KSCE Journal of Civil Engineering (2018) 22(4):1506-1514

DOI 10.1007/s12205-017-1864-3

Water Resources and Hydrologic Engineering

pISSN 1226-7988, eISSN 1976-3808
www.springer.com/12205

Development of a Hybrid Harmony Search for Water Distribution System Design

Donghwi Jung*, Doosun Kang, and Joong Hoon Kim*****

Received November 21, 2016/Accepted April 14, 2017/Published Online June 14, 2017

Abstract

Metaheuristic Optimization Algorithms (MOA) have been widely employed for optimal designs of Water Distribution System (WDS). Generally, the size, capacity, and location of WDS components (e.g., pipes, pumps, and tanks) are determined in the WDS design. Therefore, solutions with good fitness have engineeringly sound value of decision variables and share commonality. For example, a solution with transmission pipes bigger than distribution pipes is better with respect to fitness than that with the opposite case in the pipe sizing problem. However, few efforts have been made to consider good fitness rules in the optimization. In this study, we develop a hybrid harmony search (HyHS) algorithm that combines Harmony Search (HS) and a rule induction algorithm. The proposed HyHS algorithm's performance is compared with an improved GA, improved HS, and hybrid GA through optimizations of well-known benchmark functions and two WDS designs. The four algorithms are first applied to the optimization of the De Jong functions. Then, the least-cost and robustness-based design problems of New York tunnel system are solved using the four algorithms. The application results confirm that the proposed HyHS outperforms the other algorithms in terms of computational speed and effectiveness.

Keywords: *Learnable evolution model, Harmony search, Genetic algorithm, Hybridization, Water distribution system*

1. Introduction

Metaheuristic Optimization Algorithms (MOAs) have been widely used in engineering optimization to solve complex and nonlinear problems and also employed for optimal designs of Water Distribution System (WDS). Generally, the size, capacity, and location of WDS components (e.g., pipes, pumps, and tanks) are determined in the WDS design. Therefore, solutions with good fitness have engineeringly sound value of decision variables and share commonality. For example, a solution with transmission pipes bigger than distribution pipes is better with respect to fitness than that with the opposite case in the pipe sizing problem. However, few efforts have been made to consider good fitness rules in the optimization.

The mostly widely used MOA for WDS design is Genetic Algorithm (GA) (Holland, 1975), which utilizes three operators: selection, crossover, and mutation. The possible solutions in the population of GA are improved by the use of these three operators. In selection, GA is more likely to select the solutions with better fitness to the objective of the problem, i.e., "the survival of the fittest". In crossover, the selected solutions share their genetic traits in the chromosome. Finally, mutation provides an opportunity to escape from the local optima and determine a global optimum. Repetition of these three operations enables virtually all the solutions in the population to become the optimal solution for the given problem. However, GA does not directly consider the rules of making good fitness in generating new solutions.

Harmony Search (HS) (Geem *et al*., 2001; Kim *et al.,* 2001) was inspired by the musical ensemble. To obtain an acceptable harmony from musical instruments, the players meet and practice. At first, perfect harmony is not achieved because the rhythm and pitch of each instrument cannot be immediately tuned. However, continued practice to enhance the harmony enables the players to memorize the specific rhythm and pitch of each instrument, which leads to "good harmony". These sets of "good harmony" are memorized and the unacceptable sets are discarded as superior sets are found. The process of updating the sets of harmony continues until the best harmony is obtained. HS implements the harmony enhancement process and the sets of "good harmony" are saved to a solution space termed Harmony Memory (HM), which is a unique feature of HS compared to other MOAs. Although HS can consider decision variable of good solutions, each decision variable independently, a pattern of good memory cannot be considered in generating new solutions.

MOAs such as GA and HS have been widely applied to many engineering problems that cannot be solved by analytical methods or require significant computational efforts; they have provided promising results. Subsequently, the algorithms were improved and

*Member, Research Professor, Research Center for Disaster Prevention Science and Technology, Korea University, Seoul 02841, Korea (E-mil: donghwiku@gmail.com)
**Member, Assistant Professor, Dept. of Civil Engineering, Kyung Hee University, Kyunggi-do 17104, Korea (E-mil: doosunkang@gmail.com)
***Member, Professor, School of Civil, Environmental and Architectural Engineering, Korea University, Seoul 02841, Korea (Corresponding Author, E-mail: jaykim@korea.ac.kr)

hybridized with other MOAs to improve the search performance. A variety of combinations is possible and well-organized hybrid algorithms enhance the opportunity to determine a global optimum.

Recently, a new type of hybrid algorithm has been developed to utilize the learning experiences obtained in the past evaluations of the optimization process. Michalski (2000) developed the Learnable Evolution Model (LEM) that utilizes machine learning for pattern recognition. LEM is the structure of an algorithm that is composed of an MOA and a pattern recognition algorithm. It does not specify the type of algorithms that must be utilized. After the MOA has accumulated the training data (solutions), the pattern recognition algorithm utilizes the training data to extract patterns from the "good solutions" in the optimization. These are then used for updating solutions for the MOA. LEM can be an alternative to overcome the limits of traditional heuristic search algorithms by executing a learning process during optimization.

In some real-world engineering problems, computational time is a more important factor to be considered in optimization than the quality of the solution. For example, near-optimal pump scheduling could be acceptable for real-time pump operations in a Water Distribution System (WDS) (Jamieson *et al*., 2007; Rao and Salomons, 2007; Pasha and Lansey, 2009 & 2010; Jung *et al.,* 2015). More importantly, however, fast convergence near to the optimal solution is required because the optimization must be performed every time interval (e.g., 30 min). Therefore, LEM could be the best alternative for such problems.

LEM has proven to be applicable for the optimization of WDSs. Jourdan *et al.* (2006) developed one type of LEM for multi-objective optimization (LEMMO) that combines non-dominated sorting genetic algorithm-II (NSGA-II) and C4.5 as a rule induction algorithm (Quinlan 1993). The application of LEMMO to the optimal designs of the New York tunnel and Hanoi network revealed that LEMMO significantly improved the computation speed and quality of the solutions.

In this study, we develop a hybrid harmony search (HyHS) algorithm that combines HS and C4.5, a pattern recognition algorithm, in a LEM framework. HS executes a predefined number of function evaluations and the "reasons" of "good solutions" are extracted from the solutions stored in HM by C4.5, which is used for generating new populations. The proposed HyHS algorithm's performance is compared with improved GA (IGA), improved HS (IHS), and hybrid GA (HyGA) through optimizations of well-known benchmark functions and two WDS design problems. This study focuses on investigating the impact of employing machine learning on the existing MOAs' performance and identifying the most effective hybridization scheme. The four algorithms are first applied to the optimization of the De Jong (1975) functions. Then, two water distribution network design problems are solved using the four algorithms.

2. Methodology

2.1 Harmony Search

HS contains a solution storage function called HM that necessitates the definition of two parameters: HM considering rate (HMCR) and pitch-adjusting rate (PAR). The following section describes the function of the HS operators.

HM: HM is a solution space that stores "good solutions" obtained during the optimization process. The solutions stored in HM are updated over generations as HS discovers a new solution that is superior to a previous solution in HM. HS can generate new solutions for the next generation by either selecting from HM or by random generation.

- HMCR: HMCR is a user-defined parameter that determines whether new individuals are selected from HM or randomly generated. For each generation, a random number is generated and if the random number is less than the defined HMCR, all new individuals are selected from the HM; otherwise, new solutions are randomly generated.
- PAR: PAR defines the frequency of pitch adjusting. It is a similar process to the mutation in GA, except that the random change of a specific decision variable only has neighborhood decision value. For example, if a pipe diameter of 300 mm is a decision variable and the commercial pipe sizes are 100 mm, 200 mm, 300 mm, 400 mm, and 500 mm, pitch adjusting adjusts 300 mm to either 200 mm or 400 mm.
- Similar to the manner where GA adopts the three operators

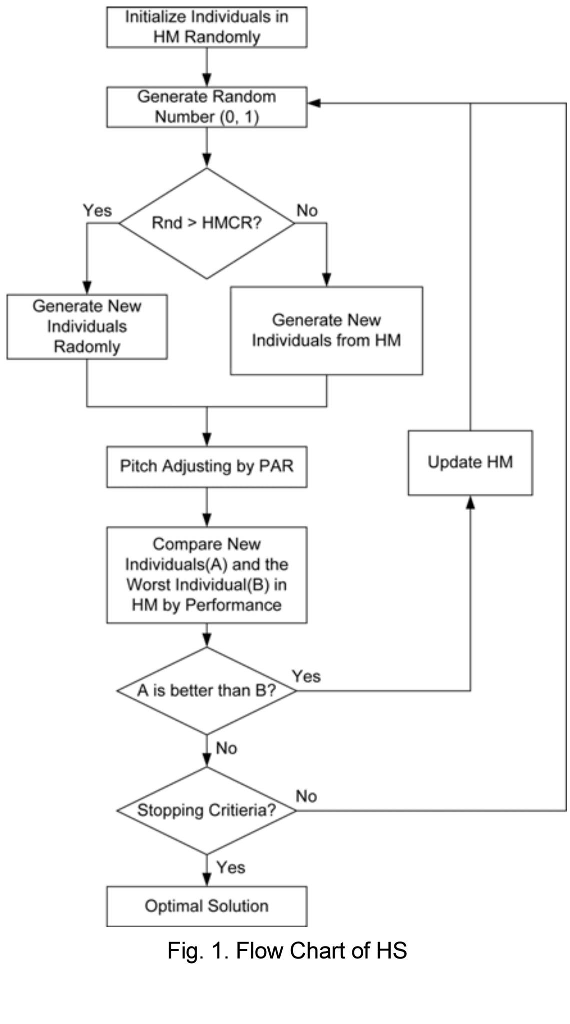

Fig. 1. Flow Chart of HS

of "selection", "crossover", and "mutation", HS converges to optimal solutions by the process of "HM updating" and "pitch adjusting" as illustrated in Fig. 1.

HS has been improved and modified in previous studies (Paik *et al*., 2005; Baek *et al*., 2005; Madhavi *et al*., 2007; Geem and Sim, 2010). Paik *et al.* (2005) developed Modified Harmony Search (MHS) and used three different strategies to perform HS: MHS1, MHS2, and MHS3. For example, MHS1 selects the solutions in HM with equal probability without considering each solution's fitness when generating a new solution. MHS2 discards overlapping harmony in HM when the solutions converge to the optimal solution and substitutes overlapping solutions to the new solutions. Paik *et al.* (2005) defined this as "elimination of overlapping harmony" for the process. MHS3 applies pitch adjusting to every decision variable in the best solution when the difference of fitness between the best solution and the worst solution is less than a predefined value. Baek *et al.* (2005) developed revised HS that can change HMCR and PAR as the number of function evaluations increase and can update multiple solutions in HM if the newly generated solutions are superior to more than one previous solution in HM. Mahdavi *et al*. (2007) proposed an improved version of HS that considers dynamic changes of PAR and its bandwidth. Geem and Sim (2010) developed the parameter-setting-free HS algorithm, where HMCR and PAR values vary as the iteration number increases.

Recently, various hybrid HS algorithms have been proposed. Fesanghary *et al.* (2008) proposed the sequential quadratic programming-based HS to strengthen HS's local search ability. The application of the hybrid HS to benchmark problems demonstrated that both the solution quality and computational speed were improved. Karahan *et al*. (2013) employed a quasi-Newton algorithm, the Broyden, Fletcher, Goldfarb, and Shannon (BFGS) method, as a local search algorithm for a hybrid HS for the parameter estimation of the Muskingum routing model. Wang *et al.* (2011) developed a hybrid HS that combines HS and a heuristic local search algorithm for solving the blocking flow shop scheduling problems of many constraints. The proposed algorithm outperformed more than 12 different algorithms. Most previous hybridizations focused on improving the local search ability of HS.

2.2 Machine Learning and C4.5

Humans can accumulate external knowledge and information and utilize their knowledge and understanding to create useful knowledge, rather than only memorizing it, via the process of "learning". Machine learning is based on the implementation of the human "learning" process into algorithms. It searches solutions using the knowledge and information gained from experience or granted by humans. To embody machine learning, many methods such as neural network, data mining, decision tree, pattern recognition, and reinforcement learning are utilized.

C4.5 (Quinlan, 1993) is a type of decision tree framework to extract patterns in the data, thus called a pattern recognition algorithm. C4.5 was developed as an advanced decision tree and in contrast to branch-and-bound (Land and Doig, 1960), can be applied to continuous variables (as summarized in Table 1) and implement information gain theory (Shannon, 1948) to build a decision tree. The detail of C4.5 is described in Quinlan (1993) with a sample example. The following section describes the approach used to combine HS and C4.5 to develop the HyHS algorithm.

Table 1. Comparison of Branch and Bound (Land and Doig, 1960) and C4.5 (Quinlan, 1993)

	Branch and Bound (Land and Doig, 1960)	C4.5 (Quinlan, 1993)
Common Feature	Decision tree	
	Search and pruning	
Difference	Search almost all the branch	Calculate entropy and decide branch first
	1,0	Use continuous value

3. Hybrid Harmony Search

This study developed HyHS by combining HS as a main search algorithm and C4.5 as a pattern recognition algorithm to include the information stored in HM to the HS searching procedure. The proposed HyHS is expected to lessen the number of function evaluations and improve the search efficiency. The proposed HyHS is the same as HS as regards the application parameters such as HMCR and PAR; however, it embeds C4.5 within the framework as illustrated in Fig. 2.

HyHS begins optimization in a similar fashion to normal HS. After 100 iterations, the solutions stored in HM are used as training data for the C4.5 pattern recognition, which then induces rules or patterns of "good solutions" for new population generation. HM is then filled with the reproduced solutions that follow the rules determined by C4.5. To accumulate sufficient training data for C4.5, the capacity of HM in HyHS is larger than that of HS. In this study, HM in HyHS stores 100 solutions and the top 10% of the solutions are considered "good solutions". C4.5 induces rules based on the "good solutions" obtained after each of the 100 HS iterations and the existing solutions are replaced with new solutions following the rules found by C4.5. This procedure is repeated until the "stopping criteria" are satisfied (Fig. 2). To address the problem of over-fitting, C4.5 in HyHS terminates expanding nodes if the information gain of the node is less than 0.15 and prunes the decision tree.

The crucial element of HyHS is the characteristic of the guided search. For enhancing the search ability, the controlling level of guidance and lesson consideration by C4.5 are the most important factors. If C4.5 over-guides the new solution, it causes local optima and if it minimizes lesson consideration, there is no solution improvement over HS. Further study is required to determine the best combination of parameters and structure in HyHS.

4. Case Study

The proposed HyHS is applied to solving the De Jong (1975)

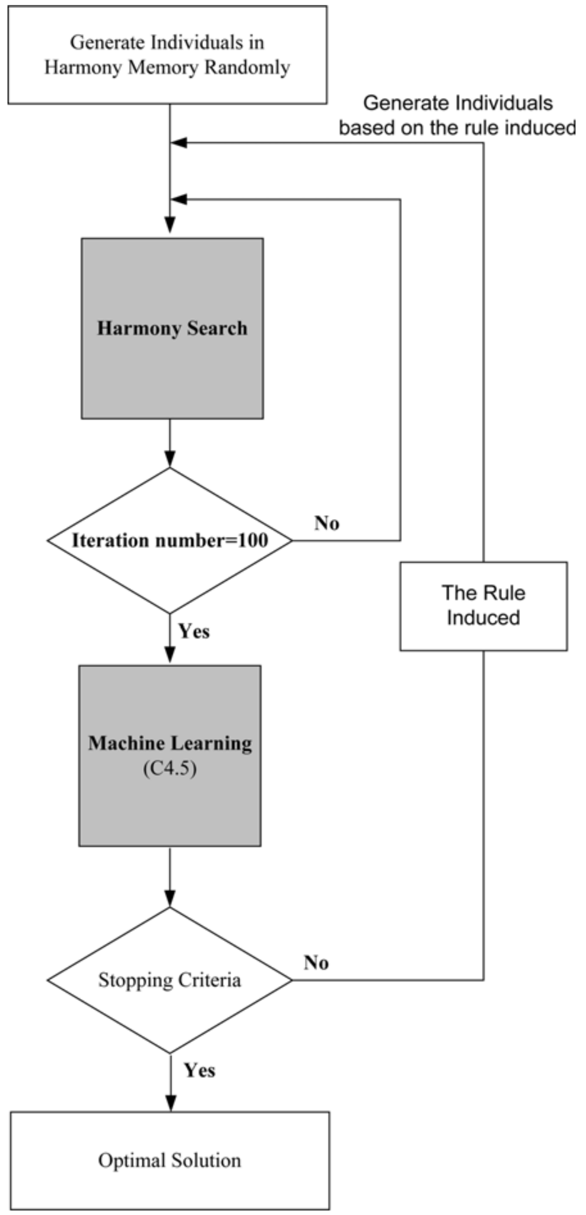

Fig. 2. Flow Chart and Structure of HyHS

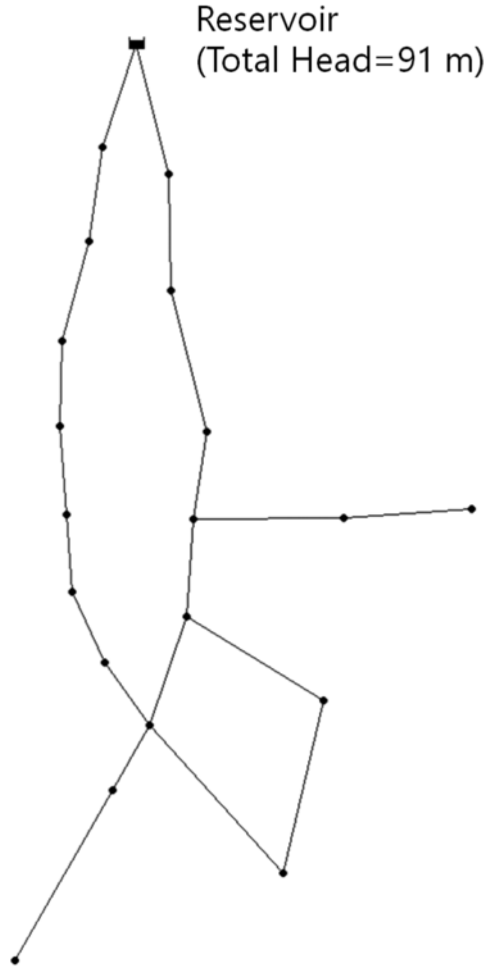

Fig. 3. A Layout of New York Tunnel System

functions and the optimal design of a WDS. For comparison, three other optimization algorithms, IGA, IHS, and HyGA, are also applied to the same problems. For the improved versions of HS, a dynamic pitch-adjusting rate is considered as suggested by Mahdavi *et al.* (2007). Similarly, the improved GA adopts a dynamic mutation rate that increases over iterations. The hybrid GA has a similar structure as the hybrid HS; however, GA replaces HS as the MOA in the hybrid framework presented in Fig. 2.

De Jong (1975) suggested five test functions to examine the performance of a GA. In this paper, four are used and summarized in Tables 2 and 3. Further, the algorithms are applied to the optimal design of a New York tunnel system (Fig. 3). Both single- and multi-objective design problems are solved to compare the performance of the selected algorithms. The single objective problem is a least-cost design of the system. A network's robustness is posed as the objective for the multi-objective design problem. The study network of the New York tunnel system has 19 nodes, one reservoir, and 21 pipes. More details on the system should refer to Quindry *et al.* (1981). Water distribution network design is formulated as an optimization problem with a selection of pipe sizes as the decision variables.

Note that the computation time required for the number of iterations presented in the result tables was not included because all the algorithms had a similar time of execution for the same number of iterations. The Visual Basic 6.0 programs were coded in a fashion such that factors other than the algorithm difference (i.e., HS with C4.5 versus GA with C4.5) could not influence the computation time. The programs were executed on a computer with a quad-core processor with a 2.2 GHz CPU and 8 G memory.

5. Application Results

5.1 De Jong (1975) Functions

The De Jong (1975) Function 1 is a continuous function with three variables. It is a convex, unimodal, and low-dimensional quadratic function. The known optimal value of the function is zero (Table 2). The average evolution of the objective function

Table 2. De Jong (1975) Functions and Their Known Optimal Values

Function Number	Function	Known Optimal Value
1	$f_1(x) = \sum_{i=1}^{3} x_i^2$	$MIN(F1) = F(0,0,0) = 0$
2	$f_2(x) = 100 \times (x_1^2 - x_2)^2 + (1 - x_1)^2$	$MIN(F2) = F(1,1) = 0$
3	$f_3(x) = \sum_{i=1}^{5} [x_i]$	$MIN(F3) = F(-5.12, -5.12, 5.12, 5.12, -5.12) = -30$
4	$f_4(x) = \sum_{i=1}^{30} (ix_i^4) + Gauss(0,1)$	$MIN(F4) = F(0,0,\ldots,0) = 0$

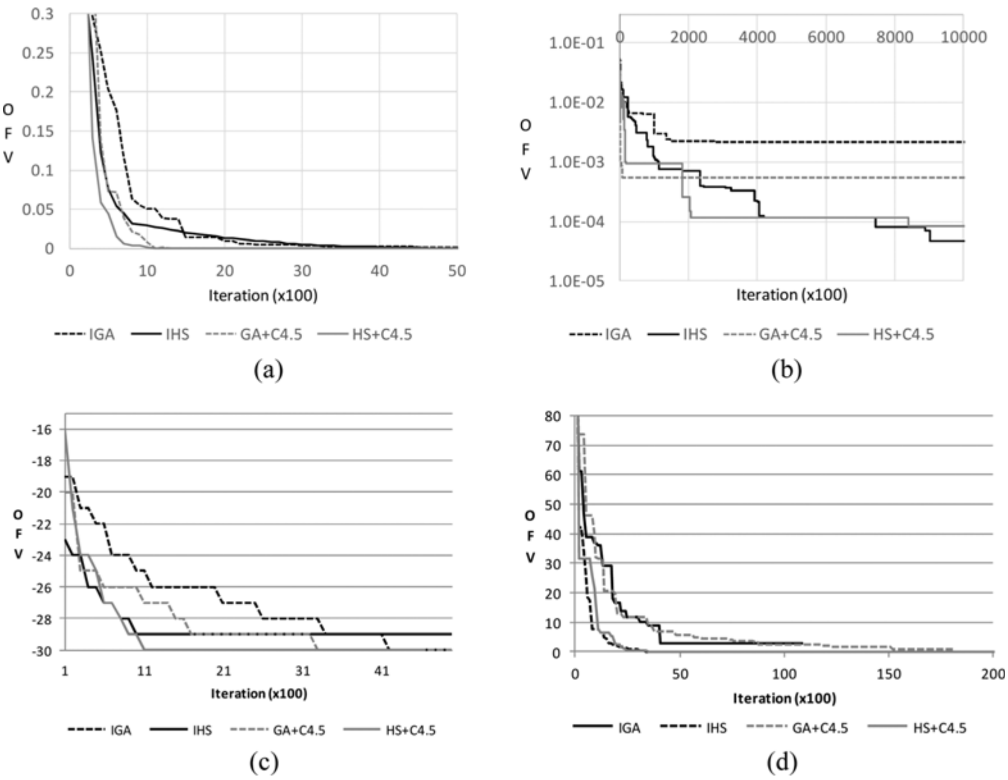

Fig. 4. Average Evolution of Objective Function Values of the De Jong Functions using four Algorithms: (a) Function 1, (b) Function 2, (c) Function 3, (d) Function 4

values of each of the four optimization algorithms is indicated in Fig. 4(a) and Table 4. Note that the initial population is identical for all the algorithms and the mean objective function value was calculated from 10 runs. The two hybrid algorithms (HyGA and HyHS) determined the optimal value at 2,000 iterations (Table 4), whereas IGA and IHS failed to find the optimum. Hence, it is observed that the hybrid framework incorporating the learning process of the C4.5 enhanced the optimization performance compared to the regular search approaches.

Table 3. Range and size of the De Jong (1975) Functions

Function Number	Limits	Resolution Factor	Problem Size
1	$-5.12 \le x_i \le 5.12$	$\Delta x = 0.01$	$(1024)^3 \cong 10^9$
2	$-2.048 \le x_i \le 2.048$	$\Delta x = 0.001$	$(4096)^2 \cong 1.7 \times 10^6$
3	$-5.12 \le x_i \le 5.12$	$\Delta x = 0.01$	$(1024)^4 \cong 10^{15}$
4	$-1.28 \le x_i \le 1.28$	$\Delta x = 0.01$	$(256)^{30} \cong 10^{72}$

Function 2 is the Rosenbrock (1960) function with two variables. The Rosenbrock function is continuous, but is nonconvex and well-known for its difficulty obtaining an optimal solution because it has a depressed parabolic valley along the curve

Table 4. Average Change of Objective Function Value of four Algorithms (Function 1)

Iterations	IGA	IHS	GA+C4.5	HS+C4.5
500	2.04E-01	7.63E-02	7.26E-02	4.49E-02
1000	5.05E-02	2.94E-02	8.02E-03	1.22E-03
1500	1.42E-02	2.07E-02	6.00E-05	2.00E-05
2000	9.50E-03	1.32E-02	0.00E+00	0.00E+00
2500	4.76E-03	9.28E-03	0.00E+00	0.00E+00

Table 5. Average Change of Objective Function Value of Four Algorithms (Function 2)

Iterations	IGA	IHS	GA+C4.5	HS+C4.5
500	5.55E-02	5.07E-02	6.28E-02	3.44E-02
1000	4.92E-02	3.97E-02	1.84E-02	5.62E-03
1500	4.49E-02	2.52E-02	7.77E-03	8.02E-03
2000	3.56E-02	2.52E-02	7.50E-03	1.53E-02
400000	2.17E-03	2.23E-04	5.52E-04	1.17E-04

$x_2 = x_1^2$ in the solution space. The average evolution of the objective function values of the algorithms are seen in Fig. 4(b) and Table 5. None of the algorithms could achieve the optimal value of zero while the final solution of IHS is superior to the other three algorithms. It is observed in Fig. 4(b) that HyHS demonstrated dramatic reductions in the objective function value in the early stage of the iterations (up to iteration 400,000). This was likely because of the lesson learning process of C4.5.

Function 3 is a step function where the solution space is discontinuous. The optimal function value is -30 when all five variables have an optimal value of -5.12. It is observed that HyHS provided improved performance compared to other algorithms while IHS failed to find the optimum (Fig. 4(c)). The objective function value of HyHS converged to the optimal value in approximately 1,100 iterations, whereas the GA-based algorithms required significantly more iterations to approach the optimum.

Function 4 is a continuous, convex, and high-dimensional quadratic function with 30 variables and randomly peaked solution space containing a white noise term. Gaussian noise adds difficulty to determining the true optimal value of the problem. The size of the solution space is the largest of the four test functions and has 10^{72} possible solutions. HS-based algorithms deliver improved performance compared to GA-based algorithms (Fig. 4(d) and Table 6). An apparent difference is observed between the GA-based algorithms and the HS-based algorithms in the slopes of the function evolutions; HS-based algorithms converge to the solution in approximately 4,000 iterations, whereas the GA-based algorithms require in excess of 18,000 iterations to converge. Further, IHS and HyHS achieve the known optimal solution at 10,000 iterations, whereas the IGA and HyGA fail to determine the optimum (Table 6).

Table 6. Average Change of Objective Function Value of four Algorithms (Function 4)

Iterations	IGA	IHS	GA+C4.5	HS+C4.5
100	85.45	82.39	73.73	95.48
500	38.89	24.76	46.21	31.38
1000	36.89	7.55	32.11	16.67
2500	11.65	1.14	11.78	0.53
10000	2.61	0.00	2.27	0.00

In summary, the HyHS provided the best performance of the four tested optimization algorithms for application to the De Jong (1975) functions. It succeeded in determining the optimal solutions for three functions. All of the algorithms failed to determine the optimum for Function 2. Moreover, HyHS converged to the optimum with significantly less iterations than the other algorithms indicating better effectiveness and efficiency.

5.2 Least Cost Design of Water Distribution Pipe Network

In this section, the proposed HyHS was applied to the optimal design of a water distribution pipe network. The problem is formulated as a least-cost design of a pipe network. The three other algorithms are also applied to the problem for comparison.

To examine the performance of the algorithms, the traditional least-cost design problem of the New York tunnel system with a simple material-based cost function was solved under a constraint of minimum pressure requirement. The objective function of the pipe construction cost (Quindry *et al.*, 1981) used in this study is presented below including the requirement of satisfying the minimum pressure requirement of 17.6 m of water (25 psi) at all consumer nodes.

$$\text{Minimize } F1 = \sum_{i}^{n}(1.1 \times D_i^{1.24} \times L_i) \tag{1}$$

$$\text{s.t. } H_j \geq 17.6 \quad j = 1, \ldots, m \tag{2}$$

where D_i is the diameter of pipe i (i = 1,..., n), n is the number of pipes in the network, L_i is the length of pipe i, H_j is the pressure head (m) of the consumer node j (j = 1, ..., m), and m is the number of consumer nodes in the network. The available commercial pipe sizes for the New York tunnel system are 1.524 m, 1.829 m, 2.591 m, 3.353 m, 4.572 m, and 5.1816 m.

As summarized in Table 7, all the algorithms determined the same optimal solution. However, the number of iterations for each algorithm to obtain the optimal solution differed. The

Table 7. Optimal Cost and Iterations Required to Reach the Optimum for Least-cost Design of the New York Tunnel System using four Algorithms

	IGA	IHS	GA+C4.5	HS+C4.5
Optimal Value (Before optimization: 1.798 × 10^8 USD)	1.134 × 10^8	1.134 × 10^8	1.134 × 10^8	1.134 × 10^8
Iterations to the Optimal Value (× 100)	3957	663	99	109

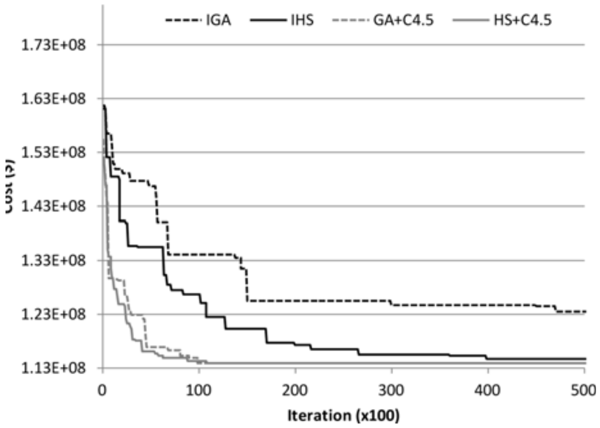

Fig. 5. Average Evolution of Solution Costs for Least-cost Design of the New York Tunnel System using four Algorithms

HyGA and HyHS achieved the optimum with 9,900 and 10,900 iterations, respectively, which is significantly less than the iterations required for IGA and IHS. The superior efficiency of the hybrid algorithms is also illustrated in Fig. 5 where the hybrid algorithms indicate dramatic reductions in their objective function values in the early iterations. Comparing IGA and IHS, the GA required considerably more iterations (395,700) than HS that required 66,300 iterations. Hence, the hybrid algorithms indicated improved optimization performance for a real-world problem compared to the regular algorithms. It can be concluded that the pattern recognition process of C4.5 in hybrid algorithms enhanced the searching ability by creating new solutions, preserving the patterns of the previous "good solutions", and appropriately altering the chromosomes to achieve superior solutions as the iterations proceeded.

5.3 Multi-objective Design of Water Distribution Pipe Network

The proposed HyHS was also applied to a multi-objective optimal design of the New York tunnel system to evaluate the performance in a multi-objective optimization framework. In addition to the system total cost considered in the least-cost design, a second objective, to maximize system robustness was posed (Fig. 6). System robustness is defined as the persistence of the network performance (supplying water in adequate pressure) against the variations of water demand and uncertainties in pipe roughness values. A system robustness index (α_c) is quantified as the inverse coefficient of variation of nodal pressure at the critical node and the second objective is defined as:

$$\text{Maximize } F2 = \alpha_c = \frac{P_c}{\sigma_c} \tag{3}$$

where P_c and σ_c are the mean and the standard deviation of the stochastic nodal pressures at the critical node, respectively.

The system cost ($F1$) and robustness ($F2$) have a trade-off relationship that results in Pareto optimal solutions (Fig. 6). Minimizing cost will sacrifice the system robustness, whereas enhancing robustness will inevitably increase system cost.

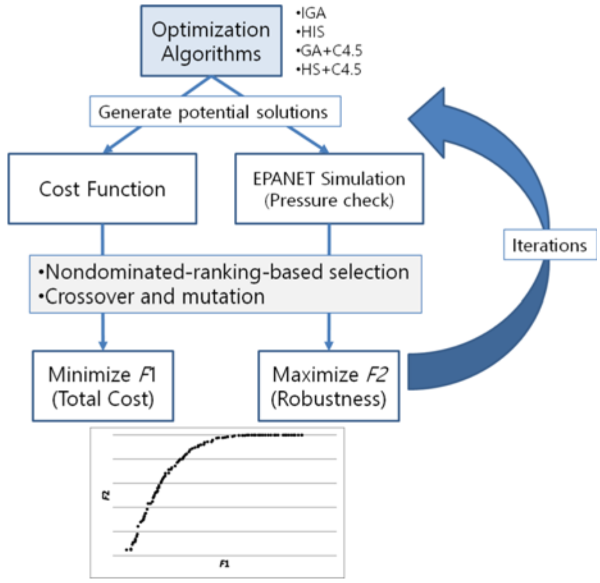

Fig. 6. Multiobjective Optimal WDS Design Problems for Testing the four Algorithms

The multi-objective hybrid algorithms updated their population or HM every 300 iterations. This is because a multi-objective optimization is more complex and difficult to converge than a single objective problem and thus necessitates more iterations to obtain effective training data for C4.5. Further, the selection of a set of "good solutions" in the process of C4.5 is based on the rank of the solutions, whereas single hybrid algorithms select a "good solution" by the single fitness value of the solutions. In this example, the ranking approach developed by Fonseca and Fleming (1993) was used to rank the solutions.

Comparing the performance of the four algorithms is not easy in the multi-objective optimization because each solution has two objective function values. In previous case studies, the objective function values of all the algorithms were compared at the discrete evaluation number of algorithms. However, in multi-objective optimization, comparing only one objective function value is insufficient to confirm the performance of the algorithms because the superiority of fitness in one objective function of an algorithm does not guarantee the superiority of fitness in the other objective function. Moreover, comparing the temporal non-dominated rank of the solutions at the discrete evaluation number of algorithms cannot be used as an alternative to compare the algorithms' performance. Therefore, this study summed the objective function values of all the solutions and compared the values at the discrete evaluation number of algorithms as indicated in Fig. 7. Because the main purpose of the multi-objective design was to minimize the cost and maximize the minimum nodal robustness index, the algorithm with the least and maximum summation values of the cost and the robustness index, respectively, is more likely to deliver superior performance compared to the other algorithms.

In the results presented in Table 8, the best performing algorithms were the Multi-objective (MO) HS and MO HyHS.

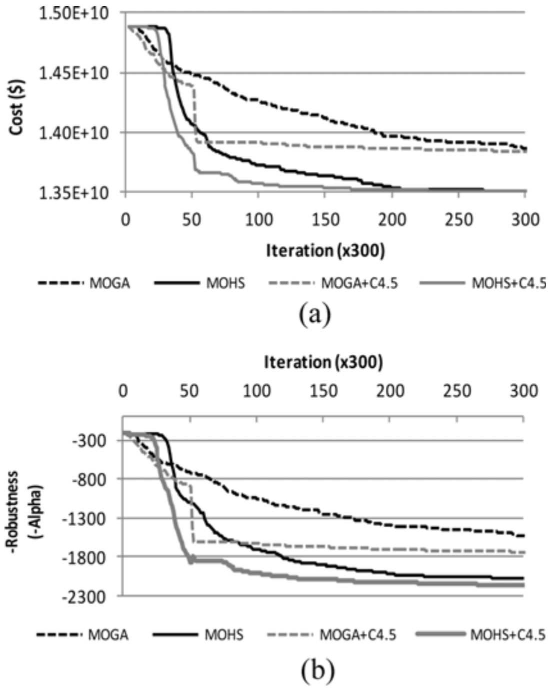

Fig. 7. Evolution of Objective Function Values: (a) Sum of Cost, (b) Sum of System Robustness for Two-objective Design of the New York Tunnel System using Four Algorithms

Before 30,000 evaluations, the algorithms had reached cost and robustness values corresponding to those of 60,000 evaluations by the MOGA and MO HyGA. As in the last experiment, the hybrid-based algorithms, HyGA and HyHS, confirmed the value of the learning process using C4.5 based on the dramatically increased fitness values around the evaluation numbers of 15,000 as presented in Fig. 7. This indicates that around 15,000 evaluations of the algorithms, the apparent differences between "good solution" and "bad solution" lead to the transition of the "bad solution" to the "good solution" by updating the solutions based on the rules from C4.5, which significantly improved the two fitness values.

However, the influence of hybridization on the solution quality was not as consistent in the multi-objective optimization as in the least cost design. In the HS-based algorithm, HyHS was more efficient than simple HS. However, HyGA indicated inferior efficiency compared to MOHS (Table 8). The superiority of HS in the multi-objective scheme has never been investigated. However, this study concluded that HM could be a factor in the multi-objective optimization by storing and keeping the various non-dominated solutions. Note that HM is the only and apparent difference with GA. Instead of hesitating by choosing the solution with the lowest rank when generating the solutions for the next generation, MOHS typically uses higher-ranking solutions from the HM to generate the new solutions. This contributes to the superior performance of the algorithm over HyGA, especially in the multi-objective design of a WDS.

6. Conclusions

This study developed a HyHS that combined an HS algorithm and a pattern recognition algorithm called C4.5. Although the structure of the hybrid algorithm can be modified to improve the performance depending on the specific problem, the rules of the extraction process using C4.5 and the update of the solutions in HM are the key concepts of HyHS. The performance of HyHS was investigated based on the algorithm's efficiency and effectiveness compared to GA, HS (the improved versions), and HyGA for the optimization of the De Jong (1975) functions, a least cost design, and a multi-objective design of the New York tunnel system.

HyHS was more efficient and effective than the pre-existing algorithms and hybridization of HS with C4.5 resulted in superior algorithm performance compared to other three algorithms. Moreover, HyHS was the most robust algorithm for the optimization problems investigated in this study.

This study has several limitations that future research must address. A future study is required to investigate the applicability of HyHS to a larger and more complex network than the New York tunnel system. For such a substantially larger network problem, the rule induction of C4.5 may require excessive time and thus weaken the advantage gained using HyHS. Comparison between overhead time increase by the network size and computation time reduction by HyHS should be conducted. In addition, HyHS can be compared to the rule induction algorithm with other algorithms than GA (e.g., Tabu search) in various optimization problems of different characteristics and level of complexity in the context of water engineering (e.g., optimal groundwater management problem in Moutsopoulos *et al.* (2017)).

An efficient and robust meta-heuristic algorithm should have both exploration and exploitation ability. Although indicating a strong exploration at the early search stages, a hybrid HS

Table 8. Change of Sum of Cost and Robustness by Algorithm (The robustness-based design of the New York tunnel system)

Iterations (× 300)	MOGA		MOHS		MOGA+C4.5		MOHS+C4.5	
	Cost (USD × 10^{10})	Robustness (α)	Cost (USD × 10^{10})	Robustness (α)	Cost (USD × 10^{10})	Robustness (α)	Cost (USD × 10^{10})	Robustness (α)
5	1.488	220.2	1.488	226.1	1.487	237.2	1.488	220.3
10	1.488	244.3	1.488	226.1	1.482	331.1	1.488	224.6
100	1.427	1049	1.373	1700	1.391	1635	1.357	2007
200	1.397	1391	1.354	2016	1.386	1704	1.352	2127
300	1.387	1536	1.350	2076	1.384	1738	1.350	2175

equipped with a local search algorithm (e.g., greedy algorithm) as a third sub algorithm could further improve the exploitation.

Acknowledgements

This work was supported by a grant from The National Research Foundation (NRF) of Korea, funded by the Korean government (MSIP) (No. 2016R1A2A1A05005306).

References

Baek, C. W., Kim, E. S., Park, M. J., and Kim, J. H. (2005). "Development of optimal decision-making system for rehabilitation of water distribution systems using ReHS." *Journal of Korea Water Resources Association*, Vol. 38, No. 3, pp. 199-212, DOI: 10.3741/JKWRA.2005.38.3.199.

De Jong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems*, PhD Thesis, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor.

Fesanghary, M., Mahdavi, M., Minary-Jolandan, M., and Alizadeh, Y. (2008). "Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems." *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, pp. 3080-3091, DOI: 10.1016/j.cma.2008.02.006.

Fonseca, C. M. and Fleming, P. J. (1993). "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization." *Genetic Algorithms: Proceedings of the Fifth International Conference* (S. Foreest, ed.), San Mateo, CA.

Geem, Z. W., Kim, J. H., and Loganathan, G. V. (2001). "A new heuristic optimization algorithm: Harmony search." *Simulation*, Vol. 76, No. 2, pp. 60-68, DOI: 10.1177/003754970107600201.

Geem, Z. W. and Sim, K. B. (2010). "Parameter-setting-free harmony search algorithm." *Applied Mathematics and Computation*, Vol. 217, No. 8, pp. 3881-3889, DOI: 10.1016/j.amc.2010.09.049.

Holland, J. (1975). *Adaptation in natural and artificial systems,* Ann Arbor, The University of Michigan Press.

Jamieson, D. Shamir, U, Martinez, F., and Franchini, M. (2007). "Conceptual design of a generic, real-time, near-optimal control system for water distribution networks." *Journal of Hydroinformatics*, Vol. 9, No. 1, pp. 3-14, DOI: 10.2166/hydro.2006.013.

Jourdan, L., Corne, D. W., Savic, D. A., and Walters, G. A. (2006). "LEMMO: Hybridising rule induction and NSGA II for multiobjective water systems design." *Proc., 8th Int. Conf. on Computing and Control for the Water Industry*, Vol. 2, (D.A. Savic, G.A. Walters, R. King, and S.-T. Khu, eds.), Exeter Press, Exeter, U.K., pp. 45–50.

Jung, D., Kang, D., Kang, M., and Kim, B. (2015). "Real-time pump scheduling for water transmission systems: Case study." *KSCE Journal of Civil Engineering*, Vol. 19, No. 7, pp. 1987-1993, DOI: 10.1007/s12205-014-0195-x.

Karahan, H., Gurarslan, G., and Geem, Z. W. (2013). "Parameter estimation of the nonlinear Muskingum flood-routing model using a hybrid harmony search algorithm." *Journal of Hydrologic Engineering*, Vol. 18, No. 3, pp. 352-360, DOI: 10.1061/(ASCE)HE.1943-5584.0000608.

Kim, J. H., Geem, Z. W., and Kim, E. S. (2001). "Parameter estimation of the nonlinear Muskingum model using harmony search." *Journal of AWRA*, Vol. 37, No. 5, pp. 1131-1138, DOI: 10.1111/j.1752-1688.2001.tb03627.x.

Land, A. H. and Doig, A. (1960). "An automatic method of solving discrete programming problems." *Econometrica*, Vol. 28, pp. 497-520, DOI: 10.2307/1910129.

Mahdavi, M, Fesanghary, M., and Damangir, E. (2007). "An improved harmony search algorithm for solving optimization problems." *Applied Mathematics and Computation*, Vol. 188, pp. 1567-1579, DOI: 10.1016/j.amc.2006.11.033.

Michalski, R. (2000). "Learnable evolution model: Evolutionary processes guided by machine learning." *Machine Learning*, Vol. 38, Nos. 1-2, pp. 9-40, DOI: 10.1023/A:1007677805582.

Moutsopoulos, K. N., Papaspyros, J. N. E., and Tsihrintzis, V. A. (2017) "Management of groundwater resources using surface pumps: Optimization using Genetic Algorithms and the Tabu Search method." *KSCE Journal of Civil Engineering,* DOI: 10.1007/s12205-017-1013-z. In Press.

Paik, K., Kim, J. H., Kim, H. S., and Lee, D. R. (2005). "A conceptual rainfall-runoff model considering seasonal variation." *Hydrological Processes*, Vol. 19, No. 19, pp. 3837-3850, DOI: 10.1002/hyp.5984.

Pasha, M. and Lansey, K. (2009). "Optimal pump scheduling by linear programming." *Proc. of World Environmental and Water Resources Congress 2009*, ASCE, Kansas City, USA, pp. 395-404, DOI: 10.1061/41036(342)38.

Pasha, M. and Lansey, K. (2010). "Strategies for real time pump operation for water distribution systems." *Proc. of Water Distribution System Analysis 2010*, Tucson, USA, pp. 1456-1469, DOI: 10.1061/41203(425)130.

Quindry, G., Brill, E. D., and Liebman, J. C. (1981). "Optimisation of looped water distribution systems." *Journal of Environmental Engineering Division, ASCE*, Vol. 107, pp. 665-679.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*, Morgan Kaufmann Publishers, Inc.

Rao, Z. and Salomons, E. (2007). "Development of a real-time, near-optimal control process for water-distribution networks." *Journal of Hydroinformatics*, Vol. 9, No. 1, pp. 51-64, DOI: 10.2166/hydro.2006.015.

Rosenbrock, H. H. (1960). "An automatic method for finding the greatest or least value of a function." *The Computer Journal*, Vol. 3, No. 175, DOI: 10.1093/comjnl/3.3.175.

Shannon, C. E. (1948). "A Mathematical theory of communication." *Bell System Technical Journal*, Vol. 27, No. 3, pp. 379-423, DOI: 10.1145/584091.584093.

Wang, L., Pan, Q., and Fatih Tasgetiren, M. (2011). "A hybrid harmony search algorithm for the blocking permutation flow shop scheduling problem." *Computers & Industrial Engineering*, Vol. 61, 76-83, DOI: 10.1016/j.cie.2011.02.013.