Study on the Performance Characteristics of Sudoku Solving Algorithms

Akta Agrawal
M.E student,
Computer Technology & Application,
S.S.T.C., Bhilai

Padma Bonde, Ph.D
Professor,
Computer Department,
S.S.T.C., Bhilai

ABSTRACT
Sudoku is a logic-based puzzle that is played by numbers from 1 to 9. The simplicity of puzzle's structure and low requirement of mathematical skills caused people to have enormous interest in accepting challenges to solve the puzzle .It is therefore of interest to study how to solve ,generate and rate such puzzles by the help of computer algorithms. This research, is carried out on limited to study of two chosen algorithm with their pros , cons and pseudo code . Paper also concluded to study further the result of testing, examine comparison, with level of puzzles and time complexity between algorithms. [1,2]

Keywords
Sudoku, Algorithm (Backtracking, Brute force)

1. INTRODUCTION
Algorithm [1] stems from the name of a Latin translation of a book written by al-Khwarizem, A Persian mathematician astronomer and geographer .The word ``Sudoku'' is short for Su-ji wa dokushin ni kagiru (in Japanese),which means ``the numbers must be single '', shown in figure 1.This game is popular in Japan since the mid 1980s.It has become extremely popular throughout the world in the last decade, triggered primarily by the publishing of Sudoku puzzles in British newspapers starting November 2004. Software programmers now supply the different programs to fulfill the growing demand for unique Sudoku puzzles. The aim of the puzzle is to enter a numerical digit from 1 through 9 in each cell of a 9×9 grid made up of 3×3 sub squares or sub grids, starting with various digits given in some cells; each row, column, and sub squares region must contain each of the numbers 1 to 9 exactly once. Players may use a wide range of strategies which deals the solutions either from zones or from the whole grid.. A sample Sudoku game and solution is depicted in figure 1.

a

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

b

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

Figure 1-Sample Sudoku Game and Solution

2. THE DIFFICULTY LEVEL OF SUDOKU PUZZLES
The difficulty level [2] of Sudoku puzzles depends on how the given numbers are placed in the Sudoku board and also how many numbers (clues) are given. Generally, the most significant aspect of difficulty ratings of Sudoku puzzles is that which techniques are required to solve the puzzles. In other words, it is important where the numbers are placed logically .Generally, a Sudoku puzzle needs at least 17-clues to solvable. It means that solving a Sudoku puzzles with 17-Clues is more difficult than a 30-clues.More given numbers, the easier and quicker the solution is. This statement may not always truth that if the number of clues becomes more the run-time of solving the puzzle would be shorter. For instance, when solving the puzzle with 28 clues the solving time is increases rapidly. The reason is that puzzle needs more techniques to solve it or the algorithm needed to iterate as long as the solution is found.

Description of levels

Extremely Easy

- Sudoku have more than 50 givens

Easy

- Sudoku generally have 36 - 49 givens (out of the total of 81 numbers in the answer).
- It has more than one given in every box.
- This level each digit from 1–9 appears as a given at least 3 times.
- It is very quick to solve, less than 10 minutes on average, and no guesses are required.

Medium

- Sudoku have around 32–35 givens.
- It has a couple of boxes with only one given.
- This level has some digits may only appear twice as a given, the rest will appear at least 3 times each.
- It take a bit longer to solve, somewhere between 10–20 minutes. No guesses are required.

Hard

- Sudoku have around 28–31 givens.
- It may have a couple of boxes with only one given
- This level has three or four digits may only appear 2 times as givens, and one digit may only appear once.
- It may take up to 45 minutes to solve, and some trial and error may be needed for example, one box may

have two candidates, and no way of determining which .

Very Hard

- Sudoku may have several boxes with less than 22 - 27 givens.
- It may have several boxes with no givens at all
- This level has most digits appear only 2 or 3 times, as well as several single occurrences.
- It can take over an hour to solve, and require trial and error.

3. EXISTING WORK

There are 6 trillion possible solutions. To solve the puzzle must have at least 17 initial numbers. Many algorithms available to solve a Sudoku puzzle. It is generally designed to be solved by human player with pencil and paper. Such players may use a range of broad strategies, but there are also a wide range of step by step algorithms that have been developed. These are typically used to allow computers both produce solution and rate the difficulty of a particular puzzle. Several research have been made to solve Sudoku problems in a more efficient way. Various exact implicit enumeration ,heuristic and meta-heuristics [3] approaches have been made to solve Sudoku efficiently .The most primitive approach to solve Sudoku has been done through brute-force technique which guarantees a logical solution of any given problems Backtracking ,on other hand ,has been found to be more promising in solving Sudoku problem [4] by reducing the search for a solution to a greater extent .A new backtrack based enumerative algorithm using Graph referencing method (GRA) has been reported by author [5] . Genetic [6] algorithms(GA) have been revealed to be an effective approach in successfully handling Sudoku as multi – objective optimization problem [7].The optimization algorithm describes a new Harmony search (HS) algorithm to solve Sudoku using meta heuristic search technique has been reported by Geem [8]. Another algorithm based on Simulated Annealing (SA) has been proposed by Henirik et al [9].

3.1 Backtracking

Backtracking is a general algorithmic technique that considers searching every possible combination in order to solve an optimization problem. The basic principle of a backtracking algorithm, in regards to Sudoku, is to work forwards, one square at a time to produce a working Sudoku grid. When a problem occurs, the algorithm takes itself back one step and tries a different path. It's nearly impossible to produce a valid Sudoku by randomly plotting numbers and trying to make them fit. Likewise, backtracking with a random placement method is equally ineffective. Backtracking best works in a linear method. It is fast, effective if done correctly.

Backtracking algorithm

Find row, column of an unassigned cell
If there is none, return true
For digits from 1 to 9
a) If there is no conflict for digit at row, column assign digit to row, column and recursively try fill in rest of grid b)
If recursion successful, return true\
c) Else, remove digit and try another
If all digits have been tried and nothing worked, return false

Pseudo code of backtracking algorithm

For standard Sudoku template (9x9) Initialize 2D array with 81 empty grids(nx=9,ny=9) Fill in some empty grid with the known values Make an original copy of the array.

Start from top left grid(nx=0,ny=0), check if grid is empty
if (grid is empty)
assign the empty grid with values (i)
if (no numbers exists in same rows & same columns same as (i) & 3x3 zone (i) is currently in)
fill in the number
if (numbers exists in same rows & same columns same as (i) & 3x3 zone (i) is currently in)
discard (i) and repick other values (i++)
}else{
while (nx<9){
Proceed to next row grid(nx++,ny)
if (nx equal 9){
reset nx = 1
proceed to next column grid(nx,ny++)
if (any equal 9){
print solutions}}}

Pros- It should be noted that the level one puzzle could be almost completely solved by the constraint propagation algorithm and required little actual search.

Cons- The search algorithm is expensive and avoiding it entirely is very valuable in those cases where it is possible.

3.2. BRUTE FORCE

Kovacs [10] describe some of the brute force methods for solving Sudoku puzzle . This is the easiest approach as it avoids computational complexities .This algorithm visits all the empty cells in a specific order and fills it with a digit from available choices .If no choice is available for a cell then it backtracks and changes the digit of previous cell .Thus the algorithm continues until all the cells are filled with appropriate digits.

Algorithm Brute Force Match(T, P)

Input text T of size n and pattern P of size m
Output starting index of a substring of T equal to P
if no such substring exists

for i ← 0 to n – m
{ test shift i of the pattern }
j ← 0
while j < m ∧ T[i + j] = P[j]
j ← j + 1
if j = m
return i {match at i}
else
break while loop {mismatch}
return -1 {no match anywhere}

Pseudo code of brute force algorithm

```
boolean solve() :=
x = 0, y = 0
for x,y in grid:
if grid[x][y].value == 0
found = true
break
if !(found)
return valid()
candidates = boolean[10]
for i := 0 -> 9:
candidates[grid[x][i].value] = true
candidates[grid[i][y].value] = true
for (cells in same box as cell x,y):
```

```
candidates[cell.value] = true
for j := 1 -> 9:
if !(candidates[j]):
grid[x][y].value = j
if solve()
return true
grid[x][y].value = 0
return false.
```

Pros- A solution is guaranteed (as long as the puzzle is valid) Solving time is mostly unrelated to degree of difficulty.

Cons- The algorithm is it may be comparatively slow when compared to computer solution methods modeled after deductive methods.

4. RESULTS AND DISCUSSIONS

The following testing result reveals the performance of algorithm which is better with respect to computing time with any levels of difficulties

Table of solving time of the backtracking and brute force method with respect to levels

Levels	Solving time on computer by algorithm	
	Backtracking	Brute force
Easy	0.0257	0.033
Medium	4.4039	3.766
Hard	5.0263	3.152
Very Hard	0.0796	0.045

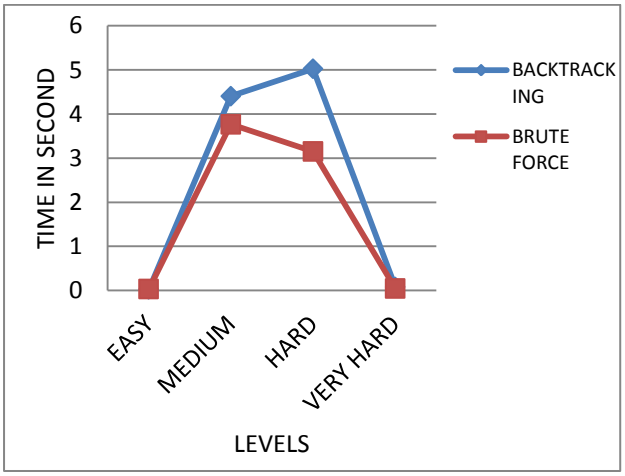

Figure 2- A Comparison between Backtracking and Brute Force algorithm

Graph shows that easy and very hard levels of puzzles look closer, little variation of time is observed, have almost same performance. In medium and hard level Brute force algorithm takes lesser time than Backtracking to solve the puzzle.

5. CONCLUSION & SCOPE OF FUTURE WORK

This study shows that brute force algorithm is more feasible to solve any Sudoku puzzles, also an appropriate method to find a solution faster and more efficient in solving ability, representation, and performances with any level of difficulties. This study will also be helpful in statistical tests and method to find some more results for comparing. Future work includes study of number of clues and run time of each difficulty level and comparing Brute Force and Backtracking algorithm based on each difficulty level.

6. REFERENCES

[1] Knuth, Donald (1997). Fundamental Algorithms, Third Edition,ISBN 0-201-89683-4.

[2] Ch. X u , W. X u ,2009, The model and algorithm to Estimate The Difficulty Levels of Sudoku puzzles, Journals of Mathematicsv Research..

[3] Lewis,R,Meta heuristics, 2007, Can solve Sudoku puzzles .Journal of heuristics,13(4),387-401.

[4] Xu,J, 2009, Using backtracking method to solve Sudoku puzzle, computer programming skills & maintainence 5,pp 17-21.

[5] Chakraborty ,r,Palidhi, S,Banerjee,, 2014, An optimized Algorithm for solving combinatorial probl;em using reference graph IOSR Journal of CS 16(3PP1-7

[6] Darwin , c, 1859, The origin of species Oxford University.

[7] Mantere T Kolljonen solving rating & generating Sudoku with GA,Proc.of IEEE 1382-1389

[8] Geem,Zong WOO, 2007, Harmony search algorithmpp371-378.

[9] Viksten ,Henrik, 2013, Performance and scalability of Sudoku solvers,Bachelor's Theses at NADA,Sweden

[10] T. Kovacs, Nov 2008., Artificial Intelligence through Search: Solving Sudoku Puzzles, Journal Papers, University of Bristol, Department of Computer Science, pp 1-14.