Harmony Search Algorithm for Solving Sudoku

Zong Woo Geem

Johns Hopkins University, Environmental Planning and Management Program,
729 Fallsgrove Drive #6133, Rockville, Maryland 20850, USA
geem@jhu.edu

Abstract. Harmony search (HS) algorithm was applied to solving Sudoku puzzle. The HS is an evolutionary algorithm which mimics musicians' behaviors such as random play, memory-based play, and pitch-adjusted play when they perform improvisation. Sudoku puzzles in this study were formulated as an optimization problem with number-uniqueness penalties. HS could successfully solve the optimization problem after 285 function evaluations, taking 9 seconds. Also, sensitivity analysis of HS parameters was performed to obtain a better idea of algorithm parameter values.

Keywords: Sudoku puzzle, harmony search, combinatorial optimization.

1 Introduction

Sudoku, which is Japanese term meaning "singular number," has gathered popularity in 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 remaining cells 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 [1], artificial intelligence [2], and genetic algorithm [3] 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 [4]. 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.

B. Apolloni et al. (Eds.): KES 2007/WIRN 2007, Part I, LNAI 4692, pp. 371–378, 2007.

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), has been developed [5] and applied to various optimization problems such as structural design [6], water network design [7], dam scheduling [8], traffic routing [9], satellite heat pipe design [10], oceanic structure mooring [11], hydrologic parameter calibration [12], and music composition [13].

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.

2 Harmony Search Model

The objective of the Sudoku problem 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:

$$\text{Minimize } Z = \sum_{i=1}^{9}\left|\sum_{j=1}^{9} x_{ij} - 45\right| + \sum_{j=1}^{9}\left|\sum_{i=1}^{9} x_{ij} - 45\right| + \sum_{k=1}^{9}\left|\sum_{(l,m)\in B_k} x_{lm} - 45\right| \tag{1}$$

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 block which contains the wrong value jointly.

To this penalty-included optimization problem, HS was applied, which originally came from the behavioral phenomenon of musicians when they together perform improvisation [5]. HS basically mimics musician's behaviors such as memory consideration, pitch adjustment, and random consideration, but it also includes problem-specific features for some applications.

For the first step of the HS algorithm, solution vectors are randomly generated as many as HMS (harmony memory size), then they are stored in HM (harmony memory) as follows:

$$\begin{bmatrix} x_{11}^1 & x_{12}^1 & \cdots & x_{19}^1 \\ x_{21}^1 & x_{22}^1 & \cdots & x_{29}^1 \\ \vdots & \cdots & \cdots & \cdots \\ x_{91}^1 & x_{92}^1 & \cdots & x_{99}^1 \end{bmatrix} \Rightarrow Z(\mathbf{x}^1) \tag{2a}$$

$$\begin{bmatrix} x_{11}^2 & x_{12}^2 & \cdots & x_{19}^2 \\ x_{21}^2 & x_{22}^2 & \cdots & x_{29}^2 \\ \vdots & \cdots & \cdots & \cdots \\ x_{91}^2 & x_{92}^2 & \cdots & x_{99}^2 \end{bmatrix} \Rightarrow Z(\mathbf{x}^2) \tag{2b}$$

$$\cdots$$

$$\begin{bmatrix} x_{11}^{HMS} & x_{12}^{HMS} & \cdots & x_{19}^{HMS} \\ x_{21}^{HMS} & x_{22}^{HMS} & \cdots & x_{29}^{HMS} \\ \vdots & \cdots & \cdots & \cdots \\ x_{91}^{HMS} & x_{92}^{HMS} & \cdots & x_{99}^{HMS} \end{bmatrix} \Rightarrow Z(\mathbf{x}^{HMS}) \tag{2c}$$

where x_{ij}^n = cell at row i and column j in n$^{\text{th}}$ vector stored in HM; and $Z(\mathbf{x}^n)$ = function value for n$^{\text{th}}$ vector in HM.

For the next step, a new harmony in Equation 3 is improvised using one of the following three mechanisms: random selection, memory consideration, and pitch adjustment.

$$\mathbf{x}^{NEW} = \begin{bmatrix} x_{11}^{NEW} & x_{12}^{NEW} & \cdots & x_{19}^{NEW} \\ x_{21}^{NEW} & x_{22}^{NEW} & \cdots & x_{29}^{NEW} \\ \vdots & \cdots & \cdots & \cdots \\ x_{91}^{NEW} & x_{92}^{NEW} & \cdots & x_{99}^{NEW} \end{bmatrix} \tag{3}$$

Random Selection. For x_{ij}^{NEW} , random value is chosen out of value range ($1 \le x_{ij}^{NEW} \le 9$) with a probability of (1-HMCR). HMCR (0 ≤ HMCR ≤ 1) stands for harmony memory considering rate.

$$x_{ij}^{NEW} \leftarrow x_{ij}, \quad x_{ij} \in \{1, 2, \ldots, 9\} \quad \text{w.p.} \quad \mathit{(1 - HMCR)} \tag{4}$$

Memory Consideration. Instead of the random selection, the value can be chosen from any values stored in HM with a probability of HMCR.

$$x_{ij}^{NEW} \leftarrow x_{ij}, \quad x_{ij} \in \{x_{ij}^1, x_{ij}^2, \ldots, x_{ij}^{HMS}\} \quad \text{w.p.} \quad \mathit{HMCR} \tag{5}$$

Pitch Adjustment. Once one pitch is obtained in memory consideration rather than random selection, the obtained value may further move to neighboring values with a probability of HMCR × PAR while the original value obtained in memory consideration does not move with a probability of HMCR × (1-PAR). PAR (0 ≤ PAR ≤ 1) stands for pitch adjusting rate. Here, x_{ij}^{NEW} in the right hand side is the value originally obtained in memory consideration; and Δ is the amount of increment (Δ equals one if x_{ij}^{NEW} is not upper limit (9) or lower limit (1). Otherwise, Δ equals zero).

$$x_{ij}^{NEW} \leftarrow \begin{cases} x_{ij}^{NEW} + \Delta & \text{w.p.} \quad HMCR \times PAR \times 0.5 \\ x_{ij}^{NEW} - \Delta & \text{w.p.} \quad HMCR \times PAR \times 0.5 \\ x_{ij}^{NEW} & \text{w.p.} \quad HMCR \times (1 - PAR) \end{cases} \tag{6}$$

If the new harmony vector $\mathbf{x}^{NEW}$ is better than the worst harmony in the HM in terms of objective function value, $Z(\mathbf{x}^{NEW})$, the new harmony is included in the HM and the existing worst harmony is excluded from the HM.

If the HS model reaches MaxImp (maximum number of improvisations), computation is terminated. Otherwise, another new harmony is improvised by considering one of three mechanisms.

3 Applications

The HS model was applied to the Sudoku puzzle proposed by Nicolau and Ryan [3] as shown in Figure 1.

	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. 1. Example of Sudoku Puzzle

The HS model found the optimal solution without any violation after 285 function evaluations using HMS = 50, HMCR = 0.7, and PAR = 0.1. Figure 2 shows the history of reaching global optimum.

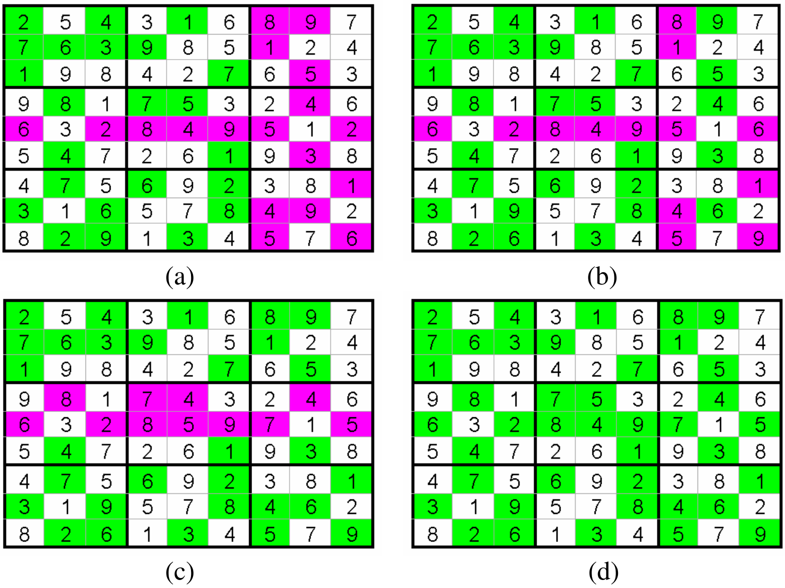

Fig. 2. Intermediate and Final Solutions of Test Sudoku Puzzle

While the green-colored cell (light-dark color in black & white) in Figure 2 means that there is no violation, the magenta-colored cell (dark color in black & white) indicates that there is at least one violation horizontally, vertically, or block-wise. Figure 2 (a) is the solution at 13 improvisations, which has a penalty of 21; Figure 2 (b) is the solution at 121 improvisations, which has a penalty of 5; Figure 2 (c) is the solution at 231 improvisations, which has a penalty of 2; and Figure 2 (d) is the solution at 285 improvisations, which has a penalty of 0.

This HS model further performed sensitivity analysis of algorithm parameters (HMS = {1, 2, 10, 50}, HMCR = {0.5, 0.7, 0.9}, PAR = {0.01, 0.1, 0.5}). Table 1 shows the analysis results. When only one vector is considered in the HM (HMS = 1), like simulated annealing or tabu search algorithm, the HS found global optimum except in one case (HMCR = 0.9, PAR = 0.1, Z = 6); When two vectors are considered in the HM (HMS = 2), partially similar to genetic algorithm, the HS also found global optimum except in two cases (HMCR = 0.7, PAR = 0.01, Z = 15; HMCR = 0.7, PAR = 0.1, Z = 27). However, when more than two vectors were considered in the HM (HMS = 10 or 50), there was no rule violation for the Sudoku example.

The HS computation was performed on Intel Celeron 1.8GHz CPU. The computing time ranged 4 - 38 seconds for HMS = 1 in order to arrive at the global optimum; for HMS = 2, it ranged 3 - 20 seconds; for HMS = 10, it ranged 3 - 8 seconds; and for HMS = 50, it ranged 7 - 12 seconds.

The HS model developed in this study was further applied to another Sudoku problem classified as "hard" as shown in Figure 3 [14]. When applied to the problem, the HS model was entrapped in one of local optima with a penalty of 14 after 1,064 function evaluations as shown in Figure 4.

Table 1. Results of Sensitivity Analysis with HS Parameters

HMS	HMCR	PAR	Iterations (Z)	Time (sec)
1	0.5	0.01	66	5
		0.1	337	10
		0.5	422	11
	0.7	0.01	287	13
		0.1	3,413	38
		0.5	56	4
	0.9	0.01	260	13
		0.1	**10,000** (6)	112
		0.5	1,003	19
2	0.5	0.01	31	3
		0.1	94	6
		0.5	175	6
	0.7	0.01	102	6
		0.1	77	6
		0.5	99	7
	0.9	0.01	**10,000** (15)	98
		0.1	**10,000** (27)	135
		0.5	1,325	20
10	0.5	0.01	49	3
		0.1	280	8
		0.5	188	5
	0.7	0.01	56	4
		0.1	146	5
		0.5	259	8
	0.9	0.01	180	5
		0.1	217	8
		0.5	350	8
50	0.5	0.01	147	9
		0.1	372	10
		0.5	649	12
	0.7	0.01	165	7
		0.1	285	9
		0.5	453	12
	0.9	0.01	87	7
		0.1	329	10
		0.5	352	11

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

Fig. 3. Another Sudoku Example (Hard Level)

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

Fig. 4. Local Optimum for Hard Example

4 Conclusion

The HS, musicians' behavior-inspired evolutionary algorithm, challenged the Sudoku puzzle with 40 given values in the literature, and could successfully find the unique global solution. The total searching space for this case is $9^{41} = 1.33 \times 10^{39}$ if integer programming formulation is considered. The proposed HS model found the global optimum without any row, column or block violation after 285 function evaluations, taking 9 seconds on Intel Celeron 1.8 GHz Processor.

When sensitivity analysis of algorithm parameters was performed, the HS could reach the global optimum 33 times out of 36 runs, taking 3 - 38 seconds (median for 33 successful cases is 8 seconds).

However, it failed to find the global optimum for hard level case with 26 given values, which has the searching space of $9^{55} = 3.04 \times 10^{52}$. The HS model was instead entrapped in one of local optima with the penalty of 14 after 1,064 function evaluations.

For study in the future, the HS model should consider additional problem-specific heuristics in order to efficiently solve a harder puzzle.

References

1. Eppstein, D.: Nonrepetitive Paths and Cycles in 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. Web Sudoku (January 19, 2007), http://www.websudoku.com/