

<https://doi.org/10.7236/JIIBC.2023.23.4.165>

JIIBC 2023-4-25

## 포함-배제 원리를 적용한 집합피복 문제의 경쟁 알고리즘

# Competitive Algorithm of Set Cover Problem Using Inclusion-Exclusion Principle

이상운\*

Sang-Un Lee\*

**요약** 본 논문은 지금까지 NP-완전 문제로 다항시간 알고리즘이 존재하지 않는 집합피복 문제에 대해 선형시간으로 해를 구할 수 있는 알고리즘을 제안하였다. 지금까지는 최대로 피복할 수 있는 집합을 선택하는 휴리스틱 탐욕 알고리즘만이 알려져 있다. 반면에 제안된 알고리즘은 최대 원소 수 2순위 또는 3순위까지의 노드들  $k$ 개를 대상으로 포함-배제 원리 규칙을 적용하여  $N$ 개의 노드 모두를 피복하는 피복집합을 구하고, 이들 중 최소 피복집합을 선택하는 경쟁 알고리즘이다. 제안된 알고리즘은 욕심쟁이 알고리즘이 최적 해를 얻지 못하는 단점을 보완하였다. 제안된 알고리즘을 다양한 응용 사례들에 적용한 결과  $O(kn^2)$ 의 다항시간으로 최적 해를 얻었다.

**Abstract** This paper proposes an algorithm that can obtain a solution with linear time for a set cover problem(SCP) in which there is no polynomial time algorithm as an NP-complete problem so far. Until now, only heuristic greed algorithms are known to select sets that can be covered to the maximum. On the other hand, the proposed algorithm is a competitive algorithm that applies an inclusion-exclusion principle rule to  $N$  nodes up to 2nd or 3rd in the maximum number of elements to obtain a set covering all  $k$  nodes, and selects the minimum cover set among them. The proposed algorithm compensated for the disadvantage that the greedy algorithm does not obtain the optimal solution. As a result of applying the proposed algorithm to various application cases, an optimal solution was obtained with a polynomial time of  $O(kn^2)$ .

**Key Words** : Set cover, Cardinality, Greedy algorithm, NP-Complete, Inclusion-exclusion principle

## 1. 서론

$m$ 개의 원소로 구성된 전체집합(universe set)  $U$ 의 모든 원소들을  $n$ 개의 부분집합  $S(S \subseteq U)$ 가 중복(overlap)되게 커버(피복, cover)하고 있는 경우,  $S$ 의 모든 부분집

합의 합은  $U$ 가 된다. 최소 집합피복(minimum set cover, MSC,  $C$ ) 문제는  $U$ 의 모든 원소들을 커버하는  $S$ 의 최소의 부분집합  $C(C \subseteq S)$ 를 찾는 문제로  $S$ 에서 중복을 가능한 최대로 제거한 최소한의 집합이 된다. 즉,  $C$ 의 모든 부분집합의 합집합도  $U$ 가 된다. 이 문제는  $U$ 의

\*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과  
접수일자 2023년 2월 1일, 수정완료 2023년 7월 13일  
게재확정일자 2023년 8월 4일

Received: 1 February, 2023 / Revised: 13 July, 2023 /  
Accepted: 4 August, 2023

\*Corresponding Author: [sulee@gwnu.ac.kr](mailto:sulee@gwnu.ac.kr)

Dept. of Multimedia Eng., Gangneung-Wonju National University, Korea

모든 원소들은 적어도  $C$ 의 어느 한 집합에 포함되어야 하고,  $U$ 의 원소들은 중복되어 포함되어도 무방하다.<sup>[1-4]</sup>

MSC 문제는 배송 및 노선 문제, 일정문제와 위치 선정 문제, 메시지 교환 이론, VLSI 회로시험, 회로 균등 문제, 항공사 운행일정 문제 등 다양한 분야에 적용되고 있다.<sup>[5]</sup>

MSC 문제는 NP-완전(NP-complete)으로 Karp의 21개 NP-완전 문제들 중 하나로 알려져 있으며<sup>[2]</sup>, 해를 구하는 가장 간단한 방법으로 욕심쟁이 알고리즘(greedy algorithm)이 제안되어 있다. 그러나 욕심쟁이 알고리즘은 최적 해(optimal solution)를 찾지 못하는 단점을 갖고 있다.<sup>[1,3]</sup>

본 논문은 경쟁 알고리즘(competitive algorithm, CA)을 적용하여 MSC의 최적 해를 찾는 방법을 제안한다. 2장에서는 MSC 관련 연구 동향과 문제점을 고찰해 본다. 3장에서는 MSC 문제를 다항시간으로 정확한 해를 구하는 경쟁 알고리즘을 제안한다. 4장에서는 다양한 데이터들을 대상으로 제안된 알고리즘의 적용성을 평가해 본다.

## II. 관련연구와 문제점

MSC 문제의 해를 구하는 욕심쟁이 알고리즘<sup>[1-3]</sup>은 그림 1에 제시되어 있다.

---

$C = \{\phi\}$   
 각 부분집합의 원소 개수 계산.  
 While  $\forall S_i = \{\phi\}$   
     최대 원소 개수 (Maximum Cardinality)  $S_i$  선택,  
 $C \leftarrow C \cup \{S_i\}$   
     선택되지 않은 집합  $S \setminus S_i$ 에 대해  $S_i$ 와 중복 원소 삭제.  
     각 부분집합의 원소 개수 계산.  
 Next

---

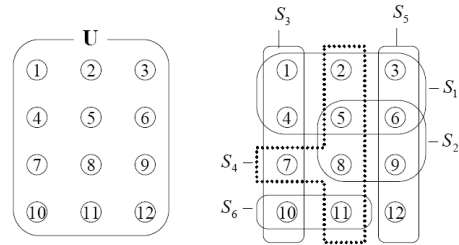
그림 1. MSC 욕심쟁이 알고리즘  
 Fig. 1. MSC Greedy algorithm

12개 원소로 구성된  $U = \{1, 2, \dots, 12\}$ 와 6개 부분집합의 모임인  $S = \{S_1, S_2, S_3, S_4, S_5, S_6\}$ 이 그림 2의  $MSC_1$ 과 같이 주어진다 가정하였을 때  $U$ 의 모든 원소를 포함하는  $S$ 의 부분집합  $C$ 를 구하여 보자.  $MSC_1$  데이터는 Wan과 Ba<sup>[3]</sup>와 Sadeghi<sup>[6]</sup>에서 인용되었다.

$MSC_1$ 의 최적 해는 그림 3과 같이  $C = \{S_3, S_4, S_5\}$ 이다. 즉,  $U$ 의 모든 원소 12개를 모두 포함하는 최소의  $C \subseteq S$ 는  $\{S_3, S_4, S_5\}$ 이며,  $S_3 \cup S_4 \cup S_5 = U$ 가 된다.

$MSC_1$ 에 대해 욕심쟁이 알고리즘을 적용한 결과 표 1

과 같이 4개의 부분집합을 얻어 3개의 부분집합이 최적 해인 값을 얻는데 실패하였다.<sup>[3]</sup>



$U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$   
 $S = \{S_1, S_2, S_3, S_4, S_5, S_6\}$   
 s.t.  $S_1 = \{1, 2, 3, 4, 5, 6\}$ ,  $S_2 = \{4, 5, 6, 7, 8, 9\}$ ,  $S_3 = \{1, 4, 7, 10\}$ ,  
 $S_4 = \{2, 5, 7, 8, 11\}$ ,  $S_5 = \{5, 6, 8, 9\}$ ,  $S_6 = \{10, 11\}$ .

그림 2.  $MSC_1$  문제  
 Fig. 2.  $MSC_1$  problem

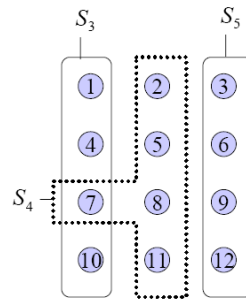


그림 3.  $MSC_1$  문제의 최적 해  
 Fig. 3. Optimal solution of  $MSC_1$

표 1.  $MSC_1$ 의 욕심쟁이 알고리즘 수행 결과  
 Table 1. Result of greedy algorithm for  $MSC_1$

수행 횟수	$S_i$	$ S_i $	선택	미 커버 원소
1	$S_1 = \{1, 2, 3, 4, 5, 6\}$	6	$S_1$	-
	$S_2 = \{5, 6, 8, 9\}$	4		$S_2 = \{8, 9\}$
	$S_3 = \{1, 4, 7, 10\}$	4		$S_3 = \{7, 10\}$
	$S_4 = \{2, 5, 7, 8, 11\}$	5		$S_4 = \{7, 8, 11\}$
	$S_5 = \{3, 6, 9, 12\}$	4		$S_5 = \{9, 12\}$
	$S_6 = \{10, 11\}$	2		$S_6 = \{10, 11\}$
2	$S_2 = \{8, 9\}$	2	$S_4$	$S_2 = \{9\}$
	$S_3 = \{7, 10\}$	2		$S_3 = \{10\}$
	$S_4 = \{7, 8, 11\}$	3		-
	$S_5 = \{9, 12\}$	2		$S_5 = \{9, 12\}$
	$S_6 = \{10, 11\}$	2		$S_6 = \{10\}$
3	$S_2 = \{9\}$	1	$S_5$	$S_2 = \{\}$
	$S_3 = \{10\}$	1		$S_3 = \{10\}$
	$S_5 = \{9, 12\}$	2		-
	$S_6 = \{10\}$	1		$S_6 = \{10\}$
4	$S_3 = \{10\}$	1	$S_3$ or $S_6$	-
	$S_6 = \{10\}$	1		$S_6 = \{\}$
$C = \{S_1, S_3, S_4, S_5\}$ or $C = \{S_1, S_4, S_5, S_6\}$				

또 다른 문제점은 임의의 수행 횟수에서 동일한 원소 개수를 가진 부분집합이 다수 존재할 경우 어떤 부분집합을 선택할 것인가가 문제가 된다. 예로, 4번째 수행횟수에서  $S_3$ 와  $S_6$ 는 모두 원소 개수가 1개로 동일하다. 이 경우  $S_6$ 을 선택하여  $C = \{S_1, S_4, S_5, S_6\}$ 을 얻은 경우,  $S_1, S_4, S_5, S_6$ 중 어느 한 부분집합이 나머지 3개 부분집합의 합에 진부분집합이 되지 않아 더 이상 최소로 할 수 없는 결과를 초래한다.

MSC의 최적 해를 구하는 욕심쟁이 알고리즘을 적용하는데 있어 다음과 같이 2가지 문제점이 있다.

[문제점 1] 욕심쟁이 알고리즘을 적용하여 얻은 해는 어느 하나의 부분집합이 다른 모든 부분집합의 합에 진부분집합이 존재하여 최적 해를 얻는데 실패할 수 있다.

[문제점 2] 임의의 단계에서 동일한 원소개수를 가진 부분집합이 다수 존재할 경우 어떤 부분집합을 선택하느냐에 따라 사색이 가능한 진부분집합을 얻지 못할 수 있다.

이러한 문제점을 개선하기 위해 Lee<sup>[7]</sup>는 동일한 최대 원소 수를 가진 경우, 탐욕 알고리즘으로 얻은 해 중에서 타 부분집합의 진부분집합이 존재하는 경우에 대해 해결책을 제시하였다. 그러나 이 방법 역시 특정 문제에 대해서는 최적 해를 찾지 못하는 경우가 발생하고 있어 완벽한 알고리즘이라고 할 수 없다.

### III. 경쟁 알고리즘

본 장에서는 욕심쟁이 알고리즘의 단점을 해결하여 MSC의 최적 해를 구할 수 있는 경쟁 알고리즘을 제안한다. 제안된 알고리즘은 그림 4의 포함-배제 원리(inclusion-exclusion principle, IEP)를 적용한다. 이 원리를 적용 시 제기되는 문제점은  $|UV| = (|U|+|V|) - (|U \cap V|)$ 에서  $|UV|$ 는  $|U \cap V|$ 에 따라 결정되며,  $|U \cap V|$ 는 사전에 알지 못한다. 따라서  $i$ 개의 부분집합들 중에서  $|U|$ 가 가장 큰 경우가 반드시  $|UV|$ 가 가장 크다고 할 수 없다. 이러한 사례는  $MSC_1$  문제에서 고찰하였으며, 이는 욕심쟁이 알고리즘으로 해결할 수 없는 문제이다.

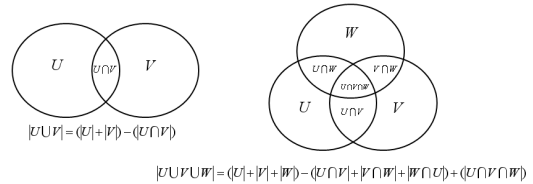
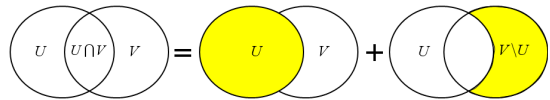


그림 4. 포함-배제 원리  
 Fig. 4. Inclusion- exclusion principle(IEP)

본 장에서는 이러한 문제점을 해결하기 위해,  $|U|$ 가 가장 큰 값 내림차순으로 2순위 또는 최대 3순위까지의 부분집합을 선택하여 식 (1)로 최적 해를 결정하는 방법을 적용한다.

$$|UV| = |U| + |V \setminus U| \tag{1}$$

$$|UVW| = |U| + |W \setminus U| + |W \setminus UV|$$



포함-배제 원리를 적용한 경쟁 알고리즘은 그림 5에 제시되어 있다.

```

C = {ϕ}
S × N 행렬 작성. cij = { 1, Nj ∈ Si
                      0, Nj ∉ Si /* 0-1 정수계획법
각 Si행의 원소 개수 (합) 계산. /* 수행 복잡도: O(n)
최대 원소 개수 Si를 내림차순으로 3순위까지 k개 선택. (2순위가
동일한 값이 다수 존재하면 2순위까지만, 1,2순위가 1개씩만 존재하면
3순위까지 선택) /* 수행 복잡도: O(nlogn)
for i = 1, 2, ..., k do /* 수행 복잡도: O(kn2)
    while N = {ϕ}
        각 Si의 cij = 1인 열 Nj 삭제. Ci ← Ci ∪ {Si}.
        남은 집합 S의 합 계산. 최대 원소 수 Si 선택.
    next
end
최소 원소개수 Ci 선택.
    
```

그림 5. 포함-배제 원리 적용 경쟁 알고리즘  
 Fig. 5. Comparative algorithm using IEP

$MSC_1$ 에 대해 3순위까지 적용한 경쟁 알고리즘을 적용한 결과 표 2와 같이 최적 해  $|C|=3$ 을 얻는데 성공하였다. 1순위  $S_1(6)$ , 2순위  $S_4(5)$ 와 3순위인  $S_2(4)$ 는 최적 해를 구하지 못한 반면, 3순위인  $S_3(4)$ 와  $S_5(4)$ 는 최적 해  $C = \{S_3, S_4, S_5\}$ 를 구하였다.

### IV. 실험 및 결과 분석

본 장에서는 MSC 문제의 최적 해를 구하는 실험 데이터로 응용 사례를 고찰해 본다. MSC 문제의 최적 해를 구하는 응용 사례는 그림 6<sup>[8-11]</sup>과 Storbeck과 Vohra<sup>[12]</sup>에서 인용된 21-노드 망인  $MSC_5$ , Daskin<sup>[13]</sup>에서 인용된  $MSC_6$ 이다.

표 2.  $MSC_1$ 의 경쟁 알고리즘 수행 결과

Table 2. Result of comparative algorithm for  $MSC_1$

MSC1	U												Cover	Rank	
	1	2	3	4	5	6	7	8	9	10	11	12			
S	1	1	1	1	1	1								6	1
	2				1	1			1	1				4	3
	3	1			1			1			1			4	3
	4		1			1		1	1			1		5	2
	5				1		1						1	4	3
	6										1	1		2	6

Rank	포함-배제원리 적용
1	$S_1(6)$ $S_1(6) + S_1(3) + S_5(2) + S_2(1)$ $S_1(6) + S_4(3) + S_5(2) + S_6(1)$
2	$S_4(5)$ $S_4(5) + S_1(4) + S_5(2) + S_6(2)$ $S_4(5) + S_5(4) + S_1(2) + S_6(2)$
3	$S_2(4)$ $S_2(4) + S_1(4) + S_3(2) + S_4(1) + S_5(1)$ $S_2(4) + S_1(4) + S_4(2) + S_3(1) + S_5(1)$ $S_2(4) + S_1(4) + S_5(2) + S_4(2)$ $S_2(4) + S_3(4) + S_1(2) + S_4(1) + S_5(1)$ $S_2(4) + S_3(4) + S_4(2) + S_5(2)$ $S_2(4) + S_3(4) + S_5(2) + S_4(2)$
	$S_3(4)$ $S_2(4) + S_1(4) + S_5(4)$ $S_2(4) + S_5(4) + S_4(4)$
	$S_5(4)$ $S_5(4) + S_4(5) + S_3(3)$
	최적해 $C = \{S_3, S_4, S_5\}$

$U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$ ,  $S = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$   
 s.t.  $S_1 = \{1, 2, 3, 5, 9, 10, 11\}$ ,  $S_2 = \{1, 2\}$ ,  $S_3 = \{2, 8\}$ ,  
 $S_4 = \{5, 6\}$ ,  $S_5 = \{9, 10\}$ ,  $S_6 = \{6, 7, 10, 11\}$ ,  
 $S_7 = \{3, 4, 7, 8, 11, 12\}$ .

(a)  $MSC_2$

$U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$   
 $S = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9, S_{10}, S_{11}\}$   
 s.t.  $S_1 = \{1, 2, 3, 4\}$ ,  $S_2 = \{1, 2, 3, 5\}$ ,  $S_3 = \{1, 2, 3, 4, 5, 6\}$ ,  
 $S_4 = \{1, 3, 4, 6, 7\}$ ,  $S_5 = \{2, 3, 5, 6, 8, 9\}$ ,  
 $S_6 = \{3, 4, 5, 6, 7, 8\}$ ,  $S_7 = \{4, 6, 7, 8\}$ ,  
 $S_8 = \{5, 6, 7, 8, 9, 10\}$ ,  $S_9 = \{5, 8, 9, 10, 11\}$ ,  
 $S_{10} = \{8, 9, 10, 11\}$ ,  $S_{11} = \{9, 10, 11\}$ .

(b)  $MSC_3$

$U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ ,  $S = \{S_1, S_2, S_3, S_4, S_5\}$   
 s.t.  $S_1 = \{2, 3, 4, 5\}$ ,  $S_2 = \{3, 4, 6, 7, 8, 9\}$ ,  $S_3 = \{1, 2, 4, 8\}$ ,  
 $S_4 = \{7, 8, 10\}$ ,  $S_5 = \{4, 6, 7, 8\}$ .

(c)  $MSC_4$

그림 6. MSC 응용 사례  
 Fig. 6. Applied examples for MSC

$MSC_2$ 는 Skiena<sup>[8]</sup>에서 인용되었다.  $MSC_3$ 는 Trick<sup>[9]</sup>에서 인용된 데이터로, 11개의 동으로 구분되어 있는 도시에 대해 소방서를 최소로 몇 개를 설치하면 11개 동을 모두 커버할 수 있는지를 결정하는 문제이다. 각 소방서는 인접한 동을 커버한다고 가정한다.  $S_i$ 는 각 동에 대해 인접한 동을 커버하는 원소들로 구성할 수 있다. Trick<sup>[9]</sup>은  $MSC_3$ 을 선형계획법을 적용하여 최적해로  $x_3 = x_8 = x_9 = 1$  즉,  $C = \{S_3, S_8, S_9\}$ 를 구하였다.

$MSC_4$ 는 Shuai와 Hu<sup>[10]</sup>에서 인용된 데이터로, 광통신 망 연결 방송의 주파수 할당 문제이다. 10개 지역 간에 5개의 주파수 광통신망이 설치되어 있다. 이는 10개 지역을 커버할 수 있는 최소의 연결된 주파수 망을 선택하는 문제로 연결 최소 집합피복(connected MSC) 문제이다.  $S_i$ 는 각 주파수  $w_i = \{v_1, v_2\}$ 의 2개 정점을 원소로 하는 집합으로 표기할 수 있다. 본 문제는 10개 지역을 연결하는 주파수 망을 찾아야 하며, 사이클이 존재하면 안된다. 이 문제의 최적 해는  $C = \{S_1, S_2, S_3, S_4\}$ 로 알려져 있다.

Storbeck과 Vohra<sup>[12]</sup>는 21-노드 망  $MSC_5$ 에 Natural Slack Covering(NSC) 알고리즘을 적용하여  $T=2$ 의 해인  $S = \{S_5, S_{11}, S_{13}, S_{20}\}$  위치를 찾았다.

$MSC_6$ 은 Daskin<sup>[13]</sup>에서 인용된 데이터로, 미국 텍사스주 오스틴시를 국제조사표준구역(census tracts)에 의해 33개의 구역(zones)으로 분할하여 각 구역 간 소요(응답) 시간(분)을 나타내고 최대 허용 응답시간  $T$ 를 10분으로 설정하여 구급차량의 위치를 선형계획법으로 해를 구하지 못하였다.

5개의 사례에 대해 경쟁 알고리즘을 적용하여 구한 최적 해는 그림 7에 제시되어 있다.  $MSC_3$ 의 경우 11개 동을 커버하는 최소한의 소방서 개수 3개를 구하는데 성공하였다.  $MSC_4$ 의 경우  $C = \{S_1, S_2, S_3, S_4\}$  또는  $C = \{S_1, S_3, S_4, S_5\}$ 를 얻어  $w_2$  또는  $w_5$  주파수 망을 삭제할 수 있다.

욕심쟁이 알고리즘은  $MSC_1, MSC_3, MSC_5$ 에서 최적 해를 얻지 못하는 반면에, 제안된 알고리즘은 6개의 데이터 모두에서 최적 해를 얻었으며, 다양한 부분집합들로 구성된 최적해  $C$ 도 얻을 수 있다.

MSC2	U												Cover	Rank	
	1	2	3	4	5	6	7	8	9	10	11	12			
S	1	1	1	1		1				1	1	1		7	1
	2	1	1											2	4
	3		1						1					2	4
	4					1	1							2	4
	5								1	1				2	4
	6						1	1			1	1		4	3
	7			1	1			1	1			1	1	6	2

Rank		포함-배제 원리 적용
1	$S_1(7)$	$S_1(7) + S_7(4) + S_4(1)$
		$S_1(7) + S_7(4) + S_6(1)$
2	$S_7(6)$	$S_7(6) + S_1(5) + S_1(1)$
		$S_7(6) + S_1(5) + S_6(1)$
3	$S_6(4)$	$S_6(4) + S_1(5) + S_7(3)$
최적해		$C = \{S_1, S_4, S_7\}$ or $C = \{S_1, S_6, S_7\}$

(a)  $MSC_2$

MSC3	U											Cover	Rank
	1	2	3	4	5	6	7	8	9	10	11		
S	1	1	1	1	1							4	7
	2	1	1	1		1						4	7
	3	1	1	1	1	1	1					6	1
	4	1		1	1		1	1				5	5
	5		1	1		1	1	1	1			6	1
	6			1	1	1	1	1	1			6	1
	7				1		1	1	1			4	7
	8					1	1	1	1	1	1	6	1
	9					1	1			1	1	5	5
	10							1	1	1	1	4	7
	11								1	1	1	3	11

Rank		포함-배제 원리 적용
1	$S_3(6)$	$S_3(6) + S_3(4) + S_{10}(1), S_{11}(1)$ $S_3(6) + S_{10}(4) + S_4(1), S_6(1), S_7(1), S_8(1)$
	$S_5(6)$	$S_5(6) + S_4(3) + S_6(2), S_{10}(2), S_{11}(2)$
	$S_6(6)$	$S_6(6) + S_9(3) + S_1(2), S_2(2), S_3(2)$ $S_6(6) + S_{10}(3) + S_1(2), S_2(2), S_3(2)$ $S_6(6) + S_{11}(3) + S_1(2), S_2(2), S_3(2)$
		$S_8(6)$
2	$S_4(5)$	$S_4(5) + S_5(4) + S_9(2), S_{10}(2), S_{11}(2)$ $S_4(5) + S_9(4) + S_5(2)$ $S_4(5) + S_{10}(4) + S_2(2), S_3(2), S_5(2)$
	$S_9(5)$	$S_9(5) + S_1(4) + S_6(2), S_7(2), S_8(2)$ $S_9(5) + S_3(4) + S_6(2), S_7(2), S_8(2)$ $S_9(5) + S_4(4) + S_5(2)$ $S_9(5) + S_6(4) + S_1(2), S_2(2), S_3(2)$
최적 해		$C = \{S_1, S_6, S_9\}$ or $C = \{S_1, S_6, S_{10}\}$ or $C = \{S_1, S_6, S_{11}\}$ or $C = \{S_1, S_7, S_9\}$ or $C = \{S_1, S_8, S_9\}$ or $C = \{S_1, S_8, S_{10}\}$ or $C = \{S_1, S_8, S_{11}\}$ or $C = \{S_2, S_4, S_{10}\}$ or $C = \{S_2, S_6, S_9\}$ or $C = \{S_2, S_6, S_{10}\}$ or $C = \{S_2, S_6, S_{11}\}$ or $C = \{S_3, S_4, S_{10}\}$ or $C = \{S_3, S_6, S_9\}$ or $C = \{S_3, S_6, S_{10}\}$ or $C = \{S_3, S_6, S_{11}\}$ or $C = \{S_3, S_7, S_9\}$ or $C = \{S_3, S_7, S_{10}\}$ or $C = \{S_3, S_8, S_9\}$ or $C = \{S_3, S_8, S_{10}\}$ or $C = \{S_3, S_8, S_{11}\}$ or $C = \{S_4, S_5, S_9\}$ or $C = \{S_4, S_5, S_{10}\}$ or $C = \{S_4, S_5, S_{11}\}$

(b)  $MSC_3$

MSC4	U										Cover	Rank	
	1	2	3	4	5	6	7	8	9	10			
S	1		1	1	1	1						4	2
	2			1	1		1	1	1			6	1
	3	1	1		1			1				4	2
	4							1	1		1	3	5
	5				1		1	1	1			4	2

Rank		포함-배제원리 적용
1	$S_2(6)$	$S_2(6) + S_1(2) + S_3(1) + S_4(1)$ $S_2(6) + S_5(2) + S_1(1) + S_1(1)$
		$S_1(4)$ $S_3(4)$ $S_5(4)$
최적해		$C = \{S_1, S_2, S_3, S_4\}$ or $C = \{S_1, S_3, S_4, S_5\}$

(c)  $MSC_4$

Rank		포함-배제 원리 적용			
1	$S_5(7)$ $S_{12}(7)$ $S_{14}(7)$ $S_{15}(7)$ $S_{16}(7)$	-			
		2	$S_5(6)$ $S_{11}(6)$ $S_{13}(6)$ $S_{20}(6)$	$S_{20}(6) + S_5(7) + S_7(4) + S_{13}(4)$ $S_{20}(6) + S_5(7) + S_{11}(4) + S_{13}(4)$	
			최적해		$C = \{S_5, S_7, S_{13}, S_{20}\}$ or $C = \{S_5, S_{11}, S_{13}, S_{20}\}$

(d)  $MSC_5$

Rank		포함-배제 원리 적용
1	$S_6(23)$	$S_6(23) + S_{23}(4) + S_4(3) + S_{31}(3)$ $S_6(23) + S_{23}(4) + S_8(3) + S_{31}(3)$ $S_6(23) + S_{23}(4) + S_{29}(3) + S_{31}(3)$ $S_6(23) + S_{25}(4) + S_4(3) + S_{31}(3)$ $S_6(23) + S_{25}(4) + S_8(3) + S_{31}(3)$ $S_6(23) + S_{25}(4) + S_{29}(3) + S_{31}(3)$
		2
	$S_7(22)$	
최적해		$C = \{S_1, S_4, S_{18}, S_{31}\}$ or $C = \{S_1, S_4, S_{25}, S_{31}\}$ or $C = \{S_4, S_6, S_{23}, S_{31}\}$ or $C = \{S_4, S_6, S_{25}, S_{31}\}$ or $C = \{S_4, S_7, S_{25}, S_{31}\}$ or $C = \{S_4, S_{11}, S_{25}, S_{31}\}$ or $C = \{S_4, S_{12}, S_{25}, S_{31}\}$ or $C = \{S_4, S_{19}, S_{25}, S_{31}\}$ or $C = \{S_4, S_{20}, S_{25}, S_{31}\}$ or $C = \{S_4, S_{25}, S_{25}, S_{31}\}$ or $C = \{S_6, S_8, S_{23}, S_{31}\}$ or $C = \{S_6, S_{23}, S_{29}, S_{31}\}$ or $C = \{S_6, S_8, S_{25}, S_{31}\}$ or $C = \{S_6, S_{25}, S_{29}, S_{31}\}$ or $C = \{S_7, S_8, S_{25}, S_{31}\}$

(e)  $MSC_6$

그림 7. 경쟁 알고리즘 적용 결과  
 Fig. 7. Applied result of CA

## V. 결론 및 향후 연구과제

최소 집합피복 문제의 최적 해를 다항시간으로 얻는 방법은 아직까지 알려지지 않고 있다. 그러나 최소 집합 피복 문제의 해를 간단히 얻는 방법으로 욕심쟁이 알고리즘이 제안되었지만 최적 해를 얻지 못하는 경우가 빈번히 발생하였다.

본 논문은 욕심쟁이 알고리즘의 단점을 보완하여 최적 해를 다항시간으로 얻을 수 있는 경쟁 알고리즘을 제안하였다. 이 방법은 엄밀히 말하면 원소 개수가 많은 상위 순위 2개 또는 3개의 부분집합들의 개수  $k$ 개에 대해 욕심쟁이 알고리즘을 반복수행하는 방법이라 할 수 있다. 이러한 방법을 적용한 결과 욕심쟁이 알고리즘으로 최적 해를 구하지 못하는 문제에 대해서도 최적 해를 구할 수 있음을 보였다.

6개의 사례를 대상으로 경쟁 알고리즘을 적용한 결과 욕심쟁이 알고리즘은 3개 데이터에서 최적 해를 구하지 못한 반면에, 제안된 알고리즘은 6개 데이터 모두에서 다양한 조합의 최적 해를 구할 수 있었다.

## References

- [1] M. Sudan, "Approximability of Optimization Problems: Set Cover, Minimum Multicut," MIT, <http://people.csail.mit.edu/madhu/FT99>, 2009.
- [2] Wikipedia, "Set Cover Problem," [http://en.wikipedia.org/wiki/Set\\_cover\\_problem](http://en.wikipedia.org/wiki/Set_cover_problem), Retrieved Apr. 2022.
- [3] V. K. Wan and K. D. Ba, "CS 105: Algorithms - Set Cover and Application to Shortest Superstring," Computer Science, Dartmouth Collage, 2005.
- [4] P. Chew, "CS 482: Introduction to Analysis of Algorithms," Cornell University, 2007.
- [5] K. Hoffman and M. Padberg, "Set Covering, Packing and Partitioning Problems," Department of Systems Engineering and Operations Research George Mason University, 2009.
- [6] I. Sadeghi, "An Introduction to Approximation Algorithms," <http://mehr.sharif.edu/~ce354/seminars/sadeghi.pdf>, 2003.
- [7] S. U. Lee, "An Improved Greedy Algorithm for Set Cover Problem", Journal of Korean Institute of Information Technology, Vol. 13, No. 2, pp. 83-91, Feb. 2015, <https://doi.org/10.14801/jkiit.2015.13.2.83>
- [8] S. Skiena, "The Stony Brook Algorithm Repository, Department of Computer Science, Stony Brook University, 2008.

- [9] M. A. Trick, "A Tutorial on Integer Programming," The Operations Research Faculty of GSIA, 1997.
- [10] T. P. Shuai and X. D. Hu, "Connected Set Cover Problem and Its Applications," AAIM 2006, LNCS 4041, pp. 243-254, 2006.
- [11] Q. Yang, J. McPeck, and A. Nofsinger, "Efficient and Effective Practical Algorithms for the Set-Covering Problem," Department of Computer Science and Software Engineering, University of Wisconsin, 2008.
- [12] J. E. Storbeck and R. V. Vohra, "A Simple Trade-off Model for Maximal and Multiple Coverage," Geographical Analysis, Vol. 20, No. 3, pp. 220-230, Jul. 1988.
- [13] M. S. Daskin, "A Hierarchical Objective Set Covering Model for Emergency Medical Service Deployment," Transportation Science, Vol. 15, No. 2, pp. 137-152, May 1981, <https://doi.org/10.1287/trsc.15.2.137>

## 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 교수
- 2007.3 ~ 2015.3 : 강릉원주대학교 멀티미디어공학과 부교수
- 2015.4 ~ 현재 : 강릉원주대학교 멀티미디어공학과 정교수
- 관심분야 : 소프트웨어 프로젝트 관리, 개발 방법론, 분석과 설계 방법론, 시험 및 품질보증, 소프트웨어 신뢰성, 인공지능과 빅데이터분석, 최적화 알고리즘
- e-mail : [sulee@gwnu.ac.kr](mailto:sulee@gwnu.ac.kr)