

# 최대 커버리지 센서 배치를 위한 유전 알고리즘

## Genetic Algorithms for Maximizing the Coverage of Sensor Deployment

윤유림\* · 김용혁\*\*

Yourim Yoon and Yong-Hyuk Kim

\* 서울대학교 컴퓨터공학부

\*\* 광운대학교 컴퓨터소프트웨어학과

### 요 약

본 논문에서는 실제 세계에서 센서를 배치할 때 발생하는 최적화 문제인 최대 커버리지 센서 배치 문제를 정의하고 문제의 해 공간의 특성을 분석하였다. 또한 최대 커버리지 센서 배치 문제의 좋은 해를 얻기 위해 유전 알고리즘을 설계하고 그 우수성을 비교 실험을 통해 보였다. 이 문제에 유전 알고리즘을 적용할 때 중요하게 고려되어야 할 부분은 평가 함수를 어떻게 구현하느냐 인데 몬테카를로법을 통해 해결할 수 있었다. 유전 알고리즘의 몬테카를로법을 이용한 평가 부분에서 샘플 생성 횟수를 조절함으로써 동일한 성능을 내면서 계산 시간을 크게 줄일 수 있었다.

**키워드 :** 유전 알고리즘, 최대 커버리지, 센서 배치, 몬테카를로법

### Abstract

In this paper, we formally define the problem of maximizing the coverage of sensor deployment, which is the optimization problem appeared in real-world sensor deployment, and analyze the properties of its solution space. To solve the problem, we proposed novel genetic algorithms, and we could show their superiority through experiments. When applying genetic algorithms to maximum coverage sensor deployment, the most important issue is how we evaluate the given sensor deployment efficiently. We could resolve the difficulty by using Monte Carlo method. By regulating the number of generated samples in the Monte Carlo evaluation of genetic algorithms, we could also reduce the computing time significantly without loss of solution quality.

**Key Words :** Genetic algorithms, maximum coverage, sensor deployment, Monte Carlo method

## 1. 서 론

센서의 성능이 점점 발전함에 따라 무선 센서 네트워크는 자연 재해 대처, 서식지 감시, 특정 지역의 기후에 대한 연구, 표적에 대한 감시, 응급 구조, 의료 등 여러 분야에 응용되어 왔다. 요즘은 감시 시스템을 구축하는 데 자동화된 센서 배치가 필요한 경우가 많아졌다. 특히 표적의 움직임을 감시하는 지상 작전에서는 자동으로 센서를 배치하고 관리하는 시스템이 필수적이다. 최적화된 센서 배치는 사람의 노동력과 시간을 절약해주고 표적의 움직임에 대한 정확한 정보를 제공해주며 동적인 전장(dynamic field)에서 빠르게 전략을 수정할 수 있도록 해 준다. 따라서 센서들을 최적으로 잘 배치하면 표적에 대해 더 명확히 파악할 수 있게 된다.

지금까지 어떤 일정한 목표에 따라 센서를 잘 배치하기 위한 문제에 관한 연구는 많이 있었다. 그러나 주로 실제 세계에서 나타나는 문제들을 다루고 있어서 센서들을 잘 배치하려는 전체적인 뼈대는 같지만 목적 함수나 대상으로 하는 환경들은 조금씩 다른 점들이 있다. 커버리지를 최대화하거나 원하는 지점을 모두 커버하는 등 목적 커버리지를 조금 더 잘 달성하기 위한 센서들의 배치를 고안한 연구들도 있었고[1-3], 목표하는 커버리지를 달성하면서 센서의 개수와 같은 배치에 드는 비용을 최소화하는 것을 목적으로 한 연구들도 있었다[4-7]. 그 외에 다양하고 복잡한 환경과 조건, 목적에 따른 센서 배치를 최적화하기 위한 방법들을 연구한 논문들도 있었다[8-11]. 특히 실시간으로 이동 가능한 센서들의 변화하는 환경에 대한 동적인 배치를 다룬 연구들[2,12]과 무선 센서 네트워크에서 만족스러운 수준의 커버리지와 연결성을 유지하는 방법들에 대한 연구도 있었다[6,13]. 센서를 배치해야 하는 대상 영역을 연속 공간이 아닌 단순화된 그리드(grid)로 표현한 연구들도 있었다[3,5,7]. 실제 세계의 문제의 해를 직접 풀어내지는 않았지만 센서들의 배치와 관련된 문제들을 이론적으로 분석하고 정리한 연구들[4,14,15]도 있었다.

본 논문에서는 일정한 넓이의 지역을 주어진 종류와 개

접수일자 : 2010년 3월 11일

완료일자 : 2010년 5월 17일

+ 교신저자

감사의 글 : 이 논문은 2008년 정부(교육과학기술부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2008-331-C00048).

커버해야 하는 영역  $A = [0,100] \times [0,100]$   
 센서 1 : 탐지 거리 20, 2개  
 센서 2 : 탐지 거리 10, 3개  
 센서 3 : 탐지 거리 5, 4개

해의 표현 : 9개 2차원 실수 좌표의 나열

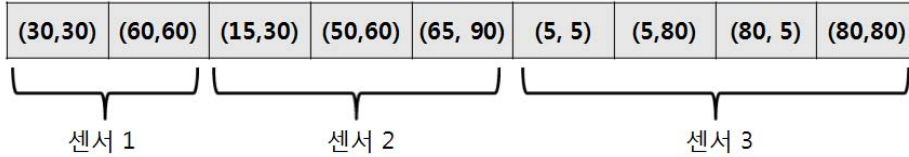


그림 1. 해의 표현

Fig. 1. Representation of a solution

수의 센서들로 최대한 많이 커버하는 것을 목적으로 하는 문제를 고려해 본다. 보통 센서 배치에 관한 연구들에서 비용 문제로 인해 센서들을 무한히 많이 생산할 수는 없으며 센서의 탐지 범위를 넓히기 위해 최대한 넓은 범위를 커버하는 것이 효율적이라는 면에서 위 문제는 실제 세계에 적용 분야가 무척 많은 문제라 할 수 있다. 이 논문에서는 위 문제를 명확하게 정의하고 좋은 해를 찾기 위해 유전 알고리즘을 이용한 효율적인 알고리즘을 모색해 본다.

## 2. 최대 커버리지 센서 배치 문제

### 2.1 문제 정의

본 논문에서는 최대 커버리지 센서 배치 문제를 다음과 같이 형식화하였다.  $k$  종류의 센서들이 주어져 있다. 각 종류의 센서들은 자신과 일정 거리 이내의 영역을 커버할 수 있으며 종류마다 커버할 수 있는 거리가 다르다. 사용할 수 있는 센서들의 개수는 전체  $n$  개이며 각 종류마다  $n_1, n_2, \dots, n_k$  개씩의 센서가 있다. (즉,  $n_1 + n_2 + \dots + n_k = n$  이다.) 최대 커버리지 센서 배치 문제의 목표는 2차원 유클리드 공간에 커버해야 하는 일정 영역이 주어졌을 때, 주어진 센서들로 커버할 수 있는 영역의 넓이, 즉, 커버리지 (coverage)를 최대화하도록 2차원 좌표 상에 센서들의 위치를 결정하는 것이다.

최대 커버리지 센서 배치 문제와 비슷한 문제들이 다양한 형태로 많이 연구되고 있다. 그러나 본 논문에서 다루는 문제는 커버하는 영역의 넓이를 최대화하는 것을 목적으로 하는 문제로, 기존의 정해진 영역에 원을 빈틈없이 채우는 원 패킹(circle packing) 문제[14]나 정해진 영역을 모두 커버하면서 그 비용을 최소화하는 점 덮기(point coverage) 문제[4]와는 다르다.

최대 커버리지 센서 배치 문제는 실제 세계에서 많은 응용 사례를 찾을 수 있다. 이동 통신 기지국을 설치할 때 최대한 많은 영역을 커버할 수 있도록 기지국의 위치를 결정해야 하는데 이는 최대 커버리지 센서 배치 문제의 응용에 해당한다. 또한 보안 목적으로 센서들을 설치할 때도 역시 최대 영역을 탐지할 수 있도록 해야 하는데 이런 문제도 역시 최대 커버리지 센서 배치 문제를 응용하여 해결할 수 있다. 이외에도 비슷한 응용 사례들을 많이 발견할 수 있을 것이다.

최대 커버리지 센서 배치 문제의 특징은 해가 실수 좌표로 표현된다는 것이다. 실수 최적화 문제 중 하나로 생각할 수 있으며 이산 공간과 다른 실수 공간의 특성상, 해의 개수가 무한하다는 점에서 좋은 해를 찾기가 그만큼 더 어렵다고 볼 수 있다.

### 2.2 가능한 해의 표현

배치해야 하는 전체 센서의 개수는  $n_1 + n_2 + \dots + n_k$  이 되고 각 센서마다 배치된 좌표를 표시해야 한다. 가장 일반적으로 생각할 수 있는 해의 표현으로는 각 센서들에 인덱스를 붙인 후 인덱스 순서로 좌표를 1차원으로 나열하는 것이다. 이러한 표현은  $n_1 + n_2 + \dots + n_k$  개의 2차원 좌표들의 1차원 배열이 될 것이다. 그림 1에 문제에 대한 해의 표현 중 한 예가 나와 있다.

### 2.3 해의 표현과 몫 공간(quotient space)과의 관계

좌표들의 1차원 배열로 해를 표현했을 때 생각해볼 수 있는 점이 같은 종류의 센서들끼리는 좌표가 바뀌어도 커버하는 영역에 전혀 영향을 주지 않는다는 점이다. 그러므로 해의 표현은 다르지만 실제로는 같은 해로 볼 수 있다. 따라서 해의 인코딩 공간과 실제 해공간 사이의 차이가 발생한다. 이것을 수학적으로 더 명확하게 해석할 수 있다. 해의 인코딩 공간을  $X$ ,  $X$ 의 두 원소  $x$ 와  $y$ 를 각각  $(x_1, x_2, \dots, x_n)$ 과  $(y_1, y_2, \dots, y_n)$ 라 하자. 여기서  $x_i$ 와  $y_i$ 는 기호 하나로 썼지만 사실은 2차원 좌표를 나타낸다.  $X$ 의 원소들 간에 동치관계  $\sim$ 를 다음과 같이 정의한다.

**정의 1.** 모든  $i = 1, 2, \dots, k$ 에 대해

$$\sigma_i((x_{n_1+\dots+n_{i-1}+1}, x_{n_1+\dots+n_{i-1}+2}, \dots, x_{n_1+\dots+n_{i-1}+n_i})) = (y_{n_1+\dots+n_{i-1}+1}, y_{n_1+\dots+n_{i-1}+2}, \dots, y_{n_1+\dots+n_{i-1}+n_i})$$

이 되는 순열(permutation)  $\sigma_i \in \Sigma_{n_i}$ 가 존재하면  $x \sim y$ 이다.

여기서  $k$ 는 센서 종류의 수,  $n_i$ 는  $i$ 번째 종류 센서의 개수,  $\Sigma_{n_i}$ 는 크기  $n_i$ 의 모든 순열들의 집합을 나타낸다.  $(x_{n_1+\dots+n_{i-1}+1}, x_{n_1+\dots+n_{i-1}+2}, \dots, x_{n_1+\dots+n_{i-1}+n_i})$ 는  $x$ 에서  $i$ 번째 종류 센서들의 좌표들만을 떼어서 나열한 것을 뜻하게 된다. 예를 들어, 그림 2의 해  $x$ 와  $y$ 는 동치 관계다. 어

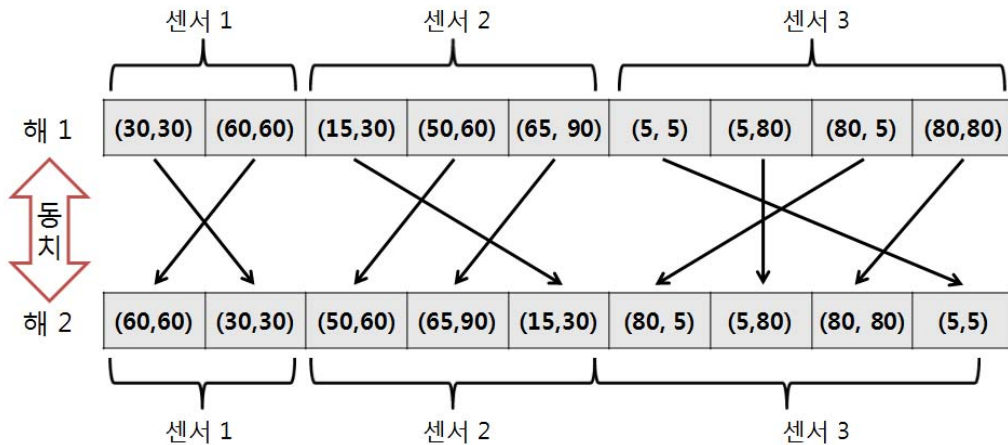


그림 2. 동치 관계에 있는 해의 예  
Fig. 2. An example of equivalent solutions

면 원소  $x$ 와 동치 관계에 있는 원소들의 개수는  $n_1!n_2! \dots n_k!$  개가 될 것이다. 그러면 실제 해의 공간은  $X$ 의 어떤 원소  $x$ 의 동치류(equivalence class)  $[x]$ 들의 집합인 몫(quotient) 공간  $X/\sim$ 로 볼 수 있다.

그러나 실제 해 공간이  $X/\sim$ 임에도 해의 표현을 정확히  $X/\sim$ 의 원소와 일대일 대응이 되도록 만들기가 어려우므로 표현 공간 상에서 해를 찾는 것이 실제 해 공간에서 해를 찾는 것과 다른 양상을 보일 수 있다. 이러한 문제점을 해결하기 위해서는 표현 공간에서의 해의 탐색을 실제 해 공간에서의 해의 탐색과 유사한 방향으로 진행되도록 하기 위한 방법을 모색할 필요가 있다.

### 3. 최대 커버리지 센서 배치 문제를 위한 유전 알고리즘

유전 알고리즘은 자연 생태계의 진화 과정과 유전 법칙을 모방한 확률적 최적해 탐색 기법으로 대부분 정해진 수의 해로 구성되는 해 집단을 갖는다. 이 해 집단으로부터 새로운 자식 해를 만들어 내는데 각각의 해는 선택, 교차, 변이의 단계를 거쳐 만들어진다. 이렇게 만들어진 새로운 해들은 해 집단 내의 부모 해들과 대치된다. 이러한 과정을 주어진 정지 조건이 만족될 때까지 수행한 후, 해 집단에 남은 해 중 가장 좋은 해를 최종 해로 삼는다.

#### 3.1 평가 함수

최대 커버리지 센서 배치의 평가 값은 배치된 센서들로 커버 가능한 전체 면적이다. 결국 각 센서의 위치를 중심으로 하고 반지름이 센서의 탐지 거리인 원들의 면적 합이 되는데 각 원들이 서로 겹치기도 하므로 원의 면적 공식 같은 수식으로는 구하기가 쉽지 않다. 따라서 본 논문에서는 몬테카를로법으로 주어진 해를 평가한다.

몬테카를로법이란, 시뮬레이션 테크닉의 일종으로, 구하고자 하는 수치의 확률적 분포를 반복 가능한 실험의 통계로부터 구하는 방법을 말한다. 확률변수에 의거한 방법이기 때문에 1949년 스타니스와프 울람(Stanisław Ulam)이 모

나코의 유명한 도박의 도시 몬테카를로(Monte Carlo)의 이름을 본떠서 명명하였다. 간단하면서도 유명한 예로, 몬테카를로법을 이용한 파이( $\pi$ )의 계산법이 있다. 먼저 정사각형 안에 한 꼭지점을 중심으로 사분원을 한 개 그린다. 이때 정사각형의 전체 넓이를 1이라고 하면 원의 넓이는  $\pi/4$ 가 된다. 이제 컴퓨터로 난수를 발생하여 무작위로 정사각형 내부에 점을 찍는다. 그리고 정사각형의 꼭지점과의 거리를 계산하여 점이 사분원의 내부에 있는지 외부에 있는지를 판단한다. 예를 들어 전체  $m$  개의 점을 찍었다고 할 때 이 중  $n$  개가 사분원의 내부에 있었다면 두 숫자의 비율, 즉,  $n/m$ 의 값은 넓이의 비인  $\pi/4$ 에 근접하리라고 예측할 수 있다. 이 값을 더 많은 점을 찍어 실험할수록 정밀해진다. 이와 같이 많은 수의 샘플을 바탕으로 통계 자료를 얻어 그 자료로부터 역산하여 어떤 특정한 수치나 확률분포를 구하는 방법을 몬테카를로법이라고 하는데 특성상 통계자료가 많을수록, 입력 값의 분포가 고를수록 결과의 정밀성이 보장된다는 것을 알 수 있다.

최대 커버리지 센서 배치 문제의 평가 함수도 몬테카를로 방법을 이용한 파이의 계산법과 유사하게 구현할 수 있다. 문제에서 주어진 영역의 전체 넓이를  $A$ 라고 하자. 어떤 해  $x$ 의 커버리지를 근사적으로 구하기 위해 난수를 발생하여 무작위로 전체 영역 내에 점을 찍는다. 그리고 해  $x$ 의 각 센서에 대해 센서의 위치로부터 그 점과의 거리를 계산해 탐지거리 내에 있는지 판단한다. 만약 어느 한 센서에 대해서 무작위로 찍은 점이 탐지거리 내에 있다면 커버리지 내부의 점이 된다. 전체  $m$  개의 점을 찍었다고 할 때, 이 중  $n$  개가 커버리지 내부의 점이라면  $n/m$ 의 값이 전체 넓이에 대한 커버리지 넓이의 비에 근접하리라고 예측할 수 있다. 따라서 커버리지 넓이를  $A \times n/m$ 로 근사할 수 있을 것이다. 이러한 방법에서 생각해야 할 점은 “얼마나 많은 점을 찍어야 통계적 안정성을 확보할 수 있을까”이다. 이것을 알아보기 위해 본 논문에서는 전체 영역  $A$ 를  $[0,100] \times [0,100]$ 의 정사각형으로 가정하고 그 안에 중심이 (50,50), 탐지거리가 40인 센서 한 개를 배치한 가장 간단한 경우에 대해 몬테카를로법을 수행해 보았다. 이때 수학적 계산에 따른 면적은 약 5027 ( $\approx \pi \times 40^2$ ) 정도이다. 그림 3은 무작위로 찍은 점의 개수의 따른 평가 함수 값의 변

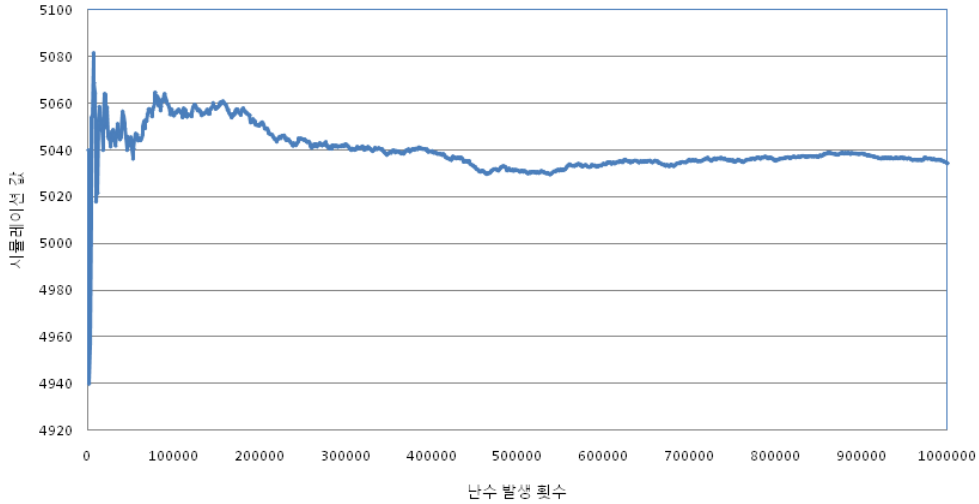


그림 3. 몬테카를로 시뮬레이션  
Fig. 3. Monte Carlo simulation

```

Extended_Box_Crossover( $p_1, p_2, o$ )
// 부모 해  $p_1 = ((x_1^1, y_1^1), (x_2^1, y_2^1), \dots, (x_n^1, y_n^1))$  과  $p_2 = ((x_1^2, y_1^2), (x_2^2, y_2^2), \dots, (x_n^2, y_n^2))$ 
// 자식 해  $o = ((x_1^0, y_1^0), (x_2^0, y_2^0), \dots, (x_n^0, y_n^0))$ 
for  $i \leftarrow 0$  to  $n$ 
     $x_i^0 \leftarrow [\min(x_i^1, x_i^2) - \alpha \times |x_i^1 - x_i^2|, \max(x_i^1, x_i^2) + \alpha \times |x_i^1 - x_i^2|]$  사이의 난수;
     $y_i^0 \leftarrow [\min(y_i^1, y_i^2) - \alpha \times |y_i^1 - y_i^2|, \max(y_i^1, y_i^2) + \alpha \times |y_i^1 - y_i^2|]$  사이의 난수;
     $(x_i^0, y_i^0)$ 가 범위를 벗어난 경우 조정;
    
```

그림 4. 확장 상자 교차 연산자의 의사 코드  
Fig. 4. Pseudo-code of extended box crossover

화를 보여준다. 그림에 따르면 점의 개수가 대략 300,000 개 이상이 되면 시뮬레이션 값이 정답에 가까워지면서 안정적이게 된다. 점을 많이 찍을수록 시뮬레이션 값이 안정적이지만 그만큼 또 수행 시간이 오래 걸리기 때문에 무조건 많은 점을 찍는 것이 좋은 방법이라 할 수는 없다.

### 3.2 사용한 유전 알고리즘

최대 커버리지 센서 배치 문제에서 커버해야 하는 전체 영역은 다양한 모양으로 나타날 수 있지만 본 연구에서는 기본적인 2차원 정사각형 모양인  $[0,100] \times [0,100]$ 을 가정하기로 한다.

사용한 유전 알고리즘의 기본 틀은 다음과 같다. 인구수가  $N$ 이라고 하자. 이  $N$  개의 염색체를  $N/2$  개의 임의의 쌍으로 짝을 짓는다. 이 임의로 만들어진  $N/2$  개의 쌍에 교차 연산자와 변이 연산자를 차례로 적용해  $N/2$  개의 자식 해를 얻는다. 이렇게 생성된  $N/2$  개의 자식 해와  $N$  개의 기존 부모 해 중 품질이 가장 좋은  $N$  개가 다음 세대를 구성한다. 본 논문에서는 50의 인구수를 사용했고 1,000 세대를 수행한 후 이제까지 찾은 해 중 가장 좋은 해를 출력한다.

교차 연산자로는 실수 문제에 대해 좋은 성능을 보이는 것으로 알려진 확장 상자 교차 연산자를 채택했고 변이 연산자로는 가우시안 변이 연산자를 사용했다. 확장 상자 연산자는 두 부모해의 각각의 좌표마다 두 값 사이의 상자를 일정 비율 늘려서 그 안에서 고른 확률로 자식 해를 선택하는 방법이고, 가우시안 변이 연산자는 변이량이 일정 표준편차를 가진 정규 분포를 따르도록 하는 방법이다. 각각의 연산자의 의사 코드(pseudo-code)가 그림 4와 그림 5에 나온다. 본 논문의 모든 실험에서 확장 상자 연산자의 확장 비율  $\alpha$ 를 0.5, 가우시안 변이 연산자의 표준 편차  $\sigma$ 를 전체 구간 길이의 반인 50, 변이 확률  $p_m$ 을 0.1로 정했다.

평가 함수에서의 난수 점 발생 횟수는 100,000 회로 고정했다. 100,000 회로는 아직 시뮬레이션 값이 안정되지 않아 실제 값과는 약간의 차이가 날 수도 있지만 유전 알고리즘에서는 약간의 변이 효과도 기대하면서 실제 값과 너무 동떨어진 엉뚱한 값을 내지는 않는 적당한 횟수다. 이보다 더 많은 횟수의 난수를 발생하게 되면 해를 평가하는데 너무 오랜 시간이 소모되어 좋은 해를 찾는다 해도 비용이 너무 많이 들게 된다. 그러나 최종적으로는 유전 알고리즘의 최종 해를 1,000,000 회의 난수 점을 발생해서 재평가하여

실제에 아주 근접한 값을 구하도록 하였다.

```

Gaussian_mutation(( $x_1, y_1$ ), ( $x_2, y_2$ ), ..., ( $x_n, y_n$ ))
  for  $i \leftarrow 0$  to  $n$ 
    if (0과 1 사이의 실수 난수 <  $p_m / n$ )
       $x_i \leftarrow x_i + N(0,1) \times \sigma$ ;
       $y_i \leftarrow y_i + N(0,1) \times \sigma$ ;
      ( $x_i, y_i$ )가 범위를 벗어난 경우 조정;
  
```

그림 5. 가우시안 변이 연산자의 의사 코드  
Fig. 5. Pseudo-code of Gaussian mutation operator

문제에서 마지막에는 실제와 유사한 정확한 평가 값이 필요하지만 사실 유전 알고리즘이 수행되는 중간에는 각 해들의 값이 정확한 것보다는 평가에 들어가는 비용을 줄이는 것이 더 효율적일 수 있다. 이 점에 착안하여 난수 발생 횟수를 일정하게 고정하지 않고 처음에는 적은 수의 난수만 발생하다가 세대가 지남에 따라 점점 더 많은 수의 난수를 발생해서 평가 함수에 적용하여 후반으로 갈수록 정확도를 높이도록 하는 방법도 함께 구현했다. 다음 절의 실험 결과에 난수 발생 횟수를 일정하게 고정하는 방식과 세대가 지남에 따라 점점 횟수를 늘리는 방식의 성능을 비교했다.

### 4. 실험 결과

#### 4.1 실험 데이터

실험을 위한 최대 커버리지 센서 배치 문제의 인스턴스를 다음과 같이 생성하였다. 일단 센서는 세 종류가 있는 경우를 다루었고 커버해야 하는 전체 영역을  $[0,100] \times [0,100]$ 의 정사각형으로 가정하였다. 센서의 탐지 거리를 0.8 비율만큼 차이가 나게 하고 센서들의 개수는 작을수록 1.5 배가 많도록 하였다. 이제 각 인스턴스마다 가장 큰 센서의 크기를 정하고 센서들의 면적 합이 전체 정사각형 면적의 0.7 배가 되게 정하면 각 센서들의 개수가 결정된다.

표 1은 가장 큰 센서의 크기를 10과 8로 정했을 때의 인스턴스를 보여준다. 여기서  $r_1, r_2, r_3$ 는 각 센서의 탐지거리이고,  $n_1, n_2, n_3$ 는 각 센서의 개수를 나타낸다.

표 1. 테스트 인스턴스  
Table 1. Test instances

	$r_1$	$n_1$	$r_2$	$n_2$	$r_3$	$n_3$
S01	10.0	7	8.00	10	6.40	15
S02	8.0	12	6.40	18	5.12	27

#### 4.2 결과 분석

그림 6과 7은 인스턴스 S01과 S02에 대한 유전 알고리즘에서 세대에 따른 전체 인구 중의 최대값과 평균값의 변화를 나타낸 것으로 각각의 인스턴스에 대해서 유전 알고리즘이 잘 수렴했음을 볼 수 있다.

유전 알고리즘의 성능을 비교해보기 위해 임의로 생성한 해들 중 가장 좋은 품질(RANDOM)과 유전 알고리즘으로 구한 해의 품질(GA)을 비교했다. RANDOM은 GA와 수행

시간이 비슷하게 걸리도록 하기 위해 25,000 개(GA를 한번 수행할 때 생성하는 자식 해의 총 개수)의 랜덤 배치들 중 가장 좋은 품질의 배치를 선택하는 것을 한번 수행하는 것으로 정했다. 또, 몬테카를로법을 통한 해의 평가에서 발생한 난수 점의 개수를 일정하게 고정하지 않고 세대수가 증가할수록 개수를 점점 늘려나가는 방식(MGA)의 성능도 같이 비교했다. 이 경우 초기 난수 발생 횟수를 10,000으로 하고 100 세대가 지날 때마다 10,000 회씩 늘려 나갔다. 이렇게 하면 처음부터 계속 100,000 개의 난수를 발생하는 것보다 시간을 훨씬 단축시킬 수 있다.

표 2와 표 3은 인스턴스 S01과 S02에 대해 각 방법을 30번 수행한 실험 결과의 통계수치를 보여준다. 두 인스턴스에 대해 모두 유전 알고리즘(GA, MGA)이 임의로 해들을 생성한 경우(RANDOM)보다 성능이 좋았음을 볼 수 있다. 몬테카를로법을 통한 해의 평가가 시간의 대부분을 차지하므로 유전 알고리즘의 수행 시간도 임의로 해들을 생성한 경우에 비교해 오래 걸리지 않았다.

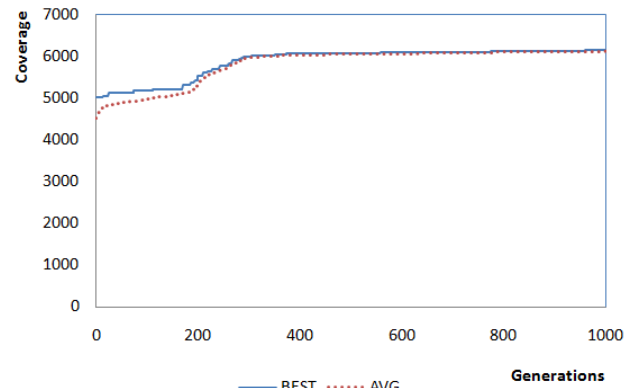


그림 6. 인스턴스 S01에 대한 유전 알고리즘의 세대에 따른 해들의 품질 변화 (한번 수행한 결과)  
Fig. 6. Convergence of a genetic algorithm for S01

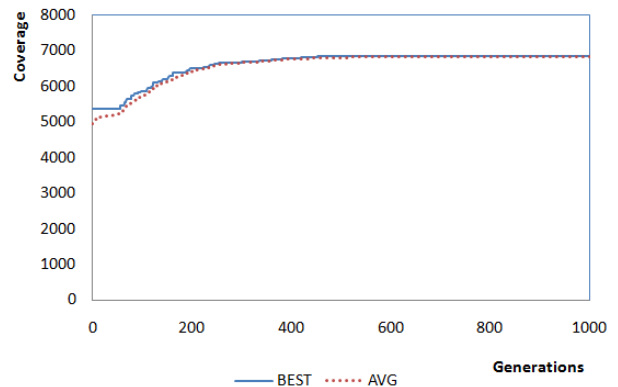


그림 7. 인스턴스 S02에 대한 유전 알고리즘의 세대에 따른 해들의 품질 변화 (한번 수행한 결과)  
Fig. 7. Convergence of a genetic algorithm for S02

MGA의 결과를 보면 GA에 비해 시간이 거의 반으로 줄어들었지만 해의 품질은 크게 차이가 나지 않았다는 것을 확인할 수 있다. 따라서 처음에는 약간 부정확하게 해를 평가하다가 세대가 지남에 따라 점점 정확도를 높여가는 방법이 더 효율적이라고 할 수 있다.

표 2. 인스턴스 S01에 대한 결과 비교

Table 2. Comparison of results for S01

	Best	Avg	Std	Time*
RANDOM	5376.01	5233.75	49.17	1932
GA	6118.62	6069.57	46.55	1580
MGA	<b>6138.80</b>	<b>6082.14</b>	41.60	<b>1074</b>

\* Avg CPU seconds on Intel(R) Core2 Duo CPU 2.66GHz

표 3. 인스턴스 S02에 대한 결과 비교

Table 3. Comparison of results for S02

	Best	Avg	Std	Time*
RANDOM	5648.50	5562.04	38.32	3074
GA	<b>6859.72</b>	<b>6780.21</b>	48.83	3196
MGA	6855.78	6774.84	58.45	<b>1430</b>

\* Avg CPU seconds on Intel(R) Core2 Duo CPU 2.66GHz

## 5. 결론

본 논문에서는 실제 센서 배치를 할 때 많이 발생하는 문제인 최대 커버리지 센서 배치 문제를 정의하고 그 문제 공간의 성질을 분석했다. 또한 유전 알고리즘을 통해 좋은 해를 구하는 방법을 모색해 보았다.

최대 커버리지 센서 배치 문제의 표현 공간과 해 공간의 관계는 몫(quotient) 공간의 개념으로 이해할 수 있으며 이러한 문제의 성질을 응용하면 좀 더 좋은 해를 구할 수 있을 것이다. 본 연구에서 고안한 유전 알고리즘의 성능이 임의로 발생한 해들보다 품질이 좋다는 것을 실험을 통해 확인함으로써 최대 커버리지 센서 배치 문제에 유전 알고리즘이 문제의 해 공간을 조금 더 효율적으로 탐색할 수 있음을 보였다. 유전 알고리즘에서 해의 평가를 몬테카를로법을 통해 구현하였고 몬테카를로법에서 사용되는 난수 집의 발생 횟수를 고정하지 않고 처음에는 적은 수로 시작해서 유전 알고리즘의 세대가 증가함에 따라 점점 늘어나가는 방식을 통해 수행 시간을 단축하는 효과를 얻을 수 있었다.

## 참고 문헌

[1] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," *Proc. IEEE INFOCOM*, pp. 1293-1303, 2003.

[2] A. Howard, M. J. Matarić and G. S. Sukhatme, "Mobile sensor network deployment using potential field: a distributed scalable solution to the area coverage problem", *Proc. International Conference on Distributed Autonomous Robotic Systems*, pp. 299-308, 2002.

[3] S. S. Dhillon, K. Chakrabarty and S. S. Iyengar, "Sensor placement algorithms for grid coverage", *Proc. International Conference on Information Fusion*, pp. 1581-1587, 2002.

[4] X. Xu and S. Sahni, "Approximation algorithms for wireless sensor deployment," *IEEE Transactions on Computers*, Vol. 56, No. 12, pp.1681-1695, 2007.

[5] Y. Xu and X. Yao, "A GA approach to the opti-

mal placement of sensors in wireless sensor networks with obstacles and preferences," *Proc. 3rd IEEE Consumer Communications and Networking Conference*, pp. 127-131, 2006.

[6] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Ad Hoc and Sensor Wireless Networks*. Vol. 1, No. 1, pp. 89-124, 2005.

[7] K. Chakrabarty, S. S. Iyengar, H. Qi and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks", *IEEE Transactions on Computers*, Vol. 51, pp. 1448-1453, 2002.

[8] 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.

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

[10] 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," *IEICE Transactions on Communications*, Vol. E91-B, No. 11, pp. 3534-3543, 2008.

[11] M. Cardei, M. Thai, Y. Li, and W. Wu, "Energy efficient target coverage in wireless sensor networks," *Proc. IEEE INFOCOM*, pp. 1976-1984, 2005.

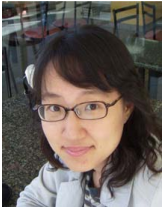
[12] G. Wang, G. Cao, T. L. Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," *Proc. IEEE INFOCOM*, pp. 2302-2312, 2005.

[13] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," *Proc. First International Conference on Embedded Networked Sensor Systems*, pp. 28-39, 2003

[14] M. Locateli and U. Raber, "Packing equal circles in a square: a deterministic global optimization approach", *Discrete Applied Mathematics*, Vol. 122, pp. 139-166, 2002.

[15] S. Meguerdichian, S. Slijepcevic, V. Karayan and M. Potkonjak, "Coverage problems in wireless ad-hoc sensor networks", *Proc. IEEE INFOCOM*, Vol. 3, pp. 1380-1387, 2001.

저 자 소 개



**윤유림 (Yourim Yoon)**

2003년 : 서울대학교 컴퓨터공학부(학사)  
2006년 : 서울대학교 컴퓨터공학부  
(박사과정수료)

관심분야 : 최적화 이론, 진화연산, 이산수학, 조합최적화  
Phone : 02) 880-1851  
E-mail : yryoon@soar.snu.ac.kr



**김용혁 (Yong-Hyuk Kim)**

1999년 : 서울대학교 전산과학 전공(학사)  
2001년 : 서울대학교 컴퓨터공학부(석사)  
2005년 : 서울대학교 컴퓨터공학부(박사)  
2005년~2007년 : 서울대학교 반도체 공동  
연구소 연구원  
2007년~현재 : 광운대학교 컴퓨터소프트  
웨어학과 조교수

관심분야 : 최적화, 진화연산, 지식공학  
Phone : 02) 940-5212  
E-mail : yhdfly@kw.ac.kr