

A Hybrid Backtracking and Pencil and Paper Sudoku Solver

Onokpasa Eva
 Department of Computer
 Science,
 University of Jos

Bisandu Desmond
 Department of Computer
 Science,
 University of Jos

Bakwa Dunka
 Department of Computer
 Science,
 University of Jos

ABSTRACT

In an earlier research [1] it was found that based on the number of iterations taken to solve puzzles from [2] the pencil and paper (PnP) was the most efficient method for solving Sudoku puzzles rated easy and medium by [2]. However for more difficult puzzles this method proved ineffective. Next, the method of alternating projections proved more efficient at solving all test puzzles compared to the backtracking (BT) algorithm. In this research work, a hybrid solver is created using the pencil and paper method and backtracking methods. This method solves all sample puzzles in finite seconds. The execution time outperforms the BT method for most of the sample puzzles.

Keywords

Hybrid, backtracking, pencil and paper, Sudoku, algorithms

1. INTRODUCTION

There are quite a number of Sudoku solving algorithms. These include the pencil and paper method, backtracking, alternating projections, genetic algorithm and simulated annealing. The pencil and paper method also known as human method is very efficient for easy and medium Sudoku puzzles but for hard, extreme and Arto Inkala's most difficult puzzles this method completely fails [2]. The backtracking method provides a solution for all valid Sudoku puzzles however it is very inefficient method as it uses brute force. This research work provides a hybrid backtracking and pencil and paper method, which guarantees a solution for all valid Sudoku puzzles and could do so in a more efficient manner than the backtracking method.

2. LITERATURE REVIEW

This section reviews the strengths and weaknesses of the pencil and paper method and the backtracking method as well as other Sudoku hybrid solvers. Hybrid method that combines Genetic Algorithm with Simulated Annealing (HGASA) [4].

2.1 Pencil and Paper Method

This algorithm tests a puzzle for certain rules that either fills in squares or eliminates candidate numbers. This algorithm is similar to the one human solvers use. These rules include naked singles, pairs, triples, hidden pairs, singles and triples can be found in [3]. The pencil and paper method when implemented as a computer application for simple, valid Sudoku puzzle, obtains a solution speedily as shown in [1]. However, if a valid Sudoku puzzle does not have any property that conforms to one of the rules it becomes unsolvable using the pencil and paper method.

2.2 Backtracking method

The backtracking method moves through empty cells of a Sudoku puzzle tests each candidate or compatible digit of the cell, if there are no violations (i.e. it conforms to the properties of a Sudoku grid), that candidate is placed in the cell. The next

cell is considered, for a suitable candidate which conforms to the properties of a Sudoku grid if one can be found it is placed in that cell, however if none can be found a backtrack is taken to the previous cells and another candidate replaces it to conform to the rules of Sudoku this is illustrated in [5]. The backtracking method guarantees a solution to all valid Sudoku puzzles however, relatively it is more time and memory consuming when compared with the pencil and paper method.

2.3 The strengths and weaknesses of the pencil and paper method and the backtracking method

The table 1 summarises the strengths and weaknesses of the backtracking and pencil and paper methods

	Backtracking method	Pencil and paper method
Strengths	Guarantees a solution and can discover multiple solutions if they exist(carries out an exhaustive search)	If puzzle conforms to one or more of its rules a solution is obtained speedily
Weakensses	It is time and memory consuming	If none of the rules can be applied to the puzzle it fails to obtain a solution

Table 1. Shows a summary of the strengths and weaknesses of the Backtracking method and Pencil and Paper method. An improved hybrid solver of the PnP and BT should combine the strengths of both methods in order to obtain an optimized method.

2.4 Hybrid Genetic Algorithm with Simulated Annealing (HGASA).

Genetic algorithm (GA) is inspired by biological genetics and the Darwinian theory of evolution (survival of the fittest and natural selection) [6]. To apply the GA to the Sudoku problem, first a population is selected, this is a set of potential solutions to the problem, and in this case a set of Sudoku solutions (that is filled Sudoku grids). Each member of this set is referred to as an individual, and each individual is made up of genes. In the Sudoku problem, the genes of an individual can be represented by the integers which fill up the empty cells of the Sudoku grid [5]. Thus if a Sudoku puzzle has 45 empty cells, then each member of the selected population has 45 genes. Next a fitness function is defined, this function determines how close an individual is to the solution. The fittest individuals in the population are selected for mating or reproduction to produce a new individual which is more fit than its parents. The reproduction involves crossover and mutation. The crossover reproduction can carried out by taking a certain percentage from the genes of one parent and

the rest from the genes of the other parent. The mutation involves taking some random genes and replacing them with new integers which comply with the rules for Sudoku grids. Finally, the algorithm terminates when the most optimal solution is attained or once a certain number of generations have been produced.

Annealing is the process by which glass or metal is heated and cooled to remove internal stresses and toughen it. Simulated annealing is an optimization algorithm inspired by the annealing process [6, 7]. Below are the steps involved in simulated annealing [7]

- Initialize an individual's state and energy.
- Initialize temperature.
- Loop until temperature is at minimum.
- Loop until maximum number of iterations has been reached.
- Determine neighbouring state by implementing the neighbourhood function.
- Determine the energy of the current and neighbouring state.
- If the neighbouring state has a lower energy than the current, then change the current state to the neighbouring state.
- Else, if the acceptance probability is fulfilled then move to the neighbouring state.
- Else stick to the current state.
- Keep track of state with lowest energy.
- End inner loop.
- Alter temperature in accordance with the cooling schedule.
- End outer loop

Applying simulated annealing to the Sudoku problem the initial state is obtained by filling the puzzle with integers from 1 to 9 ensuring that in each sub grid no digit appears more than once. The temperature corresponds to maximum cost possible, which can be taken to move from one energy state to another at a particular stage of the annealing process. An energy or cost function can be defined by determining whether an integer is repeated or is not present in a particular row or column [4]. An energy state is assigned to a possible solution based on the number of repeated or non-present integers. The more repeated or non-present integers there are in a solution's rows and columns, the higher the energy assigned to that solution.[4] A neighbourhood function is defined to determine those Sudoku boards which require minimum energy to move to from the current state. A move is made to the neighbour with least energy state or a state which fulfils an acceptance probability. The acceptance probability is a function of the difference in energies between the current and neighbouring states as well as the temperature. The process loops until temperature cools to zero and the solution to the Sudoku problem is attained.

The hybrid Genetic Algorithm and Simulated Annealing (HGASA) algorithm presented in [4], combines the parallel search capability of GA with the flexibility of SA. This algorithm starts with a search using GA, after a specific number of generations the individual Sudoku with best fitness

is selected and the SA algorithm is applied on it until a solution is found. This concept, of applying one method before the other is adapted in this research work to produce a hybrid backtrack and pencil and paper algorithm. However, this method presented in the next section applies the methods alternatingly until a solution is found.

3. THE HYBRID BACKTRACK AND PENCIL AND PAPER METHOD

In the Hybrid Backtrack and Pencil and paper method (HBPNP), the pencil and paper method is first applied to solve the puzzle, if the solution is obtained, the algorithm ends. However, if no solution is obtained, the backtrack method is applied to sub grid 2, 5 & 8 starting from sub grid 2 to 5 to 8, from left to right, and top to bottom. Sub grids 2, 5 & 8 are labelled and shaded in figure 1.

	1	2		3			
	4	5		6			
	7	8		1			

Fig 1. The numbered sub grids of a Sudoku board. The shaded region are sub grids 2, 5 & 8.

The backtracking method, fills the empty cells ensuring that the properties of a Sudoku grid are maintained (i.e. no duplicate digits appear on any row, column or sub grid). With the sub grids 2, 5 & 8 filled, the pencil and paper method is applied to obtain the solution, if the solution is attained, the algorithm is terminated. If not, the backtracking algorithm is applied from the last filled digit in the grid 8. If grids 2, 5 & 8 are filled again, with a new formation of digits, the pencil and paper method is again applied, if the solution is obtained it terminates, otherwise a backtrack is taken again. If the entire search tree is exhausted and the solution is not attained. The HBPNP is restarted, this time the backtracking algorithm is applied to sub grids 2, 5 and 8. Again, each time these sub grids are filled, the Pencil and paper method is applied to obtain the solution, otherwise it backtracks. This alternating application of the backtracking and pencil and paper method continues until the solution is attained. This method guarantees a solution for a valid puzzle and could do so at a faster rate than the backtracking method. The next section evaluates the efficiency of this hybrid method.

4. EVALUATING THE HYBRID BACKTRACKING AND PENCIL AND PAPER (HBPnP) METHOD

The HBPnP method solves all valid puzzles. As it does a backtracking search of sub grids 2, 5 & 8, it must attain the formation in these sub grids which matches the formation in the solution. When this formation is determined, the pencil

and paper method solves all the other sub grids. The HBPnP is implemented in php and the code is available at [11]. The HBPnP was tested with 40 puzzles from [2] and [9]. The minimum execution time taken to solve each puzzle after 3 executions each, on a windows 8, 64-bit system, with 3.0 GB RAM and an Intel Duo CPU (@ 2.40 GHz) using Hybrid Backtracking and Pencil and Paper (HBPnP), Backtracking (BT), Pencil and paper (PnP) are shown in table 2.

Puzzle	Difficulty Rating(S ₁ to S ₃₂ from [2], S ₃₃ to S ₄₀ from [9])	PnP (number of seconds taken to solve the problem)	BT (number of seconds taken to solve the problem)	HBPnP((number of seconds taken to solve the problem)	Difference in seconds between BT execution time and HBPnP(BT(seconds) – HBPnP(seconds))
S ₁	Gentle	0.41	3.01	0.41	2.6
S ₂	Gentle	0.37	0.97	0.37	0.6
S ₃	Gentle	0.37	4.35	0.37	1.9
S ₄	Gentle	0.41	0.87	0.41	0.46
S ₅	Gentle	0.18	4.91	0.18	4.73
S ₆	Moderate	0.41	1.89	0.41	1.48
S ₇	Moderate	0.32	0.42	0.32	0.1
S ₈	Moderate	0.36	2.41	0.36	2.05
S ₉	Moderate	0.26	0.80	0.26	0.54
S ₁₀	Moderate	0.41	2.00	0.41	1.59
S ₁₁	Moderate	0.67	0.76	0.67	0.09
S ₁₂	Moderate	0.66	11.40	0.66	10.74
S ₁₃	Tough	–	13.52	1.03	12.49
S ₁₄	Tough	–	0.61	1.59	-0.98
S ₁₅	Tough	–	8.61	5.48	3.13
S ₁₆	Tough	–	8.29	3.72	4.57
S ₁₇	Diabolical	–	2.37	4.58	-2.21
S ₁₈	Diabolical	–	3.07	0.39	2.68
S ₁₉	Diabolical	–	27.70	1.99	25.71
S ₂₀	Diabolical	–	0.36	1.71	-1.35
S ₂₁	Diabolical	–	22.54	3.55	18.99
S ₂₂	Diabolical	–	2.167	0.79	1.377
S ₂₃	Diabolical	–	4.52	0.72	3.8
S ₂₄	Extreme	–	1.61	0.73	0.88
S ₂₅	Extreme	–	4.47	3.81	0.66
S ₂₆	Extreme	–	1.32	0.85	0.47
S ₂₇	Extreme	–	40.63	0.75	39.88
S ₂₈	Extreme	–	0.16	1.22	-1.06
S ₂₉	Extreme	–	3.63	8.87	-5.24
S ₃₀	Extreme	–	5.31	0.74	4.57
S ₃₁	Extreme	–	5.47	2.46	3.01

S ₃₂	Extreme	–	0.19	0.18	0.01
S ₃₃	Most Difficult (AI Killer)	–	3.52	2.58	0.94
S ₃₄	Most Difficult (AI Lucky Diamond)	–	3.78	9.36	-5.58
S ₃₅	Most Difficult (AI Worm Hole)	–	27.08	1.85	25.23
S ₃₆	Most Difficult (AI Labyrinth)	–	30.0	4.95	25.05
S ₃₇	Most Difficult (AI Circles)	–	15.66	3.31	12.35
S ₃₈	Most Difficult (AI Squadron)	–	18.76	8.29	10.47
S ₃₉	Most Difficult (AI Tweezers)	–	13.64	5.03	8.61
S ₄₀	Most Difficult (AI Broken Brick)	–	2.09	4.03	-1.94

Table 2: shows the number of seconds taken to solve each of the 40 sample puzzles for each of the methods for solving

5. A DISCUSSION OF THE RESULT

Closely analyzing the results in table 2, the HBPnP method solved puzzles S1 to S12 at the same speed of execution as the pencil and paper method. This outcome is expected since the HBPnP method, applies the pencil and paper method first, to the puzzle to obtain the solution, which it did for puzzles S1 to S12. Next, the HBPnP solved puzzles S13 to S40 in a definite number of seconds while the pencil and paper was unable to obtain solutions to these puzzles. So, HBPnP performed better than the PnP method in these cases. Comparing the BT method with the HBPnP, for all puzzles the HBPnP solved the puzzles in fewer seconds than the BT method, except for puzzles S14, S17, S28, S29, S34 & S40, the row for these puzzles are shaded in table 2. This can be clearly seen in the graph in figure 2. The HBPnP method in grey, solves puzzles in less time than BT in red, for most puzzles. The graph of PnP method is indicated in blue, which stops abruptly at s12.

The average time taken to solve the 40 puzzles using HBPnP was 2.28 seconds, puzzle S5 and S32 were solved in the least number of seconds at 0.18 seconds and S29 took the most time to get solved at 8.87 seconds. The BT took an average of 7.62 seconds to solve the 40 puzzles. Minimum execution time of 0.16 seconds recorded at S28 and maximum execution time of 40.26 seconds recorded at S27. On average the HBPnP solved each puzzle in approximately 5 seconds less time than the BT method. Next a null hypothesis is carried out to determine if there is a significant difference between the average execution time for BT and HBPnP. The hypothesis H₀ is assumed, that there is no difference between the mean

Sudoku puzzles, Pencil and Paper, backtracking and Hybrid backtracking - pencil and paper method.

execution time of BT and HBPnP and the results obtained are shown in table 3.

	BT	HBPnP
Mean	7.621675	2.28675
Variance	95.14481	6.015863526
Observations	40	40
Hypothesized Mean Difference	0	
Df (degrees-of-freedom)	44	
t Stat	3.354691	
P(T<=t) one-tail	0.000822	
t Critical one-tail	1.68023	
P(T<=t) two-tail	0.001644	
t Critical two-tail	2.015368	

Table 3: t-Test: Two-Sample Assuming Unequal Variances with 40 observations

In table 3, the result of the t-test can be seen. If t Stat < - t Critical two-tail or t Stat > t Critical two-tail, the null hypothesis is rejected [12], and this is the case in this analysis, t Stat > t Critical two-tail. Thus the null hypothesis is rejected. This means that from the analysis, it is convincing enough to say that the average time BT takes to solve the sample puzzles differs significantly with that of HBPnP.

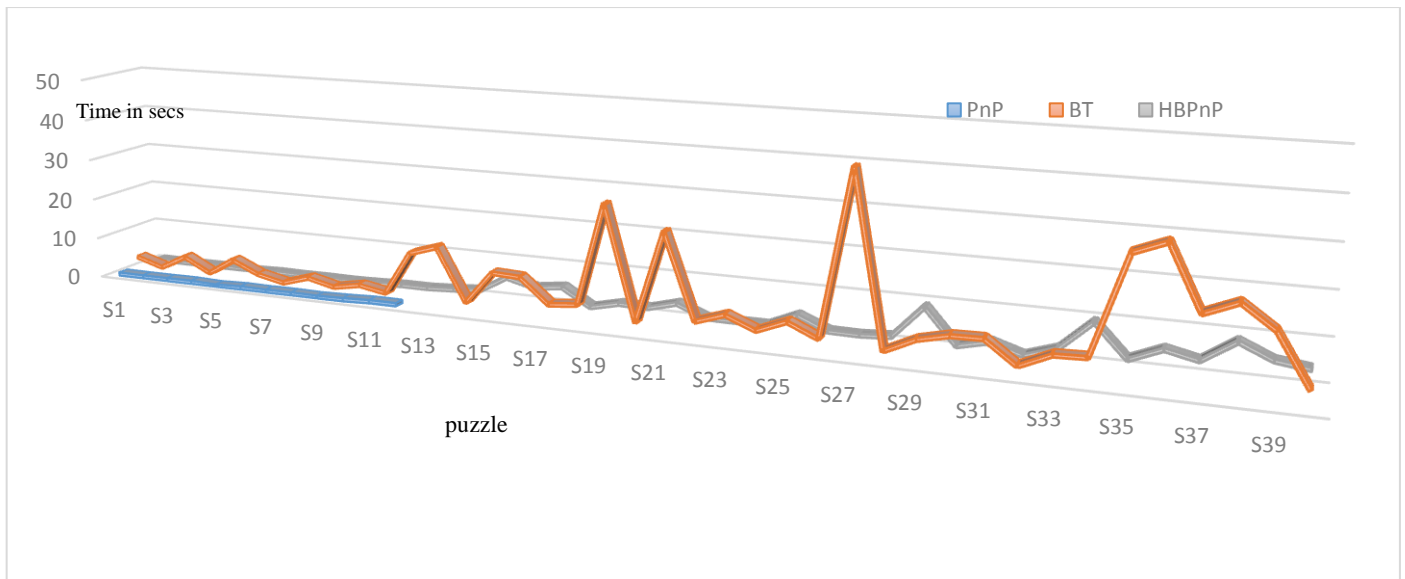


Fig 2 Graph shows the number of seconds taken to solve each puzzle 1 to 40 using PnP, BT and HBPnP

6. CONCLUSION AND FUTURE WORK

In conclusion, it is clear that the HBPnP method is an improvement to the PnP method. However, when HBPnP is compared with the BT method, for most cases the former performs better than the later, in terms of execution time. The 5 sample puzzles in which the BT took less time to solve than the HBPnP method, indicate that there is a need to fine tune or improve the HBPnP method to ensure it always performs better than the BT method in solving all valid puzzles. This fine tuning work will be part of work to be done in the future.

Since the HBPnP combines a human method (PnP) and a guessing method (Backtracking), can it be used to rate difficulty of Sudoku puzzles? This is a question that needs to be answered in future.

7. REFERENCES

- [1] Onokpasa, E. & Bakwa, D. (2019) *A Comparison of Three Sudoku Solving Methods*. International Journal of Computer Applications. 10.5120/ijca2019918439
- [2] www.sudoku.org.uk (2018) <http://www.sudoku.org.uk/Daily.asp> Accessed:20/11/2018
- [3] [Learn-Sudoku.com](https://www.learn-sudoku.com/)(2008) <https://www.learn-sudoku.com/> [Accessed 20/09/18]
- [4] Perez, M. and Marwala, T., (2011) *Stochastic optimization approaches for solving Sudoku*, Proc. IEEE Congress on Evolutionary Computing, pp 256–279, Vancouver: Piscataway.
- [5] Chi E. & Lange K. (2012) *Techniques for Solving Sudoku Puzzles*, pp 2,3 arXiv:1203.2295 [math.OC]
- [6] Cantú-Paz, E. & Goldberg, D.E. (2000). *Efficient parallel genetic algorithms: theory and practice*. Computer Methods in Applied Mechanics and Engineering, 186 (2-4), 221-238.
- [7] Kirkpatrick, S., Gelatt, C.D. & Vecchi, M. P. (1983). *Optimization by Simulated Annealing*. Science, 220(4598), 671-680.
- [8] Van Laarhoven, P. & Aarts, E. (1987). *Simulated Annealing: Theory and Application*. D Reidel Publishing Company, the Netherlands.
- [9] [www.aisudoku.com](http://www.aisudoku.com/index_en.html)(2006) http://www.aisudoku.com/index_en.html Accessed:20/11/2018
- [10] Schaad J. (2010) *Modeling the 8-Queens Problem and Sudoku using an Algorithm based on Projections onto Nonconvex Sets*, Master's thesis, The University of British Columbia. Pages 23, 93-95
- [11] Google(2019) <https://drive.google.com/open?id=1C9ym99n7IXsQASwmZcLVDo57SGpxIX-E>.
- [12] Kock, N. (2015) *One-tailed or two-tailed P values in PLS-SEM?* International Journal of e-collaboration, 11(2), 1-7.