

# Evolutionary Approaches for Resilient Surveillance Management

Ruidan Li and Errin W. Fulp  
 Department of Computer Science  
 Wake Forest University  
 Winston-Salem, NC 27109, USA  
 Email: fulp@wfu.edu

**Abstract**—The efficient management (placement and orientation) of security cameras within a floor plan is a well-known and difficult problem that has gained attention recently. The objective is to locate the minimum number of cameras in the space to ensure all walls are within the view of at least one camera. Heuristic-based approaches have been developed for this  $\mathcal{NP}$ -hard problem; unfortunately, most are only applicable to static situations. In modern applications, surveillance management must be resilient, and adapt if the environment changes.

This paper introduces evolutionary-based approaches for active surveillance camera management. Using an evolutionary-based approach, a surveillance configuration (camera locations and orientations) is encoded as a chromosome and evolutionary processes are applied to identify better solutions over successive generations. The approach has the ability to identify efficient surveillance configurations (minimum number of cameras with maximum coverage); however, another advantage is the ability to adapt if the environment unexpectedly changes. Simulation results demonstrate this type of approach can manage surveillance cameras under dynamic conditions such as camera loss and the introduction obstacles better than traditional search methods.

**Index Terms**—surveillance systems; security; cameras; resiliency; art gallery problem; genetic algorithm;

## I. INTRODUCTION

The availability of low cost cameras and the growing demand for surveillance applications has renewed the interest in how to best manage camera-based surveillance systems [9]. For example surveillance systems can be used to improve the security of an industrial complex or to detect accidents within a manufacturing site [9]. Given the low cost of cameras there are possibly thousands of cameras available at a site; however, only a small set of cameras is actually needed to view (cover) the locations of interest. The management objective is to maximize the coverage (ensure locations of interest are under surveillance) while minimizing the number of cameras in use. Coverage is important to achieve the operational goal of the system (e.g. deter crime), while minimizing the number of cameras reduces energy consumption, which is especially important if batteries are in use.

The problem of determining the location and orientation of cameras to cover a polygon space was first introduced by Victor Klee in 1973 [8]. This original problem was known as the Art Gallery Problem (AGP) and sought to locate guards (cameras) in an art gallery such that every interior wall was observed by at least one guard. Algorithms exist

to identify camera locations and orientations (cameras are typically located at the polygon interior vertices); however, finding the minimum number of cameras to provide coverage has been proven to be  $\mathcal{NP}$ -hard [7]. Since AGP was originally introduced, several variations of the problem have been proposed. Most are equally as difficult as the original, but are perhaps more applicable to a realistic situation. Given the proven difficulty of AGP, several heuristic-based approaches have been proposed. For example in [1], the floor plan, which is known in advance, is partitioned into smaller polygons and cameras are located within these smaller pieces. While many of these approaches provide good solutions to AGP-type problems, most assume the environment is static. In more modern application, it is expected that cameras will be added or removed for various reasons (e.g. unexpected failures). In addition, obstacles may appear and disappear at random times (e.g. cart moving through a warehouse). As a result, these approaches must recalculate solutions based on the new environment, which may be computationally expensive.

This paper investigates the use of evolutionary algorithms for managing cameras within a polygon space. An evolutionary approach can identify good solutions and adapt if the environment changes. Assume a set of cameras is located, perhaps randomly, within a polygon space. The location of each camera is fixed, but each can swivel 360 degrees to point towards any direction. Therefore this AGP variation seeks to determine the minimum set of cameras to use (turn on or off) and their orientation (swivel position) to maximize the wall cover. The approach encodes the on/off and swivel for each camera as a chromosome, then applies a series of evolutionary processes to find better solutions over successive generations (iterations). Simulation results with 40-sided polygons show the approach can identify good solutions under static and dynamic conditions. The continual searching nature of the approach allows the identification of good solutions if cameras are added or removed, as well as if obstacles are introduced in the space.

The remainder of this paper is structured as follows. Section II discusses surveillance management problems and the specific problem variation addressed in this paper. Evolutionary algorithms, fitness, and processes are reviewed in Section III. Simulation results of dynamic surveillance environments are

discussed in Section IV, while Section V reviews the paper and discusses some future areas of research.

## II. SURVEILLANCE MANAGEMENT

Surveillance management seeks to identify the location and orientation of cameras within a space such that all interested/targeted areas within that space are always observed by at least one camera. As described in the Introduction, this is similar to the Art Gallery Problem (AGP) which sought to place guards within a floor of a museum to ensure all walls are watched. The objective, which can be difficult to achieve, is to find the minimum number of cameras required for coverage (observe the targeted areas). The number of cameras that is always sufficient has been loosely bounded; however, this is not necessarily the minimum number [8].

For this paper, assume the polygonal space is initially populated with cameras, where each camera has the same view angle  $\alpha$ . The camera locations can be determined via an algorithm (based on a grid layout) or be random within the space as done with smart dust devices [3]. Although the camera locations are predetermined, every camera has the ability to swivel 360 degrees about their location. Let the specific orientation angle for camera  $i$  will be  $\beta_i$ . Furthermore, each camera has a binary activation state  $a_i$  that indicates if the camera is turned on or off; therefore, each camera  $i$  has two configuration settings  $(a_i, \beta_i)$ . A surveillance configuration  $s$  is then a list of camera settings, one per camera, within the polygon space. In the example given in Figure 1, the surveillance configuration for the three cameras  $A$ ,  $B$ , and  $C$  would be  $s = \{(1, \beta_A), (0, \beta_B), (1, \beta_C)\}$ . Note camera  $B$  is not activated in this example.

### A. Surveillance Objectives

The objective of this problem is to find surveillance configurations that maximizes the wall coverage using the minimum number of cameras, and as a result is considered a multi-objective problem [2]. The management approach must determine which cameras to activate ( $a_i$  state) and their orientations ( $\beta_i$  angle). This problem is similar to the AGP variant called the Floodlight Set Problem (FSP) [8], where floodlights are positioned to ensure the maximum wall space is covered. The additional requirement for proper surveillance management considered in this paper is to constantly maintain maximum coverage using the minimum cameras as the environment changes.

As described in the Introduction, a surveillance management approach must contend with the loss or addition of cameras. The loss could be the result of battery outages or network disruption, while an addition could be the installation of cameras to improve coverage or to provide redundancy. The other environment change is the introduction of obstacles (an  $m$ -sided polygon) within the space. An obstacle can be considered a hole, which means the obstacle itself is not to be monitored. For this paper the obstacle must also be covered along with the walls of the space. As with the addition and removal of cameras, a camera management approach must

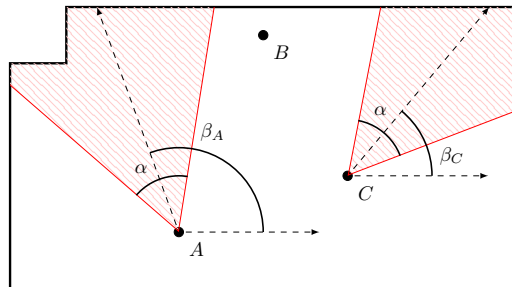


Fig. 1. Example 6-sided polygon floor space with three cameras (labeled  $A$ ,  $B$ , and  $C$ ). Each camera has the same view angle  $\alpha$  and an individual orientation (swivel) angle  $\beta_i$ . In this example cameras  $A$  and  $C$  are activated (turned-on), while camera  $B$  is not activated (turned-off).

contend with the introduction and removal of obstacles over time.

## III. AN EVOLUTIONARY APPROACH TO SURVEILLANCE MANAGEMENT

Identifying good surveillance configurations (camera activations and orientations) can be considered a search problem that attempts to locate configurations that maximize coverage with the smallest number of cameras. Given the size and complexity of the search space, search heuristics, such as Evolutionary Algorithms (EAs), are often used for this type of problem. In addition, EAs have the benefit of constantly searching for solutions. This search characteristic is helpful for problems where the search space may dynamically change [6].

EAs naively mimic evolution to find better (more fit) surveillance configurations by discovering, recombining, and altering portions of current configurations to generate new ones. This is achieved by maintaining a set of solutions (referred to as a pool) rather than a single solution. Before an EA can be applied surveillance management, a genetic representation of the problem domain, methods of determining feasibility, an understanding of configuration fitness, and the design of EA operators must be carefully addressed.

### A. Camera Configurations and Fitness

EAs represent potential solutions as a chromosome consisting of multiple traits, or parts of the solution. As described in Section II, each camera has two settings  $(a_i, \beta_i)$ . The first setting is a binary value indicating if the camera is active or inactive (or on or off) and the second is the orientation angle. A surveillance configuration  $s$  is then a list of camera settings, one per camera. Using the chromosome representation, the settings for a specific camera are a trait or gene, while the surveillance configuration is a chromosome.

A measure of fitness is also important for evolutionary algorithms to ensure fitter chromosomes are more likely to survive and influence the next generation. For surveillance management, the fitness of a chromosome (surveillance configuration) is multi-objective since the approach seeks the maximum coverage using the fewest cameras [8]. Several

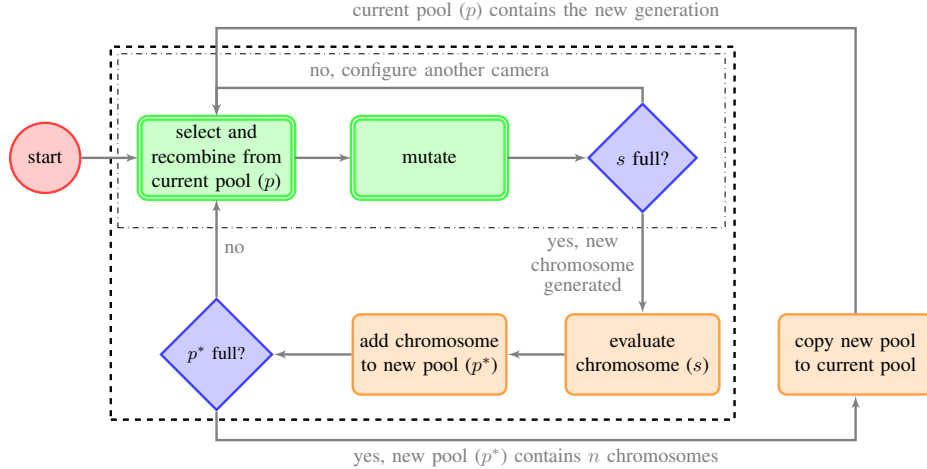


Fig. 2. Flow chart of the tasks used for identifying surveillance configurations. Tasks within the outermost dashed box create a new chromosome pool ( $p$  or  $p^*$ ) per generation, while tasks within the innermost dashed-dotted box create one configuration ( $s$ ).

approaches exist for measuring the goodness of multi-objective problems, for example the simplest and perhaps most widely used approach is the weighted sum method. This approach scalarizes the objectives into a single objective by multiplying each objective with a specified weight. This is more easily done if the objectives are all maximizations or minimizations [5]. Therefore the surveillance problem objectives would become maximizing coverage while maximizing the number of cameras not in use. Weights are chosen in proportion to the relative importance of each objective. An issue with weighted sum is that not all multi-objective problems can be easily scaled (no clear trade-off for improving one objective at the expense of another). Therefore problems can occur if the optimal solution distribution is not uniform and optimal solutions in non-convex regions are not detected [5]. These issues were observed in surveillance management using simulation.

An alternative multi-objective fitness measure is based on the Pareto Front. Essentially, a solution (chromosome or surveillance configuration) is said to be strictly Pareto optimal for a multi-objective problem if it is not dominated by all other solutions (currently considered) [5]. A surveillance configuration  $s_i$  said to dominate the other solution  $s_j$ , if both the following conditions are true. First, the solution  $s_i$  is no worse than  $s_j$  in all objectives. Second, the solution  $s_i$  is strictly better than  $s_j$ . All the points that are not dominated by any other points in the space together form the Pareto Front in the current space. The points in the front are then removed and the front of the rest of the points can be found. Using this approach, solutions in one generation can be separated into different levels.

For example, consider five different surveillance configurations for the same polygonal space. Each configuration has two performance measures, the coverage percentage and the percentage of inactive cameras. Recall the objective is to maximize both percentages. Assume the performance values

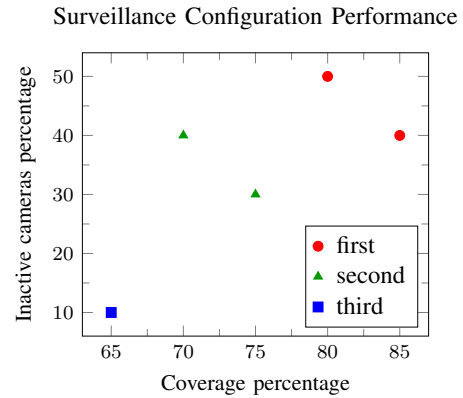


Fig. 3. Example Pareto ranking of five different surveillance configurations. Better performance occurs with higher coverage and inactive cameras percentages. The first group performs the best, second is second best, and the third group ranks the worst.

for the surveillance configurations are (80%, 50%), (65%, 10%), (75%, 30%), (85%, 40%), and (70%, 40%), where the first number is the coverage percentage and the second is the percentage of inactive cameras. For this example the Pareto rankings are depicted in Figure 3, where configurations closer to the upper righthand corner are considered better. The first level contains (80%, 50%) and (85%, 40%) as there are no other solutions that are strictly better than either (having better coverage with more inactive cameras). The solutions in the second level are (75%, 30%) and (70%, 40%) since both of them are dominated by the configurations in the first level. The last level (worst surveillance configuration) is (65%, 10%). These rankings can then be used to identify better configurations.

For this application, a combination of weighted sum and Pareto ranking was used to determine the fitness of configurations (chromosomes). A variation of the weighted sum method, which can change based on the objectives, was used to compute an initial fitness value per configuration. The average distance between Pareto levels, which contains the least wanted solutions, was then used to create relative weights. The weight for each level was then multiplied with the initial fitness value to create the final fitness. As a result, solutions are clustered by similar within-level fitness values and different between-group fitness values.

### B. Evolutionary Processes

As mentioned at the beginning of this section, an EA progresses by updating, and hopefully improving, a set of solutions called the chromosome pool. A new pool of surveillance configurations is created for each generation based on the previous generation using a series of reproduction, recombination, and mutation processes (mimicking processes observed in nature) [4]. The set of task used to find surveillance configurations is depicted in Figure 2.

Consider generating one new surveillance configuration that would initially contain an empty activation state and orientation angle per camera. For each camera in the new configuration, a configuration is selected from the current pool. This is done using roulette selection, where the fitness level of the chromosome is used to associate a probability of selection. In this case weights are assigned so more fit (higher performing) configurations will be more likely selected. The corresponding camera settings in the selected configuration are then copied to the new configuration. This type of selection also incorporates the second evolutionary process call recombination (also known as crossover in the Genetic Algorithm literature). Recombination combines portions of existing configurations to create a new configuration. Traditional recombination only occurs with two selected chromosomes; however, the process used in this paper allows the possibility of recombination using any of the current configurations. It was observed experimentally that this approach increased the breadth of the search, which provides a means to escape from local minima or maxima.

The last evolutionary process applied to the new surveillance configuration is mutation. Mutation provides the ability to explore new regions of the problem space by randomly changing camera settings in the offspring created from the recombination process. The purpose of mutation is to maintain diversity across the generations of configurations. Given the new chromosome, each camera setting will be mutated with a certain probability. If mutation occurs then the camera setting is randomly modified using a uniform distribution (done for the activation and the orientation angle). This series of selection, recombination, and mutation processes are repeated until a new pool (set) of surveillance configurations has been generated, as depicted in Figure 2.

### C. Search Refinement Using Beam Search

As previously described, heuristic search algorithms that maintain multiple solutions, for example EAs, are often well suited for complex search spaces. Beam search is another heuristic search method that maintains a population of solutions [10]. The algorithm progresses by exploring only the best solutions in the current generation (the top  $x\%$  referred to as the beam width). These best solutions are mutated to produce the next generation. This normally repeats over multiple generations until a goal is achieved, for example population convergence. Given this design, beam search has the ability to focus on a certain area of the search space; however, it can ignore other solutions or solutions that may be useful if the environment changes. In contrast, EAs can provide breadth that is useful for changing environments, but may have limited depth.

It is possible to combine an EA with beam search to provide the advantages of both techniques. This hybrid approach can be achieved on a generation basis, where the EA is employed for multiple successive generations followed by multiple successive generation of beam search. The best number of generations for EA and beam can be determined empirically. The hope is the EA will find good potential solutions and beam will refine these solutions. Given both approaches favor good solutions, better solutions should exist across generations.

## IV. EXPERIMENTAL RESULTS

In this section simulation is used to compare surveillance management using an EA, beam search, and an EA beam search hybrid (described in Section III-C). Note, the hybrid approach performed five EA iterations followed by two beam search iterations. The population size each search algorithm was 200, the mutation probability for the EA approaches was 3%, while beam search width was 25%.

Experiments simulated the effect of adding and removing cameras, as well as the introduction of obstacles within the floor space. Ten different 40-sided orthogonal polygons were used as the floor plans for experiments. The camera view angle was 90 degrees. Each floor space initially contained 80 cameras, where 10 different random camera location layouts were used. Each camera layout was simulated twice and the average results for these 200 simulations were then recorded. An example 40-sided polygon, camera locations, and obstacles are given in Figure 4. Performance was measured as the coverage percentage (percentage of the wall space covered by at least one camera) and the percentage inactive cameras (based on the number of cameras available). In both cases higher values are preferred.

### A. Dropping and Adding Cameras

The camera drop/add experiments consisted of eight consecutive events; four drop events followed by four add events. Events were space 10 generations apart and started at generation 300. Waiting until 300 generations was done to ensure the algorithms had sufficient time to find an initial surveillance

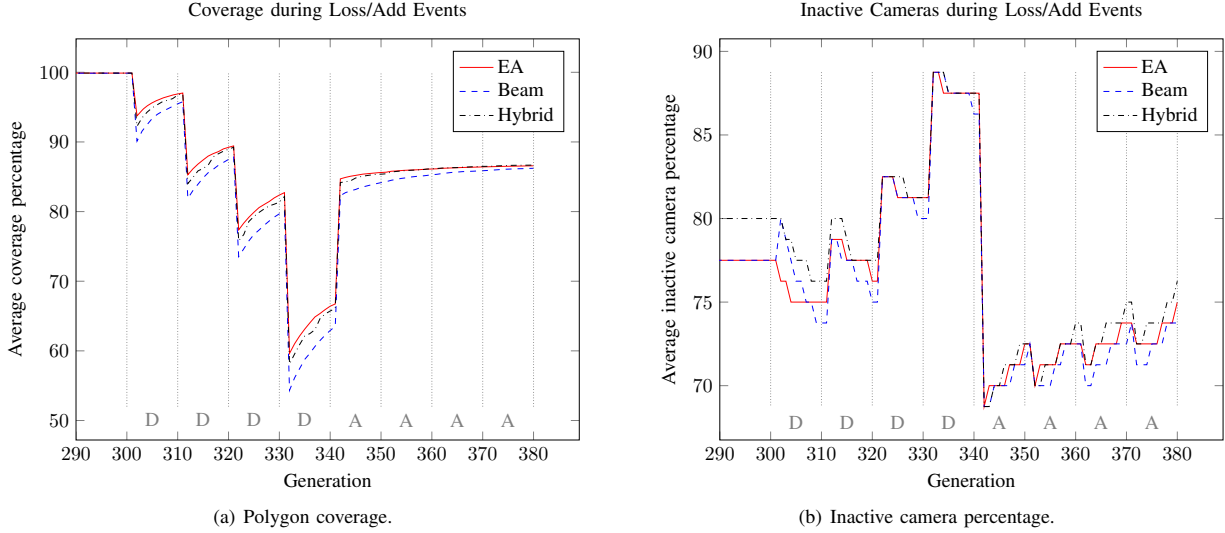


Fig. 5. The effect of dropping and adding cameras (four 20% drops followed by four 20% adds, denoted in the graph as “D” or “A” respectively) within 40-sided orthogonal polygons. Drop and add events occurred every 10 generations starting at generation 300.

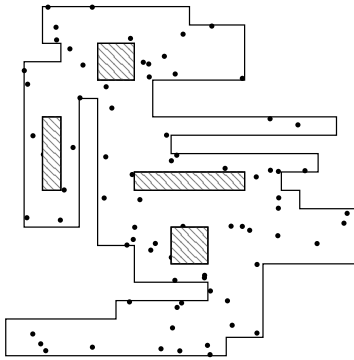


Fig. 4. An example 40-sided orthogonal polygon used as a floor plan for simulations. Circles represent one possible set of initial camera locations and gray areas are possible obstacles.

configuration using any of the 80 cameras. Each drop event randomly reduced the number of cameras in the space by 20%, while each add event allowed 20% more cameras to be active.

Figure 5 depicts the coverage and inactive camera percentage once the events begin. All algorithms start with perfect coverage; however, the EA and beam search have a lower inactive percentages (uses more cameras for the same coverage as the hybrid). As each drop event occurs, the coverage immediately declines. All algorithms also have a drop in the inactive percentage as the new configurations are being determined. Note, beam search does experience a temporarily high inactive percentage, but the coverage is lower. The evolutionary approaches (EA and hybrid) are able to recover better, providing higher coverage with fewer cameras. During the drop events beam search averaged 80% coverage

and inactive cameras, while the EA averaged 83% coverage and 80% inactive cameras. The hybrid search averaged 82% coverage and 81% inactive cameras. During the add events coverage was 85% for the evolutionary approaches and 84% for beam search. Hybrids also had higher inactive camera percentages, 73% as compared to 72% for beam search. The continual search characteristic of evolutionary approaches provided better management of camera dynamics.

### B. Adding Obstacles

Obstacle experiments introduced four 4-sided obstacles into the 40-sided polygons, as depicted in Figure 4. The obstacles appear at generation 300 and the objective required the obstacles sides and the interior of the polygon floor plan to be covered. Therefore, the introduction of the obstacles increased the overall surface to cover.

The coverage and percent inactive cameras are depicted in Figure 6. Prior to the introduction of the obstacles, all three algorithms were able to identify configurations that achieved approximately 99% coverage; however the hybrid approach had a higher inactive camera percentage. Beam search had the next higher percentage, but after 250 generations the EA inactive percentage was equal to beam search. Once the obstacles are introduced all three approaches have lower coverage and inactive percentages. Coverage improves at a similar rate for all three methods; however, the evolutionary approaches maintain higher inactive percentages, as seen in Figure 6(b). All three methods have an average 97% coverage after the obstacles are introduced. The hybrid approach has 63% of the cameras inactive, the EA has 61% while beam search is 58%. The largest difference occurs around 350 generations where beam search has 52% inactive and the evolutionary approaches have 60% inactive. Evolutionary approaches were

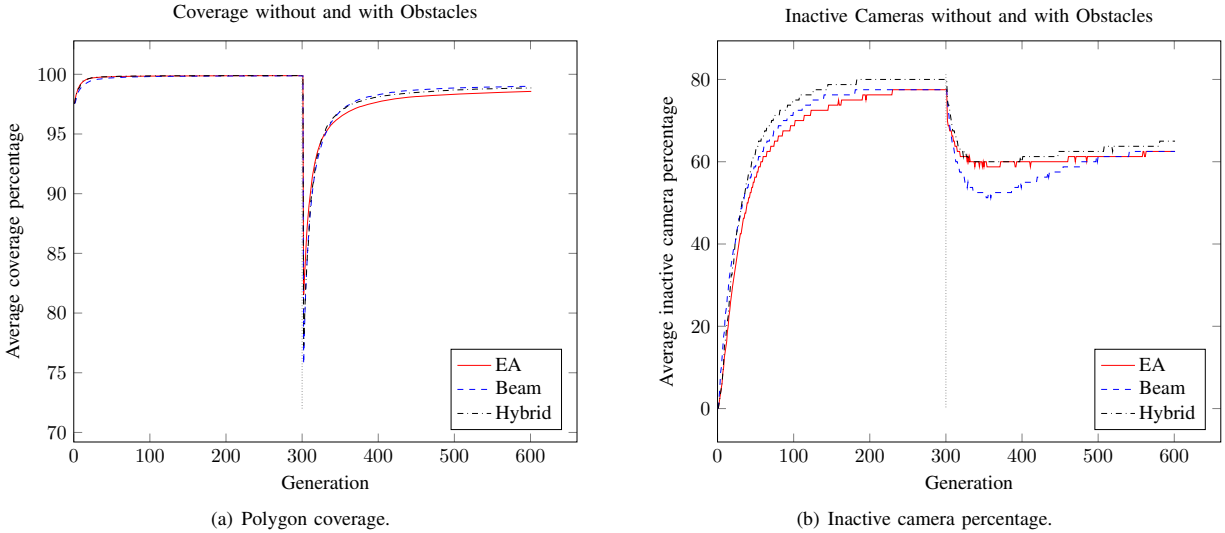


Fig. 6. The effect of adding four polygon obstacles within 40-sided orthogonal polygons at generation 300.

able to manage the introduction of obstacles better (using fewer cameras) than a beam search.

## V. CONCLUSIONS

The availability of cameras and the growing need for security has resulted in more interest in surveillance systems. For these types of systems the objective is often to determine the minimum number of cameras and orientation (direction the camera is facing) required to cover the interior walls. Unfortunately determining the best surveillance configuration is a difficult problem that is only more difficult if the environment changes. Surveillance system must contend with the availability of cameras changing (perhaps due to battery power) as well as the introduction of obstacles in the floor space.

This paper introduced evolutionary-based approaches for identifying surveillance configurations that maximize wall coverage with the fewest number of cameras. Using this approach, the activation states (on or off) and orientation angles for the cameras are encoded as a chromosome and a series of evolutionary processes are applied. Surveillance configurations are then identified that improve coverage and minimize the number of cameras required. In addition, evolutionary-based approaches can adapt if the search space changes. In the case of surveillance, evolutionary-based management approaches can quickly adjust if the number of available cameras change or if obstacles are introduced. Simulation of multiple 40-sided polygon floor plans indicates the evolutionary approaches were consistently better than a more traditional search based approach.

The experimental results indicate evolutionary-based surveillance management has promise, since the technique was capable of identifying efficient and effective surveillance configurations. However, there are several additional related

research areas that warrant addition investigation. For example, gaining better insight to the best combination of evolutionary and beam search would be helpful for deployment. The combined effect of camera additions and losses along with obstacles would also be of interest. Finally future work is needed to understand how these types of evolutionary-based approaches could be made into a distributed management system, where each camera acts individually based on global feedback.

## REFERENCES

- [1] Yoav Amit, Joseph S. B. Mitchell, and Eli Packer. Locating guards for visibility coverage of polygons. *International Journal of Computational Geometry Applications*, 20(5):601–630, 2010.
- [2] C. R. B. Azevedo and A. F. R. Araújo. Correlation between diversity and hypervolume in evolutionary multiobjective optimization. In *IEEE Congress of Evolutionary Computation (CEC)*, pages 2743–2750, June 2011.
- [3] Ioannis Chatzigiannakis, Sotiris Nikolettseas, and Paul Spirakis. Smart dust protocols for local detection and propagation. In *Proceedings of the second ACM international workshop on Principles of Mobile Computing*, pages 9–16. ACM, 2002.
- [4] Kenneth Alan De Jong. *Evolutionary Computation: A Unified Approach*. MIT Press, 2006.
- [5] Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2001.
- [6] Errin W. Fulp, H. Donald Gage, David J. John, Matthew McNiece, William H. Turkett, and Xin Zhou. An evolutionary strategy for resilient cyber defense. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2015.
- [7] Joseph O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.
- [8] Jorge Urrutia. Art gallery and illumination problems. *Handbook of Computational Geometry*, 1(1):973–1027, 2000.
- [9] Xiaogang Wang. Intelligent multi-camera video surveillance: A review. *Pattern Recognition Letters*, 34(1):3–19, January 2013.
- [10] Weixiong Zhang. *State-Space Search: Algorithms, Complexity, Extensions, and Applications*. Springer-Verlag, Inc., New York, NY, USA, 1999.