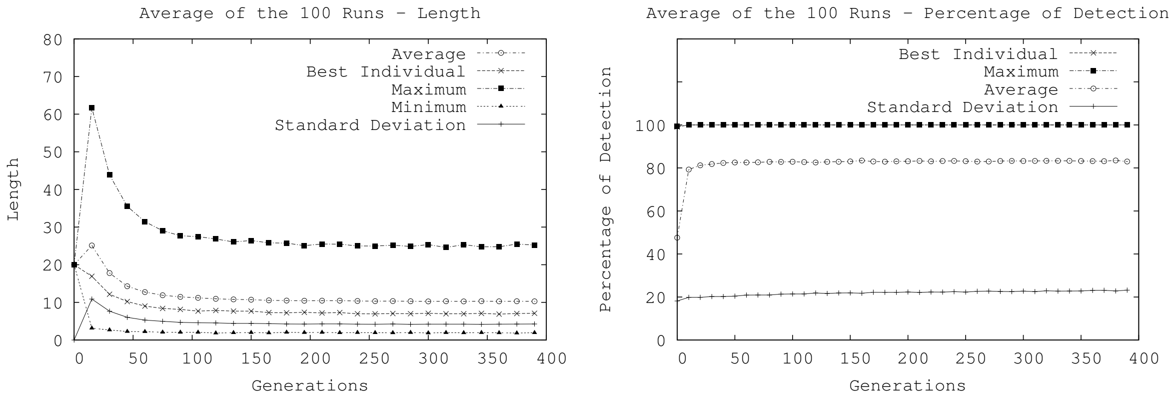
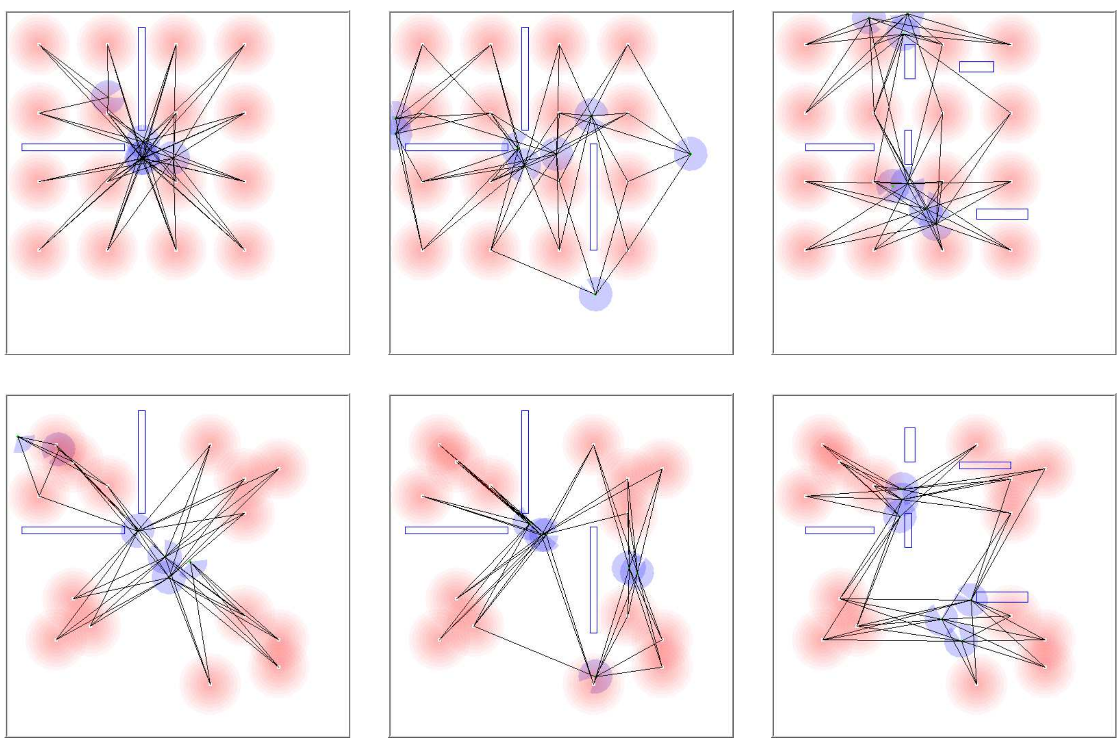


Fig. 10. Length (Number of sensors) and percentage of detection averaged over 100 runs for long range sensors in the cluster configuration with one obstacle.



Two obstacles **Three obstacles** **Multiple obstacles**

Fig. 11. Example results from experimental scenarios containing multiple obstacles.