

논문 2011-6-24

응급시설 위치 문제

Emergency Medical Service Location Problem

최명복*, 김봉경**, 한태용***

Myeong-Bok Choi, Bong-Gyung Kim, Tae-Yong Han

요약 본 논문은 하나의 도시가 여러 구역으로 분할되고, 응급환자가 발생하였을 때, 모든 구역에 대해 최대 허용 도착시간 T 를 충족시키도록 응급시설을 배치하는 문제에 대한 알고리즘을 제안하였다. 이 문제는 일반적으로 다항시간으로 해를 구하는 알고리즘이 존재하지 않아 두 구역 간 소요시간이 최대허용 도착시간이내이면 1로, 그렇지 않으면 0으로 하는 정수계획법으로 변환시키고, 선형계획법 도구를 활용하여 해를 구한다. 본 논문은 최소차수 노드의 이웃 노드들 중 최대 차수 노드를 응급시설의 위치로 결정하는 집합피복 알고리즘을 적용하였다. 제안된 알고리즘을 텍사스 오스틴 시의 33개 구역에 대한 사례에 대해 $3 \leq T \leq 20$ (분)을 적용하고, Swain의 55개 노드 망에 대해 $T=15$ 에 대해 응급시설의 위치를 결정할 수 있는지 여부를 검증하였다. 선형계획법을 활용한 전통적인 집합피복 알고리즘은 몇 개의 T 에 대해 해를 구하지 못한 반면에, 제안된 알고리즘은 18개의 모든 T 에 대해 해를 구하였다.

Abstract This paper suggests emergency medical service vehicle (ambulance) algorithm when the emergency patient occurs in order to be sufficient the maximum permission time T of arrival about all sectors in one city that is divided in the various areas. This problem cannot be solved in polynomial times. One can obtain the solution using the integer programming. In this paper we suggest vertex set (or dominating set) algorithm and easily decide the location of ambulances. The core of the algorithm decides the location of ambulance is to the maximum degree vertex among the neighborhood of minimum degree vertex. For the 33 sectors Ostin city in Texas, we apply $3 \leq T \leq 20$ minutes. The traditional set cover algorithm with integer programming cannot obtain the solution in several T in 18 cases. But, this algorithm obtains solution for all of the 18 cases.

Key Words : 집합피복, 지배집합, 응급시설, 차수, 이웃

1. 서론

어떤 한 도시 (광역, 또는 지방)에 설치되는 경찰서, 소방서와 응급의료시설과 같은 응급구조 시설은 사회적, 경제적과 정책적 목적을 만족시켜야만 한다. 특히, 한 도시에 n 개의 구역 N_i , ($i=1,2,\dots,n$)이 존재하고, 두 구역 N_i 와 N_j 간 거리 d_{ij} 를 알고 있다고 하자. 만약, 응급

상황이 발생하였을 때 임의의 응급구조시설 S_j 에서 응급환자수송차량 (응급시설, ambulance)가 임의의 구역 N_i 까지 허용되는 최대 도착시간 T 는 준수하여야 주민의 생명을 지킬 수 있다고 가정하자. 이 경우, 이 도시에 응급시설을 최소로 몇 대 배치하는 것이 가장 효율적인가를 판단해야만 한다.^[1]

본 논문은 한 도시에 응급시설을 몇 대 배치하는 것이 가장 효율적인가에 대한 연구에 한정한다. 이 문제의 해를 구하는 방법은 선형계획법 (Linear programming, LP), 집합피복 (Set covering), 중앙값 방법 (Median

*중신회원, 강릉원주대학교 멀티미디어공학과

**정회원, 남서울대학교, 스포츠경영학과

***정회원, 강릉원주대학교, 여성인력개발학과

접수일자 2011.10.14, 수정완료 2011.11.20

게재확정일자 2011.12.16

method), 중심점 방법 (Center method)이 있다.^[2] 중앙값과 중심점 방법은 모든 구역간의 최단거리의 최대값이 $t_{ij} \leq T$ 가 되도록 응급시설의 위치를 이동시키는 방법이다. 반면에, 선형계획법과 집합피복은 식 (1)과 같이 정수계획법 (Integer programming, IP)을 적용하여 $t_{ij} \leq T$ 이면 1로, $t_{ij} > T$ 이면 0으로 치환한다.^[1]

$$\text{minimize } Z = \sum_j X_j \tag{1}$$

$$\text{s.t. } a_{ij}X_{ij} \geq 1, \text{ for all } i$$

$$X_j = 0, 1 \text{ for all } j$$

where

$$X_j = \begin{cases} 0, & \text{만약, 구급차가 } j \text{ 지역에 없는 경우} \\ 1, & \text{만약, 구급차가 } j \text{ 지역에 있는 경우} \end{cases}$$

$$a_{ij} = \begin{cases} 0, & t_{ij} > T \\ 1, & t_{ij} \leq T \end{cases}, \quad t_{ij} = \frac{1}{v_{ij}} \times d_{ij}$$

Z: 응급시설 대수

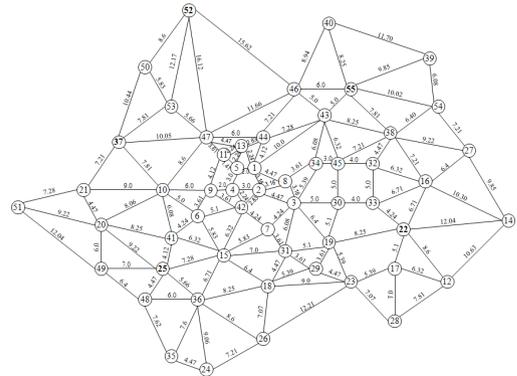
주어진 거리 행렬 데이터 D의 거리 (또는 시간)를 a_{ij} 로 치환한 행렬 데이터 A에 대해 기존 알고리즘은 선형계획법 (Linear programming, LP), 집합피복 (Set covering), 중앙값 방법 (Median method), 중심점 방법 (Center method)으로 해를 구하였다. 집합피복 문제의 정확한 해 (exact solution)를 다항시간에 구하는 알고리즘은 아직 알려지지 않아 NP-완전 (Non-deterministic Polynomial complete)에 속하는 문제이다.^[3-6] NP-완전 문제는 미국 클레이 수학재단에서 100만불의 상금을 제시한 21세기에 풀어야 할 7개 문제 중 첫 번째인 “P=NP?”이다.^[7] 반면에, 선형계획법은 단순히 선형계획법 패키지 (도구)를 적용하여 해를 구하는 방법으로 최적화 문제를 푸는데 일반적으로 사용되고 있다.^[8]

본 논문은 최소 응급시설 대수를 구하기 위해 최소 집합피복의 근사 해 (Approximate solution)를 찾는 욕심쟁이 알고리즘 (Greedy algorithm)을 제안한다. 2장에서는 관련 연구를 고찰한다. 3장에서는 최소 응급시설 대수를 $O(n^2)$ 의 다항시간으로 구하는 알고리즘을 제안한다.

II. 관련연구와 문제점

그림 1은 Swain이 제시한 55개 노드에 대한 데이터이다.^[9,10] 여기서 각 노드의 위치는 직각좌표로 제시되어 있

어 인접한 두 노드 v_i 와 v_j 간의 거리는 유클리드 거리 공식 $l(v_i, v_j) = \sqrt{[v_i(x) - v_j(x)]^2 + [v_i(y) - v_j(y)]^2}$ 으로 구할 수 있다. 그림 1의 데이터에 대해 Church와 ReVelle^[9]는 욕심쟁이 알고리즘 (Greedy algorithm)과 선형계획법 (Linear programming)을 사용하여 $T=15$ 와 $T=10$ 에 대해 인구수를 최대한 커버할 수 있도록 응급시설 위치를 그림 2와 같이 결정하였다. Church와 ReVelle^[9]의 그림 2에 따르면 $T=15$ 인 경우 52, 37, 55, 25, 22의 5곳에 응급시설을 위치하는 것으로 결정하였다. 여기서, ① 노드를 25 노드에 인접한 것으로 결정하였으나 $(25, 15, 42, 2) = 16.43$, $(25, 41, 6, 9, 4, 2) = 16.97$, $(25, 41, 6, 42, 2) = 16.29$ 로 $T=15$ 를 초과한다. 또한, ② 노드는 37, 52, 55, 22 노드로 부터도 $T=15$ 를 초과하여 어느 노드에서도 커버될 수 없어 6개의 응급시설 위치를 결정하여야 하나 5개의 위치만 결정하였다.



노드	요구량 (인구수)	위치좌표		노드	요구량 (인구수)	위치좌표	
		x	y			x	y
1	120	32	31	29	62	19	38
2	114	29	32	30	61	27	41
3	110	27	36	31	60	21	35
4	108	29	29	32	58	32	45
5	105	32	29	33	57	27	45
6	103	26	25	34	55	32	38
7	100	24	33	35	54	8	22
8	94	30	35	36	53	15	25
9	91	29	27	37	51	35	16
10	90	29	21	38	49	36	47
11	88	33	28	39	47	46	51
12	87	17	53	40	44	50	40
13	87	34	30	41	43	23	22
14	85	25	60	42	42	27	30
15	83	21	28	43	41	38	39
16	82	30	51	44	40	36	32
17	80	19	47	45	39	32	41
18	79	17	33	46	37	42	36
19	79	22	40	47	35	36	26
20	77	25	14	48	34	15	19
21	76	29	12	49	33	19	14
22	74	24	48	50	33	45	19
23	72	17	42	51	32	27	5
24	70	6	26	52	26	52	24
25	69	19	21	53	25	40	22
26	69	10	32	54	24	40	52
27	64	34	56	55	21	42	42
28	63	12	47	계	3,575		

그림 1. Swain의 55-노드 망
Fig. 1. The Swain's 55-Node Network

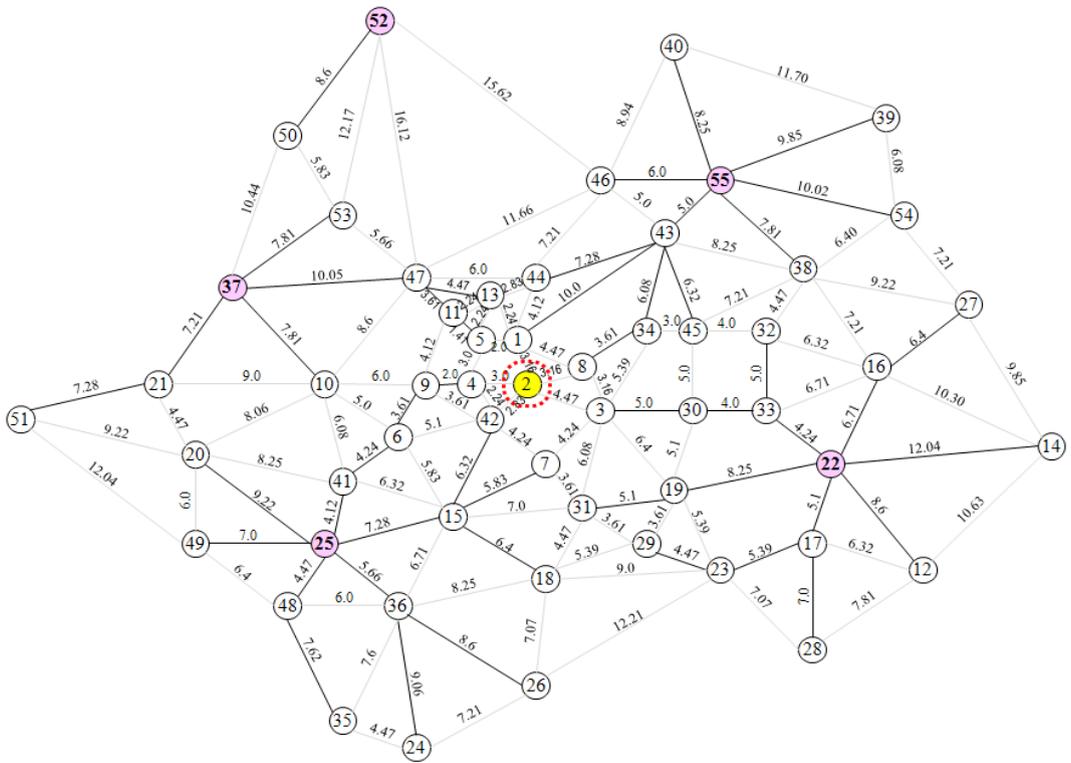


그림 2. $T=15$ 에 대한 Swain의 55-노드 망 응급시설 위치

Fig. 2. Emergency Facility Location for Swain's 55-Node Network with $T=15$

표 1은 Daskin과 Stern^[1]에서 인용된 데이터로, 미국 텍사스주 오스틴시를 국세조사표준구역 (census tracts)에 의해 33개의 구역 (Zones)으로 분할하여 각 구역 간 소요시간 (분)을 나타내고 있다. 여기서 소요시간은 응답시간이라 하며, 구역내의 소요시간은 1분으로 설정하였다.

Daskin과 Stern은 최대 허용 응답시간 T 를 3분에서 20분까지 1분 단위로 변화시키면서 응급시설 수와 위치를 선형계획법과 유사한 집합피복 (또는 지배집합) 방법으로 구하였다. 이때 지배된 집합을 포함하는 CSC (Conventional Set Covering)와 지배된 집합을 배제시키는 HOSC (Hierarchical Objective Set Covering)방법으로 하여 해를 구하였다. 이 방법을 적용하기 위해서는 응급시설 수를 최소화시키는 목적함수값 W 을 적절히 설정하는 문제도 제기된다.

표 1의 데이터에 대해 Daskin과 Stern^[1]이 $3 \leq T \leq 20$ 각각에 대해 CSC와 HOSC로 응급시설의

위치를 구한 결과는 표 2에 제시되어 있다. CSC 방법은 $T=9,10,14$ 의 3가지 경우에 대해 응급시설의 위치를 구하지 못한 반면에, HOSC는 $T=10$ 의 1가지 경우에서 응급시설의 위치를 구하지 못하였다. 두 방법 모두 $T=10$ 에서 응급시설 배치 장소 수는 정수값 대신 실수값 3.75대를 구하여 현실성이 없음을 알 수 있다.

표 2에서 선형계획법 또는 선형계획법과 유사한 집합 피복 방법은 목적함수 (응급시설수)를 어떻게 설정하느냐에 따라 최적 해를 얻을 수도 있고 얻지 못할 수도 있음을 알 수 있다. 즉, 어떠한 법칙을 따르는 알고리즘을 제안하지 않고 단지 목적함수를 설정하고 목적함수를 충족시키도록 휴리스틱한 방법으로 해를 구함으로써 특정 T 에 대해서는 해를 구하지 못하였음을 알 수 있다.

응급시설 위치 문제

표 1. 텍사스 오스틴시의 구역 간 소요시간

Table 1. Travel time for 33 Zones of Texas, Austin

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	총 시간		
1	1	5	8	11	8	8	9	12	13	13	9	8	11	12	13	6	9	10	6	7	9	16	14	16	11	10	15	13	16	18	15	19	22	373		
2	5	1	4	7	4	4	6	9	11	10	7	7	9	10	11	7	7	5	6	11	15	12	13	10	9	14	10	12	14	13	16	19	305			
3	8	4	1	6	5	5	6	7	9	9	7	11	12	12	10	8	8	6	9	10	13	19	11	10	13	16	7	10	12	12	14	18	318			
4	11	7	6	1	6	5	4	3	5	7	5	9	10	10	7	13	12	9	10	9	17	17	13	12	13	12	14	9	7	9	10	10	16	308		
5	8	4	5	6	1	3	5	7	9	9	6	9	10	10	9	9	10	8	7	8	13	17	14	13	12	11	15	9	11	13	11	14	18	314		
6	8	4	5	5	3	1	3	6	8	7	4	6	8	8	10	10	9	6	6	14	16	14	13	13	9	13	10	11	13	10	14	16	291			
7	9	6	6	4	5	3	1	4	6	5	2	5	7	7	6	11	12	10	7	6	14	15	14	13	14	9	11	10	9	12	8	12	14	277		
8	12	9	7	3	7	6	4	1	4	5	4	8	9	9	7	14	13	11	10	9	18	17	14	14	14	12	13	10	7	10	9	9	15	314		
9	13	11	9	5	9	8	6	4	1	4	5	8	9	9	6	15	15	13	11	10	19	16	17	16	16	12	13	12	7	11	8	8	13	339		
10	13	10	9	7	9	7	5	4	1	4	7	7	6	3	15	15	12	11	9	18	13	16	15	16	11	10	12	10	14	5	10	12	321			
11	9	7	7	5	6	4	2	4	5	4	1	4	5	5	5	11	13	11	7	5	15	13	15	14	15	7	9	11	9	13	7	11	13	272		
12	8	7	11	9	9	6	5	8	8	7	4	1	5	7	7	10	13	13	6	4	14	13	18	17	15	6	11	14	13	16	10	14	16	325		
13	11	9	12	10	10	8	7	9	9	7	5	5	1	3	6	13	15	14	9	7	16	10	19	18	17	7	8	15	14	17	8	14	14	347		
14	12	10	12	10	10	8	7	9	9	6	5	7	3	1	5	14	16	15	10	8	17	10	19	18	19	8	7	15	14	17	6	13	12	352		
15	13	11	10	7	9	8	6	7	6	3	5	7	6	5	1	15	16	13	10	9	18	12	17	16	17	10	9	13	11	15	4	10	11	330		
16	6	7	8	13	9	10	11	14	15	11	10	13	14	15	1	3	6	8	9	7	18	11	12	6	12	17	10	17	16	18	21	24	387			
17	9	7	8	12	10	10	12	13	15	15	13	13	15	16	16	3	1	5	10	11	9	21	10	11	6	14	19	9	16	15	18	20	24	406		
18	10	7	6	9	8	9	10	11	13	12	11	13	14	15	13	6	5	1	11	12	12	21	9	10	7	15	20	6	14	12	15	17	21	375		
19	6	5	9	10	7	6	7	10	11	11	7	6	9	10	10	8	10	11	1	4	11	14	16	17	12	7	13	14	15	17	13	17	19	343		
20	7	6	10	9	8	6	6	9	10	9	5	4	7	8	9	9	11	12	4	1	12	12	17	18	13	6	11	15	14	17	11	16	18	330		
21	9	11	13	17	13	14	14	18	19	18	15	14	16	17	18	7	9	12	11	12	1	22	10	14	7	15	20	15	21	20	21	25	27	495		
22	16	15	19	17	17	16	15	17	16	13	13	13	10	10	12	18	21	21	14	12	22	1	26	26	23	12	7	23	21	25	10	19	12	532		
23	14	12	11	13	14	14	14	14	17	16	15	18	19	19	17	11	10	9	16	17	10	26	1	6	6	20	24	10	17	14	19	21	25	489		
24	16	13	10	12	13	13	13	14	16	15	14	17	18	18	16	12	11	10	17	18	14	26	6	1	10	21	23	6	14	9	18	17	24	475		
25	11	10	10	13	12	13	14	14	16	16	15	15	17	19	17	6	6	7	12	13	7	23	6	10	1	16	21	10	17	15	19	21	25	447		
26	10	9	13	12	11	9	9	12	12	11	7	6	7	8	10	12	14	15	7	6	15	12	20	21	16	1	11	18	17	20	12	17	19	399		
27	15	14	16	14	15	13	11	13	13	10	9	11	8	7	9	17	19	20	13	11	20	7	24	23	21	11	1	19	18	22	7	16	12	459		
28	13	10	7	9	9	10	10	10	12	12	11	14	15	15	13	10	9	6	14	15	15	23	10	6	10	18	19	1	11	8	15	15	21	396		
29	16	12	10	7	11	11	9	7	7	10	9	13	14	14	11	17	16	14	15	14	21	21	17	14	17	17	18	11	1	9	13	9	16	421		
30	18	14	12	9	13	13	12	10	11	14	13	16	17	17	15	16	15	12	17	17	20	25	14	9	15	20	22	8	9	1	17	11	19	471		
31	15	13	12	10	11	10	8	9	8	5	7	10	8	6	4	18	18	15	13	11	21	10	19	18	19	12	7	15	13	17	1	11	8	382		
32	19	16	14	10	14	14	12	9	8	10	11	14	13	10	21	20	17	17	16	25	19	21	17	16	25	19	21	17	16	15	9	11	11	1	13	475
33	22	19	18	16	18	16	14	15	13	12	13	16	14	12	11	24	24	21	19	18	27	12	25	24	25	19	12	21	16	19	8	13	1	557		

표 2. 오스틴시 33-노드 망에 대한 CSC와 HOSC 적용 결과

Table 2. Results of CSC and HOSC for Austin's 33-Node Network

최대 도착 허용시간 (분)	CSC		HOSC	
	목적함수값 (응급시설 수)	응급시설 위치	목적함수값 (응급시설 수)	응급시설 위치
3	27(27)	1,2,3,4,5,9,10,11,12,13,16,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33	916(27)	1,2,3,4,6,7,9,10,12,13,16,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33
4	20(20)	1,2,8,13,15,16,18,20,21,22,23,24,25,26,27,28,29,30, 32,33	680(20)	1,2,8,13,15,16,18,20,21,22,23,24,25,26,27,28,29,30,32,33
5	16(16)	2,10,11,17,21,22,23,24,25,26,27,28,29,30,32,33	537(16)	2,10,11,17,21,22,23,24,25,26,27,28,29,30,32,33
6	12(12)	5,10,12,16,21,22,24,27,29,30,32,33	397(12)	7,10,12,16,21,22,24,27,29,30,32,33
7	8(8)	11,16,22,24,29,30,32,33	258(8)	4,11,16,24,27,30,32,33
8	6(6)	9,20,22,25,28,33	187(6)	9,20,25,27,28,31
9	6(6)		165(6)	8,11,16,24,27,31
10	3.75(3.75)		99(3.75)	
11	3(3)	19,28,31	76(3)	2,28,31
12	2(2)	15,18	57(2)	15,18
13	2(2)	11,16	43(2)	3,11
14	2(1)		39(1)	7,11
15	1(1)	7	34(1)	7
16	1(1)	6	34(1)	6
17	1(1)	4	34(1)	4
18	1(1)	4	34(1)	4
19	1(1)	2	34(1)	2
20	1(1)	2	34(1)	2

III. 응급시설 위치 결정 알고리즘

본 장에서 제안하는 방법은 다음과 같다. 첫 번째로, 표 3의 데이터와 같이 구역 간 소요시간 행렬에 대해 $t_{ij} \leq T$ 는 1로, $t_{ij} > T$ 는 0으로 치환하여 0-1 정수계획법을 적용한다. 단 t_{ij} ($i=j$)인 구역내 도착 시간은 0으로 치환한다.

다음으로, 집합 피복을 구하기 위해 적용되는 그래프 관련 용어 정의는 다음과 같다. 임의의 구역을 원으로 하는 노드 (vertex)로, 두 노드 v_i 와 v_j 간 소요시간 t_{ij} 가 $t_{ij} \leq T$ 이면 간선 (edge)으로 연결하고 간선의 가중치 w_{ij} 는 1로 하며, $t_{ij} > T$ 이면 간선이 없고, 가중치는 0이 된다. 각 노드 v_i 에 부속된 간선의 개수를 노드 v_i 의 차수 $d_G(v_i)$ 라 하며, $d_G(v_i) = 0$ 노드를 독립노드 (Independent vertex)라 한다. 그래프의 최소 차수 노드를 $\delta(G)$, 최대 차수 노드를 $\Delta(G)$ 라 하며, 임의의 노드 v_i 와 간선으로 연결된 노드를 인접 (adjacent)한다고 하며, 이웃 (Neighbourhood)이라 한다. 이웃에는 임의의 노드를 포함하는 경우 닫힌 (closed) 이웃 $N_G[v_i]$, 임의의 노드를 포함하지 않는 경우 열린 (opened) 이웃 $N_G(v_i)$ 이라 한다. 만약, 임의의 노드 v_i 를 선택하면 v_i 는 인접한 노드들을 피복 (covering) 또는 지배 (dominate)한다고 한다. 이러한 그래프 개념을 도입하여 제안된 알고리즘은 두 번째로, 0-1 정수계획법 행렬 A 의 각 행에 대해 1의 개수 (차수)의 합을 구한다. 최소차수 노드 $\delta(G) = v$ 의 이웃 $N_G(v) = u$ 들 중에서 최대차수 $\max d_G(u)$ 인 노드 u 를 선택하여 응급시설의 위치로 결정한다. 이렇게 응급시설 위치가 결정되면 $N_G[u]$ 의 모든 간선을 삭제한다. 만약, 이 경우, $d_G(v_i) = 0$ 이 존재하면 이 노드에 다음 응급시설 위치가 결정된다. 그렇지 않으면 모든 간선이 존재하지 않을 때까지 이 과정을 반복 수행한다. 제안된 방법은 그림 3에 제시되어 있다. 즉, 제안된 알고리즘은 선형계획법과 같이 목적함수를 결정하지도 않아도 쉽게 적용할 수 있는 장점을 갖고 있다.

제안된 알고리즘을 Daskin과 Stern^[1]로부터 인용된 그림 4의 2개 데이터에 적용하여 보자. 먼저 6개 노드가 존재할 경우 ③과 ②를 선택하여 응급시설 위치로 쉽게 결정할 수 있었으며, 11개 노드가 존재할 경우 ③과 ⑤를 선택할 수 있었다.

Step 1. 시간 행렬 D 의 t_{ij} 를 $t_{ij} \leq T$ 이면 1로, $t_{ij} > T$ 이면 0으로 치환한 A 행렬을 얻는다. 여기서 T 는 응급시설에서 구급차가 도착해야 하는 최대 허용시간이다. 수행시간: $O(n^2)$

Step 2. A 행렬의 각 행에 대해 1의 합을 구한다. 이는 각 정점에서 T 시간 이내에 도달할 수 있는 노드들로 차수의 합이다. 수행시간: $O(n^2)$

1. 최소 차수 노드 $\delta(G) = v$ 를 선택하여 v 의 이웃 $N_G(v) = u$ 노드들의 차수를 파악한다. 만약, 최소 차수 노드가 여러 개 있으면 모두 파악하여 최대 차수 노드 $\max d_G(u)$ 를 응급시설 위치로 결정한다. 만약, $\max d_G(u)$ 가 동일한 노드가 여러 개 존재하면 임의로 하나를 선택한다.
2. $N_G[u]$ 의 모든 간선 (행과 열)을 삭제한다. 만약, $\max d_G(u)$ 가 동일한 노드가 여러 개 존재하는 경우 $d_G(v_i) = 0$ 인 노드가 발생하면 다른 노드를 선택하여 수행시 $d_G(v_i) = 0$ 노드가 발생하지 않으면 이 노드를 응급시설 위치로 결정한다. 만약, 모든 노드에 대해 $d_G(v_i) = 0$ 가 발생하면 이들 노드의 이웃들 중에서 교집합 노드를 응급시설 위치로 결정한다.

Step 3. 만약, 선택되지 않은 행의 노드가 있으면 Step 2를 반복 수행한다. 그렇지 않으면 알고리즘을 종료한다.

그림 3. 최소 응급시설 위치 결정 알고리즘
Fig. 3. Minimum Emergency Service Location Algorithm (MESVLA)

$$A = \begin{bmatrix} & 1 & 2 & 3 & 4 & 5 & 6 & \text{계} \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 2 \\ 2 & 1 & 0 & 0 & 1 & 0 & 0 & 2 \\ 3 & 1 & 0 & 0 & 0 & 1 & 1 & 3 \\ 4 & 0 & 1 & 0 & 0 & 1 & 0 & 2 \\ 5 & 0 & 0 & 1 & 1 & 0 & 1 & 3 \\ 6 & 0 & 0 & 1 & 0 & 1 & 0 & 2 \end{bmatrix}$$

$$\delta(G) = \{1, 2, 4, 6\}, d_G(1) = 2, d_G(2) = 2, \dots$$

$$N_G(1) = \{2, 3\}, N_G(2) = \{1, 4\}, N_G(4) = \{2, 5\},$$

$$N_G(6) = \{3, 5\}$$

$$d_G(3) = 3, d_G(5) = 3 \rightarrow \textcircled{3} \text{ 선택}$$

$$A = \begin{bmatrix} & 2 & 4 & \text{계} \\ 2 & 0 & 1 & 1 \\ 4 & 1 & 0 & 1 \end{bmatrix}$$

$$\delta(G) = \{2, 4\}, d_G(2) = 1, d_G(4) = 1 \rightarrow \textcircled{2} \text{ 선택}$$

$$A = \begin{bmatrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & \text{계} \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 2 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 7 \\ 3 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 8 \\ 4 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 8 \\ 5 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 6 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 7 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 8 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 9 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 10 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 11 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

$$\delta(G) = \{1\}, d_G(1) = 2$$

$$N_G(1) = \{3,4\}, d_G(3) = 8, d_G(4) = 8 \rightarrow \textcircled{3} \text{ 선택}$$

$$A = \begin{bmatrix} & 2 & 4 & \text{계} \\ 2 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \end{bmatrix}$$

$$N_G(v_i) = 2 \cap 4 = \{5,6,7,8,9,10,11\} \rightarrow \textcircled{5} \text{ 선택}$$

그림 4. 예제 데이터
Fig. 4. Example Data

IV. 실험 및 결과 분석

본 장에서는 먼저, Daskin과 Stern^[1]이 표 1에 대해

표 3. 오스틴시 33-노드 망에 대한 T=10의 0-1 정수 계획법 데이터

Table 3. 0-1 Integer Programming Data with T=10 for Austin's 33-node Network

T=10	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	차수				
1	0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	14				
2	1	0	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	1	1	0	1	0	0	0	0	0	20				
3	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0	1	1	0	0	1	1	0	0	0	0	20				
4	0	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	0	1	0	0	0	1	1	1	1	0	21				
5	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	20				
6	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0	0	1	0	0	1	0	22				
7	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0	0	0	0	1	0	1	1	0	1	0	0	21				
8	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	20				
9	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	16			
10	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	1	1	0	18				
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	1	1	0	1	0	1	0	1	0	20				
12	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	18			
13	0	1	0	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	0	1	1	0	0	0	1	0	0	18			
14	0	1	0	1	1	1	1	1	1	1	1	1	0	1	0	0	0	1	1	0	1	0	0	0	1	1	0	0	0	0	1	0	0	1	0	18		
15	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1	0	18		
16	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	13		
17	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	12		
18	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0	0	13		
19	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	17	
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	
21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	5	
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	6	
24	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	0	0	5		
25	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	9	
26	1	1	0	0	0	1	0	0	0	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	
27	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	7	
28	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	14	
29	0	0	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	9	
30	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	4	
31	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	14
32	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

CSC와 HOSC로 해를 얻지 못한 최대 도착 허용시간 $T=9, 10, 14$ (분)를 포함한 모든 $T=[3,20]$ 에 대한 해를 구하여 본다.

표 1의 데이터에 대해 $T=10$ 을 적용하면, $t_{ij} \leq 10$ 을 1로, $t_{ij} > 10$ 로 치환하여 제안된 알고리즘을 수행한 결과는 표 3과 같으며 응급시설 위치는 $\{1,25,29,31\}$ 의 4개 노드로 4대임을 알 수 있다. 표 1에서, 각 행의 최대 값의 최소값은 15이며, 이에 해당되는 노드는 7과 11이다. 7과 11 노드 중 모든 노드까지의 총 소요시간이 최소인 11 노드를 최적 위치로 선정하였다. 즉, $15 \leq T \leq 20$ 에 대한 응급시설의 위치는 11 노드로 $16 \leq T \leq 20$ 은 구하지 않아도 됨을 알 수 있다. 제안된 알고리즘으로 구한 결과를 CSC, HOSC와 비교한 결과는 표 4에 제시되어 있다.

다음으로, Swain 55-노드 망에 대해 모든 노드를 커버할 수 있도록 $T=15$ 를 적용하여 제안된 알고리즘을 적용한 결과는 그림 5에 제시되어 있다. $T=15$ 의 경우 정확히 5개의 응급시설 위치인 53,21, 43,22,36을 얻었다. Swain 55-노드 망에 대해서는 편의상 각 노드들에서의 최단거리를 계산하지 않고 시각적으로 결정하는 방법을 적용하였다.

T=10	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	차수
1	0	1	1	1	1											1	1	1	1	1	1		0	0	0	1		0	0	0	0		10	
2	1	0	1	1	1											1	1	1	1	1	0		0	0	1	1		1	0	0	0		11	
3	1	1	0	1	1											1	1	1	1	1	0		0	1	1	0		1	1	0	0		12	
5	1	1	1	1	0											1	1	1	1	1	0		0	0	0	0		1	0	0	0		9	
16	1	1	1	1	1											0	1	1	1	1	1		0	0	1	0		1	0	0	0		11	
17	1	1	1	1	1											1	0	1	1	0	1		1	0	1	0		1	0	0	0		11	
18	1	1	1	1	1											1	1	0	0	0	0		1	1	1	0		1	0	0	0		10	
19	1	1	1	1	1											1	1	0	0	1	0		0	0	0	1		0	0	0	0		8	
20	1	1	1	1	1											1	0	0	1	0	0		0	0	0	1		0	0	0	0		7	
21	1	0	0	0	0											1	1	0	0	0	0		1	0	1	0		0	0	0	0		5	
23	0	0	0	0	0											0	1	1	0	0	1		0	1	1	0		1	0	0	0		6	
24	0	0	1	0	0											0	0	0	0	0	0		1	0	1	0		1	0	1	0		5	
25	0	1	1	0	0											1	1	1	0	0	1		1	1	0	0		1	0	0	0		9	
26	1	1	0	0	0											0	0	0	1	1	0		0	0	0	0		0	0	0	0		4	
28	0	1	1	1	1											1	1	1	0	0	0		1	1	1	0		0	0	1	0		10	
29	0	0	1	0	0											0	0	0	0	0	0		0	0	0	0		0	0	1	1		3	
30	0	0	0	0	0											0	0	0	0	0	0		0	1	0	0		1	1	0	0		3	
31																																		
32	0	0	0	0	0											0	0	0	0	0	0		0	0	0	0		0	1	0	0		1	

T=10	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	차수
1	0	1			1											1	1	1	1	1	1		0	0	0	1		0					9	
2	1	0			1											1	1	1	1	1	0		0	0	1	1		1					10	
5	1	1			0											1	1	1	1	1	0		0	0	0	0		1					8	
16	1	1			1											0	1	1	1	1	1		0	0	1	0		1					10	
17	1	1			1											1	0	1	1	0	1		1	0	1	0		1					10	
18	1	1			1											1	1	0	0	0	0		1	1	1	0		1					9	
19	1	1			1											1	1	0	0	1	0		0	0	0	1		0					7	
20	1	1			1											1	0	0	1	0	0		0	0	0	1		0					6	
21	1	0			0											1	1	0	0	0	0		1	0	1	0		0					5	
23	0	0			0											0	1	1	0	0	1		0	1	1	0		1					6	
24	0	0			0											0	0	0	0	0	0		1	0	1	0		1					3	
25	0	1			0											1	1	1	0	0	1		1	1	0	0		1					8	
26	1	1			0											0	0	0	1	1	0		0	0	0	0		0					4	
28	0	1			1											1	1	1	0	0	0		1	1	1	0		0					8	
29																																		
31																																		

T=10	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	차수	
1	0				1														1	1					1									4	
5	1				0														1	1														3	
19	1				1														0	1						1								4	
20	1				1														1	0						1								4	
25																			1	0															
26	1				0														1	1						0								3	
29																																			
31																																			

표 4. 오스틴시 33-노드 망에 대한 CSC, HOSC와 제안 알고리즘 결과 비교
 Table 4. Results of CSC and HOSC for Austin's 33-Node Network

T (분)	CSC		HOSC		제안 알고리즘	
	응급시설수	응급시설 위치	응급시설수	응급시설 위치	응급시설수	응급시설 위치
3	27	1,2,3,4,5,9,10,11,12,13,16,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33	27	1,2,3,4,6,7,9,10,12,13,16,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33	27	1,2,3,6,8,9,11,12,14,15,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33
4	20	1,2,8,13,15,16,18,20,21,22,23,24,25,26,27,28,29,30,32,33	20	1,2,8,13,15,16,18,20,21,22,23,24,25,26,27,28,29,30,32,33	20	1,2,6,8,13,15,16,18,21,22,23,24,25,26,27,28,29,30,32,33
5	16	2,10,11,17,21,22,23,24,25,26,27,28,29,30,32,33	16	2,10,11,17,21,22,23,24,25,26,27,28,29,30,32,33	16	2,11,17,21,22,23,24,25,26,27,28,29,30,31,32,33
6	12	5,10,12,16,21,22,24,27,29,30,32,33	12	7,10,12,16,21,22,24,27,29,30,32,33	12	3,10,12,16,21,22,24,27,29,30,32,33
7	8	11,16,22,24,29,30,32,33	8	4,11,16,24,27,30,32,33	8	4,12,16,24,27,30,32,33
8	6	9,20,22,25,28,33	6	9,20,25,27,28,31	6	9,19,25,27,28,33
9	6	?	6	8,11,16,24,27,31	6	2,22,25,28,29,31
10	3.75	?	3.75	?	4	1,25,29,31
11	3	19,28,31	3	2,28,31	3	19,28,31
12	2	15,18	2	15,18	2	15,18
13	2	11,16	2	3,11	2	11,25
14	2	?	2	7,11	2	7,22
15	1	7	1	7	1	11
16	1	6	1	6	1	11
17	1	4	1	4	1	11
18	1	4	1	4	1	11
19	1	2	1	2	1	11
20	1	2	1	2	1	11

- Operations Research, Vol. 4, No. 3, pp. 233-235, 1979.
- [5] R. M. Karp, "Reducibility Among Combinatorial Problems", Complexity of Computer Computations, New York: Plenum, pp. 85 - 103, 1972.
- [6] Wikipedia, "List of NP-complete Problems", http://en.wikipedia.org/wiki/List_of_NP-complete_problems, 2011.
- [7] K. J. Devlin, "The Millennium Problems: The Seven Greatest Unsolved Mathematical Puzzles of Our Time", Basic Books. 2002.
- [8] R. J. Vanderbei, "Linear Programming: Foundations and Extensions", 3rd ed., International Series in Operations Research & Management Science, Vol. 114, Springer Verlag, 2008.
- [9] R. Church and C. ReVelle, "The Maximal Covering Location Problem", Journal of Regional Science, Vol. 32, pp. 101-118, 1974.
- [10] D. Serra and C. ReVelle, "Surviving in a Competitive Spatial Market: The Threshold Capture Model", Journal of Regional Science, Vol. 39, Issue. 4, pp. 637-650, 1999.

저자 소개

최 명 복(종신회원)



- 1992년 : 호서대학교 전자계산학과 (학사)
- 1994년 : 아주대학교 컴퓨터공학과 (석사)
- 2001년 : 아주대학교 컴퓨터공학과 (박사)
- 1997~현재 : 강릉원주대학교 멀티미디어공학과 교수

• 2004. 1~현재 : 한국인터넷방송통신학회 이사
 <주관심분야 : 지능형 정보검색, 퍼지응용, 지식표현, 신경망, 임베디드 및 유비쿼터스 응용, 지능형 교통제어, 소프트웨어 공학, 알고리즘>
 • e-mail : cmb5859@gmail.com

한 태 용(정회원)



- 1979년 3월 : 경희대학교 체육대학 체육학과 학사
- 1983년 3월 : 경희대학교 교육대학원 체육교육학 석사
- 1991년 1월 : UNITED STATES SPORTS ACADEMY 교육학박사

• 강릉원주대학교 문화대학 여성인력개발학과 교수
 <관심분야 : 컴퓨터 경영분야>
 • e-mail : tyhan@gwmu.ac.kr

김 봉 경(정회원)



- 1991년, 상지대학교, 체육학과, 체육학사
- 1999년, 상지대학교, 체육학과, 체육학석사
- 2002년, 숙명여자대학교, 체육학과, 이학박사
- 남서울대학교 스포츠경영학과, 전임강사

<관심분야 : 스포츠경영, 컴퓨터경영>
 • e-mail : kbg4340@nsu.ac.kr