Genetic Algorithms for Positioning and Utilizing Sensors in Synthetically Generated Landscapes

Haluk Topcuoglu
Computer Engineering Department
Marmara University
Goztepe, Istanbul, Turkey
haluk@eng.marmara.edu.tr

Murat Ermis
Industrial Engineering Department
Turkish Air Force Academy
Yesilyurt, Istanbul, Turkey
ermis@hho.edu.tr

ABSTRACT

Positioning multiple sensors for acquisition of a a given environment is one of the fundamental research areas in various fields, such as military scouting, computer vision and robotics. In this paper, we propose a framework for locating an configuring a set of given sensors in a synthetically generated terrain with multiple objectives of maximization of visibility of the terrain, maximization of stealth of the sensors and minimization of cost of the sensors. Because of their utility-independent nature, these complementary and conflicting objectives are represented by a multiplicative global utility function based on multi-attribute utility theory. In addition to theoretic foundations, we also present how a Genetic Algorithms can be applied to maximize the global utility function for a given terrain.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods and Search—*Heuristic Methods*; J.7 [**Computers in Other Systems**]: Military

General Terms

Algorithms, Experimentation

Keywords

Sensor optimization, utility theory, genetic algorithms

1. INTRODUCTION

Searching and detection of targets form an important role in the military operations. To detect a target, a search is initiated using sensors. The repeated detection of a target during several scans over several time intervals is the acquisition phase, which is followed by identifying the type and class of the detected target. As there are different types of sensors and they use different techniques to search and detect the targets with different costs, sensor planning is one of the key research area related with military scouting.

Acquisition of an environment by positioning and utilizing of a given set of sensors is one of the fundamental research issues in computer vision and robotics. Automatic selection of sensor viewpoints to view multiple objects and regions in a given 3-D scene is an important research area in computer vision [10, 12]. Similarly, for an autonomous mobile robot navigation, a robot is equipped with a set of sensors which cooperate to detect the obstacles and the free space. Sensor planning in robotics, which is the process of deciding the types, configurations and tasks of the sensors, has been studied with various techniques [8, 9, 11]. A cooperative sensor planning system based on multi-attribute utility theory was presented recently, which unifies research from vision, sensor planning and multi-agent planning [4].

Our main motivation is to develop a framework for locating a set of given fixed number sensors and setting their behavioral parameters (i.e., tilt, heading angle) in a virtually generated landscape with multiple objectives on conflicting attributes: the maximization of visibility of the given landscape, the maximization of the stealth and the minimization of the cost of sensors used. These attributes are utility-independent; and when the multi-attribute utility theory is considered, a multiplicative function over these attributes fits for the global utility function in our study. Therefore, we target to maximize a global or unified utility function that includes the utilities of visibility, stealth and cost. Although a recent work [2] considers the sensor optimization in a virtual environment with the objectives of maximizing the visibility and stealth, it is not based on multi-attribute utility theory and it does not consider the topographic study of the given terrain. The global function presented in their paper is limited by not considering probabilistic calculations of visibility and stealth, and various types of sensor costs that are presented in our study including financial, strategic and placement costs.

As part of the sensor optimization problem, a set of locational and behavioral parameters of sensors should be set appropriately, which generate a very large solution space. As an example of 512 x 512 point landscape by assuming that the heading angle is between 0 and 360 and tilt angle is between 0 and 90, there are up to 360 x 90 x 512 x 512 possibilities for positioning a single sensor. Because of its huge solution space, the enumerative or basic optimization meth-

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.

ods can not be applicable and effective for the multi-sensor optimization problem presented in this paper.

Our framework is based on Genetic Algorithms (GA), which is an evolutionary optimization method inspired by the Darwinian evolutionary process present in nature. GA is a generic method that can be applied to any problem if the feasible solutions of the problem can be represented as strings that correspond to genetic encoding of the solutions. In this technique, good solutions are reached through application of genetic operators on individuals in the generation. Application of the operators and reproduction of individuals within themselves allows the stronger individuals to survive as the generations passed. In this paper, we present how a GA-based technique can be applied for positioning a set of sensors on a virtual landscape by optimizing the given global utility function.

The rest of the paper is organized as follows: Section 2 introduces the the details of sensor and landscape characteristics. In Section 3, we first present an overview to Multiple Attribute Utility (MAU) theory, which is followed by our multi-attribute utility function and its components. Section 4 gives our GA-based framework for solving the sensor optimization problem. Section 5 presents the experimental study and the last section give the conclusion and the future work.

2. BACKGROUND ON LANDSCAPE AND SENSORS

Our sensor optimization framework is based on a synthetically generated landscape. In this section, we first present the landscape generation process which is followed by the attributes of the landscape. Then, the details of topographic study for a given landscape is presented; finally, the locational and behavioral attributes of sensors in our framework are presented.

2.1 Landscape Generation

We developed a fractal landscape generator for creating virtual landscapes to be used in our experiments, which is based on the random mid-point displacement algorithm. In this algorithm, an initial triangle or triangles with no altitudes are generated on the X-Z plane. After initial generation of triangles, corners of the triangles are displaced toward Y-axis randomly. The amount of displacement is proportional to the edge length of the triangle, which determines the characteristic of the landscape. A lower displacement value generates a plain-like landscape and a higher displacement value generates a mountainous landscape.

After the displacement, by connecting mid-point of each edge of a triangle to all others, triangle is divided into four smaller triangles. This process is repeated for the given number of iterations for each triangle in the landscape. Figure 1 gives three landscapes generated with different number of iterations, when the displacement is equal to 200.

2.2 Landscape Attributes

In our study, a generated landscape is represented by a set of triangular polygons. For each polygon PG_i, several attributes are stored in the database; some of them are derived from the generated landscape and some require additional computation. In our framework, the major attributes of a polygon PG_i are:

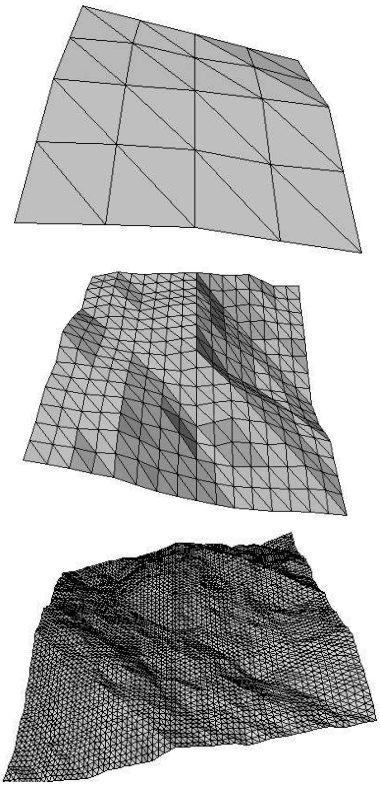

Figure 1: Generated Landscapes with (a) two iterations, (b) four iterations, and (c) six iterations

- Three vertices of the polygon PG_i, (P_1^i, P_2^i, P_3^i)
- Normal vector, N_i, which is required for the ray-tracing operation in visibility and stealth calculations.
- Gravity center of the polygon, G_i. For each polygon, gravity center is computed and stored in the database to avoid excess computing during the program execution.
- Density of the polygon, F_i, which is in the range $0 \leq F_i \leq 1$. It indicates whether the ray traversed from an origin to its destination through the given polygon is blocked by the polygon or not. If the polygon is in a forest area, it has a density of 0.25. Similarly, the density attribute can be used to set climate related issues on visibility, such as fog, snow etc.
- Weight of the polygon, W_i, which indicates the importance of the given polygon. A polygon with higher weight value increases the goodness of solution if it is seen by a sensor in that solution.

Every pixel on a polygon can act as an obstructer during the calculation of visibility of a given sensor. Therefore, the pixels of all polygons are kept in the database for obstruction checking. A scanline algorithm determined all pixels of a given polygon. Since the three corner points (P_0^i, P_1^i, P_2^i) of a polygon PG_i are known, a line is set between any two of these corners (say P_1^i, P_3^i). For every pixel on this line, another straight line is set between this pixel and P_2^i. Then, all pixels on each of these newly generated lines are stored in the database.

2.3 Vehicle Characteristics

Here, the term vehicle stands for various kinds of mobile opponent forces (tanks, trucks etc.) that exist in the given terrain. Each vehicle is associated with a set of access capabilities on a given landscape, which require the topographic analysis for setting the contour lines of all polygons. First, the polygons with the maximum and minimum altitudes are determined and the difference of altitudes is divided by n, a terrain specific parameter for the number of contour lines. Setting the contour line of each polygon gives the elevation map of the terrain, which is used for neighborhood analysis and the computation of locational probability of each vehicle of type k on each polygon i of the landscape, represented as $P_{loc}(k,i)$. Figure 2 gives an example landscape and its corresponding elevation map when six contours are used.

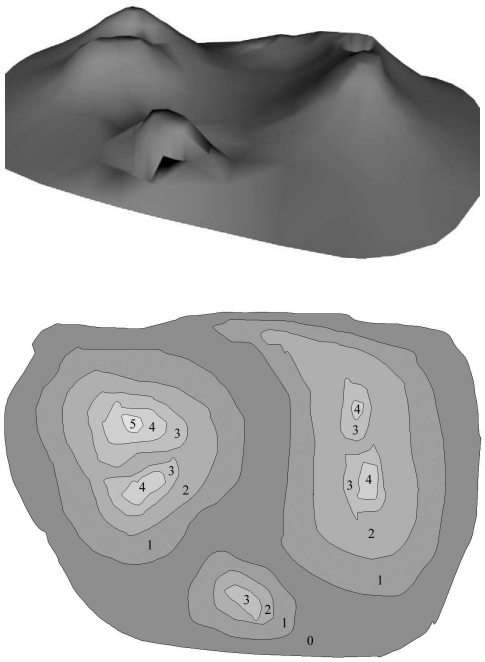

Figure 2: (a) A Synthetically Generated Landscape, and (b) Its Elevation Map with 6 Contours

In our model, each type of vehicle has different access characteristics, such as different maximum slope of climbing; therefore, a separate analysis is required to set each vehicle's locational access probabilities. The first and the main part of the topographic analysis is to determine whether the polygons of the terrain can be reachable from the first contour line, i.e., the lowest altitude contour of the terrain. First, polygons on the second contour line are examined whether they are reachable from the neighborhood polygons either on the first contour line or on the second contour line; then this step is repeated for the remaining contour lines up to the highest contour line. A partial function, G_i, keeps the results of the first part; i.e., it returns 1 if polygon PG_i can be reached from a polygon located on the first contour line; otherwise, it returns 0.

The second part determines the locational probability of each vehicle of type k on each polygon i of the landscape, $P_{loc}(k,i)$, which is computed by the following equation

$$P_{loc}(k,i) = \ln\left(e^{C_k} + \frac{(e - e^{C_k}) \times (\delta_k^v - \delta_i^{PG})}{\delta_k^v}\right) \times G_i \quad (1)$$

where δ_k^v is the maximum slope that the vehicle of type k can climb; δ_i^{PG} is the slope of the polygon i. C_k is a predefined percentage value to represent the capability of the vehicle for climbing its maximum slope. With this equation, probability values are decreased logarithmically with an increase in the slope. As an example, if a vehicle's maximum slope is 40 degrees and if it can climb its maximum slope with a 60% capability, the locational probability of this vehicle on every reachable polygon with a slope of less than or equal to 40 degrees, is in the range of $[0.4, 1.0]$.

2.4 Sensor Attributes

The visibility of a given landscape is done by a set of sensors scattered across the landscape. In general, sensors are classified as acoustic, chemical, electromagnetic and optical [6]. The acoustic sensors are generally used to detect underwater objects. The chemical sensing devices are used to sense and detect the opponents and moving vehicles; and the electromagnetic sensors (such as radar, infrared etc.) are used for detection of surface and air targets.

In our work, to simplify the experiments, we consider sensors of the same class with multiple types. Here, the term sensor stands for a device that is used to observe the landscape which is represented with two sets of attributes: locational attributes and behavioral attributes.

2.4.1 Locational Attributes.

The locational attributes of a sensor S_i are:

- Sensor position (PG_{S_i}). It is the position of the sensor S_i in the landscape in terms of landscape polygon number on which this sensor is located.
- Heading angle (α_i). It is the angle between the looking direction of the sensor carrier and X axis, which is between 0 and 360 degrees.
- Tilt angle (β_i). It is the angle between sensor and its carrier which is in the range of 0 and 90 degrees.

2.4.2 Behavioral Attributes.

The behavioral attributes of a sensor S_i are:

- Type of the sensor ($Type_i$). It is a number used for referencing a sensor with specific viewing parameters.
- Depth of view (Δ_i). It specifies the the range of visibility for the given sensor.
- Viewing angle (Θ_i). The viewing angle indicates the wideness of viewing area of the sensor.
- Cost of the sensor ($Cost^F(i)$). It is the financial cost of the sensor which depends on the sensing ability of a sensor; i.e. a sensor with a high depth of view and a large viewing angle will be more expensive. Formally, it is defined with a function h, $Cost^F(i) = h(\Delta_i, \Theta_i)$.

3. MULTI-ATTRIBUTE UTILITY FUNCTION FOR SENSOR OPTIMIZATION PROBLEM

Our formulation of sensor optimization depends on three conflicting and complementary objectives: maximization of visibility of the given terrain, minimization of visibility of sensors to the opponent objects and minimization of the total cost of the sensors. It is based on Multi-Attribute Utility (MAU) theory, which is one of the major analytical tools associated with the field of decision analysis [3]. A MAU analysis of alternatives explicitly identifies the measures that are used to evaluate the alternatives, and helps to identify those alternatives that perform well on a majority of these measures, with a special emphasis on the measures that are considered to be relatively more important [1]. The first step in a MAU analysis is to determine performance measures and their estimated values with respect to each alternative. Next, for each performance measure a single-attribute utility function is assessed that scales performance between 0 and 1. A MAU function determines how the performance on each measure affects overall performance a set of assessed weights or measures of relative importance. To determine whether a decision maker's preferences satisfy the correct conditions so that we may use the multiplicative utility function to capture her preferences as global utility function, the attributes must be utility-independent [7]. The global utility in our formulation is computed as a multiplicative function over the given three attributes. Formally, the global function to maximize, $U(A,S,P)$, of scanning an area A using a set S of sensors which are located on a set P of polygons (i.e. the sensor S_i located on the polygon P_k), is represented with the following formulation:

$$\begin{aligned} U(A,S,P) \;=\; & w_v\,U_v(A,S,P) + w_s\,U_s(S,P) + w_c\,U_c(S,P) + \\ & w_v\,w_s\,U_v(A,S,P)\,U_s(S,P) \;+ \\ & w_v\,w_c\,U_v(A,S,P)\,U_c(S,P) \;+ \\ & w_s\,w_c\,U_s(S,P)\,U_c(S,P) \;+ \\ & w_v\,w_s\,w_c\,U_v(A,S,P)\,U_s(S,P)\,U_c(S,P) \end{aligned} \quad (2)$$

where $U_v(A,S,P)$ is the utility of visibility of area A by the set of sensors S located on the set of positions P; $U_s(S,P)$ is the utility of stealth of the set of sensors S and $U_c(S,P)$ is the utility of the cost of the sensors. In this equation, w_v, w_s, w_c are the weights (coefficients) of visibility, stealth and cost utility functions, respectively, where $0 \leq w_v, w_s, w_c \leq 1$ and $w_v + w_s + w_c = 1$. These weights are set based on experimentations on a given terrain by considering various military scouting missions. The following subsections briefly describe how the values of the three utility functions are determined.

3.1 Computing the Utility of Visibility

The value of utility of visibility is derived by the calculations of the amount of visibility of the given terrain, which is computed by adding the visibility of all polygons on the given terrain. Here, some of these polygons can be reachable by an object (which is an opponent vehicle); other polygons may not be reached because of the geographic characteristics of these polygons. Formally, the utility of visibility of area A by the set of sensor S (located at set of positions P) is computed using the Equation 3.

$$U_v(A,S,P) = \sum_{PG_i \in A} V_s^m(S,P,PG_i) \times W_{PG_i} \quad (3)$$

Here, W_{PG_i} is the weight of the polygon PG_i computed by using the information generated from the topographic study, which is given at Section 2.3 in detail. In our study, the weight of a polygon depends on the the accessibility of each object to this polygon and the value of information about the given vehicle. We propose the following expression to evaluate the weight of the polygon PG_i:

$$W_{PG_i} = \frac{\sum_k P_{loc}(k,i) \times VOI_k}{\sum_k VOI_k} \quad (4)$$

where VOI_k is the value of information about the object of type k. Important or strategic objects have higher VOI values. In this equation, $P_{loc}(k,i)$ is the locational probability of each object or vehicle of type k on each polygon i, which is evaluated using the Equation 1.

In Equation 3, $V_s^m(S,P,PG_i)$ is the maximum visibility of a given polygon PG_i using the set of sensors S, which is set by the visibility of polygon PG_i from an individual sensor S_j which has the maximum value. Formally,

$$V_s^m(S,P,PG_i) = \max_{S_j \in S} \{V_s(S_j,P_k,PG_i)\} \quad (5)$$

where sensor S_j is located on position P_k. The visibility of the polygon PG_i from the sensor S_j, $V_s(S_j,P_k,PG_i)$, is calculated using basic illumination techniques in computer graphics foundations [5]. Sensors are considered as spot light source and the visibility of a polygon from a sensor depends on the distance between polygon and sensor, the angle between viewing direction of the sensor and the surface normal of the polygon. In our approach, a field of view cone is built for each sensor by using the sensor's depth of view and viewing angle attributes.

Assume that θ_i^j is the angle between the normal of the polygon PG_i and the viewing direction of the sensor S_j. Among the polygons in the field of cone, each polygon whose θ_i^j is not in the range [0..90] or [270..360] is determined and eliminated from the set. Then the visibility computations are done for the remaining polygons in the field of cone. Here, $\cos\theta$ determines whether the face of a polygon is looking toward the sensor or not. Since landscape is a solid object, only a polygon with the normal vector that looks toward to the sensor can be seen by sensor.

A separate line from the sensor to each pixel on the base circle of the view cone is constructed and the pixels on the constructed line are examined and their corresponding polygons are determined. The first pixel on each line, i.e., the closest one to the sensor, is marked *visible* and all other pixels on the same line are marked *obstructed*. Consequently, the percentage of the pixels on a given polygon that are seen by the given sensor is computed.

When these parameters are combined, the visibility of a polygon PG_i from a sensor S_j located at P_k is computed with the following equation:

$$V_s(S_j,P_k,PG_i) = \frac{cos\theta_i^j}{\delta_i^j} \times F_i \times \frac{pixels\ \ seen}{total\ pixels\ on\ PG_i} \quad (6)$$

where δ_i^j is the distance between sensor and the center of the given polygon, which is in terms of number of pixels. It should be noted that $\delta_i^j \leq \Delta_j$, where Δ_j is the depth of

view of the sensor S_j. In this equation, F_i is the density of polygon, which is in the range $0 \leq F_i \leq 1$. This visibility computation of a polygon is repeated for each sensor and the maximum visibility value for the given polygon is considered for the calculating the overall visibility of the landscape (see Equations 3 and 5).

3.2 Determining the Utility of Stealth

The computation of maintaining stealth for a given set of sensors requires two phases: scattering phase and calculation phase. In the first phase, a predefined number of objects (i.e. opponent vehicles) of different types are scattered across the terrain randomly. These vehicles are assumed to carry sensors; and it is also assumed that the sensors on these vehicles have the same behavioral attributes (i.e., viewing angle, depth of view etc.) of the best sensor in our system.

Figure 3 summarizes the steps of scattering m vehicles. The main idea is to use the locational probabilities of vehicles of different types for scattering them on polygons of the given terrain, which is derived from the topographic study. In other words a vehicle of type k is located on a randomly selected polygon PG_i, if the vehicle can reach to the polygon PG_i from a polygon at the first contour line.

1. **for** $j = 1$ to m **do**
2. $k \leftarrow$ select the type of vehicle E_j randomly.
3. $i \leftarrow$ select a polygon randomly.
4. $g \leftarrow random(0, 1)$
5. **if** $g \leq P_{loc}(k, i)$ **then**,
6. locate the vehicle E_j on polygon PG_i
7. **else** goto Step 2.

Figure 3: The Scattering Phase

After scattering, the calculation phase computes the utility of stealth for the given scenario by considering the fixed sensor locations and the vehicle locations which are set at the scattering phase. Formally, the utility of maintaining stealth for a given scenario x, where the set of sensors S are located at set of positions P, is computed with the following equation:

$$U_s^x(S,P) = 1 - \frac{\sum_{S_i \in S} V_e^m(E,P,PG_{S_i}) \times Cost^{STR}(S_i) \times W_{S_i}}{\sum W_{S_i}} \quad (7)$$

where $V_e^m(E,P,PG_{S_i})$ is the maximum visibility of the sensor S_i (located on polygon PG_{S_i}) from the set of opponent vehicles E which are located on set of positions P. As in the sensor visibility, it is computed by:

$$V_e^m(E,P,PG_{S_i}) = \max_{E_j \in E} \{V_e(E_j, P_k, PG_{S_i})\} \quad (8)$$

where opponent vehicle E_j is located on position P_k. The visibility of the sensor S_i from the opponent vehicle E_j, $V_e(E_j, P_k, PG_i)$, is the dual of $V_s(S_j, P_k, PG_i)$; and it is computed with the same equation, Equation 6. To compute the stealth value, sensor visibility value is multiplied with -1 and subtracted from 1, in Equation 7.

In Equation 7, the visibility of each sensor S_i is multiplied by its strategic cost, $Cost^{STR}(S_i)$ and its weight, W_{S_i}. It is due to the fact that a powerful sensor can be located at a non-strategic location or a less powerful sensor can be located on a strategic location on the terrain. Here, the weight of the sensor is computed by multiplying its financial cost with a coefficient q, which is $W_{S_i} = q \times Cost^F(S_i)$.

The strategic cost of the sensor S_i, i.e. the cost of being discovered by an opponent vehicle, is equal to the average visibility of the given terrain using the sensor S_i. When $|PG|$ shows the number of polygons on the terrain, the strategic cost is computed with the following equation:

$$Cost^{STR}(S_i) = \frac{\sum_{PG_i \in PG} V_s(S_i, P_k, PG_i)}{|PG|} \quad (9)$$

The process of scattering and computation phases is repeated for a predefined number of scenarios. The mean value of utilities for the given scenarios determines the overall utility of stealth. If r gives the number of different scenarios, the overall utility of maintaining stealth is computed by:

$$U_s(S,P) = \frac{\sum_{x=1}^{r} U_s^x(S,P)}{r}. \quad (10)$$

3.3 Computing the Utility of Cost

In our study, the term *cost* for a given sensor is considered with the three different meanings: a) the financial cost of sensor, (b) the cost of placement of the given sensor to its location and (c) the strategic cost of the sensor being discovered by an opponent observer. The last one was considered as part of the utility of the stealth: others are considered as part of the utility of the cost.

The utility of cost for a set of sensors S placed on a set of locations P is formally defined as

$$U_c(S,P) = \frac{\sum_{S_i \in S} P_{loc}(i,k) \times Cost^F(S_i)}{\sum_{S_i \in S} Cost^F(S_i)} \quad (11)$$

where $P_{loc}(i,k)$ is the locational probability of sensor S_i on the given polygon PG_k. Here, instead of locational cost, we use locational probabilities. As in the stealth case, the minimization of utility of cost is converted to the maximization form. To compute the locational probability of a given sensor, the Equation 1 is considered by assuming that each sensor is on the carrier which has the same access capabilities with the best opponent vehicle given in the experiments.

4. GA FORMULATION OF THE SENSOR OPTIMIZATION PROBLEM

This section summarizes our GA formulation for the sensor optimization problem. An individual or a solution consists of a set of fixed number of sensors with their attributes. In our string representation, each sensor has values for its location (in terms of the polygon number where the sensor is located), its heading and tilt angles, and the type of the sensor (see Figure 4). If an experimentation is restricted with the fixed number of sensors of the same given type, then the type field is not considered. We consider binary representation for each sensor attribute. In our current implementation, fitness proportional roulette-wheel sampling is considered in a generational-GA approach.

Since our global utility function (given in Equation 2) is in the form of maximization, it is used as the fitness function, without any adjustment. In Equation 2, since all three utility values (visibility, stealth and cost) and the coefficients

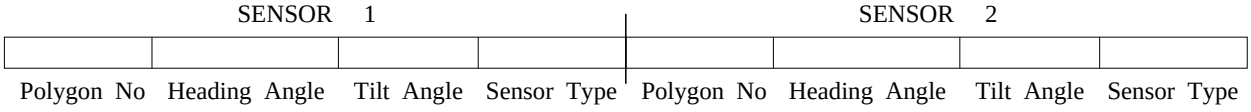

Figure 4: The String Representation for the Problem of Two Sensors

are in the range of [0..1] and $w_v + w_s + w_c = 1$, the fitness function will be in the range of [0..1].

4.1 Operators

4.1.1 Crossover Operator

There are three different strategies considered for crossover operator, which differ according to where the crossover operator is applied.

- TYPE 1: In this version, sensors of parents are considered as blocks (i.e., each sensor is preserved); and an n-point crossover operator is applied on sensors. In other words, a generated offspring consists of sensors from parents by keeping original sensor attributes in their parents.
- TYPE 2: In this type, we consider attributes as blocks, i.e., attributes of sensors are preserved. Then, a binary string is generated with the length of (number of sensors * number of sensor attributes). This string is used to implement uniform crossover by treating each attribute independently and making a random choice as to which parent it should be inherited from.
- TYPE 3: Single point crossover operation is performed on a selected attribute and the same crossover point is considered for all sensor pairs. The generated attribute values of offsprings are different than the attribute values of parents.

After the crossover operation, offsprings' attributes should be in their limits; therefore a validity check on each attribute is performed. An additional check is done for the "polygon number" attribute. The new location (say polygon PG_k) should be reachable for the sensor. Formally, $P_{z,k} \geq \epsilon$, where $P_{z,k}$ is the locational probability of the vehicle z (which carries sensors) on polygon PG_k should be greater than a predefined threshold value, ϵ. If it is not true for any of the offsprings, the crossover operator is applied for the next crossover position (by considering circular representation). If it does not generate a reachable location for the sensor in the three consecutive trials, the parents are discarded and new parents are selected from the current population. In our experiments, although initial population is generated randomly, the validity check on all attributes and the check for reachability are also performed for building the initial population.

4.1.2 Mutation

For the given individual, one of its sensors is chosen randomly and mutation is applied on the selected attribute of the sensor. There are five different types of mutation performed in the experiments: location, heading angle, tilt angle, type, and random. First four are the four attributes of the sensors; and the last one selects the mutation location anywhere on the string. As in the crossover case, the validity check on all attributes and reachability check on location attribute are performed after the mutation.

4.1.3 Local Search Extension

In our work, we consider to improve the sensor's visibility by scanning all possible neighborhood directions. In this approach, firstly, the initial visibility of a sensor is calculated by using the polygon number, type, tilt angle (which had been determined before) and the initial value of the heading angle. Then, the heading angle is increased by the amount of sensor's viewing angle, starting from 0 to 360 degrees. Thus, the local search method scans the circumference of sensor and finds the better visibility.

5. EXPERIMENTAL STUDY

In this section, we present the results of computational experiments to evaluate the effectiveness of our GA-based method for sensor optimization. The GA-based algorithm was coded in C programming language and the landscape generator was implemented in Delphi programming language. The computational experiments of these algorithms were performed by an Intel Pentium IV 1.6 GHz PC running the Linux operating system. The comparison metrics are the utility of visibility, the utility of stealth and the utility of the cost of the sensors.

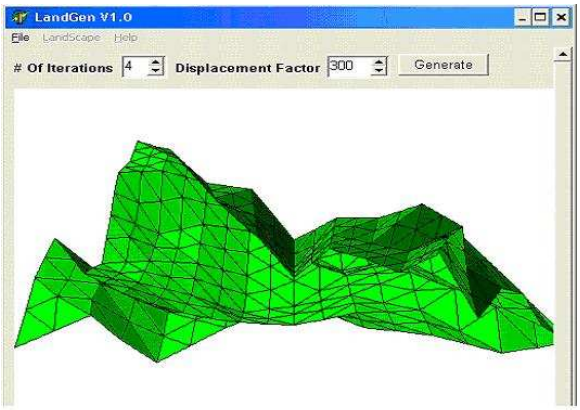

Figure 5: A randomly generated landscape with 4 iterations and a displacement factor of 300

We consider the synthetically generated landscape given in Figure 5 as the reference problem in the experiments. In this section, two separate experiments are performed to demonstrate the effect of number of observation points (i.e., the sensors) and the effect of number of opponents over the utility terms given in the multi-attribute utility function. It should be noted that number of sensors or number of opponents are varied by keeping the fixed values for other experimental and control parameters. Specifically, population size is set to 30 and the opponent distribution number is set to 4. The weight values of visibility, stealth and cost are set with the values 0.6, 0.3, and 0.1, respectively. The "TYPE 3" crossover operator and local search extension for

setting the heading angle are considered in the experiments presented in this section.

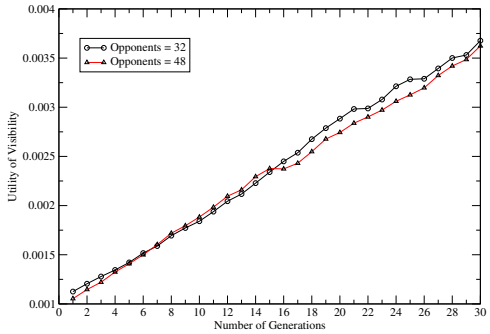

Figure 6: Change in Utility of Visibility with Different Number of Sensors

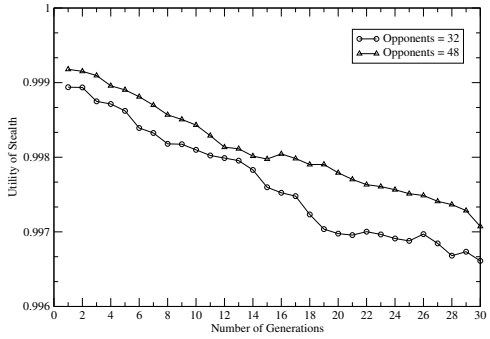

Figure 7: Change in Utility of Stealth with Different Number of Sensors

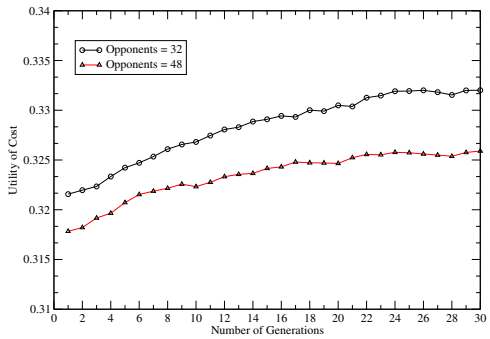

Figure 8: Change in Utility of Cost with Different Number of Sensors

In our first experiment, we vary the number of opponents (either 32 or 48) while fixing the number of sensors to 64. As can be seen from Figures 6-8, both the utility of the visibility and the utility of the cost increase whereas the utility of the stealth decreases as expected, with an increase in the number of generations. This observation is valid for different number of opponents.

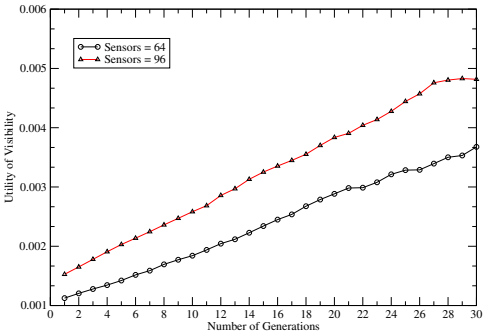

Figure 9: Change in Utility of Visibility with Different Number of Opponents

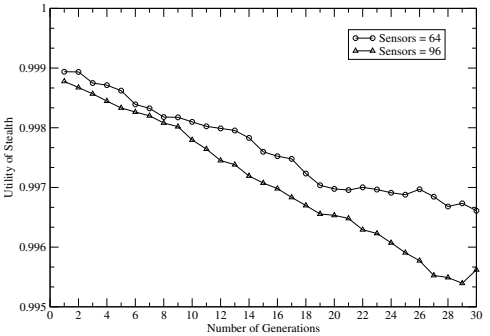

Figure 10: Change in Utility of Stealth with Different Number of Opponents

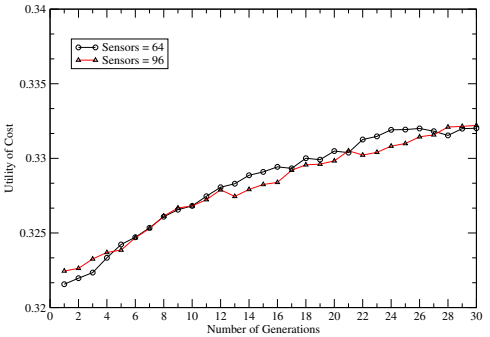

Figure 11: Change in Utility of Cost with Different Number of Opponents

In our second experiment set, we vary the number of sensors (either 64 or 96) while fixing the number of opponents to 32. As can be seen from Figures 9- 11, the utility of the visibility and the utility of the cost increase whereas the utility of the stealth decreases with the number of generations considered.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a sensor planning system that unifies research from active vision with sensor planning, decision theory, and genetic algorithms. Calculating sensor utility factors for several vehicles over a large area is computationally very expensive. So, directing and limiting the search and calculations involved in sensor planning is the main part of our extensions.

In our experiments, the initial population is set randomly and we only consider the validity of the attribute values and reachability of sensors on the selected locations. Initial population highly affects the characteristic of the convergence of the solution. We are planning to generate the initial population according to the landscape features. Additionally, local search strategy can be made more effective by changing the search mechanism. Another major extension is to add the moving capability for both sensors and opponent vehicles, which require their initial positions and the moving directions.

Our GA-based solution for the sensor optimization problem is a computationally-intensive application; and it is planned to implement its parallel version using message-passing libraries and to execute it on a cluster of machines. Genetic Algorithms can be easily parallelizable and several models in parallel genetic algorithms are suitable for the multi-attribute optimizations.

7. REFERENCES

[1] John C. Butler, Douglas J. Morrice, and Peter W. Mullarkey. A Multiple Attribute Utility Theory Approach to Ranking and Selection. *Management Science*, 47:800–816, June 2001.

[2] Tolga Can, Veysi Isler, and Ziya Ipekkan. Sensor optimization in a virtual environment. In *9th Conference on Computer Generated Forces and Behavioral Representation*, Orlando, Florida, 2000.

[3] Robert T. Clemen. *Making Hard Decision*. PWS Kent Publishing, Boston, MA, 1991.

[4] Diane J. Cook, Piotr Gmytrasiewicz, and Lawrence B. Holder. Decision-Theoretic Cooperative Sensor Planning . *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:1013–1023, October 1996.

[5] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics Principles and Practice in C - Second Edition*. Addison Wesley, Reading, Massachusetts, 1995.

[6] N. K. Jaiswal. *Military Operations Research: Quantitative Decision Making*. Kluwer Academic Publishers, Boston, MA, 1997.

[7] R. L. Keeney and H. Raiffa. *Decision with Multiple Objectives*. John Wiley and Sons, New York, 1976.

[8] Steen Kristensen. Sensor Planning with Bayesian Decision Theory. *Robotics and Autonomous Systems*, 19:273–286, 1997.

[9] L. Mihaylova, T. Lefebre, H. Bruyninckx, K. Gadeyne, and J. De Schutter. Active sensing for robotics - a survey. In *Proceedings of International Conference in Numerical Methods and Applications*, pages 316–324, 2002.

[10] Michale K. Reed and Peter K. Allen. Constraint-Based Sensor Planning for Scene Modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1460–1467, December 2000.

[11] John Spletzer and Camillo Taylor. Sensor planning and control in a dynamic environment. In *Proceedings of International Conference on Robotics and Automation*, 2002.

[12] Ioannis Stamos and Peter K. Allen. Interactive sensor planning. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pages 489–494, 1998.