Artificial Intelligence Assignment 3 Report

Pradeep Kumar 2019CSM1008

16 April 2020

1 Problem Statement

In this assignment, I have to implement solver to solve Sudoku Puzzle using two different techniques of Constraint Satisfaction Problem (CSP). One is Backtracking and second is minimum conflict.

2 Introduction

Sudoku is a logic puzzle where we have to fill a board (grid) with numbers that lie between 1 and 9. The challenge of solving Sudoku Puzzle belongs to the class of constraint satisfaction problems. This and the fact that it belongs to the class of NP-complete problems make the task of finding an efficient solver not only interesting, but also very challenging.

A Sudoku puzzle instance can be described in Fig.1 as an 9x9 grid which is divided into 81 distinct squares or cells. These squares divide the whole grid into 3x3 sub-grids. To solve a Sudoku, each cell must be filled with a number in the range of 1 to 9. Additionally, three constraints must be fulfilled to achieve a valid solution:

- In every row the numbers 1 to 9 appear exactly once.
- In every column the numbers 1 to 9 appear exactly once.
- In every 3x3 sub-grid the numbers 1 to 9 appear exactly once.

	1	2	3	4	5	6	7	8	9
A			3		2		6		
B	9			3		5			1
C			1	8		6	4		
D			8	1		2	9		
E	7								8
F			6	7		8	2		
G			2	6		9	5		
H	8			2		3			9
I			5		1		3		

(a)

	1	2	3	4	5	6	7	8	9
A	4	8	3	9	2	1	6	5	7
B	9	6	7	3	4	5	8	2	1
C	2	5	1	8	7	6	4	9	3
D	5	4	8	1	3	2	9	7	6
E	7	2	9	5	6	4	1	3	8
F	1	3	6	7	9	8	2	4	5
G	3	7	2	6	8	9	5	1	4
H	8	1	4	2	5	3	7	6	9
I	6	9	5	4	1	7	3	8	2

(b)

Figure 1: (a) A Sudoku puzzle and (b) its solution.

3 Basic Concepts and Implemented Algorithms

To solve Sudoku puzzle, I am using two different algorithms:

- Backtracking search with minimum remaining values(MRV) heuristic
- Minimum conflict

3.1 Backtracking search with MRV heuristic

Backtracking search chooses values for one variable at a time and backtracks when a variable has no legal values left to assign i.e. when variable domain is empty. It chooses unassigned variable using MRV heuristic. It repeatedly chooses an unassigned variable, and then tries all values in the domain of that variable in turn, trying to find a solution. If an inconsistency is detected, then backtrack search returns failure, causing the previous call to try another value for the variable. In minimum remaining values(MRV) heuristic, we choose those variable which contains least number of values in domain to reduce the state space of search.

3.2 Minimum conflict

In minimum conflict algorithm, First, we assign random value for each variable from its domain and try to minimizes number of conflict in puzzle by selecting variable randomly and chooses those values from its domain which minimizes conflict of current variable.
Note: Minimum conflict algorithm is a local search algorithm which can stuck at local maximum.

One drawback of using the min-conflicts heuristic in local search lies in the fact that it easily can get stuck in local-optima. I additionally use some randomness in my code to resolve local optima issue.

4 Performance Table

This performance table is calculated on given 5 input test cases in PDF file. I included a text file which contain solution of all 5 test-cases.

	Input Test-case	**Total clock time (in seconds)**	**Search clock time (in seconds)**	**Number of nodes generated/ max. steps for min-conflict**
Backtracking search	1	0.0478	0.01561	41
	2	0.0398	0.0179	56
	3	0.0917	0.0548	157
	4	0.0624	0.0312	82
	5	0.2273	0.1945	66
Minimum conflict	1	0.3478	0.3133	628
	2	7.7268	7.7062	11571
	3	4.0834	4.0504	8058
	4	6.3043	6.2804	12192
	5	1.2729	1.2417	1623

Table 1: Performance table for all test-cases for both algorithms

5 Time and Space complexity

A CSP consists of a set of variables, which must be assigned in their respective finite domain, by satisfying a set of constraints. Determining if a solution exists is a NP-complete problem. For backtrack algorithm, the time complexity is O(nd) and space complexity is O($n+d$) for an instance having n variables with domain size d but

space complexity is unstable. For min-conflict algorithm, the time complexity is the maximum number of steps and space complexity is O(n) for an instance having n variables.

6 Conclusion

We can see performance of both algorithm in performance table. I observe that performance of backtracking search algorithm is better than performance of min-conflict algorithm and backtrack algorithm is also guaranteed to find a solution if exist because in backtrack search algorithm, we are propagating constraints throughout the corresponding row, column and block variables domain and trying each domain values of variable to reach solution. While min-conflict algorithm may be or may not be find a solution if exist because it can stuck at local maximum point and not able to reach a solution i.e. global maximum.