ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/31376145

Optimal Sensor Deployment for Wireless Surveillance Sensor Networks by a Hybrid Steady-State Genetic...

Article *in* IEICE Transactions on Communications · November 2008

DOI: 10.1093/ietcom/e91-b.11.3534 · Source: OAI

CITATIONS
23

READS
342

5 authors, including:

Si-Ho Cha
Chungwoon University
54 PUBLICATIONS **194** CITATIONS

SEE PROFILE

Minho Jo
Korea University
91 PUBLICATIONS **871** CITATIONS

SEE PROFILE

All content following this page was uploaded by Minho Jo on 17 July 2017.

The user has requested enhancement of the downloaded file.

IEICE TRANS. COMMUN., VOL.E91–B, NO.11 NOVEMBER 2008

PAPER *Special Section on Emerging Technologies for Practical Ubiquitous and Sensor Networks*

Optimal Sensor Deployment for Wireless Surveillance Sensor Networks by a Hybrid Steady-State Genetic Algorithm

Jae-Hyun SEO[†], **Yong-Hyuk KIM**[†], **Hwang-Bin RYOU**[†], ***Nonmembers***, **Si-Ho CHA**[††], ***and*** **Minho JO**[†††a)], ***Members***

SUMMARY An important objective of surveillance sensor networks is to effectively monitor the environment, and detect, localize, and classify targets of interest. The optimal sensor placement enables us to minimize manpower and time, to acquire accurate information on target situation and movement, and to rapidly change tactics in the dynamic field. Most of previous researches regarding the sensor deployment have been conducted without considering practical input factors. Thus in this paper, we apply more real-world input factors such as sensor capabilities, terrain features, target identification, and direction of target movements to the sensor placement problem. We propose a novel and efficient hybrid steady-state genetic algorithm giving low computational overhead as well as optimal sensor placement for enhancing surveillance capability to monitor and locate target vehicles. The proposed algorithm introduces new two-dimensional geographic crossover and mutation. By using a new simulator adopting the proposed genetic algorithm developed in this paper, we demonstrate successful applications to the wireless real-world surveillance sensor placement problem giving very high detection and classification rates, 97.5% and 87.4%, respectively.

key words: *wireless sensor networks, surveillance sensor deployment, hybrid steady-state genetic algorithm*

1. Introduction

Wireless sensor networks (WSNs) are utilized to monitor the areas where traditional networks could not be used. WSNs have been widely implemented in practice such as disaster intervention, habitat monitoring, microclimate research, surveillance, emergency search-and-rescue and medical care, as sensor capability has improved [1], [2]. Currently, surveillance operations require an automated sensor deployment system. Especially, an automated sensor allocation and management system [3] is vital to a ground operation that monitors the movement of targets. We are much interested in the improvement of sensor capability as well as optimal sensor placement. The optimal sensor emplacement enables us to minimize manpower and time, to acquire accurate information on target movement, and to rapidly change tactics in the dynamic field. Therefore, the optimally fitted sensor emplacement allows for us to have a good awareness of targets.

Manuscript received March 1, 2008.
Manuscript revised June 3, 2008.

[†]The authors are with the Department of Computer Science and Engineering, Kwangwoon University, Korea.
[††]The author is with the Department of Information and Communication Engineering, Sejong University, Korea.
[†††]The author is with Graduate School of Information Management and Security, Korea University, Korea.
a) E-mail: minhojo@korea.ac.kr

DOI: 10.1093/ietcom/e91–b.11.3534

In general, the position of sensors affects coverage, communication costs, and resource management of surveillance sensor networks. Thus we are required to place a sensor in the best location. Nevertheless, we randomly deploy the sensors to monitor and locate the target vehicles. This random deployment may cause sensors to be centralized or to be surrounded by terrain features. It will decrease the probability of detection and classification of target movement. Most of previous literatures regarding the sensor deployment have focused their research on theoretical points without considering practical input factors. Thus in this paper, we apply more real-world input factors such as sensor capabilities, terrain features, target identification, and direction of target movements to the wireless surveillance sensor placement problem.

In addition to the more practical problem-solving contribution of this paper, we propose a novel hybrid steady-state genetic algorithm (GA) to solve the practical sensor deployment problem with more complex real-world input factors. The optimal sensor placement is a problem of maximizing coverage and minimizing the number of sensors. It is a kind of the Minimum Set Cover (MSC) problem. MSC is an NP-hard problem. We here propose a hybrid steady-state GA to find the high-quality solutions of the wireless surveillance sensor deployment and apply it to a practical case study. The proposed genetic algorithm to solve the real-world sensor deployment introduces 2-dimensional geographic crossovers and mutations, and local optimization which are unique features. In this paper, the experimental data was obtained from a real surveillance situation.

The remainder of this paper is organized as follows: Sect. 2 explains previous studies for the optimal sensor emplacement. Section 3 defines the real-world problem handled in this paper. In Sect. 4, we propose a hybrid steady-state genetic algorithm for the optimal sensor deployment. Section 5 describes the environments of experiment and analyzes the results of simulation. The paper ends with conclusions in Sect. 6.

2. Related Work

There have been a number of other studies for the optimal sensor deployment.

Lamm [4], [5] developed sensor employment rules on a system engineering framework using statistical tools such as Response Surface Methodology (RSM) [6]. Lamm used a

Copyright © 2008 The Institute of Electronics, Information and Communication Engineers

computer simulation to test experimental designs. The computer simulation uses MapInfo Professional and MapBasic software programs based on the geographic information system. The major steps of the RSM experiment are as follows. The initial step includes the design for the RSM experiment before sensor deployment. The next step is to test whether the experiment model is linear or not. The final step is to develop initial sensor emplacement rules considering tradeoff with multi-objectives and to refine them as the steps are iterated. This test requires a lot of time and moreover it is difficult to conduct elaborate experiments.

Virtual force algorithm (VFA) of Zou et al. [7] employed the repulsive and attractive forces so as to enlarge the coverage for the target region. In VFA, a cluster-head gathers the location information of every sensor. Every sensor has a force on it influenced by other sensors, obstacles, and preferential areas. The experiment of VFA shows that more sensors should be emplaced around the obstacles than in the preferential areas. However, the cost of sensor devices will be drastically increased because more and more sensors are needed as the target area is expanded. It will be required how to decrease the number of sensors.

Wang et al. [8] could meet the grid-quorum that a high-density grid sends advertisements along its row and a low-density grid sends requests along its column. They could achieve better fairness as describing the cascaded movement. A direct long movement could be decomposed into multiple short movements. The accomplished fairness could make it possible to relocate sensors evenly. The sensor relocation maximizes coverage and shows higher detection rates.

Sharp et al. [9] made an experiment on the pursuer-evader problem. The problem is that the evader moves randomly in the sensing field and the pursuer tries to capture the evader by the information from the static wireless sensor network. The experiment contributed to building the basic sensor emplacement system significantly.

There were two genetic approaches. Xu et al. [10] proposed in 2006 a genetic algorithm to optimize sensor placement and to minimize the number of sensors. Zhao et al. showed energy efficient heterogeneous sensor deployment using a general genetic algorithm in 2007 [11]. The key operators of the algorithms are order-mapped crossover and simple inversion mutation. And it is difficult to apply them to the real-world because of lack of real-world input factor modeling. They don't consider the real-world input factors such as sensor capabilities, terrain features, target identification, and direction of target movements for the sensor placement problem which are included in our problem solving. Because our proposed hybrid steady-state genetic algorithm uses 2-dimensional geographical crossover and mutation while Xu et al. and Zhao et al.'s genetic algorithms do not use them, we cannot compare performance results of both algorithms and ours.

3. Problem Definition

Sensor capabilities, terrain features, target identification, the direction of target movements, and more factors are essential to the real-world sensor placement. It is crucial to consider how the input factors will affect the target detection. Therefore, we introduce the sensor network elements and performance evaluation function, and denote input variables in this section.

3.1 Problem Statement

A seismic sensor can detect vibrations caused by the movement of vehicles or human beings. An acoustic sensor can detect acoustic signatures caused by the movement of vehicles or human beings. FLIR (Forward-Looking Infrared Radar) is a device that senses infrared radiation. An attrition sensor is used to destroy sensors which belong to the other party. A wheeled vehicle is a vehicle that moves on wheels and usually has a container for transporting things or people such as trucks. A tracked vehicle is a vehicle that runs on tracks instead of wheels such as trains.

In our experiment, the targets are classified into wheeled vehicles and tracked vehicles. The terrain features consist of rivers, trees, hills, towns, and open fields. Seismic, acoustic, FLIR, and attrition sensors are used for our experiment.

The effective sensor capabilities are obtained from U.S. Army Research Laboratory (ARL) [12]. Table 1 shows the probability of the target movements [4].

Sensor capabilities are probability of sensing a target. Table 2 and Table 3 show the degradation of sensor capabilities which is affected by the terrain features and the Euclidean distance between sensors and obstacles [4]. The criteria capability 100% is based on an open field. On the other hand, the town field gives worst capability. A feature with the more and bigger obstacles such as buildings, water, hills, and trees will degrade the more sensor capability due to refraction, reflection, diffraction, and interference. In Ta-

Table 1 Probability ranges used for target movements.

Direction	Low value	High value	Total probability
Backwards	0.00	0.02	0.02
Left	0.03	0.22	0.20
Right	0.23	0.42	0.20
Stays	0.43	0.50	0.08
Forward-Right	0.51	0.70	0.20
Forward	0.71	0.80	0.10
Forward-Left	0.81	1.00	0.20

Table 2 Degradation of sensor capabilities by terrain features (%).

Sensors	Feature				
	Open	River	Trees	Hills	Towns
Seismic	100	50	90	65	10
Acoustic	100	75	60	25	10
FLIR	100	100	75	25	25
Attrition	100	N/A	75	50	10

Table 3 Degradation of sensor capabilities by natural logarithm function.

Sensors	Targets	POD	C_1	C_2	D_1	D_2
Seismic	W	0.95	–1.3738	8.5464	250	500
	T	0.95	–1.4704	10.0920	500	900
Acoustic	W	0.95	–1.3738	8.5464	250	500
	T	0.95	–2.0922	14.6560	700	1100
FLIR	W	0.95	–1.1955	6.8983	150	300
	T	0.95	–1.1955	6.8983	150	300
Attrition	N/A	N/A	–0.1135	0.5656	0	100

* *W: wheeled vehicle, T: tracked vehicle, the others: see Table 4. D_1 and D_2 in meters*

Table 4 Notation of input variables and evaluation function.

Items	Description
D	Detection rate.
C	Classification rate.
U	The number of undetected vehicles.
W_{tgt}	The number of wheeled vehicles.
T_{tgt}	The number of tracked vehicles.
C_{class}	The number of correctly classified vehicles. Overlapping is allowed.
C_{detect}	The number of detected vehicles by sensor. Overlapping is allowed.
$C_{attrition}$	The number of sensors detected by the attrition sensors of the other party.
(α, β)	The initial coordinates of a sensor.
(x, y)	The variable coordinates of a sensor.
(a, b)	The coordinates of a moved sensor.
$S(x)$	Maximum probability that a target located on a cell detected by a sensor when the Euclidean distance between the center of the cell and the sensor is x.
POD	Effective detection rate of every sensor. See Table 3.
C_1, C_2	Constants to compute degradation of sensor capability. See Table 3.
D_1, D_2	Detection range of sensors. See Table 3.
R	Degradation of sensor capabilities by terrain features. See Table 2.
t	Probability that sensors can detect vehicles per each grid cell.
c	Probability that sensors can classify vehicles per each grid cell.
r	Probability that attrition sensors can detect sensors.
e	Probability that attrition sensors can detect sensors per each grid cell.
p	Probability of the target movement direction. See Table 1.

ble 2, notice that degradation is significant for the features like river, trees, hills, and towns.

In Table 3, effective detection rates (Power of Detection, POD) converted by the natural logarithm function, are provided. C_1 and C_2 are constant to compute degradation of sensor capability and D_1 (m) and D_2 (m) are detection range of sensors.

3.2 Input Variables and Evaluation Function

Input variables and evaluation function are defined in Table 4. Every target moves passing through a grid cell of area 5 km × 5 km. The direction of target movements in a grid is defined by a probability given in Table 1 and Formula (2). The frequency of target movements is chosen at random. Multiple target movements can occur simultaneously in a grid.

We compute the number of undetected targets by the function. Then, we calculate the detection rate and the correct classification rate as provided in Formula (7). Detection means that sensors sense a target successfully. Classification accounts for the case of classifying a type of target successfully among the detected targets. We consider the dynamic input variables for our problem, such as target movement direction (p) in Tables 1 and 4 and breakdown of sensors destroyed (detected) by the attrition sensors of the other party in Formula (2) and (4).

The following are equations describing the evaluation function to test our sensor employment capability. Formula (1) sets the detection rate for each cell based on the distance between the center of each cell and sensors. Formula (2) shows probability ranges used for targets and the coordinates of the moved sensor are given by Formula (3). Formula (4) and Formula (5) present the process of getting the numbers of detected targets (C_{detect}), correctly classified target vehicles (C_{class}), and our sensors detected by the attrition sensors of the other party ($C_{attrition}$), all of which are to be parameters of simulation.

$$S(x) = \begin{cases} POD \times R & \text{if } 0 \le x \le D_1 \\ (\log(x) \times C_1 + C_2) \times R & \text{if } D_1 \le x \le D_2 \end{cases} \tag{1}$$

$$(x, y) = \begin{cases} (0, -1) & \text{if } 0 \le p \le 0.02 \\ (-1, 0) & \text{if } 0.03 \le p \le 0.22 \\ (1, 0) & \text{if } 0.23 \le p \le 0.42 \\ (0, 0) & \text{if } 0.43 \le p \le 0.5 \\ (1, 1) & \text{if } 0.52 \le p \le 0.7 \\ (0, 1) & \text{if } 0.71 \le p \le 0.8 \\ (-1, 1) & \text{if } 0.81 \le p \le 1 \end{cases} \tag{2}$$

$$(a, b) = (\alpha, \beta) + (x, y) \tag{3}$$

$$\begin{aligned} H_l &= \begin{cases} 1 & \text{if } r \in R, r < e \\ 0 & \text{otherwise} \end{cases} \\ I_k &= \begin{cases} 1 & \text{if } r \le e, H_l = 1 \\ 0 & \text{otherwise} \end{cases} \\ F_i &= \begin{cases} 1 & \text{if } p \in R,\ p < t,\ I_k = 0 \\ 0 & \text{otherwise} \end{cases} \\ G_j &= \begin{cases} 1 & \text{if } c \le 0.95, F_i = 1 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4}$$

In Formula (4) and Formula (5), i denotes the number of detected vehicles, j accounts for the number of classified vehicles, and l and k stand for the number of our sensors detected and destroyed, respectively by the other party. I_k tells whether or not our sensor is destroyed by the other party's attrition sensor. Thus I_k will influence the number of detecting target vehicles, F_i. If the number of instances of I_k = 1 increases, the number of detecting target vehicles, F_i decreases (i.e., the case of F_i = 1 occurs less).

$$C_{detect} = \sum_{i=1}^{W_{tgt}} F_i + \sum_{i=1}^{T_{tgt}} F_i$$

$$C_{class} = \sum_{j=1}^{W_{tgt}} G_j + \sum_{j=1}^{T_{tgt}} G_j \tag{5}$$

$$C_{attrition} = \sum_{k=1}^{W_{tgt}} I_k + \sum_{k=1}^{T_{tgt}} I_k$$

$$U = \left(W_{tgt} + T_{tgt}\right) - C_{detect} \tag{6}$$

$$D = 1 - \frac{U}{\left(W_{tgt} + T_{tgt}\right)}, C = \frac{C_{class}}{C_{detect}} \tag{7}$$

When the target vehicles move passing through the area, we compute the number of undetected targets by applying Formulas (2), (3), (4), and (5). Finally, we compute the detection rate and the correct classification rate by Formula (7). Notice that $C_{attrition}$ is not for equations but only for monitoring the number of our sensors destroyed by the other party. The number of our sensors destroyed by attrition sensors is already reflected in Formula (4).

4. The Proposed Genetic Algorithm

In this section, our hybrid steady-state genetic algorithm is proposed, and the principal operators to optimize sensor emplacement are illustrated.

4.1 Hybrid Steady-State GA

Genetic algorithms (GAs) were first proposed by Holland [13]. GAs are a class of probabilistic optimization algorithms and inspired by the biological evolution process. GAs model the concept of natural selection and genetic inheritance.

Our hybrid steady-state GA is the mixture of the steady-state GA and the hybrid GA. The generational GAs replace the whole or a part of solutions with new offspring per each generation. Meanwhile, the steady-state GA replaces just one solution with a new offspring per each generation. We classify GAs into hybrid or pure ones according to whether or not local optimization is applied. The hybrid GA uses local optimization for improvement of offspring produced by crossover.

Figure 1 shows the flow diagram of our hybrid steady-state GA. The steps of the proposed GA are described in the following.

1. Step 1 generates population based on a random deployment.
2. Step 2 is the selection of Parent 1 and Parent 2.
3. Step 3 is the process of the crossover which creates an offspring by the genes' recombination of Parent 1 and Parent 2.
4. Step 4 is the process of the mutation that varies each gene of the offspring by some probability. Then we make simulation for evaluating the offspring and get the information of detection rate and classification one.
5. Step 5 is the process of the local optimization. The operator moves each sensor to the best adjacent position by trying to move the sensor to the eight neighbor cells.
6. Step 6 is the process of the replacement that replaces with the final offspring a solution of the population.
7. All steps will be iterated until the stop condition is satisfied. The stop condition is set up to the number of generations.

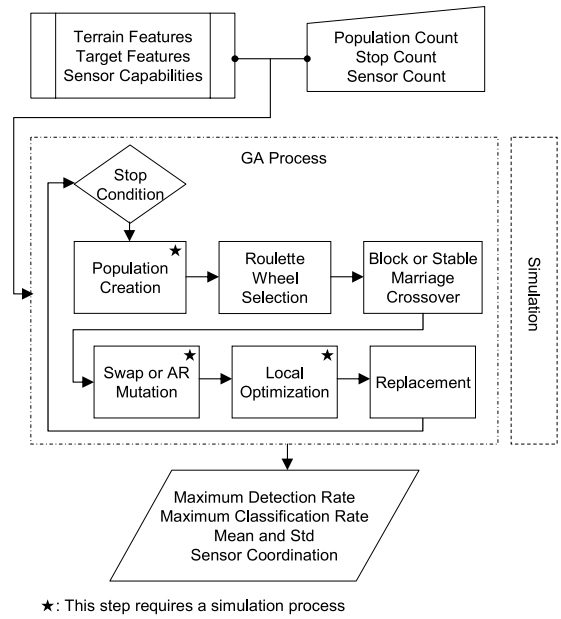


Fig. 1 A flow diagram of the proposed hybrid steady-state GA.

4.2 Crossovers and Mutations

We apply two different crossover operations and two mutation operations in the proposed hybrid steady-state GA. The two crossover operations are block crossover and table marriage crossover and the two mutation operations include swap mutation and attractive/repulsive (AR) mutation.

4.2.1 Block Crossover

Anderson et al. [14] suggested a block uniform crossover on a two-dimensional matrix chromosome that tessellated the chromosome into $i \times j$ blocks; the gene in the block is copied from a uniformly selected parent. In this paper, we do not use a random property because in our preliminary test it did seldom give any effects on the performance. Figure 2 shows the block crossover, a modified version of the original block uniform crossover. Table 5 shows the pseudo-code of this modified operator.

4.2.2 Stable Marriage Crossover

A stable marriage problem [15] has two finite sets: the set of men and that of women. It is designed for finding stable matching which is just a one-to-one mapping between both sexes. In this paper, we apply the stable marriage problem to a new crossover operator named stable marriage crossover.

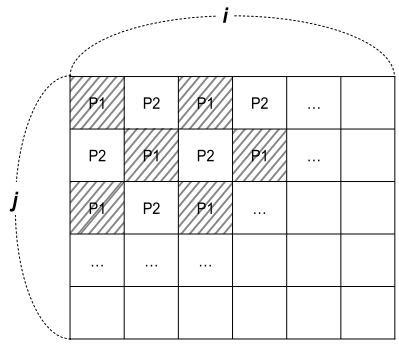
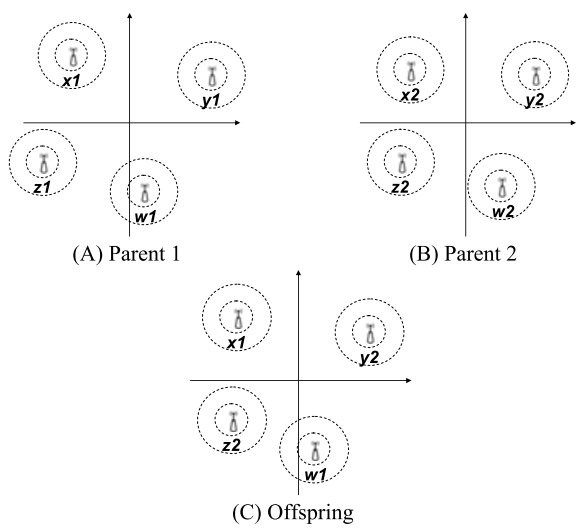


Fig. 2 A block crossover.

Table 5 The pseudo-code of the block crossover.

```
INPUT: Col_Range_Count, Row_Range_Count  // Block count
OUTPUT: Deploying P1 and P2 in a grid
For each i=0,1,...,Col_Range_Count-1
          and each j=0,1,...,Row_Range_Count-1 do
    If (i + j) mod 2 = 0 then Parent := P1;
        Else Parent := P2;
    End if
    Child_Block[i, j] := Parent;
End for

If Child_Block[i, j] = P1 then Add P1 to Child
    Else Add P2 to Child;
End if
```



(A) Parent 1 (B) Parent 2

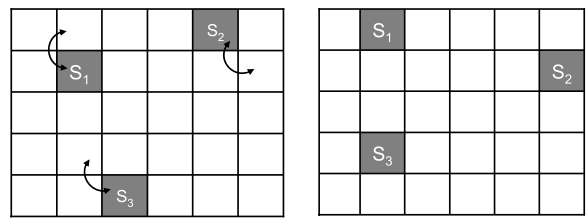
(C) Offspring

Fig. 3 Stable marriage crossover.

The operator makes pairs between nearest sensors of Parent 1 and Parent 2. For each pair, the genes are copied from a randomly selected parent. Figure 3 shows an example of paring the genes (sensors) of both parents and generating offsprings. In the first step, the marriages of nearest sensors between Fig. 3(A) and Fig. 3(B) are (x_1, x_2), (y_1, y_2), (z_1, z_2), and (w_1, w_2). In the second step, genes are determined. Based on randomly generated binary digit, '0' or '1,' the gene of an offspring is determined: cloning the gene of Parent 1 if the binary digit is '0'and the one of Parent 2 if '1.' Figure 3(C) represents the newly generated offspring.

Table 6 The pseudo-code of the stable marriage crossover.

```
INPUT: P1i , P2j ,N  // sensor i of Parent1, sensor j of Parent2, the
                     // number of sensors for each parent, respectively
OUTPUT: Oi           // sensor i of Offpring
I={1,2,...,N};     // I: a set of sensors
For each i=1,2,...,N do
    m:= argmin_{j∈I} Euclidean_distance (P1i, P2j);
    P1i marries P2m; // P2m: a sensor among P2 which is nearest
                     // from P1i
    K := random integer of 1 or 2;
    if(k = 1) Oi := P1i;
   Else      Oi := P2m;
    I := I\{m};
End for
```



(A) Before mutation (B) After mutation

Fig. 4 Swap mutation.

Table 7 The pseudo-code of the swap mutation.

```
INPUT: Grid_Number  // Column cell count x row cell count
OUTPUT: Placing a sensor into a randomly-chosen empty grid cell
For each i=1,2,...,Grid_Number do
   If (sensor is in grid_cell i) and (random(100) < 5) then
        placing a sensor and a randomly-chosen empty grid cell
   End if
End for
```

For example, a pair (x_1, x_2) gets '0,' following the gene of Parent 1, i.e., their offspring will be x_1. Table 6 shows the pseudo-code of this operator.

4.2.3 Swap Mutation

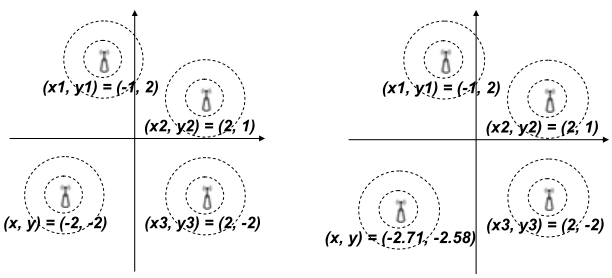
Swap mutation is the operator that exchanges the gene positions on 2-dimensional chromosome with some probability. This operator moves each individual sensor to a randomly-chosen empty cell with a specified small probability. Figure 4 shows an example of swap mutation under a virtual situation that every gene position is mutated. Table 7 shows the pseudo-code of this operator.

4.2.4 Attractive and Repulsive Mutation

In attractive and repulsive (AR) mutation, every sensor is moved to a new position which is computed by attractive or repulsive force between each pair of sensors. The procedure of this mutation is in the following. (1) Calculate distances between a sensor and the other sensors. (2) α value is calculated by sensor effective range, target region, and distance. (3) We multiply the distance and α value. Then we divide it into the number of other sensors and add it to the initial

Table 8 Notation for AR mutation.

Items	Descriptions
D	Constant to change the distance of sensors to real distance.
W	The width of target area.
r	Effective range of sensors.
N	The number of sensors.
d	Real distances between target sensor and other sensors.
α	The variable to control the coordinates of sensors.
S_x	x coordinate of target sensor.
S_y	y coordinate of target sensor.



(A) (B)

Fig. 5 An example of AR mutation.

sensor coordinates.

An example of this AR mutation is illustrated below. The notation for the AR mutation is shown in Table 8. Assume that one scale is 200 meters. The area of the target region is 5 km × 5 km. Figure 5(A) is an initial sensor deployment. Figure 5(B) is a sensor emplacement after AR mutation is completed. The example shows that an initial sensor position $(-2, -2)$ is mutated to $(-2.71, -2.58)$. The pseudo-code of the AR mutation is given in Table 9.

$$D = 200,\ W = 5000,\ r = 250,\ N = 3$$
$$d_1 = D \times \sqrt{(x - x_1)^2 + (y - y_1)^2},$$
$$d_2 = D \times \sqrt{(x - x_2)^2 + (y - y_2)^2},$$
$$d_3 = D \times \sqrt{(x - x_3)^2 + (y - y_3)^2},$$
$$\alpha_1 = \frac{d_1 - r \times 2}{W},\ \alpha_2 = \frac{d_2 - r \times 2}{W},$$
$$\alpha_3 = \frac{d_3 - r \times 2}{W},$$
$$S_x = \frac{(x - x_1) \times \alpha_1 + (x - x_2) \times \alpha_2 (x - x_3) \times \alpha_3}{N} + x$$
$$S_y = \frac{(y - y_1) \times \alpha_1 + (y - y_2) \times \alpha_2 (y - y_3) \times \alpha_3}{N} + y$$
$$\therefore \quad (S_x, S_y) = (-2.71, -2.58)$$

4.3 Parameters

Simulation is conducted using the proposed hybrid steady-state GA. The parameters used in the GA are as following:

- Encoding: It represents the coordinate of every sensor using a 2-dimensional real vector.

Table 9 The pseudo-code of the AR mutation.

```
INPUT: Si(α,β), Sj(α,β) // Initial position of sensor i in offspring
                        // selected at random. Initial position of the
                        // other sensor j in offsrping relative to sensor i
OUTPUT: Si(a,b)   // New coordinate of sensor i
Si(a,b):= Si(α,β);
For each i=1,2,...,N do
    Δ:=(0,0);
    For each j=1,2,...,N (i ≠ j) do
        Dij:=Euclidean_distance(Si(a,b) ,Sj(α,β);
        αij:=(Dij-2r)/W;
        Δ:= Δ+ αij(Si(a,b) - Sj(α,β));
    End for
    Δ:= Δ/(N-1);
    Si(a,b):= Si(a,b)+ Δ;
End for
```

- Population: It is generated by 30 random deployments of sensors.
- Selection: We use a roulette-wheel-based proportional selection in which the best individual is selected with four times higher probability than the worst.
- Block crossover: This operator divides the target area by 15 columns and 15 rows. Selected parents show a plaid- shaped rectangle.
- Stable marriage crossover: This operator pairs the nearest sensors from both parents. Then, it recombines the genes which are randomly chosen from each pair.
- Swap mutation: This operator randomly exchanges a sensor cell with an empty cell by some probability.
- AR mutation: This operator makes sensors move to the position, which is computed by attractive and repulsive forces, to maximize coverage with some probability.
- Local optimization: The operator moves each sensor to the best adjacent position by trying to move the sensor to the eight neighbor cells.
- Replacement: If Parent 1 sensor position or Parent 2 sensor position is better than offspring sensor position, inferior one of Parent 1 and Parent 2 is replaced with the offspring sensor position. If an offspring sensor position is better than the worst sensor position in the population, the worst one is replaced. If not, the offspring sensor position is discarded [16].
- Stop condition: The GA stops when the number of generations reaches a fixed value (we set the number to 100.).

5. Experiments

5.1 Simulation Environments

We tried to use the MapInfo Professional and MapBasic software for our simulation of the sensor deployment. However since the Map Info simulator showed lower time-performance, we have developed a new simulator to raise the efficiency of time-performance. Table 10 shows an average error and a standard deviation between the two simulators after we make experiments over 1,000 runs. The test result verifies the credibility of our simulator.

Table 10 Comparison of an average error between MapInfo simulator and our simulator.

		M	O
Average	Detection	0.823	0.844
	Classification	0.867	0.866
Standard Deviation	Detection	0.059	0.052
	Classification	0.015	0.013

** M: MapInfo simulator, O: our simulator*

Table 11 Simulation environment.

Parameters	Comment
Target area	5km x 5km
Terrain features	hills, trees, rivers, towns
Wheeled vehicles	30
Tracked vehicles	20
Mobile seismic sensors	5
Mobile acoustic sensors	5
Attrition sensors of the other party	6

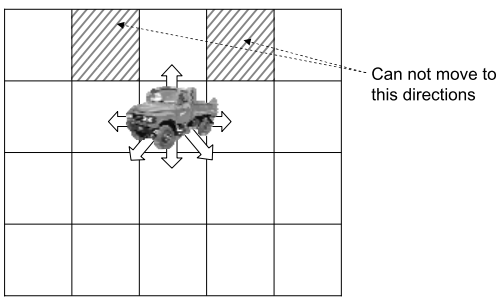


Fig. 6 Direction of target movements.

Table 12 Setup of experiments (2^3=8).

GA steps	Crossover	Mutation	L
Operators	Stable marriage	Swap	Apply
	Block	AR	N/A

** L: local optimization*

The following scenario is an example to make experiments presented in this paper. Table 11 shows the parameters used in the simulation.

The target vehicles move as in Fig. 6. Probability for the directions of target movement is shown in Table 1 of Sect. 3.1. Table 1 shows the probability for each direction of target movements depending on a random number generated during the simulation. A simulation runs based on the terrain information. The degradation of sensor capabilities is defined in Table 2 and Table 3 of Sect. 3.1.

5.2 Experimental Results

We compare eight approaches for the simulation given in Table 12. The approaches are combinations of crossovers, mutations, and local optimization. We conduct experiments on each combination.

The test results of Table 13 show the best results (D: max detection rate. C: max classification rate), average one (DA: detection average. DC: classification average), and running time of each combination.

Table 13 Test results of proposed GA.

No.	X	M	L	D	C	DA	CA	T
1	m	s	A	95.0	87.7	92.8	88.2	4396s
2	**m**	**r**	**A**	**97.5**	**87.4**	**93.2**	**87.8**	4371s
3	b	s	A	**97.5**	84.5	90.5	86.1	4396s
4	b	r	A	95.0	83.8	90.5	90.0	4350s
5	m	s	N	81.3	90.6	78.9	87.5	6597s
6	m	r	N	81.3	85.7	79.2	87.0	6642s
7	b	s	N	82.5	85.7	79.3	86.0	6480s
8	b	r	N	85.0	88.9	80.2	87.0	6583s

** X: crossover, M: mutation, L: local optimization, A: apply, N: not apply, D: max detection rate, C: max classification rate, DA: detection average, CA: classification average, T: average running time, m: stable marriage crossover, b: block crossover, s: swapping mutation, r: AR mutation.*

** Every result is from 10 runs.*

** The bold-faced figures mean the best result of 10 runs.*

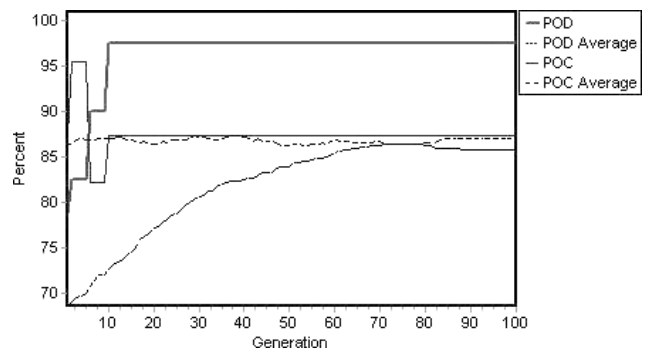


Fig. 7 Transition of detection rate and classification rate for No.2 in Table 13.

The column "T" shows the average CPU seconds taken on a Core2 2.4-GHz personal computer. Because the hybrid steady-state simulation is conducted before deployment, the CPU time (T) is NOT of interest unless it is that long like a NP-hard problem time. In the experiment, the combination of stable marriage crossover, AR mutation, and local optimization and the combination of block crossover, swap mutation, and local optimization showed the best performance. Comparison between No. 1–4 and No. 5–8 proves that local optimization shows the synergy effects around all combinations. In the experiments with No. 2 and No. 3, the results show that the combination of stable marriage crossover, AR mutation, and local optimization and the combination of block crossover, swap mutation, and local optimization have higher synergy effect than other combinations. Notice that in Table 13, the max classification rate (C) is only for the case of the max detection rate (D), while the classification average (CA) is for the case of all detection trials. Thus it is possible that the max classification rate is less than the classification average. For example, we have a total of 10 vehicles. If 10 vehicles (5 classified), 6 vehicles (4 classified), and 4 vehicles (3 classified) are detected, their detection rates (classification rates) are 100% (50%), 60% (66%), and 40% (75%), respectively. The max classification rate corresponding to the max detection rate, 100%, is 50% but the classification average will be 63.6%.

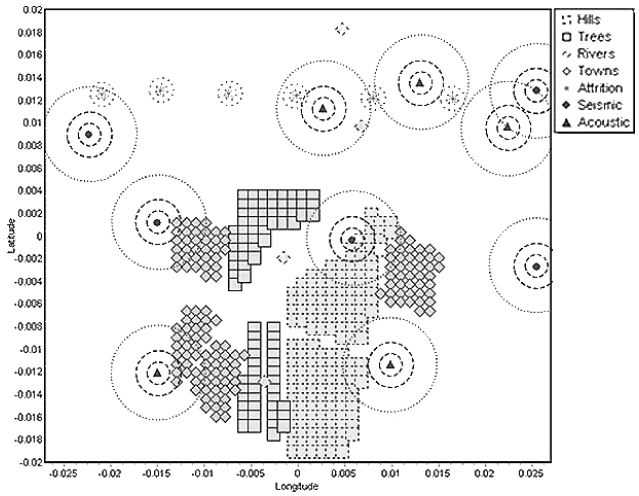


Fig. 8 An initial sensor deployment. (detection rate: 72.5%, classification rate: 84.6%)

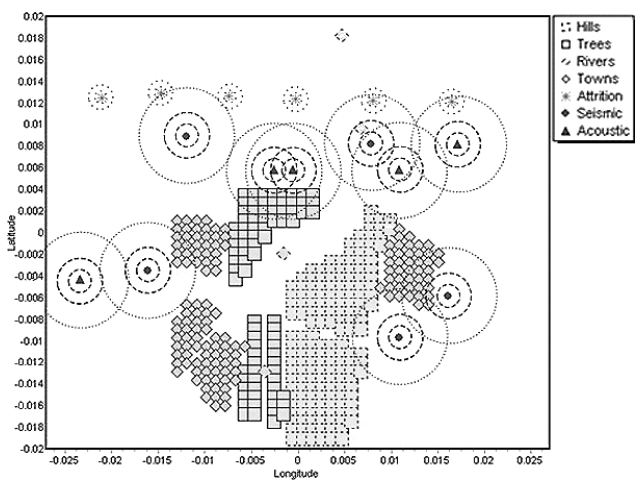


Fig. 9 An optimized sensor deployment. (detection rate: 97.5%, classification rate: 87.4%)

Figure 7 shows the evolutionary process for No. 2 of Table 13. In Fig. 7, the y axis indicates the percentage for detection and classification. The x axis is the number of generations. POD and POC are the maximum detection rate and the maximum classification rate, respectively. POD Average and POC Average means the averages of detection rate and classification rate over population, respectively. POD shows 97.5% and POC is 87.4%. Averages of POD and POC over population present 86.6% and 86.9% at the end, respectively.

Figure 8 visualizes a randomly deployed sensor placement. The detection rate and the classification rate of the initial sensor deployment are 72.5% and 84.6%, respectively. Figure 9 visualizes an optimal sensor deployment performed by using the proposed hybrid steady-state GAs. The detection rate and the classification rate are 97.5% and 87.4%, respectively.

Table 14 shows the experimental results over various numbers of sensors. In these experiments, we used GAs with stable marriage crossover, AR mutation, and local optimization. The column "Sensors" in Table 14 shows the number of seismic sensors and acoustic sensors, respectively. As the number of sensors increases, the detection rate is higher but execution time is longer. In other words, more number of sensors gives more accurate information. However the running time is influenced by the local optimization because more sensors require more local optimization process.

Table 14 Test results over various numbers of sensors.

Sensors	D	C	T
6, 6	92.5	94.1	5538s
7, 7	95.0	93.6	7359s
8, 8	97.5	93.1	8319s
9, 9	100	87.5	9142s

** D: max detection rate, C: max classification rate, T: the running time.*
** Every result is from one run.*

6. Conclusion

In this paper, we propose and apply a new genetic algorithm to optimize sensor emplacement which is suitable for wireless surveillance sensor networks. Since most of previous researches have been conducted on the sensor placement with lack of real-world input factors, we applied real-world input factors such as sensor capabilities, terrain features, target identification, and direction of target movements for the surveillance sensor placement problem. We demonstrated successful applications of the real-world through simulation. We had very good results of simulation giving detection rate of 97.5% and classification rate of 87.4% by the proposed hybrid steady-state genetic algorithm using stable marriage crossover, attractive and repulsive mutation, and local optimization. We ultimately could prove that the proposed hybrid steady-state genetic algorithm works best for the optimal surveillance sensor placement based on two-dimensional genetic operators.

Future work includes the investigation of extended sensor capabilities, additional environmental elements, and more detailed grids. In order to achieve better performance, designing other improved two-dimensional operators will be promising.

Acknowledgement

This research was supported by the Ministry of Knowledge Economy, South Korea, under the ITRC (Information Technology Research Center) support program supervised by the Institute of Information Technology Advancement (IITA-2008-(C1090-0801-0046)).

References

[1] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," Computer, vol.37, no.8, pp.41–49, Aug. 2004.
[2] X. Xu and S. Sahni, "Approximation algorithms for wireless sensor

deployment," IEEE Trans. Comput., vol.56, no.12, pp.1681–1695, Dec. 2007.

[3] S.-H. Cha, J.-E. Lee, M. Jo, H.Y. Youn, S. Kang, and K.-H. Cho, "Policy-based management for self-managing wireless sensor networks," IEICE Trans. Commun., vol.E90-B, no.11, pp.3024–3033, Nov. 2007.

[4] L.M.J. Lamm, Develop Measures of Effectiveness and Deployment Optimization Rules for Networked Ground Micro-sensors, M.S. Thesis, Division of Systems Engineering, School of Engineering and Applied Science, University of Virginia, May 2001.

[5] L.M. J. Lamm, G.A. Lamm, P.J. Driscoll, B. Klimack, and M.L. McGinnis, "Networked unattended ground sensor fields: Tradeoff study and configuration rules methodology," USMA Operations Research Center Technical Report, West Point, New York, Aug. 2002.

[6] R.H. Myers and D.G. Montgomery, Response Surface Methodology, John Wiley & Sons, New York, 1987.

[7] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," Proc. IEEE Infocom Conference, pp.1293–1303, March 2003.

[8] G. Wang, G. Cao, T.L. Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," IEEE Infocom Conference, pp.2302–2312, March 2005.

[9] C. Sharp, S. Schaffert, A. Woo, N. Sastry, C. Karlof, S. Sastry, and D. Culler, "Design and implementation of a sensor network system for vehicle tracking and autonomous interception," European Workshop on Sensor Networks, pp.93–107, 2005.

[10] Y. Xu and X. Yao, "A GA approach to the optimal placement of sensors in wireless sensor networks with obstacles and preferences," Proc. 3rd IEEE Consumer Communications and Networking Conference, pp.127–131, 2006.

[11] C. Zhao, Z. Yu, and P. Chen, "Optimal deployment of nodes based on genetic algorithm in heterogeneous sensor networks," International Conference on, Wireless Communications, Networking and Mobile Computing, 2007, pp.2743–2746, Sept. 2007.

[12] N. Srour, "Unattended ground sensors a prospective for operational needs and requirements," Sensor and Electron Devices Directorate, U.S. Army Research Laboratory Technical Report, Oct. 1999.

[13] J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975.

[14] C. Anderson, K. Jones, and J. Ryan, "A two-dimensional genetic algorithm for the Ising problem," Complex Systems, vol.5, pp.327–333, 1991.

[15] D. Gale and L.S. Shapley, "College admissions and the stability of marriage," American Mathematical Monthly, vol.69, pp.9–14, 1962.

[16] T.N. Bui and B.R. Moon, "Genetic algorithm and graph partitioning," IEEE Trans. Comput., vol.45, no.7, pp.841–855, 1996.

Jae-Hyun Seo received the B.S. degree in Computer Engineering from KEDI, Korea in 2005 and the M.S. degree in Computer Science from Kwangwoon University, Seoul, Korea in 2008. He is now an assistant manager at Korea Association of Property Appraisers. His research interests include wireless sensor networks, genetic algorithm, sensor deployment, and network security.

Yong-Hyuk Kim received the B.S. degree in computer science and the M.S. and Ph.D. degrees in computer science and engineering from Seoul National University (SNU), Seoul, in 1999, 2001, and 2005, respectively. From March 2005 to February 2007, he was a postdoctoral scholar at SNU and also a research staff member in the Inter-University Semiconductor Research Center, SNU. Since 2007, he has been an assistant professor in the Department of Computer Science and Engineering, Kwangwoon University, Seoul, Korea. He has been a reviewer for several journals of the IEEE Computer Society since 2003. His research interests include algorithm design/analysis, discrete mathematics, optimization theory, combinatorial optimization, evolutionary computation, operations research, and data/Web mining.

Hwang-Bin Ryou received the B.S. degree in Electronic Engineering from Inha University in 1968 and his M.S degree in Electronic Engineering from Yonsei University in 1975. He received his Ph.D. degree in Electronic Engineering from Kyunghee University in 1984. He worked as a manager with GoldStar Electronics, South Korea. Since 1981, He has been a Professor in the Department of Computer Science and Engineering, Kwangwoon University, Seoul, Korea. His interesting area lies in security, WSN, RFID, and ubiquitous computing.

Si-Ho Cha received the B.S. degree in Computer Science from Sunchon National University in 1995. He received the M.S. and Ph.D. degrees in Computer Science from Kwangwoon University, Seoul, Korea in 1997 and 2004, respectively. From 1997 to 2000, he worked for Daewoo Telecom as a Senior Researcher and was a Research-in-Charge at WarePlus, Korea. He had been an Adjunct Professor in the Department of Computer Engineering, Sejong University until 2006. He is now a Research Professor in the Department of Information and Communication Engineering, Sejong University, Seoul, Korea. His research interests include network management, wireless sensor networks, wireless mesh networks, and ubiquitous computing.

Minho Jo received B.S. degree in Industrial Engineering, Chosun Univ., South Korea and his Ph.D. in Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA in 1994. He worked as a staff researcher with Samsung Electronics, South Korea and was professor of School of Ubiquitous Computing and Systems, Sejong Cyber University, Seoul, South Korea. He is now a Research Professor of Graduate School of Information Management and Security, Korea Univ., Seoul, South Korea. Prof. Jo is Executive Director of Korean Society for Internet Information (KSII) and Board of Trustees of Institute of Electronics Engineers of Korea (IEEK). He is Editor-in-Chief and Chair of the Steering Committee of KSII Transactions on Internet and Information Systems, General Chair of International Ubiquitous Conference, and Co-chair of International Conference on Ubiquitous Convergence Technology. Dr. Jo is Editor of Journal of Wireless Communications and Mobile Computing, and Associate Editor of Journal of Security and Communication Networks published by Wiley and Sons, respectively. He is Associate Editor of Journal of Computer Systems, Networks, and Communications published by Hindawi. He serves as Chairman of IEEE/ACM WiMax/WiBro Services and QoS Management Symposium, IWCMC 2008. He is Technical Program Committee of IEEE ICC 2008, 2009 and IEEE GLOBECOM 2008, 2009, respectively. His interesting area lies in wireless ad-hoc sensor networks and wireless mesh networks, RFID, security in communication networks, machine intelligence in communications, and pervasive computing/networking.

View publication stats