An Efficient Genetic Algorithm and Logical Rule for Solving Nonogram Puzzle

[1]Roba Mahmoud Ali Aloglah
Al-Balqa Applied University
Ajloun University College
Ajloun, Jordan
Tel: +962-772156070
Email: robajabali@bau.edu.jo

[2]Hasan Rashaideh
Al-Balqa Applied University
Al - Salt, Jordan.
Email: rashaideh@bau.edu.jo

[3]Firas Hani Abed AL-Kharabsheh
Al-Balqa Applied University
Al - Salt, Jordan.

Abstract
Nonogram is a NP-complete problem (Puzzle) where many researchers proposed several algorithms for solving it; some researchers used a proposed genetic algorithm (GA), as a solution. In this paper, we propose an algorithm using genetic solving algorithm (GA) with Logical Rule depending on the fact that most of nonograms are condensed and adjacent, as result of that a logical rules are reasoned to shade some cells after converting it to the form of (N x M) grid and must be shaded or left blank depending on numbers at the grid sides, in which every number indicate a sequence of block will be in the solution, with at least one space between these numbers,. Then, we use the time based sequence backtracking algorithm to solve those hesitant cells and improving the search efficiency by proposing a logical rule. Experimental results show that a Nonogram puzzle can be solved using genetic algorithm and logical rule by the complete proposing fill matrix in which it computational algorithm used to full row and full column in a way there no possible to swap between row and column and since the search space is so large, we produce a modified generate population which they are depend on the Matrix row which is define by the number of group of ones in each row.

Keywords: Nonogram, Puzzle, Depth First Search, Genetic Algorithm (GA), Chronological Backtracking.

1. Introduction

Nonogram is a widely spread logical games between Japanese population , in 1987, a Japanese graphics editor, Tetsuya Nishio, and a professional Japanese puzzler; Ueda e Negao, proved that to determine if the Nonogram have an exclusive solution is a NP-Problem and the decision problem that have to be solved in a Nonogram is NP-complete and therefore not doable in polynomial time [5]. As a result of that it was named as Japanese puzzle, the question “Is a NP-complete problem puzzle solvable?” [1, 2]. Some associated papers [3, 4], proposed a non-logical algorithms as a solution for this problem by, considering slow execution speed.

1.1 Nonograms

The goal of Nonogram is to shade some squares to formulate a picture from some conditions as shown in Below Figure (1) that exhibits a simple model of nonogram puzzles in which there are two models of Nonogram Puzzle: first model is black and white and the second model is a colored one , in this paper; the black and white puzzle will be under our intention where the number at left and top sides indicates the numbers of square blocks have to be shaded from left to right in the grid by the order they appeared, if there are more than or two numbers, the group of black (shaded) squares have to be separated by at least one blank (un-shaded) square in the grid to form a picture that fulfills the following limitations:

a. Each cell has to be shaded (black) or left un-shaded (white).
b. K black runs for each row or column must be executed; If it has k numbers: $s_1, s_2, \ldots, s_k$, the first (left most for rows/top most for columns) black run with length s_1, the second black run with length s_2, and so on.
c. There must be at least one un-shaded cell between two sequential black (shaded) runs.

In Fig. 1(a); it is noticeable that the puzzle has a unique solution presented in Fig. 1(b). On the other hand, in Fig. (2); the puzzle has dual solutions, as shown in Fig. 2(a) and in Fig. 2(b), bearing in the mind that the puzzle in Fig. (3) has no solution. Therefore, there could be no solution, exact single solution, dual solution or more for a specified integral numbers. A solution for the puzzle can be a picture in a black white form. In this paper, we used shaded (black) grids for solved grid, as well as an un-shaded (white) grids for a non-solved grid.

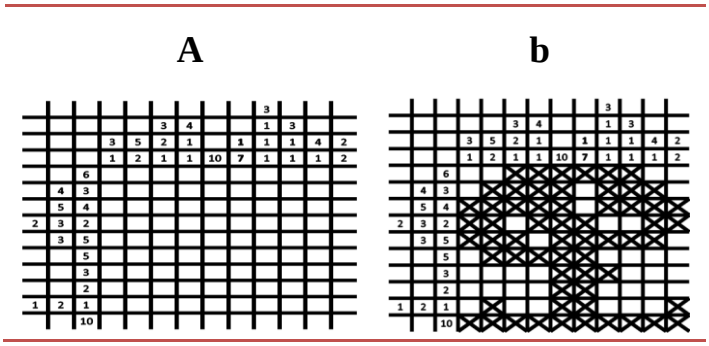


Figure 1 (a) Nonogram puzzles grid corresponding to the puzzle and (b) it's a solution

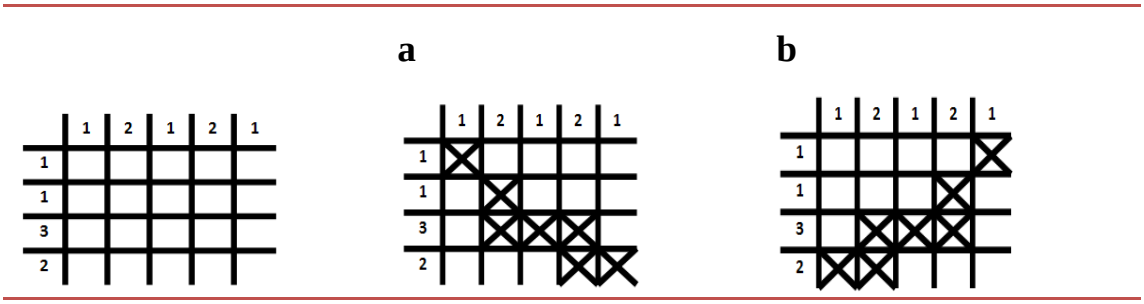


Figure 2 : Puzzle with two solution, (2.a) the first solution and (2.b) the second

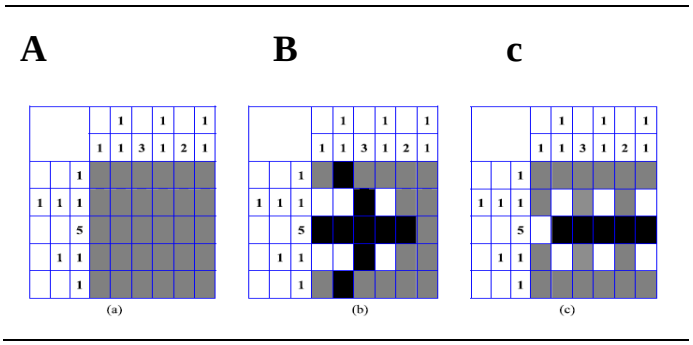


Figure 3: un-solved Puzzle, 3(**a**) proposed example. 3 (**b**) no solution in rows 2 and 4 in row 3; If the first five cells are shaded (black) 3(**c**) no solution in row 4; If the last five cells are black (shaded) in row 3.

2. Previous Studies

Batenburg, in 2003 [7] , raised up evolutionary algorithm applied on discrete tomography (DT) in which a discrete image had been reconstructed from its projections [8] in a small number of directions, then he applied his proposed algorithm to solve a Nonogram puzzle for the purpose of providing the algorithm correctness [7], after that he shared Kosters with proposing a new proposed scheme to to get Nonogram puzzle solution by altering [8] the evolutionary algorithm for the purpose of solving the nonogram. In which he discovered that the resulted solution may be incorrect because the evolutionary algorithm [7] owes coverage to a local optimum.

Wiggers [9], In 2004, came up with a new genetic algorithm (GA) conjunct with a depth first search (DFS) algorithm as a solution for nonograms based on comparing the performance of the said two algorithms are slow, and high probability for stuck in local optima could be occur in the GA algorithm during the process execution for getting a solution for the said puzzle.

Ortiz-García [10], in 2009 , categorized the algorithms that could solve Nonogram puzzle into an objective function guided algorithms as metaheuristics approaches / emergent solutions as GA, evolutionary algorithms or into a logic-type algorithms.

Also; Ortiz-García, in 2008, proposed a Solution for a very challenging puzzle with a hybrid evolutionary-logic algorithm, using a Simulated Evolution and Learning. [11] , the logical algorithm acts as local solver. While another approach proposed in the combination of relaxations of the problem, for sensing the value of pixels in the proposed puzzle. An complete solution was discovered by implementing many iteration of these relaxations for the puzzle[12].

Batenburg, in 2009; proposed a dominant solver for Nonogram puzzle has been proposed depending on an logical algorithm which can be applied to shade Nonogram puzzle and to generate puzzles in an effective way [13]. Which implemented with a set of unknown variables, and progresses by executing a logical ad-hoc algorithm solely take the benefit of puzzle logic Information without any requirement for the objective function. Taking into consideration a unique logic information of the puzzle may available for some hard puzzles, so these proposed algorithms as the algorithm propose by Ortiz-García in [6] or proposed by Batenburg, [7] may not work, or may take a lot of time to find a solution for the puzzle.

Yu, in 2011; proposed an efficient algorithm for solving nonograms the proposed algorithm was a heuristics algorithms used to solved this puzzle by applying logical rule then applying backtracking algorithm .[14]

3. Nonogram Puzzle

This kind of puzzles can be clarified using the example mentioned in fig (1) a simple Nonogram puzzle is suggested which takes the form of (N x M) grid which must be shaded or left blank (un-shaded) according to the numbers at the side of the matrix used to get a solution for a hidden picture, The goal of the puzzle is determine which square will be left blank (un-shaded) and which will be shaded (black). The numbers on the left and above of the empty puzzle show the rows and columns inputs and information indicates the

number of blocks must be shaded in the grid, in order they appear with a noting that the blocks of squares in the grid have to be separated by at least one un-shaded (white) square If there are two or more numbers.

For example the second row of the puzzle shown in figure 1. Here (4,3) indicates availability of set with Four and Three shaded squares (black) with at least one un-shaded square (white) between the sequential groups . As little by little more squares are being shaded; more information becomes clear whether the squares have to be shaded or kept un-shaded (empty). In the end the puzzle is solved if all the rows and columns are shaded with regard to their numbers on each side.

This puzzle can be easily solved by logic and that this kind of puzzles does not depend on luck when you solve the puzzle as it may be seen that it is true and when it reaches the last row or column to be solved, and then usually an error occurs and then we will come back and start from the beginning. The goal is to shade the grid to formulate image fulfilled the following boundaries [6].

Each square has to be whether shaded or not; the row or the column has to contain k black runs the first (left most for rows / top most for columns) shaded run with length s_1; If and only if , it has k numbers: $s_1,s_2,...,s_k$, after that ,the second shaded run is executed with length s_2, and continuously. At least one unshaded blank square must be available between two sequential shaded runs. And the finding of this; as shown in fig (2); some puzzles may have unique solution, one solution or more.[2]

3.1 Logical Rule to solve Nonogram Puzzle

Many simple boxes schemes could be implemented to state the quantity of figured square boxes could be done. For example fig (4, a), suggested a row of ten squares grid with only one number of 8, the quantity of bound blocks are 8 boxes that could range from the first row, with the availability of one spaces to the most left square in the grid; while in the second row, and by its obligatory to leave one spaces to the right; Consequently, the spread of the blocks will be on the eight middle most squares in each row. And this scheme is applicable despite of the numbers in the row available in the grid. If a grid with ten squares with numbers 4 & 3, proposed as an example as shown in fig (4, b), the shaded boxes may be:

- in first row , leave two squares to the right, if starting from left, one next to other,;
- is second row, leaving two squares to the left, if starting from right, one just next to other also,;

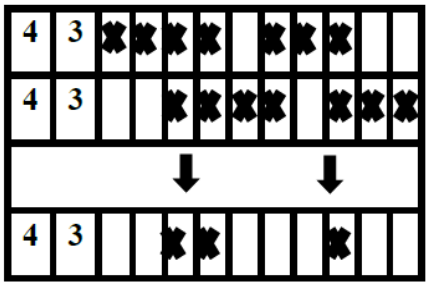


Figure 4: simple boxes, example (a)

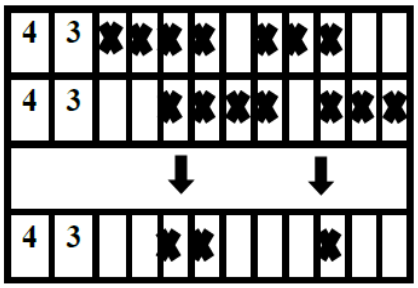


Figure 3: simple boxes, example (b)

So, the first set of four shaded boxes absolutely includes the 3rd & 4th squares, while in the 2nd set of three shaded boxes completely includes the 8th square. Shaded squares are located in the 3th, 4th & 8th squares.

3.1.1 Simple spaces

This scheme determines the spaces results from looking for squares that are unpredictable within any potential shaded boxes. In Fig (5), a ten squares row with boxes in the 3th and 8th square and block range contains 2 and 1numbers, number 2 will spread through 2nd or 4th square and number 1 will be at the 8th square .

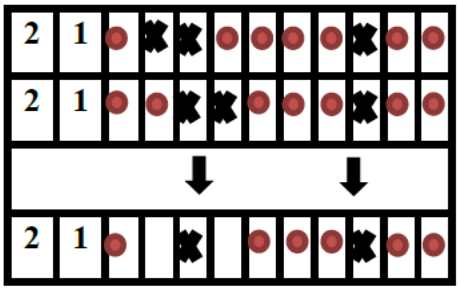


Figure 4: Simple spaces

- ✓ Since there will be a space at each side of the bound block; Number 1 is complete.
- ✓ Moreover, because it permanently must include the 3rd square; number 2 can just spread someplace between the 2nd or 4th square; nevertheless, this could leave squares that may not be shaded during any proposed solution within any scheme.

From the last example, the solver has to be vigilant since there may be unbounded blocks to each other accounted for; this is not always the solution.

3.1.2 Joining and splitting

By applying this method, closer squares may be occasionally jointed together into one block or fragmented into several blocks depending on the proposed space. Suppose availability of two blocks with an unshaded square between them, and then this square could be:

- ✓ A space; when box combining two blocks; will create a too large block.
- ✓ A box; when space separating two blocks; will create a too small block that not has plenty of free remaining squares.

suppose a row with numbers of 5, 2 and 2 of 15th squares with boxes in the 3rd, 4th, 6th, 7th, 11th, and 13th square as shown in example fig (6),:

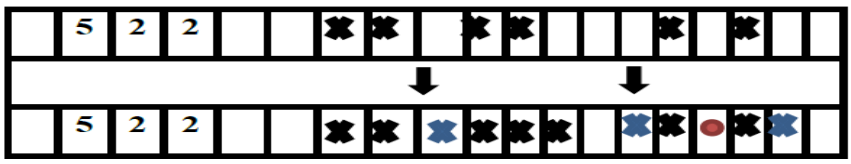


Figure 5: joining and splitting

The first two blocks will be joined by number 5 by adding a box to form one large block, since space would produce a 4 boxes block that is not adequate there. Meanwhile, the last two blocks 2 will be separated by numbers 2 by a space, a block of 3 continuous boxes because a box will be created, which is not permitted there.

3.1.3 Rule for first and end

There is a specific rule for the last and first row and for the first and last column. we will explain one situation and for our example fig (7), in the last row and because of the previous rule simple boxes then the last row will be all filled, when the last row is filled we will see at the first row in the columns number which is red row, And when we applied for the first row in the puzzle then the last row in columns number will be concerned, for the first column in the puzzle the last column in the rows numbers will be concerned and finally for the last column in the puzzle the first column in rows number will be concerned.

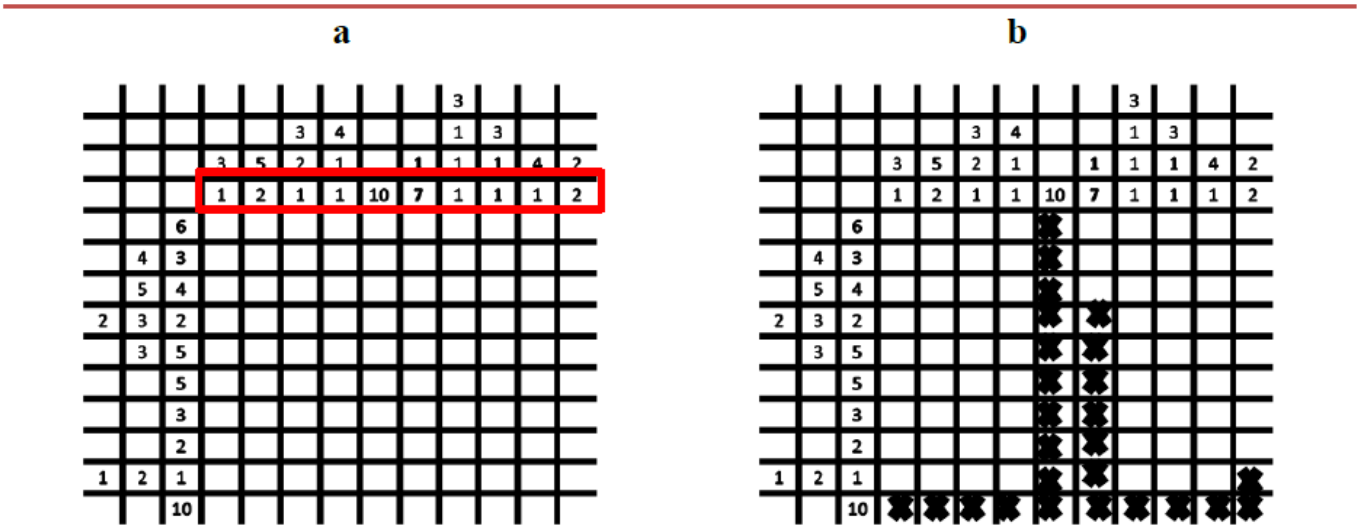


Figure 6 : rule for first and end, (a) is the puzzle; (b) is the solution by rule 3.4

4. Genetic Algorithm

4.1 What is the Genetic Algorithm

A genetic algorithm (GA) is a set of instructions that are repeated to solve both constrained and unconstrained optimization problems conceptually by processing steps inspired by the biological processes of evolution of **SURVIVAL OF THE FITTEST**- Better and better solutions advanced from previous generations until a near optimal solution is obtained.

The algorithm frequently adjusts a population of distinct solutions [15] The working principle of this algorithm is that every genetic algorithm step chooses individuals randomly from the current available population to become parents to generate children for the upcoming generation.

A genetic algorithm can be applied to solve many difficulties improvement for unfitted healthy optimization ordinary algorithms, to include the problems that the objective function is intermittent, not differentiable, stochastic, or non-linear too[20].

Fortuitous genetic algorithm is that Tim from which to address the programming problems can be mixed right, where an integer values restricting some variables .

To generate GA ; three main rules could be used and the form of the base of the rules at every step to generate the next population generation [15] as follows:

- ***Selection rules*** select. the individuals, and let the parents that contribute to the population in the next generation
- ***Crossover rules*** this is done by combining the two parents to form the next generation of children.
- ***Mutation rules*** applying random mutation changes for parents of individual things for the formation of children**.**

4.2 Example for genetic algorithm

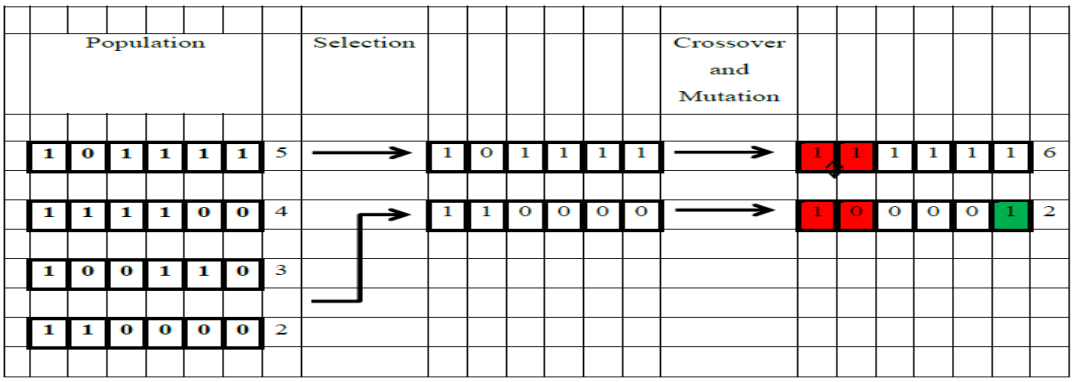


Figure 7 : example on Genetic algorithm

In Figure (8): Example of GA. Step one starts with 4 chromosomes or individuals in a selected population. Chromosomes are DNA segment encoding couple can contain values of 0 and 1. The values next to chromosomes equal to the value of fitness function. Fitness function only sums the chromosome number which is required to calculate fitness value of all the population chromosomes and then two of the chromosomes have to be selected for the imitation process (reproduction).

Often the selection process is unsystematic, but with a favoritism to the chromosomes with high fitness, after the selection / executing the exchange process and change working on two chromosomes. Transmission process of a number of genetic factors from one chromosome process changes with another. Boom operator only, in the right pane of Figure (8) is an exchange of value of one gene from chromosome through two genes killing pigment so that the exchange and mutation are less in the right side of the chromosome. But change and exchange does not always happens, but through the following mechanism which was used in [12] as the rate of transmission equals 60% and have an infectious mutation of 0.1% ,this indicates that 60% of chromosomes choose to blend with each other and only 0.1% left for the mature manner.

4.3 Flow chart for the Genetic Algorithm

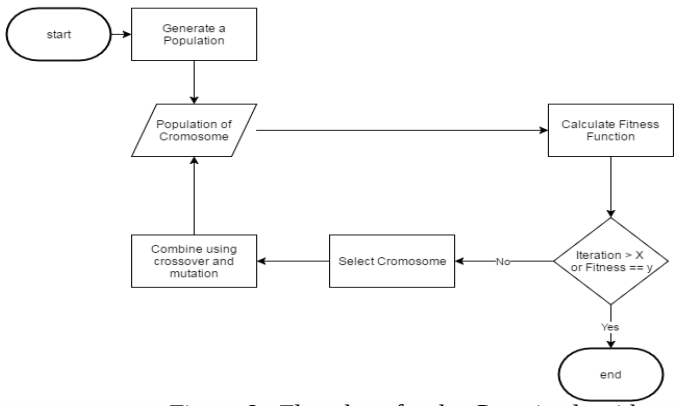


Figure 8 : Flowchart for the Genetic algorithm

Fig (9) shows: the flowchart of GA, in the first time the algorithm runs, that has a population of one or generates. Next calculate the physical Collar of chromosomes Other, as well as through an examination to ascertain whether the fitness is high enough or there was a lot of repetition has not found a solution.. When completed, generation of new residents in the process in Fig (9) starts again for the purpose of finding a chromosome has a physique that is good enough [5].

Algorithm genetic algorithm is described as follows:

Page | 7

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

The Genetic algorithm flow chart executed as follows:

1. ***[start]*** to *Produce random population of N chromosomes .*
2. *[**Fitness**]* Assess *fitness function F(x) for every single population chromosome S.*
3. *[**New Population**] generate fresh population by repeating following steps until new population completion.*
   - *[**selection**] according to better fitness (highly probability selected); select chromosome for two parent from a population*
   - *[**crossover**] cross over the parent to generate a new (children) offspring. Offspring will be an exact replica of parents without implementation crossover,*
   - *[**Mutation**] implemented to modify a new children in each chromosome position*
   - *[**Accepting**] add to the new population the new children.*
4. *[**Replace**] to repeat run of algorithm using the new generated population.*
5. *[**Test**] stop and indicate the final solution in existing population if only if end condition is satisfied ,*
6. *[Loop] Go to step 2*

5. Proposed Method

Since Nonogram is a piece of information used to formulate black and white images as a solution by defining the image as an N by M binary matrix in which the values of (0 and 1) could be used where the number 0 means unshaded square and 1 means shaded square. On the other side we can represent the images as a grid to describe the image as a row projection / column projection. These projections are more usually referred to as a description for lines in which each line description defines the following (number, size and group order) of sequential shaded squares in each line solved.

A positive integers well-ordered list represents the line projection P (P_1, P_2,..., P_k). Each integer number P_i represents the number of sequential shaded squares of a particular group. The groups have to be parted by one unshaded square at least. The beginning group may be led by zero or more unshaded squares and besides the end group may be tailed by zero also or more unshaded squares.

Each shaded square in any line has to be line chunk of these groups, so the total number of shaded squares in the line must equal D_k i=1 Pi. On the other hand the line projection P can label line with the regular expression $0\leftarrow 1d_1\ 0\ +1d_2\ 0+\ ...\ 0+1d_k\ 0\leftarrow$. Additionally; Lines lacking any shaded squares may be pointed with the special description (0).

In the end, we can summarize our proposed method as follows: A Nonogram N maybe defined as an well-arranged sequence of row descriptions labeling the rows of pictures in order (left to right) and column descriptions (c_1, c_2,...,c_m), labeling the columns of pictures from top to bottom. So , There is one Nonogram for each black/white images which can describe the image and may be one Nonogram can describe more than one image. The main aim of this paper is to propose a exclusively solvable Nonogram using the genetic algorithm and the logical rule [17], the proposed method contain two part the first part is the logical rule and the second part is using genetic algorithm .

5.1 Logical rule

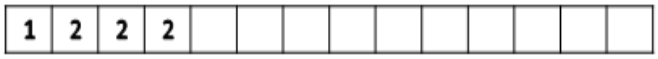
There is many method of logical rule which describe clearly in[18], in our method we proposed this rule which can be applied and named by Complete Fill Matrix .

5.2 Complete Fill Matrix

The key to solve this puzzle by this approach is to look at one line (row / column) at a time, then look at the line that have larger number as an example in fig (1), the last row have number (10) which mean that in this row there is a 10 shaded boxes and because of this puzzle is an (10 x 10) this mean this row will be completed fill, and there is a number 10 in the column will be solved like the last row.

Figure 9: example on the complete fill with one number

And if there is many number like fig (11) puzzle and fig (12) the solution



For example fig (11)

Figure 10 : the puzzle

Then if we sum these numbers (1+2+2+2) the result will be 7 and because the rule of the Nonogram is (at least one blank square between the group of black box) this mean that there is 4 element and the blank square is equal to 3 then if we sum these tow number 7 + 3 = 10 this mean this row will be completed filled like fig(12)

This rule can be writing for the rows in (1):

$$M == \sum_{k=1}^{\lceil N/2 \rceil} S_i + (n-1) \qquad (1)$$[19]

Where M indicates numbers of columns and N indicates numbers of rows and Si is summation of the puzzle numbers and n is the how many of number there are and for the column

$$N == \sum_{k=1}^{\lceil M/2 \rceil} S_i + (n-1) \qquad (2)$$

In equation (1) show the complete Row of the matrix and the equation (2) is the complete column for the matrix. After applying this rule on the matrix of puzzle and found the complete row and complete column then the genetic algorithm will be applied.

5.2 The Design of Genetic Algorithm

In this part, and for the purpose of solving Nonograms, a genetic algorithm must be developed. The flow chart shown figure (9) summarized the Nonograms solving algorithm, and now the following will be taken in consideration during the implementations process: (fitness function , crossover, mutation and selection operators).

5.2.1 Population

Because of this puzzle is NP-Complete problem there is a large number of population which can be produce, so it's a problem, to solve this problem an approach in the implementation was produce , which is depend on the numbers of rows which called Matrix_row .in fig (12) an example on the modified population.

Matrix_row			
0	0	3	2
0	1	2	1
0	0	2	2
0	0	1	3
0	2	1	1
0	0	2	3

Population 1							
1	1	1	0	0	1	1	0
0	1	0	0	1	1	0	1
1	1	0	0	0	1	1	0
0	1	0	1	1	1	0	0
0	1	1	0	1	0	0	1
1	1	0	0	1	1	1	0

Population 2							
0	0	1	1	1	0	1	1
1	0	0	1	1	0	1	0
0	0	1	1	0	0	1	1
1	0	0	0	0	1	1	1
1	1	0	0	1	0	1	0
0	1	1	0	0	1	1	1

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

Figure 11 : shows the modified population

5.2.2 Fitness function

Defining the fitness function is the primary step in the GA because this fitness shall chose the best chromosome required for solving the puzzle, depend on the fitness function in [6] this fitness compute for the similarity between the Matrix_row for the original puzzle and the Matrix_row for the population and the same thing computes for the Matrix_column.

$$F(X) = f_1(x) + f_2(x) + f_3(x) + f_4(x) \quad (3)$$

Where

$$f_1(x) = \sum_{i=1}^{N} \left| \sum_{p=1}^{M} r_{ip} - \sum_{p=1}^{M} x_{ip} \right| \quad (4)$$

$$f_2(x) = \sum_{k=1}^{M} \left| \sum_{p=1}^{N} c_{kp} - \sum_{p=1}^{N} x_{kp} \right| \quad (5)$$

$$f_3(x) = \sum_{i=1}^{N} \left| \left(M - \sum_{p=1}^{M} r_{ip} - \right) - \left(M - \sum_{p=1}^{M} x_{ip} \right) \right| \quad (6)$$

$$f_4(x) = \sum_{k=1}^{M} \left| \left(N - \sum_{p=1}^{N} c_{kp} - \right) - \left(N - \sum_{p=1}^{N} x_{kp} \right) \right| \quad (7)$$

Where $f_1(x)$, in equation (4); to get X as solution; accumulating the sum of the difference between the preferred number of 1s and the real number of 1s in rows. In equation (5); $f_2(x)$; we can get the required solution X, by accumulating the sum of the difference between the preferred number of 1s and the real number of 1s in columns,. $f_3(x)$ in equation (6) we take the sum of the differences in number of 0 s for rows between the actual matrix and the given solution X. finally in $f_4(x)$ in equation (7); we take the sum of the differences in number of 0 s for columns between the actual matrix and the given solution X. And because the population is modified then the $f_1(x)$ and $f_3(x)$ will be similar for the desired matrix in all population which is minimized the final value of F(X).

5.2.3 Crossover and mutation operator

Two basic operators for GA are crossover and mutation on which the Performance of GA is highly dependable. Moreover the operator type of implementation is highly dependable on the problem itself and its encoding. In this proposed method; by applying the binary encoded, the binary string from the commencement of our chromosome to the crossover point is imitated from their first parent, and the

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

remaining binary strings are copied from the second parent and this gives twice childrens. The crossover operator applied in which every time when a children are produce the rule of complete matrix is applied on the them ,so this approach give the crossover ability to give a enhancements on their children's ,and the same thing applied on the mutation operator which is select bits from the parent and then inverted them .

5.2.4 Main Algorithm

- *Convert Image to Matrix and find (M_row, M_column)*
- *Create initial population* P_0 *of size nPop consisting of matrices in M (M_row, M_column);*
- *Find complete matrix*
- *Perform a Copying matrix to the* P_0 *matrix*

t = 0;
While *(stop condition is not met)*
Begin

P_t = 0;

$for\ i \coloneqq 1\ to\ nPop$

Begin
Generate a child image C, by crossover or mutation;
Find (M_row_Pop, M_column_Pop)
Perform Copying matrix

$P_t \coloneqq P_t \cup \{C\};$

End

Select new population P_{t+1} from $P_{t+1} \cup P_t$

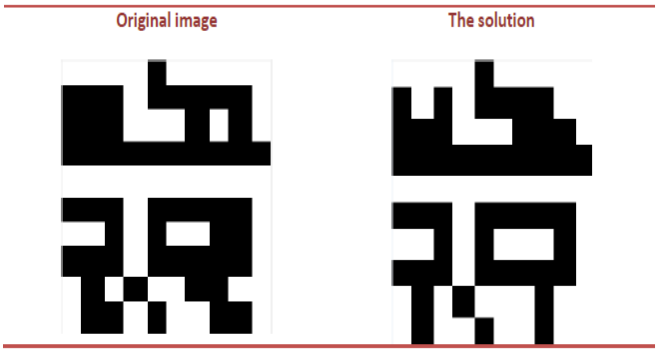
t = t + 1;
end
output the best individual found;

At the beginning of the algorithm the image is read and then convert it to a matrix and after that, the matrix row which is the vertical projections and the matrix column which is the horizontal projection will be calculated, then compute the complete matrix for the matrix row and matrix column, then generate the modified populations which are depends on the matrix row , after that the parents will be selected randomly from the populations matrices, then apply the crossover with applying the complete matrix rule on them and then compute the fitness value for each one from the children and store the value, and if the value equal zero then stop the algorithm else select the best solution matrix.

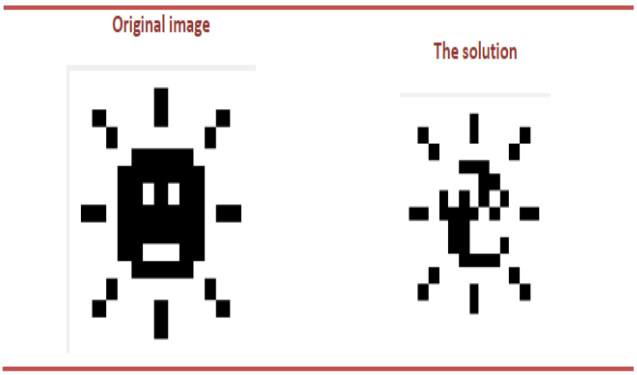
https://sites.google.com/site/ijcsis/
ISSN 1947-5500

6. Experiment Result

After applying the proposed method on (10x10) puzzle the solution with population size = 50 and number of iteration = 100



Example on (15x15)



But if the population size was increase so corresponding solution will be better

7. Conclusion

In this paper Nonogram puzzle is solved by using genetic algorithm and logical rule which we define this logical rule by the complete fill matrix which it compute the full row and full column in a way there no possible to change this row or this column.

And because the search space is so large, we produce a modified generate population which they are depend on the Matrix row which is define by the number of group of ones in each row like I explain previous in section 5.2.1.

References

1. Ueda N, Nagao T (1996) NP-completeness results for NONOGRAM via parsimonious reductions. Technical report TR96-0008, Department of Computer Science, Tokyo Institute of Technology, May 1996

2. McPhail BP (2005) Light up is NP-complete. Feb 2005. http://www.reed.edu/~mcphailb/lightup.pdf

3. Batenburg KJ (2003) An evolutionary algorithm for discrete tomography. Master thesis in computer science, University of Leiden, The Netherlands

4. Batenburg KJ, KostersWA (2004) A discrete tomography approach to Japanese puzzles. Proceedings of BNAIC, pp 243–250

5. contributors, W. *Nonogram*. 2016 [cited 2016 16 Apr]; Available from: https://en.wikipedia.org/wiki/Nonogram.

6. Ueda, N. and T. Nagao, *NP-completeness results for NONOGRAM via parsimonious reductions.*

7. Batenburg, K.J. and W.A. Kosters. *A discrete tomography approach to Japanese puzzles*. in *Proceedings of the 16th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC)*. 2004.

8. Batenburg, K.J., *An evolutionary algorithm for discrete tomography*. Discrete applied mathematics, 2005. **151**(1): p. 36-54.

9. Wiggers, W. and W. van Bergen. *A comparison of a genetic algorithm and a depth first search algorithm applied to Japanese nonograms.* in *Twente student conference on IT*. 2004. Citeseer.

10. Ortiz-García, E.G., et al., *Improving the performance of evolutionary algorithms in grid-based puzzles resolution.* Evolutionary Intelligence, 2009. **2**(4): p. 169-181.

11. Ortiz-García, E.G., et al., *Solving very difficult Japanese puzzles with a hybrid evolutionary-logic algorithm*, in *Simulated Evolution and Learning*. 2008, Springer. p. 360-369.

12. Batenburg, K.J. and W.A. Kosters, *Solving Nonograms by combining relaxations.* Pattern Recognition, 2009. **42**(8): p. 1672-1683.

13. Ortiz-García, E.G., et al., *Automated generation and visualization of picture-logic puzzles.* Computers & Graphics, 2007. **31**(5): p. 750-760.

14. Yu, C.-H., H.-L. Lee, and L.-H. Chen, *An efficient algorithm for solving nonograms.* Applied Intelligence, 2011. **35**(1): p. 18-31.

15. Mathworks. *genetic algorithm*. 2016 [cited 2016 7-may]; Available from: http://www.mathworks.com/help/gads/what-is-the-genetic-algorithm.html?refresh=true.

16. De Jong, K.A. and W.M. Spears. *Using Genetic Algorithms to Solve NP-Complete Problems*. in *ICGA*. 1989.

17. Henstra, S.J., *From Image to Nonogram: Construction, Quality and Switching Graphs*. 2011, Universiteit Leiden Opleiding Informatica. p. 57.

18. Jing, M.-Q., et al. *Solving Japanese puzzles with logical rules and depth first search algorithm*. in *Machine Learning and Cybernetics, 2009 International Conference on*. 2009. IEEE

19. *Jinn-Tsong Tsai. "Solving Japanese nonogramsby Taguchi-based genetic algorithm", Applied Intelligence, 01/26/2012.*

20 *D. B. Kulkarni. "Optimum Switching of TSCTCRUsing GA Trained ANN for MinimumHarmonic Injection", 2009 Second InternationalConference on Emerging Trends inEngineering & Technology, 12/2009*