

<https://doi.org/10.7236/JIIBC.2023.23.1.157>

JIIBC 2023-1-23

최소 빈도수 문자 우선 선택 방법의 3-SAT 다항시간 알고리즘

A 3-SAT Polynomial Time Algorithm Based on Minimum Frequency Literal-First Selection Method

이상운*

Sang-Un Lee*

요약 본 논문은 NP-완전으로 알려진 3-SAT 문제에 대해 $O(nm)$ 의 다항시간 알고리즘을 제안하였다. 여기서 n 은 3-SAT 수식 f 에 포함된 문자수이며, m 은 모든 문자의 총 발생 빈도수이다. 제안된 알고리즘은 우선순위를 정하여 문자의 진리 값을 결정하였다. 하나의 절에 포함된 최소 문자의 수가 1개($k=1$)인 문자, $k \geq 2$ 이고 진리 값이 '0' 또는 '1'만 존재하는 문자, 발생 빈도수가 최소인 문자 우선순위로 진리 값을 결정하였다. 진리 값이 결정된 문자의 T 인 절을 삭제하고, 남아 있는 절에서 해당 문자를 삭제하는 방법으로 식을 축소시킨다. 이 과정을 문자 수 l 회를 수행하면 주어진 f 의 충족 가능 여부를 결정할 수 있다. 이 방법을 적용한 결과 주어진 식 f 의 충족 가능 여부를 판단할 수 있었으며, 모든 문자의 진리 값도 결정할 수 있었다. 제안된 알고리즘을 다양한 문제들에 적용한 결과 모든 문제들에서 충족가능 여부를 정확히 결정할 수 있었다. 결국, 제안된 알고리즘은 모든 SAT 문제를 선형시간으로 풀 수 있는 일반화된 알고리즘으로 확장 가능하며, 오랜 숙원이던 P vs. NP 문제를 풀 수 있는 기반을 제공하였다.

Abstract To NP-complete 3-SAT problem, this paper proposes a $O(nm)$ polynomial time algorithm, where n is the number of literals and m is the total frequency of all literals in equation f . The algorithm firstly decides a truth value of a literal in sequence of previously-set priority. The priority order is as follows: a literal whose occurrence in a clause is 1 ($k=1$), a literal which is $k \geq 2$ and whose truth value is either 0 or 1, and a literal with the minimum frequency. Then, literals whose truth value is determined are then deleted from clause T and the remaining clauses. This process is repeated l times, the number of literals. As a result, the proposed algorithm has been successful in accurately determining the satisfiability of a given equation f and in deciding the truth value of all the literals. This paper, therefore, provides not only a linear-time algorithm as a viable solution to the SAT problem, but also a basis for solving the P versus NP problem.

Key Words : Boolean satisfiability problem(SAT), Priority, Frequency, Contraction

*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과
접수일자 2022년 10월 22일, 수정완료 2023년 1월 23일
게재확정일자 2023년 2월 3일

Received: 22 October, 2022 / Revised: 23 January, 2023 /
Accepted: 3 February, 2023

*Corresponding Author: sulee@gwnu.ac.kr

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

1. 서 론

3-충족 가능성 문제(3-SAT, Boolean satisfiability problem)는 참(true, T) 또는 거짓(false, F)의 진리 값(true value)을 가질 수 있는 3개의 문자(literals, l) 또는 변수(variables) p, q, r 이 논리합(disjunction, OR, \vee)으로 연결된 절(clause, c) 몇 개가 논리곱(conjunction, AND, \wedge)으로 구성된 부울 수식 f 가 참이 되도록 문자의 진리 값을 구할 수 있는가의 문제이다.^[1,2] 즉, 수식 f 가 충족 가능(satisfiable)이란 수식 f 가 참이 되기 위해 모든 절이 참이 되도록 문자의 진리 값을 얻어야 하며, 모든 가능한 참인 경우 중 한 가지 이상만 얻으면 된다. 반면에, 가능한 모든 문자의 진리 값에 대해 수식 f 가 참이 되지 못하면 즉, 수식이 성립하지 않으면 충족 불가능(unsatisfiable)이라 한다. 문자수를 n , 절수를 c , 절 당 문자수를 k 라 하면 $f = (p \vee \bar{q} \vee \bar{r}) \wedge (p \vee q \vee s)$ 는 $n = 4, c = 2, k = 3$ 이다.^[1,2]

SAT 문제는 명제논리, 데이터베이스 설계 및 분석, 자동화된 추론, 기계가 물체를 시각적으로 인식하는 Machine Vision, CAD(Computer Aided Design), 로보틱스, 일정, 집적회로 설계 등의 분야에 적용된다.^[3]

오늘날 사용되고 있는 가장 효율적인 모든 SAT 해결사(SAT solver)는 DL(Davis-Putnam)^[4]과 DPLL(Davis-Putnam-Logemann-Loveland) 알고리즘^[5]에 근원을 두고 있다.^[6] 3-SAT 문제는 Cook^[7]이 1971년에 NP-완전(NP-complete)으로 처음 제시하고, Karp^[8,9]이 1972년에 21개 NP-완전 문제 중 첫 번째 문제로 제시한 이후 지금까지 정확한 해를 다항시간으로 얻지 못하고 있다. 2008년에 Gubin^[10,11]은 $O(c^3)$ 의 다항시간 알고리즘을 제안하였다.

본 논문은 $O(nm)$ 의 다항시간 알고리즘을 제안한다. 여기서 m 은 수식 f 에 포함된 모든 문자의 총 빈도수이다. 2장에서는 $O(2^n)$ 의 전수검사법, DPLL^[5]과 Gubin^[10,11]의 $O(c^3)$ 다항시간 알고리즘을 고찰해 본다. 3장에서는 $O(nm)$ 의 다항시간 알고리즘을 제안한다. 4장에서는 Gubin^[11]이 제시한 사례들을 포함한 다양한 문제들을 대상으로 제안된 알고리즘을 증명하여 본다.

II. 관련연구와 연구 배경

식 (1)은 Cook^[7]이 제시한 3-SAT 문제로 $n = 4, c = 4, k = 3$ 이며, $pqr s = TFFF$ (또는 1100)의 진리 값을 가질

경우 $f = T$ 를 만족하여 충족 가능이라 하였다.

$$f_1 = (p \vee q \vee r) \wedge (\bar{p} \vee q \vee \bar{r}) \wedge (p \vee \bar{q} \vee s) \wedge (\bar{p} \vee \bar{r} \vee \bar{s}) \quad (1)$$

식 (1)의 정확한 해를 얻기 위해서는 표 1과 같이 2^n 의 가능한 모든 경우 수에 대해 f 의 진리 값을 결정해야 한다. 식 (1)은 16가지 경우 수 중에서 $pqr s = FFFF, FFFT, FTFT, FTTF, TFFF, TFFT, TTF, TTTF$ 의 9가지에서 참이 성립한다.

표 1. 식 (1)의 진리표

Table 1. Truth table for Expression (1)

p	q	r	s	$(p \vee q \vee r)$	$(\bar{p} \vee q \vee \bar{r})$	$(p \vee \bar{q} \vee s)$	$(\bar{p} \vee \bar{r} \vee \bar{s})$	f
0	0	0	0	0+0+0=0	1+0+1=1	0+1+0=1	1+1+1=1	0·1·1·1=0
0	0	0	1	0+0+0=0	1+0+1=1	0+1+1=1	1+1+0=1	0·1·1·1=0
0	0	1	0	0+0+1=1	1+0+0=1	0+1+0=1	1+0+1=1	1·1·1·1=1
0	0	1	1	0+0+1=1	1+0+0=1	0+1+1=1	1+0+0=1	1·1·1·1=1
0	1	0	0	0+1+0=1	1+1+1=1	0+0+0=0	1+1+1=1	1·1·0·1=0
0	1	0	1	0+1+0=1	1+1+1=1	0+0+1=1	1+1+0=1	1·1·1·1=1
0	1	1	0	0+1+1=1	1+1+0=1	0+0+0=0	1+0+1=1	1·1·0·1=0
0	1	1	1	0+1+1=1	1+1+0=1	0+0+1=1	1+0+0=1	1·1·1·1=1
1	0	0	0	1+0+0=1	0+0+1=1	1+1+0=1	1+1+1=1	1·1·1·1=1
1	0	0	1	1+0+0=1	0+0+1=1	1+1+1=1	1+1+0=1	1·1·1·1=1
1	0	1	0	1+0+1=1	0+0+0=0	1+1+1=1	0+0+1=1	1·1·1·1=1
1	0	1	1	1+0+1=1	0+0+0=0	1+1+1=1	0+0+1=1	1·1·1·1=1
1	1	0	0	1+1+0=1	0+1+1=1	1+0+0=1	1+1+1=1	1·1·1·1=1
1	1	0	1	1+1+0=1	0+1+1=1	1+0+1=1	1+1+0=1	1·1·1·1=1
1	1	1	0	1+1+1=1	0+1+0=1	1+0+0=1	0+0+1=1	1·1·1·1=1
1	1	1	1	1+1+1=1	0+1+0=1	1+0+1=1	0+0+0=0	1·1·1·0=0

DPLL 알고리즘^[5,6]은 기본적으로 단위 절 제거(unit clause elimination) 규칙과 순수 문자 제거(pure literal elimination) 규칙을 적용하였다. 단위 절이란 1개 문자만 포함하고 있는 절이다. 순수 문자란 f 에 문자 a 는 존재하지만 \bar{a} 가 존재하지 않는 경우이다. 따라서 단위 절이나 순수 문자가 존재할 경우, 이 문자를 '1'로 하면 해당 절을 삭제할 수 있다. 또한, 부울 수식이 '충족 불가능'으로 판단되면 지금까지 선택된 문자의 진리 값을 반대로 적용하면서 충족 여부를 판단하는 역추적법(backtracking)을 적용한다. 역추적 개념은 하나의 문자를 선택하여 참 값을 할당하고 수식을 단순화시킨다. 단순화된 수식에 대해 충족가능 여부를 반복적으로 확인한다. 만약, 충족 불가능이면 반대되는 진리 값을 할당하면서 동일한 반복 확인 과정을 거친다. DPLL 알고리즘은 다음과 같이 수행된다.

- (1) 부울 수식을 CNF(conjunctive normal form)로 변환시킨다.
- (2) 단위 절 또는 순수 문자가 존재하면 해당 문자 x 를 우선 선택한다. 만약, 그렇지 않으면 임의의 문자 x 를 선택한다. 만약, 선택된 문자가 x 이면 $x = 1, \bar{x} = 0$ 을, \bar{x} 이면 $x = 0, \bar{x} = 1$ 을 할당한다. 단순화 과정을 수행한다. 부울 수식 f 의 충족 가능성 여부를 판단한다.
- (3) 만약, $f =$ 충족 불가능이 결정되면 현재 할당된 문자

의 진리 값을 반대로 적용하여 단순화 과정을 다시 수행한다. 그래도 $f =$ 충족 불가능이면 현재까지 선택된 문자들을 역으로 추적하면서 진리 값을 반대로 설정하면서 단순화 과정을 반복적으로 수행한다.

[단순화 과정]

- (1)'1'값이 할당된 문자를 포함하고 있는 절 c_i 를 삭제한다.
- (2)'0'값이 할당된 문자를 포함하고 있는 절 c_j 에 해당 문자를 삭제한다.

f_1 에 DPLL 알고리즘을 적용하여 보자. 단위 절 또는 순수 문자가 존재하면 우선 선택 법칙을 적용하고, 존재하지 않으면 첫 번째 절의 첫 번째에 존재하는 문자를 선택하는 방법을 적용하기로 하자. f_1 은 $k=3$ 인 3-SAT 문제로 단위 절은 존재하지 않는다. 또한 p, q, r, s 의 4개 문자 모두 순수 문자는 존재하지 않는다. 따라서 첫 번째로 p 를 선택하면 $p=1, \bar{p}=0$ 이 할당되어 $f_1 = (q\vee\bar{r}) \wedge (\bar{r}\vee\bar{s})$ 로 축소된다. 두 번째로 이 수식에는 순수 문자 q 가 존재한다. 따라서 순수 문자 q 를 선택하면 $q=1$ 이 할당되어 $f_1 = (\bar{r}\vee\bar{s})$ 가 된다. 세 번째로 2개의 순수 문자가 존재하며, \bar{r} 를 선택하고 $\bar{r}=1$ 이 할당되면 $f_1 = (\bar{s})$ 가 된다. 마지막으로 $\bar{s}=1$ 이 할당되면 $f_1 = \phi$ 이 되어 충족 가능하다. 결국, 4회가 수행된다.

식 (2)는 Boulé^[12]가 제시한 “충족 가능” 문제이다.

$$f_2 = (p\vee q\vee r) \wedge (p\vee q\vee \bar{r}) \wedge (\bar{p}\vee q\vee \bar{r}) \wedge (p\vee r\vee s) \wedge (\bar{p}\vee r\vee s) \wedge (\bar{p}\vee r\vee \bar{s}) \wedge (\bar{q}\vee \bar{r}\vee \bar{s}) \wedge (\bar{q}\vee \bar{r}\vee s) \quad (2)$$

표 2. f_2 의 DPLL 알고리즘 적용 결과

Table 2. Applied result of DPLL algorithm for f_2

수행횟수 (n)	문자 선택 (값 할당)	부울 수식
1	$\frac{p}{p=1, \bar{p}=0}$	$f_2 = (q\vee\bar{r}) \wedge (r\vee s) \wedge (\bar{r}\vee\bar{s}) \wedge (\bar{q}\vee\bar{r}\vee s) \wedge (\bar{q}\vee\bar{r}\vee\bar{s})$
2	$\frac{q}{q=1, \bar{q}=0}$	$f_2 = (r\vee s) \wedge (\bar{r}\vee\bar{s}) \wedge (\bar{r}\vee\bar{s}) \wedge (\bar{r}\vee s)$
3	$\frac{r}{r=1, \bar{r}=0}$	$f_2 = (\bar{s}) \wedge (s)$: 충족 불가능
4	$\frac{r}{r=0, \bar{r}=1}$	$f_2 = (\bar{s}) \wedge (s)$: 충족 불가능
5	$\frac{q}{q=0, \bar{q}=1}$	$f_2 = (\bar{r}) \wedge (r\vee s) \wedge (r\vee\bar{s})$
6	$\frac{r}{r=0, \bar{r}=1}$	$f_2 = (s) \wedge (\bar{s})$: 충족 불가능
7	$\frac{p}{p=0, \bar{p}=1}$	$f_2 = (q\vee r) \wedge (q\vee\bar{r}) \wedge (r\vee s) \wedge (\bar{q}\vee\bar{r}\vee\bar{s}) \wedge (\bar{q}\vee\bar{r}\vee s)$
8	$\frac{q}{q=1, \bar{q}=0}$	$f_2 = (r\vee s) \wedge (\bar{r}\vee\bar{s}) \wedge (\bar{r}\vee s)$
9	$\frac{r}{r=1, \bar{r}=0}$	$f_2 = (\bar{s}) \wedge (s)$: 충족 불가능
10	$\frac{r}{r=0, \bar{r}=1}$	$f_2 = (s)$
11	$\frac{s}{s, \bar{s}=1}$	$f_2 = 1$: 충족 가능

f_2 에 DPLL 알고리즘을 적용하면 $p \rightarrow q \rightarrow r$ (UNSAT) (UNSAT) $\rightarrow \bar{q} \rightarrow \bar{r}$ (UNSAT) $\rightarrow \bar{p} \rightarrow q \rightarrow r$ (UNSAT) $\rightarrow \bar{r} \rightarrow s$ (SAT) 순서로 역 추적 과정을 3회 수행하여 총 11회에 “충족 가능”임을 판단할 수 있다. 이에 대한 수행과정은 표 2에 제시되어 있다.

Gubin^[10,11]은 식 (1)의 해를 구하기 위해 다음과 같은 방법을 적용하였다.

- (1) 진리표(truth-tables)에 기반하여 상삼각의 절 호환성 행렬(clauses' compatibility matrices) $c_i : c_j$ 를 작성한다.

$c_1 : c_2$	$c_1 : c_3$	$c_1 : c_4$
	$c_2 : c_3$	$c_2 : c_4$
		$c_3 : c_4$

- (2) 절 호환성 행렬 $c_i : c_j, i \geq 2$ 에 대해 2번째와 3번째 행을 축소시킨다.

$c_1 : c_2$	$c_1 : c_3$	$c_1 : c_4$
	$c_1 \wedge c_2 : c_1 \wedge c_3$	$c_1 \wedge c_2 : c_1 \wedge c_4$
		$c_1 \wedge c_3 : c_1 \wedge c_4$

- (3) 절 호환성 행렬 $c_i : c_j, i \geq 2$ 에 대해 3번째 행을 축소시킨다.

$c_1 : c_2$	$c_1 : c_3$	$c_1 : c_4$
	$c_1 \wedge c_2 : c_1 \wedge c_3$	$c_1 \wedge c_2 : c_1 \wedge c_4$
		$c_1 \wedge c_2 \wedge c_3 : c_1 \wedge c_2 \wedge c_4$

- (4) 어떤 하나 이상의 행렬에서 “1”을 전혀 포함하고 있지 않으면 충족 불가능이며, 그렇지 않으면 충족 가능하다.

이 방법은 호환성 행렬을 작성하는 과정이 매우 난해하며, 매우 복잡한 행 축소 계산 과정을 거쳐야 한다. 식 (1)에 대해 해를 구하는 상세한 과정은 Gubin[11]을 참고하기 바란다.

3장에서는 $O(nm)$ 의 다항시간으로 해를 구하는 알고리즘을 제안한다.

III. 3-SAT 선형 알고리즘

편의상 주어진 수식 f 의 모든 절에 존재하는 문자 p 를 ‘1’, \bar{p} 를 ‘0’이라 표기한다. 제안된 알고리즘은 우선순위(priority)를 정하여 문자의 진리 값을 결정하였다. 첫 번째 우선순위는 하나의 절에 포함된 최소 문자의 수가 1

개 ($k=1$)인 단위 절 문자의 진리 값을 결정한다. 두 번째는 $k \geq 2$ 이고 진리 값이 '0' 또는 '1'만 존재하는 순수 문자의 진리 값을 결정한다. 마지막으로 발생 빈도수가 최소인 문자의 진리 값을 결정한다. 따라서 DPLL의 문자 선택 방법 중에서 단위 문자 또는 순수 문자 선택 방법을 적용한다. 그러나 임의의 문자 선택 방법 대신 최소 빈도수 문자를 선택하는 방법을 적용한다. 또한 DPLL 알고리즘은 '충족 불가능'으로 판단 시 역추적 기법을 적용하는데 비해 제안된 알고리즘은 역추적 기법을 적용하지 않는 차이점이 있다.

진리 값이 결정된 문자의 진리 값이 'T'인 절을 삭제한다. 예로, 만약 $p=F$ 이면 \bar{p} 를 포함한 절을, $p=T$ 이면 p 를 포함한 절을 삭제한다. 또한, 남아 있는 절에서 해당 문자의 'F'를 삭제한다. 예로, $p=F$ 이면 p 를, $p=T$ 이면 \bar{p} 를 삭제한다. 이와 같이 하여 축소된 수식을 위의 방법을 n 회 반복 적용하면 모든 문자의 진리 값을 결정할 수 있으며, f 의 충족 가능 여부를 판단할 수 있다. 제안된 알고리즘의 상세한 내용은 그림 1과 같다.

```

While  $i=1$  to  $n$  do /*  $O(n)$ 
    각 문자의 발생 빈도수 계산. /*  $O(m)$ 
    절의 문자수  $k$  계산.
    각 문자의 진리값  $\{0,1\}$  확인.
    if  $\exists k=1$  then /* 단위 절
        if  $\exists (p) \wedge (\bar{p})$  then  $f=F$  /* 충족 불가능
            알고리즘 종료
        else if  $p=\{0\}$  then  $p=F$  /*  $\bar{p}$  존재
        else if  $p=\{1\}$  then  $p=T$  /*  $p$  존재
    else if  $\forall k \geq 2$  and  $\exists \{0\}$  or  $\{1\}$  then /* 순수 문자
        if  $p=\{0\}$  then  $p=F$ 
        else if  $p=\{1\}$  then  $p=T$ 
    else if  $\forall k \geq 2$  and  $\forall \{0,1\}$  then /* 최소 빈도수 문자
        최소 발생 빈도수 문자  $p=T$ .
    T 진리값 문자 포함 절 삭제.
    남은 절에서 F 진리값 문자 삭제.
    진리값이 미결정된 문자는 don't care.
Return

```

그림 1. SAT 알고리즘
Fig. 1. SAT algorithm

제안된 알고리즘을 적용하여 Gubin^[11]이 제시한 식 (3)의 Exact 3-SAT 해를 구하여 보자.

$$f_3 = (p \vee q \vee r) \wedge (p \vee q \vee \bar{r}) \wedge (p \vee \bar{q} \vee r) \wedge (p \vee \bar{q} \vee \bar{r}) \wedge (\bar{p} \vee q \vee r) \wedge (\bar{p} \vee q \vee \bar{r}) \wedge (\bar{p} \vee \bar{q} \vee r) \quad (3)$$

$k=3, p=q=r=\{0,1\}$, 발생 빈도수 $p(7), q(7), r(7)$ 로 모두 동일하여 $p=T$ 로 결정하고, p 가 포함된 절을 삭제하고, 나머지 절에서 \bar{p} 를 삭제한다. 왜냐하면 $\bar{p}=F$ 는 해당 절을 T 로 만드는데 기여를 하지 못하기 때문이다. 이로

인해 $f_3 = (q \vee r) \wedge (q \vee \bar{r}) \wedge (\bar{q} \vee r)$ 로 축소되어 $q=r=T$ 가 된다. 결국, $f=T$ 로 충족 가능하며, $pqr=TTT$ 의 진리 값을 얻는다.

식 (1)의 $f_1 = (p \vee q \vee r) \wedge (\bar{p} \vee q \vee \bar{r}) \wedge (p \vee \bar{q} \vee s) \wedge (\bar{p} \vee \bar{r} \vee \bar{s})$ 은 $k=3, p=q=r=s=\{0,1\}$, 발생 빈도수는 $p(4), q(3), r(3), s(2)$ 이다. 최소 발생 빈도수인 $s=T$ 로 설정하면 $f_1 = (p \vee q \vee r) \wedge (\bar{p} \vee q \vee \bar{r}) \wedge (p \vee \bar{q}) \wedge (\bar{p} \vee \bar{r})$ 로 축소된다. $k \geq 2, p=r=\{0,1\}, q=\{1\}$, 발생 빈도수는 $p(3), q(2), r(3)$ 이다. 따라서 $q=T$ 로 설정하면 $f_1 = (\bar{p} \vee \bar{r})f =$ 로 축소된다. $k \geq 2, p=r=\{0\}$ 이므로 $p=F$ 로 설정하면 r 은 don't care이다. 결국 $pqr=FTFT, FTTT$ 로 표 1에서 이들 진리 값은 $f=T$ 이기 때문에 f 는 충족 가능하다. 참고로, Gubin^[11]이 제시한 식 (4)의 2-SAT 해를 구하여 보자.

$$f_4 = \bar{p} \wedge \bar{q} \wedge \bar{r} \wedge (p \vee q) \wedge (p \vee r) \wedge (q \vee r) \quad (4)$$

$k=1$ 인 (\bar{p}) 절에 따라 $p=F$ 로 결정하면 $f_4 = (\bar{q}) \wedge (\bar{r}) \wedge (q) \wedge (r) \wedge (q \vee r)$ 로 축소된다. $k=1$ 인 (\bar{q})와 (q)절이 존재하여 $f=F$ 로 충족 불가능하다.

결국, 제안된 알고리즘은 Gubin^[10,11]의 복잡한 $O(m^3)$ 다항시간 알고리즘을 $O(mn)$ 의 다항시간으로 축소시킨 결과를 얻었으며, f 의 충족 가능 여부를 쉽게 결정할 수 있음을 알 수 있다. 또한, 제안된 알고리즘은 k -SAT, ($2 \leq k$)의 모든 문제를 다항시간으로 해결할 수 있는 확장성을 갖고 있다.

IV. 실험 및 결과 분석

본 장에서는 다양한 문제들을 대상으로 제안된 알고리즘의 적용성을 평가해 본다.

Wikipedia^[1]에서 제시한 식 (5)의 3-SAT 해를 구하여 보자. 여기서는 6개 문자가 존재한다.

$$f_5 = (x \vee y \vee \bar{z}) \wedge (a \vee \bar{b} \vee c) \wedge (\bar{x} \vee b \vee z) \quad (5)$$

$k=3, x=z=b=\{0,1\}, y=a=c=\{1\}$, 발생 빈도수 $x(2), y(1), z(2), a(1), b(2), c(1)$ 로 $y=a=c=T$ 로 결정하면 x, z, b 의 값만 구하면 된다. y, a, c 를 포함하는 절을 제거하면 $f_5 = (\bar{x} \vee b \vee z)$ 만 남게 되며, b, z 중 어느 하나만 T 가 되면 된다. $b=T$ 로 하면 x, z 는 don't care이다. 결국,

$f = T$ 로 충족 가능하다.

3-SAT이 하나의 절에 3개 이하의 문자가 존재하는 경우에도 동일한 방법을 적용할 수 있다. 식 (6)은 Boule^[12]에서 인용되었다.

$$f_6 = (p \vee q \vee r) \wedge (p \vee q \vee \bar{r}) \wedge (\bar{p} \vee q \vee \bar{r}) \wedge (p \vee r \vee s) \wedge (\bar{p} \vee r \vee s) \wedge (\bar{p} \vee r \vee \bar{s}) \wedge (\bar{q} \vee \bar{r} \vee s) \wedge (\bar{q} \vee \bar{r} \vee \bar{s}) \quad (6)$$

$k=3, p=q=r=s=\{0,1\}$, $p(6), q(5), r(8), s(5)$ 의 빈도수를 가지므로 $q = T$ 라 하고, q 포함 절을 삭제하고 나머지 절에서 \bar{q} 를 삭제하면 $f_6 = (p \vee r \vee s) \wedge (\bar{p} \vee r \vee s) \wedge (\bar{p} \vee r \vee \bar{s}) \wedge (\bar{r} \vee s) \wedge (\bar{r} \vee \bar{s})$ 로 축소된다. $(\bar{r} \vee s) \wedge (\bar{r} \vee \bar{s})$ 에 따라 $s = T$ 로 결정하면 $f = (\bar{p} \vee r) \wedge (\bar{r})$ 로 축소된다. $p(1), r(2)$ 이며, $p = \{0\}$ 이므로 $p = F$ 로 설정하면 $f = (\bar{r})$ 로 축소된다. 결국, $r = F$ 로 설정되며, $f = T$ 로 충족 가능하다.

Gubin^[8]이 제시한 식 (7)의 해를 구하여 보자.

$$f_7 = (p \vee q \vee r) \wedge (p \vee q \vee \bar{r}) \wedge (\bar{p} \vee s) \wedge (\bar{p} \vee \bar{s}) \wedge (\bar{q}) \quad (7)$$

$k=1$ 인 (\bar{q}) 절에 따라 $q = F$ 로 설정한다. \bar{q} 절을 제거하고 q 문자를 제거하면 $f_7 = (p \vee r) \wedge (p \vee \bar{r}) \wedge (\bar{p} \vee s) \wedge (\bar{p} \vee \bar{s})$ 이다. $k=2, p=r=s=\{0,1\}$, $p(4), r(2), s(2)$ 의 빈도수이므로 $r = T$ 로 설정하면 $f_7 = (p) \wedge (\bar{p} \vee s) \wedge (\bar{p} \vee \bar{s})$ 로 축소된다. $k=1$ 인 (p) 절에 따라 $p = T$ 로 설정하면 $f_7 = (s) \wedge (\bar{s})$ 로 축소된다. 결국, $(s) \wedge (\bar{s})$ 를 만족시키는 s 의 진리 값은 없으므로 $f = F$ 로 충족 불가능하다.

Jelliffe^[13]가 제시한 식 (8)의 3-SAT 해를 구하여 보자.

$$f_8 = (a \vee b \vee \bar{c}) \wedge (\bar{b} \vee d \vee \bar{e}) \wedge (\bar{a} \vee c \vee e) \wedge (\bar{c} \vee \bar{d} \vee \bar{e}) \quad (8)$$

$k=3, a=b=c=d=e=\{0,1\}$, $a(2), b(2), c(3), d(2), e(3)$ 의 빈도수로 $a = T$ 로 설정하면 $f_8 = (\bar{b} \vee d \vee \bar{e}) \wedge (c \vee e) \wedge (\bar{c} \vee \bar{d} \vee \bar{e})$ 로 축소된다. $k \geq 2, b = \{0\}, c=d=e=\{0,1\}$ 으로 $b = F$ 로 설정하면 $f_8 = (c \vee e) \wedge (\bar{c} \vee \bar{d} \vee \bar{e})$ 로 축소된다. $d = \{0\}, c=e=\{0,1\}$ 로 $b = F$ 로 설정하면 $f_8 = (c \vee e)$ 로 축소된다. 결국, $c = T$ 로 하면 e 는 don't care이다. 따라서 $f = T$ 로 충족 가능하다.

Krasnoger^[14]가 제시한 식 (9)의 3-SAT 해를 구하여 보자.

$$f_9 = (p \vee q \vee r) \wedge (\bar{q} \vee \bar{r} \vee s) \wedge (\bar{p} \vee q \vee \bar{s}) \wedge (p \vee \bar{q} \vee r) \quad (9)$$

$k=3, p=q=r=s=\{0,1\}$, $p(3), q(4), r(3), s(2)$ 의 빈도수이므로 최소 빈도수 $s = T$ 로 설정하면 $f_9 = (p \vee q \vee r) \wedge (\bar{p} \vee q) \wedge (p \vee \bar{q} \vee r)$ 로 축소된다. $k \geq 2, p=q=\{0,1\}, r=\{1\}$ 로 $r = T$ 로 설정하면 $f_8 = (\bar{p} \vee q)$ 로 축소된다. 따라서 $q = T$ 로 하면 s 는 don't care이며, $f = T$ 로 충족 가능하다.

Lai^[15]가 제시한 식 (10)의 3-SAT 해를 구하여 보자.

$$f_{10} = (p \vee \bar{q} \vee \bar{r}) \wedge (\bar{p} \vee q \vee r) \wedge (p \vee q \vee r) \quad (10)$$

$k=3, p=q=r=\{0,1\}$, $p(3), q(3), r(3)$ 으로 발생 빈도수가 모두 동일하므로 첫 번째 문자 $p = T$ 로 설정하면 $f_{10} = (q \vee r)$ 로 축소된다. 따라서 $q = T$ 로 하면 r 은 don't care이며, $f = T$ 로 충족 가능하다.

Kochenberger^[3]가 제시한 식 (11)의 3-SAT 해를 구하여 보자.

$$f_{11} = (p \vee q \vee r) \wedge (p \vee \bar{q} \vee r) \wedge (\bar{p} \vee q \vee \bar{r}) \wedge (q \vee \bar{r} \vee s) \wedge (\bar{q} \vee r \vee s) \wedge (\bar{q} \vee \bar{r} \vee \bar{s}) \wedge (q \vee s \vee t) \wedge (\bar{q} \vee r \vee t) \wedge (q \vee \bar{r} \vee t) \wedge (r \vee s \vee t) \wedge (r \vee \bar{s} \vee \bar{t}) \wedge (\bar{r} \vee \bar{s} \vee \bar{t}) \quad (11)$$

$k=3, p=q=r=s=t=\{0,1\}$, $p(3), q(9), r(11), s(7), t(6)$ 발생 빈도수로 최소 발생 빈도수 $p = T$ 로 설정하면 $f_{11} = (q \vee \bar{r}) \wedge (q \vee \bar{r} \vee s) \wedge (\bar{q} \vee r \vee s) \wedge (\bar{q} \vee \bar{r} \vee \bar{s}) \wedge (q \vee s \vee t) \wedge (\bar{q} \vee r \vee t) \wedge (q \vee \bar{r} \vee t) \wedge (r \vee s \vee t) \wedge (r \vee \bar{s} \vee \bar{t}) \wedge (\bar{r} \vee \bar{s} \vee \bar{t})$ 로 축소된다. $k \geq 2, q=r=s=t=\{0,1\}$, $q(7), r(9), s(7), t(6)$ 로 최소 발생 빈도수 $t = T$ 로 설정하면 $f_{11} = (q \vee \bar{r}) \wedge (q \vee \bar{r} \vee s) \wedge (\bar{q} \vee r \vee s) \wedge (\bar{q} \vee \bar{r} \vee \bar{s}) \wedge (r \vee s) \wedge (\bar{r} \vee \bar{s})$ 로 축소된다. $k \geq 2, q=r=s=\{0,1\}$, $q(4), r(6), s(5)$ 로 최소 발생 빈도수 $q = T$ 로 설정하면 $f_{11} = (r \vee s) \wedge (\bar{r} \vee \bar{s}) \wedge (r \vee s) \wedge (\bar{r} \vee \bar{s})$ 로 축소된다. $k \geq 2, r=s=\{0,1\}$, $r(4), s(4)$ 로 동일한 발생 빈도수를 가지므로 $r = T$ 로 설정하면 $f_{11} = (\bar{s}) \wedge (\bar{s})$ 로 축소되어 $s = F$ 로 설정하면 $f = T$ 로 충족 가능하다.

제안된 알고리즘을 11개의 2-SAT, 3-SAT에 적용한 결과 절에 포함된 문자수와 식에 포함된 전체 문자 수 k 에 상관없이 모든 SAT에서 f 의 충족 가능 여부를 결정할 수 있었다. 결국, 제안된 알고리즘은 일반적인 k -SAT 알고리즘으로 확장 가능하다.

V. 결론 및 향후 연구과제

본 논문은 3-SAT 문제가 NP-완전이 아닌 $O(mn)$ 의

다항시간 알고리즘임을 증명하였다.

제안된 알고리즘은 $k=1$, '0' 또는 '1' 진리 값만 가지는 경우, 최소 발생 빈도수 순으로 우선순위를 결정하여 문자의 진리 값을 결정하였다. 결정된 문자의 진리 값이 T 인 절을 삭제하고, 남아 있는 절에서 진리 값이 F 인 해당 문자를 삭제하는 방법으로 식을 축소시키는 방법을 적용하였다. 또한, DPLL 알고리즘과 같이 역추적 기법을 적용하지 않아 수행 시간이 빠르다. 이러한 간단한 방법을 적용한 결과 $O(nm)$ 의 다항시간으로 주어진 식 f 의 충족 가능 여부를 판단할 수 있었으며, 모든 문자의 진리 값도 결정할 수 있었다.

제안된 알고리즘을 다양한 문제들에 적용한 결과 모든 문제들에서 충족가능 여부를 정확히 결정할 수 있었다. 결국, 제안된 알고리즘은 모든 SAT 문제에 확장하여 적용할 수 있을 것이다. 단지 3-SAT을 일반적인 SAT 문제로 확장하면 계산 복잡도가 현저히 증가할 것으로 판단되며, 이러한 문제점은 실제 프로그램으로 구현하여 실험을 통해 추후 밝힐 예정이다. 따라서 추후 일반적인 SAT 문제에 대해 본 제안된 알고리즘이 적합한지 아니면 개선할 부분이 있는지 여부를 연구할 예정이다.

References

- [1] Wikipedia, "Boolean Satisfiability Problem," http://en.wikipedia.org/wiki/Boolean_satisfiability_problem, Retrieved Jul. 2022.
- [2] Wikipedia, "Satisfiability and Validity," http://en.wikipedia.org/wiki/Satisfiability_and_validity, Retrieved Jul. 2022.
- [3] G. Kochenberger, "Solution of Satisfiability Problems by Unconstrained Quadratic Programming," Hearin Center for Enterprise Science, <http://hces.bus.olemiss.edu/conference/militaryworkshop-2003/Kochenberger.ppt>, 2003.
- [4] M. Davis and H. Putnam, "A Computing Procedure for Quantification Theory," *Journal of the ACM*, Vol. 7, No. 3, pp. 201-215, Jul. 1960.
DOI: <https://doi.org/10.1145/321033.321034>
- [5] M. Davis, G. Logemann, and D. Loveland, "A Machine Program for Theory Proving," *Communications of the ACM*, Vol. 5, No. 7, pp. 394-397, 1962.
- [6] M. Qasem, "SAT and MAX-SAT for the Lay-Researcher," School of Electronics and Computer Science, University of Southampton, 2010.
- [7] S. A. Cook, "The Complexity of Theorem Proving Procedures," *Proceedings of the Third Annual ACM Symposium on the Theory of Computing*, pp.

151-158, May 1971.

DOI: <https://doi.org/10.1145/800157.805047>

- [8] R. M. Karp, "Reducibility Among Combinatorial Problems," in R. E. Miller and J. W. Thatcher (editors). *Complexity of Computer Computations*. pp. 85-103, New York: Plenum, 1972.
- [9] R. M. Karp, "Reducibility Among Combinatorial Problems," In R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, "Complexity of Computer Computations," New York, Plenum. pp. 85-103, 1972.
DOI: https://doi.org/10.1007/978-1-4684-2001-2_9
- [10] S. Gubin, "A Polynomial Time Algorithm for 3-SAT," Cornell University Library, <http://arxiv.org/abs/cs.cc/0701023>, 2008.
- [11] S. Gubin, "Polynomial Time Algorithm for 3-SAT Examples of Use," Cornell University Library, <http://arxiv.org/abs/cs.cc/0703098>, 2008.
- [12] M. Boulé, "MBSat Satisfiability Program and Heuristics Brief Overview," Electrical and Computer Engineering, McGill University, <http://www.ece.mcgill.ca/~zzilic/649/MB.ppt>, 2001.
- [13] R. Jelliffe, "3-SAT," <http://lists.xml.org/archives/xml-dev/200103/msg00724.html>, 2001.
- [14] N. Krasnoger, "Computability (and Complexity): Reductions and Completeness," School of Computer Science, The University of Nottingham, 2009.
- [15] K. J. Lai, "Design and Analysis of Algorithms," MIT, http://web.mit.edu/k_lai/www/6.046/r12-handout.pdf, 2008.

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