ICIC Express Letters
Volume **10**, Number **6**, June **2016**

ICIC International ©2016 ISSN 1881-803X
pp. 1451–1456

AN R-BASED GENETIC ALGORITHM FOR PLACEMENT OF SURVEILLANCE CAMERAS

Hyun-Lim Yang and Tai-Woo Chang*

Department of Industrial and Management Engineering
Kyonggi University
154-42, Gwanggyosan-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do 16227, Korea
*Corresponding author: keenbee@kgu.ac.kr

Received December 2015; accepted March 2016

Abstract. *Video surveillance with CCTV is widely used for structure monitoring. In this study, we deal with a placement problem of multi-type cameras. We suggest an R-based genetic algorithm to find a feasible solution of surveillance cameras placement problem that satisfies required coverage with minimum cost. We considered 2-dimensional floor plan of structure and available installation area. We could find an approximate solution with required coverage.*
Keywords: Optimal camera placement, CCTV deployment, Video surveillance, Genetic algorithm

1. **Introduction and Related Work.** With performance improvement and diversification of closed-circuit television (CCTV) cameras, video surveillance with CCTV is widely used for many fields. Recently, it has been used not only for crime prevention, traffic control, inside surveillance, but also for structure monitoring and safety management [1]. Many researches have studied to improve the performance of video surveillance. However, efficient placement of surveillance cameras was not considered enough. Yoon and Kim [2,4] and Xu and Sahni [3] suggested optimal solution in sensor deployment problem with genetic algorithm and approximation algorithm. And Seo et al. [6] suggested one in wireless sensor deployment problem with hybrid genetic algorithm. One difference of deployment problem between sensor networking and video surveillance is the fact that CCTV has limited visible range and a direction. Also, it has more constraints such as occlusion. It is difficult to relocate the cameras after installing them, and it costs a lot for maintenance. In this study, we deal with a problem of placement of surveillance cameras around the large-structure.

The goal of this study is to find an efficient solution of the problem with R-based genetic algorithm (GA). R is a programming language for statistical computing and data analytics. The objective of the problem is to minimize total cost of multi-type cameras with minimum coverage. We considered their cost, visible range, angle and direction of view. We considered 2-demensional floor plan.

Chang [1] suggested a linear programming model for the placement problem of single-type cameras and a simulation tool for relocating the cameras. This study expands the study by dealing with multi-type cameras and develops a genetic algorithm with R to increase availability. Also we use real number coordinate for representation of solutions. A large-scale placement problem with multi-type cameras is one of NP-hard problems. So, we suggest a heuristic method using GA.

This paper is organized as follows. Section 2 defines the camera placement problem with its variables and algorithm procedure. Section 3 introduces our R-based GA and its experimental results. Finally, we conclude the paper in Section 4.

2. **Problem Modeling and Algorithm.** The main variables of camera placement problem are effective sight range (R), pan angle (θ), sight direction (d_j), and cost of camera (c_l). We assign total number of cameras (NC) and number of directions (ND) in advance. Also, to evaluate coverage, we make surveillance spots (t_k) on exterior line segment at regular interval. With all spots (NT), we test each point with camera location (x_i) in terms of range, angle, and direction. Camera location, direction and target point have x- and y-axis values and they form a matrix. In this study, the candidate points of camera location and target monitoring points are spaced apart at regular intervals.

As shown in Figure 1, if a camera located in x_i and it has range R, angle θ with direction d_j, then it covers a spot t_k. If total covered points exceed 80% of NT, we can say that it has 80% of coverage. In Figure 1, one camera has been used to cover point t_k. This assessment will do $NC \times NT$ times, and each assessment requires occlusion test. In this problem, i is 1 to NC, j is 1 to ND, k is 1 to NT, and l is 1 to the number of camera types.

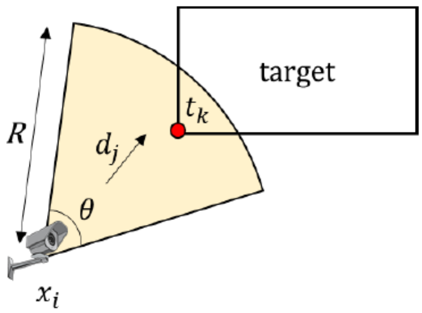

FIGURE 1. Target surveillance of a specific camera at its location

2.1. **Visibility test.** It is necessary to test whether t_k is covered by a camera, x_i of which has range R. It is also necessary to test whether t_k is covered by a camera, x_i of which has pan angle θ and a direction d_j, and to test whether t_k is concealed or not from other exterior segments (S). For all camera location points, if each point returns *TRUE* at all tests, it becomes a covered point. According to this process, we can calculate coverage.

2.2. **Cover Range Test.** It is necessary to test whether t_k is covered by a camera, x_i of which has range R. We denoted x-axis value of x_i by x_{i1} and t_k by t_{k1}, and y-axis value of x_i by x_{i2} and t_k by t_{k2}. The pseudo code of Cover Range Test algorithm is as follows.

for $i = 1$ **to** NC **do**
  for $k = 1$ **to** NT **do**
    if $\left(\sqrt{(t_{k_1} - x_{i1})^2 + (t_{k_2} - x_{i2})^2} < R\right)$ **then**
      return *TRUE*
    else return *FALSE*

2.3. **Cover Angle Test.** It is also necessary to test whether t_k is covered by a camera, x_i of which has pan angle θ and a direction d_j. We denoted the x-axis value of d_j by d_{j1}, y-axis value of d_j by d_{j2}. The algorithm of Angle Test is as follows.

for $i = 1$ **to** NC **do**
  for $k = 1$ **to** NT **do**
    if $\left(\frac{d_{j1}\times(t_{k1}-x_{i1})+d_{j2}\times(t_{k2}-x_{i2})}{\sqrt{d_{j1}^2+d_{j2}^2}+\sqrt{(t_{k1}-x_{i1})^2+(t_{k2}-x_{i2})^2}} > \cos\theta\right)$ **then**
      return *TRUE*
    else return *FALSE*

2.4. **Occlusion Test.** It is also necessary to test whether t_k is concealed or not from other exterior segments (S). We denoted 2-endpoints of S by S_a and S_b. We denoted the x- and y-axis values of them by S_{a1}, S_{a2}, S_{b1}, S_{b2} each.

Consider a segment S_{ik} from x_i to t_k. If other exterior segments intersect S_{ik}, then t_k will be a concealed point. We can represent equations of segment S and S_{ik} as S_a, S_b and x_i, t_k with notation u and v.

$$S(u) = (1-u)S_a + uS_b$$
$$S_{ik}(v) = (1-v)x_i + vt_k$$

Then, we can find u and v by a simultaneous equation of x- and y-coordinate. Hence, t_k is an invisible point if $0 < u < 1$ and $0 < v < 1$. The pseudo code of Occlusion Test is as follows.

for $i = 1$ **to** NC **do**
  for $k = 1$ **to** NT **do**
    if ($0 < u < 1$ **and** $0 < v < 1$) **then**
      return *FALSE*
    else return *TURE*

2.5. **Cover point estimation.** For all camera location points, if each point returns *TRUE* at all tests, it becomes a covered point. Therefore, the pseudo code of cover point estimation is as follows. According to this process, we can calculate coverage.

for $k = 1$ **to** NT **do**
  if (*Cover Range Test & Cover Angle Test & Occultation Test*)
    then t_k is *covered point*
  else t_k is *not covered*

2.6. **Algorithm procedure.** As mentioned before, we used GA to solve this problem. It finds best solution from multiple solutions which called population, and preserves elite solution with crossover and mutation. Through a number of generations, bad solution will fall behind and elite solution will derive an approximate optimal solution.

Algorithm procedure is as follows. First, input camera specification, directions, and floor plan. Second, set parameter of GA and minimum coverage. Next, generate target point t_k. With initial population, calculate its quality. Select elite solutions for next generation, and produce offspring with crossover and mutation. GA repeats it until the generation ends.

3. **Experiments and Genetic Algorithm.** We define a floor plan of the camera placement problem as in Figure 2. Available area for deployment is coordinate of (1000m × 300m). All x_i are spaced apart at least 10m and t_k have 5m regular intervals. 8 directions are available. Three types of cameras are used: 200, 180 and 160 for range, 80, 140 and 180 for angle, and 10, 12 and 10 for cost. We set 0.8 as minimum coverage and set $NC \times \max\{(c_l)\}$ as a cost value for prohibiting a solution, which could not satisfy the minimum coverage, not to be selected.

Environmental values of GA are as follows. Population is 100, mutation rate is 0.1, and crossover rate is 0.75. GA will end with 2000 generations.

3.1. **Generation of initial solution.** We define a function $f_m(NC)$, which makes an initial solution for GA. Available area for placement is outside region of blue dotted line in Figure 2. And the type and direction of camera are determined with random uniform function.

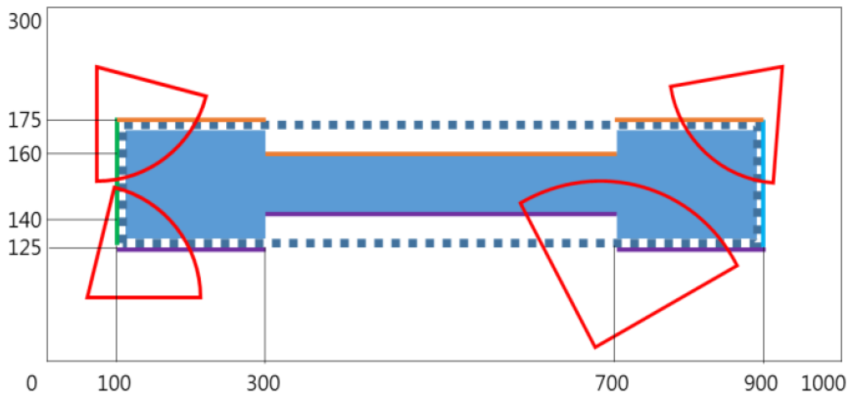

FIGURE 2. Two-dimensional floor plan of structure and available area for camera placement

3.2. **Selection.** For selection, we used roulette wheel. We set best solution's fitness to have five times more of share than the worst solution's fitness for rapid convergence.

3.3. **Crossover.** We used 2-points crossover. After generating two random integer values between 1 and NC, we rearranged them in ascending order of (rn_1, rn_2). In parent solution A and B, we made offspring A′ with A for the first row to rn_1 row, B for $rn_1 + 1$ row to rn_2 row, A for $rn_2 + 1$ row to NC row. And we made offspring B′ with reverse combination. Crossover operation will run only if a random uniform number between 0 and 1 is lower than the crossover rate.

3.4. **Mutation.** For mutation, we made totally new solution. If a random uniform number between 0 to 1 is lower than the mutation rate, this solution will be replaced with new solution by $f_m(NC)$. With this operation, we tried to keep the probability open for a future intervention of solution which cannot be generated by crossover operation.

3.5. **Replacement.** We replaced A′ and B′ with A and B in the next generation. New A′, B′ and other non-selected solutions would be estimated by the cost. And an elite solution will be selected by roulette wheel again, and GA will go on in this procedure.

In this study, to solve this problem, we used R. All variables are saved in arrays to be able to call or operate them easily. Especially, we defined one generation data of GA as a large array, such as right-side exemplary one in Figure 3, to be able to access all components of camera specifications. Each row of 'thisGen' matrix means single gene, and 'thisGen' consists of one hundred rows, which are the same as population. The first

```
> caminfo
  angle range cost
1    80   200   10
2   140   180   12
3   180   160   10
4     0     0    0
```

(a)

```
> thisGen[,,1]
     [,1] [,2] [,3] [,4] [,5]
[1,]  940  270    3    4    4
[2,]  180  110    2    1    3
[3,]  290   50    5    4    3
[4,]  950  230    5    2    4
[5,]  190   80    2    3    3
[6,] 1000   20    4    2    4
[7,]  480  120    8    4    3
  ⋮     ⋮    ⋮    ⋮    ⋮    ⋮
```

(b)

FIGURE 3. Expression of algorithm variables: (a) camera specifications of each type; (b) 100 sets of genes with coordinate values and camera features

```
> bestsolution
    V1  V2 V3 V4 V5
2  910 200  5  3  4
3  700  40  2  2  3
4  710  70  6  3  3
6  670 290  2  2  2
8  990 240  8  2  4
12  90  40  2  2  1
13 260  90  8  2  3
14 680 200  5  3  2
16  70 130  2  3  1
18 710  50  4  1  3
19 100  10  7  3  1
20  80 120  1  3  1
21  50 280  7  3  1
22   0 180  6  1  1
```

Figure 4. An approximate solution of GA

two columns indicate x- and y-axis coordinate values of each population, and the rest of them indicate the camera features of the population.

With a constraint of minimum coverage 0.8, we got a solution of cost 150. As it is an NP-hard problem, which requires abundant resources to solve it through mathematical approach, a feasible solution has significance to this problem. Total number of used cameras was 14; they are 2 cameras of type 1, 5 cameras of type 2, 8 cameras of type 3, and their coverage was 0.8191. As shown in Figure 4, we could obtain location of 14 cameras. The first two columns of 'bestsolution' matrix mean x- and y-axis coordinate values. The third column means direction of camera, and the fourth column means type of camera. And this feasible solution could be visualized and improved by a simulation tool of the previous study [1].

4. **Conclusion and Future Research.** For the purpose of structure monitoring, cost minimization with required coverage is a critical issue. Since camera placement problem is an NP-hard problem, it is very difficult to find optimal solution in multi-type camera placement though it might be possible to solve a problem of single-type cameras with linear programming model.

In this study, we suggested an approximate solution of genetic algorithm for a large-size problem which cannot be solved by mathematical modeling. We could deal with more practical problems of using multi-type cameras and found a surpassing placement rule. As solution tools we used R and a freeware RStudio for compatibility with other programs. In addition, using R made it easy to represent variables and solutions as matrices and to understand variables after end of GA.

Some components in our model may not be practical and should be elaborated with real world problems and more sophisticated algorithms.

Acknowledgement. This research was supported by a grant (14SCIP-B065985-02) from Smart Civil Infrastructure Research Program funded by Ministry of Land, Infrastructure and Transport (MOLIT) of Korea government and Korea Agency for Infrastructure Technology Advancement (KAIA).

REFERENCES

[1] T.-W. Chang, Optimization and simulation tool for placement of surveillance cameras, *ICIC Express Letters, Part B: Applications*, vol.6, no.12, pp.3183-3188, 2015.

[2] Y. Yoon and Y. H. Kim, An efficient genetic algorithm for maximum coverage deployment in wireless sensor network, *IEEE Trans. Cybernetics*, vol.43, no.5, pp.1473-1483, 2013.

[3] X. C. Xu and S. Sahni, Approximation algorithm for sensor deployment, *IEEE Trans. Computers*, vol.56, no.12, pp.1681-1695, 2007.

[4] Y. Yoon and Y. H. Kim, Genetic algorithms for maximizing the coverage of sensor deployment, *Korea Intelligent Information System Society*, vol.20, no.3, pp.406-412, 2010.
[5] K. Chakarabarty, S. S. Iyengar, S. Qi and E. Cho, Grid coverage for surveillance and target location in distributed sensor networks, *IEEE Trans. Computers*, vol.51, no.12, pp.1448-1453, 2002.
[6] J. H. Seo, Y. H. Kim, H. B. Ryou, S. H. Cha and M. Jo, Optimal sensor deployment for wireless surveillance sensor networks by a hybrid steady-state genetic algorithm, *IEEE Trans. Communications*, vol.E91-B, no.11, pp.3534-3543, 2008.