Basically, the algorithm will be broken into 3 parts:

1. The base raw filled coordinate list will be generated varied by the specified filled coordinate spacing in the same row only(to be explained later)
2. The generated data will have its portion groups and portions in groups swapped by a specified combination only(to be explained later)
3. The filled coordinate spacing in the same row and swapping combination will be generated by random

**Filled Coordinate Spacing In The Same Row**

Suppose a row has length l and f filled coordinates. If the filled coordinate spacing is s, then the formula for the column index c with the filled coordinate index i, where 0 <= I <= f, with the filled coordinate, without initial offset o, is (o + is) % l.

**Portion Group**

For a sudoku board with size X, a portion group is a group of the 1st to Xth/(X + 1)th to (2X)th/(2X + 1)th to (3X)th/…/[(X - 1)X + 1]th to (X ^ 2)th portions. So a row group is that of such rows and a column group is that of such columns.

The importance of this concept is that, no matter how these portion groups are swapped within each other, every grid will still remain unchanged, meaning that the board won’t become invalid.

**Portions In Group**

Portions in group are those In the same portion group. So rows in group are those in the same row group and columns in group are those in the same column group.

The importance of this concept is that, no matter how these portions are swapped within the same portion group, every grid will still remain unchanged, meaning that the board won’t become invalid.

**Swapping Combination**

Given n elements to be swapped within each other, a swapping combination is a list of n consecutive indices xi, where 1 <= I <= n, arranged in a specific order. To simplify its usage, the smallest xi is 0 and the largest one is n - 1.

For instance, for 6 elements, [0, 1, 2, 3, 4, 5], [2, 5, 4, 3, 1, 0] and [5, 4, 3, 2, 1, 0] are swapping combinations, meaning that the new index of the element with old index i should be l[i], where l is the list being the swapping combination.

**Base Raw Filled Coordinate List**

Basically, for a board with size n and filled coordinate spacing k, the column index of a filled coordinate with row index y and filled coordinate index in that row, i, is (y % n \* n + [y / n] + i \* k) % (n ^ 2), where y % n \* n is the row index within the row group multiplied by the board size, [y / n] is the row group index, i \* k is the filled coordinate spacing per filled coordinate index increase, and n ^ 2 is the board length. Note that y % n \* n + [y / n] is the base column offset for each row.

A row, column or grid is said to be valid if it has [f / l] or [f / l] + 1 number of filled coordinates.

If x and n are [coprime](https://en.wikipedia.org/wiki/Coprime_integers), then the above column index formula will ensure that all rows, columns and grids are valid.

Proof: