

# Sudoku puzzle design algorithms computer science essay

[Technology](#), [Computer](#)



Our project is called generating, rating and solving Sudoku puzzle. The aim of the project is to design and develop Sudoku puzzle solving program by generate the puzzle with five differences difficulty rating level. The program is able to solve the Sudoku puzzle by using some Artificial Intelligent algorithm.

The final year project development period is given two trimester and we divided the development process into two part. The first part is to develop the problem specification, design, generating and study on some Sudoku solving algorithm which to be done by trimester one. Part 2 of the project is developing the complete Sudoku solving program and this part will be done by trimester 2.

## Chapter 1: Introduction

### 1. 1 Introduction

Sudoku is logic based number-placement puzzle. It designs with 9x9 grids so that the digits 1-9 appear exactly once in each row, each columns, and each 3x3 sub grids (also called as blocks) that compose the grid. Each Sudoku puzzle created with a unique solution, which typically means that there is only one correct way to fill in the 81 grids.

The aim of the project is to design and develop Sudoku puzzle solving program using the artificial intelligence algorithm. First at all the Sudoku configuration program would be able to generate the puzzle with five differences difficulty rating level, which are very easy, easy, medium, difficult and expert. The program also able to solve the Sudoku puzzle as well as

providing the chances of having tips while they achieve the specify condition which could help the player to solved the puzzle when they get stuck.

An interesting Sudoku puzzle is the one with the minimal hints given. The puzzle should be generated with a single solution. So, knowledge for the rules of Sudoku is very important which the digit 1-9 need to be appear exactly once in each rows, each columns, and each 3x3 sub-grid. Then a 9x9 empty grid need to be created and filled in with the digit 1-9 by following the rules to create a complete puzzle.

The next step is to begin removing digits from the completed Sudoku puzzle randomly to create the game. To generate a symmetrical puzzle, we need to randomly remove the singleton from the row, column, or block, following by naked pairs, naked triples, hidden pairs, hidden triples, and pointing pairs accordingly. The techniques used to remove the cells decided the difficulty level of the Sudoku puzzle. Lastly, the program is able to solve the Sudoku game by using the AI algorithm like simulated annealing.

## 1. 2 Problem statement

People nowadays always busy for working and cope for the challenges on their life. With the increasing busyness of the life, stress become a common issue that faced by the society. A long term of facing a particular problem will make a people to feel pressure and stress themselves badly. Therefore, they need some break to take their mind off from working materials before they continue to work. Sometimes, a challenges mini game like Sudoku can helps to exercise their brain and reduce stress.

Besides, with the fast growing of the technologies today, most people are able to obtain the advance electronic device like PDA, laptop, and so on. So the Sudoku puzzle game is not the game that is only published in the hardcopy such as newspaper or Sudoku book. Therefore Sudoku game can be installed in their electronic device so that they can bring it anywhere.

## 1. 2 Scope

Scope of the project is generating the Sudoku puzzle with different difficulty level and solves the Sudoku game. The scope covers:

Allow user to select the difficulty level of the Sudoku game.

Generate a complete Sudoku puzzle by following the Sudoku rules which are the digits 1-9 appear exactly once in each row, each columns, and each 3x3 sub grids.

Remove cells to create a valid Sudoku games according to the difficulty level that the user selected.

Allow user to input the digit 1-9 into the empty cells of the puzzle.

Highlight the incorrect input of digit with red colour.

Solve the Sudoku games.

## 1. 2 Objectives

Understanding the algorithm behind Sudoku puzzle.

Identify an algorithm to generate a complete Sudoku puzzle with the rules that the digit 1-9 appear exactly once in each row, each columns, and each 3x3 sub grids.

Identify the ways of removing cells from the complete Sudoku puzzle to create a valid game.

Generate Sudoku with different difficulty level.

Identify the algorithm to solve the Sudoku puzzle.

### 1. 3 Project Schedule

#### Project Schedule Management - Gantt Chart

Tasks

Activity

Start date

End date

Duration(days)

Task 1

Research and study relevant material

7/6/2010

27/6/2010

21

Task 2

Planning development process

28/6/2010

4/7/2010

7

Task 3

Implement Prototype for Generating

5/7/2010

8/8/2010

35

Task 4

Test Program and Fix error

9/8/2010

14/8/2010

7

Task 5

Amends report and prototype

15/8/2010

29/8/2010

14

Task 6

Validate interim report and Submission

30/8/2010

1/9/2010

1

Chapter 2: Literature Review

2. 1 Nature of the problem

Sudoku puzzle is a famous game among the worldwide. Most of the newspaper has a long history of publishing the Sudoku games such as The Times, Sin Chew Daily, and so on. The games in hardcopy required user to use pen or pencil to finish on the paper and the number of games are limited. So, in this project, a program is created to generate the Sudoku puzzle games. User does not need to use an extra writing material like pen or pencil to complete the Sudoku games. They just need a mouse to interact with the game.

Other than that, the program is able to solve the Sudoku puzzle as well as identify the puzzle is solvable and have a unique solution. User also can enter the Sudoku games that are not generated by the program and solve it by using this program.

## 2. 2 Background

The origin of Sudoku is from Switzerland (By a Mathematician Leonhard Euler) and then travels to Japan by way of America. “ Su” is the Japanese character with the meaning of number and “ doku” means single. In another word, it called single number which means that for each column, each row, and each sub grid, it can only contain the digit 1-9 without repetition.

In this project, we need to consider some important things like the cost that we need, the purpose of this project, the target user of this program and so on. Cost that we need to use is to complete the project in 24 week period. The purpose of this project is to create a program which can generate the Sudoku games with different difficulty level as well as able to solve the Sudoku games.

Sudoku is a game that suitable for all generation of the people. The program is able to generate the game in 5 different difficulty levels which is very easy, easy, medium, hard, and expert. Level “ very easy” and “ easy” are suitable for the children, medium is suitable for the adults, hard and expert are suitable for those who are experienced in the Sudoku games.

## 2. 3 Literature Review

<https://assignbuster.com/sudoku-puzzle-design-algorithms-computer-science-essay/>



## 2. 3. 1 The Science behind Sudoku

Sudoku is a game of numbers, but it does not have anything to do with mathematics such as addition or multiplication. A Sudoku grid is a special kind of Latin square that named by an 18th century mathematician-Leonhard Euler. Latin square is an  $n \times n$  matrices that are filled in with  $n$  symbols in a way that the same symbols appear exactly once in same row and column.

The number of valid Sudoku grids is 6, 670, 903, 752, 021, 072, 936, 960 which proved by the use of logic and computers. The minimum numbers of hints that a  $9 \times 9$  Sudoku puzzle can start with and the solution can still remain unique seem to be 17. This mean for any Sudoku which is having the hints that less than 17 will make it cannot guarantee a unique solution.

C: UsersVinZDesktop1. png C: UsersVinZDesktop2. png

Figure 1. An example of the minimal hints for a Sudoku puzzles and still guarantee with the unique solution.

C: UsersVinZDesktop3. png C: UsersVinZDesktop4. png

Figure 2. An example of a Sudoku game with 16 hints provided and leads to two solutions.

To solve a Sudoku puzzle, the most common way is backtracking. First, the program places a number " 1" in the first empty cell of the Sudoku puzzle. If the number is compatible with the existing hints, it moves to the second empty cell and place a " 1" again. When it encounters a conflict or clash with

other existing hints, it erases the number which just placed and enters number " 2". If it is invalid again then proceed with 3 or the next legal number. After the legal number placed, it moves to the next cell and start again with number " 1".

If the number that placed and it is still not compatible with the existing hints is " 9" (" 9" is the largest number in Sudoku puzzle), the program will backtrack and increase the previous cell's number by one. After that it moves forward until it is clash with other existing hints and backtracks again. Sometimes the program needs to backtrack for several times before it can move forward.

Backtracking technique is good and fast when apply in a program or machine, but a human player need to spend more time to finish the game using this technique. So they used smarter ways for solving Sudoku and generally turn to method " trial and error" method at last of the solving part.

### 2. 3. 2 A Pencil-and-Paper Algorithm for Solving Sudoku Puzzle

To solve a Sudoku puzzle, it involve with the widely known concept of matching numbers across cells. In another mathematical point of view, we name it as pre-emptive sets. The most important technique to solve the Sudoku puzzles is based on two things, which are the (i) definition of pre-emptive sets and (ii) the occupancy theorem. A mark up puzzle can help to solve the Sudoku puzzle with more efficiently by writing down the possible number that can be filled into the empty cell.

C: UsersVinZDesktopmarkup. png

Figure 3. Example of a mark-up Sudoku puzzle.

Pre-emptive sets:

A pre-emptive set composed of the digits 1-9 and is a set of size  $m$ , where:

$2 \leq m \leq 9$ . There is a property that no numbers other than the  $m$  numbers from the list can occupy the  $m$  cells. It can be determined easily by using the mark up puzzle. The range of pre-emptive set is a column, row, or sub grid in which all of the cells of the pre-emptive set are located. It can be presented by

$$\{[n_1, n_2, \dots, n_m], [c(i_1, j_1), c(i_2, j_2), \dots, c(i_m, j_m)]\}$$

Where  $1 \leq n \leq 9$ , for  $i, j = 1, 2, \dots, 9$

C: UsersVinZDesktop852. png

3459

459

6

23478

1

2348

234579

24579

2345

Figure 2. Pre-emptive set

For example, the figure above consist of a pre-emptive set

{[3, 4, 5, 9], [c(7, 1), c(7, 2), c(8, 3), c(9, 3)]} with size 4.

Occupancy theorem

Suppose that  $x$  is a number that listed in the pre-emptive set and it lies completely in one column, or row, or a sub grid, so there can be no occurrence of  $x$  for the column, or row, or sub grid that outside of the pre-emptive set.

Firstly, the algorithm use method 1 and method 2 to complete the puzzle until it cannot continue further. Method 1 is to figure out the " Force cell" which is the cell that can be determine by eliminate the impossible number (number that already existed in the same column, row and sub grid) that occupy the cell. Method 2 is to figure out the possible number for the empty cell that can be determined by checking the same number which occupies the cells in the column and row of the sub grid.

After that we look for the pre-emptive set from each column, each row, and each sub grid. Try to break it into several smaller pre-emptive set and cross

out the number according to the occupancy theorem. For example in Figure 2, the pre-emptive set is  $\{[3, 4, 5, 9], [c(7, 1), c(7, 2), c(8, 3), c(9, 3)]\}$  with size 4, it cross out the number that are not inside the pre-emptive set. In figure 2, digits 8 and 2 in cell  $c(8, 1)$  and  $c(9, 1)$  can be determined directly after the number had been crossed out. So apply again method 1 and method 2 to figure out other possible number.

Clearly then, repeat again the process until it cannot find out any pre-emptive tes ot until the sudoku puzzle is solved. if the result seem to be violated, then the sudoku game is a unsolable game.

#### Metaheuristics can solve Sudoku puzzles

Before the simulated annealing can straightforward, representation, neighbourhood operator and evaluation of candidate solution need to be defined. Representation used to assign each empty cell a value randomly, but every sub grid contain the digit 1-9 exactly once.

During the process, neighbourhood operator choose two different non-fixed cells (the cell which filled in by randomly) in the sub grid and swap them.

Two of the non fixed cells to be swap must be in the same sub grid to make sure the criterion of each sub grid contain the digit 1-9 exactly once.

Cost function used to evaluate the candidate solution by looking at each row and column, and then calculate the number of values that are not present.

For example in figure 3, 1 and 2 had repeated two times in the first row, so

the score of that row is 2. Note that the complete solution of the Sudoku puzzle will have zero of cost.

C: UsersVinZDesktop854. png

Figure 3

Here, simulated annealing start with a candidate solution  $s$  and a neighbour  $s'$ .  $s'$  is accepted if  $s'$  is better than  $s$  (according to cost function) or with the probability:

$\text{Exp}(-\Delta C/t)$ , which

$\Delta C$  = proposed change in the cost function,

$t$  = control parameter (as known as temperature).

$t$ , temperature is very important to expend the Simulated Annealing and lead to the success. The initial temperature should allow majority if the moves to be accepted, approximately 80% (according to Van Laaehoven Aart (1987)).

During the process, temperature is reduced according to the simple geometric cooling schedule. A formula,  $\alpha$  = use to make sure the temperature decrease according to the cooling rate. Note that  $\alpha$ , so the largest value of  $\alpha$  is less than 1, such as 0.9999. For example, value of  $\alpha$  is 0.9999 will cause the temperature drop slowly, and the value of  $\alpha$  is 0.5 will make it drop faster.

Furthermore, the use of homogenous Simulated Annealing schema takes the form of a sequence of Markov Chains that generated at the fixed value of  $t$ .

The  $t$  is then changed in-between subsequent chains (Van Laarhoven and

Aart, 1987). The value of  $t$  here is also very important as it decided the time to explore of the search tree and also the run times.

The size of the problem-instance defined by the non-fixed number in the grid where:

$ml =$

$ml =$  size of the problem-instance.

$=$  the function that identify the number of cells in the are non-fixed.

By using this formula, the value can allow each pair of the non-fixed cells have a good chance to considered at least once per Markov chain.