e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

# GENERATING SUDOKU PROBLEMS FOR DIFFERENT DIFFICULTY LEVELS WITH AN AUTO SOLVING FEATURE

**Akshay Kamkhalia[*1], Vicky Jha[*2], Aarti Ghagre[*3], Vaishnavi Amage[*4], Sayalee Narkhede[*5]**

[*1,2,3,4,5]Student, Department Of Computer Engineering, Terna Engineering College, India.

## ABSTRACT

Sudoku puzzle game is one of the most famous and most played puzzle games in the world. This popularity attracts Researches to constantly evolve techniques to generate Sudoku problems based on different difficulty levels. determining the difficulty of a Sudoku problem is indeed a challenging task. The objectives of this paper are to discuss different methods of generating Sudoku problems based on the difficulty levels and understanding the problem generation phase of the game. Our proposed project successfully fulfills the problem generation phase of the game by generating Sudoku instances based on different difficulty levels and an auto solver feature to solve the generated problems

**Keywords:** Sudoku Problem Generation, Puzzle Generation, Digging Hole Strategy, Sudoku, Difficulty Level, Self-Solver, Sudoku.

## I. INTRODUCTION

Sudoku puzzles have grown extremely popular among players of all cerebral levels all around the world. In this chapter, we'll look at alternative methods for constructing a large number of Sudoku cases with variable levels of difficulty, based solely on the number of clues rather than their position. A Sudoku instance can be created in a variety of ways. The most popular method is to take a solved Sudoku puzzle and remove some of the numbers from the cells based on the required difficulty level, then check whether the created Sudoku instance has a unique solution after each removal, though no such technique has yet been discovered that tells about the uniqueness of a solution. the most popular method is the digging hole strategy. creates Sudoku puzzles. Other methods include digit exchanging, rows-in-a-band exchanging, band exchanging, stack exchanging, and the creation of a Sudoku instance from an existing Sudoku instance using shuffling techniques like digit exchanging, rotation, rows-in-a-band exchanging, band exchanging, stack exchanging, and the. A solid comprehension of the difficulty level is very crucial for building an instance of the problem according to different levels of complexity. So, in the next section, we'll go over the various levels of difficulty in the Sudoku puzzle.

**Proposed system**

1. Read the input file.
2. Search patterns.
3. Provide a solution.
4. Resolve Backwards.
5. Compare working time.

When the user clicks to play the game, he/she will get an option to choose the grid size between 4x4 and 9x9 which the user can choose based on his wish. The next option encountered will be the choice between difficulty levels between easy, medium, and hard. Here the choice between difficulty levels and grid size is important as it will decide the game further based on the user's past experiences with the game. Next, the user gets his selected choices to confirm and is later taken to the screen where he can solve and play the game by his capabilities. In another case where the user fails to find the solution on its own, he can opt for the self-solver feature where he will see the cell by cell solution of the entire game with a backtracking algorithm. and hence can get his problem solved.

**The Total Amount of Given Cells**

The total number of supplied cells in an initial Sudoku problem is the first component that influences level estimation. Three constraints in the game rule can drastically reduce the number of possible digits in each cell, ensuring that each row, column, and mini-grid include 1 through 9 exactly once. In general, it's logical to

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

assume that the emptier cells (and fewer hints) offered at the start of a Sudoku game, the higher the puzzle's level. For each level of difficulty, different studies have moderately scaled the number of ranges of givens; in summary, such data are shown in Table 5.1. Table 5.1: The amount ranges givens in each difficulty level.

| Level | Number of Clues |
|---|---|
| (Easy) | 36-46 |
| (Medium) | 32-35 |
| (Difficult) | 28-31 |

**Digit Exchanging**

It is simple to accomplish the method of digit exchanging because what is necessary here is to exchange all the digits in the cells of one existing Sudoku instance in some well-defined fashion.

To a surprise when this process is executed this will not have an impact on the uniqueness of the Sudoku problem. Thus a new instance of the Sudoku puzzle is produced.

Now, the same method can be applied to multiple pairs of numbers that need to be exchanged. Figure 5.8 depicts a newly generated Sudoku instance derived from the Sudoku instance shown in Figure 5.6, in which values 1 and 9, 5 and 6, and 7 and 8 are pair-wise exchanged to create a new Sudoku instance. In the diagram, all of the replacements are highlighted.

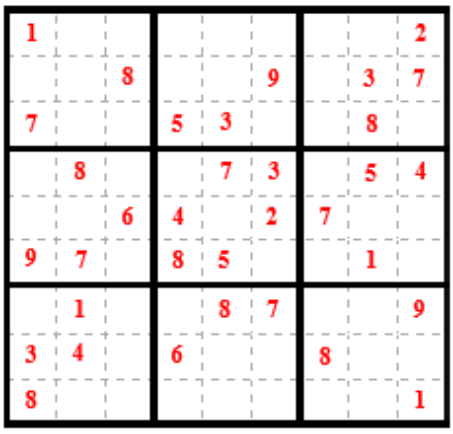



**Figure 5.6:** An instance of Sudoku puzzle.

A Sudoku puzzle is depicted in Figure 5.6. Let's replace all of the 1s in this puzzle with 9 and vice versa. Then, as shown in Figure 5.7, the modified Sudoku instance can be created. In this diagram, all 9s in cells [6,1], [2,6], and [7,9] have been replaced by 1, whereas all 1s in cells [1,1], [7,2], [6,8], and [9,9] have been replaced by 9. All of these substitutions are highlighted in the diagram. As a new Sudoku instance is generated, this process of exchanging digits is valid throughout the puzzle, and it is valid for exchanging any number of values within the puzzle.

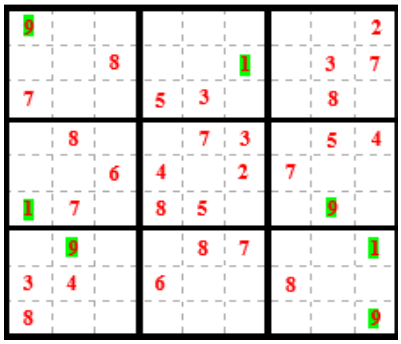



**Figure 5.7:** A new instance is generated from the Sudoku instance shown in Figure 5.6 after interchanging 1 and 9

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

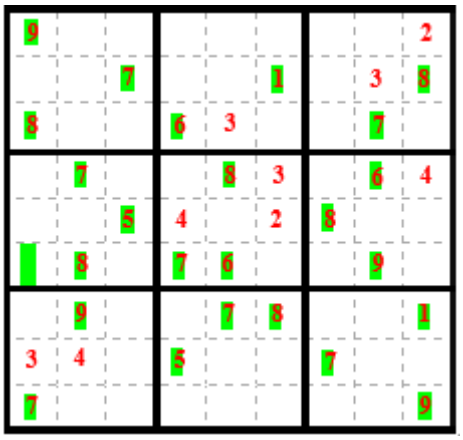


**Figure 5.8:** A new instance is generated from the Sudoku instance shown in Figure 5.6 after exchanging three pairs of values.

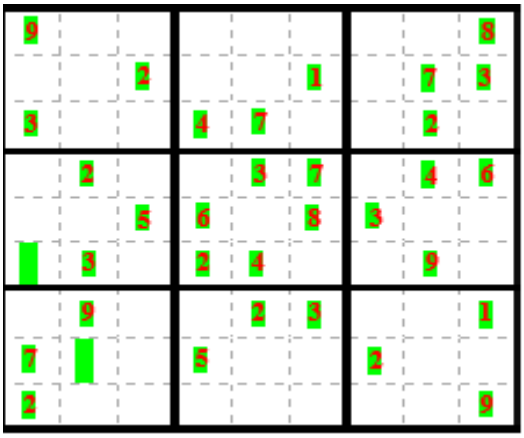


**Figure 5.9:** A new Sudoku instance is generated after the replacement of all the digits for the Sudoku instance shown in figure 5.6

**Replacement of All Digits**

By exchanging the values of all the clues in an existing Sudoku instance, a new Sudoku instance can be constructed. Take a look at Figure 5.6, which shows a Sudoku puzzle. (1, 9), (2, 8), (3, 7), (4, 6), (5, 4), (6, 5), (7, 3), (8, 2), and (1, 9), (2, 8), (3, 7), (4, 6), (5, 4), (6, 5), (7, 3), (8, 2), and (1, 9), (2, 8), (3, 7), (4, 6), (5, 4), (6, 5), (7, 3), (8, 2), and (9, 1). We can then create a new instance, as seen in Figure 5.9. The 1s in cells [1,1], [7,2], [6,8], and [9,9] have been replaced by 9s, the 2s in cells [1,9] and [5,6] have been replaced by 8, and so on.

**Rows-in-a-Band Exchanging**

Rows-in-a-band Changing two or three rows in the same band in any way we like is what we mean by exchanging (or randomly). As seen in Figure 5.15, three consecutive mini-grids form a band. Band 1 is formed by mini-grids 1, 2, and 3, whereas band 2 is formed by mini-grids 4, 5, and 6, and band 3 is formed by mini-grids 7, 8, and 9. Now we can see if all of the rows in each band can be swapped out to make new Sudoku puzzles.

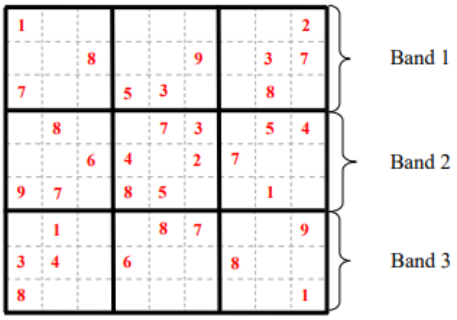



**Figure 5.15:** Concept of band in Sudoku puzzle.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

The freshly created puzzle generated after the swap of rows in the same band is still valid. Figure 5.16 depicts a new Sudoku instance constructed by swapping the first and second rows in band 1 for the Sudoku instance depicted in Figure 5.6. This concept can be expanded by swapping rows in corresponding bands, either alone or in combination, to create an infinite number of Sudoku puzzles.

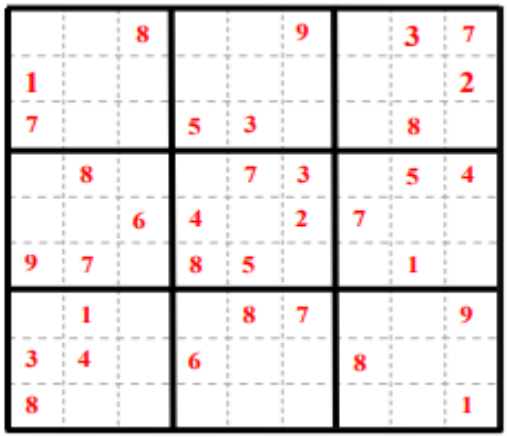


**Figure 5.16:** After swapping the values in rows 1 and 2 of band 1 for the Sudoku instance illustrated in Figure 5.6, a new Sudoku instance is formed.

**Band-Exchanging**

It should now be self-evident that we can swap the position of an entire band with another to create a new Sudoku instance. To create this instance, bands 1 and 3 of the Sudoku instance in Figure 5.6 have been swapped, as shown in Figure 5.19. To create this new Sudoku instance, mini-grids 7, 8, and 9 in Figure 5.6 have been swapped out for mini-grids 1, 2, and 3.

**Stack-Exchanging**

To create a new Sudoku instance, we can swap the positions of entire stacks with another stack. As a result, swapping stacks 1 and 3 of the Sudoku puzzle shown in Figure 5.6 yields a new Sudoku instance as shown in Figure.

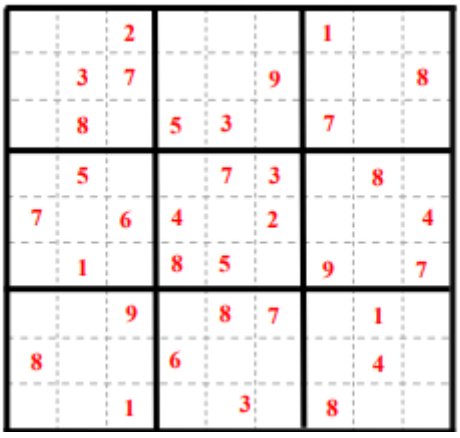


**Figure 5.20:** A new Sudoku instance is formed after the values in stack 1 and stack 3 in the Sudoku instance shown in Figure 5.6 are swapped.

The clues in mini-grids 1, 4, and 7 are now the clues in mini-grids 3, 6, and 9, respectively, in the newly produced instance, and the clues in mini-grids 3, 6, and 9 are now the clues in mini-grids 1, 4, and 7.

The project created successfully generates Sudoku problems based on difficulty levels that are solvable

**Creating Terminal Pattern: Las Vegas Algorithm**

All algorithm manufacturing procedures revolve around the creation of a terminal pattern (also known as a solution pattern). Using a random algorithm, we generate a random terminal pattern. The Algorithm of Las Vegas A solvent can be used to turn a terminal design into a blank pattern. To increase the diversity of

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

complexity generated, we choose n cells in an empty grid at random and then fill these empty cells with 1 to 9 numbers while creating game rules. In this step, you'll be given a problem with the number n. The positions and values of the supply are determined by whether or not this dilemma can be resolved in an acceptable amount of time (for example, 0.1s).

Assume P (x) is the chance that a problem with a collection of objects will be solved. it's possible to fix The Las Vegas algorithm by repeating event x until it occurs, during which time the event is meant to finish producing the terminal pattern within the time limit. The function las vegas (P (x)) in the program names will return TRUE if the event occurs with the probability of P (x), else FALSE, and the event will be repeated until it returns TRUE as the following codes:

while (! las vegas (P (x)) The number of random objects n influences event opportunities P (x). As a result, we set a few n's and repeat the event 10000 times each n

**Solving Algorithm: Depth-first Search:**

In order to generate a judgment algorithm that determines whether the produced problem has a unique solution, a good solution algorithm that searches for all potential Sudoku puzzle solutions is required. Because the Sudoku puzzle is presented as a 9-by-9 grid with enough information from the given clues and rigorous game rules, a computational search can be used to find all of the complex solutions. We first generated a Sudoku solution in the style of Advanced Search in order to identify a solution like a terminal pattern or to determine whether a created puzzle can be solved differently. From left to right, up and down the grid, the solution searches for vacant cells. It attempts to fill in each blank with between 1 and 9 digits. When inputting another potentially untapped power, the solution returns to the previous empty cell and updates the completed digit to fit the parameters. The solver continues in this manner until all viable solutions have been found and documented.

**Easy level Sudoku instance 9x9**

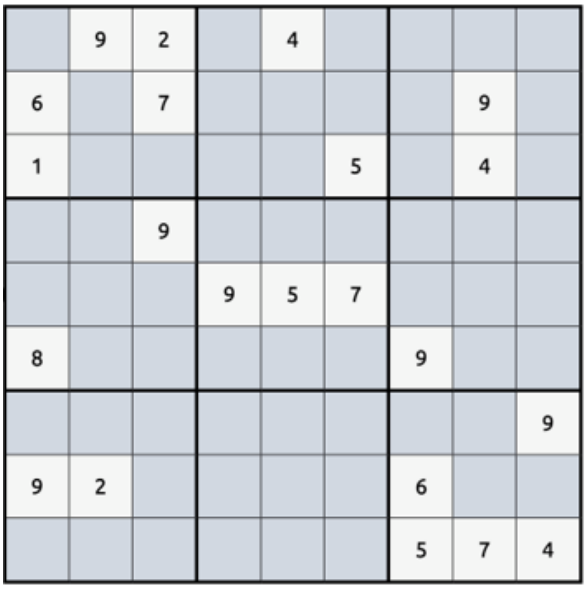
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 6 | | | | 2 | 9 | | 5 | |
| 5 | 7 | 4 | | | 3 | | | |
| | 9 | | | | | 8 | 6 | 3 |
| | 5 | 7 | 3 | | | 9 | | |
| 4 | 2 | 9 | 7 | 8 | 5 | 3 | 1 | 6 |
| | 6 | | | 4 | 2 | 7 | | |
| 3 | 8 | 5 | | | | | 7 | |
| | 4 | | 5 | | | 6 | | 1 |
| | 1 | |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

**Medium level Sudoku instance 9x9**

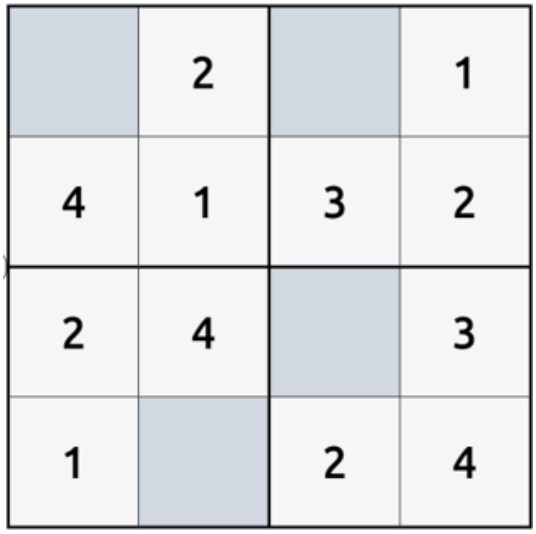
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | | | | | 2 | 9 | |
| | | 2 | 7 | | | | 6 | 3 |
| 6 | | | | | | 1 | | 7 |
| | | | 9 | 3 | 5 | | | 8 |
| | 8 | 4 | 6 | 7 | 1 | | 5 | |
| 5 | | | 8 | 4 | 2 | | | |
| 3 | | 9 | | | | | | 5 |
| 7 | 5 | | | | 3 | 8 | | |
| | 1 | 8 | | | | | 3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

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

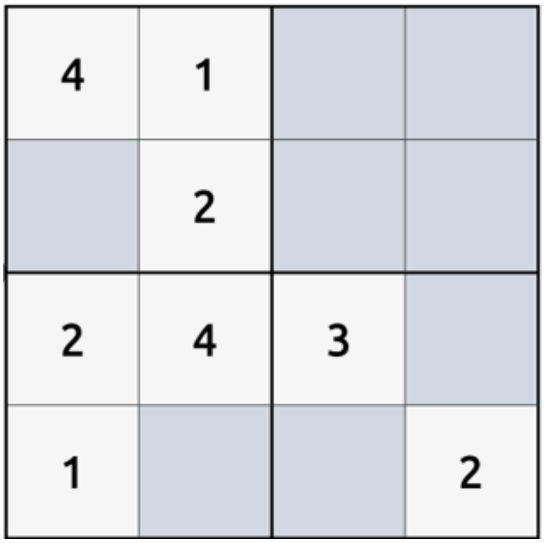
**Hard level Sudoku instance 9x9**



**Easy level Sudoku instance 4x4**

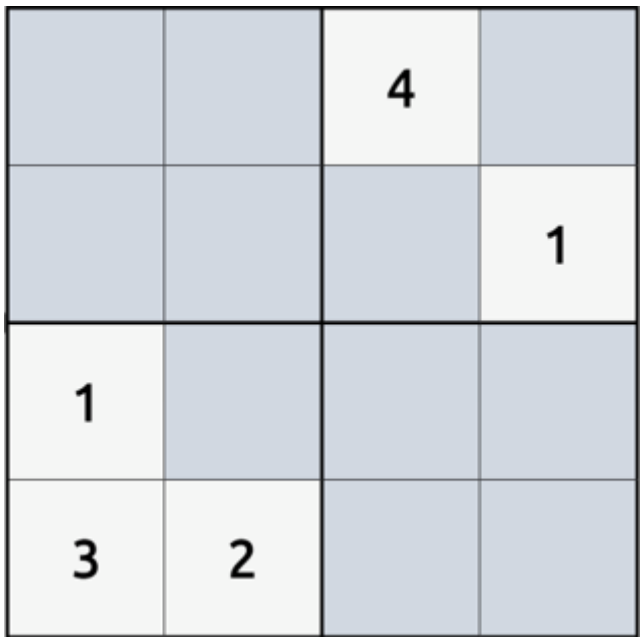


**Medium level Sudoku instance 4x4**



e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

**Hard level Sudoku instance 4x4**




## II. CONCLUSION

We've gone over three different types of Sudoku problem generation techniques in-depth, with a step-by-step approach that provides a clear understanding of the game's problem generation phase concerning difficulty levels. And the self-solver feature will solve every problem using the above-explained algorithm successfully giving unique solutions to every problem.

## III. REFERENCES

[1] Sudoku Puzzle Generation Publications by Tao Song, Shanchan Pang, Eryan Li, and Peng Zangh on Researchgate.net -2019 Beijing.

[2] Janne Koljonen, Timo Mantere With GA, solve, rate, and create Sudoku puzzles. IEEE Congress on Evolutionary Computation, pp. 1382-1389, September 25-28, 2018.

[3] Method for Generating Strategy-Solvable Sudoku Clues, Kohei Nishikawa, Takahisha Toda, Graduate School of Informatics and Engineering, University of Electro-Communications, Tokyo 182-8585, Japan, 2020.