

<https://doi.org/10.7236/JIIBC.2023.23.2.185>

JIIBC 2023-2-24

완전 피복 문제의 원소 수 기반 알고리즘

Algorithm Based on Cardinality Number of Exact Cover Problem

이상운*

Sang-Un Lee*

요약 본 논문은 지금까지 NP-완전 문제로 다항시간 알고리즘이 존재하지 않는 완전피복 문제에 대해 선형시간으로 해를 구할 수 있는 알고리즘을 제안하였다. 제안된 알고리즘은 "행과 열에는 동일한 값이 존재하면 안된다"는 완전피복문제의 특징을 이용하였다. 이를 위해 먼저 최소 원소 개수를 가진 부분집합을 선택하고 선택된 부분집합의 원소를 가진 부분집합을 삭제하였다. 남은 부분집합들을 대상으로 반복적으로 수행하면 해를 구한다. 만약, 해를 구하지 못하면 최대 원소 개수를 가진 부분집합을 선택하여 동일한 과정을 수행하였다. 제안된 알고리즘은 일반적인 완전피복 문제의 해를 쉽게 구하였다. 추가로, 완전피복 문제를 보다 일반화한 N-퀸 문제를 대상으로 제안된 알고리즘을 적용할 수 있음을 보였다. 결국, 제안된 완전피복 알고리즘은 완전피복 문제에 대해 P-문제임을 증명하였다.

Abstract To the exact cover problem that remains NP-complete to which no polynomial time algorithm is made available, this paper proposes a linear time algorithm that yields an optimal solution. The proposed algorithm makes use of the set cover problem's major feature which states that "no identical element shall be included in more than one covering set". To satisfy this criterion, the proposed algorithm initially selects a subset with the minimum cardinality and deletes those that contain the cardinality identical to that of the selected subset. This process is repeatedly performed on remaining subsets until the final solution is obtained. Provided that the solution is unattainable, it selects subsets with the maximum cardinality and repeats the same process. The proposed algorithm has not only obtained the optimal solution with ease but also proved its wide applicability on N-queens problems, hence disproving the NP-completeness of the exact cover problem.

Key Words : Exact cover, Cardinality, N-Queens problem, NP-complete

*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과
접수일자 2022년 10월 22일, 수정완료 2023년 3월 3일
게재확정일자 2023년 4월 7일

Received: 22 October, 2022 / Revised: 3 March, 2023 /
Accepted: 7 April, 2023

*Corresponding Author: sulee@gwnu.ac.kr
Dept. of Multimedia Eng., Gangneung-Wonju National University,
Korea

1. 서 론

집합 X 와 X 의 부분집합 $S(S \subseteq X)$ 의 모임(collection)이 주어졌을 때, 완전 피복(exact cover, EC) S^* 는 X 의 모든 원소들이 S^* 에 속한 하나의 집합 S_i 에만 정확히 포함된 S 의 부분 모임(sub-collection)인 S^* ($\cup S^* = X$)를 찾는 문제이다.^[1] 즉, $S_i \cap S_j = \{\emptyset\}$ 가 되어야만 한다. 완전 피복 문제는 완전 적중 집합(exact hitting set, EHS) 문제와 동치 개념이다. EHS는 EC를 역으로 치환한 것으로 생각할 수 있다. 즉, S 의 모든 원소들이 X^* 에 속한 하나의 집합 X_j 에만 정확히 포함되어야만 한다.

완전 피복 문제를 응용하고 있는 주목할 만한 예로는 비디오 게임의 일종인 테트리스(tetrix)와 유사한 펜토미노 타일 붙이기(pentomino tiling), 숫자 넣기(sudoku)와 체스 게임의 퀸을 배치하는 N Queen 문제가 있다.^[1]

완전 피복 문제는 NP-완전(NP-complete)으로 Krap의 21개 NP-완전 문제들 중 하나로 알려져 있으며, 완전 피복 문제의 모든 해를 찾는 방법은 반복적, 비결정적, 깊이우선탐색(DFS)과 백트래킹(backtracking) 방법을 적용하여 가능한 조합을 모두 시도해서 문제의 해를 찾는 Knuth의 알고리즘 X가 존재한다.^[1-12] 결국, 지금까지 정확한 해법(exact solution)을 다항시간에 얻을 수 있는 알고리즘이 제안되지 않고 있다. 참고로 Knuth의 알고리즘 X는 출발 지점 결정 시 최소 빈도수를 가진 X_j 를 선택한다는 규칙만을 제시하고 있으며, 이후부터는 해를 찾아가는 규칙이 제시되지 않는 방법이다.

본 논문은 완전피복 문제의 정확한 해를 다항시간으로 구할 수 있는 알고리즘을 제안한다. 2장에서는 완전 피복 관련 연구 동향과 문제점을 고찰해 본다. 3장에서는 완전 피복 문제를 다항시간으로 정확한 해를 구하는 알고리즘을 제안한다. 4장에서는 N개의 퀸을 배치하는 실험을 통해 제안된 알고리즘의 적용성을 평가해 본다.

II. 관련연구와 문제점

7개 원소로 구성된 $X=\{1,2,3,4,5,6,7\}$ 과 6개 부분집합의 모임인 $S=\{A,B,C,D,E,F\}$ 가 그림 1의 EC_1 과 같이 주어진다 가정하였을 때 X 의 모든 원소를 정확히 하나씩만 포함하는 S 의 부분집합 S^* 를 구하여 보자. EC_1 데이터는 Wikipedia^[1,2]와 Yun^[4]에서 인용되었다.

$X=\{1,2,3,4,5,6,7\}$, $S=\{A,B,C,D,E,F\}$
such that $A=\{1,4,7\}$; $B=\{1,4\}$; $C=\{4,5,7\}$;
 $D=\{3,5,6\}$, $E=\{2,3,6,7\}$, $F=\{2,7\}$.

		X						
		1	2	3	4	5	6	7
S	A	1			1			1
	B	1			1			
	C				1	1		1
	D			1			1	1
	E		1	1			1	1
	F		1					1

그림 1. 완전 피복 문제 EC_1

Fig. 1. EC_1 exact cover problem

그림 1 완전 피복 문제 EC_1 의 해 $S^*=\{B,D,F\}$ 이다. 즉, X 의 모든 원소를 정확히 하나씩만 포함하는 S 의 부분집합은 $BUDUF=X$ 가 되며, X 의 모든 원소 $\{1,2,3,4,5,6,7\}$ 은 $B=\{1,4\}$, $D=\{3,5,6\}$, $F=\{2,7\}$ 의 어느 한 집합에만 유일하게 속해 있다.

그림 2의 Knuth 알고리즘은 Wikipedia^[2], Yun^[3]과 Tjensvold^[4]에 기술되어 있다. Knuth 알고리즘으로 EC_1 의 해를 구하는 과정은 Wikipedia^[2]와 Yun^[3]에 자세히 기술되어 본 논문에서는 생략한다.

행 열 S , 열을 X 로 하여 값이 존재하면 '1', 존재하지 않으면 '0'으로 하는 행렬 A 작성.

```
While  $A \neq \{\emptyset\}$  RIGHT
  if 행렬  $A = \{\emptyset\}$  then 해를 얻음, 알고리즘 종료.
  else 열  $c$  선택 /* 1의 개수가 최소인 열 결정론적 선택
         $A_{r,c} = 1$ 인 행  $r$  선택 /* 비결정론적 선택.
         $S^* \leftarrow S^* \cup \{r\}$ .
         $A_{r,j} = 1$ 인 열  $j$ 와  $A_{i,j} = 1$ 인 행에 대해 행  $i$ 와
        열  $j$ 를 행렬  $A$ 에서 삭제
```

```
Next
if 해를 얻지 못함 then 1의 개수가 최소인 다른 열  $c$  선택  $A$ 
에 대해 반복 수행.
```

그림 2. Knuth의 알고리즘 X

Fig. 2. Knuth's algorithm X

Knuth 알고리즘으로 완전피복 문제를 풀기 위해서는 모든 가능한 경우수를 고려하여 결정론과 비결정론으로 선택과정을 반복하면서 해를 구한다. 즉, 첫 번째 출발지점인 열을 선택할 때는 1의 개수가 최소인 열을 결정론적으로 선택하지만 이후부터는 비결정론적으로 랜덤하게 선택한다. 이와 같이 하는 이유는 X 의 모든 원소를 피복시켜야 하기 때문에 1의 개수가 최소인 열 x_j 를 선택하면 전수탐색법의 가능한 경우 수를 해당 열의 '1'의 개수로 축소시킬 수 있기 때문이다. Knuth 알고리즘 X는 일

단 $\{S_i, X_j\}$ 가 선택되면 S_i 가 커버하는 모든 X_k 를 커버하는 X_j 들을 삭제해야만 하며 이는 순환 이중연결리스트(circular doubly linked list)가 춤을 추는 방식과 유사하여 DLX(dancing Links)라 명명하였다.^[7-10]

결국, Knuth 알고리즘 X는 전적으로 규칙이 없는 시행착오 법(trial-and-error)이라 할 수 있다. 3장에서는 시행착오 법에서 탈피하여 출발지점을 비롯하여 목표 지점인 해를 얻는 전 과정에서 규칙을 적용하여 움직이는 알고리즘을 제안한다.

III. 원소수 기반 완전 피복 알고리즘

본 장에서 제안되는 완전 피복 알고리즘은 의 부분 집합들 S_i 의 원소 수(cardinality)를 기준으로 항상 결정론적으로 선택한다. 선택 방법은 첫 번째로, 최소 원소 수를 가진 부분집합 S_j 를 선택하고 S_j 의 원소를 포함하는 부분집합 S_i 를 삭제한다. $S = \{\phi\}$ 이 될 때까지 남아 있는 S_i 를 대상으로 이 과정을 반복 수행한다. 만약, 이 방법을 수행하여 $X = \{\phi\} \cap S = \{\phi\}$ 이 되면 정확한 해를 구한다. 만약, 결과가 $X = \{\phi\} \cap S = \{\phi\}$ 가 되지 못하면 두 번째로, 최대 원소 개수를 가진 부분집합 S_j 를 선택하

$X, S_i \subseteq S, S^* = \{\phi\}$.
 $c_i = |S_i|$: S_i 의 원소 수(cardinality).
 $c_{\min} = \min |S_i|$: S 부분집합 중 최소 원소 수 집합 S_i 의 원소 수.
 $c_{\max} = \max |S_i|$: S 부분집합 중 최대 원소 수 집합 S_i 의 원소 수.
 While $X \neq \{\phi\}$ or $S \neq \{\phi\}$
 S 의 모든 부분집합 S_i 들에 대해 c_i 계산.
 $c_{\min} = \min |S_i|$ 인 S_i 선택.
 $S^* \leftarrow S^* \cup \{S_j\}$, $X \leftarrow X \setminus S_j$ /* X 에서 S_j 원소 삭제.
 S_j 원소를 포함한 S 의 부분집합 S_i 삭제($S \leftarrow S \setminus S_i$).
 Next S^* .
 if $X = \{\phi\} \cap S = \{\phi\}$ then 알고리즘 종료
 else
 While $X \neq \{\phi\}$ or $S \neq \{\phi\}$
 S 의 모든 부분집합 S_i 들에 대해 c_i 계산.
 $c_{\max} = \max |S_i|$ 인 S_i 선택.
 $S^* \leftarrow S^* \cup \{S_j\}$, $X \leftarrow X \setminus S_j$ /* X 에서 S_j 원소 삭제.
 S_j 원소를 포함한 S 의 부분집합 S_i 삭제
 ($S \leftarrow S \setminus S_i$).
 Next S^* .
 endif

그림 3. 원소 수 기반 완전 피복 알고리즘
 Fig. 3. Cardinality-based exact cover algorithm

고 위 과정을 반복 수행한다. 이 과정은 최악의 경우 $2^{|S|}$, 최적의 경우 $|S|$ 회 이하를 수행하여 알고리즘 수행 복잡도는 선형인 $O(n)$ 으로 다항시간 알고리즘임을 알 수 있다. 제안된 원소 수 기반 완전 피복 알고리즘(cardinality-based exact cover algorithm, CECA)은 그림 3에 제시되어 있다.

$X = \{A, B, C, D\}$, $S = \{1, 2, 3, 4, 5, 6\}$
 such that 1={A}; 2={B,C}; 3={A,D};
 4={C,D}; 5={B}; 6={C}.

그림 4. 완전 피복 문제 (EC_2)
 Fig. 4. EC_2 exact cover problem

표 1. EC_1 의 원소 수 기반 완전 피복 알고리즘

Table 1. CECA for EC_1

수행 기준	S_j	S^*	S_i	X	S
초기치	ϕ	ϕ	A={1,4,7} B={1,4} C={4,5,7} D={3,5,6} E={2,3,6,7} F={2,7}	{1,2,3,4,5,6,7}	{A,B,C,D,E,F}
$c_{\min} = 2$	B	{B}	D={3,5,6} F={2,7}	{2,3,5,6,7}	{D,E,F}
$c_{\min} = 2$	F	{B,F}	D={3,5,6}	{3,5,6}	{D}
$c_{\min} = 3$	D	{B,F,D}	ϕ	ϕ	ϕ

표 2. EC_2 의 원소 수 기반 완전 피복 알고리즘

Table 2. CECA for EC_2

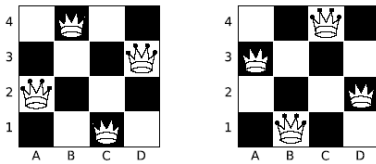
수행 기준	S_j	S^*	S_i	X	S
초기치	ϕ	ϕ	1={A} 2={B,C} 3={A,D} 4={C,D} 5={B} 6={C}	{A,B,C,D}	{1,2,3,4,5,6}
$c_{\min} = 1$	1	{1}	2={B,C} 4={C,D} 5={B} 6={C}	{B,C,D}	{2,4,5,6}
$c_{\min} = 1$	5	{1,5}	4={C,D} 6={C}	{C,D}	{4,6}
$c_{\min} = 1$	6	{1,5,6}	ϕ	{D}	ϕ
				$X \neq \{\phi\}, S = \{\phi\}$	
초기치	ϕ	ϕ	1={A} 2={B,C} 3={A,D} 4={C,D} 5={B} 6={C}	{A,B,C,D}	{1,2,3,4,5,6}
$c_{\max} = 2$	2	{2}	1={A} 3={A,D}	{A,D}	{1,3}
$c_{\max} = 2$	3	{2,3}	ϕ	ϕ	ϕ

그림 1의 EC_1 과 그림 4의 EC_2 문제에 대해 제안된 알고리즘을 적용한 과정은 각각 표 1과 표 2에 제시되어 있다.

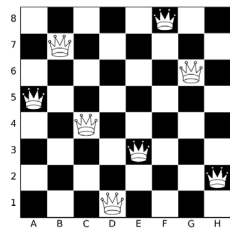
EC_2 는 Tjensvold^[4]에서 인용되었다. EC_1 은 단지 3회 수행으로 $S^* = \{B, D, F\}$ 를 얻었으며, EC_2 는 3회 수행 시 해를 얻지 못해 다시 2회 수행으로 $S^* = \{2, 3\}$ 을 얻었다. 결국, 두 문제에 대해 $|S|$ 이하의 수행 횟수로 해를 얻는데 성공하였다.

IV. 알고리즘 적용 및 결과 분석

완전피복 문제를 보다 일반화한 그림 5의 4-퀸 문제인 EC_3 와 8-퀸 문제인 EC_4 에 대해 제안된 알고리즘의 적용성을 평가해 본다.



(a) 4-퀸 문제 (4×4 체스보드)



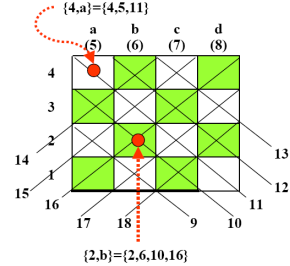
(b) 8-퀸 문제 (8×8 체스보드)

그림 5. 실험 데이터^[4]
Fig. 5. Experimental data

이들 문제는 Tjensvold^[4], Wikipedia^[5]와 Takaoka^[6]로부터 인용되었다. 체스 게임에서 퀸은 행과 열, 대각선으로도 이동이 가능하다. 따라서 퀸들 사이에는 동일한 행, 열과 대각선에 존재하면 상대 퀸의 공격을 받기 때문에 배치될 수 없다. 이 문제를 해결하기 위해 퀸의 위치는 4차원 좌표 $\{x, y, w, z\}$ 로 표현될 수 있다. 4×4 체스보드에 대해 4-퀸을 배열할 수 있는 집합은 그림 6과 같이 주어진다.

S 의 부분집합 S_j 는 4개 모서리 좌표를 제외하고는 모두 4차원의 좌표로 원소 개수는 4이다. 4개 좌표를 대상

으로 첫 번째 자리를 임의로 선택하여 4-퀸을 배열할 수 있는 가능한 해를 구하는 과정은 표 3과 같다. 4-퀸의 2개 대각선의 개수는 4개가 아닌 5개 씩 존재하여 X 의 원소개수는 16개가 아닌 18개이다. 따라서 알고리즘 종료 기준은 $X = \{\phi\} \cap S = \{\phi\}$ 이 아닌 $X = 2 \cap S = \{\phi\}$ 이 된다.



- $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18\}$
 $S = \{S_{15}, S_{16}, S_{17}, S_{18}, S_{25}, S_{26}, S_{27}, S_{28}, S_{35}, S_{36}, S_{37}, S_{38}, S_{45}, S_{46}, S_{47}, S_{48}\}$
 $S_{15} = \{1, 5, 16\}, S_{16} = \{1, 6, 9, 17\},$
 $S_{17} = \{1, 7, 10, 18\}, S_{18} = \{1, 8, 11\},$
 $S_{25} = \{2, 5, 9, 15\}, S_{26} = \{2, 6, 10, 16\},$
 $S_{27} = \{2, 7, 11, 17\}, S_{28} = \{2, 8, 12, 18\},$
 $S_{35} = \{3, 5, 10, 14\}, S_{36} = \{3, 6, 11, 15\},$
 $S_{37} = \{3, 7, 12, 16\}, S_{38} = \{3, 8, 13, 17\},$
 $S_{45} = \{4, 5, 11\}, S_{46} = \{4, 6, 12, 14\},$
 $S_{47} = \{4, 7, 13, 15\}, S_{48} = \{4, 8, 16\}.$

그림 6. 4-퀸 문제의 집합
Fig. 6. Sets of 4-Queen problem

표 3은 $S_{16} = \{1, 6, 9, 17\}$ 을 선택할 경우, $S^* = \{S_{16}, S_{28}, S_{35}, S_{47}\}$ 을 얻은 경우이다. 만약, $S_{17} = \{1, 7, 10, 18\}$ 을 선택할 경우 $S^* = \{S_{17}, S_{25}, S_{38}, S_{46}\}$ 을 얻는다. 이 둘 해는 Tjensvold^[4]이 얻은 그림 5의 (a) 결과와 일치한다.

8×8 체스보드에 대해 8-퀸을 배열할 수 있는 유일한 해는 12개 존재한다. 11개의 각 해에 대해 좌우 대칭으로 배열하면 2개, 90°, 180°, 270°로 회전시키면 6개씩 존재하여 총 8개씩 존재한다. 결국, 11개의 유일한 해에 대해 88개의 해가 존재한다. 또한, 상하 위치를 바꾸어 회전시키면 1개의 유일한 해에 대해 4개의 다른 해를 구할 수 있다. 따라서 총 92개의 해가 존재한다. 12개의 유일한 해의 위치를 표시하면 그림 7과 같으며, 8×8 체스보드에 대해 8-퀸을 배열할 수 있는 집합은 그림 8과 같이 주어진다. 결국, 8-퀸 문제는 모든 X_j 가 동일하여 Knuth 알고리즘 X 를 적용할 경우 출발지점 선택도 결정론적이 아닌 비결정론적으로 선택해야 하므로 완전 비결정론적 방법이라고 할 수 있다.

표 3. EC_3 의 원소 수 기반 완전 피복 알고리즘

Table 3. CECA for EC_3

수행 기준	S_j	S^*	S_i	X	S
초기치	\emptyset	\emptyset	$S_{15} = \{1, 5, 16\}$ $S_{16} = \{1, 6, 9, 17\}$ $S_{17} = \{1, 7, 10, 18\}$ $S_{18} = \{1, 8, 11\}$ $S_{19} = \{2, 5, 9, 15\}$ $S_{20} = \{2, 6, 10, 16\}$ $S_{21} = \{2, 7, 11, 17\}$ $S_{22} = \{2, 8, 12, 18\}$ $S_{23} = \{3, 5, 10, 14\}$ $S_{24} = \{3, 6, 11, 15\}$ $S_{25} = \{3, 7, 12, 16\}$ $S_{26} = \{3, 8, 13, 17\}$ $S_{27} = \{4, 5, 11\}$ $S_{28} = \{4, 6, 12, 14\}$ $S_{29} = \{4, 7, 13, 15\}$ $S_{30} = \{4, 8, 16\}$	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18\}$	$\{S_{15}, S_{16}, S_{17}, S_{18}, S_{19}, S_{20}, S_{21}, S_{22}, S_{23}, S_{24}, S_{25}, S_{26}, S_{27}, S_{28}, S_{29}, S_{30}\}$
$c_{max} = 4$	$\{S_{23}\}$	$\{S_{23}\}$	$S_{23} = \{2, 8, 12, 18\}$ $S_{25} = \{3, 5, 10, 14\}$ $S_{27} = \{3, 7, 12, 16\}$ $S_{29} = \{4, 5, 11\}$ $S_{47} = \{4, 7, 13, 15\}$ $S_{48} = \{4, 8, 16\}$	$\{2, 3, 4, 5, 7, 8, 10, 11, 12, 13, 14, 15, 16, 18\}$	$\{S_{23}, S_{25}, S_{27}, S_{29}, S_{47}, S_{48}\}$
$c_{max} = 4$	$\{S_{23}\}$	$\{S_{23}, S_{25}\}$	$S_{25} = \{3, 5, 10, 14\}$ $S_{45} = \{4, 5, 11\}$ $S_{47} = \{4, 7, 13, 15\}$	$\{3, 4, 5, 7, 10, 11, 13, 14, 15, 16\}$	$\{S_{23}, S_{45}, S_{47}\}$
$c_{max} = 4$	$\{S_{23}\}$	$\{S_{23}, S_{25}, S_{27}\}$	$S_{47} = \{4, 7, 13, 15\}$	$\{4, 7, 11, 13, 15, 16\}$	$\{S_{47}\}$
$c_{max} = 4$	$\{S_{47}\}$	$\{S_{47}, S_{23}, S_{25}, S_{47}\}$	\emptyset	$\{11, 16\}$	\emptyset

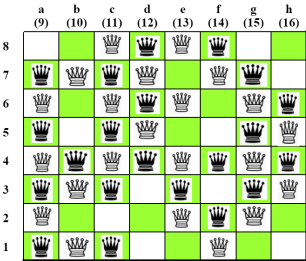
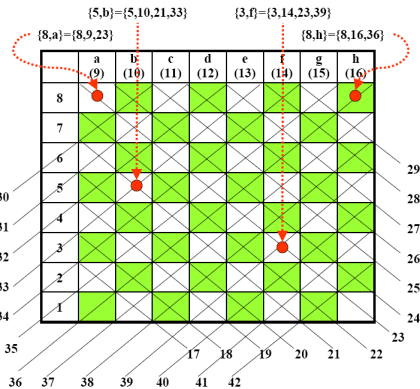


그림 7. 8×8 체스보드의 8-퀸 배열 위치
 Fig. 7. 8-Queen assignment location for 8×8 chessboard



$X = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42\}$
 $S = \{S_{10}, S_{11}, S_{12}, S_{13}, S_{14}, S_{15}, S_{16}, S_{17}, S_{18}, S_{19}, S_{20}, S_{21}, S_{22}, S_{23}, S_{24}, S_{25}, S_{26}, S_{27}, S_{28}, S_{29}, S_{30}, S_{31}, S_{32}, S_{33}, S_{34}, S_{35}, S_{36}, S_{37}, S_{38}, S_{39}, S_{40}, S_{41}, S_{42}, S_{43}, S_{44}, S_{45}, S_{46}, S_{47}, S_{48}, S_{49}, S_{50}, S_{51}, S_{52}, S_{53}, S_{54}, S_{55}, S_{56}, S_{57}, S_{58}, S_{59}, S_{60}, S_{61}, S_{62}, S_{63}, S_{64}, S_{65}, S_{66}, S_{67}, S_{68}, S_{69}, S_{70}, S_{71}, S_{72}, S_{73}, S_{74}, S_{75}, S_{76}, S_{77}, S_{78}, S_{79}, S_{80}, S_{81}, S_{82}, S_{83}, S_{84}, S_{85}, S_{86}, S_{87}, S_{88}, S_{89}, S_{90}\}$
 $S_{10} = \{1, 9, 36\}$, $S_{11} = \{1, 10, 17, 37\}$, $S_{12} = \{1, 11, 18, 38\}$,
 $S_{13} = \{1, 12, 19, 39\}$, $S_{14} = \{1, 13, 20, 40\}$, $S_{15} = \{1, 14, 21, 41\}$,
 $S_{16} = \{1, 15, 22, 42\}$, $S_{17} = \{1, 16, 23\}$,
 $S_{20} = \{2, 9, 17, 35\}$, $S_{21} = \{2, 10, 18, 36\}$, $S_{22} = \{2, 11, 19, 37\}$,
 $S_{23} = \{2, 12, 20, 38\}$, $S_{24} = \{2, 13, 21, 39\}$, $S_{25} = \{2, 14, 22, 40\}$,
 $S_{26} = \{2, 15, 23, 41\}$, $S_{27} = \{2, 16, 24, 42\}$,
 $S_{30} = \{3, 9, 18, 34\}$, $S_{31} = \{3, 10, 19, 35\}$, $S_{32} = \{3, 11, 20, 36\}$,
 $S_{33} = \{3, 12, 21, 37\}$, $S_{34} = \{3, 13, 22, 38\}$, $S_{35} = \{3, 14, 23, 39\}$,
 $S_{36} = \{3, 15, 24, 40\}$, $S_{37} = \{3, 16, 25, 41\}$,
 $S_{40} = \{4, 9, 19, 33\}$, $S_{41} = \{4, 10, 20, 34\}$, $S_{42} = \{4, 11, 21, 35\}$,
 $S_{43} = \{4, 12, 22, 36\}$, $S_{44} = \{4, 13, 23, 37\}$, $S_{45} = \{4, 14, 24, 38\}$,
 $S_{46} = \{4, 15, 25, 39\}$, $S_{47} = \{4, 16, 26, 40\}$,
 $S_{50} = \{5, 9, 20, 32\}$, $S_{51} = \{5, 10, 21, 33\}$, $S_{52} = \{5, 11, 22, 34\}$,
 $S_{53} = \{5, 12, 23, 35\}$, $S_{54} = \{5, 13, 24, 36\}$, $S_{55} = \{5, 14, 25, 37\}$,
 $S_{56} = \{5, 15, 26, 38\}$, $S_{57} = \{5, 16, 27, 39\}$,
 $S_{60} = \{6, 9, 21, 31\}$, $S_{61} = \{6, 10, 22, 32\}$, $S_{62} = \{6, 11, 23, 33\}$,
 $S_{63} = \{6, 12, 24, 34\}$, $S_{64} = \{6, 13, 25, 35\}$, $S_{65} = \{6, 14, 26, 36\}$,
 $S_{66} = \{6, 15, 27, 37\}$, $S_{67} = \{6, 16, 28, 38\}$,
 $S_{70} = \{7, 9, 22, 30\}$, $S_{71} = \{7, 10, 23, 31\}$, $S_{72} = \{7, 11, 24, 32\}$,
 $S_{73} = \{7, 12, 25, 33\}$, $S_{74} = \{7, 13, 26, 34\}$, $S_{75} = \{7, 14, 27, 35\}$,
 $S_{76} = \{7, 15, 28, 36\}$, $S_{77} = \{7, 16, 29, 37\}$,
 $S_{80} = \{8, 9, 23\}$, $S_{81} = \{8, 10, 24, 30\}$, $S_{82} = \{8, 11, 25, 31\}$,
 $S_{83} = \{8, 12, 26, 32\}$, $S_{84} = \{8, 13, 27, 33\}$, $S_{85} = \{8, 14, 28, 34\}$,
 $S_{86} = \{8, 15, 29, 35\}$, $S_{87} = \{8, 16, 36\}$.

그림 8. 8-퀸 문제의 집합
 Fig. 8. Set of 8-Queen problem

그림 7에서 배치될 수 없는 자리의 공통점은 29와 42가 없다. 따라서 이 값을 갖고 있는 집합 $S_{1,d}, S_{2,h}, S_{7,h}, S_{8,i}$ 를 제외하면 다음과 같이 다시 얻을 수 있다.

$$\begin{aligned}
 X &= \{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, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41\} \\
 S &= \{S_{1a}, S_{1b}, S_{1c}, S_{1d}, S_{1e}, S_{1f}, S_{1g}, S_{1h}, S_{2a}, S_{2b}, S_{2c}, S_{2d}, S_{2e}, S_{2f}, S_{2g}, S_{2h}, S_{3a}, S_{3b}, S_{3c}, S_{3d}, \\
 &S_{3e}, S_{3f}, S_{3g}, S_{3h}, S_{4a}, S_{4b}, S_{4c}, S_{4d}, S_{4e}, S_{4f}, S_{4g}, S_{4h}, S_{5a}, S_{5b}, S_{5c}, S_{5d}, S_{5e}, S_{5f}, S_{5g}, S_{5h}, \\
 &S_{6a}, S_{6b}, S_{6c}, S_{6d}, S_{6e}, S_{6f}, S_{6g}, S_{6h}, S_{7a}, S_{7b}, S_{7c}, S_{7d}, S_{7e}, S_{7f}, S_{7g}, S_{7h}, \\
 &S_{8a}, S_{8b}, S_{8c}, S_{8d}, S_{8e}, S_{8f}, S_{8g}, S_{8h}\} \\
 S_{1a} &= \{1, 9, 36\}, S_{1b} = \{1, 10, 17, 37\}, S_{1c} = \{1, 11, 18, 38\}, \\
 S_{1d} &= \{1, 12, 19, 39\}, S_{1e} = \{1, 13, 20, 40\}, S_{1f} = \{1, 14, 21, 41\}, \\
 S_{1g} &= \{1, 16, 23\}, \\
 S_{2a} &= \{2, 9, 17, 35\}, S_{2b} = \{2, 10, 18, 36\}, S_{2c} = \{2, 11, 19, 37\}, \\
 S_{2d} &= \{2, 12, 20, 38\}, S_{2e} = \{2, 13, 21, 39\}, S_{2f} = \{2, 14, 22, 40\}, \\
 S_{2g} &= \{2, 15, 23, 41\}, \\
 S_{3a} &= \{3, 9, 18, 34\}, S_{3b} = \{3, 10, 19, 35\}, S_{3c} = \{3, 11, 20, 36\}, \\
 S_{3d} &= \{3, 12, 21, 37\}, S_{3e} = \{3, 13, 22, 38\}, S_{3f} = \{3, 14, 23, 39\}, \\
 S_{3g} &= \{3, 15, 24, 40\}, S_{3h} = \{3, 16, 25, 41\}, \\
 S_{4a} &= \{4, 9, 19, 33\}, S_{4b} = \{4, 10, 20, 34\}, S_{4c} = \{4, 11, 21, 35\}, \\
 S_{4d} &= \{4, 12, 22, 36\}, S_{4e} = \{4, 13, 23, 37\}, S_{4f} = \{4, 14, 24, 38\}, \\
 S_{4g} &= \{4, 15, 25, 39\}, S_{4h} = \{4, 16, 26, 40\}, \\
 S_{5a} &= \{5, 9, 20, 32\}, S_{5b} = \{5, 10, 21, 33\}, S_{5c} = \{5, 11, 22, 34\}, \\
 S_{5d} &= \{5, 12, 23, 35\}, S_{5e} = \{5, 13, 24, 36\}, S_{5f} = \{5, 14, 25, 37\}, \\
 S_{5g} &= \{5, 15, 26, 38\}, S_{5h} = \{5, 16, 27, 39\}, \\
 S_{6a} &= \{6, 9, 21, 31\}, S_{6b} = \{6, 10, 22, 32\}, S_{6c} = \{6, 11, 23, 33\}, \\
 S_{6d} &= \{6, 12, 24, 34\}, S_{6e} = \{6, 13, 25, 35\}, S_{6f} = \{6, 14, 26, 36\}, \\
 S_{6g} &= \{6, 15, 27, 37\}, S_{6h} = \{6, 16, 28, 38\}, \\
 S_{7a} &= \{7, 9, 22, 30\}, S_{7b} = \{7, 10, 23, 31\}, S_{7c} = \{7, 11, 24, 32\}, \\
 S_{7d} &= \{7, 12, 25, 33\}, S_{7e} = \{7, 13, 26, 34\}, S_{7f} = \{7, 14, 27, 35\}, \\
 S_{7g} &= \{7, 15, 28, 36\}, \\
 S_{8a} &= \{8, 9, 23\}, S_{8b} = \{8, 10, 24, 30\}, S_{8c} = \{8, 11, 25, 31\}, \\
 S_{8d} &= \{8, 12, 26, 32\}, S_{8e} = \{8, 13, 27, 33\}, S_{8f} = \{8, 14, 28, 34\}, \\
 S_{8g} &= \{8, 16, 36\}.
 \end{aligned}$$

이들 주어진 집합을 대상으로 임의의 셀을 선택하여 8-퀸을 배열할 수 있는 방법을 찾으면 된다. 알고리즘 종료 기준은 8개의 퀸 셀의 좌표 값의 합 32개를 제거하였을 경우 $X=8 \cap S = \{\phi\}$, 3개 좌표 값 셀이 선택될 경우 $X=9 \cap S = \{\phi\}$ 가 된다. 임의의 셀 S_{1f} 을 선택하여 8-퀸을 배열할 수 있는 해는 $S^* = \{S_{1f}, S_{2a}, S_{3e}, S_{4d}, S_{5h}, S_{6c}, S_{7a}, S_{8d}\}$ 이다. 동일 방법으로 Wikipedia^[5]가 제시한 12개의 유일한 해를 찾을 수 있을 것이다. 또한, 또 다른 유일한 해가 존재하는지 검증하기 위해서는 그림 6에서 퀸이 배치되지 않은 모든 셀을 대상으로 8-퀸을 배치할 수 있는지 제안된 알고리즘을 적용하여 검증할 수 있다. 예로, S_{8a}, S_{8h} 와 S_{8b}, S_{8g} 는 상호 대칭되는데도 불구하고 퀸이 배치될 수 없다. 만약, S_{8a} 또는 S_{8b} 를 선택하여 제안된 알고리즘으로 8-퀸이 배치될 수 있는지 검증 가능하다. 제안된 알고리즘을 4-퀸에 적용한 결과 해를 쉽게 구하는데 성공하였다. 또한, 이를 확장하여 8-퀸에 적용할 수 있는 방법을 제안하였다. 결국, 완전 피복 문제는 NP-완전으로 근사 알고리즘만이 존재하는 것이 아니라 P-문제로 다항시간 알고리즘이 존재함을 알 수 있다.

V. 결론 및 향후 연구과제

완전피복 문제는 지금까지 정확한 해법을 다항시간에

찾는 알고리즘이 존재하지 않는 NP-완전 문제로 알려져 근사 알고리즘만을 연구하였다. 본 논문은 각 행과 열 (N-퀸 문제의 경우 대각선까지 고려)에서 중복되지 않게 반드시 1개씩만 존재해야 하는 완전피복 문제의 특징을 이용하여 특정 부분집합을 선택(최소 원소 개수 또는 최대 원소 개수 부분집합)하였을 경우 선택된 부분집합의 원소를 포함하는 다른 부분집합을 삭제하고 남은 부분 집합들 중에서 다시 선택하는 방법으로 완전피복 문제의 해를 선형시간에 구하는 알고리즘을 제안하였다.

제안된 알고리즘을 완전피복 문제를 보다 일반화한 4-퀸과 8-퀸 문제에 적용 가능성을 보였다. 추후 본 알고리즘을 펜토미노 타일 붙이기와 숫자 넣기 문제에 적용할 수 있는지 연구할 계획이다.

References

- [1] Wikipedia, "Exact Cover," http://en.wikipedia.org/wiki/Exact_cover, Retrieved Apr. 2022.
- [2] Wikipedia, "Knuth's Algorithm X," http://en.wikipedia.org/wiki/Knuth's_algorithm, Retrieved Apr. 2022.
- [3] K. Y. Yun, "ECE 30: Introduction to Computer Engineering," Dept. of Electrical and Computer Eng., University of California, San Diego, 2007.
- [4] J. M. Tjensvold, "Genetic Distributed Exact Cover Solver," <http://decs.googlecode.com/decs/downloads/list/decs-report.pdf>, 2007.
- [5] Wikipedia, "Eight Queens Puzzle," http://en.wikipedia.org/wiki/Eight_queens_puzzle, Retrieved Apr. 2022.
- [6] T. Takaoka, "Solution to the 8-Queens Problem," <http://bridges.canterbury.ac.nz/features/eight.html>, 1998.
- [7] D. E. Knuth, "Dancing Links," The Art of Computer Programming, Vol. 4, Pre-Fascicle 5c, pp. 1-270, Addison-Wesley, Jul. 2019.
- [8] D. E. Knuth, "Dancing Links," Proceedings of the 1999 Oxford-Microsoft Symposium in Honour of Sir Tony Hoare, pp. 187-214, arXiv : cs/0011047, pp. 1-26, Nov. 2000.
- [9] M. Nishino, N. Yasuda, S. I. Minato, and M. Nagata, "Dancing with Decision Diagrams: A Combined Approach to Exact Cover," Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, Vol. 31, No. 1, pp. 868-874, Feb. 2017.
- [10] M. Nishino, N. Yasuda, S. I. Minato, and M. Nagata, "Efficient Algorithm for Enumerating All Solutions to an Exact Cover Problem," NTT Technical Review, Vol.

15, No. 11, pp. 1-5, Nov. 2017.

- [11] M. Chabert and C. Solnon, "A Global Constraints for the Exact Cover Problem: Application to Conceptual Clustering," *Journal of Artificial Intelligence Research*, Vol. 67, pp. 509-547, hal-02507186, 2020, <https://doi.org/10.1613/jair.1.11870>
- [12] M. Nishino, N. Yasuda, and K. Nakamura, "Compressing Exact Cover Problems with Zero-Suppressed Binary Decision Diagrams," *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 1996-2004, Aug. 2021, <https://doi.org/10.24963/ijcai.2021/275>

저 자 소 개

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
- 관심분야 : 소프트웨어 프로젝트 관리, 개발 방법론, 분석과 설계 방법론, 시험 및 품질보증, 소프트웨어 신뢰성, 인공지능과 빅데이터분석, 최적화 알고리즘
- e-mail : sulee@gwnu.ac.kr