Mine Blast Harmony Search and Its Applications

Ali Sadollah, Ho Min Lee, Do Guen Yoo and Joong Hoon Kim

Abstract A hybrid optimization method that combines the power of the harmony search (HS) algorithm with the mine blast algorithm (MBA) is presented in this study. The resulting mine blast harmony search (MBHS) utilizes the MBA for exploration and the HS for exploitation. The HS is inspired by the improvisation process of musicians, while the MBA is derived based on explosion of landmines. The HS used in the proposed hybrid method is an improved version, introducing a new concept for the harmony memory (HM) (i.e., dynamic HM), while the MBA is modified in terms of its mathematical formulation. Several benchmarks with many design variables are used to validate the MBHS, and the optimization results are compared with other algorithms. The obtained optimization results show that the proposed hybrid algorithm provides better exploitation ability (particularly in final iterations) and enjoys fast convergence to the optimum solution.

Keywords Harmony search · Mine blast algorithm · Hybrid metaheuristic methods · Global optimization · Large-scale problems

1 Introduction

Among optimization methods, metaheuristic algorithms have shown their potential for detecting near-optimal solutions when exact methods may fail, especially

---

A. Sadollah · D.G. Yoo
Research Center for Disaster Prevention Science and Technology,
Korea University, Seoul 136-713, South Korea
e-mail: ali_sadollah@yahoo.com, godqhr425@korea.ac.kr

H.M. Lee · J.H. Kim(✉)
School of Civil, Environmental, and Architectural Engineering,
Korea University, Seoul 136-713, South Korea
e-mail: {dlgh86,jaykim}@korea.ac.kr

J.H. Kim and Z.W. Geem (eds.), *Harmony Search Algorithm*,
Advances in Intelligent Systems and Computing 382,
DOI: 10.1007/978-3-662-47926-1_16

when the global minimum is surrounded by many local minima. Hence, the need to use such approaches is understood by the optimization community.

Harmony search (HS) algorithm, developed by Geem et al. [1-3], is derived from the concepts of musical improvisations and harmony knowledge, and is a well-known metaheuristic algorithm. To date, the HS has proved its advantages over other optimization methods [4-6], and many improved versions have been developed in the literature [7-10].

In recent years, it has become clear that concentrating on a sole optimization method may be rather restrictive. A skilled combination of concepts from different optimizers can provide more efficient results and higher flexibility when dealing with large-scale problems. Thus, a number of hybrid metaheuristic algorithms have been proposed.

There are many hybrid optimization methods that employ the concept of the HS [11-14]. For instance, Kaveh and Talatahari [11] developed a hybrid optimization method for the optimum design of truss structures. Their proposed algorithm was based on a particle swarm optimization (PSO) with passive congregation (PSOPC), ant colony optimization, and the HS scheme.

Geem [12] proposed a hybrid HS incorporating the PSO concept. Known as particle swarm harmony search (PSHS), this algorithm was applied to the design of water distribution networks.

The mine blast algorithm (MBA) was developed to solve discrete and continuous optimization problems [15, 16]. The concept of the MBA was inspired by the process of exploding landmines. The results obtained by the MBA demonstrate its superiority in finding near-optimum solutions in early iterations and its fast mature convergence rate [16].

However, the exploitation (local search) ability of the MBA is not good as its exploration phase. Also, it suffers from a serious problem, that is, the MBA is almost memory-less optimizer. Though, the HS has many obvious advantages, it can be trapped in performing local search for solving optimization problems [8]. Moreover, its optimization performance is quite sensitive to its key control parameters.

Therefore, how to effectively fine-tune the key control parameters (i.e., HMS, HMCR, PAR, and bw) in the process of improvisation is a key research focus in the HS. In addition, its search precision and convergence speed are also an issue in some cases. Indeed, a reasonable balance between exploration and exploitation are beneficial to the performance of an algorithm [17].

Since, many modified and hybrid HS still cannot escape local minimum and adjust algorithm parameters effectively, so the relationship between the search mechanism of HS and the parameters is a very significant area for future research [18]. That deserves a lot more attention and this paper is thus motivated to focus on this research. Therefore, we propose the mine blast harmony search (MBHS), which embeds the HS into MBA to improve the exploitation phase in the MBA and exploration phase in the HS.

2 Mine Blast Harmony Search

The following sections provide detailed descriptions of the HS and its variants, MBA, and MBHS. The MBA and HS used in the MBHS are slightly improved.

2.1 Harmony Search Algorithm

Since the HS was first developed and reported in 2001 [1], it has been applied to various research areas and obtained considerable attention in different fields of research [6]. The HS intellectualizes the musical process of searching for a perfect state of harmony.

As musical performances search a fantastic harmony determined by aesthetic estimation, hence the optimization technique seeks a best state (global optimum) measured by an objective function value.

Further details of the HS can be found in the work of Geem et al. [1]. The main steps of the HS algorithm are summarized as below:

Step 1: Generate random vectors (x_1, x_2, …, x_{HMS}) up to the harmony memory size (HMS) and store them in the harmony memory (HM) matrix:

$$HM = \left[\begin{array}{ccc|c} x_1^1 & \cdots & x_n^1 & f(x^1) \\ \vdots & \ddots & \vdots & \vdots \\ x_1^{HMS} & \cdots & x_n^{HMS} & f(x^{HMS}) \end{array}\right]. \tag{1}$$

Step 2: Generate new harmony. For each component:

• With probability HMCR (harmony memory considering rate; $0 \le \text{HMCR} \le 1$), pick a stored value from the HM: $x_i' \leftarrow x_i^{\text{int}(u(0,1)\times HMS)+1}$.

• With probability (1-HMCR), pick a random value within the allowed range.

Step 3: If the value in Step 2 came from the HM:

· With probability PAR (pitch adjusting rate; $0 \le \text{PAR} \le 1$) change x_i':

$x_i' \leftarrow x_i' + bw \times (2rand - 1)$,

where rand is a uniformly distributed random number between zero and one and bw is the maximum change in pitch adjustment.

• With probability (1-PAR), do nothing.

Step 4: Select the best harmonies up to the HMS and consider them as the new HM matrix.

Step 5: Repeat Steps 2 to 5 until the termination criterion (e.g., maximum number of function evaluations) is satisfied.

To mention a few examples of improved versions of HS, Mahdavi et al. [8] proposed an improved HS (IHS) in which bw and PAR are not fixed values. During the optimization process, values of bw and PAR decrease and increase, respectively. This approach helps the exploitation phase of the IHS in the final iterations.

Afterwards, Geem and Sim [9] developed another improved variant, called parameter-setting-free HS (PSF-HS). In their improved method, the values of user parameters HMCR and PAR vary during the optimization process.

2.2 Mine Blast Algorithm

The MBA is inspired by the process of landmines explosion; shrapnel pieces are thrown away and collided with other landmines in the vicinity of the explosion area causing further explosions. Consider a landmine field where the goal is to clear all landmines. To clear all the landmines, the position of the most explosive mine must be determined.

This position corresponds to the optimal solution and its casualties considers as cost function [15]. Indeed, the MBA is developed to find the most explosive mine (i.e., the mine with the most casualties), and the aim is to reduce the casualties caused by the explosion of mines.

Similar to other population-based methods, the MBA requires an initial population of individuals. This population is generated by a first shot explosion. The population size is the number of shrapnel pieces (N_s) caused by an explosion. To begin, the MBA uses the lower and upper bound values (i.e., LB and UB) specified for a given problem for generating a random first shot solution (point).

At initialization step, we assume that the first shot point (X_0) is the best solution ($X_{Best} = X_0$) so far. The MBA starts with the exploration phase, which is responsible for comprehensively exploring the search space. Exploration (global search) and exploitation (local search) are the two critical steps for metaheuristic algorithms. The difference between the exploration and exploitation phases is how they affect the whole search process in finding the optimal solution.

To explore the search space from both small and large distances, an exploration factor, μ, is introduced [15]. This parameter, used in early iterations of MBA, is compared to an iteration number index (t). Explosion of a shrapnel piece triggers another landmine explosion at location $\vec{X}_t^e$. Hence, updating equations for the exploitation and exploration phases in the MBA are given in Equations (2) and (3), respectively [16]:

$$\vec{X}_t^e = \vec{X}_{Best} + \vec{d}_{t-1} \times ra\vec{n}dn^2 \times \cos\theta \qquad t \le \mu, \tag{2}$$

$$\vec{X}_t^e = \vec{X}_{best\_t}^e + \exp\left(-\sqrt{\frac{M_t}{D_t}}\right)\vec{X}_{Best} \qquad \mu < t, \tag{3}$$

Where $\vec{X}_{best\_t}^e$ in Equation (3) is the best exploded landmine at iteration t given as follows:

$$\vec{X}_{best\_t}^e = \vec{X}_{Best} + \vec{d}_{t-1} \times ra\vec{n}dn \times \cos(\theta) \qquad \mu < t. \tag{4}$$

randn is normally distributed random number and d_{t-1} is distance of each shrapnel piece. The Euclidean distance (D_t) and direction (M_t) between the current and previous best landmines (X_{Best} and X_{Best-1}) in m dimensions are given by:

$$D_t = \left[\sum_{i=1}^{m} (X_i^{Best} - X_i^{Best-1})^2 \right]^{1/2}, \quad \mu < t \, , \tag{5}$$

$$M_t = \frac{F_{Best}^t - F_{Best-1}^t}{D_t}, \quad \mu < t \cdot \tag{6}$$

When the Euclidean distance in Equation (5) between the current and previous best solutions is near zero (at final iterations), the exponential term in Equation (3) is equal to zero. The shrapnel angle of incidence, denoted by θ in Equations (2) and (4), is given by:

$$\theta = k \times \Delta \quad k = 0,1,2,...,N_s - 1 \, , \tag{7}$$

where Δ=360/N_s. The value of θ ranges from 0 to 360; the resulting value of cos(θ) ranges between -1 and 1, which generates solutions having harmonic orders. To improve MBA's global and local search abilities, the initial distance of shrapnel pieces (d_0=UB-LB) is gradually reduced at each iteration to quickly detect near location of the most explosive mine as follows:

$$\vec{d}_t = \frac{\vec{d}_{t-1}}{e^{(t/\alpha)}} \qquad t = 1,2,3,...,Max\_It \, , \tag{8}$$

where *Max_It* is maximum number of iteration and α is the reduction factor, the only sensitive user defined parameter of MBA, which depends on the complexity of the optimization problem. At the end of the optimization process, shrapnel distances are close to zero.

Indeed, the MBA starts with initial standard deviation named as initial distance of shrapnel pieces (d_0). By iteration continues, the MBA adaptively reduces the standard deviation in order to increase the exploitation and convergence effects.

Finally, steps of MBA are as follows:

Step 1: Choose initial parameters α, N_s (N_{pop}), and *Max_It*.

Step 2: Check the condition of the exploration factor (μ).

Step 3: If the condition of the exploration factor is satisfied, calculate the location of the exploded mine using Equation (2). Then, go to Step 8. Otherwise, continue to Step 4.

Step 4: Calculate the location of exploded landmine in the exploitation phase using Equation (4).

S*tep 5*: Does the shrapnel piece have a lower function value than the best temporal solution? If true, archive it.

Step 6: Calculate the Euclidian distance and direction between current and previous best solutions using Equations (5) and (6).

Step 7: Calculate improved location of exploded landmine in the exploitation phase using Equation (3).

Step 8: Does the shrapnel piece have a lower function value than the best temporal solution? If true, archive it.

Step 9: Update the X_{Best} (Best=Archive).

Step 10: Reduce the distance of shrapnel pieces adaptively using Equation (8).

Step 11: Check the stopping condition. If the stopping criterion is satisfied, the MBA stops. Otherwise, return to Step 2.

2.2.1 Setting Initial Parameters of MBA

Poor choices of algorithm parameters may result in a low convergence rate and undesired solutions. The following guidelines are suggested to fine tune the user-defined parameters.

The reduction factor (α) depends on the complexity of the problem, maximum number of iteration, and problem bounds. The value of α should be chosen so that at the final iteration, the distance of shrapnel pieces is approximately zero.

It is worth mentioning that being close to zero varies from one problem to another (depends on desire accuracy and tolerance). The following formula computes a suggested value for α used in the MBA given as follows:

$$\alpha_{Suggested} = \frac{M^2 + M}{2} \times \frac{1}{\ln(\vec{d}_0 / Tol)}, \tag{9}$$

where *Tol* is tolerance, a small value close to zero and *M* is maximum number of iteration. The exploration factor (μ) defines the number of iterations for the exploration phase. Increasing μ may result in getting trapped in a local minimum. For the MBA, we recommend μ be equal to the maximum number of iterations divided by five.

2.3 Mine Blast Harmony Search

Performance of HS is good at local search compared with its global search, and its convergence performance may also be an issue in some cases [18]. To overcome these drawbacks, combining the concepts and formulations of the MBA with the HS can improve the exploration and exploitation performances of both algorithms. The exploitation phase in the MBA is not as efficient as the exploration phase. Therefore, embedding the HS into the MBA can be considered to improve the exploitation phase in the MBA and exploration phase in the HS.

Since the MBA is a memory-less algorithm, almost no information is extracted dynamically during the search, whereas the HS uses memory to store information extracted during the search process (i.e., harmony memory matrix, Equation (1)).

The proposed hybrid MBHS involves two phases: (*i*) exploration phase using the strategy in the MBA and (*ii*) exploitation phase using the concepts of the HS, whereby memory consideration and pitch adjustment are employed along with the MBA operators.

For the MBHS, the updating exploitation equation in the MBA (Equation (3)) for avoiding problems with the dimension of the search space (m) is modified. Indeed, the perception of direction is replaced by moving to the best solutions in the MBHS. Hence, the new updating equations used in the MBHS are given as follows:

$$\vec{X}_t^e = \vec{X}_{Best} + \vec{d}_{t-1} \times ra\vec{n}dn \times \cos(\theta) \qquad \mu < t, \tag{10}$$

$$\vec{X}_t^e = \vec{X}_t^e + \exp\left(-\sqrt{\frac{1}{D_t}}\right) \times r\vec{a}nd \times \left\{\vec{X}_{Best} - \vec{X}_{Best-1}\right\}, \quad \mu < t' \tag{11}$$

In addition, the HS used in the MBHS is not the standard HS. The HS utilized in the MBHS has borrowed some features of IHS [8] and PSF-HS [9] for adaptively reducing and increasing the user parameters of HS. In this research, we also define the new concept for HM having variables size, so called dynamic harmony memory (DHM). Indeed, the HMS is not fixed parameter in the MBHS.

Increasing the value of HMS causes more exploration in the search space, and sometimes causes the optimization results to diverge. In the current hybrid MBHS, the value of HMS is changed at early and final iterations and it is fixed in between, as shown in Fig. 1.

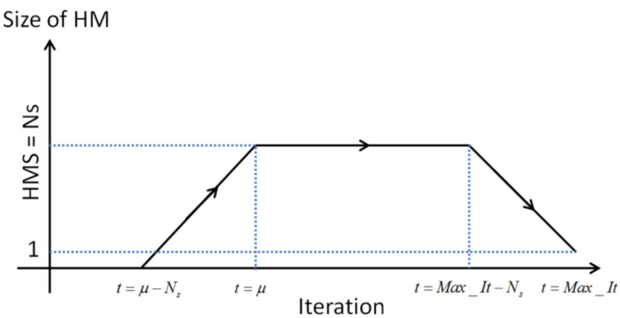

Fig. 1 Size of DHM during optimization process

For the sake of reducing the user parameters in the MBHS, value of HMS is considered to be the population size (N_s). By decreasing the value of HMS in the final iterations, further exploitation close to the current best solution can be achieved. In general, there is only one user parameter in the MBHS, the reduction factor (α) as for used in the MBA.

In addition, we assume that the bandwidth (bw) user parameter in the HS acts similarly to the distance of shrapnel pieces (d_{t-1}). Therefore, the bw has been merged with d_{t-1}, and adaptively reduces at each iteration as follows:

$$b\vec{w}_t = \frac{b\vec{w}_{t-1}}{e^{(t/\alpha)}} \qquad t = 1,2,3,...,Max\_It. \tag{12}$$

Initial values of HMCR and PAR are automatically tuned in the optimization process as given in Equations (13) and (14). The values of HMCR and PAR in HS phase are changed right after the exploration phase. The following equations describe the variation of these user parameters:

$$HMCR(t) = \begin{cases} \dfrac{t}{Max\_It} & \mu < t \\ 0 & t \leq \mu \end{cases}, \tag{13}$$

$$PAR(t) = \begin{cases} 1 - \dfrac{t}{Max\_It} & \mu < t \\ 0 & t \leq \mu \end{cases}. \tag{14}$$

In this research, we assume that values of HMCR and PAR linearly increase and decrease, respectively, at each iteration. Therefore, there is no need to tune these parameters during the optimization process. The (probability) value of HMCR goes from zero to 0.99, and from one to near zero for the PAR parameter. The reason to choose maximum value of 0.99 for the HMCR is for having one percent chance to generate random solutions at final iteration.

By progressing the optimization in the MBHS, the exploration approach decreases in importance and the exploitation phase becomes dominant ($\mu < t$). Indeed, in the final iterations, the MBHS executes only a local search near to the best current solution.

3 Numerical Optimization Results

MATLAB was used to code and implemented the algorithms. To ensure statistically significant results, 50 independent optimization runs were carried out for each test problem in this paper.

3.1 Benchmark Optimization Problems

The MBHS has been tested on eleven unconstrained benchmark functions. In order to observe the effects of proposed MBHS and having fair discussion, the MBA and HS also have been implemented for considered benchmarks. The dimensions of benchmark functions were 200 and 500. Properties of these functions are represented in Table 1.

Table 1 Properties of F1 to F11

Function	Range	Optimum ($f(x^*)$)
F_1 (Hyper Sphere)	$[-100,100]^m$	0
F_2 (Schwefel 2.21)	$[-100,100]^m$	0
F_3 (Rosenbrock)	$[-100,100]^m$	0
F_4 (Rastrigin)	$[-5,5]^m$	0
F_5 (Griewank)	$[-600,600]^m$	0
F_6 (Ackley)	$[-32,32]^m$	0
F_7 (Schwefel 2.22)	$[-10,10]^m$	0
F_8 (Schwefel 1.2)	$[-65.536,65.536]^m$	0
F_9 (Bohachevsky)	$[-15,15]^m$	0
F_{10} (Schaffer)	$[-100,100]^m$	0
F_{11} (Extended f_{10})	$[-100,100]^m$	0

Talking about maximum number of function evaluations (NFEs), considered as stopping condition in this paper, the predefined NFEs is 5000 multiple by dimension size for each function.

User parameters of MBA and MBHS were set to the recommended values for μ and α given in Section 2.2.1 and population size of 50 (*Tol.* = 1.00e-14). Accordingly, the user parameters of the HS for the considered benchmarks were: a harmony memory size of 50, and HMCR, PAR, and bw values of 0.98, 0.1, and 0.01, respectively, as suggested by [1].

The obtained statistical results (i.e., error values: $f(x) - f(x^*)$) for dimensions 200, and 500 are represented in Tables 2 and 3, respectively. The best obtained result (error) at the end of each optimization process is recorded during each run. The best, average, and worst errors and standard deviation (SD) are shown in Tables 2 and 3. In Tables 2 and 3, 0.00e+00 means 1.00e-324 (defined accuracy for zero in MATLAB).

By observing Tables 2 and 3, the MBHS considerably has reduced the error compared with its original optimizers (i.e., MBA and HS). From the obtained optimization results especially for large-scale problems, we can infer that the combination of HS with the MBA leads us to develop a hybrid optimization method having better performance and efficiency.

Furthermore, Table 4 summarizes the average error values of MBHS, MBA, and HS and compares those findings with the results using other optimizers. The PSO [19], imperialist competitive algorithm (ICA) [20], and gravitational search algorithm (GSA) [21] have been coded and implemented in this paper for comparison purposes. In this study, all error values below 1.00e-14 assume to be 0.00e+00 in Table 4. Looking at Table 4, the MBHS shows its superiority not only against the HS and MBA, also represented competitive results compared with other optimizers.

Table 2 Statistical optimization results for m = 200 for the MBA, HS, and MBHS

Function	Method	Best	Average	Worst	SD
F_1	HS	1.22e+03	1.29e+03	1.44e+03	7.30e+01
	MBA	7.96e-13	8.75e-13	9.66e-13	6.48e-14
	MBHS	5.68e-14	5.85e-14	5.91e-14	1.21e-14
F_2	HS	2.38e+01	2.46e+01	2.63e+01	7.21e-01
	MBA	2.25e+01	4.67e+01	8.55e+01	2.47e+01
	MBHS	6.82e-13	9.89e-13	1.71e-12	4.30e-13
F_3	HS	4.44e+06	6.55e+06	8.02e+06	1.10e+06
	MBA	2.37e+02	1.25e+03	3.87e+03	1.52e+03
	MBHS	1.98e+02	1.98e+02	1.98e+02	2.87e-02
F_4	HS	8.30e+01	9.77e+01	1.07e+02	7.32e+00
	MBA	6.40e+02	9.83e+02	1.51e+03	3.60e+02
	MBHS	0.00e+00	4.54e-14	5.68e-14	2.54e-14
F_5	HS	1.09e+01	1.28e+01	1.50e+01	1.47e+00
	MBA	9.94e-13	1.97e-03	9.86e-03	4.41e-03
	MBHS	0.00e+00	1.71e-14	2.84e-14	1.55e-14
F_6	HS	4.34e+00	4.54e+00	4.74e+00	1.46e-01
	MBA	4.18e+00	1.68e+01	2.00e+01	7.06e+00
	MBHS	1.99e-13	2.33e-13	2.56e-13	2.38e-14
F_7	HS	2.43e+01	2.56e+01	2.69e+01	7.13e-01
	MBA	4.84e+00	4.53e+02	8.00e+02	4.10e+02
	MBHS	3.36e-13	4.07e-13	4.70e-13	4.79e-14
F_8	HS	3.45e+04	3.62e+04	3.77e+04	1.15e+03
	MBA	3.30e+00	1.24e+01	2.46e+01	1.10e+01
	MBHS	1.17e-26	1.29e-26	1.40e-26	1.58e-27
F_9	HS	1.91e+02	2.06e+02	2.25e+02	1.11e+01
	MBA	7.96e+01	8.73e+01	9.55e+01	7.19e+00
	MBHS	0.00e+00	0.00e+00	0.00e+00	0.00e+00
F_{10}	HS	3.87e+02	4.36e+02	4.61e+02	2.98e+01
	MBA	1.12e+03	1.42e+03	1.71e+03	2.88e+02
	MBHS	1.09e-05	1.18e-05	1.24e-05	6.01e-07
F_{11}	HS	4.09e+02	4.41e+02	4.70e+02	2.86e+01
	MBA	1.11e+03	1.65e+03	1.88e+03	3.20e+02
	MBHS	1.06e-05	1.22e-05	1.37e-05	1.10e-06

Table 3 Statistical optimization results for *m*=500 using the HS, MBA, and MBHS

Function	Method	Best	Average	Worst	SD
F_1	HS	4.89e+04	5.38e+04	5.76e+04	2.65e+03
	MBA	1.99e-12	2.21e-12	2.39e-12	1.63e-13
	MBHS	5.11e-14	5.23e-14	5.68e-14	2.32e-14
F_2	HS	5.72e+01	5.79e+01	5.85e+01	4.23e-01
	MBA	1.17e+01	4.17e+01	8.59e+01	3.37e+01
	MBHS	6.82e-13	1.39e-12	2.39e-12	6.23e-13
F_3	HS	6.67e+09	7.39e+09	7.91e+09	3.96e+08
	MBA	4.91e+02	7.43e+02	1.36e+03	3.51e+02
	MBHS	4.97e+02	4.97e+02	4.97e+02	4.21e-02
F_4	HS	8.30e+02	8.58e+02	9.13e+02	2.51e+01
	MBA	2.24e-02	2.47e-02	2.95e-02	2.85e-03
	MBHS	5.22e-014	5.45e-14	5.94e-14	4.23e-14
F_5	HS	4.42e+02	4.88e+02	5.42e+02	2.70e+01
	MBA	2.90e-12	1.48e-03	7.40e-03	3.31e-03
	MBHS	2.84e-14	2.99e-14	3.12e-14	3.53e-14
F_6	HS	1.09e+01	1.12e+01	1.14e+01	1.52e-01
	MBA	1.52e+01	1.68e+01	2.00e+01	1.89e+00
	MBHS	2.84e-13	2.91e-13	2.95e-13	2.84e-13
F_7	HS	1.92e+02	2.02e+02	2.13e+02	7.14e+00
	MBA	1.20e+96	1.54e+96	1.88e+96	2.32e+95
	MBHS	4.12e-13	4.48e-13	4.79e-13	2.95e-14
F_8	HS	4.55e+06	4.89e+06	5.11e+06	2.31e+03
	MBA	6.65e+02	9.25e+02	1.11e+03	2.33e+02
	MBHS	1.62e-26	1.83e-26	2.03e-26	2.95e-27
F_9	HS	3.83e+03	3.98e+03	4.26e+03	1.22e+02
	MBA	2.08e+02	2.24e+02	2.35e+02	1.13e+01
	MBHS	0.00e+00	0.00e+00	0.00e+00	0.00e+00
F_{10}	HS	1.65e+03	1.68e+03	1.71e+03	2.30e+01
	MBA	2.46e+02	3.29e+03	5.32e+03	1.96e+03
	MBHS	1.83e-05	1.88e-05	1.94e-05	5.13e-07
F_{11}	HS	1.71e+03	1.92e+03	2.10e+03	2.67e+02
	MBA	2.96e+03	4.09e+03	4.70e+03	6.65e+02
	MBHS	1.85e-05	1.89e-05	1.98e-05	5.32e-07

Table 4 Comparison of average error values for different optimizers for F_1 to F_{11}

Function	m	PSO	ICA	GSA	HS	MBA	MBHS
F_1	200	1.10e+04	2.20e+04	1.06e-12	1.29e+03	8.75e-13	5.85e-14
	500	4.51e+09	4.18e+05	8.25e-12	5.38e+04	2.21e-12	5.23e-14
F_2	200	2.22e+01	8.97e+01	8.08e+00	2.46e+01	4.67e+01	9.89e-13
	500	2.59e+01	9.64e+01	1.1.7e+01	5.79e+01	4.17e+01	1.39e-12
F_3	200	1.07e+08	6.10e+08	1.83e+02	6.55e+06	1.25e+03	1.98e+02
	500	7.11e+08	9.96e+10	9.75e+02	7.39e+09	7.43e+02	4.97e+02
F_4	200	1.12e+03	1.62e+03	1.17e+02	9.77e+01	8.39e-03	4.54e-14
	500	3.73e+03	5.28e+03	3.62e+02	8.58e+02	2.47e-02	5.45e-14
F_5	200	9.55e+01	1.23e+02	8.00e-01	1.28e+00	1.97e-03	1.71e-14
	500	4.12e+02	3.85e+03	9.07e-01	4.88e+02	1.48e-03	2.99e-14
F_6	200	1.01e+01	1.96e+01	4.60e-09	4.54e+00	1.68e+01	2.33e-13
	500	1.06e+01	2.01e+01	9.51e-09	1.12e+01	1.68e+01	2.84e-13
F_7	200	9.04e+01	8.90e+02	2.02e-07	2.56e+01	4.53e+02	4.07e-13
	500	3.20e+02	2.35e+03	9.22e-07	2.02e+02	1.54e+96	4.48e-13
F_8	200	4.83e+05	1.06e+06	2.04e-14	3.62e+04	1.24e+01	0.00e+00
	500	4.01e+06	3.39e+07	4.64e-13	4.89e+06	9.25e+02	0.00e+00
F_9	200	9.84e+02	1.55e+03	2.44e+00	2.06e+02	8.73e+01	0.00e+00
	500	3.76e+02	2.70e+04	2.09e+01	3.98e+03	2.24e+02	0.00e+00
F_{10}	200	8.25e+02	1.82e+03	1.11e+02	4.36e+02	1.42e+03	1.18e-05
	500	2.27e+03	5.01e+03	6.57e+02	1.68e+03	3.29e+03	1.88e-05
F_{11}	200	8.18e+02	1.82e+03	1.00e+02	4.41e+02	1.65e+03	1.22e-05
	500	2.29e+03	4.97e+03	6.61e+02	1.92e+03	4.09e+03	1.89e-05

4 Conclusions

A hybrid metaheuristic optimization method has been introduced in this paper. The combination of mine blast algorithm (MBA) and harmony search (HS) algorithm produced a hybrid optimization method with excellent exploration and exploitation capabilities.

The MBA is memory-less optimization method, while the HS is memory-based algorithm. Using the advantages of MBA in global search and HS in local search and thinking about combing the HS and MBA led to develop new hybrid optimization method, so called mine blast harmony search (MBHS). Furthermore, various improvements were applied to the standard HS and MBA.

A new concept for harmony memory (HM) in HS phase, so called dynamic HM, has been proposed. Also, the perception of direction in the MBA phase has been replaced by the concept of moving toward the best solutions in the MBHS.

Eleven unconstrained benchmarks, widely used in the literature, with different design variables (i.e., from 200 to 500) have been tackled. The optimization results obtained by the proposed hybrid method show that it surpasses both the

MBA and HS in terms of solution quality and having better statistical results. Moreover, further comparisons with other optimizers indicate that the MBHS attains the optimal solution more accurately and efficiently.

Acknowledgment This work was supported by the National Research Foundation (NRF) of Korea under a grant funded by the Korean government (MSIP) (NRF-2013R1A2A1A01013886).

References

1. Geem, G.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. Simulation **76**(2), 60–68 (2001)
2. Kim, J.H., Geem, G.W., Kim, E.S.: Parameter estimation of the nonlinear Muskingum model using harmony search. J. of Amer. Wat. Res. Assoc. **37**(5), 1131–1138 (2001)
3. Geem, G.W., Kim, J.H., Loganathan, G.V.: Harmony search optimization: application to pipe network design. Int. J. Modelling Simul. **22**(2), 125–133 (2002)
4. Fesanghary, M., Damangir, E., Soleimani, I.: Design optimization of shell and tube heat exchangers using global sensitivity analysis and harmony search algorithm. Appl. Therm. Eng. **29**(5–6), 1026–1031 (2009)
5. Degertekin, S.: Optimum design of steel frames using harmony search algorithm. Struct. Multidiscip. Optim. **36**(4), 393–401 (2008)
6. Geem, Z.W. (ed.): Harmony Search Algo. for Structural Design Optimization. SCI, vol. 293. Springer, Berlin (2009)
7. Kim, J.H., Baek, C.W., Jo, D.J., Kim, E.S., Park, M.J.: Optimal planning model for rehabilitation of water network. Water Sci. and Technol. Water Supply **4**(3), 133–147 (2004)
8. Mahdavi, M., Fesanghary, M., Damangir, E.: An improved harmony search algorithm for solving optimization problems. Appl. Math. Comput. **188**(2), 1567–1579 (2007)
9. Geem, Z.W., Sim, K.B.: Parameter-setting-free harmony search algorithm. Appl. Math. Comp. **217**(8), 3881–3889 (2010)
10. Wang, L., Yang, R., Xu, Y., Niu, Q., Pardalos, P.M., Fei, M.: An improved adaptive binary Harmony Search algorithm. Inform. Sciences **232**, 58–87 (2013)
11. Kaveh, A., Talatahari, S.: Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. Comput. Struct. **87**(5–6), 267–283 (2009)
12. Geem, Z.W.: Particle-swarm harmony search for water network design. Eng. Optim. **41**(4), 297–311 (2009)
13. Ayvaz, M.T., Kayhan, A.H., Ceylan, H., Gurarslan, G.: Hybridizing the harmony search algorithm with a spreadsheet 'solver' for solving continuous engineering optimization problems. Eng. Optim. **41**(12), 1119–1144 (2009)
14. Fesanghary, M., Mahdavi, M., Minary-Jolandan, M., Alizadeh, Y.: Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems. Comput. Methods Appl. Mech. Eng. **197**(33–40), 3080–3091 (2008)
15. Sadollah, A., Bahreininejad, A., Eskandar, H., Hamdi, M.: Mine blast algorithm for optimization of truss structures with discrete variables. Comput. Struct. **102–103**, 49–63 (2012)

16. Sadollah, A., Bahreininejad, A., Eskandar, H., Hamdi, M.: Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. Appl. Soft Comput. **13**(5), 2592–2612 (2013)
17. Yadav, P., Kumar, R., Panda, S.K., Chang, C.S.: An intelligent tuned harmony search algorithm for optimization. Inf. Sciences **196**, 47–72 (2012)
18. Yang, H.O., Gao, L., Li, S., Kong, X., Zou, D.: On the iterative convergence of harmony search algorithm and a proposed modification. Appl. Math. Comput. **247**(15), 1064–1095 (2014)
19. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. of IEEE Int. Conf. on Neural Networks, vol. IV, pp. 1942–1948 (1995)
20. Atashpaz-Gargari, E.: Lucas, C: Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. IEEE CEC **7**, 4661–4666 (2007)
21. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: A gravitational search algorithm. Inform. Sciences **179**, 2232–2248 (2009)