An Efficient Approach to Solve Sudoku Problem by Harmony Search Algorithm

Satyendra Nath Mandal[1], Soumi Sadhu[2]
[1]Dept. of I.T, [2]Dept. of CSE,
Kalyani Govt. Engg College, Kalyani ,Nadia(W.B),India,
satyen_kgec@rediffmail.com, soumi19kgec@gmail.com

Abstract: *Harmony Search Algorithm originally came from the analogy between the music improvisation and optimization process. The harmony search algorithm is searching the optimal solution using the novel stochastic derivative equation. The equation is based on density information of multiple solution vectors, instead of calculus based gradient information of single solution vector. In other words, if a certain value frequently appears in multiple vectors, then the value has higher chance to be selected rather than the other candidate values. In this paper, an algorithm has been proposed to solve the Sudoku problem based on harmony search. A comparative study has been made with the proposed algorithm with the existing solution of Sudoku problem by harmony search. It has been proved that the proposed algorithm minimized the iterations to solve the Sudoku problem.*

1. Introduction

Suduko Puzzle is a problem which can be solved in very eassy manner by using HS algorithm. Sudoku, which is Japanese term meaning “singular number,” has gathered popularityi n Japan, the UK, and the USA. The Sudoku puzzle consists of 9 × 9 grid and 3 × 3 blocks for all the 81 cells. Each puzzle, which has a unique solution, has some cells that have already been filled in. The objective of the puzzle is to fill in the remainingcells with the numbers 1 through 9 so that the following three rules are satisfied:• Each horizontal row should contain the numbers 1 - 9, without repeating any. Each vertical column should contain the numbers 1 - 9, without repeating any. Each 3 × 3 block should contain the numbers 1 - 9, without repeating any.

In recent years, researchers have started to apply various methods such as graph theory, artificial intelligence, and genetic algorithm to solve the Sudoku puzzle. Eppstein[1] used the transformation from a directed or undirected graph to an unlabeled digraph to solve the puzzle. Although, it was successful to the undirected case, the method is not successful to a directed one because the latter is NP-complete. Caine and Cohen[2] proposed an artificial intelligent model named MITS (Mixed Initiative Tutoring System for Sudoku), in which the tutor takes the initiative to interact when the student lacks knowledge and makes moves that have low utility. Nicolau and Ryan[3] developed a system named GAuGE (Genetic Algorithm using Grammatical Evolution) for Sudoku, which uses a position independent representation. Each phenotype variable is encoded as a genotype string along with an associated phenotype position to learn linear relationships between variables.

Recently, a musicians' behavior-inspired evolutionary algorithm, harmony search(HS)([6]-[13]), has been developed and applied to various optimization problems such as structural design, water network design , dam scheduling, traffic routing ,satellite heat pipe design , oceanic structure mooring, hydrologic parameter calibration , and music composition From its success in various applications, HS in this study tackles the board game Sudoku, which can be formulated as an optimization problem with minimal violations of the above-mentioned three rules.Zong Woo Geam has worked on it by using some different parameter.(HMS=50,HMCR=0.9,Par=0.01).

The paper is divided into the following sections. In third section, the Sudoku problem and some basic terms related to harmony search are defined. The proposed algorithm based on harmony search is given in section 4. The algorithm is implemented at section 5. The result and conclusion with future score are furnished at section 6 and 7. Finally, the related references are given at last section.

3. Theory

3.1 Sudoku problem

The objective of the Sudoku problem[14] is to fill in the cells with the numbers 1 through 9 only once while satisfying the above-mentioned three rules. In other words, the problem can be formulated as an optimization problem as follows:

$$\text{Minimize } z=\sum_{i=1}^{9} \left| \sum_{i=1}^{9} x(i,j)-45 \right| + \sum_{j=1}^{9} \left| \sum_{i=1}^{9} x(i,j)-45 \right| + \sum_{k=1}^{9} \left| \sum_{(l,m) \in B(k)} x(l,m) - 45 \right| \quad (1).$$

where x_{ij} = cell at row *i* and column *j* , which has integer value from 1 to 9; and

B_k = set of coordinates for block *k*.

The first term in Equation 1 represents the penalty function for each horizontal row; the second term for each vertical column; and the third term for each block. It Should be noted that, although the sum of each row, each column, or each block equals 45, it does not guarantee that the numbers 1 through 9 are used exactly once. However, any violation of the uniqueness affects other row, column, or blocks which contains the wrong value jointly. The instance of Sudoku problem and Sudoku solution are Fig1. and Fig 2.

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

Fig 1. Sudoku Problem

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

Fig 2. Sudoku Problem

3.2 Harmony search and its Parameters

3.2.1 Harmony search

Harmony search (HS) algorithm is based on natural musical performance processes that occur when a musician searches for a better state of harmony, such as during jazz improvisation. The engineers seek for a global solution as determined by an objective function, just like the musicians seek to find musically pleasing harmony as determined by anesthetic . In music improvisation, each player sounds any pitch within the possible range, together making one harmony vector. If all the pitches make a good solution, that experiences stored in each variable's memory, and the possibility to make a good solution is also increased next time. HS algorithm includes a number of optimization operators, such as the harmony memory (HM), the harmony memory size (HMS, number of solution vectors in harmony memory), the harmony memory considering rate (HMCR), and the pitch adjusting rate (PAR). In the HS algorithm, the harmony memory (HM) stores the feasible vectors, which are all in the feasible space. The harmony memory size determines how many vector sit stores. A new vector is generated by selecting the components of different vectors randomly in the harmony memory.

The steps in the procedure of harmony search are as follows:

Step1. Initialize the problem and algorithm parameters.

Step2. Initialize the harmony memory (HM).

Step3. Improvise a new harmony from the HM.

Step4. Update the HM.

Step5. Repeat Steps 3 and 4 until the termination criterion is satisfied.

3.2.2 Parameters

The various parameters that are used in Harmony Search algorithm are defined as follows:

3.2.2.1. HMS (Harmony Memory Size): Number of simultaneous solution vectors in harmony memory. It generally varies from 1 to 100. (typical value = 30)

3.2.2..2. HMCR (Harmony Memory Considering Rate): It indicates the rate or the percentage with which HSA will choose the decision variables from the historically stored values in the harmony memory. It generally varies from 0.7 to 0.99. (typical value = 0.9)

3.2.2.3. PAR (Pitch Adjustment Rate): It indicates the rate or the percentage with which the algorithm will choose the neighboring values of the harmonies chosen from harmony memory. It generally varies from 0.1 to 0.5. (typical value = 0.3)

3.2.2.4. Harmony Memory Initialization: The harmony memory matrix is initially filled with randomly generated solutions vectors within the defined range.

4. Proposed algorithm to Solve Suduko Problem Based on Harmony Search:

Input : Incomplete Soduko

Output : Comple sudoku

Method :

Step 1: a) Suduko is divided 9 , 3X3 blocks starting from upper left corner.
b) Choosing a cell randomly from incomplete soduko.

Step 2: Put any number from 1 to 9 which is not present in the corresponding row, coloumn or block.

Step 3: If more than one number is available ,then the value of the cell will be calculated by the equation no 1.

***x'(i,j) <- x(lower)+(HMCR).(x(upper)-x(lo*wer)). ——————-→1.**

Step 4: Repeat step 2 and step 3 until all value of the corresponding cell will be filled up.

Step 5: If any number is repeated after step 4 in same row,same coloumn or the same block,we have to choose the selected value by equation 2. (PAR=0.5,HMCR=0.8)

X''(i.j) ← x'(i.j) +? HMCR*PAR*0.5————————————————→2(a).

***X''(i.j) ← x'(i.j) -? HMCR*PAR*0.5———————————————-→* 2(b).**

Step 6 : Repeat step 5 until ,the selected cell will get the suitable value.

Step 7: Repeat step 5 and step 6 to replace all duplications in each cell.

Step 8 : End.

5. Implementation

The algorithm has been implemented by using an Sudoku Problem is shown fig 3.

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

Fig 3. Sudoku Problem

To implement the proposed algorithm, the value of parameters have been taken as HMS = 40, HMCR = 0.8, and PAR = 0.5.

The vacant cells have been filled up by step3 as furnished in fig4.

1.8	5	3.6	3	1	6	6.6	9	7
5.8	6.8	5.4	8.6	8	5	5.8	2	4
5.8	9	8	4	2	5.8	6	4.2	3
9	7.2	1	7.8	4.8	3	2	4.8	6
5.2	3	5.2	8.4	4.8	7.4	7.2	1	5
5	4	7	2	6	1	9	3.8	8
4	6	5	6.8	9	1.8	3	8	1
5.8	1	7.8	5	7	8	7.2	8.4	2
8	5.2	7.6	1	3	4	5	7	8.2

Fig4. Fill up vacant cell

The fig4 has been filled up by following way, the cell x(2,1) may have any value between 1,3,6,7 as corresponding row, column and block contains 2,4,6,8,9, ,so as per apply equation 1: The value of cell has been calculated as: 1+0.8*(7-1)=5.8.where 1=x(lower),7=x(upper).In this way , the 40 remaining cells have been filled up.

The value of all vacant cells of fig 4 sholud be changed in integer value. The value of the cells that have been changed by equation 2 are shown in fig 5.

The details of calculation have been done as follows.

Suppose, the value of cell(2,1) has been considered. The floating value of cell (2,1) should be replaced by any available value of the corresponding cell i.e 1,3,6,7.

As per the fig 3. value 1 is to be calculated by using equation 2.And required number of iteration to compute 1 from 5.8 is (5.8-1)/(0.8*0.5*0.5)=24. The pink color shows that the respective cell value has been appeared either in the same row or same column or same block.. Green color cell shows the value is fixed and final for that cell after the iteration.

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

Fig 5. The updated cell value by equation 2.

The value in cell x(2,1) has been repeated in cell x(3,1).so generate 3 from 1 using equation 2 .Required no of generation is=(3-1)/(0.8*0.5*0.5)=10. The repeated value has been replaced and shown in fig 6.

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

Fig 6.The updated cell value using equation 2.

The replace value 3 must be changed as it is again repeated its corresponding column. To change the value of cell (2,1) from 3 to 7 by using equation 2 as 3 has been repeated in cell (8,1).Required no of iteration is 20. The final Sudoku solution is furnished at fig 7.

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

Fig 7. Sudoku Solution

So for cell x(2,1) total no of iteration is=54 iterations.

In this way calculation for the remaining cells. So The max 57 iteraion is required for cell x(5,3).

6. Result

The number of iteration has been counted to put the suitable value in the blank cell. The number of required iteration for all cells is furnished in table 1. To put the suitable value in each sudoku cell, the number of iterations For each cell in shown in tabular form is furnish table 1.

Table 1: The number of iteration to put suitable value in each blank cell.

X(1,1)	2	X(1,3)	3	X(1,5)	1	X(1,7)	8
X(5,5)	5	X(2,2)	7	X(2,3)	43	X(2,4)	19
X(3,1)	26	X(3,6)	2	X(3,8)	5	X(4,2)	5
X(4,4)	6	X(4,5)	10	X(4,8)	5	X(5,1)	37
X(5,3)	57	X(5,4)	13	X(5,5)	5	X(5,6)	13
X(5,7)	32	X(5,9)	1	X(6,2)	1	X(6,6)	1
X(6,8)	5	X(7,2)	44	X(7,4)	5	X(7,9)	2
X(8,1)	1	X(8,7)	17	X(8,8)	13	X(9,2)	57
X(9,3)	50	X(9,5)	1	X(9,7)	1	X(9,9)	5

Form table 1, it has been observed that the maximum number of iteration to comple the sudoku is 57. The same Sudoku has been solved by Zong Woo Geem[14], he has used maximum iteration is 285. So, the result has been computed in this paper is better than the previous approach.

7. Conclusion and future work

In this paper, an algorithm has been proposed to solve the Sudoku problem. It has been proved that the algorithm has been produced the result with fewer number of iteration. The algorithm will be used in all level Sudoku and the performance of this algorithm will be verified. The heuristic algorithm will be applied in future to predict the harmony parameters based on Sudoku problem instances.

References:

[1] Eppstein, D.: Nonrepetitive Paths and Cycles ,” Graphs with Application to Sudoku”. ACM Computing Research Repository. cs.DS/0507053 (2005)

[2] Caine, A., Cohen, R.: A Mixed-Initiative Intelligent Tutoring System for Sudoku. In: Lamontagne, L., Marchand, M. (eds.) Canadian AI 2006. LNCS (LNAI), vol. 4013, pp. 550–561. Springer, Heidelberg (2006)

[3] Nicolau, M., Ryan, C.: Solving Sudoku with the GAuGE System. In: Collet, P., Tomassini,M., Ebner, M., Gustafson, S., Ekárt, A. (eds.) EuroGP 2006. LNCS, vol. 3905, pp. 213– 224. Springer, Heidelberg (2006)

[4] Yato, T., Seta, T.: Complexity and Completeness of Finding Another Solution and its Application to Puzzles. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 86, 1052–1060 (2003)

[5] Geem, Z.W., Kim, J.H., Loganathan, G.V.: A New Heuristic Optimization Algorithm: Harmony Search. Simulation. 76(2), 60–68 (2001)

[6] Lee, K.S., Geem, Z.W.: A New Structural Optimization Method Based on the Harmony Search Algorithm. Computers and Structures. 82(9-10), 781–798 (2004)

[7] Geem, Z.W.: Optimal Cost Design of Water Distribution Networks using Harmony Search. Engineering Optimization. 38(3), 259–280 (2006)

[8] Geem, Z.W.: Optimal Scheduling of Multiple Dam System Using Harmony Search Algorithm. In: Lecture Notes in Computer Science, vol. 4507, pp. 316–323 (2007)

[9] Geem, Z.W., Lee, K.S., Park, Y.: Application of Harmony Search to Vehicle Routing. American Journal of Applied Sciences. 2(12), 1552–1557 (2005)

[10] Geem, Z.W., Hwangbo, H.: Application of Harmony Search to Multi-Objective Optimization for Satellite Heat Pipe Design. In: Proceedings of 2006 US-Korea Conference on Science, Technology, & Entrepreneurship (UKC 2006). CD-ROM (2006)

[11] Ryu, S., Duggal, A.S., Heyl, C.N., Geem, Z.W.: Mooring Cost Optimization via Harmony Search. In: Proceedings of the 26th International Conference on Offshore Mechanics and Arctic Engineering (OMAE 2007), ASME. CD-ROM (2007)

[12] Kim, J.H., Geem, Z.W., Kim, E.S.: Parameter Estimation of the Nonlinear Muskingum Model Using Harmony Search. Journal of the American Water Resources Association. 37(5), 1131–1138 (2001)

[13] Geem, Z.W., Choi, J.–Y.: Music Composition Using Harmony Search Algorithm. In: Lecture Notes in Computer Science, vol. 4448, pp. 593–600 (2007)

[14] Zong Woo Geem," Harmony Search algorithm for Solving Sudoku ",Johns Hopkins University, Environmental Planning and Management Program, 729 Fallsgrove Drive #6133, Rockville, Maryland 20850, USA