EFFECT OF USER DEFINED PARAMETERS OF HARMONY SEARCH ALGORITHM (HSA) FOR UNCONTRAINED OPTIMIZATION PROBLEMS

Tanu Puri[1], Souvik Ganguli[2]
[1]ME (PS) Student, [2]Assistant Professor,
Department of Electrical & Instrumentation Engineering, Thapar University, Patiala
Email: [1]tanupuri89@gmail.com, [2]souvik.ganguli@thapar.edu

Abstract— This paper presents the efficacy of Harmony Search Algorithm (HSA) on some nonlinear benchmark functions namely sphere and rosenbrock functions. The parameters of harmony search algorithm are varied and their effects are studied. The results show that reosenbrock function gives better results by taking more number of iterations.

Index Terms— Harmony Search Algorithm (HSA), Metaheuristic Algorithm, Sphere Function, Rosenbrock Function

I. INTRODUCTION

From engineering design to business planning optimization is everywhere. Optimization implies finding the best possible solution of a given problem. The aim is to compute the values that maximize or minimize an objective function subjects to certain constraints. Optimization algorithms are broadly classified as deterministic and stochastic algorithms. Deterministic algorithms i.e. hill-climbing method will produce the same set of solutions if the iterations start with the proper values of input whereas, stochastic algorithms do not repeat the same set of solutions even they are called with the same initial starting point due to randomness component attached with it, though the final results obtained by stochastic algorithms will usually converge to the same optimal solutions within a given accuracy. Stochastic algorithms often have a deterministic component and a random component attached with it so that it is possible to find global optimal solution in less time. The stochastic component can take many forms such as simple randomization by randomly sampling the search space or by random walks [1].

Most of the stochastic algorithms are considered as meta-heuristic. Some good examples are Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Firefly Algorithm (FFA), Harmony Search (HS) etc .Most meta-heuristic algorithms are nature-inspired. Two major components of any meta-heuristic algorithms are: intensification (selection of the best solutions) and randomization or diversification. The selection of the best solution defines that the solutions will converge to the optimality but randomization avoids the solutions to be trapped at local optima and, at the same time, increase the diversity of the solutions to find the global optima. The good combination of these two components will usually decide the capacity of the algorithms.

An algorithm, developed by Zong woo Geem in the year 2001 known as Harmony Search Algorithm (HAS) [2-3] is a music-based meta-heuristic optimization algorithm. The algorithm is inspired by the music harmony for getting optimal harmony in every new improvisation. The musician finds the best harmony in three ways: picks up the best harmony from memory, or adjusts the pitch or does randomization to get optimal solution of the problem.

The objective of this paper is to find the performance of harmony search algorithms for some non-linear standard functions. Section II discusses the details of harmony search algorithm. Section III presented the representation of some non-linear benchmark functions. Section IV shows the results through the variation of harmony search parameters while Section V discusses the conclusion and proposes future scope of work.

II. Harmony Search Algorithm

The HS algorithm, developed by Zong woo Geem, mimics a musical improvisation process in which, the musicians in an orchestra try to find a fantastic harmony through musical improvisations [2].

Musical performances seek a best state (fantastic harmony) determined by aesthetic estimation, as the optimization problem seek an optimum state [3]. When a musician improvises new pitch, he (or she) has to follow three steps: Memory Consideration, pitch adjustment, randomization. HS has successfully been applied to a wide variety of practical optimization problems like pipe-network design, structural optimization [4], the vehicle routing problem [5] etc.

A. Harmony Search Algorithm Steps

The main step in the procedure of Harmony Search Algorithm (HSA) is as follows [6].

Step 1.Define objective function.

Step 2.Initialize HSA algorithm parameters like Harmony memory (HM), Harmony memory considerating rate (HMCR), Pitch adjusting rate (PAR), Number of improvisations (NI), stopping criteria, number of decision variables (N).

Step 3.Generate the initial harmony memory (i= 1 to number of harmony vectors) randomly within the range.

Step 4.Determine functional value of initial Harmony memory.

Step 5.Set iteration count, iteration =1

Step 6.Starting of Harmony Search, if generated random value > HMCR, Then select the value of parameter randomly as given by,

$$x_{inew} = L(x_{old}) + rand\,\varepsilon(0,1) * bandwidth$$

(where, L(x_{old}) is the lower limit of Harmony memory solution vector and rand is a random number generator uniformly distributed in [0, 1]), Otherwise, choose value from Harmony memory and adjust the pitch as follows:

$$x_{new} = x_{old} + bandwidth(rand - 0.5)$$

Step 7.Update the value of objective function and replace the worst solution with new better solution.

Step 8.Check the stopping criteria and iteration > maximum iteration, if it is satisfied GOTO step 10

Step 9.Advance the iteration count by iteration = iteration+1 and GOTO step 6.

Step 10.Find the optimal value of the function.

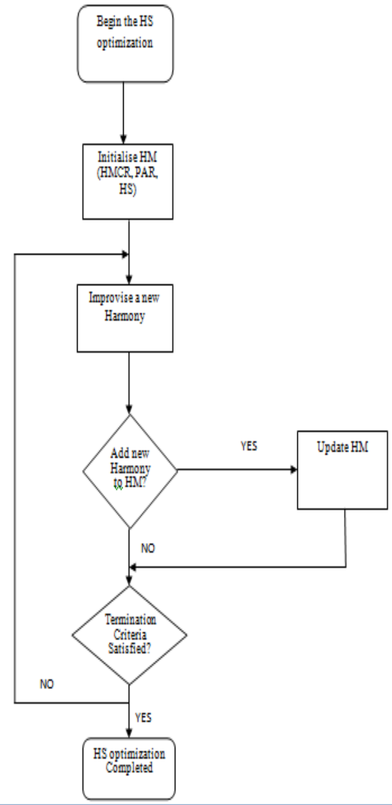

Fig. 1: Flow Chart of HSA

III.STANDARD FUNCTIONS

A. Sphere Function

Sphere function is also known as De Jong's function since it is the first function of De Jong. It is one of the simplest benchmark functions. This function is continuous, unimodal and convex. It has following general definition as per [7].

$$f(x) = \sum_{i=1}^{n} x_i^2$$

Test area is usually restricted to hypercube $-5.12 \leq x_i \leq 5.12$, i = 1. . n. Global minimum f(x) = 0 is obtainable for $x_i = 0$, i = 1 .. n. The variation of harmonies and the 3D plot for sphere function are shown in Fig. 2 and Fig. 3 respectively.

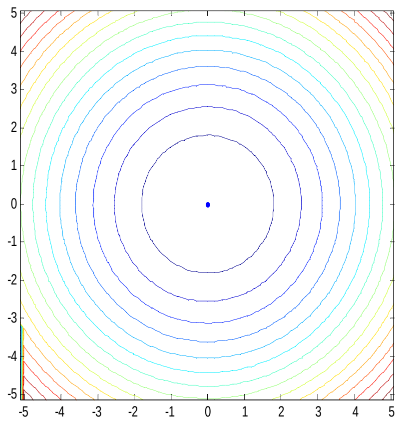

Fig. 2: Variation of harmonies in sphere function

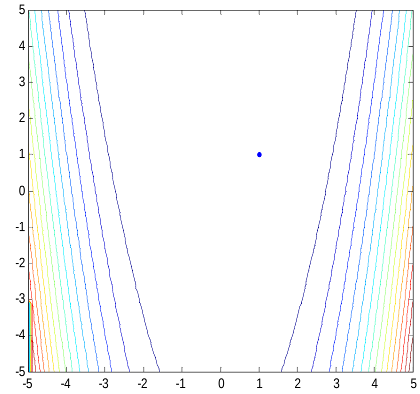

Fig.4: Variation of Harmonies of Rosenbrock Function

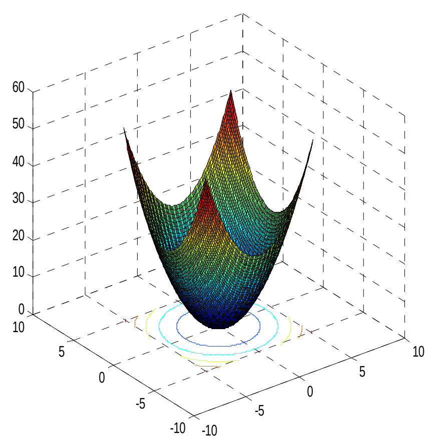

Fig. 3: 3D plot of sphere function

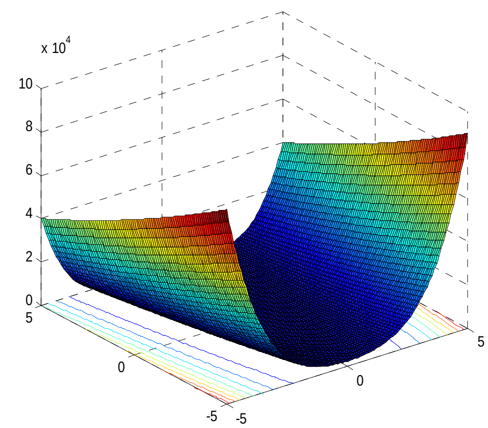

Fig. 5: 3D plot of Rosenbrock Function

B. Rosenbrock Function

The Rosenbrock function, also referred to as the Valley or Banana function is a popular test problem for gradient-based optimization algorithm [8]. The function is described by the formula:

$$f(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^{2})^2+(1-x_i)^2]$$

Test area is usually restricted to hyphercube $-5 \le x_i \le 5$, i = 1,. . . , n. Its global minimum equal f(x) = 0 is obtainable for x_i , i = 1, . . . , n. The variation of harmonies and the 3D plot for rosenbrock function are shown in Fig. 4 and 5 respectively.

IV. RESULTS

Table I shows the best estimates of sphere and rosenbrock functions using harmony search algorithm. Table II to VII report the effect of Harmony Search parameters namely harmony memory accepting rate, harmony memory size and pitch adjusting rate on the non-linear benchmark functions.

Table I Results of Harmony Search Algorithm

Function	Dimension (N)	HMS	PAR	HMCR	Search Domain	x1	x2	fmin	count	CPU time (seconds)
Rosen brock	2	20	0.7	0.95	[-5 5]	0.9975	0.9950	6.6876×10^{-6}	3224	1.559414
Sphere	2	20	0.7	0.95	[-5.12 5.12]	0.0029	0.0001	8.4162×10^{-6}	560	1.846967

Table II
Results of Rosenbrock function by varying Pitch adjusting rate (PAR)

Function	Dimension(N)	HMS	PAR	HMCR	Search Domain	x1	x2	fmin	count	CPU Time (seconds)
Rosen brock	2	20	0.7	0.7	[-5 5]	1.0014	1.0025	6.8230×10^{-6}	18898	5.177858
				0.75		1.0011	1.0023	2.0686×10^{-6}	21406	5.768367
				0.8		0.9989	0.9979	2.2546×10^{-6}	22378	6.004149
				0.85		1.0009	1.0017	1.9592×10^{-6}	5530	2.141367
				0.9		0.9971	0.9941	0.9941×10^{-6}	21064	5.683221

Table III
Results of Rosenbrock function by varying Harmony memory Accepting rate (HMCR)

Function	Dimension (N)	HMS	PAR	HMCR	Search Domain	x1	x2	fmin	count	CPU Time (seconds)
Rosen brock	2	20	0.7	0.7	[-5 5]	1.0014	1.0025	6.8230×10^{-6}	18898	5.177858
				0.75		1.0011	1.0023	2.0686×10^{-6}	21406	5.768367
				0.8		0.9989	0.9979	2.2546×10^{-6}	22378	6.004149
				0.85		1.0009	1.0017	1.9592×10^{-6}	5530	2.141367
				0.9		0.9971	0.9941	0.9941×10^{-6}	21064	5.683221

Table IV
Results of Rosenbrock function by varying Harmony Memory Size (HMS)

Function	Dimension (N)	HMS	PAR	HMCR	Search Domain	x1	x2	fmin	count	CPU Time (seconds)
Rosenbrock	2	10	0.7	0.95	[-5 5]	0.9986	0.9969	8.7541×10^{-6}	9197	2.970938
		30				0.9989	0.9975	6.4426×10^{-6}	21211	5.632414
		40				1.0008	1.0017	7.7780×10^{-7}	8674	2.786523
		50				0.9988	0.9975	1.5890×10^{-6}	20793	5.577444
		60				1.0023	1.0047	6.4308×10^{-6}	16240	4.515762

Table V
Results of Sphere function by varying Harmony Memory Considering rate (HMCR)

Function	Dimension (N)	HMS	PAR	HMCR	Search Domain	x1	x2	fmin	count	CPU Time (seconds)
Sphere	2	20	0.7	0.7	[-5.12 5.12]	0.0018	-0.0008	3.9856×10^{-6}	371	0.865585
				0.75		0.9484	0.1737	9.2960×10^{-7}	1049	1.069324
				0.8		-0.0003	-0.0015	2.4409×10^{-6}	302	0.859589
				0.85		0.0014	-0.0019	5.4515×10^{-6}	497	0.884250
				0.9		0.0021	0.0012	5.9454×10^{-6}	257	0.835925

Table VI
Results of Sphere function by varying Harmony memory Size (HMS)

Function	Dimension (N)	HMS	PAR	HMCR	Search Domain	x1	x2	fmin	count	CPU Time (seconds)
Sphere	2	20	0.5	0.95	[-5.12 5.12]	0.0013	0.0023	7.1269×10^{-6}	395	0.963389
			0.55			-0.0022	-0.0023	9.7974×10^{-6}	406	0.888991
			0.6			0.0002	-0.0011	1.3386×10^{-6}	312	0.886139
			0.65			0.6366	0.3009	4.9575×10^{-6}	288	0.900525

Table VII
Results of Sphere function by varying Harmony memory Size (HMS)

Function	Dimension (N)	HMS	PAR	HMCR	Search Domain	x1	x2	fmin	count		CPU Time (seconds)
Sphere	2	10	0.7	0.95	[-5.12 5.12]	-0.0022	-0.0003	4.8402×10^{-6}	284		0.871129
		30				-0.0009	0.0019	4.4837×10^{-6}	334		0.908296
		40				-0.7569	0.5456	8.7058×10^{-7}	920		1.021674
		50				-0.0010	0.0011	2.2599×10^{-6}	561		1.020708
		60				0.0026	0.0002	6.8373×10^{-6}	1219		1.090346

CONCLUSION

In this paper, we have studied the effects of harmony search parameters on nonlinear standard functions such as: sphere, and rosenbrock. It can be concluded that by varying harmony search parameters, the rosenbrock function takes more time to give the accurate results. The proposed method can further be applied to some constrained optimization problems.

REFERENCES

[1] Yang X. S., "Firefly algorithm, stochastic test functions and design optimisation", *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010.

[2] Geem, Z. W., "Nature inspired Harmony Search Algorithm- Theory and application", *Springer publication, Studies in computational intelligence*, vol.191, 2009.

[3] Geem, Z. W., Kim, J.H., and Loganathan G.V., "A New Heuristic Optimization Algorithm: Harmony Search", *Simulation 76,*

[4] Lee, K.S., and Geem, Z. W., "A new structural optimization method based on the harmony search algorithm," *Computers and Structures*, vol.82, no.9, pp.781-798, Apr. 2004.

[5] Geem, Z.W., Lee, K.S., and Park, Y. "Application of Harmony Search to vehicle routing," *American Journal of Applied. Sciences*, vol.2, no.12, pp.1552-1557, 2005.

[6] Arul, R., Ravi, G., Velusami, S., "Non convex Economic Dispatch with heuristic Load patterns using Harmony Search Algorithm", *International journal of Computer Applications*, volume.16, no.1, Feb. 2011.

[7].*ict.pwr.wroc.pl/files/docs/functions.pdf, 2005.*