A New Fuzzy Based Piecewise Opposition Harmony Search

A thesis submitted in partial fulfilment of the requirement for the
Degree of Master of Computer Application
of
Jadavpur University

By
Shreya Bakshi
Registration No: 133663 of 2015-16
Examination Roll No: MCA 186001

Under the guidance of

Dr. Nibaran Das
Assistant Professor
Department of Computer Science and Engineering
Jadavpur University, Kolkata-700032
India

May, 2018

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

CERTIFICATE OF RECOMMENDATION

This is to certify that the thesis entitled "**A New Fuzzy Based Piecewise Opposition Harmony Search**" has been satisfactorily completed by Shreya Bakshi (Registration No. **133663 of 2015-16,** Examination Roll No **MCA 186001**). It is a piece of work that is carried out under the guidance and supervision and be accepted for completing my degree of Master of Computer Application, Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jadavpur University, Kolkata.

Dr. Nibaran Das (Thesis Supervisor)
Assistant Professor
Department of Computer Science and Engineering
Jadavpur University, Kolkata-700032

Countersigned

Prof. Ujjawal Maulik
Head, Department of Computer Science and Engineering
Jadavpur University, Kolkata-700032

Prof. Chiranjib Bhattacharjee
Dean, Faculty of Engineering and Technology
Jadavpur University, Kolkata-700032

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

CERTIFICATE OF APPROVAL

This is to certify that the thesis entitled "**A New Fuzzy Based Piecewise Opposition Harmony Search**" has been satisfactorily completed by Shreya Bakshi (Registration No. **133663 of 2015-16,** Examination Roll No. **MCA 186001**) that is carried out under the guidance and supervision. It is understood that by this approval the undersigned do not necessarily approve any statement made, opinion expressed but approve the thesis only for the purpose for which it has been submitted.

Signature of the Examiner
Date:

Signature of the Supervisor
Date:

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

DECLARATION OF ORIGINALITY COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that this thesis entitled "**A New Fuzzy Based Piecewise Opposition Harmony Search**" contains literature survey and original research work by the undersigned candidate, as part of her Degree of Master of Computer application.
All information in this document has been obtained and presented according to the academic rules and ethical conduct.
I also declare that as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Shreya Bakshi
Registration No: **133663 of 2015-16**
Examination Roll No: **MCA 186001**

Thesis Title: A New Fuzzy Based Piecewise Opposition Harmony Search

Signature:

____

Date:

____

ACKNOWLEDGEMENT

First of all, I would like to thank God for showering me to the earth with full of energy, strength and potential for completing my Masters journey from such a wonderful place and completing my thesis work satisfactorily.

Next, I am very grateful to have a thesis guide **Dr. Nibaran Das**, Department of Computer Science Engineering, **Jadavpur University**, Kolkata for his invaluable guidance, constant motivation and inspiration during my whole work. I am also thankful to Ms. **Piyali Chandra**, my senior for helping me to develop my thesis work.

I am highly thankful to Jadavpur University for providing me the opportunity to work in such a nice working environment and provide a nice infrastructure and services from the **'Centre for Microprocessor Application for Training, Education and Research**' also. I would like to thanks all my labmates to give me constant motivation and mental strength and have a wonderful journey in these six months.

Last but not the least, I extend my gratitude to my parents for their love and support, giving me the mental strength. Also, I am thankful to my friends, my classmates, and seniors for encouraging me always to my work and motivate me also.

Shreya Bakshi

Registration No: **133663 of 2015-2016**

Examination Roll No: **MCA 186001**

Master of Computer Application

Department of Computer Science and Engineering

Jadavpur University

TABLE OF CONTENTS

1 Introduction 9

1.1 PARAMETERS USED IN HARMONY SEARCH: 9

1.1.1 HMS (Harmony Memory Size): 9

1.1.2 PAR (Pitch Adjustment Rate): 10

1.1.3 BW (Bandwidth): 10

1.2 IMPORTANCE OF HARMONY SEARCH: 10

1.3 ADVANTAGES OF HARMONY SEARCH: 12

1.4 DRAWBACKS OF HARMONY SEARCH: 12

1.5 OVERVIEW OF THE THESIS: 12

2 Variations of Harmony Search 14

2.1 BASIC HARMONY SEARCH ALGORITHM: 14

2.2 TYPES OF HARMONY SEARCH: 15

2.2.1 IHS (Improved Harmony Search): 15

2.2.2 EHS (Enhanced Harmony Search): 15

2.2.3 OHS (Opposition based Harmony Search): 16

2.2.4 GDHS (Global Dynamic Harmony Search): 16

2.2.5 SAHS (Self Adaptive Harmony Search): 17

2.2.6 GHS (Global Best Harmony Search): 17

2.2.7 NGHS (Novel Global Best Harmony Search): 18

2.2.8 IGHS (Improved Global Best Harmony Search): 18

2.2.9 LHS (Local Best Harmony Search): 19

2.3 HYBRID HS: 20

2.3.1 HS+ DMO: 20

2.3.2 HS+PSO: 20

2.4 BENCHMARK FUNCTIONS FOR OPTIMIZATION: 21

2.5 GRAPHICAL REPRESENTATION OF THE BENCHMARK FUNCTIONS: 33

3 Proposed Methodology..36

3.1 FUZZY BASED PIECEWISE OPPOSITION HARMONY SEARCH ALGORITHM:..........37

3.2 EXPERIMENTS AND RESULTS: 41

3.2.1 Parameters used in the present work: 41

3.2.2 Comparison between the previous hs and the fuzzy based pohs: 43

3.2.3 Success Rate and Absolute Errors over the Benchmark Functions:.........45

3.3 GRAPHICAL IMPLEMENTATIONS OF SOME BENCHMARK FUNCTIONS:................46

3.4 ANALYSIS: 50

4 Conclusion and Future work...51

4.1 CONCLUSION: 51

4.2 FUTURE WORK: 51

References...53

CHAPTER ONE

INTRODUCTION

Harmony Search is a metaheuristic[1] optimization strategy based on the music. In the music, the pitch of each musical instrument determines its quality, similarly here also by the fitness value determines the quality of the decision variables. A musician tries to find the best harmony altogether, likewise in the Harmony Search, we also try to find the best global optimal value (minimum or maximum) value[2]. Here each decision variable generates a value. It was proposed by Zong Woo Geem, Joong Hoon Kim, and G. V. Loganathan in 2001[3]. It was inspired by the improvisation of the jazz musicians.

1.1 PARAMETERS USED IN HARMONY SEARCH:

Harmony Search uses many parameters for its input. The basic components[4][5] of the Harmony Search is defined as follows:

1.1.1 HMS (HARMONY MEMORY SIZE):

Generally, a random vector (x^1,x^2,...,x^{hms}) is initialized as many as Harmony memory size is defined. Generally, HMS[6] varies between 1 to 100[7]. For the convenience of the implementation of the program, a rate of value is chosen from the Harmony Memory. It is ideally set to value between 0.7 to 0.99. The lower value of it defines that only a few no of best harmony memory is chosen for the next iteration which leads the algorithm to be slow. The higher value of the HMCR(Harmony Memory Consideration Rate) result into a lack of diversity other than those belonging to HM (Harmony Memory) are not explored efficiently. That is why its value lies within that particular range. The Harmony memory is initialized in the following way:

$$HM=\begin{pmatrix} x_1^1 & x_2^1 & \cdots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_{N-1}^2 & x_N^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \cdots & x_{N-1}^{HMS-1} & x_N^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_{N-1}^{HMS} & x_N^{HMS} \end{pmatrix}$$

Figure 1: **Initializing the harmony memory**[7]

1.1.2 PAR (Pitch Adjustment Rate):

Pitch Adjustment Rate is the rate of choosing a neighbouring value which helps in the Harmy Search for further mutate a solution. It generally varies from 0.1 to 0.5. A low value of PAR may fail the convergence of the solution in the Harmony Search due to the limited exploration of the search space while a higher value of PAR may cause the solution to disperse like a random search.

1.1.3 BW (Bandwidth):

It is the amount of maximum change in pitch adjustment. This can be (0.01 × allowed range) to (0.001 × allowed range). It possible to vary it also as search progress also.

Besides these, there are also other parameters like Dimension, JR(Jumping Rate) etc.

1.2 Importance of Harmony Search:

Day by day, along with the technology is developing, our demand is also increasing. Also, we need to do every work in less amount of time. Here optimization comes.

Harmony Search is one of the popular optimization techniques which has solved already a plenty amount of problems.

Harmony Search may be a music-based algorithm, but as long as it is being famous throughout the world, the researchers started to work on this. So besides the music, it has a wide variety of applications till now:

1. Harmony Search is applied in the School Bus Routing Problem (SBRP) and tries to optimize the number of buses and the travel time also with two major constraints (bus capacity and the time window)[8] so that the SBRP aims to provide efficient transport for the students without losing their business.

2. Harmony Search was inspired by the Soft Computing algorithm. Cluster Analysis[9][10], one of the popular data mining technique is also a greatly affected by the Harmony search. K -means is used to be the popular clustering method for its simplicity and high speed in large clustering datasets.

3. Low connectivity is one of the major problems in non-uniform density WSNs(Wireless Sensor Networks)[11]. Harmony Search is applied in this field also to overcome this issue. Network connectivity with and without the Harmony Search was calculated and at last, it ensured to improve the performance.

Apart from that, Harmony search has also a lot of applications in the following areas: structural design[12] ,dam scheduling[13] ,RNA structure prediction [14], soil stability analysis, structural design, groundwater modeling, energy system dispatch, medical physics, tour planning, solving sudoku puzzles[15], transportation energy modeling, heat exchanger design, satellite heat pipe design, optimal power flow, container-storage problem etc. Besides these, in NP-hard combinatorial optimization problems, train neural network[16], manufacturing scheduling, for reliability problems[17], training neural network[18], numerical optimization[19] nurse scheduling problem Harmony search is used also.

1.3 ADVANTAGES OF HARMONY SEARCH:

Harmony Search takes an input as random numbers this is really an advantage. Apart from that, it uses fewer numbers of adjustable parameters throughout the algorithm. The implementation of the algorithm is also easy.

Also, Harmony Search has quick convergence too. Besides being a music based metaheuristic algorithm, it can be applied over a huge number of fields from engineering to real life problems. One of the big advantages of Harmony Searches is that the functions used in it are derivative free.

1.4 DRAWBACKS OF HARMONY SEARCH:

But it has some drawbacks too. Sometimes its converging rate becomes too slow and premature also. And before starting the algorithm, it has no sufficient information. So sometimes it may be stuck to the local optimum also.

To overcome these problems, researchers are trying to improve it in various ways, sometimes adjusting the parameters, sometimes merging another kind of algorithm with Harmony Search a new version of Harmony Search has been improved. In most of the cases researchers have proposed the methodology by adjusting the parameters like HMCR, PAR etc.

1.5 OVERVIEW OF THE THESIS:

The first chapter ‘Introduction’ goes through the brief concept of the Harmony Search, what type of parameters are used in the Harmony Search. Also it describes the applications of it, advantages and drawbacks also. In the chapter ‘Variations of Harmony Search’, the variety of Harmony Searches have been drawn. Also, there are some other algorithms like GA (Genetic Algorithm), PSO (Particle Swarm Optimization) which have been merged with the Harmony Search and a new type of Hybrid algorithms have been improved. Sometimes also adjusting the parameter in a

logical way the algorithm has developed also. It also describes about a lot of benchmark functions which may be used for the optimization problem. In the third

The first chapter ‘Introduction’ goes through the brief concept of the Harmony Search, what type of parameters are used in the Harmony Search. Also it describes the applications of it, advantages and drawbacks also. In the chapter ‘Variations of Harmony Search’, the variety of Harmony Searches have been drawn. Also, there are some other algorithms like GA (Genetic Algorithm), PSO (Particle Swarm Optimization) which have been merged with the Harmony Search and a new type of Hybrid algorithms have been improved. Sometimes also adjusting the parameter in a logical way the algorithm has developed also. It also describes about a lot of benchmark functions which may be used for the optimization problem. In the third chapter ‘Proposed Methodology' we have proposed a new fuzzy-based accelerating factor and imposed it on the POHS and observed the better performance in most of the cases. At last ‘Conclusion' gives the idea we get from our experiments.

CHAPTER TWO

VARIATIONS OF HARMONY SEARCH

2.1 BASIC HARMONY SEARCH ALGORITHM:

Harmony Search is one of the popular search fields in optimization. Its main advantage is that it uses less no of parameters in the algorithm. Researchers are working on till now to improve it by so many techniques. Here is a brief overview of the basic Harmony Search algorithm.

Algorithm 1: Basic Harmony Search algorithm[20]

Input: *Parameters like HMS (Harmony Memory Size), HMCR(Harmony Memory Consideration Rate), PAR(Pitch Adjustment Rate), BW(Bandwidth), Max_NFc(Maximum number of Iterations)*

Output: *Achieved the best fitness value*

1. Initialize the Parameters HMS, HMCR, Par, BW, Max_NFC as stated above

2. Intialize the initial population of Harmony Memory(HM) with random harmonies, say m, such that $h \in [lb_i, ub_i]$

3. If $r_1 \leq HMCR$ *where* $r_1 = rand()$ *such that* $r_1 \in U(0,1)$

3.1 Randomly select a harmony from HM, say M

3.2 If $r_2 \leq PAR$ *where* $r_2 = rand()$ *such that* $r_2 \in U(0,1)$

3.2.1 $H_{new} = H \pm r * BW$ *where r=rand() such that* $r \in (0,1)$

4. Compare the H_{new} *and the worst harmony of HM, say* H_{worst} *in terms of the fitness function* $f((H_{new}))$ *and* $f(H_{worst})$ *respectively.*

5. If H_{new} *is better than* M_{worst} *then it is replaced by* H_{new} *in the Harmony Memory.*

6. Step 3to step 5 is repeated until some termination criteria are met or the number of Iterations have been reached to the value of Max_NFC

2.2 TYPES OF HARMONY SEARCH:

Over the past few years, to improve the performance of the Harmony Search, researchers are trying applying different modification. As a result, some category of Harmony Search is published. Here are some of the reviews:

2.2.1 IHS (IMPROVED HARMONY SEARCH):

The main two parameters of the Harmony Search i.e the PAR and the BW have been modified slightly. In this case, a justification has been given that the PAR was increasing linearly and the value of BW was decreasing exponentially with respect to the number of iterations[21]. So it is modified as:

$$PAR_k = PAR_{min} + ((PAR_{max} - PAR_{min})/K) \times k$$

......................**Equation 1**

$$BW_k = BW_{max} \exp(c.k),$$

$$\text{where } c = (\ln(BW_{min}/BW_{max}))/K$$

......................**Equation 2**

Where k is the current iteration and K is the max number of iterations. Also, PAR_{max} and PAR_{min} are the maximum and minimum pitch adjustment rate respectively. BW_{max} and BW_{min} are the maximum and minimum bandwidth respectively.

2.2.2 EHS (ENHANCED HARMONY SEARCH):

In the original Harmony Search at the last step, the harmony vector is updated until the termination criteria are satisfied. But here in the EHS[22], a checking condition has been kept i.e if the objective function of the new harmony factor is better than the worst harmony memory, then the worst harmony memory is replaced by the new harmony memory. Here also the BW is being updated dynamically with the standard deviation where

$$BW = \tau\sqrt{Var(x)}$$

…………………**Equation 3**

Where τ is a constant taken as 1.17. Based on the previous work t has been chosen.

2.2.3 OHS (Opposition based Harmony Search):

Opposition based Harmony Search[23] is very popular. When there is no sufficient information about the solution, then the solution is started with a random guess. It can be started with a guess by estimating the value of $\breve{x}$ where

$$\breve{x} = a_i + b_i - x$$ ………………..**Equation 4**

In every iteration, it tries to improve to get the solution. It can be shown as below:

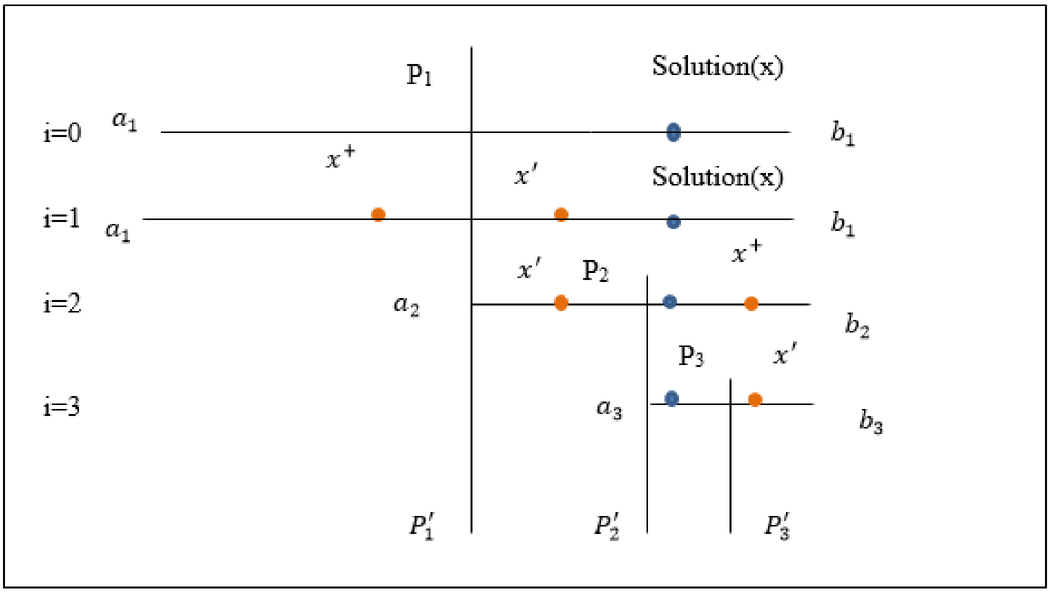


Figure 2: **Opposition based harmony Search**[20]

2.2.4 GDHS (Global Dynamic Harmony Search):

The advantage of GDHS is that there is no need to predefine any parameters. All the parameters are changed into the dynamic mode. This Harmony Search has improved a dynamic method to adjust PAR, HMCR, BW[24]. The dynamic method of tuning BW is same for GDHS and IHS. Here the domain is also dynamically changed. One more advantage of the search is faster convergence. Here HMCR and PAR are adjusted as:

$$HMCR_k = 0.9 + 0.2 \times \sqrt{\frac{(k-1)}{(K-1)} \times (1 - \frac{(k-1)}{(K-1)})}$$ ……………………Equation 5

$$PAR = 0.9 + 0.2 \times \sqrt{\frac{(k-1)}{(K-1)} \times (1 - \frac{(k-1)}{(K-1)})}$$ …………………Equation 6

2.2.5 SAHS (Self Adaptive Harmony Search):

Since the PAR and the BW have a great influence on the final solutions, therefore the value of PAR has been adjusted to provide better solutions. Actually, the parameter BW is completely replaced. The parameter PAR is adjusted as follows: given as:

$$PAR_k = PAR_{max} - (PAR_{max} - PAR_{min}) \times k/K$$ ……………..Equation 7

2.2.6 GHS (Global Best Harmony Search):

Here also the adjusting factor is the PAR. The process is same as IHS. It develops the concept of Swarm Intelligence and improves Harmony Search. Recently GHS proved that it outperformed the other Harmony Searches while tested on 10 benchmark functions[25]. One of the advantages of GHS is that it can be applied to both discrete and continuous problems. GHS also provides better results for the higher dimensional problems. Here the two parameter HMCR and HMS was investigated, it was seen that HMCR improves the performance of GHS except for the lower dimensionality problems where a small value of HMCR is recommended.

But the value of HMS should be small here. It was proved that a small constant value of PAR can improve the performance of GHS.

2.2.7 NGHS (NOVEL GLOBAL BEST HARMONY SEARCH):

A very interesting thing is that NGHS can solve also the 0-1 Knapsack problems. But here Harmony Search faces some difficulties. To improve here comes the NGHS. Some changes have been imported in the NGHS[26]. The algorithm is as follows:

Algorithm 2: A brief overview of the NGHS algorithm

---

i)PAR and HMCR are not included in the NGHS, rather a genetic mutation probability r(p_m) is included here.

ii)The improvisation step of the HM is as follows:

for each i $\in$ [1, N]do

$step_i = |x_{besti} - x_{worsti}|$ *% Calculating the adaptive step*

$x_i{}' = xi_{best} \pm r \times step_i$ *%position updating*

if $rand() \leq p_m$ *then*

$x_i' = xi_L + rand() \times (x_{iU} - x_{iL})$ *%genetic mutation*

End

End

iii)After improvisation, the worst harmony xworst is replaced by the new harmony memory x' even if x' is worse than xworst.

2.2.8 IGHS (IMPROVED GLOBAL BEST HARMONY SEARCH):

As proposed in the SAHS, the value of PAR is linearly decreasing and BW is exponentially decreasing as proposed in the IHS. After so many experiments, it works better than the GHS. Here the parameters HMCR and PAR have been improvised[27]. They are updated dynamically based on a composite function. The improvisation process is taken from [21]. Here is a small overview of the IGHS algorithm:

Algorithm 3: A brief overview of the IGHS algorithm

For j=1to D do
    If rand≤HMCR
        $x_{new,j}=x_{r,j}+Gauss(0,1)*BW_k$, *r belongs to (1,2,3,…,HMS)*
        If rand≤PAR_k *then*
            $x_{new,j}=x_{best,j}+rand()*BW_k$
        end if
    else
        $x_{new,j}=x_{j,L}+rand()*(x_{j,U}-x_{j,L})$
    end if

2.2.9 LHS (Local Best Harmony Search):

Here exploitation and exploration are required that means the algorithm should have the exploitation ability has to use all the information and try to find a search space in the neighbourhood of the global optimum. Another one is the exploration ability to find the region around the global optimum quickly

In this search, some properties, better to say three key features[21] has been proposed as follows:

i) Pitch adjustment rate
ii) Opposition based learning technique
iii) The selection mechanism is proposed in such a way that it can escape from the local optimum.

Besides all these types, there are also other types of Harmony Search like chaotic HS[28][29][30], Ant colony Algorithm [31][32][33]based on Harmony Search, Cellular HS[34], Adaptive Harmony Binary search (ABHS)[35], Intelligent Tuned Harmony Search Algorithm(ITHS)[36] etc.

2.3 HYBRID HS:

To improve the Harmony Search the parameters have been adjusted through many logics and it has been improved also. So after the original Harmony Search, its so many improved versions are available also. Likewise, to improve the performance of Harmony Search in various fields, it has been hybridized[21][37] with other algorithms also and better results have been got also rather than the previous ones. Here are some of the hybrid HS as follows:

2.3.1 HS+ DMO:

Harmony search with Differential Mutation Operator has been introduced also to improve the result of Harmony Search. An operator like differential mutation operator is replaced to adjust the value of PAR. The operation is as follows[21]:

$$x_{new} = x_{new,j} + F \times (x_{r1,j} - x_{r2,j})$$ ....**Equation 8**

Where r1,r2 belongs to (1,2,3,…,HMS) , r1≠r2 and F is the scale factor

2.3.2 HS+PSO:

IPSO (Improved Particle Swarm Optimization) is based on basically the improvement of PSO algorithm which has been done by merging the PSO with the Harmony Search. It is based on the common characteristics of the PSO[38][39] and Harmony Search algorithms. The improvement has been done as follows:

i) Initializing the parameters of Harmony Search and PSO and the parameters also.

ii) Evaluate the particles according to their fitness value and sort them in descending order.

iii) Perform Harmony Search and generate a new solution.

iv) If the new solution is better than the worst, then it worst is replaced by the new one.

v) Update particles according to the rules of PSO.

vi) Repeat from step (iii) to (vi) or some termination criteria is finished.

2.4 BENCHMARK FUNCTIONS FOR OPTIMIZATION:

Normally Harmony Search and their modifications are evaluated over a set of benchmark functions. There are a huge number of benchmark functions exists in the literature which re popularly used by the researchers. Some of the important benchmark functions which have been used for the Optimization purposes are given in the table:

Table 1: **Benchmark Functions for the Optimization problems** [40][21][1][41]

Function#	Function Name	Mathematical Expression	Range	Global Optimum
1	Sphere	$f_1 = \sum_{i=1}^{D} x_i^2$	[-100,100]^D	0
2	Schwefel 2.2	$f_2 = \sum_{i=1}^{D} \|\|x_i\| + \prod_{i=1}^{D} \|\|x_i\|\|\|$	[-100,100]^D	0
3	Schwefel 1.2	$f_3 = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)$	[-100,100]^D	0
4	Schwefel 2.21	$f_4 = \max_{D-1}\{\|x_i\|, 1 \leq i \leq D\}$	[-100,100]^D	0

[1] https://www.sfu.ca/~ssurjano/optimization.html

Function#	Function Name	Mathematical Expression	Range	Global Optimum
5	Rosenbrock	$f_5 = \sum[100(x_{i+1} - x_i)^2 + (x_{i-1})^2]$	[-30,30]^D	0
6	Step	$f_6 = \sum_{i=1}^{D}(\lvert x_i + 0.5 \rvert)^2$	[-100,100]^D	0
7	Quartic	$f_7 = \sum_{i=1}^{D} ix^4 \; + random(0,1)$	[-1.28,1.28]^D	-418.9829 D
8	Schwefel	$f_8 = \sum_{i=1}^{D} -x_i \times sin\sqrt{\lvert\lvert x_i \rvert\rvert}$	[-500,500]^D	0
9	Rastrigin	$f_9 = 10D + \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i)]$	[-5.12,5.12]^D	0
10	Ackley	$f_{10} = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$	[-32,32]^D	0
11	Griewank	$f_{11} = \frac{1}{4000}\left(\sum_{i=1}^{D} x_i^2\right) - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600,600]^D	0

Function#	Function Name	Mathematical Expression	Range	Global Optimum
12	Penalized	$f_{12} = \frac{\pi}{D}\left\{10sin^2(\pi y_1) + \sum_{i=1}^{D}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{D} u(x_{i,}10, 100, 4)y_i = 1 + \frac{1}{4}(x_i + 1)$	$[-50,50]^D$	0
13	Penalized 2	$f_{13} = 0.1\left\{sin^2(3\pi x_1)15 + \sum_{i=1}^{D-1}(x_i - 1)^2\,[1 + sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\right\} + \sum_{i-1}^{D} u(x_i, 5, 100, 4)$ Where $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a \\ 0 \quad -a \leq x_i \leq a \\ k(-x_i - a)^m\,, x_i < -a \end{cases}$	$[-50,50]^D$	0
14	Foxholes	$f_{14} = [\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}]^{-1}$	[- 65356,653	0.998

Function#	Function Name	Mathematical Expression	Range	Global Optimum
			56]^2	
15	Kowalik	$f_{15} = \sum_{i=1}^{D}\left(a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right)^2$	[-5,5]^4	3.08e-04
16	Six Hump Camelback	$f_{16} = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	[-5,5]^2	-1.316285
17	Branin	$f_{17} = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_1 + 10$	[-5,10]^2 or [0,15]^2	0.398
18	Goldstein Price	$f_{18} = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	[-5,5]^2	3
19	Hartman 3	$f_{19} = -\sum_{i=1}^{4} c_i \exp[-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2]$	[0,1]^3	-3.86278
20	Hartman 6	$f_{20} = -\sum_{i=1}^{4} c_i \exp[-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2]$	[-10,10]^2	-3.32
21	Shubert	$f_{21} = \prod_{i=1}^{2}\sum_{i=1}^{5} jcos(j+1)(x_i + j)$	[–10,10]2	−186.7309

Function#	Function Name	Mathematical Expression	Range	Global Optimum
22	Goldstein Price2	$$f_{22} = \exp\left(0.5(x_1^2 + x_2^2 - 25)^2\right) + \sin^4(4x_1 - 3x_2) + 0.5(2x_1 + x_2 - 10)^2$$	$[-5,5]^2$	1
23	Schaffer	$$f_{23} = \frac{\sin^2\sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)^2]^2} - 0.5$$	$[-100,100]^2$	-1
24	Schafferf 7	$$f_{24} = \sum_{i=1}^{n-1}[(x_i^2 + x_{i+1}^2)^{0.25}(\sin(50(x_i^2 + x_{i+1}^2)^{0.1})^2 + 1)]$$	$[-100,100]^D$	0
25	Zakharov	$$f_{25} = \sum_{i=1}^{2} x_i^2 + (\sum_{i=1}^{2} 0.5ix_i)^2 + (\sum_{i=1}^{2} 0.5ix_i)^4$$	$[-100,100]^D$	0
26	Shifted Rotated High Conditioned Elliptic Function	$$f_{26} = \sum_{i=1}^{N}(10^6)^{\frac{i-1}{N-1}}{z_i}^2 - 450, z = x - o$$	$[-100,100]^D$	-450
27	Shifted Schwefel's Problem 1.2 with	$$f_{27} = \sum_{i=1}^{N}\left(\sum_{j=1}^{i} z_j\right)^2 (1 + 0.4 \times \lvert N(0,1)\rvert) - 450, z = x - o$$	$[-100,100]^D$	-450

Function#	Function Name	Mathematical Expression	Range	Global Optimum
	Noise			
28	Stepint	$f_{28} = 25 + \sum_{i=1}^{5} [x_i]$	[-5.12,5.12]^D	0
29	SumSquares	$f_{29} = \sum_{i=1}^{D} i x_i^2$	[-10,10]^D	0
30	Beale	$f_{30} = (2.25 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$	[-4.5,4.5]^5	0
31	Easom	$f_{31} = -\cos(x_1)\cos(x_2)\exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$	[-100,100]^2	-1
32	Matyas	$f_{32} = 0.26(x_1^2 x_2^2) - 0.48 x_1 x_2$	[-10,10]^2	0
33	Colville	$f_{33} = 100(x_1^2 - x_2) + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8((x_2 - 1)(x_4 - 1))$	[-10,10]^4	0
34	Trid	$f_{34} = \sum_{i=1}^{D} (x_i - 1)^2 - \sum_{i=1}^{D} x_i x_{i-1}$	[-D^2,D^2]^6	-50

Function#	Function Name	Mathematical Expression	Range	Global Optimum
35	Powell	$f_{35} = \sum_{i=1}^{D/k}[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$	$[-4,5]^{24}$	0
36	Bohachevsky 1	$f_{36} = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	$[-100,100]^2$	0
37	Bohachevsky 2	$f_{37} = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3$	$[-100,100]^2$	0
38	Booth	$f_{38} = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	$[-10,10]^2$	0
39	Michalewicz 2	$f_{39} = -\sum_{i=1}^{2}\sin(x_i)\sin(\frac{ix_2^2}{\pi})^{20}$	$[0,\pi]^2$	-1.8013
40	Michalewicz 5	$f_{40} = -\sum_{i=1}^{5}\sin(x_i)\sin(\frac{ix_2^2}{\pi})^{20}$	$[0,\pi]^5$	-4.687658
41	Michalewicz 10	$f_{41} = -\sum_{i=1}^{10}\sin(x_i)\sin(\frac{ix_2^2}{\pi})^{20}$	$[0,\pi]^{10}$	-9.66015
42	Perm	$f_{42} = \sum_{k=1}^{D}[\sum_{i=1}^{D}(i^K + b)((\frac{x_i}{i})^k - 1)]^2$	$[-D,D]^4$	0
43	Powersum	$f_{43} = \sum_{k=1}^{D}[(\sum_{i=1}^{D} x_i^k) - b_k]^2$	$[0,D]^4$	0
44	Dixon-Price	$f_{44} = (x_1 - 1)^2 + \sum_{i=1}^{D} i(2x_i^2 - x_{i-1})^2$	$[-10,10]^D$	0

Function#	Function Name	Mathematical Expression	Range	Global Optimum
45	Rotated Hyper Ellipsoid Function	$f_{45} = \sum_{i=1}^{N} (\sum_{j=1}^{i} x_j)^2$	$[-100,100]^D$	0
46	Weierstrass Function	$f_{46} = \sum_{i=1}^{N} (\sum_{k=0}^{kmax} [a^k \cos(2\pi b^k (x_i + 0.5))]) = N \sum_{k=0}^{kmax} [a^k \cos(2\pi b^k \times 0.5)]$ $where\ a = 0.5, b = 3, kmax = 20$	$[-0.5,0.5]^D$	0
47	Cross-in tray Function	$f_{47} = -0.0001(\lvert\sin(x)\sin(y)\exp(\lvert 100 - \frac{\sqrt{x^2+y^2}}{\pi}\rvert)\rvert + 1)^{0.1}$	$[-10,10]^2$	-2.06261218
48	Keane Function	$f_{48} = -\frac{sin^2(x-y)sin^2(x-y)}{\sqrt{x^2+y^2}}$	$[0,10]^2$	0.673667521146855
49	Leon Function	$f_{49} = 100(y - x^3)^2 + (1-x)^2$	$[0,10]^2$	0
50	Salomon Function	$f_{50} = 1 - \cos\left(2\pi \sqrt{\sum_{i=1}^{D} x_i^2}\right) + 0.1 \sqrt{\sum_{i=1}^{D} x_i^2}$	$[-100,100]^D$	0

Function#	Function Name	Mathematical Expression	Range	Global Optimum
51	Shubert 3 Function	$f_{51} = \sum_{i=1}^{5}\sum_{j=1}^{5} jsin((j+1)x_i + j)$	$[-10,10]^D$	-29.6733337
52	Shubert 4 Function	$f_{52} = \sum_{i=1}^{5}\sum_{j=1}^{5} jcos((j+1)x_i + j)$	$[-10,10]^D$	-25.740858
53	Ackley N.2Function	$f_{53} = -200e^{-0.2\sqrt{x^2+y^2}}$	$[-32,32]^2$	-200
54	Ackley N.3Function	$f_{54} = -200e^{-0.2\sqrt{x^2+y^2}} + 5e^{\cos(3x)+\sin(3y)}$	$[-32,32]^2$	-195.62902823841 9
55	Levy FUnction N.13	$f_{55} = sin^2(3\pi x_1) + (x_1-1)^2[1 + sin^2(3\pi x_2)] + (x_2-1)^2[1 + sin^2(2\pi x_2)]$	$[-10,10]^2$	0
56	Holder Table Function	$f_{56} = -\lvert \sin(x_1)\cos(x_2)\exp(\lvert 1 - \frac{\sqrt{x_1^2+x_2^2}}{\pi}\rvert)\rvert$	$[-10,10]^2$	-19.2085

Function#	Function Name	Mathematical Expression	Range	Global Optimum
57	Egg Holder Function	$f_{57} = -(x_2+47)\sin\left(\sqrt{\left\lvert x_{2+\frac{x_1}{2}+47}\right\rvert}\right) - x_1\sin(\sqrt{\lvert x_1-(x_2+47)\rvert})$	[-512,512]^2	959.6407
58	Drop Wave Function	$f_{58} = -\frac{1+\cos(12\sqrt{x_1^2+x_2^2})}{0.5(x_1^2+x_2^2)+2}$	[-5.12,5.12]^2	-1
59	Bird Function	$f_{59} = \sin(x)\, e^{(1-\cos(y))^2} + \cos(y)\, e^{(1-\sin(x))^2} + (x-y)^2$	[-2π,2π]^2	-106.764537
60	Exponential Function	$f_{60} = -\exp(0.5\sum_{i=1}^{D} x_i^2)$	[-1,1]^D	0
61	Alpine N.1 Function	$f_{61} = \sum_{i=1}^{D} \lvert x_i\sin(x_i)+0.1x_i\rvert$	[0,10]^D	0
62	Alpine N.2 Function	$f_{62} = \prod_{i=1}^{D} \sqrt{x_i}\sin(x_i)$	[0,10]^D	2.808^D
63	Bartless Conn Function	$f_{63} = \lvert x^2+y^2+xy\rvert + \lvert\sin(x)\rvert + \lvert\cos(y)\rvert$	[-500,500]	1
64	Gramacy		[0.5,2.5]	-

Function#	Function Name	Mathematical Expression	Range	Global Optimum
	& Lee Function	$f_{64} = \frac{\sin(10\pi x)}{2x} + (x-1)^4$		0.869011 1349895 00
65	Qing Function	$f_{65} = \sum_{i=1}^{D} (x^2 - i)^2$	$[-500,500]^D$	0
66	Wolfe Function	$f_{66} = \frac{4}{3}(x^2 + y^2 - xy)^{0.75} + z$	$[0,2]^3$	0
67	Deckkerts Arts Function	$f_{67} = 10^5 x^2 + y^2 - (x^2 + y^2)^2 + 10^{-5}(x^2 + y^2)^4$	$[-20,20]^D$	-24771.09375
68	Himmelblau Function	$f_{68} = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$	$[-6,6]^2$	0
69	Styblinski-Tank Function	$f_{69} = \frac{1}{2}\sum_{i=1}^{D} (x_i^4 - 16x_i^2 + 5x_i)$	$[-5,5]^D$	-39.16599 D
70	Sum-Squares Function	$f_{70} = \sum_{i=1}^{D} i x_i^2$	$[-10,10]^D$	0
71	Egg-Crate Function	$f_{71} = x^2 + y^2 + 25(sin^2(x) + sin^2(y))$	$[-5,5]^2$	0
72	McCormick		$x \in [-1.5,4]$	-1.9133

Function#	Function Name	Mathematical Expression	Range	Global Optimum
	Function	$f_{72} = \sin(x+y) + (x-y)^2 - 1.5x + 2.5y + 1$	$y \in [-3,3]$	
73	Bartels Conn Function	$f_{73} = \lvert x^2 + y^2 + xy \rvert + \lvert \sin(x) \rvert + \lvert \cos(y) \rvert$	$[-500,500]^2$	1
74	Adjiman Function	$f_{74} = \cos(x)\sin(y) - \frac{x}{y^2+1}$	$x \in [-1,2]$ $y \in [-1,1]$	-2.02181
75	Xin-She-Yang N.2 Function	$f_{75} = (\sum_{i=1}^{D} \lvert x_i \rvert)\exp(-\sum_{i=1}^{D} \sin(x_i^2))$	$[-2\pi,2\pi]^D$	0
76	Xin-She-Yang N.4 Function	$f_{76} = (\sum_{i=1}^{D} sin^2(x_i) - e^{-\sum_{i=1}^{D} x_i^2})e^{-\sum_{i=1}^{D} sin^2\sqrt{\lvert x_i \rvert}}$	$[-10,10]^D$	-1
77	Sum of different power function	$f_{77} = \sum_{i=1}^{D-1} \lvert x_i \rvert^{i+1}$	$[-1,1]^D$	0
78	Levy Function	$f_{78} = 0.1(sin(23\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 \ [1 + sin(23\pi x_{i+1})] + (x_n - 1)^2[1 + sin(23\pi x_n)])$	$[-5,5]^D$	0
79	Rectangular Step Function	$f_{79} = \begin{cases} 1 & if\ \lvert x \rvert < -1/2 \\ 1 & if\ \lvert x \rvert > 1/2 \\ 0 & otherwise \end{cases}$	$[-50,50]^D$	0

2.5 GRAPHICAL REPRESENTATION OF THE BENCHMARK FUNCTIONS:

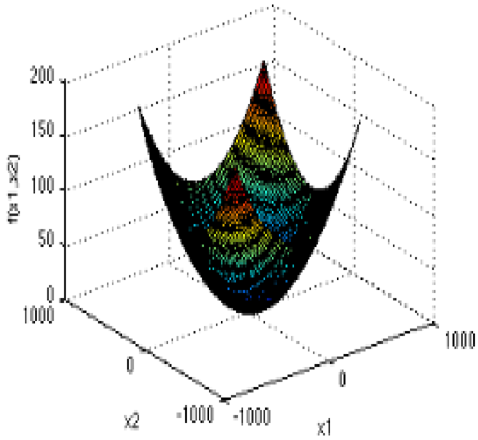


Figure 3: **Sphere Function**

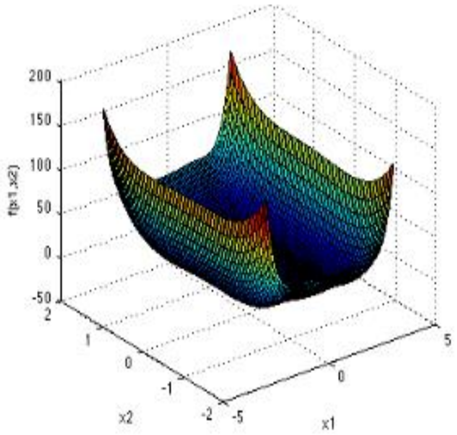


Figure 4: **Six Hump Camel Back Function**

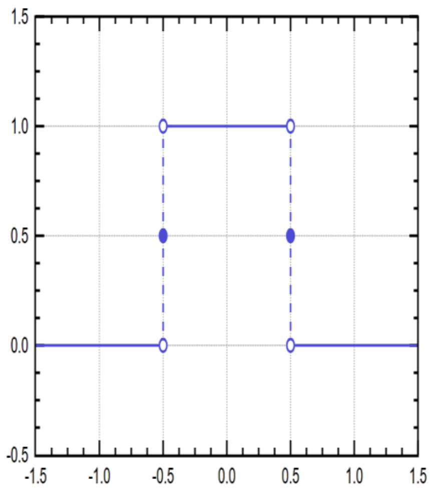


Figure 5: **Rectangular Step Function**

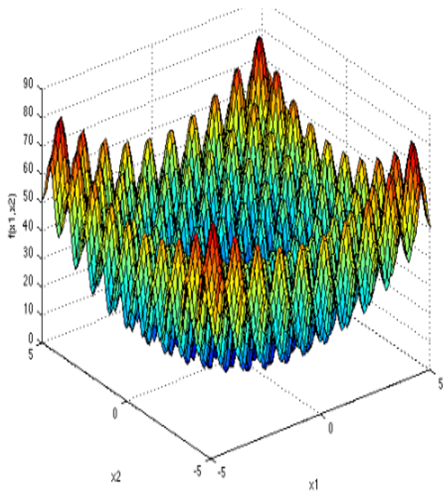


Figure 6: **Rastrigin Function**

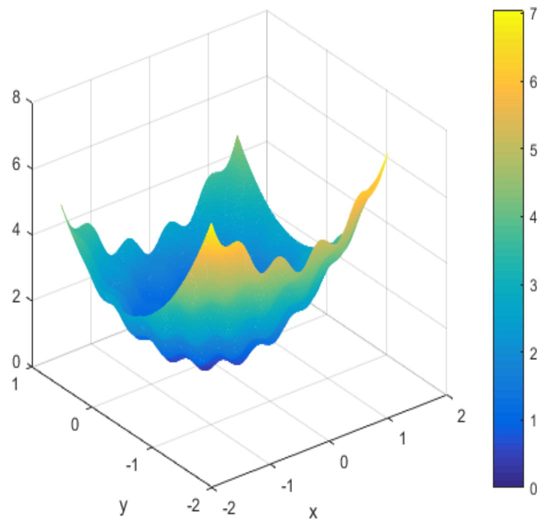


Figure 7: **Bohachevsky's 2 Function**

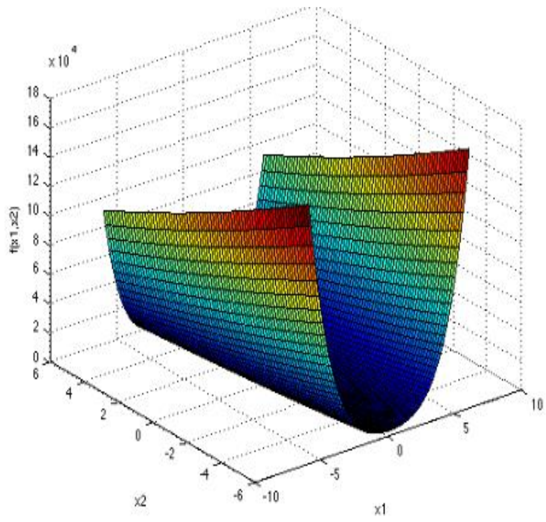


Figure 8: **Rosenbrock's Function**

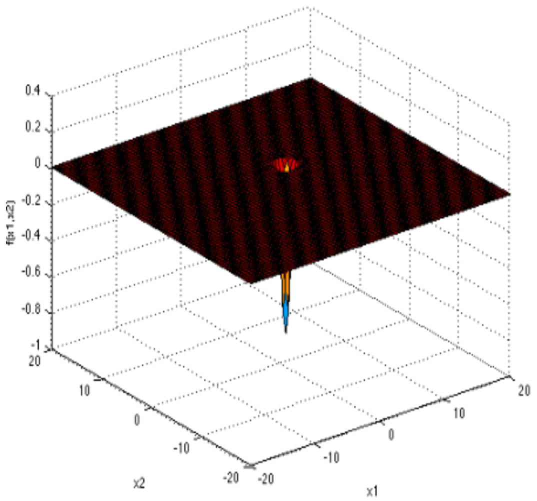


Figure 9: **Easom Function**

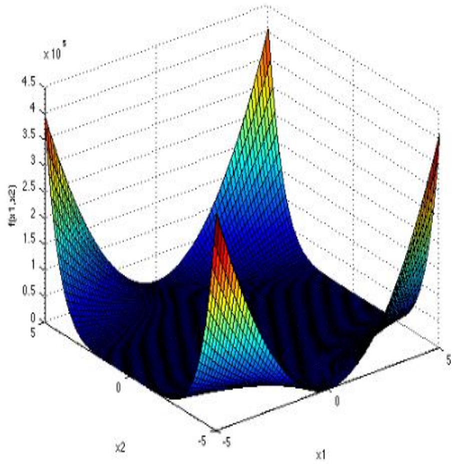


Figure 10: **Beale Function**

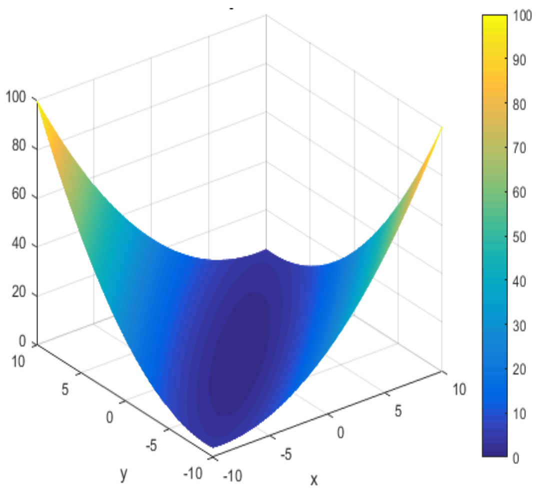


Figure 11: **Matyas Function**

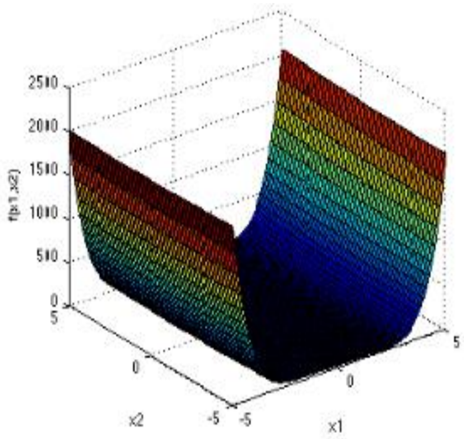


Figure 12: **Three Hump Camel Back Function**

CHAPTER THREE

PROPOSED METHODOLOGY

In the previous chapter of this thesis details of the Harmony Search has been fully described. To improve the performance of the Harmony search its various types are also portrayed. POHS (Piecewise Opposition Harmony Search) is one type of Harmony Search. Here we have also tried to propose a method to improve the performance of the Harmony Search. We have proposed a fuzzy [42][43][44]based accelerating factor in such a way and impose it into the algorithm of POHS. We have chosen the accelerating factor in such a way that in the next iteration we can reach closer to the global optimum than the previous iteration. The accelerating factor is:

$$acc = \left[\frac{(OVTR - best_{fit})}{(OVTR - worst_{fit} + eps1)}\right] * eps$$ **………….Equation 9**

Where, bestfit = the achieved bestfitness value, worstfit=the worst fitness value, OVTR = Optimum Value to reach

Initially, we have evaluated this algorithm with some arbitrary eps values. After some tests, we got better performance in eps and eps1 respectively 10^{-5} and 10^{-8}. We have proposed that acc equation in such a way that at first iteration best $_{fit}$ will be close to worst$_{fit.}$ That means acc will be near to 1. But as iteration increases, then also best$_{fit}$ will start to decrease by the algorithm, and when tends to the OVTR

(Optimum Value to reach), then acc will be near to 0, thus the optimization process becomes faster.

3.1 FUZZY BASED PIECEWISE OPPOSITION HARMONY SEARCH ALGORITHM:

Algorithm: A new Fuzzy based Piecewise Opposition Harmony search

---

Input: Harmony size(n), Max number of Function Calls(MAX_{NFC}), fitness function (f(x)), Max value of Jumping rate(JR_{max}), Min value of jumping rate(JR_{min}), global optimum value(OVTR), Max value of Bandwidth(BW_{max}), Min value of Bandwidth(BWmin),Harmony Memory Consideration Rate(HMCR)

Output: The best achieved fitness value close to the global optimum

---

1. Begin

2. Initialize all the parameters Harmony Memory Size (HMS), Pitch Adjustment Rate (PAR_{max} and PAR_{min}), Harmony Memory Consideration Rate (HMCR), Jumping Rate (JR_{max} and JR_{min}), Bandwidth (BW_{max} and BW_{min}), Max number of function calls(MAX_{NFC}).Set the NFC to initially zero.

3. **/*Piecewise Opposition based Harmony Memory Initialization*/**

4. Randomly generate the Harmony memory HM. Let the population be HM(n)={h_1, h_2, …, h_n} where each h_i is a t-dimensional variable

5. for (i=1 to n){

6. **/*Determine the piecewise opposite point of the for the Harmony h_i*/**

7. Initialize a variable g_i as the corresponding piecewise opposite point of the harmony h_i

8. for (j=1 to n){

9. Determine the piecewise opposite point of $g_{i,j}$ of the j-th variable of h_i i.e $h_{i,j}$ i.e $h_{i,j} = a_i + b_i - g_{i,j}$

10. Update the search interval

11. }

12. }

13. **/*Update the harmony memory HM(n)*/**

14. Calculate fitness value for each harmony in current Harmony Memory (HM(n))

15. while (best_fitness_value>OVTR and NFC<MAX_{NFC}){

16. **/*Determine the value of jumping rate*/**

17. $JR_{NFC} = JR_{max}$ - NFC*(JR_{max} -JR_{min})/MAX_{NFC}

18. if (rand(0,1)<JR_{NFC}){

19. for (i=1 to n){

20. **/*Determine piecewise opposite point for each harmony h_i of the harmony memory HM (n) */**

21. Initialize a variable gop_i as the corresponding piecewise opposition point of the individual m_i

22. for (i=1 to n){

23. Calculate the search interval for the j-th variable from the current population

24. Determine the piecewise opposition point gop_i of the j-th variable m_i i.e $m_{i,j}$

25. Update the search interval

26 }

27. }

28. } else{

29. **/*Apply the fuzzy based POHS Algorithm on the current harmony memory*/**

30. if(rand(0,1)<HMCR){

31. Randomly select a harmony m from the harmony memory HM(n)

32. Compute $PAR_{NFC} = (PAR_{max} - PAR_{min}) \times NFC/MAX_{NFC}$

33. if(rand(0,1)<PAR_{NFC}){

34. Compute$BW_{NFC} = \begin{cases} \frac{(BW_{max} - BW_{min})}{1000} \; if\ the\ selected\ Harmony\ h = 0 \\ h \;\; if\ the\ selected\ harmony\ h \in (0,1) \\ \max\left[\frac{BW_{max} - BW_{min}}{50{,}1}\right] \; otherwise \end{cases}$

35. if (h>0){

36. h_new=h-rand(0,1)×BW_{NFC}

37. if(h_new<OVTR)

38. h_new=h

39. }

else{

h_new=h+rand(0,1)× BW_{NFC}

if (h_new>OVTR):

h_new=h

}

}

else{

epsilon=1e-5

eps1=1e-8

acc=((OVTR-best_fitness)/(OVTR-worst_fitness +eps1))*epsilon

A=(best_fitness-(worst_fitness-best_fitness)*rand(0,1))*acc

if (h>0) {

$$compute\ h_{new} = \begin{cases} h + A \times acc & if\ A < 0 \\ h - A \times acc & if\ A > 0 \\ h - 0.1 & otherwise \end{cases}$$

if (h_new<OVTR)

h_new=h

}

else{

$$compute\ h_{new} = \begin{cases} h + A \times acc & if\ A < 0 \\ h - A \times acc & if\ A > 0 \\ h - 0.1 & otherwise \end{cases}$$

if(h_new>OVTR)

h_new=h

}

Calculate the best_fitness value and the worst_fitness value from the updated harmony memory

}

}else{

epsilon=1e-5

eps1=1e-8

acc=((OVTR-best_fitness)/(OVTR-worst_fitness+eps1))*epsilon

A=(best_fitness-(worst_fitness-best_fitness)*rand(0,1))*acc

```
                    Calculate h_new value for each harmony in harmony memory
       Calculate the best_fitness value and the worst_fitness value from the updated
       harmony memory
                 }
    NFC=NFC+1
        }
     }
 Determine fitness value for each harmony
 Return the best solution
 End
```

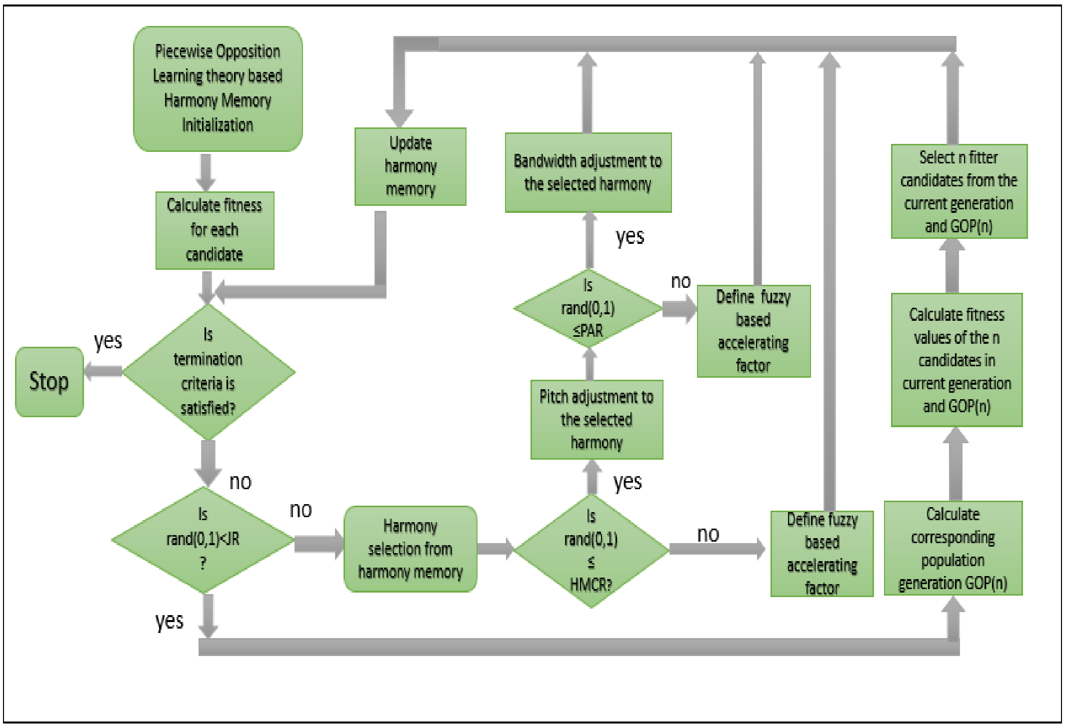


Figure 13: **Flowchart of the developed Fuzzy-POHS algorithm**

To test the newly proposed method, we have selected some benchmark functions which are of different characteristics. F_2, F_3, F_4, F_7, F_8, F_{10}, F_{11} are two-dimensional functions. The rest are multi-dimensional. The selected benchmark functions have been run over 100 trials and 50000 iterations per trial. Initialization techniques and parameters are used as suggested by respective author. Parameters are set and described in section 3.2.1.

3.2 EXPERIMENTS AND RESULTS:

A set of benchmark functions has been utilized to experiment the performance of the proposed algorithm. Here we have gone through 15 benchmark functions and determine the fitness value of the functions by the new fuzzy based[45] piecewise opposition harmony search algorithm which has been discussed above and measured the time complexity of each function. As a result, we observed that for 9 functions the proposed algorithm gives better fitness value. The functions have been selected from [20]. The functions are of different characteristics, some are unimodal[46], some are multimodal. Then the Success Rate is also calculated using:

$$SR = a/b$$

…….………….Equation 10

Where, a =Number of experimental trails which achieved the desired value before reaching MAX_{NFC}

b=Total number of experimental trials

3.2.1 PARAMETERS USED IN THE PRESENT WORK:

A few amounts of parameters have been taken for the Harmony Search algorithm which is initialized at the first of the program.

Table 2: **Parameters of the proposed Algorithm**

Parameter Name	Parameter Value
Population Size(HMS)	100
Dimension of the test functions(D)	30
Harmony Memory Consideration Rate(HMCR)	$1 - \frac{1}{(HMS \times dimension\ of\ the\ test\ functions)}$
Max value of Pitch Adjustment Rate(PAR_{max})	0.99
Min value of Pitch Adjustment Rate(PAR_{min})	0.35
Max value of Bandwidth(BW_{max})	$\frac{1}{20}(X^U - X^L)$
Min value of Bandwidth(BW_{min})	$1e^{-6}$
Max value of Jumping Rate(JR_{max})	0.3
Min value of Jumping Rate(JR_{min})	0
Max value of Number of Iterations(MAX_{NFC})	50000
Termination Criteria	Number of function call reaches MAX_{NFC} or difference between the best value achieved in the algorithm and the global optimum should be 1% of the global optimum

3.2.2 Comparison between the previous HS and the fuzzy based POHS:

We have selected 15 functions for testing the performance of our algorithm. Here is the comparison between the results of the original POHS (Piecewise Opposition Harmony Search) and our proposed fuzzy based POHS algorithm.

Table 3:**Results of the benchmark Functions for testing**[20]

Serial No.	**Function No**	**Function Name**	**OHS**	**POHS**	**Fuzzy Based POHS**	**Improvement over POHS**
1	F_{11}	Griewank Function	8.76×10^{-1}	6.76×10^{-1}	1.90×10^{-8}	+99.99%
2	F_{16}	Six Hump CamleBack Function	-9.91×10^{-1}	-1.031628	-9.84×10^{-1}	-4.61%
3	F_{36}	Bohachevsky's Function 1	4.87×10^{-1}	3.30×10^{-2}	1.11×10^{-16}	+100%
4	F_{37}	Bohachevsky's Function 2	1.75×10^{-1}	3.30×10^{-2}	8.65×10^{-12}	+99.99%
5	F_{5}	Rosenbrocks valley	6.64×10^{3}	5.04×10^{3}	2.90×10^{1}	+94.25%

		Function				
6	F_{78}	Sum of different power Function	5.06×10^{-21}	2.30×10^{-2}	2.10×10^{-23}	+100%
7	F_{33}	Colville Function	4.90×10^{2}	2.51×10^{1}	4.20×10^{1}	-67.33%
8	F_{31}	Easom Function	-8.02×10^{-5}	-9.99×10^{-1}	-1.98×10^{-1}	-119.19%
9	F_{79}	Levy Function	0.86×10^{1}	0.13×10^{1}	3.30×10^{-3}	+99.74%
10	F_{32}	Matyas Function	3.52×10^{-21}	3.48×10^{-21}	0	+100%
11	F_{11}	Three Hump Camel back Function	5.06×10^{-21}	2.75×10^{-2}	0	+100%
12	F_{1}	Sphere Function	4.46×10^{-21}	1.39×10^{-23}	8.42×10^{-9}	-6.05×10^{16} %
13	F_{4}	Schwefel's 2.21 Function	5.76×10^{-21}	5.49×10^{-21}	2.37×10^{-5}	-8.94×10^{20} %
14	F_{80}	Rectangular Step Function	6.68×10^{-21}	6.05×10^{-21}	4.91×10^{-2}	-8.11×10^{-20} %

15	F_9	Rastrigin Function	2.88×10^1	2.08×10^1	1.99×10^1	+4.33 %

3.2.3 Success Rate and Absolute Errors over the Benchmark Functions:

Success Rate is calculated over the 15 benchmark functions and for 9 benchmark function the result s has been improved. So success Rate has been given for 9 benchmark functions:

Table 4: **Success Rate and Absolute errors of the Benchmrk Functions**

Function No.	**Function Name**	**Success Rate**	**Absolute Error**
1	Griewank Function	1	1.90×10^{-8}
3	Bohachevsky's Function 1	1	1.11×10^{-16}
4	Bohachevsky's Function 2	1	8.65×10^{-12}
5	Rosenbrocks valley Function	0.80	2.90×10^{1}
6	Sum of different power Function	1	2.10×10^{-23}
9	Levy Function	0.97	3.30×10^{-3}

10	Matyas Function	1	0
11	Three Hump Camel back Function	1	0
15	Rastrigin Function	0.80	1.99×10^{1}

3.3 Graphical Implementations of some Benchmark Functions:

Function 1(Greiwank's Function):

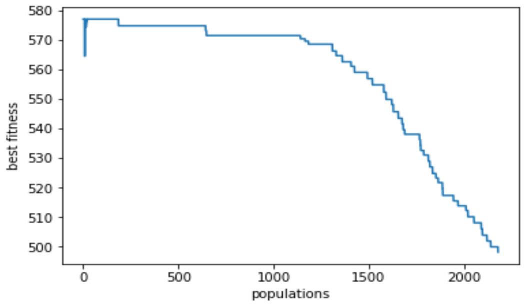


Figure 14: **After 2000 iterations**

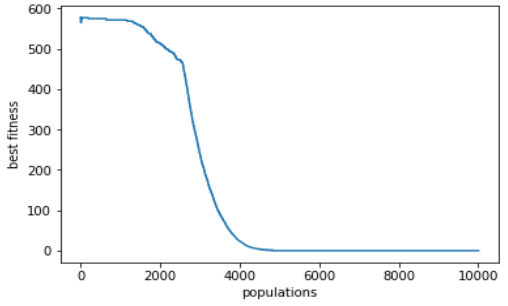


Figure 15: **After 10000 iterations**

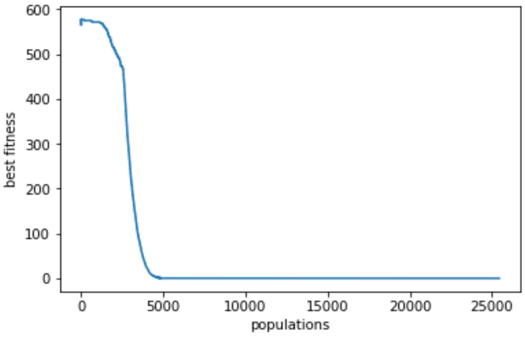


Figure 16: **After 25000 iterations**

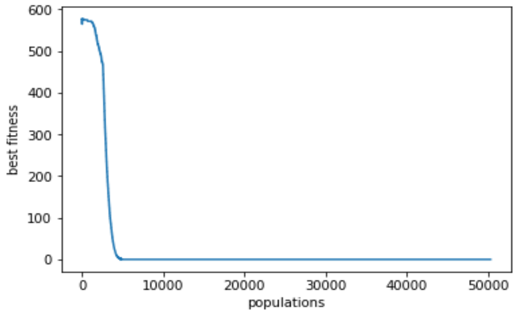


Figure 17: **After 50000 iterations**

<u>Function 3 (Bohachevsky's Function1):</u>

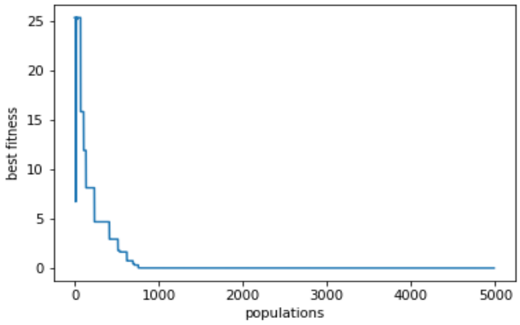


Figure 18: **After 5000 iterations**

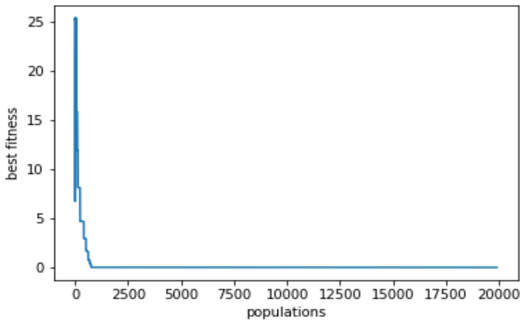


Figure 19: **After 20000 iterations**

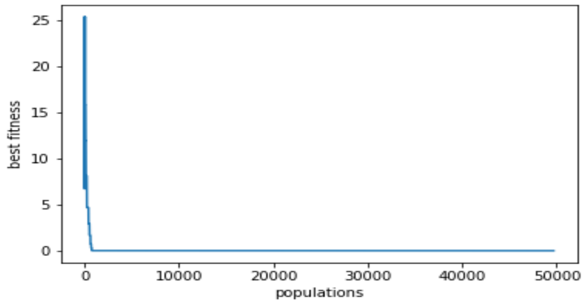
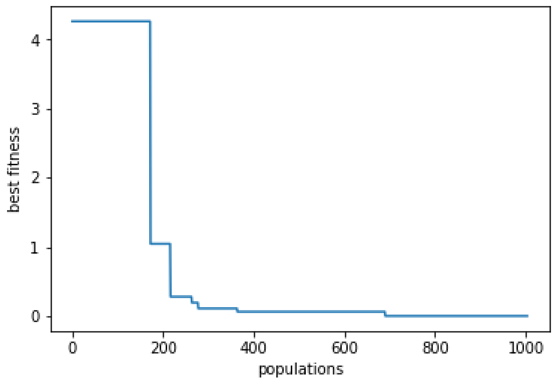


Figure 20: **After 50000 iterations**

Function 4 (Bohachevsky's Function2):



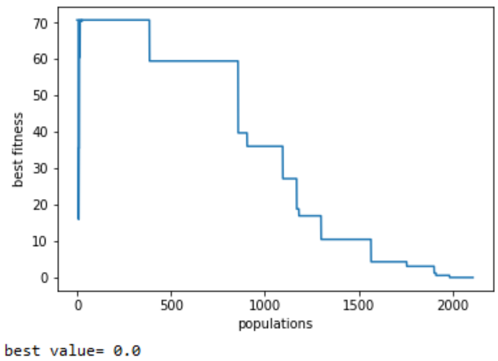


Figure 21: **After 10000 iterations**

Figure 22: **After 2000 iterations it has reached the global optimum value**

Function 6(Sum of different power function):

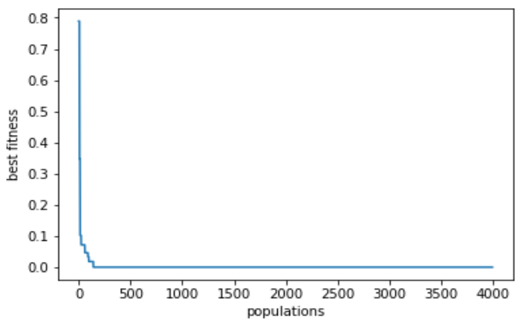


Figure 23: **After 4000 iterations**

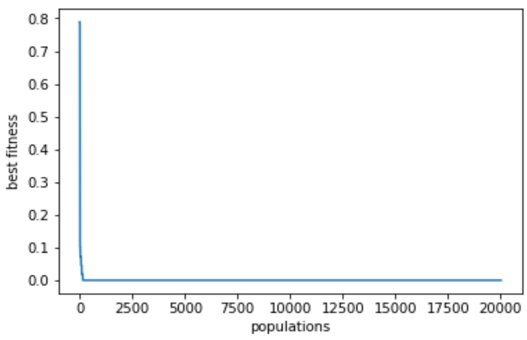


Figure 24: **After 20000 iterations**

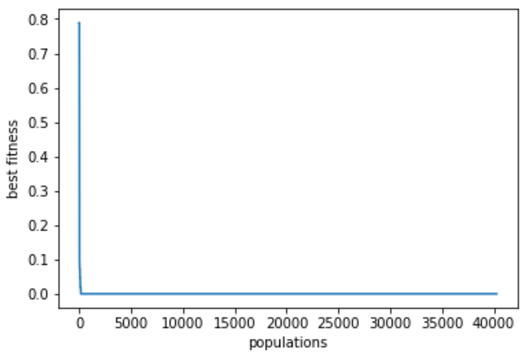


Figure 25: **After 40000 iterations**

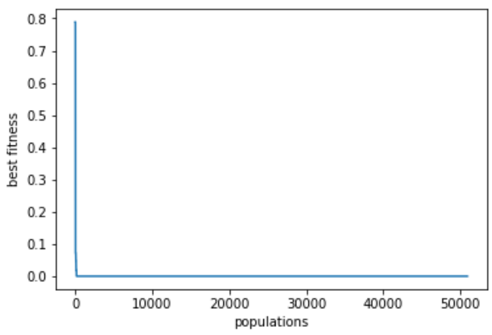
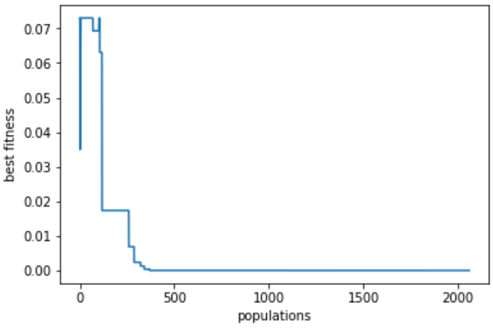


Figure 26: **After 50000 iterations**

Function 10(Matyas Function):



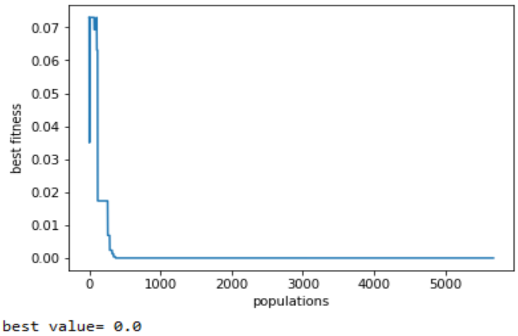


Figure 27: **After 2000 iterations**

Figure 28: **After 6000 iterations(appox) it has reached to the global optimum value**

3.4 ANALYSIS:

There are so many benchmark functions for optimization. From them, we have selected 15 benchmark functions. Each benchmark functions evaluated over 100 trials and 50000 iterations. This proposed algorithm comapared the performance with previous POHS (R. Sarkhel et al. 2018). It gave better results for 9 functions with respect to previous. We have also observed that for most of the periodic functions do not porvide better results in our algorithm.

CHAPTER FOUR

CONCLUSION AND FUTURE WORK

4.1 CONCLUSION:

Day by day, people want to make their work in less amount of time. So reducing the time in every aspect has been a great challenge to the researchers based on their demands. Here, the concept of "*optimization*" comes and Harmony Search is one of the most popular search algorithms in the field. After emerging the Harmony Search in 2001, researchers have taken tremendous efforts to improve the algorithm. Here, we have proposed a methodology introducing a new fuzzy-based accelerating factor, which has been embedded in the POHS algorithm and evaluated on different benchmark functions. As a result, we have observed that the proposed algorithm gives better results rather than the original POHS in most of the casesOur algorithm performs well on 9 benchmark functions over 15 different types of benchmark functions selected for evaluation purpose.

4.2 FUTURE WORK:

From Industry to Science, optimization is required everywhere. In addition, Harmony Search has a broad area of applications. Therefore, there is a huge future scope of this field. It can be developed further in other ways introducing different types of fuzzy membership functions. There are very few research works are exists related to fuzzy based Harmony search[48]. Harmony search can be applied for feature selection[49][50][51]. So there are a lot of potentials to work on Harmony search.

Apart from this, like the hybrid Harmony Searches, one can also develop other types of hybrid algorithms (SAHS, GHS, IGHS etc which have been discussed earlier) in this field. Also, a comparative discussion of some Hybrid algorithms and their performance can be analyzed. This algorithm can be applied to integer programming problem[52], industrial other works and we can also invent some new applications of harmony searches also.

REFERENCES

[1] X. Z. Gao, V. Govindasamy, H. Xu, X. Wang, and K. Zenger, "Harmony search method: Theory and applications," *Computational Intelligence and Neuroscience*. 2015.

[2] K. S. Lee and Z. W. Geem, "A new structural optimization method based on the harmony search algorithm," *Comput. Struct.*, 2004.

[3] Z. Kong, L. Wang, Z. Wu, S. Qi, and D. Zou, "An Improved Adaptive Harmony Search Algorithm *," no. Iccse, pp. 743–747, 2012.

[4] Z. W. Geem and K. B. Sim, "Parameter-setting-free harmony search algorithm," *Appl. Math. Comput.*, vol. 217, no. 8, pp. 3881–3889, 2010.

[5] O. M. D. Alia and R. Mandava, "The variants of the harmony search algorithm: An overview," *Artif. Intell. Rev.*, vol. 36, no. 1, pp. 49–68, 2011.

[6] T. Puri, S. Ganguli, and I. Engineering, "EFFECT OF USER DEFINED PARAMETERS OF HARMONY SEARCH ALGORITHM (HSA) FOR UNCONTRAINED," no. 6, pp. 2–7, 2015.

[7] X. Wang, X. Z. Gao, and K. Zenger, "The overview of harmony search," in *SpringerBriefs in Applied Sciences and Technology*, 2015.

[8] Z. W. Geem, K. S. Lee, and Y. Park, "Application of Harmony Search to Vehicle Routing," *Am. J. Appl. Sci.*, 2005.

[9] B. Amiri, L. Hossain, and S. E. Mosavi, "Application of harmony search algorithm on clustering," vol. vol.1, no. October, pp. 460–465, 2010.

[10] O. M. d. Alia, "Dynamic relocation of mobile base station in wireless sensor networks using a cluster-based harmony search algorithm," *Inf. Sci. (Ny).*, vol. 385–386, pp. 76–95, 2017.

[11] A. S. Alsaadi, T.-C. Wan, and A. Munther, "Application of Harmony Search Optimization Algorithm to Improve Connectivity in Wireless Sensor Network with Non-uniform Density," *J. Inf. Sci. Eng.*, vol. 31, pp. 1475–1489, 2015.

[12] M. Mahdavi, M. H. Chehreghani, H. Abolhassani, and R. Forsati, "Novel meta-heuristic algorithms for clustering web documents," *Appl. Math. Comput.*, vol. 201, no. 1–2, pp. 441–451, 2008.

[13] Z. W. Geem, "Optimal Scheduling of Multiple Dam System Using Harmony Search Algorithm," *Comput. Ambient Intell.*, pp. 316–323, 2007.

[14] *Recent Advances in Harmony Search Algorithm Studies in Computational Intelligence , Volume 270*. 2010.

[15] O. Access, "We are IntechOpen , the first native scientific publisher of Open Access books TOP 1 %."

[16] Z. W. Geem and J.-Y. Choi, "Music Composition Using Harmony Search Algorithm," *Lncs*, vol. 4448, pp. 593–600, 2007.

[17] D. Zou, L. Gao, S. Li, and J. Wu, "An effective global harmony search algorithm for reliability problems," *Expert Syst. Appl.*, vol. 38, no. 4, pp. 4642–4648, 2011.

[18] S. Kulluk, L. Ozbakir, and A. Baykasoglu, "Training neural networks with harmony search algorithms for classification problems," *Eng. Appl. Artif. Intell.*, vol. 25, no. 1, pp. 11–19, 2012.

[19] D. Karaboga and B. Akay, "Artificial Bee Colon (ABC), harmony search and bees algorithms on numerical optimization," *Proc. IPROMS-2009 Innov. Prod. Mach. Syst.*, no. January 2009, 2009.

[20] R. Sarkhel, N. Das, A. K. Saha, and M. Nasipuri, "An improved Harmony Search Algorithm embedded with a novel piecewise opposition based learning algorithm," *Eng. Appl. Artif. Intell.*, 2018.

[21] H. bin Ouyang, L. qun Gao, S. Li, X. yong Kong, Q. Wang, and D. xuan Zou, "Improved Harmony Search Algorithm: LHS," *Appl. Soft Comput. J.*, vol. 53, pp. 133–167, 2017.

[22] M. R. Maheri and M. M. Narimani, "An enhanced harmony search algorithm for optimum design of side sway steel frames," *Comput. Struct.*, 2014.

[23] P. Upadhyay, R. Kar, D. Mandal, S. P. Ghoshal, and V. Mukherjee, *A novel design method for optimal IIR system identification using opposition based harmony search algorithm*, vol. 351, no. 5. Elsevier, 2014.

[24] X. Kong, L. Gao, H. Ouyang, and S. Li, "A simplified binary harmony search algorithm for large scale 0-1 knapsack problems," *Expert Syst. Appl.*, vol. 42, no. 12, pp. 5337–5355, 2015.

[25] M. G. H. Omran and M. Mahdavi, "Global-best harmony search," *Appl. Math. Comput.*, 2008.

[26] D. Zou, L. Gao, S. Li, and J. Wu, "Solving 0-1 knapsack problem by a novel global harmony search algorithm," in *Applied Soft Computing Journal*, 2011.

[27] W. L. Xiang, M. Q. An, Y. Z. Li, R. C. He, and J. F. Zhang, "An improved global-best harmony search algorithm for faster optimization," *Expert Syst. Appl.*, 2014.

[28] B. Alatas, “Chaotic harmony search algorithms,” *Appl. Math. Comput.*, vol. 216, no. 9, pp. 2687–2699, 2010.

[29] A. Askarzadeh, “A discrete chaotic harmony search-based simulated annealing algorithm for optimum design of PV/wind hybrid system,” *Sol. Energy*, vol. 97, pp. 93–101, 2013.

[30] Q. K. Pan, L. Wang, and L. Gao, “A chaotic harmony search algorithm for the flow shop scheduling problem with limited buffers,” *Appl. Soft Comput. J.*, vol. 11, no. 8, pp. 5270–5280, 2011.

[31] “Application of Hs on clustering.”

[32] M. Mernik, S. H. Liu, D. Karaboga, and M. Črepinšek, “On clarifying misconceptions when comparing variants of the Artificial Bee Colony Algorithm by offering a new implementation,” *Inf. Sci. (Ny).*, vol. 291, no. C, pp. 115–127, 2015.

[33] A. Kumar and V. Kumar, “Hybridized ABC-GA optimized fractional order fuzzy pre-compensated FOPID control design for 2-DOF robot manipulator,” *AEU - Int. J. Electron. Commun.*, vol. 79, pp. 219–233, 2017.

[34] M. A. Al-Betar, A. T. Khader, M. A. Awadallah, M. H. Alawan, and B. Zaqaibeh, “Cellular harmony search for optimization problems,” *J. Appl. Math.*, vol. 2013, no. i, 2013.

[35] L. Wang, R. Yang, Y. Xu, Q. Niu, P. M. Pardalos, and M. Fei, “An improved adaptive binary Harmony Search algorithm,” *Inf. Sci. (Ny).*, vol. 232, pp. 58–87, 2013.

[36] P. Yadav, R. Kumar, S. K. Panda, and C. S. Chang, “An Intelligent Tuned Harmony Search algorithm for optimisation,” *Inf. Sci. (Ny).*, vol. 196, pp. 47–72, 2012.

[37] G. Wang, L. Guo, H. Duan, H. Wang, L. Liu, and M. Shao, “Hybridizing harmony search with biogeography based optimization for global numerical optimization,” *J. Comput. Theor. Nanosci.*, vol. 10, no. 10, pp. 2312–2322, 2013.

[38] H. Li, L. Li, T. Kim, and S. Xie, “An improved PSO-based of harmony search for complicated optimization problems,” *Int. J. …*, vol. 1, no. 1, pp. 57–64, 2008.

[39] A. Kaveh, *Harmony Search Algorithms for Structural Design Optimization*, vol. 239, no. July 2015. 2009.

[40] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham, and B. K. Panigrahi, “Exploratory power of the harmony search algorithm: Analysis and improvements for global numerical optimization,” *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, 2011.

[41] R. Venkata Rao and G. G. Waghmare, “A comparative study of a teaching-

learning-based optimization algorithm on multi-objective unconstrained and constrained functions," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 26, no. 3, pp. 332–346, 2014.

[42] C. Peraza, F. Valdez, M. Garcia, P. Melin, O. Castillo, and Y.-C. Liang, "A New Fuzzy Harmony Search Algorithm Using Fuzzy Logic for Dynamic Parameter Adaptation."

[43] M. T. Ayvaz, "Simultaneous determination of aquifer parameters and zone structures with fuzzy c-means clustering and meta-heuristic harmony search algorithm," *Adv. Water Resour.*, vol. 30, no. 11, pp. 2326–2338, 2007.

[44] M. H. Mashinchi, M. a. Orgun, M. Mashinchi, and W. Pedrycz, "A Tabu–Harmony Search-Based Approach to Fuzzy Linear Regression," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 3, pp. 432–448, 2011.

[45] A. A. Taleizadeh, F. Barzinpour, and H. M. Wee, "Meta-heuristic algorithms for solving a fuzzy single-period problem," *Math. Comput. Model.*, vol. 54, no. 5–6, pp. 1273–1285, 2011.

[46] X. Z. Gao, X. Wang, and S. J. Ovaska, "Uni-modal and multi-modal optimization using modified harmony search methods," *Int. J. Innov. Comput. Inform. Control*, vol. 5, no. 10, pp. 2985–2996, 2009.

[47] M. Fesanghary, M. Mahdavi, M. Minary-Jolandan, and Y. Alizadeh, "Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems," *Comput. Methods Appl. Mech. Eng.*, vol. 197, no. 33–40, pp. 3080–3091, 2008.

[48] H. Wang, X. Yuan, Y. Wang, and Y. Yang, "Harmony search algorithm-based fuzzy-PID controller for electronic throttle valve," *Neural Comput. Appl.*, vol. 22, no. 2, pp. 329–336, 2013.

[49] C. C. O. Ramos, A. N. Souza, G. Chiachia, A. X. Falcão, and J. P. Papa, "A novel algorithm for feature selection using Harmony Search and its application for non-technical losses detection," *Comput. Electr. Eng.*, vol. 37, no. 6, pp. 886–894, 2011.

[50] Y. Wang, Y. Liu, L. Feng, and X. Zhu, "Novel feature selection method based on harmony search for email classification," *Knowledge-Based Syst.*, vol. 73, no. 1, pp. 311–323, 2014.

[51] R. Diao and Q. Shen, "Two new approaches to feature selection with harmony search," *Fuzzy Syst. (FUZZ), 2010 IEEE …*, pp. 18–23, 2010.

[52] A. F. and G. SHAFIPOUR, "Harmony Search Algorithm Based 0-1 Integer Programming for Generation Maintenance Scheduling in Power Systems," *Expert Syst.*, pp. 1–10, 2005.