

Set Covering 문제의 해법을 위한 개선된 Simulated Annealing 알고리즘

이현남 · 한치근

경희대학교 전자계산공학과

An Enhanced Simulated Annealing Algorithm for the Set Covering Problem

Hyun-Nam Lee · Chi-Geun Han

The set covering(SC) problem is the problem of covering all the rows of an $m \times n$ matrix of ones and zeros by a subset of columns with a minimal cost. It has many practical applications of modeling of real world problems. The SC problem has been proven to be NP-Complete and many algorithms have been presented to solve the SC problem. In this paper we present hybrid simulated annealing(HSA) algorithm based on the Simulated Annealing(SA) for the SC problem. The HSA is an algorithm which combines SA with a crossover operation in a genetic algorithm and a local search method. Our experimental results show that the HSA obtains better results than SA does.

1. 서론

Set Covering(SC) 문제는 전체집합을 커버하는 최소한의 부분 집합을 구하는 문제로 소방서, 응급실, 이동통신 기지국, 회사 지점 등의 위치를 설정하는 문제, 항공기 승무원, 버스 운전기사의 스케줄링 문제, 자원을 할당하는 문제 등 실제세계에서의 많은 문제에 적용되고 있다[1, 2, 7, 8].

그 중 실제상황에서 가장 많이 고려되는 위치설정문제의 해는 공간확보와 설치비용에 관련되어 설비구축에 매우 민감한 결과를 산출한다. 불필요한 위치에까지 설비를 설치하게 되면 설치 및 공간확보에 드는 비용을 낭비하여 손해를 일으키게 되는 것이다. 따라서, 고객의 편의를 고려하면서도 설비의 설치비용을 최소화할 수 있도록 최적의 해를 구하는 것이 매우 중요하다. SC 문제는 NP-Complete 문제[6]로, 이를 해결하기 위하여 다양한 휴리스틱(heuristic) 접근법이 이용되고 있다. 그 중 그리디(greedy) 휴리스틱 방법은 빠른 시간내에 해를 구하지만, 국부탐색을 하기 때문에 지역 최적해에 빠지기 쉬운

단점이 있다. 그리고 전역탐색을 하는 Simulated Annealing (SA)은 해공간에서의 작은 이동으로 인해 해공간을 탐색하는 데 시간이 오래 걸리는 단점이 있다.

본 논문에서는 SA의 단점을 보완하고 보다 좋은 해를 얻기 위하여 Hybrid Simulated Annealing(HSA) 알고리즘을 제안하였다. HSA는 SA를 기반으로 하여 유전자 알고리즘(Genetic Algorithm: GA)의 교차연산(crossover)과 지역탐색(local search)을 결합한 방법이다. HSA의 성능을 평가하기 위하여 이를 다양한 SC 문제에 적용하고, 기존에 해결방안으로 제시되었던 그리디 휴리스틱 방법, SA 등과 결과를 비교함으로써 그 효율성을 증명하고자 한다.

2. Set Covering 문제와 기존해법

2.1 Set Covering 문제의 정의

SC 문제는 1과 0의 값으로 구성된 $m \times n$ 행렬에서, 최소비용

으로 1~ m 의 모든 행을 커버하는 열들의 부분집합을 구하는 문제이다[2]. <그림 1>에 $m \times n$ 행렬의 예를 나타내었다.

행렬내에서 하나의 열을 구성하는 m 개의 요소는 m 개의 각 행과 매치되고, 그 열이 커버할 수 있는 행과 매치되는 요소에는 1의 값이, 커버할 수 없는 행과 매치되는 요소에는 0의 값이 설정되게 된다. 또한, 각 열에는 각기 다른 비용이 주어져 있어서, 열들을 조합하여 모든 행을 커버하면서도 열들의 비용의 합은 최소가 되도록 해를 구성하는 것이 SC 문제의 목적인 것이다.

	1	2	3	4	...	n (열)
1	1	0	0	1		0
2	0	1	0	1		1
3	1	0	0	0		0
⋮						
m (행)	0	0	1	0		1

그림 1. SC 문제에서의 $m \times n$ 행렬.

SC 문제는 지금까지 자원할당문제, 위치설정문제, 스케줄링 문제 등의 다양한 실제계 문제의 모델링에 적용되어왔다. 이 중, 위치설정문제는 많은 사람들에게 서비스를 제공하면서도 설치비용을 최소화할 수 있도록 적절한 위치에 설비를 설치하는 문제이다. 일반적으로 위치설정문제는 그래프로 변형하여 푸는데, 이 때 노드는 설치대상인 위치를 나타내고, 각 노드를 연결하는 에지의 가중값은 노드간의 거리를 나타내어 SC 문제에 적용한다.

2.2 수식 정의

앞의 2.1 절에서 설명한 것과 같이, SC 문제는 주어진 행렬의 모든 행을 커버하면서도 비용은 최소가 되도록 열들의 부분집합으로 해를 구성하는 문제이다.

SC 문제를 수식으로 정의하면 다음과 같다.

$$A = \{a_{ij}\} \in R^{m \times n},$$

$$a_{ij} = \begin{cases} 1, & \text{만약 열 } j \text{ 가 해에 포함되면} \\ 0, & \text{그렇지 않으면} \end{cases}$$

$$x = \{x_j\} \in R^n,$$

$$x_j = \begin{cases} 1, & \text{만약 열 } j \text{ 가 해에 포함되면} \\ 0, & \text{그렇지 않으면} \end{cases}$$

$$c = \{c_j\} \in R^n,$$

c_j 는 열 j 가 해에 포함될 때의 비용 0

SC 문제는

$$\text{Minimize } \sum_{j=1}^n c_j x_j \tag{1}$$

$$\text{subject to } \sum_{j=1}^n a_{ij} x_j \geq 1, \quad i = 1, \dots, m$$

위의 (식 1)에서, 모든 열 j 에 대해 비용 c_j 가 같은 경우를 단위비용(unicost) SC 문제라 한다. 단위비용 SC 문제는 위치설정 문제에서 설치위치나 설치지점 간의 거리에 관계없이 설치비용이 같은 경우에 적용된다. 즉, 설치비용을 고려하지 않고 해에 포함되는 열의 수를 최소화하는 것이 목적인 것이다. 모든 열에 대해 비용이 같으므로 목적식 또한

$$\text{Minimize } \sum_{j=1}^n x_j \tag{2}$$

$$\text{subject to } \sum_{j=1}^n a_{ij} x_j \geq 1, \quad i = 1, \dots, m$$

가 된다.

본 논문에서는 일반적인 위치설정문제를 단위비용 SC 문제에 적용하여 해결하였다.

2.3 기존해법

SC 문제를 해결하기 위해 지금까지 많은 휴리스틱 알고리즘들이 제안되어 왔다.

그 중, 그리디 휴리스틱 방법[1]은 현재상태에서 최선의 선택을 한 후 다음상태로 이동하는 과정을 반복함으로써 근사 최적해를 얻는 방법이다. 여기에서 최선의 선택이란 커버되지 않은 모든 열들 중에서, 가장 많은 행을 커버하는 열을 우선적으로 선택하는 것을 말한다. 결국, 모든 행이 커버되는 순간까지 이 과정을 반복함으로써 해를 얻는 것이다. 그러나 한 상태에서 가장 많은 행을 커버하는 열을 선택하는 것은 그 상태에서는 최선의 선택이지만 전체적으로 보면 최선이 아닌 결과를 가져올 수도 있다. 이런 점 때문에 그리디 휴리스틱 방법은 빠른 시간내에 해를 얻을 수는 있지만 지역 최적해에 빠지기 쉬운 단점이 있다.

최근에는 SC 문제의 최적해를 구하기 위해 Jacobs와 Brusco[8]가 SA에 기반을 둔 휴리스틱 방법을 제안하였고, Beasley와 Chu[2]는 GA에 기반을 둔 휴리스틱 방법을 제안하였다. SA와 GA는 해공간을 전역탐색함으로써 전역 최적해에 접근하는 방법이다[11]. Gonzalez는 SC 문제를 이용하여 버스 운전기사의 스케줄링을 해결하였다. 운전기사가 제공해야 하는

서비스는 행으로 표현하고 운전기사들의 교대조는 열로 표현하여, 모든 서비스를 제공하면서도 운전기사의 작업량은 최소화하는 해를 구한 것이다[7]. Caprara et. al.은 이탈리아 철도회사의 스케줄을 5,000개의 행과 1,000,000개의 열을 갖고 있는 SC 문제로 바꾸어 해결하였다[3].

본 논문에서는 그리디 휴리스틱 방법의 지역 최적해에 빠지기 쉬운 단점을 보완하고, SA의 해공간 탐색시간이 오래 걸리는 단점을 보완하기 위하여 HSA를 제안하였다.

3. Hybrid Simulated Annealing 알고리즘의 정의 및 적용

3.1 벡터 구성

본 논문에서는 노드 전체의 집합인 V 와 0보다 큰 가중값이 있는 에지의 집합 E 로 구성되어 있는 무향 그래프 $G=(V, E)$ 가 주어졌을 때, 각 노드들의 커버관계를 나타내는 $|V| \times |V|$ 행렬 A 를 생성하여 해를 구하였다. 즉, 그래프내의 모든 노드들 중에서 전체노드를 커버할 수 있는 최소수의 노드를 선택하는 것이다.

우선, <그림 2>에 간단한 그래프를 나타내었다. 그래프 상에서의 각 노드는 설비설치의 대상인 위치를 나타내고, 각 에지는 설정대상인 위치들간의 거리를 나타낸다.

$|V| \times |V|$ 행렬의 행과 열은 각각 그래프내의 모든 노드들로 구성되고, 각 노드마다 다른 노드들이 포함가능한 거리내에 위치하고 있는지를 조사하여 그 여부를 나타내는 1 또는 0의 값으로 열을 나타낸다. <그림 2>에서 만약 위치설정된 노드가 커버할 수 있는 거리가 50이라고 하면, $|V| \times |V|$ 행렬은 다음과 같이 된다.

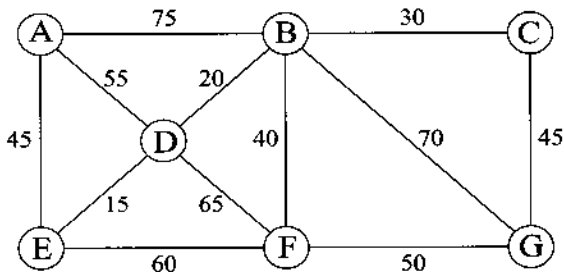


그림 2. 위치설정문제를 위한 그래프 예제.

해를 표현하는 벡터 x 는 전체노드의 개수 $|V|$ 의 크기를 갖는 비트 스트링으로 표현된다. 벡터의 각 비트는 노드를 표현

하는 행렬에서의 각 열과 매치되어 그 노드에 위치설정이 결정되면 1의 값으로, 아니면 0의 값으로 설정된다. <그림 3>의 행렬을 기반으로 하여 생성한 해의 예를 들면 다음 그림과 같다.

해벡터에 E열을 포함시킴으로써 A, B, D, E행을 커버하고, G열을 포함시킴으로써 C, F, G행을 커버한다. 따라서 모든 행을 커버할 수 있는 것이다.

	A	B	C	D	E	F	G
A	1	0	0	0	1	0	0
B	0	1	1	1	1	1	0
C	0	1	1	0	0	0	1
D	0	1	0	1	1	0	0
E	1	1	0	1	1	0	0
F	0	1	0	0	0	1	1
G	0	0	1	0	0	1	1

그림 3. 그림 2에 대한 7×7 행렬.

A	B	C	D	E	F	G
0	0	0	0	1	0	1

그림 4. SC 문제의 벡터 예제.

3.2 Simulated Annealing

SA는 연속적으로 해들을 생성하고 한정된 수의 계산을 하여 최적의 해를 찾는 방법이다. 해공간에서 현재해 x 를 선택하고, x 의 이웃해 x' 를 얻어 이웃해로 이동하는 과정을 반복함으로써 전체해공간을 탐색하며, 이 탐색결과 근사 최적해를 얻는 것이다.

이웃해로의 이동은 이웃해의 비용이 현재해의 비용보다 적은 경우에 허용되는데, 이 과정을 반복하다보면 지역 최적해에 빠지는 결과를 가져올 수 있다. 따라서, SA는 온도 T 에 대한 에너지변화에 따른 확률 P 에 의해 비용이 큰 이웃해로의 이동도 허용한다. 이것은 SA가 지역 최적해에 빠졌을 경우, hill climbing을 통해 이를 탈피할 수 있다는 것을 의미한다[5, 8, 9].

SA를 기술하면 <그림 5>와 같다. 12번 줄이 지역 최적해에서 빠져 나올 수 있는 기회를 부여하는 부분으로, 확률 P 는 $\exp(-\frac{\Delta E}{T})$ 이다. 즉, 이웃해의 비용이 현재해의 비용보다 크더라도 임의로 발생시킨 수가 확률 P 보다 작으면 이웃해로 이동하는 것이다. SA는 15번 줄에서처럼 온도변화를 이용하여 현재해의 비용이 최종적으로 전역 최적해에 도달하도록 하는

방법이다. SA에서 초기 온도값은 전역 최적해의 수렴에 큰 영향을 준다. 초기 온도값이 너무 크면 초기에 향상이 일어나지 않아 안정된 해를 얻지 못하거나 전체 수행시간이 너무 많이 들게 된다. 또한, 초기 온도값이 너무 작으면 해가 깊이있게 탐색되지 않아 전역 최적해로의 수렴이 이루어지기 힘들다. 따라서, 적당한 값으로 설정하는 것이 중요하다. 본 논문에서는 각 상태에서의 온도를 다음 냉각 스케줄에 의하여 구한다.

$$T_k = \alpha T_{k-1}, \quad 0 < \alpha < 1 \quad (3)$$

SA는 현재해 x 에 작은 변화를 주어 이웃해 x' 을 생성하여 해공간을 탐색하면서 전역 최적해에 수렴하는 방법이다. 그러나 이웃해로의 이동은 해공간내에서의 작은 이동의 효과를 내므로, 해공간이 큰 문제에서는 전체해공간을 탐색하는데 너무 많은 시간이 걸린다는 단점이 있다[10].

```

1: procedure SA
2: begin
3:   Get an initial solution  $x$  and
   an initial temperature  $T$ 
4:   While not stopping-criterion do
5:     begin
6:       While not inner-loop-criterion do
7:         begin
8:           Select a neighbor  $x'$  of  $x$ 
9:            $\Delta E = \text{cost}(x') - \text{cost}(x)$ 
10:          if ( $\Delta E \leq 0$ ) then
11:             $x \leftarrow x'$ 
12:          else if (random[0,1] <  $P$ ) then
13:             $x \leftarrow x'$ 
14:          end
15:          Reduce temperature  $T$ 
16:        end
17:      Return  $x$ 
18:    end

```

그림 5. SA 알고리즘.

3.3 균일(uniform) 교차연산

본 논문에서는 해공간내에서의 큰 이동의 효과를 내어 해공간의 탐색시간을 단축시키고자 GA의 균일 교차연산을 이용하였다. 균일 교차연산은 GA에서 흔히 사용되는 일점(one-point) 또

는 이점(two-point) 교차연산과는 달리 하나의 자식해를 생성한다. 균일 교차연산의 과정은 다음과 같다. 우선, 해와 같은 길이인 비트 스트링 B 를 1과 0의 값으로 랜덤하게 생성한다.

$$B : b_1 b_2 \dots b_{n-1} b_n, \quad b_i \in \{0,1\}, \quad i=1, \dots, n$$

스트링 B 는 교차연산 마스크(mask)의 역할을 한다. 두 해 x_1 과 x_2 를 연산하여 자식해 x' 를 생성하고자 할 때, B 의 각 비트 값을 조사하여 그 값이 0이면 x' 는 x_1 의 값을, 1이면 x_2 의 값을 갖게 되는 것이다.

즉, 균일 교차연산은 모든 $i = 1, \dots, n$ 에 대해

$$\begin{aligned} \text{if } b_i = 0 \text{ then } x'[i] &= x_1[i] \\ \text{if } b_i = 1 \text{ then } x'[i] &= x_2[i] \end{aligned}$$

와 같이 동작하는 것이다. 균일 교차연산의 예를 들면 <그림 6>과 같다.

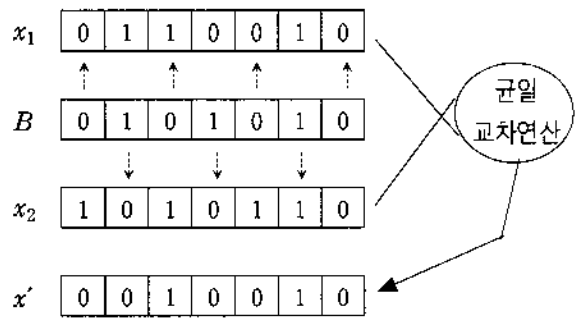


그림 6. 균일 교차연산 예.

일점 교차연산은 두 해 x_1 과 x_2 의 일부분을 교환함으로써 자식해 x' 를 생성하는 방법이기 때문에 x' 는 x_1 이나 x_2 중 하나와 유사한 해로 생성되기 쉽다. 반면, 균일 교차연산은 마스크 B 의 각 비트값을 조사하여 두 해 x_1 과 x_2 의 값을 결합하는 방법이므로, 해 벡터가 비트 스트링이고 교차연산할 두 해의 차이가 크지 않은 경우에 효율적이라고 할 수 있다.

3.4 지역탐색

HSA는 전역 최적해에 보다 근사하게 수렴하기 위하여 지역탐색을 이용한다. 현재해 x_1 으로부터 지역탐색을 수행하여 지역 최적해 x_1' 을 얻고, x_1' 의 이웃해 x_2 로부터 지역탐색을 수행하여 지역 최적해 x_2' 을 얻는다. 이들 두 해 x_1' 과 x_2' 가 비용비교에 이용되므로, 매 비교시마다 각 지역 최적해가 고려되게 된다. 즉, <그림 7>에서처럼 해공간을 탐색하면서 비용을 비교할 때마다 각 지역 최적해에 접근하므로, 전역 최적해로의 수렴도 더 쉬워지게 되는 것이다.

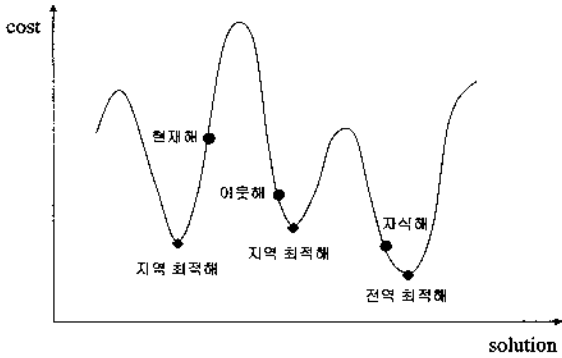


그림 7. 해공간 표현

또한, HSA에서의 지역탐색은 GA에서 적은 비용 함수식을 가진 부모해를 선택하는 것과 유사한 역할을 한다. 현재해와 이웃해를 지역탐색함으로써 두 지역 최적해를 얻고 좋은 두 해를 결합하여 자식해를 생성하게 되는 것이다.

본 논문의 지역탐색은 주어진 해에서 1로 설정된 열들만을 대상으로, 현재상태에서 최선의 선택을 하고 다음상태로 전이하는 방식으로 수행된다. 이 때 최선의 선택이란 가장 많은 행을 커버하는 열을 선택하는 것을 말한다. 이러한 과정을 모든 행이 커버될 때까지 반복함으로써 1로 설정된 열의 수를 줄이는 것이다.

지역탐색의 예를 들면 다음과 같다. <그림 8>은 3.1절에 있는 <그림 3>의 행렬을 바탕으로 생성한 임의의 해이다.

A	B	C	D	E	F	G
1	0	0	0	1	0	1

그림 8. HSA에서 지역탐색 전의 벡터.

주어진 벡터에서 1로 설정되어 있는 열들은 A, E, G이고, 이들은 각각 2개, 4개, 3개의 행을 커버할 수 있다. 이 중 가장 많은 행을 커버하는 열 E를 선택하면 A, B, D, E 행이 커버되고 다음으로 많은 행을 커버하는 열 G를 선택하면 C, E, G 행이 커버됨으로써, 다음 그림처럼 열 A를 해에 포함시키지 않아도 모든 행들이 커버되게 되는 것이다.

이런 과정을 통해 해에 포함되는 열의 수를 최소화하고, 그 결과 얻은 해의 비용이 지역 최적해가 되는 것이다.

A	B	C	D	E	F	G
0	0	0	0	1	0	1

그림 9. HSA에서 지역탐색 후의 벡터

3.5 Hybrid Simulated Annealing 알고리즘의 정의

SA는 해공간내에서의 작은 이동을 통해 전체해공간을 탐색하기 때문에 탐색시간이 오래 걸린다는 단점이 있다. 따라서, 보다 큰 이동의 효과를 냄으로써 SA의 단점을 보완하고자 GA의 교차연산을 결합하고, 또한 더욱 전역 최적해에 근접수렴하기 위하여 지역탐색방법을 결합하여 HSA를 정의하였다. <그림 10>에 HSA의 구조를 보기 쉽게 나타내었다.

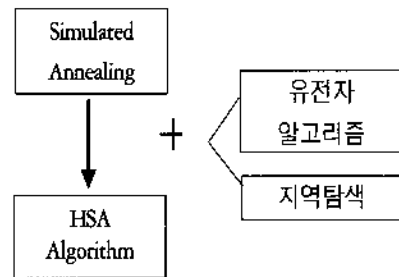


그림 10. HSA 알고리즘의 구조

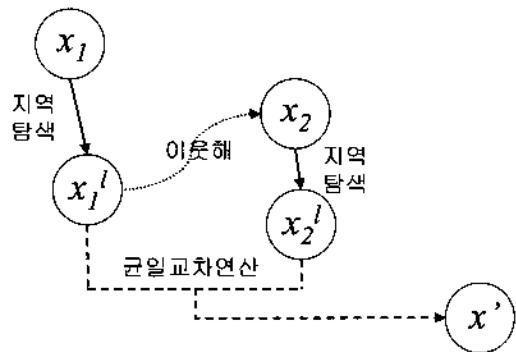


그림 11. HSA의 주요연산과정.

HSA에서는 현재해와 이웃해에 지역탐색을 적용하여 구한 서로 다른 두 지역 최적해를 교차연산함으로써 둘과 다른 새로운 자식해를 생성하는데, 이 과정이 해공간에서의 큰 이동의 효과를 가져오는 것이다. 그리고 지역 최적해들을 대상으로 연산을 하므로 그 자식해도 적은 비용을 가진 해가 생성되는 것이다. HSA는 다음과 같은 원리를 이용한다. 우선, 연속적인 해들을 생성하여 현재해 x_1 을 선택하고 지역탐색하여 지역 최적해 x_1' 을 구하고, x_1' 으로부터 일정비율의 변화에 의해 이웃해 x_2 를 생성하여 지역탐색한 결과값 x_2' 을 구한다. 그리고 두 지역 최적해 x_1' , x_2' 에 교차연산을 하여 자식해 x' 을 얻은 후, 현재해 x_1' 과의 비용 비교를 통하여 비용이 적은 해로 이동을 한다. 이러한 과정을 반복함으로써 전역 최적해에 접근하게 되는 것이다. 물론, SA와 마찬가지로 지역 최적해에

머무르지 않도록 하기 위해서 온도 T 에 대한 에너지변화에 따른 확률을 이용하여 hill climbing을 허용한다.

```

1: procedure HSA
2: begin
3:   Get an initial solution  $x_1$  and
   an initial temperature  $T$ 
4:   While not stopping-criterion do
5:     begin
6:        $x_1' \leftarrow \text{local\_search}(x_1)$ 
7:       Select a neighbor  $x_2$  of  $x_1'$ 
8:        $x_2' \leftarrow \text{local\_search}(x_2)$ 
9:        $x' \leftarrow \text{crossover}(x_1', x_2')$ 
10:       $\Delta E = \text{cost}(x') - \text{cost}(x_1')$ 
11:      if ( $\Delta E \leq 0$  or
          random[0,1]  $< \exp(-\frac{\Delta E}{T})$ ) then
12:         $x_1 \leftarrow x'$ 
13:      Reduce temperature  $T$ 
14:    end
15:  Return  $x_1'$ 
16: end

```

그림 12. HSA 알고리즘.

이 때의 지역탐색은 앞의 3.4절에서 다른 방법을 사용하고, 교차연산은 3.3절에서 기술하였던 균일 교차연산을 사용한다. HSA에서의 주요연산인 교차연산과 지역탐색을 <그림 11>에 간단하게 표현하였다.

<그림 12>는 HSA를 기술한 것이다. 전체적인 구조는 3.2절에서 다른 SA를 기반으로 하고, 6번 줄과 8번 줄처럼 현재해와 이웃해에 각각 지역탐색을 적용하고, 9번 줄에서 그 두 지역 최적해에 균일 교차연산을 적용함으로써 좋은 자식해를 생성하여 현재해와 비용비교를 하는 것이다. 그리고 온도변화식은 3.2절의 (식 3)과 같게 하였다.

4. 실험 및 성능평가

4.1 문제정의

본 논문에서는 위치설정문제를 단위비용 SC 문제에 적용하고, 그 최적해를 구하기 위하여 HSA를 제안하였다. 그리고 HSA의 성능평가를 위하여 여러 방법들과도 그 결과를 비교하

였다.

방법들을 적용할 다양한 문제들은 그래프의 노드 개수를 각각 30개, 50개, 80개, 110개, 140개로 하고, 위치설정된 노드가 커버할 수 있는 에지 가중치를 40, 45, 50, 55, 60으로 달리 구성하여 25개의 문제로 정의하였다. 각 그래프에서의 에지 가중치는 균등(uniform) 분포로 30~70 사이의 임의의 정수를 발생시켜 얻었다.

4.2 기존방법과의 성능비교

HSA는 SA의 시간적 단점을 보완하여 보다 짧은 시간안에 보다 좋은 해에 수렴하기 위한 방법이다. 이 HSA의 성능을 평가하기 위하여, 4.1절에서 정의한 여러 문제에 그리디 휴리스틱 방법, SA, 그리고 HSA를 적용하여 실험결과를 비교하였다.

그리디 휴리스틱 방법은 단 한번의 루프를 통하여 가장 많은 노드를 커버하는 노드부터 선택하면서 근사 최적해를 구하도록 하였다.

SA와 HSA는 정확한 비교를 위하여 초기온도를 비롯한 모든 조건을 같게 하고, 3.2절에서 기술했던 냉각 스케줄인 (식 3)에 의해 α 값은 0.95로 하여 각 상태에서의 온도를 구하였다.

SA를 기반으로 하는 알고리즘에서의 초기온도는 전역 최적해의 수렴에 큰 영향을 준다. 적절한 초기온도를 구하기 위하여 50°, 70°, 100°, 130°, 150°로 다르게 SA에 적용하여 실험한 결과, 100°일 때 가장 좋은 근사 최적해를 얻을 수 있었다. 따라서 SA와 HSA의 초기온도를 100°로 설정하였다.

HSA에서는 SA와 마찬가지로 현재해의 일부분을 변경하여 이웃해를 얻는다. 그러나 HSA는 현재해를 지역탐색하여 지역 최적해를 얻은 후 이웃해를 생성하기 때문에, SA에서의 작은 변화로는 현재해와 비교될 수 있는 정도의 적당한 이웃해를 얻을 수 없다. 따라서, 적당한 변화로 이웃해를 생성하기 위하여 본 논문에서는 HSA에서의 이웃해를 생성하기 위한 현재해의 변경비율을 최소한 한 비트의 변화에 추가로 2%, 5%, 8%, 10%로 주고 실험하였다. 그 결과, 변화비율이 근사 최적해의 수렴값에는 별로 영향을 끼치지 않았다. 따라서 해 수렴 시간면에서 다소 좋은 결과를 보인 5%를 현재해의 변경비율로 결정하였다.

각 방법의 평가기준으로는 근사 최적해값과 수렴시간을 이용하였는데, 우선 <표 1>에 그리디 휴리스틱 방법과 SA, HSA의 해 수렴결과를 나타내었다. SA와 HSA의 해를 그리디 휴리스틱 방법과 각각 비교하기 위해 다음의 식에 적용하여 구한 그리디 휴리스틱 방법에 대한 향상도를 이용하였다.

$$\frac{(\text{그리디의 해}) - (\text{SA 또는 HSA의 해})}{(\text{그리디의 해})} \times 100(\%)$$

표 1. 그리디, SA와 HSA의 해 비교

문제	방법	그리디의 근사최적해	SA의 향상도(%)	HSA의 향상도(%)
1		9	11.1	11.1
2		6	16.7	16.7
3		4	-25.0	0.0
4		4	0.0	25.0
5		3	0.0	0.0
6		12	8.3	25.0
7		9	22.2	22.2
8		6	16.7	16.7
9		4	0.0	0.0
10		4	0.0	0.0
11		12	16.7	16.7
12		9	22.2	11.0
13		7	0.0	0.0
14		7	28.6	28.6
15		5	20.0	20.0
16		14	8.3	8.3
17		12	0.0	11.1
18		9	0.0	0.0
19		8	12.5	12.5
20		5	0.0	20.0
21		17	5.9	5.9
22		12	8.3	8.3
23		7	-14.3	14.3
24		6	0.0	0.0
25		5	0.0	20.0

<표 1>에 나타나 있는 향상도를 보면 다수의 문제에서 HSA의 결과가 그리디 휴리스틱 방법 및 SA보다 좋게 나올 것을 알 수 있다. 이것은 지역탐색을 사용함으로써 각 지역 최적해로의 접근이 쉬워져, 전역 최적해로 수렴하는 비율도 커진 결과이다.

다음의 <표 2>에는 SA와 HSA의 해 수렴시간을 비교한 결과가 나타나 있다. 해 수렴시간은 CPU clock 수와 비용 비교횟수로 표현하였다. 표의 HSA에서 *로 표시된 부분은 clock 수가 50 이하의 적은 시간이 걸린 문제이다. 해 수렴시간 및 비용의 비교횟수를 보면, HSA가 SA보다 월등하게 좋음을 알 수 있다.

이는 현재해와 이웃해의 차이가 너무 작아 큰 해공간을 탐색하는 시간이 많아 걸리는 SA의 단점을 교차연산으로 보완한 결과이다.

표 2. SA와 HSA의 해 수렴시간 비교

문제	방법	SA		HSA	
		clock 수	비교횟수	clock 수	비교횟수
1		710	295	820	204
2		1260	648	*	1
3		60	42	220	39
4		50	35	60	17
5		110	42	*	1
6		220	72	110	23
7		1430	515	60	15
8		1050	370	*	4
9		760	297	*	2
10		770	250	60	14
11		6540	1150	170	34
12		29050	6150	170	39
13		3680	743	330	54
14		10980	2325	*	1
15		11100	2325	4780	413
16		17300	2120	*	3
17		11370	1401	280	30
18		9520	742	220	24
19		18620	3821	*	1
20		54270	6413	280	28
21		35590	3996	110	12
22		17580	1395	2630	389
23		11420	2325	*	1
24		11700	847	390	48
25		14120	1016	60	4

5. 결론 및 추후 연구방향

SC 문제는 다양한 실세계 문제의 모델링에 많이 적용되고 있다. 그 중, 위치설정문제는 적절한 곳에 설비를 설치함으로써 최대한의 서비스를 제공하면서도 설치비용을 최소화하는 문제이다.

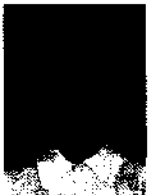
본 논문에서는 SC 문제를 해결하기 위해 SA에 GA의 교차연산과 지역탐색을 결합한 HSA를 제안하였다. HSA는 지역탐색을 통해 매 비용 비교시마다 각 지역 최적해들에 쉽게 접근함으로써 전역 최적해에 보다 근접한 해를 얻을 수 있고, 교차연산을 수행하여 해 공간에서의 큰 이동의 효과를 넘으로써 전

역 최적해에 수렴하는 시간이 너무 길어지는 SA의 단점을 보완하는 방법이다. 제안한 HSA의 효율성을 증명하기 위해 여러 문제를 정의하고 같은 문제에 HSA와 그리디 휴리스틱 방법, SA 등의 기존방법들에 각각 구현하여 실험결과를 비교하였다. 그 결과, HSA가 그리디 휴리스틱 방법보다 좋은 근사 최적해를 SA보다 훨씬 빠른 시간 내에 구할 수 있었다.

본 논문에서 다룬 SC 문제를 실생활에서의 설비를 설치하는데 적용하기 위해서는 보다 많은 제약조건들이 필요하다. 그러므로 다양한 부가목적이 추가된 SC 문제에 HSA를 적용하는 연구가 필요하겠다.

참고문헌

1. Balas, E. and Ho, A., "Set covering algorithms using cutting planes, heuristics and subgradient optimization: a computational study," *Mathematical Programming Study*, Vol. 12, pp. 37-60, 1980.
2. Beasley, J. E. and Chu, P., "A genetic algorithm for the set covering problem," *European Journal of Operational Research*, Vol. 94, pp. 392-404, 1996.
3. Caprara, A., Fischetti, M. and P. Toth, "A heuristic method for the set covering problem," *Working Paper, DEIS*, University of Bologna, Italy, 1995.
4. Daskin, M., *Network and Discrete Location Models, Algorithms and Applications*, John Wiley & Sons, Inc., 1995.
5. Desai, R. and Patil, R., "SALO: Combining simulated annealing and local optimization for efficient global optimization," *Proceedings of the 9th FLAIRS-96*, Key West, FL, pp. 233-237, 1996.
6. Garey, M. R. and Johnson, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, W. H. and Co., New York, 1979.
7. Gonzalez, L. F., "Evolutionary divide and conquer for the set covering problem," *MSc. Information Technology: Knowledge Based System*, Department of Artificial Intelligence, Edinburgh Univ., September, 1995.
8. Jacobs, L. W. and Brusco, M. J., "A simulated annealing-based heuristic for the set covering problem," *Working Paper Operations Management and Information Systems Department*, Northern Illinois Univ., March, 1993.
9. Kirkpatrick, S., Gellat, C. and Vecchi, M., "Optimization by simulated annealing," *Science*, Vol. 220, pp. 671-680, 1983.
10. Koakutsu, S., Kang, M. and Dai, W. W., "Genetic simulated annealing and application to non-slicing floor-plan design," *Fifth ACM/SIGDA PDW '96*, Virginia, 1996.
11. Muhlenbein, H., Georges-Schlechter and Kramer, O., "Evolution algorithms in combinatorial optimization," *Parallel Computing*, Vol. 7, pp. 65-85, 1988.



이현남

1996년 경희대학교 전자계산 공학 학사
1998년 경희대학교 전자계산공학 석사
관심분야: 그래프이론, Hybrid 알고리즘 설계



한치근

1983년 서울대학교 산업공학과 학사
1985년 서울대학교 산업공학과 석사
1991년 미국 펜실베니아주립 대학교
전산학 Ph. D.

현재: 경희대학교 전자계산공학과 부교수
관심분야: Combinatorial Optimization, 통신망
설계, 문서관리 등