The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 18, No. 5, pp.251-257, Oct. 31, 2018. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2018.18.5.251
JIIBC 2018-5-33

독점시장에서 경쟁우위 확보를 위한 신설점포 위치 결정 알고리즘

New Store Positioning Algorithm to ensure Competitive Advantage in Monopoly Market

이상운*

Sang-Un Lee*

요 약 경쟁업체 F_A가 k개 점포로 시장을 독점하고 있는 상황에서, 동종업계의 신규 업체 F_B가 경쟁업체보다 고객을 보다 많이 확보하여 경쟁우위를 가질 수 있는 k 점포를 신규로 개설하고자 한다. 이 경우 어느 위치에 점포를 개설해야 하는가가 문제로 대두된다. 이 문제에 대해 Serra와 Revelle의 PREMAL은 $k-$Median과 LP+BB를 사용한 MAXCAP 정수계획법으로 F_B의 점포 위치 k개를 결정한 다음 F_B의 점포 위치가 보다 많은 고객을 확보하는 위치로 변경하는 방법을 제안하였다. 본 논문에서는 정수계획법 도움 없이 단지 MS-Excel을 활용하여 최 외곽 노드들을 제외하고, F_A 점포 위치 u에 대해 $v \not\in N_G(u)$들을 대상으로 F_B의 점포 위치를 결정하는 비근방 탐색법을 제안하였다. 실험 결과 제안된 알고리즘은 정수계획법에 비해 쉽고 빠르게 해를 구할 수 있었다.

Abstract We will be establish the new k stores of identical product firm F_B to gain competitive advantage over rival firm F_A that has already monopolize a market with k stores. In this situation, how we can decide the location of k stores? For this problem, Serra and Revelle proposes $k-$Median and MAXCAP integer programming using LP+BB to decide the k stores of firm F_B. Then they exchange the k stores to another location that cover more customers. This paper suggests non-neighborhood search method that finds the $v \not\in N_G(u)$nodes for u of firm F_A without most outer loop nodes using just MS-Excel. As a result of experiment, the proposed algorithm can be get the optimal solution easier and faster than integer programming.

Key Words : Monopoly market, Location, Independent degree of contribution, coupling degree of contribution, non-neighborhood search

Ⅰ. 서 론

n개 노드(지점)이 복잡한 도로들로 연결된 하나의 도시에 p명의 주민(population)이 거주하고 있다. 이 도시에 동일 업종의 경쟁업체(firm) F_A가 k개의 점포를 운영하여 도시 전체 주민 수 p명을 독점하여 고객으로 확보하고 있다. 이 상황에서 동일 업종의 동일한 경쟁력과 제품 인지도를 갖는 업체 F_B가 신규로 k개의 점포를 개설

*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과
접수일자 : 2018년 5월 30일, 수정완료 : 2018년 9월 30일
게재확정일자 : 2018년 10월 5일

Received: 30 May, 2018 / Revised: 30 September, 2018 / Accepted: 5 October, 2018
*Corresponding Author: sulee@gwnu.ac.kr
Dept. of Multimedia Eng., Gangneung-Wonju National University, Korea

하여 F_A보다 많은 고객을 확보하고자 한다. 여기서, 고객은 동일 업종의 동일한 경쟁력과 제품 인지도를 갖고 있어 가장 가까운 점포에서 물건을 구입한다고 가정한다. 이 경우 F_B의 신규 개설 점포 k개를 어느 노드에 개설해야 하는가가 문제로 발생한다. 이를 점포 개설 위치 선정 문제(facility location problem, FLP)라 한다.[1]

FLP에 대해, n이 작아 단순한 문제인 경우 전수탐색법(brute-force, BF), 선형계획법(linear programming, LP)이나 분기 한정 법(branch-and-bound, BB)을 적용하여 해를 구할 수 있다.[2,3] 그러나 대규모의 n에 대해서는 이 방법들을 적용하기가 쉽지 않다.

Serra와 Revelle[1]는 $n-k$개 중 신규로 개설할 점포 수 k개와 반복적으로 상호 교환하는 PREMAL (PRe-EMptive heuristic ALgorithm)을 제안하였다. Choi et al.[4]은 망을 k개 영역으로 분할하여 각 영역에서 최대 고객을 확보할 수 있는 지점을 선택하는 분할정복 알고리즘(divide-and-conquer algorithm, DCA)을 제안하였다. 또한, Lee et al.[5]은 구성 알고리즘(constructive algorithm, CA)을 제안하였다. 그러나 이들 방법 역시 n과 p가 항상 변화하는 실제 상황에서 컴퓨터 프로그램의 도움 없이는 해를 구하기가 쉽지 않다.

Choi et al.[4]은 경쟁업체 F_A가 k개 점포를 운영시 F_B가 동일한 k개 점포를 개설하는 경우이며, Lee et al.[5]은 경쟁업체 F_A가 k개 점포를 운영시 F_B가 $1,2,\cdots,k$개 점포를 개설하는 경우로 차이점이 있다. 또한, 경쟁업체가 전무한 상황에서 F_B가 시장을 독점하기 위해 $1,2,\cdots,k$개 점포를 개설하는 경우에 대해서는 Lee[6]이 연구하였다. 따라서 Lee[6]의 연구는 본 논문이 다루는 경쟁업체가 존재하는 경우와 상이하여 이후로는 다루지 않는다.

본 논문에서는 PREMAL, DCA나 CA에 비해 단순하면서도 n과 p가 변하여도 누구나 쉽게 MS-Excel을 활용하여 해를 구할 수 있는 가장 단순한 알고리즘을 제안한다. 2장에서는 관련 연구와 문제점을 고찰한다. 3장에서는 경쟁업체와는 인접하지 않고, 신규 개설 점포들 간에도 인접하지 않아 가장 많은 고객을 확보할 수 있는 위치를 선정하는 알고리즘을 제안한다. 4장에서는 PREMAL, DCA, CA로 해를 얻은 실험 데이터에 대해 제안된 알고리즘을 적용하여 알고리즘 적합성을 검증하여 본다.

II. 관련 연구와 문제점

Serra와 Revelle[1]의 PREMAL이 수행되는 과정은 그림 1과 같다.[4] PREMAL은 $n-k$개 중 신규로 개설할 점포 수 k개와 반복적으로 상호 교환하는 단순한 방법으로 보이지만 실제로는 F_A와 무관한 F_B의 초기 위치를 선정하기 위해 Teiz-Bart, Covering 또는 $k-$Median 방법을 적용해야 하며, F_A를 고려한 F_B 위치를 결정하기 위해 LP+BB를 사용한 MAXCAP 정수계획법을 적용해야 한다. 다시 F_B의 해를 개선할 수 있는 위치로 변경하기 위해 LP+BB를 사용한 MAXCAP 정수계획법을 사용해야 한다. 따라서 PREMAL 알고리즘은 정수계획법의 컴퓨터 프로그램 도움 없이는 해를 구할 수 없다.

Choi et al.[4]의 DCA가 수행되는 과정은 그림 2와 같다. 이 방법은 Step 3을 수행하는 과정이 복잡하여 전문가가 아닌 일반인들은 적용하는데 어려움이 많다.

Step 1. F_A의 k개 점포 위치와 무관하게 F_B의 k개 점포 초기 위치를 결정.[Teiz-Bart, Covering 또는 $k-$Median 방법]
Step 2. F_A의 k개 점포 위치를 고려하여 F_B의 k개 점포 위치를 결정.[LP+BB를 사용한 MAXCAP 정수계획법]
Step 3. F_A와 F_B 각 점포의 시장점유율 계산.
for $i=1$ to $n-2k$
  Step 4. F_A의 k개 점포 위치를 고려하여 F_B의 k개 점포 중 1개 위치를 해를 개선할 수 있는 지역(노드)으로 대체.
  Step 5. F_A의 k개 점포 위치를 고려하여 F_B의 k개 점포 위치 중 1개의 위치를 해를 개선할 수 있는 지역(노드)으로 대체.[LP+BB를 사용한 MAXCAP 정수계획법]
  Step 6. F_A와 F_B 각 점포의 시장점유율 계산. 만약, F_B 시장점유율이 개선되면 새로운 해로 대체.
end

그림 1. PREMAL
Fig. 1. PREMAL

Step 1. F_A의 k개 점포 각각의 시장점유율 (확보고객수와 노드) 계산.
Step 2. F_A 점포 위치를 두 점포가 포함되는 신규 입점 F_B 점포 k개 영역 집합으로 분할.
Step 3. 각 집합이 확보한 노드들 중 최단거리 노드들의 주민 수 합이 최대인 노드를 고객을 최대로 확보할 수 있는 F_B의 점포 위치로 결정. 만약, 해당 점포 위치가 기존에 구한 위치와 중복되면 상대 업체의 시장점유율이 최소로 감소된 노드로 결정.

그림 2. 분할정복 알고리즘
Fig. 2. Divide-and-conquer algorithm

그림 3은 Serra와 Revelle[1]와 Choi et al.[4]와 Lee et al.[5] 이 해를 얻는데 사용한 Swain-55 노드 망이다. 그림 3의 망에 대해, F_A={6,8,13,16,31}인 경우 Serra와 Revelle[1]와 Choi et al.[4], F_A={4,21,22,36,38}인 경우 Lee et al.[5]이 얻은 결과인 F_B는 표 1과 같이 경쟁업체 F_A에 비해 보다 많은 고객을 확보할 수 있는 지점으로 신규 개설 점포 위치를 결정할 수 있었다.

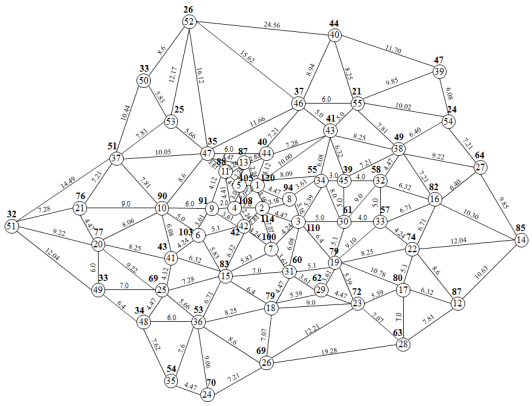


그림 3. Swain 도시 망
Fig. 3. Swain-55 network

표 1. 알고리즘 수행 결과
Table 1. Result of algorithm

연구 결과	F_A 점포 위치	F_B 점포 위치
Serra와 ReVelle[1] PREMAL 알고리즘	{6,8,13,16,31} (1,648)	{4,13,17,25,45} (1,927)
Choi et al.[4] 분할정복 알고리즘(DCA)	{6,8,13,16,31} (1,648)	{4,13,17,25,45} (1,927)
	{6,8,13,16,31} (1,718)	{4,13,17,41,45} (1,857)
Lee et al.[5] 경쟁 알고리즘(CA)	{4,21,22,36,38} (1,344)	{11,17,27,30,42} or {5,17,27,30,42} (2,231)

만약, n개 지점들 중에서 k개를 전수탐색(brute-force, exhaustive search)할 경우 가능한 경우 수는 ${}_nC_k = \frac{n!}{k!(n-k)!}$가 된다. Lotto의 1등에 당첨될 확률은 $n=45, k=6$으로 ${}_{45}C_6 = \frac{45!}{6! \times 39!}$=8,145,060이 된다. 본 논문에서 예를 들은 그림 1의 $n=55, k=5$인 경우, ${}_{55}C_5 = \frac{55!}{5! \times 50!}$=3,478,761회를 수행하여야만 한다. 이는 컴퓨터 프로그램 도움 없이는 사실상 해를 얻는데 불가능하다.

PRIMAL[1]이나 DCA[4] 기법 모두 이러한 가능한 경우 수를 가급적 줄이려고 시도하였으나 현저하게 줄이는 데는 실패하였다.

Ⅲ. 비 근방 탐색 알고리즘

본 장에서는 경쟁업체 점포의 고객을 최대로, 신규개설 점포의 고객을 최소로 빼앗는 위치를 결정하는 방법을 제안한다. 이러한 목적을 달성하기 위해서는 경쟁업체 점포와 인접하고 있으면 최대 고객을 빼앗을 수 없기 때문에 인접하지 않아야 하며, 신규 개설 업체의 점포들 간에도 인접하면 안된다. 이러한 개념에 기반하여 본 장에서 제안되는 알고리즘을 비 근방 탐색(non-neighborhood search, NNS) 알고리즘이라 한다. NNS는 n과 주민 수가 변경되는 어떠한 경우라도 쉽게 적용하여 빠르게 해를 찾을 수 있도록 유연한 확장성도 갖도록 한다.

NNS 알고리즘은 먼저, 직관적으로 볼 때 경쟁업체 k개 점포는 가능한 중심가를 기준으로 원형으로 배치되어 있으며, 이들 지점의 고객 수를 가급적 많이 빼앗을 수 있는 지점으로 최 외곽 지점들이 될 수 없다. 따라서 Step 1에서는 이들 m개 지점 을 제외하여 주어진 문제의 망의 영역 수 n을 $n-m$개로 축소시킨다.

Step 1. 주어진 망에 대해 $v \not\in N_G(u), u \not\in F_A$를 제외한 최 외곽 지점들 u와 u의 이웃인 $N_G(u)$들 중에서 $\deg(u) \le 4$인 지점 m개를 을 제외시킨다. 이와 같이 하여 n개 지점을 $n-m$지점으로 축소시킨다.

다음으로, 경쟁업체 지점 u를 포함하여, u와 $N_G(u)$지점들에 점포를 개설할 경우 경쟁 업체 지점의 고객을 최대로 빼앗을 수 없다는 개념에 근거하여, Step 2에서는 $N_G[u]$를 제외한 나머지 지점들 $v \not\in N_G[u]$에 대해 경쟁업체 고객을 최대로 빼앗을 수 있는 확보 고객 수를 구한다. 각 경로에서 최대 고객 확보 지점들을 신규 개설 점포로 결정한다.

Step 2. $N_G[u]$를 제외한 나머지 지점들 $v \not\in N_G[u]$ 각각에 대해 경쟁업체 k개의 F_A 고객을 최

대로 빼앗을 수 있는 확보 고객 수를 구한다. 이로부터 각 경로에서 최대 고객 확보 지점들 l개를 F_B의 신규 개설 점포로 결정한다.

만약, $k=l$이면 F_A의 각 지점의 고객 수/2와 F_B 지점의 고객 수를 비교하여 F_B 지점의 고객 수가 보다 적은 경우 이를 보다 많은 F_A 지점으로 변경시킨다. 만약, $k>l$이면 $q=k-l$개 지점을 Step 3과 같이 신규 개설 업체 점포들과 이웃하지 않은 경쟁업체 점포들 영역에서 결정한다.

Step 3. k개의 F_A와 l개의 F_B 가 존재하는 경우에 대해 $u \notin N_G[v]$인 지점을 대상으로 각 지점 u의 확보 고객 수를 계산한다. 이들 중 최대 고객 확보 지점 q개를 결정한다.

Step 4. k개의 F_B가 함께 개설시 총 확보 고객 수에서 각 노드를 제외시 확보 고객 수를 뺀 값(기여도)을 계산하여, 최소 기여도를 갖는 노드에 대해, 나머지 F_B의 노드 u의 $v \notin N_G[u]$들 중 보다 많은 고객을 확보할 수 있는 노드로 변경한다.

위치	주민수	F_A					F_B					F_A 확보 고객 수					F_B 확보 고객 수				
		6	8	13	16	31	13	17	25	4	45	6	8	13	16	31	13	17	25	4	45
1	120	10.61	4.47	2.24	21.40	13.71	2.24	24.81	19.59	5.00	11.08	0	0	120	0	0	120	0	0	0	0
2	114	8.61	3.16	5.40	20.09	10.68	5.40	21.65	16.29	3.00	9.77	0	0	0	0	0	0	0	0	114	0
3	110	12.40	3.16	9.87	15.71	6.08	9.87	17.18	17.35	7.47	8.39	0	110	0	0	0	0	0	0	0	0
4	108	5.61	6.16	5.24	23.09	10.09	5.24	24.65	13.97	1.00	12.77	0	0	0	0	0	0	0	0	108	0
5	105	8.61	6.47	2.24	25.34	13.09	2.24	26.81	16.97	3.00	13.08	0	0	105	0	0	105	0	0	0	0
6	103	1.00	11.09	9.97	28.11	12.83	9.97	26.30	8.36	5.61	17.70	103	0	0	0	0	0	0	0	0	0
7	100	9.34	7.40	12.47	19.95	3.61	12.47	17.08	13.11	6.48	12.63	0	0	0	0	100	0	0	0	0	0
8	94	11.09	1.00	6.71	16.93	9.24	6.71	20.34	19.45	6.16	6.61	0	94	0	0	0	0	0	0	0	0
9	91	3.61	8.16	6.36	25.18	11.46	6.36	24.93	11.97	2.00	14.77	0	0	0	0	0	0	0	0	91	0
10	90	5.00	14.16	12.36	31.18	17.46	12.36	30.93	10.20	8.00	20.77	90	0	0	0	0	0	0	0	0	0
11	88	7.73	7.88	2.24	26.75	14.50	2.24	28.22	16.09	4.41	14.49	0	0	88	0	0	88	0	0	0	0
12	87	32.62	25.00	31.71	15.31	19.79	31.71	6.32	34.07	30.72	21.84	0	0	0	0	0	0	87	0	0	0
13	87	9.97	6.71	1.00	23.64	16.08	1.00	27.05	18.33	5.24	13.32	0	0	87	0	0	87	0	0	0	0
14	85	38.22	27.23	33.94	10.30	25.39	33.94	16.95	39.67	32.75	20.62	0	0	0	85	0	0	0	0	0	0
15	83	5.83	12.31	13.80	25.78	7.00	13.80	20.47	7.28	8.56	18.46	83	0	0	0	0	0	0	0	0	0
16	82	28.11	16.93	23.64	1.00	20.91	23.64	11.81	33.06	23.09	10.32	0	0	0	82	0	0	0	0	0	0
17	80	26.30	20.34	27.05	11.81	17.08	27.05	1.00	28.30	24.65	18.34	0	0	0	0	0	0	80	0	0	0
18	79	12.23	13.71	20.42	23.96	4.47	20.42	14.39	13.68	14.56	18.94	0	0	0	0	79	0	0	0	0	0
19	79	17.93	9.56	16.27	14.96	5.10	16.27	10.78	19.38	13.87	10.10	0	0	0	0	79	0	0	0	0	0
20	77	12.49	22.22	20.42	39.24	21.57	20.42	35.04	9.22	16.06	28.83	0	0	0	0	0	0	0	77	0	0
21	76	14.00	23.16	21.36	40.18	26.46	21.36	39.51	13.69	17.00	29.77	0	0	0	0	0	0	0	76	0	0
22	74	26.18	16.40	23.11	6.71	13.35	23.11	5.10	27.63	20.71	13.24	0	0	0	0	0	0	74	0	0	0
23	72	20.91	14.95	21.66	17.20	8.08	21.66	5.39	22.91	18.19	15.49	0	0	0	0	0	0	72	0	0	0
24	70	21.60	27.99	29.57	36.62	18.75	29.57	24.81	14.72	24.33	33.22	0	0	0	0	0	0	0	70	0	0
25	69	8.36	19.45	18.33	33.06	14.28	18.33	28.30	1.00	13.97	25.74	0	0	0	0	0	0	0	69	0	0
26	69	19.30	20.78	27.49	29.41	11.54	27.49	17.60	14.26	21.63	26.01	0	0	0	0	69	0	0	0	0	0
27	64	34.13	23.04	27.58	6.40	26.46	27.58	18.21	40.74	29.58	16.43	0	0	0	64	0	0	0	0	0	0
28	63	27.98	22.02	28.73	18.81	15.15	28.73	7.00	29.75	25.26	22.56	0	0	0	0	0	0	63	0	0	0
29	62	16.44	12.85	19.88	23.96	3.61	19.88	9.86	18.97	13.70	13.71	0	0	0	0	62	0	0	0	0	0
30	61	17.40	8.16	14.87	10.71	10.20	14.87	13.34	22.35	12.47	5.00	0	0	0	0	0	0	0	0	0	61
31	60	12.83	9.24	16.08	20.91	1.00	16.08	17.08	14.28	10.09	14.47	0	0	0	0	60	0	0	0	0	0
32	58	21.70	10.61	17.32	6.32	18.47	17.32	14.34	29.74	16.77	4.00	0	0	0	0	0	0	0	0	0	58
33	57	21.40	12.16	18.87	6.71	14.20	18.87	9.34	26.35	16.47	9.00	0	0	0	57	0	0	0	0	0	0
34	55	14.70	3.61	10.32	13.32	11.47	10.32	21.34	22.74	9.77	3.00	0	0	0	0	0	0	0	0	0	55
35	54	20.14	26.62	28.11	41.09	20.32	28.11	29.28	12.09	22.87	32.77	0	0	0	0	0	0	0	54	0	0
36	53	12.54	19.02	20.51	32.21	12.72	20.51	22.64	5.66	15.27	25.17	0	0	0	0	0	0	0	53	0	0
37	51	12.81	21.23	14.52	38.47	25.27	14.52	38.74	18.01	15.81	28.15	51	0	0	0	0	0	0	0	0	0
38	49	24.91	13.82	18.36	7.21	21.68	18.36	19.02	32.95	19.98	7.21	0	0	0	24.5	0	0	0	0	0	24.5
39	47	34.93	24.54	24.96	19.69	32.40	24.96	31.50	43.67	30.20	19.69	0	0	0	23.5	0	0	0	0	0	23.5
40	44	28.95	22.94	18.98	23.27	30.80	18.98	35.08	42.07	24.22	19.57	0	0	44	0	0	44	0	0	0	0
41	43	4.24	15.33	14.21	32.10	13.32	14.21	26.79	4.12	9.85	21.94	0	0	0	0	0	0	0	43	0	0
42	42	5.10	5.99	7.48	22.92	7.85	7.48	21.32	13.46	2.24	12.60	0	0	0	0	0	0	0	0	42	0
43	41	20.08	9.69	10.11	15.46	17.55	10.11	24.66	28.82	15.35	6.32	0	0	0	0	0	0	0	0	0	41
44	40	12.80	8.59	2.83	22.74	17.83	2.83	29.37	21.16	8.07	13.60	0	0	40	0	0	40	0	0	0	0
45	39	17.70	6.61	13.32	10.32	14.47	13.32	18.34	25.74	12.77	1.00	0	0	0	0	0	0	0	0	0	39
46	37	20.01	14.69	10.04	20.46	22.55	10.04	29.66	28.37	15.28	11.32	0	0	37	0	0	37	0	0	0	0
47	35	11.34	11.18	4.47	28.42	18.11	4.47	31.52	18.80	8.02	17.79	0	0	35	0	0	35	0	0	0	0
48	34	12.83	23.92	22.80	37.53	18.72	22.80	28.64	4.47	18.44	30.21	0	0	0	0	0	0	0	34	0	0
49	33	15.36	26.45	25.33	40.06	21.28	25.33	34.75	7.00	20.97	32.65	0	0	0	0	0	0	0	33	0	0
50	33	22.83	22.67	15.96	39.91	29.60	15.96	43.01	28.45	19.51	29.28	0	0	33	0	0	33	0	0	0	0
51	32	21.28	30.44	28.64	48.46	30.79	28.64	44.26	18.44	24.28	37.05	0	0	0	0	0	0	0	32	0	0
52	26	27.46	27.30	20.59	36.08	34.23	20.59	47.64	34.92	24.14	26.94	0	0	26	0	0	26	0	0	0	0
53	25	17.00	16.84	10.13	34.08	23.77	10.13	37.18	24.46	13.68	23.45	0	0	25	0	0	25	0	0	0	0
54	24	31.31	20.22	24.76	13.61	29.34	24.76	25.42	39.35	26.38	13.61	0	0	0	12	0	0	0	0	0	12
55	21	25.08	14.69	15.11	15.02	22.55	15.11	26.83	33.82	20.35	11.32	0	0	0	0	0	0	0	0	0	21
	3575											327	204	320	348	449	320	376	541	355	335
												1648					1927				

그림 4. NNS의 MS-액셀 검증
Fig. 4. MS-Excel verification for NNS

NNS는 Step 1에서 ${}_nC_k$를 ${}_{n-m}C_k$로 축소시켰다. Step 2에서는 ${}_{n-m}C_k$를 수행하지 않고 단지 $n-m-p$, $p=N_G[u], u \in F_A$에 대해 수행하였으며, Step 3에서는 $n-m-q$, $q=N_G[v], v \in F_B$에 대해 수행하였다. 결국 제안된 알고리즘은 $(n-m-p)+(n-m-q)=2(n-m)-(p+q)$회 수행하는 $O(n)$선형 복잡도를 갖는 단순한 알고리즘이며, Step 2 ~ Step 4는 컴퓨터 프로그램을 작성하지 않고 단지 그림 4와 같이 MS-Excel을 활용하여 누구나 쉽게 구할 수 있다.

Ⅳ. 알고리즘 적용성 평가

본 장에서는 그림 3의 Swain-55 망에서 첫 번째로, F_A ={6,8,13,16,31}인 경우 F_B를 NNS로 구하여 PREMAL과 DCA로 구한 표 1의 결과와 비교한다. 두 번째로, F_A ={4,21,22,36,38}인 경우 F_B를 NNS로 구하여 Lee et al.[5]의 CA와 비교한다.

F_A ={6,8,13,16,31}에 대해 NNS의 Step 1을 수행한 결과는 그림 5와 같다. 55개 노드 중 최 외곽 노드인 {51,37,50,52,40,39,54,27,14,12,28,26,24,35,48,49}가 제외되고, 이들 노드의 인접 노드이면서 $\deg(u) \le 4$인 {21, 53}이 제외되었다. 따라서 남은 노드 수는 $n-m=55-18=37$개 이다.

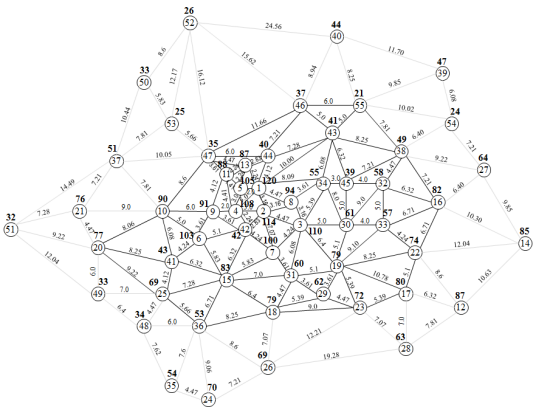
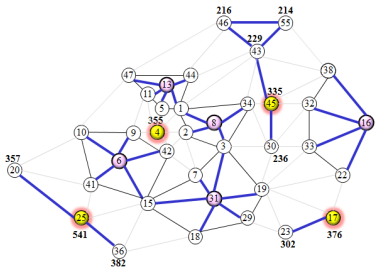


그림 5. 외곽 노드 제외
Fig. 5. Exclusion of Outer nodes

NNS의 Step 2를 수행하여 1차로 선정한 F_B의 신규 개설 점포 위치는 그림 6과 같다. 여기서는 대상 노드를 {20,25,36}, {4}, {46,55,43,45,30}와 {23,17}의 4개경로를 설정하고, 각 노드에 신규점포를 개설할 경우 F_A = {6,8,13,16,31}의 고객을 빼앗는 고객 수를 계산하였다. 이들로부터 각 경로에서 최대 고객 확보 지점인 {4,17,25,45}가 1차적으로 결정된 F_B가 된다.

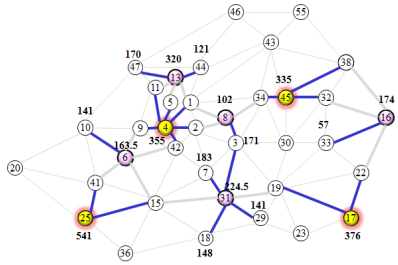
- max{20(357),25(514),36(382)} = 25(541)
- max{4(355)} = 4(355)
- max{17(376),23(302)} = 17(376)
- max{46(216),55(214),43(229),45(335),30(236)}=45(335)

k개의 F_B를 얻는 것이 목적이므로, 부족한 한 곳은 Step 3을 수행하여 얻었으며, 이는 그림 7에 제시되어 있다.



F_A={6,8,13,16,31}, F_B={α}
대상 : {20,25,36}, {4}, {46,55,43,45,30}, {23,17}
max{20(357),25(514),36(382)}=25(541)
max{4(355)}=4(355)
max{17(376),23(302)}=17(376)
max{46(216),55(214),43(229),45(335),30(236)}=45(335)
⇒ F_B={4,17,25,45}

그림 6. 대상 점포 선정
Fig. 6. Choice of candidate stores



F_A={6,8,13,16,31}, F_B={4,17,25,45}+{β}
대상 : {6,10,47,13,44,8,3,7,31,18,29, 16,33}
⇒ max{6(163.5), 10(141), 47(170),13(320),44(121),8(102),3(171),7(183), 31(224.5),18(148),29(141),16(174),33(57)}=13(320)

F_A= **1648{6,8,13,16,31}**: F_B =**1927{4,13,17,25,45}**
6(327), 8(204), 13(320), 16(348), 31(449)
4(355), 17(376), 25(541), 45(335), 13(320)

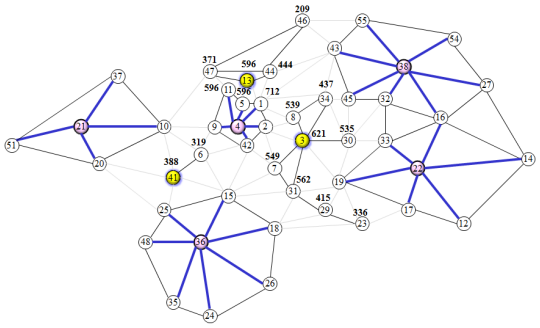
그림 7. 대상점포 추가 선정
Fig. 7. Additional choice of candidate stores

그림 7에서, 나머지 한 곳을 선정하는 대상은 {6,10,47,13, 44,8,3,7,31,18,29,16,33}으로 결정되었으며, F_A ={6,8,13,16,31}과 F_B ={4,17,25,45}가 존재하는 상태에서 각 노드의 확보 고객 수를 계산하여 최대 확보 고객 지점 위치는 13(320)으로 결정되었다.

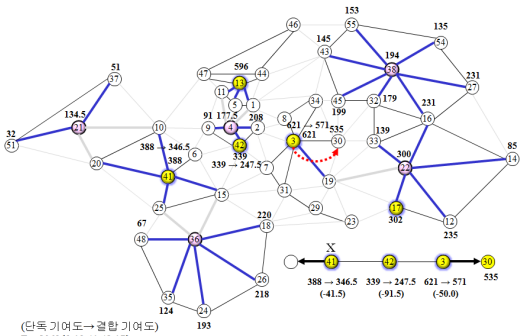
- max{6(163.5),10(141),47(170),13(320),44(121),8(102),3(171),7(183),31(224.5),18(148),29(141),16(174),33(57)}=13(320)

제안된 알고리즘은 $n=55$에서 $k=5$개를 선정하는 문제에 대해 ${}_{55}C_5$ =3,478,761회를 수행하지 않고, 단지 {20,25,36}, {4}, {46,55,43,45,30}와 {23,17}, {6,10,47,13,44,8,3,7,31,18,29,16,33}의 24개 노드에 대해서만 엑셀을 이용하여 해를 구하는데 성공하였다.

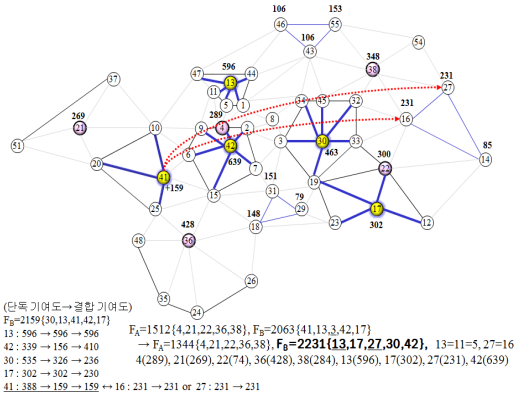
다음으로, F_A ={4,21,22,36,38}에 대해 NNS를 수행한 결과는 그림 8에 제시하였다.



(단독 기여도)
max{13(596),47(371),44(444),46(209)}=13(596)
max{3(621),31(562), 7(549),8(539),30(535),34(437),29(415),23(326)}=3(621)
max{41(388),6(319)}=41(388)
F_A=1970{4,21,22,36,38}, F_B=1605{41,13,3}



(단독 기여도→결합 기여도)
F_B=2063{3,13,41,42,17}
13 : 596 → 596
17 : 302 → 302
41 : 388 → 159
42 : 339 → 156
3 : 621 → 140 ↔ 30 : 535 → 326
F_A=1512{4,21,22,36,38}, F_B=2063{41,13,3,42,17}
→ F_A=1575{4,21,22,36,38}, F_B=2000{41,13,30,42,17}



(단독 기여도→결합 기여도)
F_B=2159{30,13,41,42,17}
13 : 596 → 596 → 596
42 : 339 → 156 → 410
30 : 535 → 326 → 236
17 : 302 → 302 → 230
41 : 388 → 159 → 159 ↔ 16 : 231 → 231 or 27 : 231 → 231
F_A=1512{4,21,22,36,38}, F_B=2063{41,13,3,42,17}
→ F_A=1344{4,21,22,36,38}, **F_B=2231{13,17,27,30,42}**, 13=11=5, 27=16
4(289), 21(269), 22(74), 36(428), 38(284), 13(596), 17(302), 27(231), 42(639)

그림 8. F_A ={4,21,22,36,38}에 대한 NNS
Fig. 8. NNS for F_A ={4,21,22,36,38}

단독으로 결정하는 경우, Step 2에서 F_B ={41,13,3}을 얻고, Step 3에서 추가로 {42,17}을 얻어 F_B ={3,13,17,41,42}가 되었다. Step 4에서는 F_B ={3,13,17,41,42}의 5개 노드가 결합된 경우 각 노드의 기여도(총 확보 고객 수 - 각 노드 제외시 확보 고객 수)를 계산한 결과 3(140)의 최소 기여도를 가져 이를 30(326)으로 대체하였다. 다시 F_B ={13,17,30,41, 42}에 대해 각 노드의 기여도를 계산한 결과 41(159)로 최소 기여도를 나타냈으며, 이 노드를 16(231) 또는 27(231)로 대체하였다. 결론적으로 F_B ={13,16,17,30,42} 또는 F_B ={13,17,27,30,42}을 얻었으며, 여기서 30은 5 또는 11과 동일한 기여도를 가져, 6가지 경우 수에 대한 최적 해를 구할 수 있었다.

V. 결론

본 논문에서는 아직까지도 다항시간으로 해를 구하는 규칙을 발견하지 못해 NP-난제로 알려진 경쟁업체가 k개 점포로 시장을 독점하고 있는 상황에서, 동일한 점포 수 k개를 신규로 개설하여 경쟁업체 고객 수보다 많은 고객을 확보할 수 있는 점포 위치를 결정하는 문제를 다루었다.

지금까지 알려진 대부분의 방법들은 넣고 빼기를 수많은 반복을 수행하여 해를 구하는 방법을 적용하였다. 반면에, 본 논문에서 제안된 알고리즘은 주어진 망의 n개 지점(노드)들 중에서 신설 점포위치로 될 수 없는 최외곽 노드들을 제외시키고, 나머지 노드들을 대상으로 경쟁업체 점포와 인접하지 않은 영역에서 먼저 신설 점

포 후보들을 선정하고, 다음으로, 신설 점포 후보들과 인접하지 않은 노드들에서 나머지 점포들을 선정하였다. 이렇게 선정된 점포들을 모두 운영할 경우 각 점포의 실제적인 확보 고객 수를 구하여 최소 고객을 가진 점포를 보다 많은 확보 고객을 가진 위치로 변경하는 방법을 적용하였다. 이런 방법을 적용한 결과 제안된 알고리즘은 $O(n)$ 수행 복잡도로 컴퓨터 프로그램의 도움없이 단지 MS- Excel을 활용하여 해를 구하는데 성공하였다.

제안된 알고리즘을 2가지 경우의 사례 데이터에 대해 적용한 결과, 기존의 PREMAL, DCA, CA와 동일한 결과인 최적 해를 얻었으며, 특정 데이터에 대해서는 다양한 경우 수에 대한 여러 가지 해를 구하기도 하였다.

References

[1] 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.

[2] C. ReVelle, "Facility Siting and Integer-Friendly Programming", European Journal of Operational Research, Vol. 65, pp. 147-158, 1993.

[3] V. Beresnev, "Branch-and-Bound Algorithm for a Competitive Facility Location Problem," Computers & Operations Research, Vol. 40, No. 8, pp. 2062-2070, Aug. 2013, doi:/10.1016/j.cor.2013.02.023

[4] S. B. Choi, B. G. Kim, and T. Y. Han, "Sports Shop Location Strategy to Maximize the Market Share Using Divide-and-Conquer Strategy," Journal of the Korean Society of Sports Science, Vol. 23, No. 2, pp. 309-320, Apr. 2014, uci: G704- 001369.2014.23.2.018

[5] S. U. Lee, Y. S. Lee, S. B. Choi, and T. Y. Han, "Location Strategy of Sports Outlets to Maximize the Market Share," Journal of the Institute of Internet, Broadcasting and Communication, Vol. 13, No. 3, pp. 93-101, Jun. 2013, doi:0.7236/ JIIBC. 2013.13.3.93

[6] S. U. Lee, "A Constructive Algorithm for p-Median Facility Location," Journal of The Korea Society of Computer and Information, Vol. 20, No. 6, pp. 77-85, Jun. 2015, doi:10.9708/ jksci.2015.20.6.077

저자 소개

이 상 운(정회원)

- 1987년 : 한국항공대학교 항공전자공학과 (학사)
- 1997년 : 경상대학교 컴퓨터과학과 (석사)
- 2001년 : 경상대학교 컴퓨터과학과 (박사)
- 2003년 : 강원도립대학 컴퓨터응용과 전임강사
- 2004년 ~ 2007.2 : 국립 원주대학 여성교양과 조교수
- 2007.3 ~ 2015.3 : 강릉원주대학교 멀티미디어공학과 부교수
- 2015.4 ~ 현재 : 강릉원주대학교 멀티미디어공학과 정교수
- e-mail : sulee@gwnu.ac.kr

<주관심분야 :소프트웨어 프로젝트 관리, 개발 방법론, 분석과 설계 방법론, 시험 및 품질보증, 소프트웨어 신뢰성, 최적화 알고리즘>