The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 17, No. 4, pp.155-161, Aug. 31, 2017. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2017.17.4.155
JIIBC 2017-4-20

스도쿠 퍼즐을 위한 이진역추적 알고리즘

Binary Backtracking Algorithm for Sudoku

이상운*

Sang-Un, Lee*

요 약 본 논문은 지금까지 NP-완전 문제로 다항시간 알고리즘이 존재하지 않는 스도쿠 퍼즐 문제의 해를 다항시간으로 구하는 알고리즘을 제안하였다. 제안된 알고리즘은 빈칸들에 $[1,2,\cdots,9]$중에서 행, 열과 블록에 존재하는 실마리 숫자를 제외한 후보 집합을 초기치로 설정하였다. 빈칸의 후보 집합에 대해 Stuart이 제시한 기본적인 규칙들과 더불어 2개의 추가 규칙을 제시하고, 마지막으로 이진 역추적 기법(BBT)을 적용하였다. 다양한 부류의 해를 갖는 실험데이터들에 대해 적용한 결과 제안된 BBT 알고리즘은 어떠한 부류의 해를 갖던지에 상관없이 주어진 스도쿠 퍼즐을 풀 수 있음을 보였다.

Abstract This paper suggests polynomial time solution algorithm for Sudoku puzzle problem. This problem has been known NP (non-deterministic polynomial time)-complete. The proposed algorithm set the initial value of blank cells to value range of $[1,2,\cdots,9]$.Then the candidate set values in blank cells deleted by preassigned clue in row, column, and block. We apply the basic rules of Stuart, and proposes two additional rules. Finally we apply binary backtracking(BBT) technique. For the experimental Sudoku puzzle with various categories of solution, the BBT algorithm can be obtain all of given Sudoku puzzle regardless of any types of solution.

Key Words : Sudoku, NP-Complete, Single value, Unique value, Common value

Ⅰ. 서 론

신문 일간지의 퍼즐란은 물론, 스도쿠만을 전문적으로 다루는 잡지나 TV방송도 있을 정도이다. 한국에서도 수많은 스도쿠 서적이 나오고, 네이버에서 매일 하나씩 스도쿠 문제를 제공할 정도로 스도쿠는 두터운 매니아층을 가지고 있으며, 2010년 3월 애플 선정 '새롭고 주목할 만한 어플', 한국 애플 앱스토어 1위, 미국 애플 앱소토어 퍼즐 20위 등을 차지하고 있다. 이와 같이 방송통신의 발전에 따른 퍼즐 등의 앱들이 점차 수요가 증대되는 사회적 현상에 발맞추어 본 논문에서는 스도쿠 퍼즐을 푸는데 있어서 규칙(법칙)이 존재함을 밝히고, 이 법칙에 따라 퍼즐을 보다 쉽게 풀 수 있는 방법을 제안한다.

전형적인 스도쿠 퍼즐(Sudoku puzzle)은 $n\times n(n=9)$개의 정방행렬 형태를 취하고 있으며, 행렬 내에 3×3의 9개의 블록으로 분할되어 있다. 스도쿠 문제는 문제를 풀 수 있도록 사전에 주어진 숫자를 실마리 (clue)라 하며, 스도쿠 퍼즐은 k개의 실마리가 주어진 상황에서 m^2-k개의 빈칸(blank cells)의 각 행, 열과 블록에 $[1,2,\cdots,9]$의 숫자가 중복되지 않도록 배정하는 문제이다[1].

스도쿠 퍼즐은 유일 해(unique solution)를 갖도록 유효한(valid) $k-$실마리 스도쿠 퍼즐을 생성하고자 한다[2]. 그러나 문제를 풀면 유일 해를 갖는 경우는 극소수에

*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과
접수일자 : 2017년 5월 28일, 수정완료 : 2017년 7월 28일
게재확정일자 : 2017년 8월 11일

Received: 28 May, 2017 / Revised: 28 July, 2017 /
Accepted: 11 August, 2017
*Corresponding Author: sulee@gwnu.ac.kr
Dept. of Multimedia Eng., Gangneung-Wonju National University, Korea

불과하고, 대부분은 회피 불가능 집합(unavoidable set)이 존재하는 다중 해(multiple solution)를 갖는 경우[1]와 회피 불가능 집합은 존재하지 않지만 별개의(distinct) 다중 해를 갖는 경우[3]도 발생할 수 있다.

유일 해를 얻을 수 있는 최소 실마리 개수는 17로, 17-clue스도쿠 퍼즐까지 알려져 있다.[1] 스도쿠 퍼즐의 9×9 정방행렬에 대해 약 6.671×10^{21} 개의 가능한 숫자 배열이 존재[4-6]하여 간단히 풀 수 있는 수학적 방법은 존재하지 않는 것으로 NP-완전(NP-Complete) 문제로 등재되었다.[7] 이와 같이 풀기 어려운 근거로, Maria와 Toroczkai[8]는 스도쿠 퍼즐의 빈칸에 들어갈 수 있는 숫자의 다양성을 스도쿠의 혼돈(chaos)으로 제시하였다. 또한, Lewis[9]는 메타휴리스틱 방법으로 스도쿠 문제를 풀 수 있음을 제시하였다.

반면에, Reigh[10]는 수학자들은 스도쿠 퍼즐에 대한 돌파구(breakthrough)가 존재함을 제시하였다. 2005년부터 이 문제에 대해 본격적으로 연구되기 시작하여 휴리스틱 방법으로 역추적(backtracking, BT), 분기한정법(branch-and-bound, BB), 정확피복(exact cover, EC), 무작위 공격(brute-force, BF)과 통계적 탐색/최적화기법 들을 적용하여 해를 구하려고 시도되고 있다[11]. 그러나 정확피복 문제도 NP-완전으로, 아직까지 가능한 모든 경우 수를 열거하는 무작위 공격이나 가능한 숫자들 중에서 임의의 숫자를 배정하고, 만약, 실패하면 이전의 분기로 복귀하여 다른 숫자들을 배정하는 역추적이나 분기한정법을 적용해야만 한다.[12]

Crook[12]은 역추적 기법으로 스도쿠 퍼즐을 푸는 연필과 종이(pencil-and-paper) 알고리즘을 제안하였다. Delahaye[3]는 단지 셀(*only* cell), 강제 셀(*forced* cell), 가능성 범위 단순화(simplifying the range of possibilities)와 시행착오법(trial-and-error)의 4가지 규칙을 제안하였다. Stuart[13]의 Sudoku Solver(SS1)는 기본적으로 naked single(NS)/pairs(NP)/triples(NT)/quads(NQd)/quint(NQt),hidden single(HS)/pairs(HP)/triples(HT)/quads(HQd)/quint(HQt), pointing pairs(PP)/triples (PT)과 더불어 기타 28가지의 다양한 규칙과 시행착오법을 제안하였다. Sudoku-Solutions. com[14]의 Sudoku Solver (SS2)는 Stuart의 기본 규칙과 더불어 4가지의 추가 규칙만을 제안하였다. Lee[15]는 시행착오법을 제안하였다.

본 논문에서는 SS1과 SS2의 기본 규칙인 NS/NP/NT/NQd/NQt,HS/HP/HT/HQd/HQt,PP/PT에 추가로 2가지 규칙을 제안하고, 마지막으로 이진 역추적(binary backtracking, BBT)을 추가한 방법으로 $O(n^2)$의 다항시간으로 스도쿠 퍼즐을 푸는 알고리즘을 제안한다.

제안된 알고리즘은 SS1[13]과 같이 각 빈칸에 [1,2,…,9]의 숫자 중에서 행, 열과 블록에 사전에 배정된 실마리 숫자들을 삭제한 후보 집합(candidate set)을 적용한다. 후보 집합에 대해 기본 규칙들과 이진 역추적 방법을 적용하여 $O(n^2)$의 다항시간 복잡도로 정확한 해를 구한다. 2장에서는 스도쿠 퍼즐의 해 형태와 더불어 스도쿠 문제를 푸는 방법들을 고찰한다. 3장에서는 스도쿠 문제를 다항시간 복잡도로 해를 구하는 이진 역추적 알고리즘을 제안한다. 4장에서는 스도쿠 퍼즐들 중에서 풀기 어려운 다양한 사례를 대상으로 제안된 알고리즘의 적용성을 평가해 본다.

II. 스도쿠 퍼즐의 해와 푸는 방법

전형적인 스도쿠 퍼즐은 셀 보드 상에서 게임을 수행한다. 이 보드에는 9×9의 정방행렬로 81개의 칸이 존재하며, 내부에는 3×3의 9개의 서브보드인 블록으로 분할되어 있다. 이 블록은 맨 좌측상단부터 순서대로 1부터 9까지 번호가 부여된다.[4]

정의 1 (실마리). 스도쿠 퍼즐은 9×9개의 칸들 중에서 k개의 칸에 1부터 9까지의 숫자들 중 행, 열과 블록에 동일한 숫자가 존재하지 않도록 사전에 실마리 숫자가 주어진다.

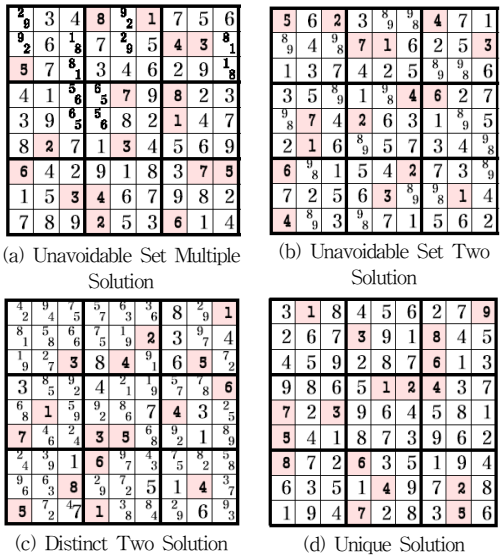
정의 2 (스도쿠 해). 스도쿠 퍼즐의 해는 모든 행, 열과 블록에 [1,2,…,9]에 있는 모든 숫자를 포함하고 있어야 하며, 모든 칸에는 단지 하나의 숫자, 오로지 하나의 숫자만 갖고 있어야 한다.[4]

정의 3 (유효한). 주어진 스도쿠 퍼즐 문제의 해가 유일하게 (uniquely) 1개만 존재하면, 유일 해를 가진 스도쿠 문제로 이를 유효한(valid) 또는 올바른 (proper)이라 한다. 유효한 스도쿠 퍼즐은 단지 하나의 완료(completion)를 가져야만 한다. 즉, 오로지 하나의 해(only single solution)만을 가져야 하며, 단지 하나의 정확한 답만을 가져야만 한다.[1]

정의 4 (회피 불가능 집합과 다중 해). 스도쿠 퍼즐의 해가 다수가 존재할 경우, 2가지 경우가 발생한다. 첫 번

째는 한 쌍의 숫자를 가진 4개 칸 또는 다수의 칸이 존재하는 경우, 이 쌍의 숫자를 상호 교환하면 2가지 해를 얻는다. 이를 회피 불가능 집합이라 한다. 두 번째는 전혀 다른 해(distinct solution)를 다수 얻는 경우 이를 다중해(multiple solution)라 한다.

스도쿠 퍼즐을 풀면 일반적으로 해는 그림 1과 같이 4가지 형태를 나타낸다. 16-clue 스도쿠 퍼즐은 3.3×10^{16}의 가능한 위치가 존재한다. 이와 같이 16-clue 스도쿠는 다양한 퍼즐을 만들 수 있지만 유일한 해를 갖는 경우는 없다. 17-clue 스도쿠 퍼즐의 경우 7.02×10^{26}개의 다양한 경우수가 발생하지만 유일한 해를 갖는 경우는 극소수에 불과하다.



(a) Unavoidable Set Multiple Solution (b) Unavoidable Set Two Solution

(c) Distinct Two Solution (d) Unique Solution

그림 1. 스도쿠 퍼즐의 해 부류
Fig. 1. Solution category of Sudoku puzzle

Sudoku Solver (SS1)[13]은 주어진 규칙들에 의해서만 문제가 풀리는지 여부를 검증하며, 단지 유일 해를 가진 스도쿠 퍼즐 문제에 대해서만 해를 구한다. Sudoku Solver(SS2)[14]는 유일 해뿐 아니라 회피 불가능 집합과 다중 해를 가진 스도쿠 퍼즐 문제에 대해서도 해를 구한다. 그러나 SS2[14]의 단점은 $[1,2,\cdots,9]$중 8개 이상의 숫자가 있을 경우와 $k\geq16$에 대해서만 해를 구하며, 2개 이상의 숫자가 없으면 해를 구하지 못한다. 따라서 주어진 스도쿠 퍼즐 문제가 유일 해를 갖고 있는지 여부는 SS1과 SS2가 동일한 해를 얻는지 여부로 판단할 수 있다.

유일 해를 갖는 스도쿠 퍼즐 문제가 극소수에 불과하므로, 우리의 관심은 모든 스도쿠 퍼즐 문제에 대해 해를 구할 수 있는 알고리즘이 요구된다. 유일 해를 갖지 않는 스도쿠 퍼즐을 풀기 위해서는 일반적으로 역추적이나 분기한정법을 적용하고 있다.

만약, 분기한정법의 하한과 상한 값을 실패(fail)라고 정의하면 분기한정법은 역추적법과 동일하다. 역추적과 분기한정법의 최대 단점은 첫 번째 빈칸에서 시작한다는 점이다. 만약, 첫 번째 빈칸과 연속해서 탐색하는 빈칸들이 다수의 숫자들을 후보로 갖고 있는 경우 다중 분기가 발생하며, 이는 일반 트리를 탐색하는 경우로 가능한 경로가 기하급수적으로 증가할 수 있다. 이러한 문제점을 해결하고자 3장에서는 단지 2개의 분기만을 가진 경우의 역추적을 할 수 있는 이진트리 역추적법을 제안한다.

Ⅲ. 이진 역추적 알고리즘

본장에서 제안하는 이진 역추적 알고리즘(BBT)은 다음의 후보 집합 이론과 4개의 범주를 가진 열쇠에 기반하고 있다. 4개의 열쇠 범주는 성공 열쇠(success key), 성공적 감소 열쇠(success reducing key), 매우 성공적 감소 열쇠(very high success reducing key)와 50% 성공 열쇠(half success key)이다.

Stuart[13]의 SS1은 반드시 모든 규칙을 적용하여 유일 해를 얻는 경우에만 문제를 풀 수 있다. 그러나 유일 해를 갖는 스도쿠 퍼즐 문제가 극소수에 불과하기 때문에 대부분의 스도쿠 퍼즐에 대해서는 해를 구하지 못하는 경우가 일반적이다. SS2는 이러한 문제점을 보완하여 유일 해를 갖지 않는 대부분의 스도쿠 퍼즐 문제에 대해서도 해를 제공하지만 $[1,2,\cdots,9]$의 숫자들 중 실마리로 8개 숫자는 있어야만 해를 제공한다.

본 장에서는 어떠한 종류의 해를 갖는 스도쿠 퍼즐에 대해서도 해를 구할 수 있는 알고리즘을 제안한다. 제안된 알고리즘은 SS1과 SS2에서 적용된 기본적인 규칙인 성공 열쇠와 성공적 감소 열쇠 범주에 속한 규칙들을 적용하고, 추가적으로 매우 성공적 감소 열쇠와 50% 성공 열쇠의 범주를 적용하여 빠르게 해를 구할 수 있도록 하였다.

정리 1 (후보 집합). 빈칸에는 행, 열과 블록에 있는 실마리 숫자들은 넣을 수 없다. 따라서 $[1,2,\cdots,9]$의 숫자들 중 행, 열과 블록에 있는 실마리 숫자들을 제외한 해당

빈칸에 넣을 수 있는 숫자들을 후보 집합이라 하자.

제안된 알고리즘은 n^2-k개 빈칸의 후보집합에 대해 다음의 4개 범주 열쇠들을 적용한다.

[범주 1: 성공 열쇠]

규칙 1 (Naked Single, NS). Delahaye[3]의 forced cell과 Stuart[13]의 naked single 규칙.

규칙 2 (Hidden Single, HS). Delahaye[3]의 only cell과 Stuart[13]의 hidden single 규칙.

[범주 2: 성공적 감소 열쇠]

규칙 3 (Naked Pairs, NP). Delahaye[3]의 가능성 범위 단순화 방법과 Stuart[13]의 naked pairs 규칙.

규칙 4부터 규칙 11까지는 Stuart[13]에서 인용되었다.

규칙 4 (Hidden Pairs, HP).

규칙 5 (Naked Triples, NT).

규칙 6 (Hidden Triples, HT).

규칙 7 (Naked Quads, NQd).

규칙 8 (Hidden Quads, HQd).

규칙 9 (Naked Quint, NQt).

규칙 10 (Pointing Pairs, PP).

규칙 11 (Pointing Triples, PT).

[범주 3: 매우 높은 성공적 감소 열쇠]

규칙 12 (Two Cardinality Cell, R2). 만약, 하나의 빈칸이 2원소를 갖고 있고, 이 2개 원소들을 갖고 있는 빈칸과 일부 원소만을 가진 빈칸이 존재하는 경우, 이 2개 원소들의 일부 원소를 가진 빈칸의 일부 원소 숫자는 삭제할 수 있다.

만약, $(i,1)=[a,b],(i,2)=[a,b,c,d],(i,3)=[a,b,c,d,e]$와 $(i,4)=[b,c,d,e]$을 가진 경우, 숫자 $[a,b]$는 $(i,1),(i,2)$와 $(i,3)$이 공통으로 갖고 있어 이 3개 빈칸 중에 어느 하나의 빈칸이 이들 숫자를 가질 확률이 매우 높다. 따라서 $[b,c,d,e]$를 가진 $(i,4)$ 빈칸은 $[c,d,e]$로 감소된다.

규칙 13 (Three Cardinality Cell, R3). 만약, 하나의 빈칸이 3원소를 갖고 있고, 이 3개 원소들을 갖고 있는 빈칸과 일부 원소만을 가진 빈칸이 존재하는 경우, 이 3개 원소들의 일부 원소를 가진 빈칸의 일부 원소 숫자는 삭제할 수 있다.

만약, $(i,1)=[a,b,c],(i,2)=[a,,b,c,d],(i,3)=[a,b,c,d,e]$와 $(i,4)=[b,c,d,e]$를 가진 경우, 규칙 12와 같이 $[a,b,c]$는 $(i,1),(i,2),(i,3)$에 공통으로 갖고 있어 $[b,c,d,e]$를 가진 $(i,4)$는 $[d,e]$로 감소된다.

[범주 4: 50% 성공 열쇠]

규칙 14 (Binary Backtracking, BB). 만약, 2개 원소 $[a,b]$를 가진 하나의 빈칸 또는 유일 숫자 $[a]$를 가진 단지 2개 빈칸만이 존재하고, 위의 규칙 1 ~ 11을 적용할 수 없는 경우, 우리는 이진 역추적 규칙을 적용하여 $[a,b]$ 중 임의의 값 $[a]$를 해당 빈칸에, 또는 2개의 빈칸 중 임의의 빈칸에 $[a]$를 배정할 수 있다.

이 규칙은 최후의 수단으로 적용하는 양자택일법 (choose either a or b method)으로 단지 2가지 경우 중 어느 하나는 성공할 수 있다. 만약, 첫 번째 숫자를 배정하고 탐색하는 도중에 실패하면 이 빈칸의 분기로 복귀하여 다른 숫자를 배정하고 탐색할 수 있다. 이 규칙은 Delahaye[3]과 Stuart[13]의 시행착오법을 양자택일법으로 탐색 경로 수를 크게 감소시키는 결과를 얻는다.

위에서 제안된 14가지 규칙을 적용하여 제안된 BBT 알고리즘은 그림 2와 같이 수행된다.

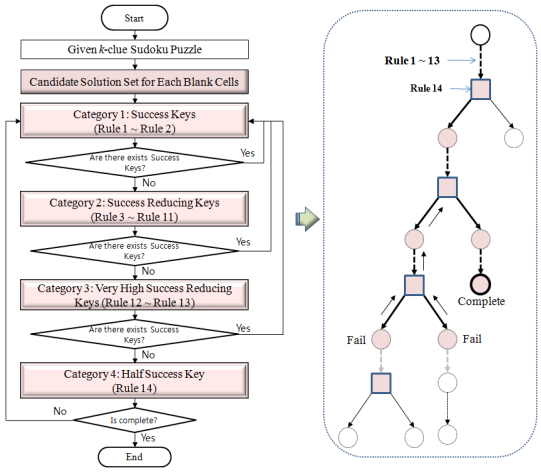


그림 2. BBT 스도쿠 알고리즘
Fig. 2. BBT Sudoku Algorithm

제안된 BBT 알고리즘은 범주 1을 최우선적으로 적용하며, 범주 1이 없으면 범주 2를 적용한다. 이와 같이 차례로 적용하고 마지막으로 부득이하게 적용할 규칙이 없을 경우 범주 4의 BBT를 적용한다. BBT를 적용하고 일단, 범주 1 ~ 3의 규칙이 존재하면 해당 규칙을 계속적으로 적용하는 방식이다.

Wikipedia[11]에서 인용된 그림 3의 30-clue 스도쿠 퍼즐에 제안된 알고리즘을 적용하여 얻은 결과는 표 1에 제시되어 있다. 301 스도쿠 퍼즐은 단지 규칙 1의 NS와 규칙 2의 HS 열쇠만을 적용하여 4회 연속 NS와 HS 빈칸들을 찾아 해를 구할 수 있음을 알 수 있다.

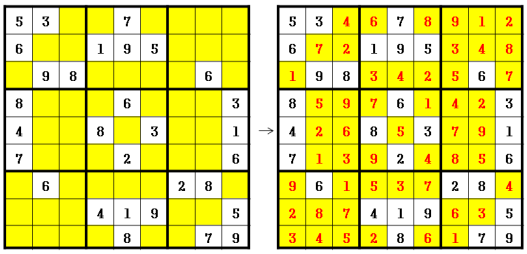


그림 3. 30-clue 수도쿠 (301)와 해
Fig. 3. 30-clue Sudoku (301) and solution

표 1. BB T알고리즘을 적용한 V30의 해
Table 1. The Solution of V30 using BBT algorithm

5	3	(1,2,4)	(2,6)	7	(2,4,6,8)	(1,4,8,9)	(1,2,4,9)	(2,4,8)
6	(2,4,7)	(2,4,7)	1	9	5	(3,4,7,8)	(2,3,4)	(2,4,7,8)
(1,2)	9	8	(2,3)	(3,4)	(2,4)	(1,3,4,5,7)	6	(2,4,7)
8	(1,2,5)	(1,2,5,9)	(5,7,9)	6	(1,4,7)	(4,5,7,9)	(2,4,5,9)	3
4	(2,5)	(2,5,6,9)	8	(5)	3	(5,7,9)	(2,5,9)	1
7	(1,5)	(1,3,5,9)	(5,9)	2	(1,4)	(4,5,8,9)	(4,5,9)	6
(1,3,9)	6	(1,3,4,5,7,9)	(3,5)	(3,5)	(7)	2	8	(4)
(2,3)	(2,7,8)	(2,3,7)	4	1	9	(3,6)	(3)	5
(1,2,3)	(1,2,4,5)	(1,2,3,4,5)	(2,3,5,6)	8	(2,6)	(1,3,4,6)	7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Ⅳ. 실험 결과 및 분석

본 장에서는 스도쿠 퍼즐들 중 난해한 그림 4의 실험 데이터들에 대해 BBT 알고리즘을 적용하여 본다.

스도쿠 퍼즐은 일반적으로 실마리 개수가 적을수록 풀기 어렵다. 그러나 특정 형태의 스도쿠 퍼즐은 실마리 개수가 충분히 많더라도 풀기 어려운 경우가 발생한다. (a)의 171은 McGuire et al.[1]이 제시한 회피 불가능 집합을 가진 17-clue 스도쿠 문제이며, (b)의 172는 Lin과 Wu[4]이 제시한 유효한 17-clue 스도쿠 문제이다. (c)는 Swain[16]

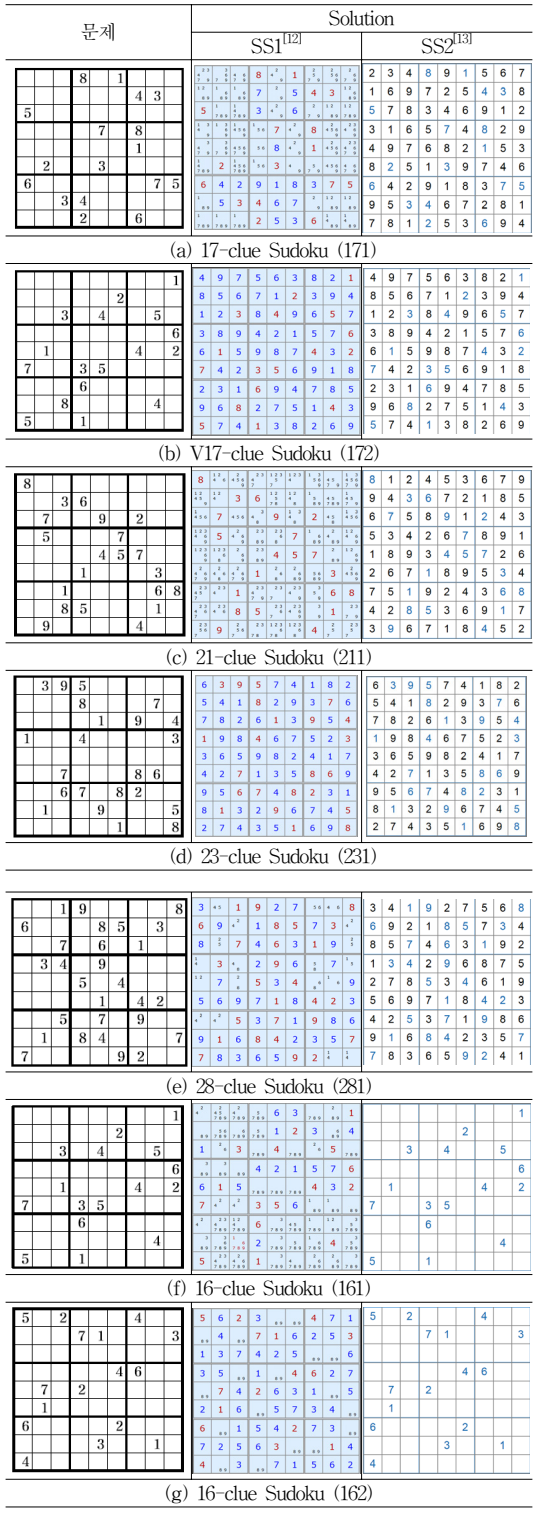
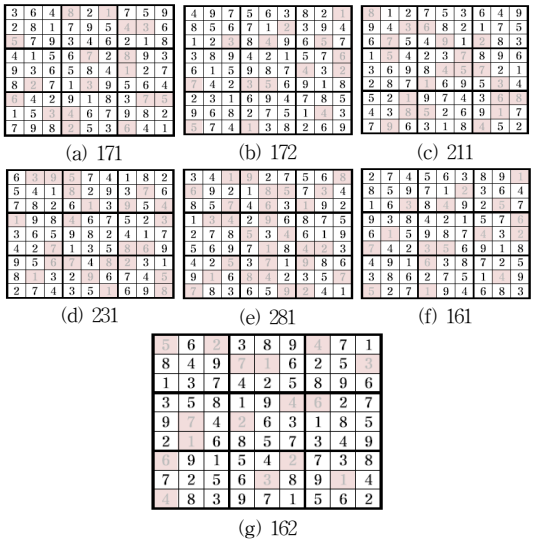


그림 4. 실험 데이터
Fig. 4. Experimental Data

에서 인용되었으며, 수학 천재인 Arto Inkalar가 세계에서 가장 풀기 어려운 스도쿠 문제로 제시한 것이다. (d)와 (e)는 Crook[12]이 제시한 문제로 매우 도전적인 스도쿠 퍼즐 문제이다. (f)는 (b)의 17-clue 스도쿠 문제로부터 (8,3)=8을 삭제한 16-clue 스도쿠 문제를 유도한 것이다. (g)는 Delahaye[3]으로부터 인용된 16-clue 스도쿠 문제이다. (f)와 (g)는 실마리 숫자가 $[1,2,\cdots,9]$ 중 7개만 존재하는 경우이다.

그림 4의 스도쿠 퍼즐들에 대해 제안된 BB 알고리즘을 적용하여 해를 얻은 결과는 그림 5에 제시하였다.



(a) 171 (b) 172 (c) 211

(d) 231 (e) 281 (f) 161

(g) 162

그림 5. 실험 데이터에 대한 BB 알고리즘의 해
Fig. 5. The solution of BB algorithm for experimental data

그림 4와 그림 5를 비교한 결과는 표 2에 제시되어 있다. 표 2로부터 SS1은 유일 해를 가진 유효한 스도쿠 퍼즐에 대해서만 해를 얻을 수 있으며, SS2는 다중 해를 가진 경우도 해를 얻을 수 있지만 실마리에 $[1,2,\cdots,9]$ 중 8개 이상의 숫자가 포함된 경우에만 해를 얻을 수 있음을 알 수 있다. 반면에, 제안된 BB 알고리즘은 다중 해를 갖는지에 상관없이, 실마리 숫자가 $[1,2,\cdots,9]$ 중에서 7개인 경우를 포함해 모든 경우에 대해서 해를 얻을 수 있음을 알 수 있다.

각 퍼즐 문제에 대해 BBT 규칙을 적용한 횟수를 살펴보면 171은 2회, 172는 1회, 211은 7회, 231은 0회, 281은 1회, 161은 10회, 162는 4회이다. 결국, 양자택일은 대략 n회 이하로 수행됨을 알 수 있다.

표 2. 알고리즘의 해 비교
Table 2. Compare with the solution of algorithms

문제	알고리즘			결과
	SS1[13]	SS2[14]	BB	
171	Fail	Complete	Complete	Unavoidable set multiple solution
172	Complete	Complete	Complete	Valid unique solution
211	Fail	Complete	Complete	Distinct multiple solution
231	Complete	Complete	Complete	Valid unique solution
281	Fail	Complete	Complete	Unavoidable set multiple solution
161	Fail	Fail	Complete	Distinct multiple solution
162	Fail	Fail	Complete	Unavoidable set multiple solution

V. 결론

스도쿠 문제는 지금까지 정확한 해법을 다항시간에 찾는 알고리즘이 존재하지 않는 NP-완전 문제로 알려져 모든 가능한 경우수를 고려하여 해를 구하는데 많은 시간이 소요되었다.

본 논문은 빈 칸들에 $[1,2,\cdots,9]$의 숫자들 중 행, 열과 블록에 존재하는 실마리 숫자들을 제외한 후보 집합을 이용하였다. 후보 집합에 대해 Stuart[13]의 기본 규칙들을 적용하고, 추가적으로 2개의 규칙과 이진 역추적 기법을 적용하였다.

SS1은 유일 해를 갖는 스도쿠 퍼즐에 대해서만 해를 제시하며, SS2는 유일 해 뿐 아니라 회피 불가능 해, 다중 해인 경우에도 해를 구할 수 있지만 실마리가 $[1,2,\cdots,9]$ 중 8개 이상 포함되어 있는 경우와 $k \geq 16$에 한해 해를 제시한다.

본 논문에서는 다양한 부류의 스도쿠 퍼즐 문제들에 대해 실험을 수행한 결과, 제안된 BBT 알고리즘은 유일 해, 회피 불가능 해, 다중 해의 어떠한 해 유형을 갖는 스도쿠 문제 뿐 아니라 실마리가 $[1,2,\cdots,9]$ 중 7개 숫자만 포함된 경우에 대해서도 해를 구할 수 있음을 보였다.

References

[1] G. McGuire, B. Tugemann, and G. Givario, "There is no 16-Clue Sudoku: Solving the Sudoku Minimum Number of Clues Problem," School of

Mathematical Sciences, University College Dublin, Ireland, pp. 1-36, Jan, 2012. arXiv: 1201.0749

[2] The Physics arXiv Blog, "Mathematicians Solve Minimum Sudoku Problem," http://www.technologyreview.com/view/509091/best-of-2012-mathematicians-solve-minimum-sudoku-problem/, MIT Technology Review, Jan. 2012.

[3] J. P. Delahaye, "The Science Behind Sudoku," Scientific American, Vol. 294, No. 6, pp. 80-87 Jun. 2006, doi:10.1038/scientificamerican0606-80

[4] H. H. Lin, and I. C. Wu, "Solving the Minimum Sudoku Problem," International Conference on Technologies and Applications of Artificial Intelligence (TAAI), pp. 456-461, Nov. 2010, doi:10.1109/TAAI.2010.77

[5] A. M. Herzberg and M. R. Murty, "Sudoku Squares and Chromatic Polynomials," Notices of the American Mathematical Society (AMS), Vol. 54, No. 6, pp. 708-717, Jun. 2007.

[6] B. Felgenhauer and F. Jarvis, "Enumerating Possible Sudoku Grids," http://www.afjarvis.staff.shef.ac.uk/sudoku. pdf, 2005.

[7] T. Yato and T. Seta, "Complexity and Completeness of Finding Another Solution and Its Application to Puzzles," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol .E86-A, No. 5, pp. 1052-1060, May 2003, ISSN: 0916-8508

[8] E. R. Maria and Z. Toroczkai, "The Chaos Within Sudoku," Scientific Reports, Vol. 2, pp. 1-8, Oct. 2012, doi:10.1038/srep00725.

[9] R. Lewis, "Metaheuristics Can Solve Sudoku Puzzles," Journal of Heuristics, Vol. 13, No. 4, pp. 387-401, Aug. 2007, doi:10.1007/s10732-007-9012-8

[10] E. S. Reich, "Mathematician Clams Breakthrough in Sudoku Puzzle," Nature News, Jan. 2012, doi:10.10.1038/nature.2012.9751

[11] Wikipedia, "Algorithmics of Sudoku," http://en.wikipedia.org/wiki/Algorithmics_of_Sudoku, Wikipedia Foundation Ltd., 2015.

[12] J. F. Crook, "A Pencil-and-Paper Algorithm for Solving Sudoku Puzzles," Notices of the American Mathematical Society (AMS), Vol. 56, No. 4, pp. 460-468, Apr. 2009.

[13] A. Stuart, "Sudoku Solver for Android and iPhone: Sudoku Solver for Mobile Devices. Ver. 1.97," http://www.sudokuwiki.org/sudoku.htm, Syndicated Puzzles Inc, Jul. 2013.

[14] Sudoku-Solutions.com, "Sudoku Solutions: Sudoku Solver," http://www.sudoku-solutions. com/, Aire Technologies, Wiesbaden, Germany, 2013.

[15] S. U. Lee, "Sudoku Algorithm," Journal of the IIBC, Vol. 15, No. 1, pp. 207-215, Feb. 2015, doi: 10.7236/JIIBC.2015.15.1.207

[16] M. Swain, "World's Hardest Sudoku Puzzle: It's the Most Baffling Brainteaser Ever Devised... Can You Solve it?," http://www.mirror.co.uk/ news/weired-news/worlds-hardest-sudoku-puzzle-ever-942299, Jun. 2012.

저자 소개

이 상 운(정회원)

- 1987년 : 한국항공대학교 항공전자공학과 (학사)
- 1997년 : 경상대학교 컴퓨터과학과 (석사)
- 2001년 : 경상대학교 컴퓨터과학과 (박사)
- 2003년 : 강원도립대학 컴퓨터응용과 전임강사
- 2004년 ~ 2007년 2월 : 국립 원주대학 여성교양과 조교수
- 2007년 3월 ~ 2015년 3월 : 강릉원주대학교 멀티미디어공학과 부교수
- 2015년 4월 ~ 현재 : 강릉원주대학교 멀티미디어공학과 정교수

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

- e-mail : sulee@gwnu.ac.kr