

p-중앙 시설 위치선정 구성 알고리즘

이 상 운*

A Constructive Algorithm for p-Median Facility Location

Sang-Un Lee *

요 약

본 논문은 n 개의 행정구역으로 구성된 도시에 p 개의 시설을 신규로 설치하는 경우, 비용이 최소가 되는 최적의 시설 위치를 선정하는 알고리즘을 제안하였다. 이 문제는 정확한 해를 찾는 다항시간 알고리즘이 제안되지 않아 NP-난제로 분류되어 있다. 제안된 방법은 $p = [1, n-1]$ 에 대해 먼저 노드들을 증가시키는 방법으로 p 개를 선택하고, p 번째 선택된 시설 위치를 교체하는 방법을 적용하여 기존의 Myopic 알고리즘의 단점을 개선하였다. 제안된 알고리즘은 $n = 5, 7, 10, 55$ 인 데이터에 적용한 결과 역-삭제 방법에 비해 최적 해에 가장 근사한 해를 구할 수 있었으며, 엑셀을 활용해 간단히 구현할 수 있는 장점도 있다.

▶ Keywords : 시설 위치, 휴리스틱, 최단거리, 구성 알고리즘

Abstract

This paper proposes a location algorithm that locates newly built p -facilities in the optimal area with minimum cost in a city of n districts. This problem has been classified as NP-hard, to which no polynomial time algorithm exists. The proposed algorithm improves the shortcomings of existing Myopic algorithm by constructing until p -facilities and exchanging locations of p -th facility for $p = [1, n-1]$. When applied to experimental data of $n = 5, 7, 10, 55$, the proposed algorithm has obtained an approximate value nearest possible to the optimal solution take precedence of reverse-delete method. This algorithm is also simply executable using Excel.

▶ Keywords : Facility location, Heuristics, Shortest distance, Constructive algorithm

• 제1저자 : 이상운

• 투고일 : 2015. 02. 11, 심사일 : 2015. 03. 13, 게재확정일 : 2015. 04. 19.

* 강릉원주대학교 멀티미디어공학과 (Dept. of Multimedia Eng., Gangneung-Wonju National University)

I. 서론

최적 시설 위치를 찾는 위치 이론은 수학, 컴퓨터과학, 운용과학 (operations research, OR), 경제, 지리학 등 많은 학문들에 근거를 둔 학제적 분야로 경제적 목적 또는 공공서비스 측면에서 많은 관심의 대상이 되고 있다.

n 개의 행정구역 (동, 구 등) N_j , ($j=1,2,\dots,n$)으로 구성된 도시에 시설 F_i , ($i=2,3,\dots,p$)을 신규로 p 개 설치하는 경우, 최적의 시설 위치를 찾는 문제를 p -중앙 문제 (p -median problem)라 한다[1,2]. 여기서 대상은 공공과 상업 등 어떠한 시설도 될 수 있다. 특히 공공 시설물의 최적 위치는 지역사회의 안전과 복지에 심각한 영향을 미치며[2-4], 상업 시설 위치는 상품 판매 수익을 결정하는데 결정적인 요인으로 작용한다.

고객이 위치한 n 개 구역과 p 개 시설간의 거리가 감소하면 시설 접근성은 증가하며, 반응시간은 감소한다. 이 개념이 p -중앙값 (p -median) 이다. 따라서 p -시설의 위치는 고객이 위치한 구역과 가장 인접한 시설간의 평균 거리를 최소화시키도록 결정한다[4].

이 문제에서, 후보시설 F_i 와 노드들 N_j 사이의 최단 거리 l_{ij} 와 각 노드의 요구량 (주민 수, 유동인구수) d_j 를 알고 있다고 가정하자. 이 경우 시설 위치의 효율성은 고객이 위치한 구역 (노드)과 시설 간의 최단거리 l_{ij} 와 해당지역의 고객 수 d_j 를 합한 최소비용 $\min \sum c_{ij}$, ($c_{ij}=l_{ij} \times d_j$)으로 측정한다 [2]. 여기서, 동일 구역 내의 비용은 "0"으로, 두 지점간의 거리는 해당 구역의 중심 위치 간 거리로 계산된다.

p -중심 (p -center)문제의 해는 반드시 노드들에 위치하지 않는 반면에, p -중앙값 문제의 해는 항상 노드들에 위치하는 장점이 있다. 따라서 시설 위치 결정 문제는 p -중앙값 문제로 발전하였다[3]. Serra와 Reville[5-7]은 선형계획법 (linear programming, LP)과 분기 한정법 (branch-and-bound, BB)을 결합한 p -중앙값 알고리즘을 제안하였다.

또한, Myopic 알고리즘 (MA), 이웃탐색 (neighbourhood search, NS), 교환 휴리스틱 (exchange heuristic, EH), Tabu 탐색, 유전자 알고리즘, 담금질 방법 (simulated annealing, SA) 등 다양한 메타휴리스틱 방법들도 적용되고 있다[1,4,8,9].

만약, $p=1$ 인 경우의 최적의 시설 위치는 $\min \sum l_{ij}$ 또는 $\min \sum c_{ij}$ 가 되는 F_i 로 쉽게 결정될 수 있다. 그러나 $p \geq 2$ 에 대해 1960이후 지금까지도 ${}_n C_p$ 의 경우 수를 확인하는 열거

법 이외에는 정확한 해를 구하는 다항시간 알고리즘이 알려져 있지 않아 Kariv와 Hakimi는 p -중심과 p -중앙값 방법은 NP-난제 (NP-hard) 문제임을 증명하였다[2-4].

이 문제에 대해 다항시간으로 해를 구하는 방법으로, Choi et al.[10]은 n 개 후보 위치들을 대상으로 한 번에 하나씩 삭제하는 방식의 역-삭제 (reverse-delete, RD) 알고리즘을 제안하였다. 이 방법은 하나의 지점이 일단 삭제되면 추후 해당 지점은 더 이상 고려 대상이 되지 않아 최적 해를 찾지 못하는 단점과 더불어 우리가 원하는 $p < n/2$ 로 소수의 지점들만을 대상으로 하기 때문에 수행 횟수가 많이 소요되는 단점을 갖고 있다.

본 논문은 p 개의 시설에 대한 최적 위치를 결정하는 p -중앙 문제에 대해 근사 해를 빠르게 구하는 알고리즘을 제안한다. 제안된 방법은 $p=1$ 부터 $p=n-1$ 까지 한 번에 하나씩 증가시키면서 찾는 구성 알고리즘 (constructive algorithm, CA)으로 Choi et al.[10]의 RD를 역으로 수행하는 방식으로 RD에 비해 보다 빠르게 p 에 도달할 수 있으며, 일단 이전에 후보로 선정된 지점이라도 다른 지점으로 대체시키는 과정을 통해 RD에 비해 보다 최적 해를 찾을 수 있는 장점을 갖고 있다. 2장에서는 p -중앙 문제와 Myopic[8,9], Caccetta와 Dzator[4]가 제안한 RH1+RH2+RRH 알고리즘을 고찰한다. 3장에서는 구성 알고리즘을 제안하고, 4장에서는 사례들을 중심으로 제안된 방법의 적용성을 RD의 비교하여 평가해 본다.

II. 관련연구와 문제점

시설에 대한 최적의 위치를 찾는 기준은 응답시간 또는 비용 최소화를 시키는 것이다. 응답시간 (또는 비용) 최소화는 시설과 주민이 거주하는 구역간의 거리에 의존한다. 따라서 이들 시설을 설치할 위치는 이 시설들을 방문하는 사람들의 이동거리를 최소화시켜야 한다.

p -중앙 문제는 해당 구역과 가장 근접한 응급시설 간의 거리를 최소화시키도록 p 개의 시설 위치를 결정하는 문제이다. 이 문제는 식 (1)로 표현된다.

$$\begin{aligned} F_i, i &= \{1,2,\dots,p\} : \text{시설 후보 위치} \\ L_j, j &= \{1,2,\dots,n\} : \text{주민이 거주하는 구역} \\ l_{ij} &: \text{해당 구역 } j \text{와 시설 } i \text{ 간 최단거리} \\ x_{ij} &= \begin{cases} 1, & F_i \text{에서 } L_j \text{에 위치한 고객과 최단거리인 경우} \\ 0, & \text{그렇지 않은 경우} \end{cases} \\ y_i &= \begin{cases} 1, & \text{시설이 } i \text{에 설치되는 경우} \\ 0, & \text{그렇지 않은 경우} \end{cases} \end{aligned}$$

p : 설치될 시설 개수
 d_j : 노드 j 에 거주하는 주민수 (또는 요구량)
 c_{ij} : 시설 i 와 해당 구역 j 간 소요비용 ($= d_j \times l_{ij}$)

$$\min \sum_{i=1}^p \sum_{j=1}^n l_{ij} x_{ij} \text{ or } \min \sum_{i=1}^p \sum_{j=1}^n c_{ij} x_{ij} \quad (1)$$

subject to

$$\sum_{i \in F_i} x_{ij} = 1, \forall j \in L_j, \sum_{i \in F_i} y_i = p$$

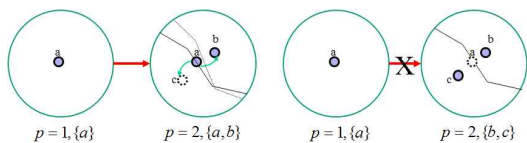
$$x_{ij} \leq y_i, \forall j \in L_j, \forall i \in F_i,$$

$$y_i \in \{0,1\}, x_{ij} \in \{0,1\}$$

MA는 그림 1과 같이 간단히 수행된다. 이 알고리즘은 쉬운 반면에 일단 해로 결정되면 더 이상의 변경이 되지 않으면서 계속적으로 시설 위치가 추가되는 개념으로 p 번째로 얻은 최종 해가 최적 해에서 멀어질 수 있는 단점이 있다[4,8,9]. 이는 그림 2에서 설명될 수 있다. $p=1$ 인 경우 n 개 노드들 중에서 가장 중심에 위치하는 노드를 a 라 하면, $p=2$ 인 경우, n 개 노드들을 양분하는 2 구역의 중심점들인 b, c 로 결정되어야 한다. 그러나 MA의 경우 2개 노드중 하나인 a 가 고정된 상태에서 나머지 하나를 선택하는 관계로 a, b 또는 a, c 가 될 수 있으며, 최악의 경우 a, d 로 $p=2$ 의 최적 노드 위치인 b, c 중 어느 것도 포함하지 않을 수도 있다.

- Step 1. 모든 구역의 고객들에 대해 최소 비용인 시설을 $p=1$ 로 선택한다.
- Step 2. 시설이 p 개가 될 때까지 최소 비용 시설을 한 번에 하나씩 추가한다.

그림 1. MA
 Fig. 1. Myoptic Algorithm



(a) $p-1$ plus one added (b) random selected
 그림 2. 최적 해 결정 방법

Fig. 2. Optimal solution decision making methods

Caccetta와 Dzator[4]는 감소 휴리스틱 (reduction heuristic, RH) 알고리즘을 제안하였다. MA는 최단거리행렬에 존재하는 모든 값들을 사용하는 반면에, RH 알고리즘은 극단적인 값 (이상점, outliers)을 삭제하는 방법이다. 왜냐하면 이상점들은 최적 해에 강력한 영향을 미치기 때문이다. RH 알고리즘은 n 이 증가함에 따라 그림 3의 RH1+RH2+RRH (repeated RH)를 수행하도록 설계되어 있다.

만약, n 이 작은 경우는 RH1으로 최적 해를 얻을 수 있지만 n 이 증가함에 따라 RH2로도 해를 얻지 못해 RRH까지도 수행해야 하는 경우가 발생할 수 있다. 표 1의 데이터에 대해 $p=2$ 인 경우 RH1은 그림 4와 같이 수행된다. 참고로, Caccetta와 Dzator[4]는 표 1의 C_1 행렬에 대해 $p=2$ 인 경우 $p = \{1,2\}$, $p=3$ 인 경우 $p = \{1,2,5\}$ 를 얻었다. MA는 표 2와 같이 수행된다.

- (RH1)
- Step 1. 시설 수 p (열)과 노드 수 n (행)을 동일하게 설정한 비용 행렬 C 를 작성한다.
 - Step 2. 각 열에서 최대값 내림차순으로 p 개를 "0"으로 치환한다.
 - Step 3. 각 열의 합을 계산한다. 열의 합을 오름차순으로 하여 최소값 p 개를 초기해 집합으로 선택한다.
 - Step 4. 원래의 C 비용행렬에서 선택된 p 개의 노드 행과 열의 모든 c_{ij} 를 "0"으로 치환하고, 열의 합을 계산한다.
 - Step 5. 합이 0이 아닌 열의 최소값 노드 1개를 초기값 노드 p 개의 초기 노드와 각각 치환한 조합 집합을 생성한다. /* p 회
 - Step 6. 각 집합의 합이 최소인 집합을 선택한다. 그렇지 않으면 Step 3으로 복귀하고, 보다 작은 값을 가진 집합을 최종 해로 선택한다.
- (RH2)
- Step 1. 시설 수 p (열)과 노드 수 n (행)을 동일하게 설정한 비용 행렬 C 를 작성한다.
 - Step 2. 각 열에서 최대값 내림차순으로 p 개를 "0"으로 치환한다.
 - Step 3. 각 열의 합을 계산한다. 열의 합을 오름차순으로 하여 최소값 p 개를 초기해 집합으로 선택한다.
 - Step 5. 초기 집합 p 에 속하지 않는 모든 노드를 대상으로 p 에 속한 1개 노드와 치환한 조합 집합을 생성한다. /* $(n-p)p$ 회
 - Step 6. 집합의 합이 최소인 집합을 선택한다. 그렇지 않으면 보다 작은 값을 가진 집합을 최종 해로 선택한다.

- (RRH)
- 최종 해를 개선할 수 있을 때까지 다음을 반복 수행한다.
- Step 1. RH2의 최종 해를 초기집합으로 설정한다.
 - Step 2. 원래의 C 비용행렬에서 선택된 p 개의 노드 행과 열의 모든 c_{ij} 를 "0"으로 치환하고, 열의 합을 계산한다.
 - Step 3. 초기 집합 p 에 속하지 않는 모든 노드를 대상으로 p 에 속한 1개 노드와 치환한 조합 집합을 생성한다. /* $(n-p)p$ 회
 - Step 4. 집합의 합이 최소인 집합을 선택한다. 그렇지 않으면 보다 작은 값을 가진 집합을 최종 해로 선택한다.

그림 3. RH 알고리즘
 Fig. 3. RH Algorithm

표 1. 비용 행렬 데이터 C_1

Table 1. Cost Matrix Data C_1

$L_j \backslash F_i$	1	2	3	4	5
1	0	82	37	51	100
2	67	0	78	93	97
3	74	18	0	20	49
4	20	87	27	0	66
5	62	37	51	87	0

	1	2	3	4	5
1	0	82	37	51	100
2	67	0	78	93	97
3	74	18	0	20	49
4	20	87	27	0	66
5	62	37	51	87	0

(a) Step 1

	1	2	3	4	5
1	0	0	37	51	0
2	0	0	0	0	0
3	0	18	0	20	49
4	20	0	27	0	66
5	62	37	0	0	0

(b) Step 2

	1	2	3	4	5
1	0	0	37	51	0
2	0	0	0	0	0
3	0	18	0	20	49
4	20	0	27	0	66
5	62	37	0	0	0
합	82	85	64	71	115

(c) Step 3

	1	4	5
1	0	51	100
4	20	0	66
5	62	87	0
합	82	138	166

(d) Step 4

{2,3} → {2,1}, {1,3}

(e) Step 5

	1	2	3
1	0	82	0
2	67	0	67
3	74	18	0
4	20	87	27
5	62	37	51
합	75		145

최적 해 : {1,2}

(f) Step 6

그림 4. C_1 에 대한 $p = 2$ RH1 알고리즘

Fig. 4. RH1 Algorithm for $p = 2$ of C_1

표 2. C_1 에 대한 MA

Table. 2. MA for C_1

	1	2	3	4	5	합
1	0	67	74	20	62	223
2	82	0	18	87	37	224
3	37	78	0	27	51	193
4	51	93	20	0	87	251
5	100	97	49	66	0	312
$p = \{3\}, c = 193$						Min: 193
$p = 1$	1	2	3	4	5	합
3	37	78	0	27	51	193
1	0	67	74	20	62	
2	82	0	18	87	37	
4	51	93	20	0	87	
5	100	97	49	66	0	
1	-37	-11		-7		-55
2		-78			-14	-92
4				-27		-27
5					-51	-51
$p = \{3,2\}, c = 193 - 92 = 101$						Min: -92
$p = 2$	1	2	3	4	5	합
3	37	78	0	27	51	
2	82	0	18	87	37	
Min:	37	0	0	27	37	101
1	0	67	74	20	62	
4	51	93	20	0	87	
5	100	97	49	66	0	
1	-37			-7		-44
4				-27		-27
5					-37	-37
$p = \{3,2,1\}, c = 101 - 44 = 57$						Min: -44

$p = 3$	1	2	3	4	5	합
3	37	78	0	27	51	
2	82	0	18	87	37	
1	0	67	74	20	62	
Min:	0	0	0	20	37	57
4	51	93	20	0	87	
5	100	97	49	66	0	

4				-20		-20
5					-37	-37
$p = \{3,2,1,5\}, c = 57 - 37 = 20$						Min: -37

$p = 4$	1	2	3	4	5	합
3	37	78	0	27	51	
2	82	0	18	87	37	
1	0	67	74	20	62	
5	100	97	49	66	0	
Min:	0	0	0	20	0	20
4	51	93	20	0	87	

$p = \{3,2,1,5,4\}, c = 20 - 20 = 0$

III. 시설 위치 선정 구성 알고리즘

본 장에서는 MA가 최적 해에서 멀어지는 단점을 보완한 개선된 MA를 제안한다. MA는 단순히 $p-1$ 개의 시설을 고정시키고, 비용이 최소가 되는 위치에 p 번째 시설을 결정하는 방식이다. 제안된 방법은 $p-1$ 의 노드들로부터 p 번째 노드를 추가하는 방식으로, $p = 1, 2, \dots$ 로 증가 (증대)하는 구성방식을 채택한다.

$[1, p-1]$ 에 대해 $p-1$ 번째를 u , $[1, p-1]$ 에 기반하여 찾은 p 번째 시설을 v 라 하자. $p_1 + p_2 + \dots + u + v$ 를 집합 P_0 로, u 와 v 를 교환한 $p_1 + p_2 + \dots + v$ 에 기반하여 찾은 p 번째 시설을 w 라 할 때, $p_1 + p_2 + \dots + v + w$ 를 집합 P_1 이라 하자.

제안된 알고리즘은 $p \geq 2$ 에 대해 P_0 와 P_1 의 비용이 최소인 시설 집합을 p 로 선택한다. 이 알고리즘은 $p=1$ 부터 $p=n-1$ 까지 순차적으로 후보를 추가하는 방법으로 이후부터는 구성 알고리즘 (CA)이라 칭한다. CA는 그림 5와 같이 수행된다.

사전처리 (Pre-processing)

- 시설 수 F_i ($i = 1, 2, \dots, m$) 행과 노드 수 L_j ($j = 1, 2, \dots, n$) 열의 거리행렬 L 작성.
- L 행렬을 F_i 와 L_j 간 최단거리 l_{ij} 행렬 S 로 변환.
- S 행렬을 F_i 와 L_j 의 비용 $c_{ij} = d_j \times l_{ij}$ 행렬 C 로 변환.

초기치 : $p = \{1, 2, \dots, m\}$.

Step 1. F_i 행의 합이 최소인 노드를 $p = 1$ 로 결정.

Step 2. for $2 \leq p \leq n-1$ do

- (1) $p-1$ 개 노드를 열의 최소값 $\min c_j$ 를 구함.
- (2) $n - (p-1)$ 개 노드들의 열 c_{ij} 가 $\min c_j$ 보다 작으면 $c_{ij} = \min c_j - c_{ij}$ 를 계산하여 합을 구하고 최소 합 노드 u 를 p 번째 시설로 결정한 집합 P_0 생성.
- (3) P_0 집합에서 p 번째 u 와 $p-1$ 번째 v 를 교체한

$p-1$ 개 노드들에 대해 열 c_{ij} 가 $\min c_j$ 보다 작으면 $c_{ij} = \min c_j - c_{ij}$ 를 계산하여 합을 구하고 최소 합 노드 w 를 p 번째 시설로 결정한 집합 P_1 생성.
 (4) $\min\{c(P_0), c(P_1)\}$ 을 p 로 결정.
 end

그림 5. 구성 알고리즘
 Fig. 5. Constructive algorithm (CA)

제안된 CA를 Choi et al.[10]의 RD와 비교한 결과는 그림 6과 같다. CA는 Choi et al.[10]이 제안한 $p=n, n-1, n-2, \dots$ 의 역 삭제 (RD) 방법을 역으로 수행하는 차이가 있다. 또 한 가지 차이점은 p 번째 노드를 그대로 결정하지 않고, 나머지 노드들과 교환 (대체)하는 기법을 적용하여 보다 최적의 노드를 찾는 특징이 있다. ${}_n C_p$ 의 조합 이외에는 p 개의 노드를 정확히 찾을 수 있는 방법은 존재하지 않는다. 따라서, CA는 기존에 알려진 방법들에 비해 보다 간단하면서도 근사하게 p 를 찾는 방법임을 보이고자 한다.

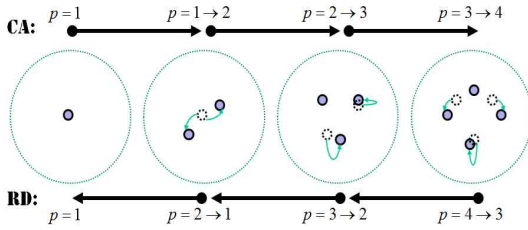


그림 6. CA 대 RD
 Fig. 6. CA vs. RD

표 1의 C_1 행렬 데이터에 제안된 CA를 적용하는 과정은 표 3과 같다.

표 3. C_1 에 대한 CA
 Table. 3. CA for C_1

	1	2	3	4	5	합
1	0	67	74	20	62	223
2	82	0	18	87	37	224
3	37	78	0	27	51	193
4	51	93	20	0	87	251
5	100	97	49	66	0	312
$p = \{3\}, c = 193$						Min: 193
$p=1$	1	2	3	4	5	합
3	37	78	0	27	51	193
1	0	67	74	20	62	
2	82	0	18	87	37	
4	51	93	20	0	87	
5	100	97	49	66	0	
1	-37	-11		-7		-55
2		-78			-14	-92
4				-27		-27
5					-51	-51
$p = \{3,2\}, c = 193 - 92 = 101$						Min: -92

	1	2	3	4	5	합
2	82	0	18	87	37	224
1	0	67	74	20	62	
3	37	78	0	27	51	
4	51	93	20	0	87	
5	100	97	49	66	0	

1	-82			-67		-149
3	-45		-18	-60		-123
4	-31			-87		-118
5				-21	-37	-58
$p = \{2,1\}, c = 224 - 149 = 75$						Min: -149

$p=2$	1	2	3	4	5	합
2	82	0	18	87	37	
1	0	67	74	20	62	
Min:	0	0	18	20	37	75
3	37	78	0	27	51	
4	51	93	20	0	87	
5	100	97	49	66	0	

3			-18			-18
4				-20		-20
5					-37	-37
$p = \{2,1,5\}, c = 75 - 37 = 38$						Min: -37

	1	2	3	4	5	합
2	82	0	18	87	37	
5	100	97	49	66	0	
Min:	82	0	18	66	0	166
1	0	67	74	20	62	
3	37	78	0	27	51	
4	51	93	20	0	87	

1	-82			-46		-128
3	-45		-18	-39		-102
4	-31			-66		-97
$p = \{2,5,1\}, c = 166 - 128 = 38$						Min: -128

$p=3$	1	2	3	4	5	합
2	82	0	18	87	37	
1	0	67	74	20	62	
5	100	97	49	66	0	
Min:	0	0	18	20	0	38
3	37	78	0	27	51	
4	51	93	20	0	87	

3			-18			-18
4				-20		-20
$p = \{2,1,5,4\}, c = 38 - 20 = 18$						Min: -20

	1	2	3	4	5	합
2	82	0	18	87	37	
1	0	67	74	20	62	
4	51	93	20	0	87	
Min:	0	0	18	0	37	55
3	37	78	0	27	51	
5	100	97	49	66	0	

3			-18			-18
5					-37	-37
$p = \{2,1,4,5\}, c = 55 - 37 = 18$						Min: -37

표 3에서, $p=1$ 인 경우 당연히 $p=\{3\}$ 이 되어야 한다. 왜냐하면 노드 3으로부터 노드 1,2,4,5와의 거리 합이 최소 (가장 중심에 위치)가 되기 때문이다. $p=2$ 인 경우, $p=\{3\}$ 에 나머지 1,2,4,5 중 2 노드를 추가하여 $p=\{2,3\}$ 이 될 경우 $p=1$ 인 3 노드가 커버하던 1,2,4,5들 중에서 2와 5를 2 노드가 커버하여 최대의 거리인 92를 감소시킬 수 있음을 알 수 있다. 따라서, $p=\{2,3\}$ 의 $p=\{2\}$ 에 대해 다시 수행한 결과 $p=\{2,1\}$ 의 조합이 나머지 3,4,5와의 거리가 최소인

$c=75$ 를 나타내어 $p=2$ 는 $p=\{2,3\}$ 이 아닌 $p=\{1,2\}$ 로 변경되었다. 이러한 방식으로 수행한 결과 $p=3$ 인 경우 $p=\{2,5,1\}, p=4$ 인 경우 $p=\{2,1,4,5\}$ 를 얻었다.

제안된 CA와 MA는 엑셀을 활용하여 수행하였다. C_1 데이터에 대해 알고리즘 수행 결과를 비교하면 표 4와 같다. 제안된 CA는 RH1+RH2+RRH 알고리즘에 비해 매우 간단함에도 불구하고, 최적 해를 구할 수 있음을 알 수 있다. 또한, MA가 최적 해에서 멀어지는 단점을 보완하였음을 알 수 있다. 또한, 제안 알고리즘은 전문가가 아니더라도 엑셀을 활용하여 해를 구할 수 있어 일반적으로 적용이 가능하다.

표 4. C_1 에 대한 알고리즘 결과 비교
Table. 4. Result of algorithms for C_1

p	알고리즘		
	RH1+RH2+RRH	MA	CA
1	-	$p=\{3\}, c=193$	$p=\{3\}, c=193$
2	$p=\{1,2\}, c=75$	$p=\{3,2\}, c=101$	$p=\{2,1\}, c=75$
3	$p=\{1,2,5\}, c=38$	$p=\{3,2,1\}, c=57$	$p=\{2,1,5\}, c=38$
4	-	$p=\{3,2,1,5\}, c=20$	$p=\{2,1,5,4\}, c=18$
5	-	$p=\{3,2,1,5,4\}, c=0$	$p=\{2,1,5,4,3\}, c=0$

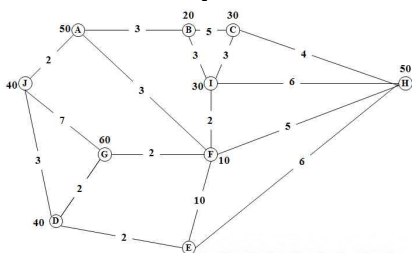
IV. 실험 및 결과 분석

본 장에서는 그림 7의 Eiselt와 Sandblom[1]의 C_2 , Dibble[11]의 C_3 와 Swain의 55-노드 망[5-7] 데이터에 적용하여 본다.

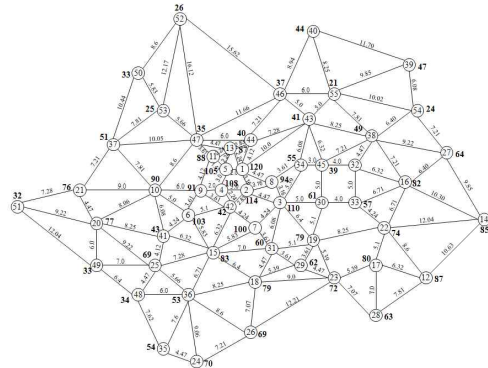
C_2, C_3 와 Swain의 55-노드 망 데이터에 제안된 CA를 적용한 결과는 표 5 ~ 표 7에 제시되어 있다. 각 표에서 얻은 방법은 3장에서 제안된 CA에 기반하고 있으며, 표 3의 C_1 수행 방법과 동일하여 여기서는 상세한 설명을 생략한다.

l_{ij}	L_{ij}							합
	1	2	3	4	5	6	7	
F_1	1	0	3	7	7	10	15	49
	2	3	0	4	10	9	7	45
	3	7	4	0	14	8	3	44
	4	7	10	14	0	6	14	73
	5	7	9	8	6	0	8	54
	6	10	7	3	14	8	0	51
	7	15	12	8	22	16	9	82

$p=3, p=\{1,3,7\}$ or $\{1,6,7\}$
(a) C_2 데이터



(b) C_3 데이터



(c) Swain의 55-노드
그림 7. 실험 데이터

Fig. 7. Experimental Data

표 5. C_2 에 대한 CA
Table. 5. CA for C_2

p	CA
1	$p=\{3\}, c=44$
2	$3(44)+1(-16)=28, 3(44)+4(-16)=28, 3(44)+5(-16)=28$ $1(49)+3(-21)=28, 4(73)+3(-45)=28, 5(54)+3(-26)=28$ $p=\{3,1\}, \{3,4\}, \{3,5\}, c=28$
3	$3+1(28)+4(-8)=20, 3+1(28)+5(-8)=20, 3+1(28)+7(-8)=20$ $p=\{3,1,4\}, \{3,1,5\}, \{3,1,7\}, \{3,4,2\}, \{3,4,7\}, \{3,5,2\}, \{3,5,7\}, c=20$
4	$3+1+4(20)+7(-8)=12, 3+1+7(20)+5(-8)=12, 3+1+7(20)+4(-8)=12$ $p=\{3,1,4,7\}, \{3,1,7,5\}, c=12$
5	$3+1+4+7(12)+5(-6)=6, 3+1+4+5(14)+7(-8)=12$ $p=\{3,1,4,7,5\}, c=6, c=12$
6	$3+1+4+7+5(6)+2(-3)=3, 3+1+4+7+5(6)+6(-3)=3$ $3+1+4+7+6(9)+5(-6)=3$ $p=\{3,1,4,7,5,2\}, \{3,1,4,7,5,6\}, c=3$
7	$p=\{1,2,3,4,5,6,7\}, c=0$
p	Eiselt & Sandblom[1] with p-Center
3	$p=\{1,3,7\}$ or $\{1,6,7\}, c=20$

표 6. C_3 에 대한 CA
Table. 6. CA for C_3

p	CA
1	$p=\{F\}, c=1490$
2	$F(1490)+D(-440)=1050, D(1690)+C(-670)=1020, p=\{D,C\}, c=1,020$ $D+C(1020)+A(-340)=680, D+A(1180)+H(-520)=660$
3	$p=\{D,A,H\}, c=660$ $D+A+H(660)+I(-190)=470, D+A+I(770)+H(-300)=470$
4	$p=\{D,A,H,I\}, c=470$ $D+A+H+I(470)+G(-120)=350, D+A+H+G(500)+I(-150),$ or $C(-150)=350$ $p=\{D,A,H,I,G\}, \{D,A,H,G,C\}, c=350$
6	$D+A+H+I+G(350)+E(-100)=250, D+A+H+I+E(370)+G(-120)=250$ $p=\{D,A,H,I,G,E\}, c=250$
7	$D+A+H+I+G+E(250)+C(-90)=160, D+A+H+I+G+C(260)+E(-100)=160$ $p=\{D,A,H,I,G,E,C\}, c=160$
8	$D+A+H+I+G+E+C(160)+J(-80)=80, D+A+H+I+G+E+J(170)+C(-90)=80$ $p=\{D,A,H,I,G,E,C,J\}, c=80$
9	$D+A+H+I+G+E+C+J(80)+B(-60)=20, D+A+H+I+G+E+C+B(100)+J(-80)=20$ $p=\{D,A,H,I,G,E,C,J,B\}, c=20$
10	$p=\{D,A,H,I,G,E,C,J,B,F\}, c=0$

p	Dibble[11] by Hand
2	$p=\{G,H\}, c=1,130$

Eiselt와 Sandblom[1]는 C_2 데이터의 $p=3$ 에 대해 p -중심 방법으로 해를 구하였으며, 제안된 CA는 $p=[1,7]$ 에 대해 해를 구하였다. $p=3$ 에 대해 Eiselt와 Sandblom[1]는 2개 해를 구한 반면 CA는 7개를 구하였다. C_3 데이터에 CA를

적용한 결과 Dibble[11]가 $p=2$ 에 대해 수기식으로 해를 잘못 구하였음을 알 수 있다.

표 7. Swain의 55-노드 망의 CA
Table 7. CA for Swain's 55-Node Network

p	CA	
	시설 위치	비용
1	$p(1) = \{3\}$	48,690.8
2	$3(48690.8)+4(-9183.2)=-3607.6$ 4(49303.1)+22(-13143.3)=-36159.8 $\rightarrow p(2) = \{4, 22\}$	36,159.8
3	$4+22(36159.8)+36(-5623.9)=-30535.9 \rightarrow p(3) = p(2) + 36$	30,535.9
4	$p(2)+36(30535.9)+38(-3391.1)=-27144.8 \rightarrow p(4) = p(3) + 38$	27,144.8
5	$p(3)+38(27144.8)+21(-3201.2)=-23943.6 \rightarrow p(5) = p(4) + 21$	23,943.6
6	$p(4)+21(23943.6)+3(-2287.5)=-21656.1 \rightarrow p(6) = p(5) + 3$	21,656.1
7	$p(5)+3(21656.1)+13(-1747.3)=-19908.8 \rightarrow p(7) = p(6) + 13$	19,908.8
8	$p(6)+13(19908.8)+29(-1664.2)=-18244.6 \rightarrow p(8) = p(7) + 29$	18,244.6
9	$p(7)+29(18244.6)+12(-1103.0)=-17141.6 \rightarrow p(9) = p(8) + 12$	17,141.6
10	$p(8)+12(17141.6)+46(-985.2)=-16156.4 \rightarrow p(10) = p(9) + 46$	16,156.4
11	$p(9)+46(16156.4)+6(-979.7)=-15176.7 \rightarrow p(11) = p(10) + 6$	15,176.7
12	$p(10)+6(15176.7)+14(903.6)=-14273.1 \rightarrow p(12) = p(11) + 14$	14,273.1
13	$p(11)+14(14273.1)+24(-899.1)=-13374.0 \rightarrow p(13) = p(12) + 24$	13,374.0
14	$p(12)+24(13374.0)+50(-816.7)=-12577.3 \rightarrow p(14) = p(13) + 50$	12,577.3
15	$p(13)+50(12577.3)+16(-730.7)=-11826.6 \rightarrow p(15) = p(14) + 16$	11,826.6
16	$p(14)+16(11826.6)+39(-594.3)=-11232.3 \rightarrow p(16) = p(15) + 39$	11,232.3
17	$p(15)+39(11232.3)+25(-562.1)=-10702.2 \rightarrow p(17) = p(16) + 25$	10,702.2
18	$p(16)+25(10702.2)+23(-497.5)=-10172.7 \rightarrow p(18) = p(17) + 23$	10,172.7
19	$p(17)+23(10172.7)+28(-492.0)=-9880.7 \rightarrow p(19) = p(18) + 28$	9,880.7
20	$p(18)+28(9880.7)+15(-483.9)=-9196.8 \rightarrow p(20) = p(19) + 15$	9,196.8
21	$p(19)+15(9196.8)+34(-407.7)=-8736.1 \rightarrow p(21) = p(20) + 34$	8,736.1
22	$p(20)+34(8736.1)+10(-450.0)=-8286.1 \rightarrow p(22) = p(21) + 10$	8,286.1
23	$p(21)+10(8286.1)+18(-425.8)=-7800.3 \rightarrow p(23) = p(22) + 18$	7,800.3
24	$p(22)+18(7800.3)+7(-424.0)=-7436.3 \rightarrow p(24) = p(23) + 7$	7,436.3
25	$p(23)+7(7436.3)+27(-400.6)=-7026.7 \rightarrow p(25) = p(24) + 27$	7,026.7
26	$p(24)+27(7026.7)+17(-408.0)=-6618.7 \rightarrow p(26) = p(25) + 17$	6,618.7
27	$p(25)+17(6618.7)+40(-384.4)=-6225.3 \rightarrow p(27) = p(26) + 40$	6,225.3
28	$p(26)+40(6225.3)+20(-377.2)=-5848.1 \rightarrow p(28) = p(27) + 20$	5,848.1
29	$p(27)+20(5848.1)+37(-357.7)=-5480.4 \rightarrow p(29) = p(28) + 37$	5,480.4
30	$p(28)+37(5480.4)+2(-342.0)=-5138.4 \rightarrow p(30) = p(29) + 2$	5,138.4
31	$p(29)+2(5138.4)+5(-337.0)=-4801.4 \rightarrow p(31) = p(30) + 5$	4,801.4
32	$p(30)+5(4801.4)+23(-321.8)=-4479.6 \rightarrow p(32) = p(31) + 23$	4,479.6
33	$p(31)+23(4479.6)+30(-318.7)=-4160.9 \rightarrow p(33) = p(32) + 30$	4,160.9
34	$4160.9+8(-297.0)=-3863.9 \rightarrow p(34) = p(33) + 8$	3,863.9
35	$3863.9+19(-285.2)=-3578.7 \rightarrow p(35) = p(34) + 19$	3,578.7
36	$3578.7+32(-239.3)=-3319.4 \rightarrow p(36) = p(35) + 32$	3,319.4
37	$3319.4+35(-241.4)=-3078.0 \rightarrow p(37) = p(36) + 35$	3,078.0
38	$3078.0+1(-240.0)=-2838.0 \rightarrow p(38) = p(37) + 1$	2,838.0
39	$2838.0+51(-233.0)=-2605.0 \rightarrow p(39) = p(38) + 51$	2,605.0
40	$2605.0+33(-228.0)=-2377.0 \rightarrow p(40) = p(39) + 33$	2,377.0
41	$2377.0+43(-226.0)=-2151.0 \rightarrow p(41) = p(40) + 43$	2,151.0
42	$2151.0+52(-223.6)=-1927.4 \rightarrow p(42) = p(41) + 52$	1,927.4
43	$1927.4+31(-216.6)=-1710.8 \rightarrow p(43) = p(42) + 31$	1,710.8
44	$1710.8+49(-198.0)=-1512.8 \rightarrow p(44) = p(43) + 49$	1,512.8
45	$1512.8+9(-182.0)=-1330.8 \rightarrow p(45) = p(44) + 9$	1,330.8
46	$1330.8+41(-177.2)=-1153.6 \rightarrow p(46) = p(45) + 41$	1,153.6
47	$1153.6+47(-160.8)=-992.8 \rightarrow p(47) = p(46) + 47$	992.8
48	$992.8+48(-152.0)=-840.8 \rightarrow p(48) = p(47) + 48$	840.8
49	$840.8+54(-145.9)=-694.9 \rightarrow p(49) = p(48) + 54$	694.9
50	$694.9+53(-141.5)=-553.4 \rightarrow p(50) = p(49) + 53$	553.4
51	$553.4+11(-124.1)=-429.3 \rightarrow p(51) = p(50) + 11$	429.3
52	$429.3+45(-117.0)=-312.3 \rightarrow p(52) = p(51) + 45$	312.3
53	$312.3+44(-113.2)=-199.1 \rightarrow p(53) = p(52) + 44$	199.1
54	$199.1+55(-105.0)=-94.1 \rightarrow p(54) = p(53) + 55$	94.1
55	$94.1+42(94.1)-0 \rightarrow p(55) = p(54) + 42$	0.0

Swain-55 노드 망에 대해 CA로 얻은 표 7의 해에서 $p=4, p=5$ 에 대해 Serra et al.[5], Serra와 Revelle[6,7]와 Choi et al.[10]의 RD로 구한 결과와 비교하면 표 8과 같다.

$p=4$ 인 경우 Serra et al.[5]의 26,922.7에 비해 CA의 해는 0.8250% 초과하였으며, $p=5$ 인 경우 Serra와 Revelle[6]의 26,116.1에 비해 8.32% 감소한 결과를 얻었다. 또한,

RD에 비해 CA는 $p=4, p=5$ 모두에서 보다 좋은 결과를 얻었다.

표 8. Swain의 55-노드 망의 p 결과 비교
Table 8. Comparison of p results for Swain's 55-Node Network

p	p	알고리즘		소요비용
		알고리즘	소요비용	
4	$p = \{1, 16, 29, 41\}$ $p = \{4, 22, 36, 38\}$ $p = \{5, 16, 29, 36\}$	Serra et al.[5]	26,922.7	
		CA	27,144.8	
		RD[10]	27,870.0	
5	$p = \{4, 22, 36, 38, 21\}$ $p = \{5, 16, 20, 29, 36\}$ $p = \{6, 8, 13, 16, 31\}$ $p = \{5, 11, 12, 54, 55\}$	CA	23,943.6	
		RD[10]	24,391.8	
		Serra-Revelle [6]	26,116.1	
		Serra-Revelle [7]	36,246.1	

결론적으로, 우리는 n 개 노드 중에서 p 개를 선택하는 경우, 대부분 $2 \leq p < n/2$ 의 범위를 갖고 있기 때문에, $n, n-1, n-2, \dots$ 의 RD 보다는 1, 2, ...의 CA가 보다 수행횟수가 적은 장점을 갖고 있다. 또한, RD를 역으로 수행하는 CA의 구성 방법이 보다 좋은 결과를 얻을 수 있음을 표 9로부터 알 수 있다. 따라서, RD 보다는 CA를 추천한다.

표 9. Swain의 55-노드 망의 CA vs. RD 비용
Table 9. CA vs RD cost for Swain 55-node network

p	소요비용		p	소요비용	
	RD[10]	CA		RD[10]	CA
1	48,690.8	48,690.8	29	8,150.4	5,480.4
2	40,234.7	36,159.8	30	7,968.4	5,138.4
3	32,263.5	30,535.9	31	7,791.2	4,801.4
4	27,870.0	27,144.8	32	7,634.7	4,479.6
5	24,391.8	23,943.6	33	7,510.6	4,160.9
6	22,504.8	21,656.1	34	7,393.6	3,863.9
7	20,536.4	19,908.8	35	7,280.4	3,578.7
8	18,560.3	18,244.6	36	7,186.3	3,319.4
9	17,126.1	17,141.6	37	-	3,078.0
10	16,076.1	16,156.4	38	-	2,838.0
11	15,343.8	15,176.7	39	-	2,605.0
12	14,472.0	14,273.1	40	-	2,377.0
13	13,940.6	13,374.0	41	-	2,151.0
14	13,393.4	12,577.3	42	-	1,927.4
15	12,945.8	11,826.6	43	-	1,710.8
16	12,450.5	11,232.3	44	-	1,512.8
17	12,146.6	10,670.2	45	-	1,330.8
18	11,696.2	10,172.7	46	-	1,153.6
19	11,212.3	9,690.7	47	-	992.8
20	10,745.9	9,196.8	48	-	840.8
21	10,295.9	8,736.1	49	-	694.9
22	10,063.9	8,286.1	50	-	553.4
23	9,676.5	7,860.3	51	-	429.3
24	9,379.5	7,436.3	52	-	312.3
25	9,094.3	7,026.7	53	-	199.1
26	8,835.0	6,618.7	54	-	94.1
27	8,595.0	6,225.3	55	-	0.0
28	8,387.0	5,848.1			

제한된 CA는 지금까지 제안된 알고리즘들 중에서 가장 간단히 역셀을 활용해 근사 해를 구할 수 있는 알고리즘이며, 최적 해에 가장 근사한 결과를 얻었다. 제안된 알고리즘을 적용한 결과, 시설 위치 변화의 특징은 $p=2$ 또는 $p=3$ 에서 발생함을 알 수 있었다. 따라서 이 부분에 대해서는 CA로 해를 구하고, $p \geq 4$ 에 대해서는 해가 변경되지 않아 MA와 동

일하게 수행되어도 무방함을 알 수 있다.

V. 결론 및 추후 연구과제

본 논문은 p 개의 시설을 설치할 최적의 위치를 결정하는 간단한 알고리즘을 제안하였다. 이 문제는 다항시간에 최적해를 구할 수 있는 알고리즘이 제안되지 않아 NP-난제로 알려져 있다.

제안된 CA는 MA와 동일하게 $p=1$ 부터 $p=n$ 까지 한 번에 하나씩 p 를 증가시키면서 해를 찾는 구성 알고리즘인 반면에, MA가 최적 해에서 점점 멀어지는 단점을 보완하여 보다 최적 해에 근사한 해를 찾을 수 있었다.

제안된 알고리즘은 $n=5, 7, 10$ 에 대해서는 최적 해를 찾는 데 성공하였지만, $n=55$ 의 $p=4$ 에 대해서는 다른 알고리즘으로 찾은 해에 비해 0.8250%의 오차를 보였다. 그러나 CA와 반대로 수행되는 역 삭제 (RD) 알고리즘과 비교시 $p=4$, $p=5$ 모두에서 보다 좋은 결과를 얻었으며, $n=55$ 에서 5가지 감소시키는 50회에 비해 $p=1$ 부터 5가지 증가시키는 5회로 보다 간단히 해를 구할 수 있음을 알 수 있다.

CA의 특징은 최적의 시설 위치를 찾는 전문가가 아닌 초보자도 상용 패키지인 엑셀을 활용하여 쉽게 구현할 수 있는 장점이 있다.

제안된 CA는 Swain-55 노드 망의 $p=4$ 에 대해 Serra et al.[5] 보다는 나쁜 결과를 얻었다. 따라서, 추후 CA의 기본 개념에는 충실하면서 Serra et al.[5]의 결과를 얻을 수 있는 방법을 연구할 예정이다.

REFERENCES

- [1] H. A. Eiselt and C-L, Sandblom, "Decision Analysis, Location Models, and Scheduling Problems, Part II: Location and Layout Decisions, Chapter 2. Location models on Networks", Springer-Verlag Berlin Heidelberg, pp. 169-210, 2004.
- [2] H. Jia, F. Ordóñez, and M. Dessouky, "A Modeling Framework for Facility Location of Medical Services for Large-Scale Emergencies", IIE Transactions, Vol. 39, Issue. 1, pp. 41-55, 2007.
- [3] H. K. Smith, G. Laporte, and P. R. Harper, "Locational Analysis: Highlights of Growth to Maturity", Journal of the Operational Research Society, Vol. 60, Supplement 1, pp. s140-s148, 2009.
- [4] L. Caccetta and M. Dzator, "Heuristic Methods for Locating Emergency Facilities", Annual Conference of Economists ACE07, The Economic Society of Australia, 2007.
- [5] D. Serra, C. Revelle, and K. Rosing, "Surviving in a Competitive Spatial Market: The Threshold Capture Model", Journal of Regional Science, Vol. 39, Issue. 4, pp. 637-650, 1999.
- [6] D. Serra, and C. Revelle, "Market Capture by Two Competitors: The Pre-Emptive Location Problem", Economics Working Paper 39, Dept. of Economics and Business, Universitat Pompeu Fabra, 1993.
- [7] D. Serra, and C. Revelle, "Competitive Location in Discrete Space", Economics Working Paper 96, Dept. of Economics and Business, Universitat Pompeu Fabra, 1994.
- [8] S. H. Owen and M. S. Daskin, "Strategic Facility Location: A Review", European Journal of Operational Research, Vol. 11, Issue. 3, pp. 423-447, 1998.
- [9] M. Dzator and J. A. Dzator, "Locating Emergency Facilities: Targeting Efficiency and Cost-Effectiveness", 36th Australian Conference of Economists, Hobart, 2007.
- [10] M. B. Choi, S. U. Lee, B. K. Kim, S. S. Jeung, and T. Y. Han, "p-Facility Location Models," Journal of IIBC, Vol. 11, No. 6, pp. 193-205, Dec. 2011.
- [11] C. Dibble, "GEOG 430/788P, Location Theory and Spatial Analysis," http://www.geog.umd.edu/homepage/courses/cdibble/geog430/lectures/geog430_09_x1.pdf, 2003.

저자 소개



이 상 운

1983년 ~ 1987년 :
한국항공대학교 항공전자공학과 (학사)

1995년 ~ 1997년 :
경상대학교 컴퓨터학과 (석사)

1998년 ~ 2001년 :
경상대학교 컴퓨터학과 (박사)

2003.3 ~ 2015.3 :
강릉원주대학교 멀티미디어공학과 부교수

2015.4 ~ 현 재 :
강릉원주대학교 멀티미디어공학과 정교수
관심분야 : 소프트웨어 프로젝트 관리,

소프트웨어 개발 방법론,
소프트웨어 신뢰성,
그래프 알고리즘

e-mail : sulee@gwnu.ac.kr