Hyperdize Jaya Algorithm for Harmony Search Algorithm's Parameters Selection

Alaa A. Al-Omoush[1], AbdulRahman A. Alsewari[2], Ameen Bahomaid[3], Hammoudeh S. Alamri[4], Kamal Z. Zamli[5]

Faculty of Computer Systems and Software Engineering
University Malaysia Pahang
Gambang, Pahang, Malaysia
E-mail: [1]allaa1030@gmail.com, [2]alsewari@ump.edu.my, [3]ameenalib@gmail.com, [4]Ha.amri@gmail.com
[5]kamalz@ump.edu.my

Abstract—this paper present a successful method to tune the parameters of the harmony search algorithm, which is a well-known meta-heuristic algorithm. Choosing the values of HS parameters consider as a difficulty when we use it in different cases, and we tried to solve this problem by using another meta-heuristic algorithm to pick the right value for one of these parameters, which is harmony memory consideration rate. To evaluate the performance of the amended algorithm, we have applied it on different bench-mark functions that shows its good performance, in terms of result accuracy in contrast with the original algorithm Harmony Search with less number of iterations, and when we compare it with two other well-known variants of HS.

Keywords—Hyperdize; Tuning; Metahuristic algorithms; Harmony search Algorithm; stochastic search methods; mathematical functions minimization.

1. INTRODUCTION

Harmony Search (HS) consider as famous Evolutionary algorithms (EA), and basically these algorithms begin by creating a random values that's possibly provide a solution for specific problem. The fitness of every value evaluated based on evaluation function. Through every rotation inside the EA, should be a nominating process to create a better population. The main goal of the nomination process is to deviate toward the fitter values to catch it, and to insert it in the next population. Each value will be modified using mutation and alteration based on two parent values[1].

Optimization algorithms used to find the most optimal solutions for a well-defined problems that has a difficulty to be solved using a traditional techniques[2], There are two main categories of optimization: the first category known as exact algorithm which is guaranteed to find an optimal solution and to prove its optimality for every instance of the hard optimization problem, but its drawback is consuming huge amount of time and resources, on the other hand the second category is heuristic algorithm which care about finding good solution in limiting time more than guarantee of finding optimal solutions[3]. "Met-heuristic algorithms consider as high-level problem-independent algorithmic framework of the heuristic optimization techniques"[4]. Different optimization methods influenced by the nature to solve a well-known NP problem like genetic algorithm, particle swarm optimization, Tabu search, ant colony optimization, bees' algorithm, artificial immune system and simulated annealing, which are widely used in different science and to solve engineering issues[5]. HS is well known meta-heuristic algorithm that shows a significant result in engineering fields and computer science including AI & SE[6].

The rest of the paper will be organized as: Section II we will illustrate the explanation and the problem of original HS and the well-known previous variants of HS and how they fix the selection of parameters. In section III, we propose and describe the new modified HS Algorithms. Experiments and discussion will be elaborated in section IV. Finally, the conclusion will be in section V.

2. HARMONY SEARCH & ITS VARIANTS

A. HS Description

HS algorithm mimic the idea of searching and improvisation process of the musician to find new harmony in music[7], and its containing few parameters like harmony consideration rate (HMCR), which have an important effect on the algorithm efficiency, and must be chosen before the algorithm started its search. Choosing the right value of these parameters is really vital to get the right result[8], and to accomplish that we used another meta-heuristic algorithm to pick the values of HMCR and give the good result. The HS algorithm as describe by Geem [9] contain 5 main steps:

1. Step 1: create initial values of HS parameters and values: The optimization problem will be determined either maximum or minimum of the objective function f(x), x is the prospect solution from N decision variables (x_i), within (lower bound ≤ x_i ≥ upper bound), of all the decision variables. In this step HS parameters will be initialized, such as Harmony memory consideration rate (HMCR), Bandwidth (BW), Pitch adjustment rate (PAR), Number of iteration (NI), Harmony memory size (HMS).
2. Step 2: in this step harmony memory values will be initialized within the upper and lower range, where:
   x_i = lower bound + r * (upper bound - lower bound) where r value between (0&1).
3. Step 3: this step will have the improvisation of new harmony in this way:

   > For (i to n)
   >> If (random value between (0&1) ≤ HMCR
   >>> $x_i{}^\grave{} = x_{ij}$, where j =(1….HMS)
   >>> If (random value between (0&1) ≤PAR)
   >>>> Then ($x_{ij} = x_{ij} \pm r * bw$)
   >>> Else
   >>>> x_{ij} = lower bound + r* (upper bound-lower bound)

4. Step four we update the memory if the generated vector is better than worst vector in the harmony memory, based on the objective function.
5. Step five: we repeat step 3&4 until we face the stopping criteria.

B. Previous Variants of HS

Too many researches tried to find a solution for selecting the right value of the HS parameters like Mahdavi[10], Omran[1],but in this work we tried new idea by using another meta-heuristic algorithm, which is known as Jaya algorithm , to do the selection of the HMCR value for the HS algorithm.

Mahdavi [10] presented an improved harmony search algorithm (IHS). This new algorithm is to automatically modify the value of PAR & BW through every iteration.

The update of PAR and BW is by using these formulas:

PAR (t) = PAR_{min} + (($PAR_{max} - PAR_{min}$)/NI)*t.

BW (t) = bw_{max} (ln (bw_{min}/bw_{max})/NI)* t.

Meanwhile t is the number of generation, PAR_{max} the maximum pitch adjustment rate and PAR_{min} the minimum pitch adjustment rate. BW (t) is the generation bandwidth. The issue with this formulas is the need to dictate the value of bw_{min} and bw_{max}, which are hard to predict and problem conditioned.

Meanwhile the Global best harmony search(GHS)[1] aimed to enhance the performance of HS by utilizing the intelligence of swarm from the Particle Swarm Optimization[11]. GHS altered the PAR in the IHS, and the other parts are remain the same, the new PAR step as follow:

> While (i<N)
>> If (random value between (0&1) ≤ HMCR
>>> $x_i{}^\grave{} = x_i$
>>> If (random value between (0&1) ≤ PAR (t)
>>>> $x_i{}^\grave{}= x_k$best , best indicate the index of best harmony in the HM, k=(0~HMS)
>> End if
>> Else
>>> $x_i{}^\grave{}$ = lower bound + r * (upper bound – lower bound)

Combining two algorithms to enhance its efficiency, like what we have done in this work called Hybridization.

3. POPOSED METHOD

A. Hybridization

The idea of hybrid algorithm has been widely used, and to create a hybrid algorithm we need to combine two algorithms together to get new algorithm. The new algorithm will combine the strength characteristics of both algorithms to accomplish the work [10], and this is what we tried to do in this work.

B. Jaya Algorithm

Jaya Algorithm is new meta-heuristic algorithm, that created by R. Venkata Rao and give good results and consider as very low parameters metaheuristic as the author shows in his paper [12].

C. Our New Variant of HS

Most of the previous HS variants tried to improve the original HS efficiency by finding the best value of its constant variables such as HMCR and PAR, because it's affecting the whole performance of the algorithm in finding the best optimal value, and in this work we combined the Jaya algorithm with HS to tune the HMCR value to get a better results than the previous variants.

4. EXPERIMENTS ON UNCONSTRAINED BENCHMARK PROBLEMS

Several functions taken from literature mainly from [1] and [13], these objective function exercised to manifest the effectiveness of the proposed algorithm. Al the exercised functions have been used before by the original HS or GHS, and we compare the result to present soundness and performance of this work. For the comparison with HS we used some of the objectives function that used by Geem[13] to present that we get a good results. For the HMS we used the same value (20) for all the tests similar to the original HS, and for the PAR = 0.35 also similar to the value used by Geem, and HMCR we used variable values that generated by our hybrid algorithm, but the initial value we used for HMCR = 0.99.

A. HS vs Our algorithm

The Table 1 present the mathematical functions minimization we used in our comparison, to prove the powerful and effectiveness of our work. The table 1 show the function name, formula, used boundaries, and the optimal values of this function. The Table 2 show the number of iteration and the best obtained result of the objective function in both the HS algorithm and our hybrid algorithm. As shown in the table below we can see that we have the less number of iteration number of our algorithm than the HS, and we get the most optimal value that's similar HS. The table three compare between the GHS and HIS algorithm with our hybrid algorithm which presents a competitive results.

Table 1: Unconstrained function minimization

Functions name	Formula	Boundaries	Optimal solutions
Rosenbrock function	$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$	(-10 , 10)	F(x) = 0.0
Goldstein and Price function I	Goldstein and Price function I (with four local minima) $f(x) = \{1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\}$ $\times \{30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\}.$	(-5 , 5)	F(x) = 3.0
Goldstein and Price function II	$f(x) = \exp\left\{\frac{1}{2}(x_1^2 + x_2^2 - 25)^2\right\} + \sin^4(4x_1 - 3x_2) + \frac{1}{2}(2x_1 + x_2 - 10)^2.$	(-5 , 5)	F(x) = 1.0
Easton and Fenton function	$f(x) = \left\{12 + x_1^2 + \frac{1 + x_2^2}{x_1^2} + \frac{x_1^2 x_2^2 + 100}{(x_1 x_2)^4}\right\}\left(\frac{1}{10}\right)$	(0 , 10)	F(x) = 1.74
Schwefel's Problem 2.22	$f(x) = \sum_{i=1}^{N_d} |x_i| + \prod_{i=1}^{N_d} |x_i|,$	(-10,10)	F(x) = 0.0
Rastrigin Function	$f(x) = \sum_{i=1}^{N_d} (x_i^2 - 10\cos(2\pi x_i) + 10),$	(-5.12 , 5.12)	F(x) = 0.0
Ackley's	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{N_d} x_i^2}\right) - \exp\left(\frac{1}{30}\sum_{i=1}^{N_d}\cos(2\pi x_i)\right) + 20 + e,$	(-32,32)	F(x) = 0.0
Griewank	$f(x) = \frac{1}{4000}\sum_{i=1}^{N_d} x_i^2 - \prod_{i=1}^{N_d}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$	(-600,600)	F(x) = 0.0

Table 2: Optimum scores of testing unconstrained function minimization cases gained from our algorithm and HS algorithm

Function Name	Optimum Solutions	Number of Iteration HS	Number of Iteration Our Algorithm	Best result of HS	Best result of Our Algorithm
Rosenbrock function	f(x) = 0.0	50,000 iteration	25000 iteration	f(x)= 5.6843418860E-10	f(x) = 3.369536694846016E-10
Goldstein and Price function I	f(x) = 3.0	40,000 iteration	20,000 iteration	f(x)=3.000,000,000	f(x) = 3.0000000000200746
Goldstein and Price function II	f(x) = 1.0	45,000 iteration	20000 iteration	f(x)= 1.000,000,000	f(x) = 1.000,000,0000998082
Easton and Fenton function	f(x) = 1.74	800 iteration	400 iteration	f(x)= 1.74415	f(x) = 1.74415

Table 3 Optimum scores of testing unconstrained function minimization cases gained from our algorithm and GHS & IHS algorithms

Function Name	Optimum Solutions	Mean result of IHS	Mean result of GHS	Mean result of Our Algorithm
Rosenbrock function	$f(x) = 0$	$f(x)= 624.323216$	$f(x)= 49.669203$	$f(x) = 37.65528767172753$
Schwefel's Problem 2.22	$f(x) = 0$	$f(x)= 1.097325$	$f(x)= 0.072815$	$f(x) = 0.188365882169454$
Rastrigin Function	$f(x) = 0$	$f(x)= 3.499144$	$f(x)= 0.008629$	$f(x) = 0.017937494096941838$
Ackley's	$f(x) = 0$	$f(x)= 1.893394$	$f(x)= 0.020909$	$f(x) = 0.007679186329152987$
Griewank	$f(x)=0$	$f(x)= 1.120992$	$f(x)= 0.102407$	$f(x)= 0. 10354414611249534$

5. CONCLUSION

In this paper we presented new variant of HS, which apply the hybridization of two algorithm, HS and JAYA to solve the issue of choosing HS variable (HMCR) and we test it using minimization functions, and compare it with original HS, IHS and GHS, and the results show the robustness and effectiveness of the new algorithm, because we get a similar results with less number of running of the algorithm in contrast with HS and competitive results of GHS and IHS.

REFERENCES

1. Omran, M.G.H. and M. Mahdavi, *Global-best harmony search.* Applied Mathematics and Computation, 2008. **198**(2): p. 643-656.
2. Rothlauf, F., *Design of modern heuristics: principles and application*. 2011: Springer Science & Business Media.
3. Monteiro, M.S.R., *Ant colony optimization algorithms to solve nonlinear network flow problems*. 2012, Ph. D. thesis, Faculdade de Economia da Universidade do Porto, Porto, Portugal.
4. Sörensen, K., *Metaheuristics—the metaphor exposed.* International Transactions in Operational Research, 2015. **22**(1): p. 3-18.
5. Huang, L., et al., *Particle Swarm Optimization for Traveling Salesman Problems [J].* Acta Scientiarium Naturalium Universitatis Jilinensis, 2003. **4**: p. 012.
6. Gao, X., et al., *Harmony search method: theory and applications.* Computational intelligence and neuroscience, 2015. **2015**: p. 39.
7. Geem, Z.W. and J.H. Kim, *A New Heuristic Optimization Algorithm: Harmony Search.* Simulation, 2001. **76**(2): p. 60-68.
8. Ashrafi, S.M. and A.B. Dariane, *Performance evaluation of an improved harmony search algorithm for numerical optimization: Melody Search (MS).* Engineering Applications of Artificial Intelligence, 2013. **26**(4): p. 1301-1321.
9. Geem, Z.W., J.H. Kim, and G. Loganathan, *A new heuristic optimization algorithm: harmony search.* Simulation, 2001. **76**(2): p. 60-68.
10. Mahdavi, M., M. Fesanghary, and E. Damangir, *An improved harmony search algorithm for solving optimization problems.* Applied Mathematics and Computation, 2007. **188**(2): p. 1567-1579.
11. Eberhart, R.C. and J. Kennedy. *A new optimizer using particle swarm theory*. in *Proceedings of the sixth international symposium on micro machine and human science*. 1995. New York, NY.
12. Rao, R., *Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems.* International Journal of Industrial Engineering Computations, 2016. **7**(1): p. 19-34.
13. Lee, K.S. and Z.W. Geem, *A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice.* Computer methods in applied mechanics and engineering, 2005. **194**(36): p. 3902-3933.