Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Sudoku Solver – A Novel Approach Using Recognition of Printed Text Image

Manisha SanthanaGopalaKrishnan
Independent Researcher, Toronto, Canada
Corresponding Author: Manisha SanthanaGopalaKrishnan, **E-mail**: manishassg@gmail.com

| ABSTRACT

Detecting and analyzing text in multimedia is required for recognition of characters which can be used for performing multi-objectives in numerous research areas and applications. Characters or numbers in images can be hand-written, printed document, typewritten or artificially added for the purpose of searching or indexing, editing or image understanding. Various applications such as License Plate Detection, Handwritten documents recognition, Sudoku Solving, multi-oriented recognition focuses on character recognition. Image Sudoku solver works on Sudoku puzzle images. In this paper, the printed text Sudoku image is used as the input. The image is enhanced by filling the horizontal and vertical gaps in the grids that may occur due to shadow or distortion of the scanned image such that all the 81 grids are extracted. Secondly, the numbers and grids in the enhanced image are extracted and a dancing link approach using Algorithm X is adopted to solve the Sudoku. The proposed work is compared with a backtracking approach and dancing link approach without enhancing the image and its computation time for various ranks of difficulty of a puzzle taken. The Sudoku puzzle can be used to secure information stored in an image as the number of solutions to solve a Sudoku puzzle is very large, allowing it to be used as a security tool resistant to brute force attacks.

| KEYWORDS

Sudoku solver, Text detection, Recognition, Printed text, Dancing link.

| ARTICLE INFORMATION

ACCEPTED: 01 August 2024 **PUBLISHED:** 01 September 2024 **DOI:** 10.32996/jcsts.2024.6.4.6

1. Introduction

Optical Character Recognition (OCR) is a technique used to read, analyze and understand the characters in multimedia which can be used for indexing, searching, editing or understanding an image. The text can be in the form of printed, handwritten or caption text. The OCR application can work in online or offline mode. In the offline mode, the input is an image or a video frame that is processed and the text is recognized. In the online mode, the processing of the image and the recognition of the text characters are done simultaneously. Various applications of OCR are navigation assistance system, sudoku solver [Dutta, 2016, Ly, 2015], sport video analysis, content based image retrieval (CBIR), license plate recognition etc., [Jain, 2015]. Sudoku is a number game popularized with a 9 X 9 grid with few squares of the grid initially filled with numbers 1 to 9. The objective of the game is to fill the remaining empty grids from numbers 1 to 9, such that no number should be repeated in a single row or column or a 3 X 3 box. A sample Sudoku problem is given in Fig 1.

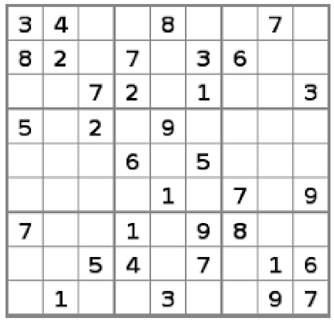


Figure 1: A sample Sudoku puzzle

The Sudoku solver is an NP-complete problem. The solution to a grid is valid, when there is only one way in which the numbers can be filled for the given clues. The least number of clues required to solve a Sudoku is 17. Few logic techniques are used by researchers to solve the problem. The level of difficulty for a Sudoku puzzle can be ranked into 5 major categories (extremely easy, easy, medium, hard and evil) based on the number of clues and the position of each empty cell [9]. The comparison chart of the number of clues and the rank of difficulty are given in Table 1. Similarly, Table 2 gives the comparison of the minimum number of clues in each row and column in a Sudoku instance and their corresponding rank of difficulty.

Difficulty level	Number of Clues (out of 81)
1 (Extremely Easy)	46
2 (Easy)	36
3 (Medium)	32
4 (Difficult / Hard)	28
5 (Extremely Hard / Evil)	17

Table 1: Comparison chart of the number of clues and the rank of difficulty [9]

Difficulty level	Number of Clues (out of 81)
1 (Extremely Easy)	5
2 (Easy)	4
3 (Medium)	3
4 (Difficult / Hard)	2
5 (Extremely Hard / Evil)	0

Table 2: Comparison chart of the minimum number of clues in a row and column and the rank of difficulty [9]

The paper is organized as follows. The related research work is presented in section II. The various algorithms to solve a Sudoku are discussed in section III. The proposed enhanced DLX algorithm and its working principles are explained in section IV. Section V discusses the experimental results of the proposed algorithm and its comparison with the state-of-the art methods and also in terms of time complexity. The conclusion and future research directions are written in section VI.

2. Literature Review

The basic logic used to solve a Sudoku puzzle is using the depth-first search (DFS) and backtracking [Kamal, 2015]. Most of the existing approaches use backtracking combined with other approaches to solve a puzzle [Kamal, 2015, Maji, 2014, Schottlender, 2014]. The brute-force approach uses filling of numbers in a cell based on random filling and ordered filling. The Fisher et al., [n.d] discusses on sorting the array of elements randomly. But the main drawback of this algorithm is that, the brute-force approach is time consuming and is harder for puzzles with higher degree of difficulty. However a solution to the puzzle is guaranteed. This is an iterative approach and the solution becomes deterministic and recursive. The technique of permutations can also be used to randomly generate possible numbers to be filled in the empty cell.

A pencil and paper algorithm uses preemptive sets to assign values to an empty cell. These numbers are termed differently by different authors. The markup numbers are named as partnerships and number sharing. These pre-emptive sets can be used to solve the puzzle at various degrees of difficulty as well as it is easier to scale the size of the puzzle.

A backtrack based enumerative algorithm is proposed using Graph Referencing method (GRA) [Chakraborty, 2014]. Genetic Algorithms are found effective in solving the puzzle as a multi-objective optimization problem [Darwin,1968, Mantere, 2007]. Meta-heuristic algorithms such as Harmony search use continuous optimization to solve the puzzle. This works by using the musical process of search to identify the perfect state of harmony to give better solutions for combinatorial optimization problems. The efficient modification on the Harmony search algorithm proposed by Mandal et al., [2011] is effective in the number of iterations required to solve a puzzle. The algorithm using Simulated Annealing (SA) technique is developed based on the puzzle solving strategy and GA based algorithm also suggests better options in solving a given Sudoku puzzle [Mandal, 2011, Vikstén, 2013, Volume, 2010].

3. State-of-the art Sudoku Solver Techniques and Algorithms:

Although a Sudoku puzzle with 81 squares has a complete solution, there are 6,670,903,752,021,072,936,960 valid Sudoku grids. The basic logic used to solve a Sudoku puzzle is using the depth-first search (DFS) and backtracking [Kamal, 2015]. Therefore, for a Sudoku puzzle that is ranked difficult, the basic DFS algorithm is very slow to solve. The brute-force and the heuristic are the major approaches for solving a Sudoku puzzle [Taruna, 2015]. Some of the Sudoku solver techniques and algorithms discussed in this section are: brute-force approach, pencil and paper approach and rule-based approach.

3.1 Brute-force approach:

Brute-force approach is the most common algorithm used for solving a Sudoku puzzle. It is an exhaustive search algorithm as it finds all possible solutions for a given problem and but is not considered effective [Taruna, 2015]. The idea of the brute-force approach is to fill an empty cell with a number without violating the rules, and proceeds with the next empty cell and repeats the process until a violation occurs. On violation it backtracks to the previously filled cell and fills the next possible number. The backtracking approach can be categorized into ordered choices and random choices. In the ordered choices based backtracking approach, the numbers are ordered in an array which can be sorted or shuffled. For example, the number to fill the first empty cell is 1 to 9, and the next empty cell is 1 to 9 minus the number filled in the first empty cell and so on therefore giving a deterministic number of backtracking and recursions. In the random choice based backtracking approach, Fisher – Yates shuffle [n.d] is used to shuffle the choices in the array randomly. This leads to a non-deterministic number of backtracks and recursions to solve a Sudoku. However, choosing the right value for the first empty cell leads to fewer backtracks and recursions thus optimizing the backtracking algorithm [Maji, 2014].

The advantages of this method are:

1. Solution is guaranteed for a valid puzzle.
2. The time taken to solve the puzzle is independent to the difficulty ranking.
3. Simple approach for execution compared to many algorithms used to solve a difficult puzzle.

The major disadvantage of the method is the time taken to solve a complex puzzle is slow compared to other methods that use backtracking after decomposing the problem into less complex problem [Schottlender, 2014].

3.2 Pencil and Paper approach:

The pencil and paper algorithm works such that given a partially filled puzzle, all possible numbers that can be filled in the empty cells without violating the rules of Sudoku solving. These possible numbers are termed as the markup numbers. If a particular markup number appears only once (singleton) in a cell in the entire row, column and the box, then that number can be assigned to that particular cell. This can be done using the pre-emptive sets, occupancy theorem, hidden sets and random choice [Schottlender, 2014]. By filling all such singleton numbers, the size of the problem is reduced. The same process is repeated until a violation occurs. On violation, all the markings are undone and backtrack to the previously non-violating cell and proceed again [Michael, 2016]. If a number gets filled using any of the above said approaches and does not violate the rule, then the markup numbers in the remaining cells are recalculated and the process is repeated until the solution for the puzzle is complete without any violations.

Algorithm 1: Pencil and Paper approach
(1) Find all forced numbers in the puzzle.
(2) Markup the puzzle (and attempt to identify remaining singletons).
(3) do
Search for pre-emptive sets in rows, columns and boxes, make corrections
continue step 3
while (solution == TRUE)
(4) if (solution == FALSE)
make random choice
if (solution == TRUE)
Return
else
violation backtrack to step 4.

The advantage of the algorithm is that apart from finding the solution to the puzzle, they can also rate the difficulty of a puzzle. The major disadvantage is that these algorithms are comparatively slower and it is important that the forced random choice used to fill a cell is correct to get a solution for the puzzle.

3.3 Rule based Approach:

The rule based approach works based 7 rules that are derived from the existing approaches [Gustavo, 2007]. They mainly work on the principle of elimination of numbers in a particular empty cell based on the partially filled puzzle. The list of rules and their explanation are given below.

Rule 1: First order simplification rule – If only one number between 1 to 9 is possible in a particular empty cell, then that number can be removed from all other empty cells in that particular row, column or box.

Rule 2: Second order simplification rule – If two cells in a row, column or box has only two possible numbers, then those numbers can be removed from all other empty cells in that row, column or box.

Rule 3: Third order simplification rule - If three cells in a row, column or box has only three possible numbers, then those numbers can be removed from all other empty cells in that row, column or box.

Rule 4: Only one number rule – If a number is not possible in a cell except one, and the number of possible numbers in that cell is greater than one, then the number is called a singleton number.

Rule 5: Only two number rule – If two numbers n1 and n2 are not possible in a cell except two, and the number of possible numbers in that cell is greater than two, then the possible set of numbers are n1 and n2.

Rule 6: Twin rule – If a number is possible in a cell in one row or column, then that number can be removed from the list of all possible numbers to be filled in that row or column, but not in the other boxes.

Rule 7: Sudoku split rule – If none of the other rules can be applied, then the puzzle is split. A cell with minimum number of possibility is taken as a sub problem and is solved using the rules 1 to 6.

Algorithm 2: Rule Based Approach

```
(1) if (only one number == TRUE)
        enter a number between 1 to 9 is possible, remaining = num-1
(2) else if (only two numbers == TRUE)
        enter two numbers between 1 and 9 is possible, remaining = num-2
(3) else if (only three numbers == TRUE)
        enter three numbers between 1 and 9 is possible, remaining = num-3
(4) else if (one number == FALSE)
        that number between 1 to 9 is not possible, remaining = num-1
(5) else if (two numbers == FALSE)
        two numbers between 1 to 9 are not possible, remaining = num-2
(6) else if (two numbers in two cells == TRUE)
            in a row or column two numbers are possible in two cells (cell1 and cell2 = num1|num2)
(7) else
      problem = sub problem
      goto step 1.
```

The major disadvantage of the rule based approach is that multiple solutions are possible and the derived solution is dependent on the order of execution of these rules thus leading to a non-confluent solution. Therefore, it is important to identify the right method to solve the puzzle. Main application is that these puzzles can be used as a security tool for data hiding and transmitting the information securely, since the number of possible keys to decrypt the message is equal to the number of solutions to the puzzle transmitted [Thai-Son, 2015].

4. Methodology

Sudoku puzzle is a NP complete problem, however the popular 9 X 9 puzzle has finite solution. In general, the puzzle is considered an exact cover problem. The exact cover problem is a decision problem used to find the exact cover. The idea of the exact cover problem is to decide an element from a set of elements, such that an element can be used only once throughout the problem. The proposed work, performs the following steps: preprocessing, finding the square boundaries, finding the total grids and recognition of digits and solving the Sudoku puzzle.

4.1 Preprocessing:

The input Sudoku image taken is from a printed document, therefore there is a need to perform few pre-processing steps to avoid issues that occur due to shadowing, distortion etc., The image is filtered using a median filter and is binarized using Otsu's thresholding to produce a binary image. The intensity of all the pixels are equalized using these filters. The vertical and horizontal edges are extracted and these edge points are connected using vertical and horizontal gap filling approach [Manisha, 2017]. Every pixel is compared with its nearby pixel and if the intensity of a pixel varies from its neighbouring pixels, then the value of the pixel is replaced by the mean of its nearby pixels [Manisha, 2017]. The importance of the pre-processing step can be understood better in finding the total grids. For a 9X9 puzzle, the pre-processing step improves in finding all the 81 grids, which is the novelty of the proposed work.

4.2 Finding square boundaries:

The square boundaries are interpreted by identifying the connected components in the image. These connected components are classified as the 3X3 boxes from the 81 grids. The identification of these square boundaries contributes in identifying the centres of the components.

4.3 Finding grids and recognition of digits:

The component centres from the previous step helps in finding the grids such that it works well for any transformation such as rotation, distortion or translation. The coordinates of all the grid points are identified thus breaking down the grid into 81 individual squares [Dutta, 2016]. These squares can be classified as numbers between 1 and 9 or can be an empty square that needs to be solved. These digits can be recognized using OCR and the result is fed into a Sudoku solver [Ly, 2015].

4.4 Sudoku Solver – DLX using Algorithm X:

Dancing Links (DLX) method is used to implement Algorithm X efficiently. This method works such that, any known data is taken as the initial condition and the corresponding links are calculated with their neighbouring cell. If any link does not correspond to the initial condition, then the link is attached to the suitable cell. The algorithm works using doubly linked list approach. A depth-first search is done throughout the cells that are linked to the initial condition and the possible links are attached to these conditions.

The Algorithm X is a non-deterministic, recursive algorithm that uses DFS for backtracking. This is used to identify ideal solutions for any exact cover problem. In Sudoku puzzle, there are four major constraints that constitute into an exact cover problem with 81 grids. The position constraint is that only a single value between 1 and 9 can be allocated in a cell. The row constraint is that, any number can be assigned to only a single cell in the entire row. The column constraint is that, any number can be assigned to only a single cell in the entire column. The region constraint is that, any number can be assigned to only a single cell in the entire region of 3X3 box. Before assigning a value to a cell the position constraint should have the highest priority and the row, column and region constraints have equal priority.

Algorithm 3: Algorithm X

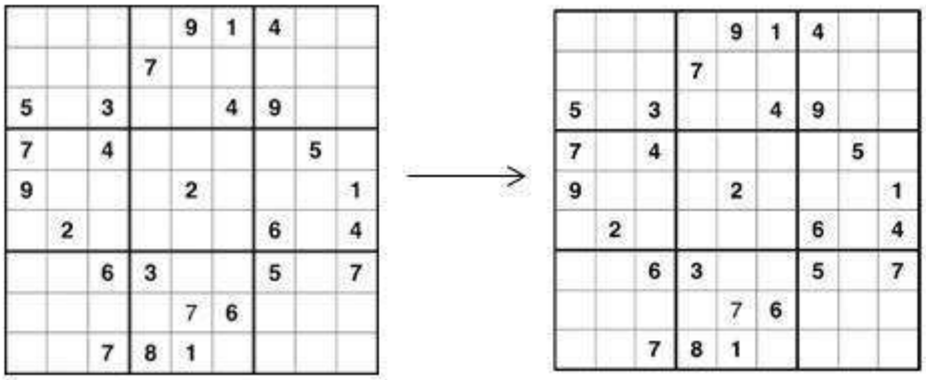
```
(1)  if j in A[i][j] ==0
         return

     else if A[i][j] ==1

         include A[i] in partial solution

(2)  for (j in Ai,j ==1)
         for (i in Ai,j ==1)

             delete A[i] in A and A[i][j] =A[x][j]

         delete A[j] in A and A[x][j] = A[x][y]

(3)  goto step 1on the reduce matrix A[x][y].
```

5. Experimental Results

In this section, the experimental setup and the results obtained by the proposed method is discussed and the performance of the proposed method is compared with the state-of-the art approaches such as brute-force approach and rule based approach in terms of time taken to solve a puzzle. Here the noise in the input image is removed using a median filter and Otsu's thresholding is applied and the image is converted into a binary image by equalizing the intensity of the image throughout and edges are filled using horizontal and vertical gap filling technique.

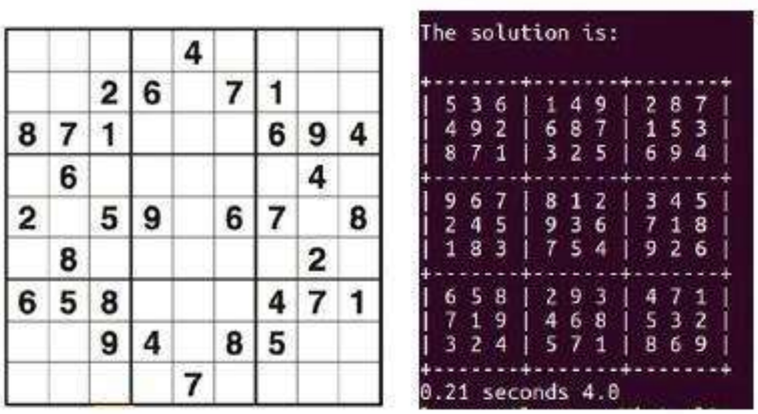
Figure 2 consists of the input image and its corresponding pre-processed image after filling the edges. Figure 3 shows the sample input image and its corresponding solution using brute-force approach. In figure 4, the sample input image and its corresponding solution using the proposed method is given. On comparing figure 3 and figure 4 it is evident that the proposed method outperforms the brute-force method in terms of the computation time taken for solving the puzzle at the same level of difficulty ranking.

Figure 5 shows the failure case of the DLX algorithm as the total number of contours obtained is greater than the actual number of contours available in the puzzle. Thus it clearly assures the need for pre-processing the input image before feeding into the algorithm for solving the puzzle. The comparison graph on the total computation time for solving the puzzles at various degrees of difficulty using the proposed method with the state-of-the art methods [Thenmozhi, 2017, Sankhadeep, 2015, Akta, 2015] is given in figure 6. This further confirms that the DLX algorithm using Algorithm X gives finite solutions to the puzzle and is efficient than the other approaches.



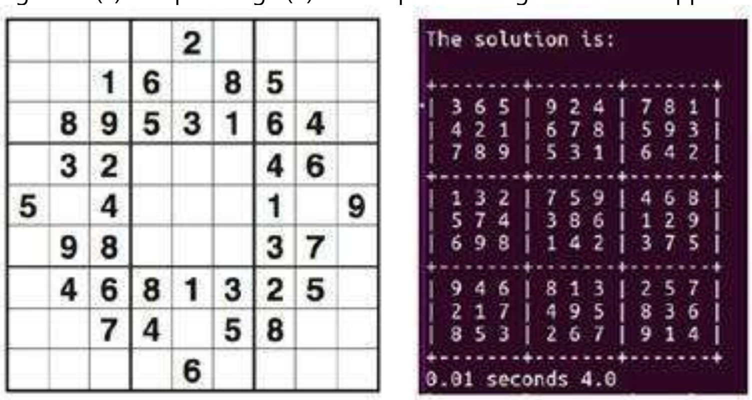
(a) (b)

Figure 2: (a) Sample input image (b) pre-processed image



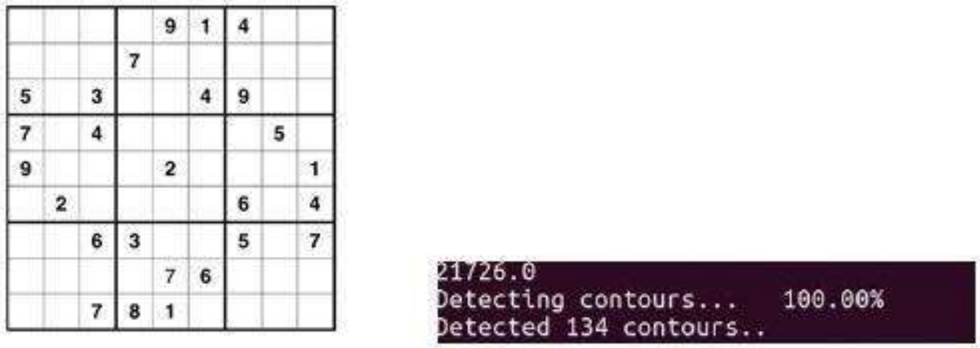
(a) (b)

Figure 3: (a) Sample image (b) Solved puzzle using brute-force approach



(a) (b)

Figure 4: (a) Sample image (b) Solved puzzle using proposed method



(a) (b)

Figure 5: (a) Sample image (b) Failure case in finding contours

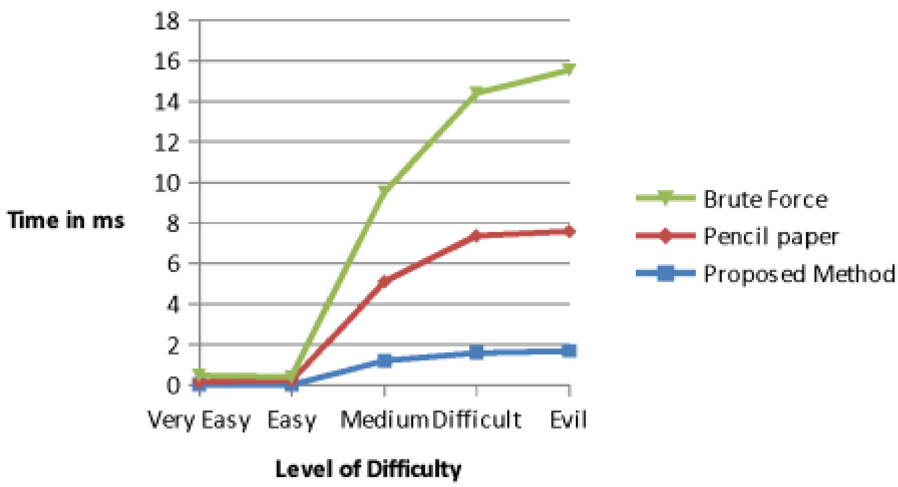


Figure 6: Comparison of proposed method with brute-force and pencil paper method

The novelty of the proposed method is twofold. Firstly, the proposed method affirms the importance of pre-processing the input image containing the printed text. The finding of all the grids in a puzzle is an important aspect in solving the puzzle. This paper evaluates the performance of the DLX algorithm using Algorithm X without enhancing the input image and it is evident that there is a possibility of not identifying all the grids due to distortion or a shadow in the input image taken. Secondly, the comparison of the proposed method is done with the state-of-the art approaches such as brute-force approach and rule based approach for different degrees of difficulty ranking in terms of computation time. This further clearly enhances the significance of the proposed work over the existing approaches.

6. Conclusion

The objective of this study was to evaluate the efficiency of the DLX method utilizing Algorithm X for solving Sudoku puzzles. The key findings indicate that this method effectively solves Sudoku puzzles and highlights the significance of pre-processing the input image to accurately identify the 81 grid cells, which is crucial for solving the puzzle. The study also compares this approach with state-of-the-art techniques in terms of computation time, demonstrating that the time required to solve the puzzle is influenced by the number of initial clues provided. Despite the NP-complete nature of the Sudoku problem, the proposed method successfully confirms that a 9x9 puzzle can be solved within a finite timeframe. Limitations of the study include the reliance on initial clues and the focus on a single puzzle size. Future research could explore the application of Sudoku in steganography and investigate methods for improving efficiency with different puzzle configurations and clue densities.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.
ORCID iD (if any) https://orcid.org/0009-0004-8590-9199
Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Akta A M., and Padma B, (2015). Study on the Performance Characteristics of Sudoku Solving Algorithms, *International Journal of Computer Applications,* 122(1), 2015, 10 -12.
[2] Chakraborty, R., Paladhi, S., Chatterjee, S., and Banerjee, S. (2014). An Optimized Algorithm for Solving Combinatorial Problems using Reference Graph, *IOSR Journal of Computer Engineering*, 16(3) 1-7 (2014)
[3] Crook J.F., (n.d). A Pencil-and-Paper Algorithm for Solving Sudoku Puzzles, Notices of the AMS, 56(4), 460 – 468.
[4] Darwin, C. (1968). The Origin of Species: By Means of Natural Selection orThe Preservation of Favoured Races in the Struggle for Life, OxfordUniversity Press, London, 1859, A reprint of the 6th edition (1968)
[5] Dutta and Ghosh A., (2016). Development of a character recognition software to solve a Sudoku puzzle, 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, 2016, 1-5.
[6] Fisher R, and Yates F. (n.d) Statistical tables for biological, agricultural and medical research (3rd ed.). London: Oliver & Boyd. 26–27. OCLC 14222135
[7] Gustavo S, and Miguel P, (2007) Solving Sudoku Puzzles with Rewriting Rules, In Electronic Notes in Theoretical Computer Science, 176, 4, 2007, 79-93, ISSN 1571-0661, https://doi.org/10.1016/j.entcs.2007.06.009.
[8] Jain and Sharma J., (2014) Classification and interpretation of characters in multi-application OCR system, 2014 International Conference on Data Mining and Intelligent Computing (ICDMIC), New Delhi, 2014, 1-6.
[9] Kamal S., Chawla S. S. and Goel N., (2015) Detection of Sudoku puzzle using image processing and solving by Backtracking, Simulated Annealing and Genetic Algorithms: A comparative analysis, 2015 Third International Conference on Image Information Processing (ICIIP), Waknaghat, 2015, 179-184.
[10] Ly T. M. and Vo D. T., (2015) A novel and automatic character extraction and recognition for Sudoku puzzle solving, 2015 International Conference on Advanced Technologies for Communications (ATC), Ho Chi Minh City, 2015,546-550.
[11] Lee, S, K.., and Geem, W, Z. (2005) A new meta-heuristic algorithm for continuous engineering Optimization, Comput. Methods Appl. Mech. Engrg, 194, 3902–3933 (2005).
[12] Liu, Y. and Liu, S. (2010) Algorithm based on genetic algorithm for Sudoku puzzles, Computer Science, 37(3), pp. 225-226 (2010).
[13] Mandal, S., and Sadhu, S. (2011) An Efficient Approach to Solve Sudoku Problem by Harmony Search Algorithm, An *International Journal of Engineering Sciences* ISSN: 2229-6913 Issue September 2011, 4, 312-323 (2011)
[14] Michael H, (2016) Reducing the generalised Sudoku problem to the Hamiltonian cycle problem, In *AKCE International Journal of Graphs and Combinatorics,* 13, 3, 2016, 272-282, ISSN 0972-8600, https://doi.org/10.1016/j.akcej.2016.10.001.
[15] Maji K. and Pal R. K., (2014) Sudoku solver using minigrid based backtracking, 2014 IEEE International Advance Computing Conference (IACC), Gurgaon, 2014, 36-44. doi: 10.1109/IAdCC.2014.6779291
[16] Manisha S, and Sree S T, (2017) TEXT FRAME CLASSIFICATION AND RECOGNITION USING SEGMENTATION TECHNIQUE, 2nd ICATCCT, IEEE, 2017
[17] Manisha S, and Sree S T, (2017) Effective Printed Tamil Text Segmentation and Recognition using Bayesian Classifier, LNCS, Springer, 2017.
[18] Mantere, T., and Koljonen, J. (2008) Solving, rating and generating Sudoku puzzles with GA, Proc. of IEEE Congress on Evolutionary Computation (CEC 2008), IEEE, New York, pp. 1382-1389 (2007).
[19] Schottlender M., (2014) The effect of guess choices on the efficiency of a backtracking algorithm in a Sudoku solver, IEEE Long Island Systems, Applications and Technology (LISAT) Conference 2014, Farmingdale, NY, 2014, 1-6. doi: 10.1109/LISAT.2014.6845190
[20] Sankhadeep C, Saubhik P, and Raktim C, (2015) A Comparative Study On The Performance CharacteristicsOf Sudoku Solving Algorithms, *IOSR Journal of Computer Engineering*, 16(5), 2015, 69-77.
[21] Thai-Son N, and Chin-Chen C, (2015) A reversible data hiding scheme based on the Sudoku technique, In Displays, 39, 2015, 109-116, ISSN 0141-9382, https://doi.org/10.1016/j.displa.2015.10.003.
[22] Thenmozhi M., Palash J, Sai A R, and Saketh R B, (2017) Analysis of Sudoku Solving Algorithms *International Journal of Engineering and Technology (IJET)* , 9(3), 2017, DOI: 10.21817/ijet/2017/v9i3/170903043
[23] Taruna K, and Preeti y, L, (2015) Study of Brute Force and Heuristic Approach to Solve Sudoku, *International Journal of Emerging Trends & Technology in Computer Science*, 4(5), 2015.
[24] Vikstén, H., and Mattsson, V (2013) Performance and Scalability of Sudoku Solvers, Bachelor's Thesis at NADA, Royal Institute of Technology, University in Stockholm, Sweden (2013)
[25] Volume 6146, (2010). 461-467. Sudoku Using Parallel Simulated Annealing, Zahra Karimi-Dehkordi, Kamran Zamanifar, Ahmad Baraani-Dastjerdi, Nasser Ghasem-Aghaee, First International Conference, ICSI 2010, Beijing, China, Proceedings, Part II, June 12-15, (2010).