A Novel Hybrid Flower Pollination Algorithm with Chaotic Harmony Search for Solving Sudoku Puzzles

Osama Abdel-Raouf [*1], Mohamed Abdel-Baset[2] and Ibrahim El-henawy[3]
[1]*Department of Operations Research, faculty of Computers and Information, Menoufia University, Menoufia, Shebin-el-Kome, Egypt.*
Postal code: 32511.
[2]*Department of Operations Research, faculty of Computers and Informatics, Zagazig University, El-Zera Square, Zagazig, Sharqiyah, Egypt*
Postal code: 44519.
[3]*Department of Computer Science, faculty of Computers and Informatics, Zagazig University, El-Zera Square, Zagazig, Sharqiyah, Egypt.*
Postal code: 44519.

***Abstract*— Flower Pollination algorithm (FP) is a new nature-inspired algorithm, based on the characteristics of flowering plants .In this paper, a new hybrid optimization method called improved Flower Pollination Algorithm with Chaotic Harmony Search (FPCHS) is proposed. The method combines the standard Flower Pollination algorithm (FP) with the chaotic Harmony Search (HS) algorithm to improve the searching accuracy. The FPCHS algorithm is used to solve Sudoku puzzles. Numerical results show that the FPCHS is accurate and efficient in comparison with standard Harmony Search, (HS) algorithm.**

***Keywords*—Flower pollination algorithm, Harmony search, meta-heuristics, optimization, Sudoku, chaos, evolutionary algorithms.**

I. INTRODUCTION

Sudoku was invented by American Howard Garns in 1979 and published by Dell Magazines as "Numbers in place" [1]. In 1984, Maki Kaji of Japan published it in the magazine of his puzzle company Nikoli. He gave the game its modern name of Sudoku, which means "Single Numbers." The puzzle became popular in Japan and was discovered there by New Zealander Wayne Gould, who then wrote a computer program that would generate Sudokus. He was able to get some puzzles printed in the London newspaper The Times beginning in 2004. Soon after, Sudoku-fever swept England. The puzzle finally became popular in the U.S. in 2005. It has become a regular feature in many newspapers and magazines and is enjoyed by people all over the globe.

The standard version of Sudoku consists of 9 × 9 grid and 3 × 3 blocks for all the 81 cells. Each puzzle starts with some cells that already have numbers as shown in figure 3(the numbers in white cells are originally given). The goal of the puzzle is to find numbers for the remaining cells with three rules:

i. Each horizontal row should contain the numbers 1 - 9, without repeating any.
ii. Each vertical column should contain the numbers 1 - 9, without repeating any.
iii. Each 3 × 3 block should contain the numbers 1 - 9, without repeating any.

This problem can be mathematically represented as an optimization problem as follows [2]:

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

where x_{ij} is a cell at row i and column j, which has an 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 [2]. 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 [2].

Harmony Search (HS) is an evolutionary algorithm, which mimics musicians' behaviors, such as random play, memory-based play, and pitch-adjusted play when they perform. HS has proved to be a powerful tool for solving several optimization problems [22-33]. It does not need any mathematical calculations to obtain the optimal solutions. In recent years, HS was been applied to many optimization problems, demonstrating its efficiency

compared to other heuristic algorithms and other Meta mathematical optimization techniques. Continuous development improvements to the algorithm and various applications to new types of problems (operations research, economy, computer science**,** civil engineering and electrical engineering), indicate that HS is a preferred choice. As a result in this, many studies have been proposed to increase its efficiency.

Sudoku puzzles in this researchwere formulated as an optimization problem with number-uniqueness penalties. In order to compare the performance of proposed algorithm and standard HS, proposed algorithm was applied to solving the Sudoku puzzles taken from Figure 1 (easy level) and Figure 3 (hard level) of Geem [2] and it was found that the proposed algorithm in example1 takes time less than standard HS, it takes3 second after 76 function evaluations, but The standard HS takes285 function evaluations, taking 9 seconds. in example 2 the proposed algorithm take 6 second after237function evaluations with the penalty of 0 but standard harmony search entrapped in one of local optima with the penalty of 14 after 1,064 function evaluations.

This paper is organized as follows: after introduction, Literature review of Sudoku is shortly displayed. In section 3, the Flower Pollination Algorithms briefly introduced. In section 4, the harmony search is briefly introduced. In section 5, the proposed algorithm is described, while the results are discussed in section 6. Finally, conclusions and future work are presented in section 7.

II. LITERATURE REVIEW OF SOLUTION TECHNIQUES FOR SUDOKU PUZZLES

In recent years researchers have been conducted into the derivation of methods for solving Sudoku problems, such as graph theory [3], artificial intelligence [4], and genetic algorithm [5]. Eppstein [3] used the transformation from a directed or undirected graph to an unlabeled digraph to solve the puzzle. Although it was successful in the undirected case, the method was not successful in a directed one, because the latter is NP-complete [6]. Caine and Cohen [4] planned 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 [5] developed a system named GAuGE (Genetic Algorithm using Grammatical Evolution) for Sudoku, which used a position independent representation. Each phenotype variable was encoded as a genotype string along with an associated phenotype position to learn linear relationships between variables. Moraglio et al. [7] Proposed geometric crossover operators to be used with an evolutionary algorithm for solving Sudoku. The algorithm was tested on five problems, three easy, one medium and one hard and was able to find solutions for all problems except the medium problem.

Bee colony optimization has also been applied to solving Sudoku puzzles [8], in which each Sudoku puzzle was represented as a two-dimensional array. The algorithm used solves the puzzle by emulating the process used by bees when foraging for food. Pacurib et al. [9] have also successfully applied a bee colony optimization algorithm to solve Sudoku puzzles. Lewis [10] employed a process using simulated annealing to solve Sudoku puzzles published in daily UK newspapers. The process began by randomly assigning numbers of empty cells in such a manner that each square contains just one occurrence of each number. This initial potential solution was then improved using simulated annealing. The approach found solutions for all puzzles it was applied to. Moraglio et al. [11] evaluate geometric particle swarm optimization as a means of solving Sudoku puzzles. This approach was able to find a solution to the problem it was tested on but the success rate over fifty runs was not as high as other methods applied to the same problems. Machado and Chaimowicz [12] combined a constraint satisfaction algorithm and simulated annealing to solve Sudoku puzzles. Deng and Li [13] used a hybrid approach by combining genetic algorithms and particle swarm optimization to solve Sudoku puzzles. Gem [2] used harmony search algorithm for solving Sudoku it was successful for easy level, it failed to find the global optimum for the hard level case**.**

III. THE ORIGINAL FLOWER POLLINATION ALGORITHM

Flower Pollination Algorithm (FPA) was founded by Yang in the year 2012. Inspired by the flow pollination process of flowering plants are the following rules:

> ***Rule 1*:** Biotic and cross-pollination can be considered as a process of global pollination process, and pollen-carrying pollinators move in a way that obeys Le'vy flights.
> *Rule 2:* For local pollination, a biotic and self-pollination are used.
> *Rule 3:* Pollinators such as insects can develop flower constancy, which is equivalent to a reproduction probability that is proportional to the similarity of two flowers involved.
> *Rule 4:* The interaction or switching of local pollination and global pollination can be controlled by a switch probability $p \in [0,1]$, with a slight bias toward local pollination .

In order to formulate updating formulas, we have to convert the aforementioned rules into updating equations. For example, in the global pollination step, flower pollen gametes are carried by pollinators such as insects, and pollen can travel over a long distance because insects can often fly and move in a much longer range[7].Therefore, Rule 1 and flower constancy can be represented mathematically as:

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)(x_i^t - B) \qquad (2)$$

Where x_i^t is the pollen i or solution vector $x_i$at iteration t, and $B$is the current best solution found among all solutions at the current generation/iteration. Here γ is a scaling factor to control the step size. In addition, $L(\lambda)$ is the parameter that corresponds to the strength of the pollination, which essentially is also the step size. Since insects may move over a long distance with various distance steps, we can use a Le'vy flight to imitate this characteristic efficiently. That is, we draw $L > 0$ from a Levy distribution:

$$L \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{S^{1+\lambda}}, (S >> S_0 > 0) \quad (3)$$

Here, Γ(λ) is the standard gamma function, and this distribution is valid for large steps $s > 0$.

Then, to model the local pollination, both Rule 2 and Rule 3 can be represented as

$$x_i^{t+1} = x_i^t + U(x_j^t - x_k^t) \qquad (4)$$

Where x_j^t and x_k^t are pollen from different flowers of the same plant species. This essentially imitates the flower constancy in a limited neighborhood. Mathematically, if x_j^t and x_k^t comes from the same species or selected from the same population, this equivalently becomes a local random walk if we draw U from a uniform distribution in [0, 1].Though Flower pollination activities can occur at all scales, both local and global, adjacent flower patches or flowers in the not-so-far-away neighborhood are more likely to be pollinated by local flower pollen than those faraway. In order to imitate this, we can effectively use the switch probability like in Rule 4 or the proximity probability p to switch between common global pollination to intensive local pollination. To begin with, we can use a naive value of $p = 0.5$as an initially value. A preliminary parametric showed that $p = 0.8$ might work better for most applications [7].

The basic steps of FP can be summarized as the pseudo-code shown in Figure 1.

Flower pollination algorithm
Define Objective function f (x), x = (x1, x2, ..., xd)
Initialize a population of n flowers/pollen gametes with random solutions
Find the best solution B in the initial population
Define a switch probability p ∈ [0, 1]
Define a stopping criterion (either a fixed number of generations/iterations or accuracy)
while *(t <MaxGeneration)*
for *i* = 1 : *n (all n flowers in the population)*
if *rand <p,*
Draw a (d-dimensional) step vector L which obeys a L´evy distribution
Global pollination via $x_i^{t+1} = x_i^t + L(B - x_i^t)$
else
Draw U from a uniform distribution in [0,1]
Do local pollination via $x_i^{t+1} = x_i^t + U(x_j^t - x_k^t)$
end if
Evaluate new solutions
If new solutions are better, update them in the population
end for
Find the current best solution B
end while
Output the best solution found

Fig. 1 Pseudo code of the Flower pollination algorithm

IV. HARMONY SEARCH ALGORITHM

Harmony search (HS) algorithm is a metaheuristic algorithm for solving optimization problems [14]. It has been applied to solve diverse types of problems in the past years and the results were very effective compared with other meta-heuristic algorithms and traditional techniques that computationally expensive.

In HS, the Harmony Memory (HM) stores feasible vectors, which are all in the feasible space. When a musician improvises one pitch, usually one of three rules is used:

i. Generating any one pitch from his/her memory, i.e. choosing any value from harmony memory, defined as memory consideration;

ii. Generating a nearby pitch of one pitch in his/her memory, i.e. choosing an adjacent value of one value from harmony memory, defined as pitch adjustments;

iii. Generating totally a random pitch from possible sound ranges, i.e. choosing totally random value from the possible value range, defined as randomization.

Similarly, when each decision variable chooses one value in HS, it can apply one of the above mentioned rules in the entire algorithm. If a New Harmony vector is better than the worst harmony vector in HM, then the New Harmony vector replaces it. This procedure is repeated until a stopping criterion is satisfied [14]. The procedure of HS is composed of five steps:

Step 1: Parameters Initialization

The optimization problem is specified as follows:

Minimize $f(x)$

Subject to

$$x_i \in X_i = 1,2,\dots,N \qquad (5)$$

Where $f(x)$ is an objective function; x the set of each decision variable x_i; N the number of decision variables, X_i a set of the possible range of values for each decision variable, that is, $x_i^{lower} \le x_i \le x_i^{upper}$ and x_i^{lower} and x_i^{upper} are the lower and upper boundaries for each decision variable, respectively.

The algorithm requires several parameters: Harmony Memory Size (HMS), Maximum number of Improvisations (MaxImp), Harmony Memory Considering Rate (HMCR), Pitch Adjusting Rate (PAR), Bandwidth vector (Bw).

Step 2: Harmony Memory Initialization and Evaluation

The harmony memory (HM) is a memory location where all the solution vectors and corresponding objective function values are stored. In this step, the HM matrix is filled with solution vectors by taking the possible ranges into account and fitness function values are calculated for each solution vector.

Step 3: Improvise a new harmony from the HM

A new harmony vector $X' = (x_1', x_2', x_3', \dots, x_N')$ is generated based on three mechanisms, namely, random selection, memory consideration, and pitch adjustment. The value of a design variable can be selected from the values stored in HM with a probability HMCR. It can be further adjusted by moving it to a neighbor value of a selected value from the HM with a probability of pitch adjusting rate (PAR), or, it can be selected randomly from the set of all candidate values without considering the stored values in HM, with the probability of $(1 - HMCR)$.

Step 4: Update the HM

If the new generated harmony vector is better than the current worst vector, based on the objective value and/or constraint violation, the new vector will replace the worst one.

Step 5: Stopping criterion check

If the stopping criterion (or maximum number of improvisations) is satisfied, the computation is terminated. Otherwise, Steps 3 and 4 are repeated.

V. THE PROPOSED ALGORITHM (FPCHS) FOR SOLVING SUDOKU

Newly chaos extends to various optimization areas, for example combinatorial optimization, because it can more easily escape from local minima in comparison with other stochastic optimization algorithms [15-20]. Using chaotic sequences in HS can be helpful to improve the global convergence, and to prevent getting stuck in local solutions than the classical HS algorithm which uses fixed values for HMCR, PAR and BW. In the proposed algorithm we used chaotic HS algorithm, where HMCR, PAR and BW values have been not fixed they have been modified by selecting chaotic maps. The steps of the FPCHS algorithm for solving Sudoku puzzles are shown in the figure 2.

The proposed algorithm

Begin
Define objective function $f(\mathbf{x})$, $\mathbf{x}=(x_1,x_2,\dots,x_d)^T$
Define the HS algorithm parameters.
Generate Harmony Memory by iterating the selected chaotic maps until reaching to the HMS.
while *(t<max number of iterations)*
while *(i<=number of variables)*
if *(rand<HMCR), Choose a value from HM for the variable i*
if *(rand<PAR), Adjust the value by adding certain amount*
end if
else *Choose a random value*
end if
end while
Accept the new harmony (solution) if better
end while
Find the current best solution
end
The best solution found by HS is regarded as initial for FP algorithm B
Define a switch probability $p \in [0, 1]$
while *(t <MaxGeneration)*
for *i = 1 : n (all n flowers in the population)*
if *rand < p,*
Draw a (d-dimensional) step vector L which obeys a L´evy distribution
Global pollination via $x_i^{t+1} = x_i^t + L(B - x_i^t)$
else
Draw U from a uniform distribution in [0,1]
Do *local pollination via* $x_i^{t+1} = x_i^t + U(x_j^t - x_k^t)$
end if
Evaluate new solutions
If *new solutions are better, update them in the population*
end for
Find the current best solution B
end while
End

Fig. 2 Pseudo code of the FPCHS algorithm

VI. APPLICATIONS

Numerous examples have been used to test the performance of the proposed algorithm.
To begin with, the initial parameters are set as follows: HMS=50and MaxImp=10000 We performed 70 runs on all puzzles. The selected chaotic map for all examples is the logistic map, whose equation is shown below:

$$Y_{n+1} = \mu Y_n(1 - Y_n) \qquad (6)$$

Clearly, $Y_n \in [0,1]$ under the conditions that the initial $Y_0 \in [0,1]$, where n is the iteration number and $\mu = 4$. All the experiments were performed on a Windows 7 Ultimate 64-bit operating system; processor Intel Core i5 760 running at 2.81 GHz; 4 GB of RAM and codes were written in Visual c#.

A. Example 1

This example has been introduced by Geem [2] and was successfully solved by the standard harmony search with 285 function evaluations, taking 9 seconds. In this paper the proposed algorithm could successfully solve it with a fewer number of function evaluation (only 76 function evaluations) and within three seconds.

	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 Example 1 Sudoku puzzle

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	3	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. 4 Final solution of example 1 Sudoku puzzle

B. Example 2

In this example, standard HS algorithm applied to this problem [20] by Geem [2] and failed to find the global optimum, as shown in figure 5[20]

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.4Example 2 of Sudoku puzzle

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. 5 Standard HS solution for example 2

When applied the proposed algorithm for this example we found that the proposed algorithm could successfully find the global optimal without any row, column or block violation after 237 function evaluations taking 6 seconds. As shown in figure 6

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

Fig.6. Final solution of example 2Sudoku puzzle with proposed algorithm

C. Example 3

Another Sudoku problem classified as "hard" as shown in Figure 7[21]. Also the proposed algorithm successfully found the global optimal without any row, column or block violation after 281 function evaluations taking 9 seconds. As shown in figure 8

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

Fig.7. Example3Sudoku puzzle

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

Fig.8. Final solution of example 3 Sudoku puzzle

VII. CONCLUSION AND FUTURE WORK

In this paper, the better results (i.e. solve a hard level of Sudoku and minimum numbers of iterations have been computed as a compare with paper [2] in same Sudoku examples).This paper presented an improved flower pollination algorithm by integrating it with chaotic harmony search algorithm for solving Sudoku puzzles. The HM members were fine-tuned by the chaos operator to improve their affinities. The proposed algorithm has been provided the better and clear way to find the solution of hard level of any given Sudoku.

The algorithm has been tested on a set of Sudoku problems. The results proved the superiority of the proposed methodology. The reason for getting better results in comparison with other algorithms considered is the search power of FP. In addition to that, using Chaos with harmony search helps the algorithm to escape from local solutions.

Future work includes different puzzles such as Futoshiki and KenKen can be solved using this algorithm. Another direction can be the study of different mechanisms for adapting the HMCR , PAR , bw and P parameters. Also we can use this algorithm for solving most popular engineering optimization problems.

REFERENCES

[1] Web http://www.math.cornell.edu/~mec/Summer2009/Mahmood/Intro.html . Nov. 30, 2013.

[2] Z. W. Geem, "Harmony search algorithm for solving sudoku," in *Knowledge-Based Intelligent Information and Engineering Systems*, 2007, pp. 371-378.

[3] D. Eppstein, "Nonrepetitive paths and cycles in graphs with application to Sudoku," *arXiv preprint cs/0507053*, 2005.

[4] A. Caine and R. Cohen, "MITS: A mixed-initiative intelligent tutoring system for sudoku," in *Advances in Artificial Intelligence*, ed: Springer, 2006, pp. 550-561.

[5] M. Nicolau and C. Ryan, "Solving sudoku with the gAuGE system," in *Genetic Programming*, ed: Springer, 2006, pp. 213-224.

[6] Y. Takayuki and S. Takahiro, "Complexity and completeness of finding another solution and its application to puzzles," *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 86, pp. 1052-1060, 2003.

[7] A. Moraglio, J. Togelius, and S. Lucas, "Product geometric crossover for the sudoku puzzle," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, 2006, pp. 470-476.

[8] A. Kaur and S. Goyal, "A Survey on the Applications of Bee Colony Optimization Techniques," *International Journal on Computer Science and Engineering (IJCSE)*, vol. 3, pp. 3037-3046, 2011.

[9] J. A. Pacurib, G. M. M. Seno, and J. P. T. Yusiong, "Solving sudoku puzzles using improved artificial bee colony algorithm," in *Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on*, 2009, pp. 885-888.

[10] R. Lewis, "Metaheuristics can solve sudoku puzzles," *Journal of heuristics*, vol. 13, pp. 387-401, 2007.

[11] A. Moraglio and J. Togelius, "Geometric particle swarm optimization for the sudoku puzzle," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, 2007, pp. 118-125.

[12] M. C. Machado and L. Chaimowicz, "Combining Metaheuristics and CSP Algorithms to Solve Sudoku," in *Games and Digital*

Entertainment (SBGAMES), 2011 Brazilian Symposium on, 2011, pp. 124-131.

[13] X. Q. Deng and Y. Da Li, "A novel hybrid genetic algorithm for solving Sudoku puzzles," *Optimization Letters*, vol. 7, pp. 241-257, 2013.

[14] Z. W. Geem, J. H. Kim, and G. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, pp. 60-68, 2001.

[15] H.-O. Peitgen, H. Jürgens, and D. Saupe, *Chaos and fractals: new frontiers of science*: Springer, 2004.

[16] B. Alatas, "Chaotic harmony search algorithms," *Applied Mathematics and Computation*, vol. 216, pp. 2687-2699, 2010.

[17] G. Heidari-Bateni and C. D. McGillem, "A chaotic direct-sequence spread-spectrum communication system," *Communications, IEEE Transactions on*, vol. 42, pp. 1524-1527, 1994.

[18] P. Gaspard, *Chaos, scattering and statistical mechanics* vol. 9: Cambridge University Press, 2005.

[19] C. Letellier, *Chaos in nature* vol. 81: World Scientific Publishing Company, 2013.

[20] P. Knight, "Deterministic Chaos: An Introduction," 1988.

[21] Web Sudoku. http://www.websudoku.com/ (Nov. 24, 2013)

[22] Z. Geem, J. Kim, and G. Loganathan, "Harmony search optimization: application to pipe network design," *International journal of modelling & simulation*, vol. 22, pp. 125-133, 2002.

[23] Z. W. Geem, C.-L. Tseng, and Y. Park, "Harmony search for generalized orienteering problem: best touring in China," in *Advances in natural computation*, ed: Springer, 2005, pp. 741-750.

[24] J. H. Kim, Z. W. Geem, and E. S. Kim, "Parametersestimationofthenonlinear muskingummodelusingHarmonysearch," *JAWRA Journal of the American Water Resources Association*, vol. 37, pp. 1131-1138, 2001.

[25] K. S. Lee and Z. W. Geem, "A new structural optimization method based on the harmony search algorithm," *Computers & Structures*, vol. 82, pp. 781-798, 2004.

[27] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied mathematics and computation*, vol. 188, pp. 1567-1579, 2007.

[28] M. G. Omran and M. Mahdavi, "Global-best harmony search," *Applied Mathematics and Computation*, vol. 198, pp. 643-656, 2008.

[29] Q.-K. Pan, P. N. Suganthan, M. F. Tasgetiren, and J. J. Liang, "A self-adaptive global best harmony search algorithm for continuous optimization problems," *Applied Mathematics and Computation*, vol. 216, pp. 830-848, 2010.

[30] Q.-K. Pan, P. Suganthan, J. Liang, and M. F. Tasgetiren, "A local-best harmony search algorithm with dynamic subpopulations," *Engineering Optimization*, vol. 42, pp. 101-117, 2010.

[31] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham, and B. K. Panigrahi, "Exploratory power of the harmony search algorithm: analysis and improvements for global numerical optimization," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 41, pp. 89-106, 2011.

[32] D. Zou, L. Gao, J. Wu, S. Li, and Y. Li, "A novel global harmony search algorithm for reliability problems," *Computers & Industrial Engineering*, vol. 58, pp. 307-316, 2010.

[33] Z. W. Geem, K. S. Lee, and Y. Park, "Application of harmony search to vehicle routing," *American Journal of Applied Sciences*, vol. 2, p. 1552, 2005.