TECHNICAL ARTICLE

SIMULATION 76:2,60-68

ISSN 0037-5497/01
Printed in the United States of America

A New Heuristic Optimization Algorithm: Harmony Search

Zong Woo Geem and **Joong Hoon Kim**[1]
Department of Civil and Environmental Engineering
Korea University, Seoul, South Korea 136-701

G. V. Loganathan
Department of Civil and Environmental Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

Many optimization problems in various fields have been solved using diverse optimization algorithms. Traditional optimization techniques such as linear programming (LP), non-linear programming (NLP), and dynamic programming (DP) have had major roles in solving these problems. However, their drawbacks generate demand for other types of algorithms, such as heuristic optimization approaches (simulated annealing, tabu search, and evolutionary algorithms). However, there are still some possibilities of devising new heuristic algorithms based on analogies with natural or artificial phenomena. A new heuristic algorithm, mimicking the improvisation of music players, has been developed and named Harmony Search (HS). The performance of the algorithm is illustrated with a traveling salesman problem (TSP), a specific academic optimization problem, and a least-cost pipe network design problem.

Keywords: Harmony search, optimization, heuristic algorithm, combinatorial optimization, music

1. Introduction

Today's highly capitalized societies require "maximum benefit with minimum cost." For achieving this goal, we usually depend on optimization techniques. Many problems in various fields are formulated as optimization problems and solved using various optimization algorithms. Over the decades, the development and application of optimization models have attracted growing attention among engineers.

Traditional mathematical techniques, such as linear programming (LP), non-linear programming (NLP), and dynamic programming (DP), have been frequently used for solving the optimization problems. All three techniques can guarantee global optima in simple and ideal models. However, in real world problems, there are some drawbacks: in LP, considerable losses occur when a linear ideal model from a non-linear real world problem is developed; in DP, an increase in the number of variables would exponentially increase the number of evaluations of the recursive functions and tax the core-memory (the "curse of

[1]Please direct all correspondence to
Zong Woo Geem (zwgeem@yahoo.com).

dimensionality"); in NLP, if the functions used in computation are not differentiable, the solving algorithm may not find the optimum. Careful attention is also required in selecting the initial values in order to guarantee convergence to the global optimum and not into local optima.

In order to overcome the above deficiencies of mathematical techniques, heuristic optimization techniques based on simulation have been introduced. These allow a good solution (near optimum) to be found within a reasonable computation-time and with reasonable use of memory without any loss of subtle nonlinear characteristics of the model and without any requirement of complex derivatives or careful choice of initial values.

Since the 1970s, many heuristic algorithms have been developed that combine rules and randomness mimicking natural phenomena. These techniques include: simulated annealing (SA), tabu search (TS), and evolutionary algorithms (EA).

In 1983, Kirkpatrick et al. [1] proposed the innovative idea of the simulated annealing algorithm. The simulated annealing algorithm is based on an analogy of the physical annealing process. In physical annealing, a material's temperature is increased to give mobility to the molecules, and gradually those molecules will form a crystalline structure, that is, an optimal solution. They modeled their approach after the stochastic thermal equilibrium process by Metropolis et al. [2] to solve a classic combinatorial optimization problem (the traveling salesman problem) and good results were obtained. Since then, many engineering problems have been successfully solved by this kind of algorithm. Tabu search is an iterative procedure for solving discrete combinatorial optimization problems. It was first suggested by Glover [3] and since then, has become widely used to obtain optimal solutions. The basic idea of the algorithm is to explore the search space of all feasible solutions by a sequence of moves. A move from one solution to another results in the best available solution. However, to escape from local optima and to prevent cycling, some moves are classified as forbidden or tabu. Tabu moves are based on the history of the sequence of moves.

Evolutionary algorithms (or evolutionary computation methods), based on a principle of evolution (survival of the fittest), and mimicking some natural phenomena (genetic inheritance), consist of basically four heuristic algorithms: genetic algorithm, evolution strategies, evolutionary programming and genetic programming.

The genetic algorithms are search algorithms based on natural selection and the mechanisms of population genetics. The theory was proposed by Holland [4] and further developed by Goldberg [5] and others. The simple GA is comprised of three operators: reproduction, crossover, and mutation. Reproduction is a process of survival-of-the-fittest selection. Crossover is the partial swap between two parent strings to produce two offspring strings. Mutation is the occasional random inversion of bit values, generating non-recursive offspring. The main characteristic of GA, which differs from traditional optimization techniques or heuristic methods such as simulated annealing and tabu search, is the simultaneous evaluation of many solutions; other techniques evaluate only one solution at every iteration. This feature can be an advantage, enabling a wide search and potentially avoiding convergence to a non-global optimum.

Evolution strategies (ES) were developed to solve parameter optimization problems [6]. ES uses deterministic ranking to select a basic set of solutions for a new trial [7]. A chromosome represents an individual as a pair of float values, $v = (x, \sigma)$. Here, the first vector $\mathbf{x}$ is a point in the search area, and the second vector σ is a vector of standard deviations. The major operation, mutation, is represented as

$$x^{t+1} = x^t + N(0, \sigma)$$

where $N(0, \sigma)$ is a vector of independent random Gaussian variables with means of zero and standard deviations of σ.

Evolutionary programming (EP) algorithms were originally developed by Fogel et al. [8], and described for the evolution of finite state machines to solve prediction tasks. The state transition tables in these machines are modified by uniform random mutations on the corresponding alphabet. The algorithms utilize selection and mutation as main operators, and the selection process is a stochastic tournament.

The genetic programming (GP) algorithm was developed relatively recently by Koza [9]. He suggested that the desired program should evolve itself during the evolution process. Genetic programming is similar to genetic algorithms. The main difference between the two algorithms is the representation of the solution. Genetic programming creates computer programs in LISP or scheme computer languages as the solution, while genetic algorithms create a string of numbers that represent the solution.

Simulation-based heuristic methods, discussed above, have powerful searching abilities, which can occasionally overcome the several drawbacks of traditional mathematical methods. The purpose of this paper is to propose a new heuristic optimization algorithm, which can produce better solutions than other existing algorithms in less number of iterations.

2. New Heuristic Algorithm: Harmony Search

Now it is time to ask a question. Is it possible to develop a new heuristic algorithm with better performance (better solutions, fewer iterations) than existing heuristic algorithms? Such an algorithm would serve as an attractive alternative to other already established algorithms.

As observed previously, existing heuristic methods mimic natural phenomena; SA, physical annealing; TS, human memory; EA, evolution. Therefore, a new algorithm might also be found in natural phenomena, or in artificial ones. An artificial phenomenon, musical harmony, can serve as the model for devising a new technique. Music is one of the most satisfying processes generated by human endeavors. A new heuristic algorithm derived from an artificial phenomenon found in musical performance (for example, a jazz trio), namely the process of searching for better harmony, can be introduced.

Music harmony is a combination of sounds considered pleasing from an aesthetic point of view. Harmony in nature is a special relationship between several sound waves that have different frequencies. Since the Greek philosopher and mathematician Pythagoras (582 BC-497 BC), many people have researched this phenomenon. The French composer and musicologist Jean-Philippe Rameau (1683-1764) established the classical harmony theory; the musicologist Tirro [10] has documented the thorough history of American jazz.

Musical performances seek a best state (fantastic harmony) determined by aesthetic estimation, as the optimization algorithms seek a best state (global optimum—minimum cost or maximum benefit or efficiency) determined by objective function evaluation. Aesthetic estimation is determined by the set of the sounds played by joined instruments, just as objective function evaluation is determined by the set of the values produced by component variables; the sounds for better aesthetic estimation can be improved through practice after practice, just as the values for better objective function evaluation can be improved iteration by iteration. A brief presentation of these observations is shown in Table 1.

The new algorithm is named Harmony Search (HS) and the steps in the procedure of HS are as follows:

- Step 1. Initialize a Harmony Memory (HM).
- Step 2. Improvise a new harmony from HM.

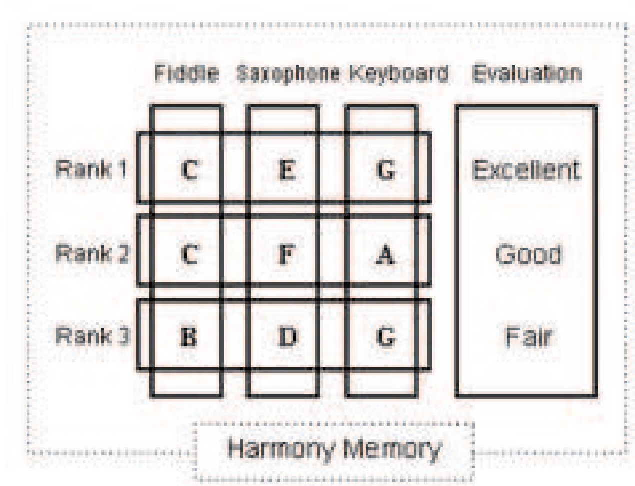


Figure 1. Structure of Harmony Memory (HM)

- Step 3. If the new harmony is better than minimum harmony in HM, include the new harmony in HM, and exclude the minimum harmony from HM.
- Step 4. If stopping criteria are not satisfied, go to Step 2.

The structure of Harmony Memory (HM) is shown in Figure 1. Consider a jazz trio composed of fiddle, saxophone and keyboard. Initially, the memory is stuffed with random harmonies: (C, E, G), (C, F, A), and (B, D, G) that are sorted by aesthetic estimation. In the improvising procedure, three instruments produce a new harmony; for example, (C, D, A): fiddle sounds {C} out of {C, C, B}; saxophone sounds {D} out of {E, F, D}; and keyboard sounds {A} out of {G, A, G}. Every note in HM has the same opportunity to be selected, for example, each of the notes E, F, or D of the saxophone in HM has a selection probability of 33.3%. If the newly made harmony (C, D, A) is better than any of the existing harmonies in the HM, the new harmony is included in HM and the worst harmony (in this example, (B, D, G)) is excluded from the HM. This process is repeated until satisfying results (near optimum) are obtained.

Table 1. Comparison between Optimization and Musical Performance

COMPARISON FACTOR	OPTIMIZATION PROCESS	PERFORMANCE PROCESS
Best state	Global Optimum	Fantastic Harmony
Estimated by	Objective Function	Aesthetic Standard
Estimated with	Values of Variables	Pitches of Instruments
Process unit	Each Iteration	Each Practice

For further understanding, consider the optimization problem expressed as:

$$Min\ f(x) = (x_1 - 2)^2 + (x_2 - 3)^4 + (x_3 - 1)^2 + 3$$

This is a minimization problem for which one can easily find the solution vector (2, 3, 1) for the global minimum. However, the Harmony Search finds the solution vector in another way.

As given in Figure 2a, the HM is initially structured with randomly generating values that are sorted by the value of objective function. Next, the new harmony (1, 2, 3) is improvised after consideration of the HM: x_1 chooses {1} out of {2, 1, 5}; x_2 chooses {2} out of {2, 3, 3}; x_3 chooses {3} out of {1, 4, 3}. Because the function value of the new harmony is 9, the new harmony (1, 2, 3) is included in the HM and the worst harmony (5, 3, 3) is excluded from the HM, as shown in Figure 2b. Finally, Harmony Search improvises the harmony (2, 3, 1), which has the function value of 3, the global minimum.

Of course, the above assumes that all the parts of the global solution exist initially in HM. When this is not the case, in order to find global optimum, Harmony Search initiates a parameter, Harmony Memory Considering Rate (HMCR), which ranges from 0 to 1. If a uniformly generated value between 0 to 1 occurs above the current value of the HMCR, then HS finds notes randomly within the possible playable range without considering HM. A HMCR of 0.95 means that at the next step, the algorithm chooses a variable value from HM with a 95% probability.

For improving solutions and escaping local optima, yet another option may be introduced. This option mimics the pitch adjustment of each instrument for tuning the ensemble. For computation, the pitch adjustment mechanism is devised as shifting to neighboring values within a range of possible values. If there are six possible values such as {1, 3, 4, 6, 7, 9}, {6} can be moved to neighboring {4} or {7} in the pitch adjusting process. A Pitch Adjusting Rate (PAR) of 0.10 means that the algorithm chooses a neighboring value with 10% probability (an upper value with 5% or a lower value with 5%).

Suppose that the set of possible values of an instrument (a variable) is {C, D, E, F, G}, HMCR is 0.95, PAR is 0.10, and the instrument now has {C, E, G} in HM. In the improvisation process of HS algorithm, the algorithm uniformly chooses one note out of {C, E, G} with 95% probability or one note out of {C, D, E, F, G} with 5% probability, and {E} can be shifted to {D} or {F} with 10% probability when {E} is chosen.

In order to demonstrate the convergence capability of harmony search, let us consider the harmony memory with the parameters: the size of HM (the number of harmonies in HM) = M, the number of instruments (variables) = N, the number of possible notes (values) of instruments = L, the number of optimal note (value) of instrument i in the HM = H_i, harmony memory considering rate = Hr, and the optimal harmony (optimal vector) = (C, E, G). The probability to find the optimal harmony, Pr(**H**) is

$$\Pr(\mathbf{H}) = \prod_{i=1}^{N}\left[Hr\frac{H_i}{M} + (1 - Hr)\frac{1}{L}\right]$$

where the pitch adjusting rate is not considered because it is an optional operator.

Initially, the HM is stuffed with random harmonies. If there is not any optimal note of all instruments in the HM,

$$H_1 = H_2 = ... = H_N = 0$$

and

$$\Pr(\mathbf{H}) = \left[(1 - Hr)\frac{1}{L}\right]^N$$

This means the probability Pr(**H**) is very low.

However, if the schema of optimal harmony such as (*, E, G), (C, *, G), (C, E, *) have better fitness (better evaluation) than other ones, the number of optimal notes of instrument i in the HM, H_i will be increased iteration by iteration. Consequently, the probability of finding the optimal harmony, Pr(**H**) will be increased.

Figure 2a. Initial Harmony Memory

	X_1	X_2	X_3	F
Rank 1	2	2	1	4
Rank 2	1	3	4	13
Rank 3	5	3	3	16

Figure 2b. Subsequent Harmony Memory

	X_1	X_2	X_3	F
Rank 1	2	2	1	4
Rank 2	1	2	3	9
Rank 3	1	3	4	13

As explained above, the Harmony Search incorporates, by nature, the structure of existing heuristic methods. It preserves the history of past vectors (Harmony Memory) similar to TS, and is able to vary the adaptation rate (Harmony Memory Considering Rate) from the beginning to the end of computation resembling SA, and manages several vectors simultaneously in a manner similar to GA. However, the major difference between GA and HS is that HS makes a new vector from all the existing vectors (all harmonies in the Harmony Memory), while GA makes the new vector only from two of the existing vectors (the parents). In addition, HS can independently consider each component variable in a vector while it generates a new vector, but GA cannot because it has to keep the structure of a gene.

3. Applications

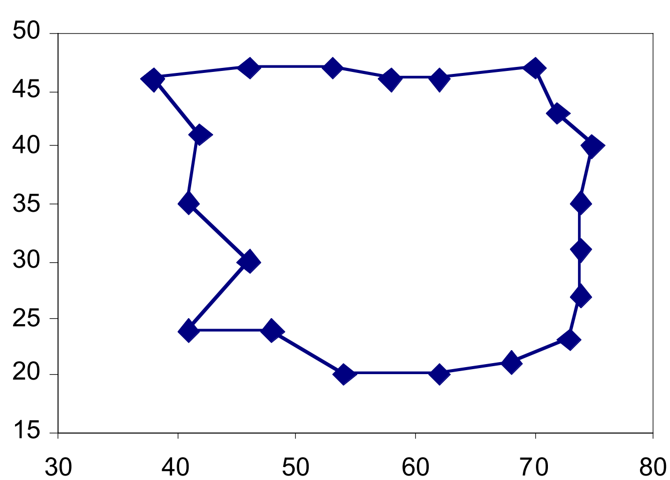
Three problems are presented in this paper to demonstrate the searching ability of the Harmony Search algorithm.

3.1 Problem 1

The HS is applied to a Traveling Salesman Problem (TSP). The objective of TSP is to find the shortest path for a travelling salesman who has to visit every city precisely once. The difficulty in this problem is the huge number of possible tours: (n-1)! / 2 for *n* cities.

The 20-city TSP applied is shown in Figure 3. The number of possible tours is $(20-1)!\ /\ 2 = 6.08 \times 10^{16}$. Nonetheless, the best route in this case can be easily measured with the eye. For solving a TSP using HS, each musical instrument in HM is substituted with a variable assigned for each city. Linking each city to its next assigned city creates one of the possible tours. The length of the tour is compared with those of existing tours in HM. If the new length is shorter than any of existing tour lengths, the new tour is included in HM, and the worst tour (longest tour) is excluded from HM.

Figure 3. 20-city traveling salesman problem and the shortest route measured by eye



Thirty runs using the HS were performed with different values of the parameters (HMCR = 0.85 - 0.99, the size of HM = 10 - 100). Seven out of 30 runs have reached global optimum (minimal length = 117) after up to 20,000 iterations. For faster convergence to the global optimum, two operators ((neighboring city)-going operator, city-inverting operator) have been introduced. The (neighboring city)-going operator makes the salesman visit the closest neighboring city; the city-inverting operator makes the salesman visit, for example, 1-3-2 instead of 1-2-3, if the former is shorter than the latter. Twenty-one runs employing these two operators were performed. After up to 5,000 iterations, 11 out of 21 runs could reach the shortest tour.

3.2 Problem 2

The second problem, originally given by Braken and McCormick [11] and cited by Himmelblau [12], is a relatively simple constrained minimization problem and involves continuous variables. The problem statement is:

$$\textit{Minimize}\ f(x) = (x_1 - 2)^2 + (x_2 - 1)^2$$

$$\textit{Subject to}\ g_1(x) = 0$$
$$g_2(x) \geq 0$$

with

$$g_1(x) = x_1 - 2x_2 + 1$$
$$g_2(x) = -x_1^2/4 - x_2^2 + 1$$

Homaifar et al. [13] attacked this problem using GA and compared the result of GA with the exact solution and the result obtained by the Generalized Reduced Gradient (GRG) method [14]. Fogel [7] compared the result of evolutionary programming with that of GA. The comparison of these methods to Harmony Search is shown in Table 2.

For the computation using HS, the number of objective function evaluations and the structure of the penalty functions are the same as those of GA and EP [13, 7]; the number of maximum evaluations (iterations) is set to 40,000. For the Harmony Search computation, the size of HM is 30, HMCR is 0.5, and the steps of pitches (the number of possible values for one variable) are 3,000. The components x_1 and x_2 are both initialized randomly with uniform distribution over the range [-10.0, 10.0]. After the first half of computing (20,000 iterations), the range has been narrowed to whatever values have been stored in HM, and the computation freshly starts again for the second half of computing. The computation time of 40,000 iterations is about 1 hour on a Pentium 100 MHz computer.

Table 2 shows the comparison of HS solutions with other solutions. The first HS solution, HS(1) (first one in HM) is in terms of the objective. However, HS(2) is

Table 2. Comparison of Results from Various Methods for Problem 2

	EXACT	GRG	GA	EP	HS(1)	HS(2)
f(x)	1.3935	1.3934	1.4339	1.3772	1.3771	1.3965
%	0.0000	-0.0072	+2.8992	-1.1697	-1.1769	+0.2153
x_1	0.82288	0.8229	0.8080	0.8350	0.8348	0.8290
x_2	0.91144	0.9115	0.8854	0.9125	0.9124	0.9080
g_1	7.05×10^{-9}	1.0×10^{-4}	3.7×10^{-2}	1.0×10^{-2}	1.0×10^{-2}	1.3×10^{-2}
g_2	1.73×10^{-8}	-5.2×10^{-5}	5.2×10^{-2}	-7.0×10^{-3}	-6.7×10^{-3}	3.7×10^{-3}

also the best solution among the heuristic approaches within terms of accuracy; third row (% row) in Table 2 shows the relative errors between the algorithm results and the exact function value. Although the GRG solution shows the closest objective value, it violates the inequality constraint (all ten solutions of the EP also violate the inequality constraint).

Harmony Search proves to outperform the other methods in this continuous-variable problem, but it is believed to show better performance in discrete and combinatorial problems as well.

3.3 Problem 3

HS has also been applied to the design of a pipeline network for water supply in Hanoi, Vietnam (Figure 4). Fujiwara and Khang [15, 16] first presented the network, consisting of 32 nodes, 34 links, and 3 loops. No pumping station is considered since a single fixed-head source at an elevation of 100 m is available. The minimum pressure-head requirement at all nodes is fixed at 30 m.

The objective of the application is to determine the minimal diameters of all pipelines that can satisfy the water pressure requirements in all nodes. Each of 34 pipelines (links) can be assigned one of 6 available commercial-diameter pipes (ready-made pipes in 6 different diameters) shown in Table 3. The total number of possible network designs is $6^{34} = 2.87 \times 10^{26}$.

Fujiwara and Khang [15, 16] solved the problem using the Nonlinear Programming Gradient (NLPG) and the local improvement method. The final optimal (continuous) diameters were converted to the discrete set of commercial diameters using the standard split pipe technique. Table 4 shows optimal solutions ob-

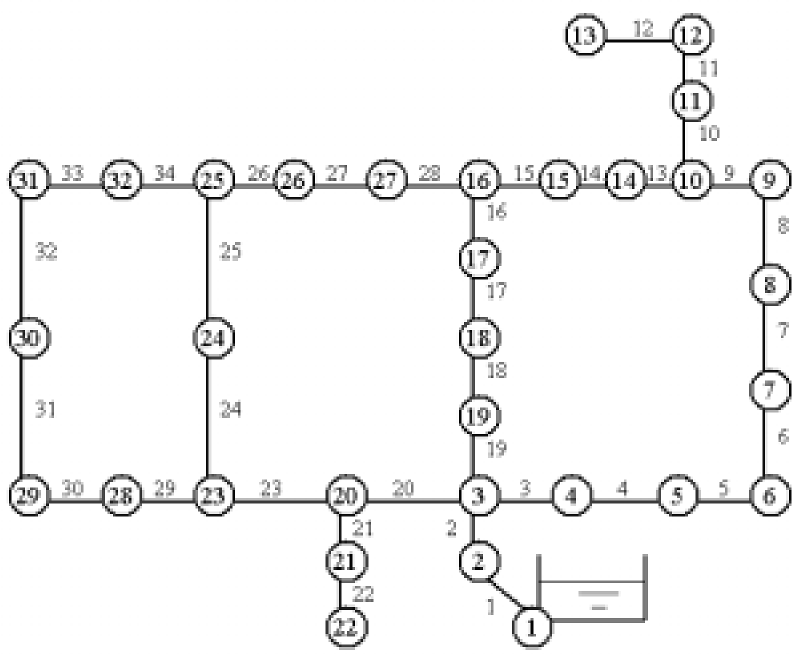


Figure 4. Hanoi Water Distribution Network

Table 3. Commercial Pipe Diameters and Construction Costs

DIAMETER (inch)	DIAMETER (mm)	COST (units)
12	304.8	45.726
16	406.4	70.400
20	508.0	98.378
24	609.6	129.333
30	762.0	180.748
40	1,016.0	278.280

tained from the three methods including NLPG [16], GA [17], and HS. Fujiwara and Khang [16] obtained an optimal cost of $6,320,000. Savic and Walters [17] solved the same problem using genetic algorithm, resulting in an optimal cost of $6,073,000. Harmony Search also solved the problem and could find the least cost ($6,056,000) with the algorithm parameters (size of HM = 50; HMCR = 0.93; and PAR = 0.18) after 162,636 iterations that took about 4.1 hours on a Pentium 333 MHz computer.

4. Conclusions and Discussion

This paper reviewed several traditional optimization techniques and existing heuristic methods, and showed the potential of the development of a new algorithm. The algorithm was devised using the analogy of the music performance process and was named Harmony Search. Advantageous features of Harmony Search that are different from other methods are: HS

Table 4. Comparison of Optimal Diameters and Costs

Pipe	Length (m)	Diameter [16] (inch)	Diameter [17] (inch)	Diameter (HS) (inch)
1	100	40	40	40
2	1,350	40	40	40
3	900	40	40	40
4	1150	40	40	40
5	1450	40	40	40
6	450	40	40	40
7	850	38.16	40	40
8	850	36.74	40	40
9	800	35.33	40	40
10	950	29.13	30	30
11	1200	26.45	24	24
12	3500	23.25	24	24
13	800	19.57	20	20
14	500	15.62	16	16
15	550	12.00	12	12
16	2,730	22.50	12	12
17	1,750	25.24	16	16
18	800	29.01	20	20
19	400	29.28	20	20
20	2,200	38.58	40	40
21	1,500	17.36	20	20
22	500	12.65	12	12
23	2,650	32.59	40	40
24	1,230	22.06	30	30
25	1,300	18.34	30	30
26	850	12.00	20	20
27	300	22.27	12	12
28	750	24.57	12	12
29	1,500	21.29	16	16
30	2,000	19.34	16	12
31	1,600	16.52	12	12
32	150	12.00	12	16
33	860	12.00	16	16
34	950	22.43	20	24
Cost ($)	-	6,320,000	6,073,000	6,056,000

makes a new vector after considering all existing vectors rather than considering only two (parents) as in the genetic algorithm, and HS does not require the setting of initial values of decision variables (for example, initial pipe diameters). These features help HS in increasing flexibility and in finding better solutions.

The three previous applications of the Harmony Search algorithm show that it can solve a continuous-variable problem as well as combinatorial problems, and outperforms other existing heuristic methods in two specific applications. In addition to these applications, Harmony Search has also been applied to other combinatorial or continuous problems (for example, optimal layout of a pipe network, optimal expansion of a pipe network, and optimal parameter calibration of a hydrologic model), in which Harmony Search has outperformed existing mathematical and heuristic methods. Table 5 summarizes the results of various Harmony Search applications.

To demonstrate the faster convergence ability, HS has been applied to the design of a simple network consisting of 7 nodes, 8 links, and 2 loops. The procedures of the computation are the same as those of Problem 3. The full combinations of the different designs are $14^8 = 1.48 \times 10^9$ and the optimal solution is \$419,000. While genetic algorithm [17] and simulated annealing [18] found the solution with up to 250,000 and 70,000 iterations, respectively, HS found the same solution after only 1,095 iterations.

It is expected that this algorithm can be successfully applied to various fields by other researchers. For polishing the new algorithm, Douglas Hofstadter [19] suggests some potential avenues. In his Pulitzer Prize book, Hofstadter suggests a philosophy of science and combines the works of the mathematician Goedel, the artist Escher, and the musician Bach. It is expected that other innovative ideas related to iterative or recursive computation algorithms can be obtained from this analysis, especially from his analysis of fugue (or counterpoint).

5. Acknowledgments

This work was partially supported by grant 971-1205-021-2 from the Basic Research Program of KOSEF. Also, the authors would like to thank many people at Virginia Tech, who have proposed comments and suggestions for this paper: Joel A. Donahue in the Electrical and Computer Engineering Department; Joseph A. Ball and Kyehong Kang in the Mathematics Department; Robert D. Krebs and David F. Kibler in the Civil and Environmental Engineering Department; and John R. Howell and Lisanne E. Lyons in the Music Department.

6. References

[1] Kirkpatrick, S., Gelatt, C., and Vecchi, M. "Optimization by Simulated Annealing." *Science*, Vol. 220, No. 4598, pp 671-680, 1983.

[2] Metropolis et al., "Equations of state calculations by fast computing machines." *The Journal of Chemical Physics*, Vol. 21, pp 1087-1092, 1953.

[3] Glover, F. "Heuristic for Integer Programming Using Surrogate Constraints." *Decision Sciences*, Vol. 8, No. 1, pp 156-166, 1977.

[4] Holland, J. H. *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.

[5] Goldberg, D. E. *Genetic Algorithms in Search Optimization and Machine Learning*, Addison Wesley, 1989.

[6] Schwefel, H. -P. "On the Evolution of Evolutionary Computation." In *Computational Intelligence: Imitating Life*, Zurada, J., Marks, R., and Robinson, C. (Editors), pp 116-124, IEEE Press, 1994.

[7] Fogel, D. B. "A Comparison of Evolutionary Programming and Genetic Algorithms on Selected Constrained Optimization Problems." *SIMULATION*, Vol. 64, No. 6, 399-406, 1995.

Table 5. Results Obtained from Various Harmony Search Applications

EXAMPLE	VAR. #	COMBINATION #	MAX. ITER. #	MIN. SOLUTION	COMMENTS
Problem 1 (TSP)	20	$(20-1)!/2 = 6.08 \times 10^{16}$	5,000	117	Global Optimum
Problem 2 (Function)	2	$3000^2 = 9.0 \times 10^6$	40,000	1.3771	GA=1.4339 EP=1.3772
Problem 3 (Pipe Design)	34	$6^{34} = 2.87 \times 10^{26}$	200,000	\$6.056M	GA=\$6.073M NLPG=\$6.320M
Layout of Pipe Network	112	Possible trees = 1.26×10^{26}	2,000	5062.8	GA=5218.0 EP=5062.8
Expansion of Pipe Network	21	$16^{21} = 1.93 \times 10^{25}$	10,000	\$36.66M	GA=\$37.13M
Parameter Calibration	3	$1000^3 = 1.0 \times 10^9$	5,000	36.78	GA=38.23

[8] Fogel, L. J., Owens, A. J., and Walsh, M. J. *Artificial Intelligence Through Simulated Evolution*, John Wiley, Chichester, UK, 1966.

[9] Koza, J. R. *Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems*, Report No. STAN-CS-90-1314, Stanford University, 1990.

[10] Tirro, F., *Jazz: A History*, W. W. Norton & Company, 1977.

[11] Bracken, J. and McCormick, G. P., *Selected Applications of Nonlinear Programming*, John Wiley & Sons, 1968.

[12] Himmelblau, D. M. *Applied Nonlinear Programming*, McGraw-Hill Book Company, 1972.

[13] Homaifar, A., Qi, C. X. and Lai, S. H. "Constrained Optimization via Genetic Algorithms." *SIMULATION*, Vol. 62, No. 4, pp 242-254, 1994.

[14] Gabriele, G. A. and Ragsdell, K. M. "Large Scale Nonlinear Programming Using the Generalized Reduced Gradient Method." *ASME J. Mech. Des*, Vol .102, 566-573, 1980.

[15] Fujiwara, O. and Khang, D. B. "A Two-Phase Decomposition Method for Optimal Design of Looped Water Distribution Networks." *Water Resources Research*, Vol. 26, No. 4, pp 539-549, 1990.

[16] Fujiwara, O. and Khang, D. B. "Correction to a Two-Phase Decomposition Method for Optimal Design of Looped Water Distribution Networks." *Water Resources Research*, Vol. 27, No. 5, pp 985-986, 1991.

[17] Savic, D. A. and Walters, G. A. "Genetic Algorithms for Least-Cost Design of Water Distribution Networks." *Journal of Water Resources Planning and Management*, ASCE, 123:2, 67-77, 1997.

[18] Cunha, M. da C. and Sousa, J. "Water Distribution Network Design Optimization: Simulated Annealing Approach." *Journal of Water Resources Planning and Management, ASCE*, Vol. 125, No, 4, pp 215-221, 1999.

[19] Hofstadter, D. R. *GOEDEL, ESCHER, BACH: An Eternal Golden Braid*, Vintage Books, New York, 1979.

Dr. Zong Woo Geem received his Bachelor's degree in Civil Engineering from Chungang University in Seoul, Korea, and Master's and Ph.D. degrees in Hydrosystems Engineering from Korea University in Seoul, Korea. He was a visiting scholar at Virginia Tech in 1999, supported by the Korea Science and Engineering Foundation (KOSEF) and is now a faculty research associate for the University of Maryland at College Park. His research interest is the application of optimization techniques to design and maintenance problems in pipeline network systems. He hopes to see further development of the Harmony Search and its applications to various engineering problems. He also writes classical music columns for several Korean magazines.

Dr. G. V. Loganathan is an associate professor in the Department of Civil and Environmental Engineering at Virginia Polytechnic Institute and State University. He received his B.E. degree in Civil Engineering from Madras University, India. He has an M.Tech. from the Indian Institute of Technology, Kanpur and a Ph.D. from Purdue University. His research interest is in the area of hydro-systems engineering with particular emphasis on system design, operation, and replacement.

Dr. Joong Hoon Kim is an associate professor at Korea University in the Department of Civil and Environmental Engineering. He earned the B.S. degree in Civil Engineering from Korea University. He received an M.S.C.E. from Virginia Polytechnic Institute and State University and Ph.D. from the University of Texas at Austin. He was a branch manager in the Engineering Division at the Flood Control District of Maricopa County, Arizona, until he joined the faculty of Korea University in March 1993. His major areas of interest are Application of Mathematical Programming and Computer Applications in Water Resources Engineering.