PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAISO
FACULTAD DE INGENIERÍA
ESCUELA DE INFORMÁTICA

Genetic Algorithm to Parameter Tuning for the Bat Optimization Approach to resolve the Set Covering Problems

Claudia Montserrat Olea Valdebenito
Teacher Guide: Broderick Crawford
October 12, 2014

Abstract

In this work, we present a Bat Optimization Approach to solve the Set Covering Problem. The Bat Algorithm is inspired in the echo-localization used by microbats to find food and avoid obstacles. Eight different transfer functions will be used to solve the binary problem. Genetic Metaheuristic will be used as a search technique to find approximate solutions for the configuration parameters of the Binary Bat Algorithm. We select the best transfer function and illustrate this approach with 65 instances of the problem.

Contents

1. Introduction and Bibliographic Discussion

As the theorem "No Free Lunch" says, there's not a solver able to find good solutions for all the problems. There's not a universal multipurpose solver for optimization. That's why we have to keep searching for better and new options to resolve problems.

An Autonomous Search system must be able to autonomously solve problems, acquire and discover new knowledge, which can be reused later. This type of system has as ultimate goal to provide an easy-to-use interface for end users, who could provide a possibly informal description of their problem and obtain solutions with minimum interaction and technical knowledge.

In the present we will do an Autonomous search system that uses Parameter Tuning. Parameter Tuning is used to find Off-line configurations for parameters values for a search technique, in this case for a Binary Bat Algorithm.

This type of work is called a Hyper-heuristic, is a Meta-heuristic that is used to find parameters for another meta-heuristic that can resolve a problem. My master thesis is about finding a solution for the Set Covering Problem (SCP) using Binary Bat algorithm and Genetic algorithm to find good parameters for the Binary Bat Algorithm. This case can be considered a Hyper-heuristic.

The problem is a minimization one, the SCP main proposes is to minimize the cost sum of the selected columns, when $x_{j=}1$ a column j belongs to the solution. The constraints allow ensuring that every row i is cover for at least one column.

The Binary Bat Algorithm (BBA) will perform differently based on his setting of parameters.

A Genetic Algorithm will be used to determine the best configuration for the set of parameters that the BBA uses. The proposed Algorithm will evaluate different combinations of the BBA instead of doing a manual parameterization.

GA allows simulating the evolution of the lower-level meta-heuristic, in this case BBA. GA solutions are represented by each member of the population; each solution represents an instance of the BBA.

2. Goals Analysis

2.1. Main Goal

- Build a system that can autonomously solve the Set Covering Problem using a Meta-Optimization.

2.2. Specific Goals

- Use the Bat Algorithm Methaheuristic
- Use the Genetic Algorithm as a Hiperheuristic to set the configuration Values
- Probe the solution using the Beasley Benchmarks
- Use eight different binarization techniques and define witch one is the best for this case.

3. Methodology

In this work we use an incremental software development methodology, this means that the implementation and testing will be done incrementally, adding a new part/functionality each time.

4. Work Plan

4.1 Thesis Seminar

The following plan was made considering eight weeks; the reason is to have time to show the Teacher guide the reports for corrections before the official dates.

Task	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8
Binarization implementation of eight techniques into Bat Algorithm implementation								
Test Plan Elaboration Part 1								
Test Plan Implementation Part 1								
Analyze Tests Results Part 1								
Elaboration of Progress Report Official Format								
Elaboration of Progress Report Paper Format								
Genetic Algorithm implementation for parameter tunning								
Test Plan Elaboration Part 2								
Test Plan Implementation Part 2								
Analyze Tests Results Part 2								
Elaboration of Final Report Official Format								
Elaboration of Final Report Paper Format								

4.2 Thesis Project

The following plan was made considering fourteen weeks of work.

Task	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14
Compare Results with other techniques														
Test Plan Elaboration Part 1														
Test Plan Implementation Part 1														
Analyze Tests Results Part 1														
Elaboration of Progress Report Official Format														
Elaboration of Progress Report Paper Format														
Integration														
Test Plan Elaboration Part 2														
Test Plan Implementation Part 2														
Analyze Tests Results Part 2														
Elaboration of Final Report Official Format														
Elaboration of Final Report Paper Format														

5. Problem Description

The Set Covering Problem [2] is a classic problem in computer science. There are many problems from the real world that can be modeled as a Set Covering Problem: facility location, airline crew scheduling, nurse scheduling, and resource allocation, among others. Incomplete techniques are used to resolve this problem.

If $A = (a_{ij})$ has m-rows y n-columns, a matrix with 0 y 1 values. The j column covers a row i if $a_{ij} = 1$. Each column j is related to a real not negative cost c_j .

If $I = \{1, \dots, m\}$ y $J = \{1, \dots, n\}$ are a row set and a column set respectively. The main goal is to minimize the cost of S sub-set J, each row $i \in I$ most be cover for at least one column $j \in S$.

The mathematical model of the Set Covering Problem is:

$$Minimaze\ Z = \sum_{j=1}^{n} c_j x_j \quad j = \{1,2,3, \dots, n\}$$

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \quad i = \{1,2,3, \dots, m\}$$

$$x_j = \{0,1\}$$

Find the correct algorithm to resolve this problem and find the correct configuration to this algorithm is a big part of the problem complexity.

6. Proposed Solution

6.1. Binary Bat Algorithm to resolve the Set Covering Problem

The Binary Bat Algorithm (BBA) [3] will be used to solve the Set Covering Problem.

The Bat Algorithm [1] is bio-inspired in the echo-localization used by micro-bats to find food and avoid obstacles. They send a sound and listen the echo that produce this sound when gets to an object.

There are three rules in which this Metaheuristic is based:

1. Each bat flies using echo-localization to know the distance to food, walls and obstacles. The difference between these objects is "magically" known by the bat
2. Bats fly randomly with a v_i velocity, in a x_i position, with a f_{min} frequency, changing its wavelength λ and A_0 minimum loudness to a constant loudness A_{min}
3. Loudness can change in different ways. We have to assume that varies from a maximum A_0 to a minimum and constant value A_{min}.

6.1.1 Parameters, Constants and Equations Involve

The involve parameters are:

- x_i^j: $bat\ i\ position$
- v_i^j: $bat\ i\ velocity\ in\ j\ variable$
- f_i: $bat\ i\ frequency$
- f_{min}: minimum $bat\ i\ frequency$
- f_{max}: maximum $bat\ i\ frequency$
- r_i: $bat\ i\ pulse\ rate$
- A_i: $bat\ i\ loudness$

The constants values used are:

- β: $random\ number\ between\ 0\ and\ 1$

The bat algorithm equations that we use are

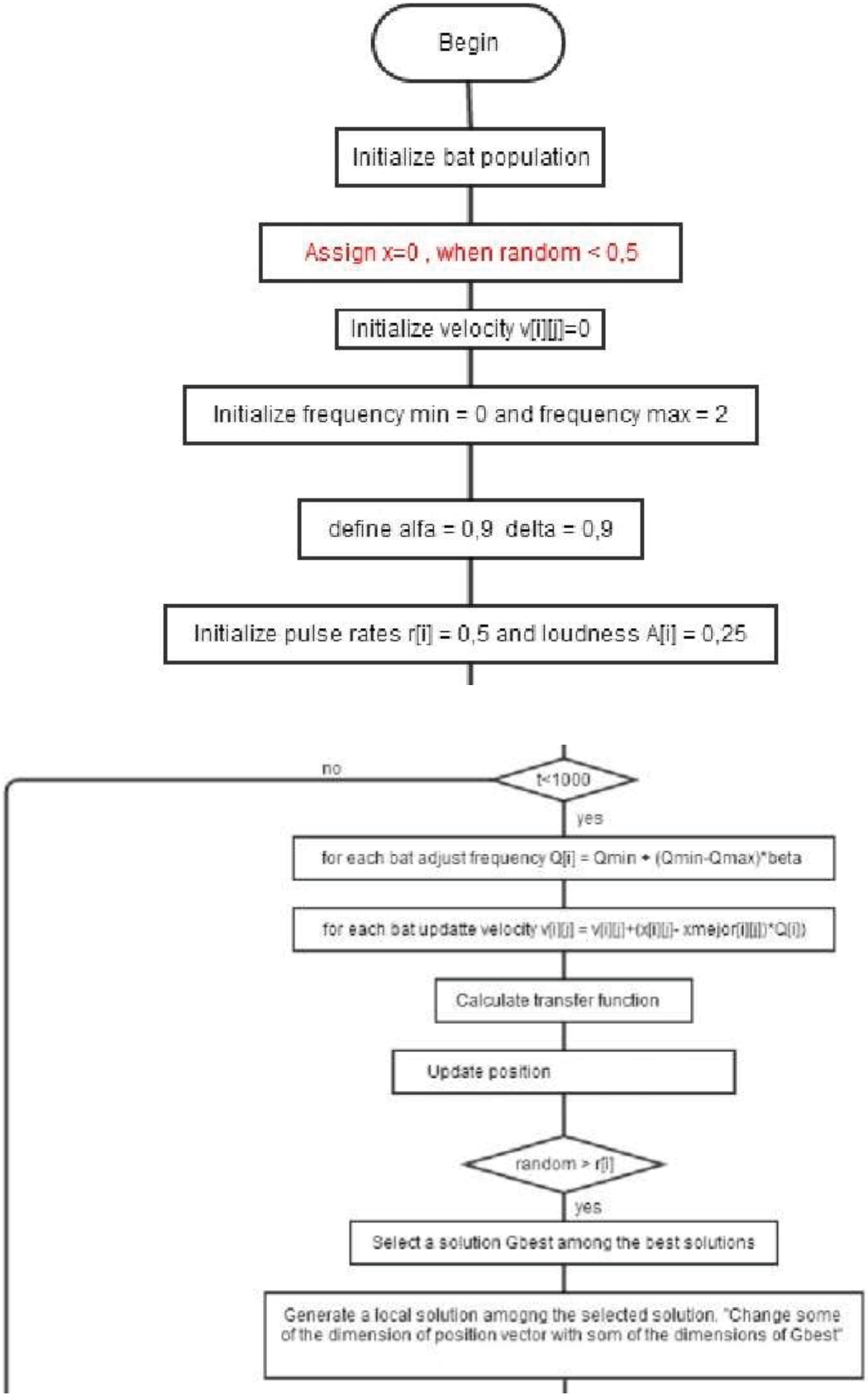
- $\boldsymbol{f_i = f_{min} + (f_{max} - f_{min})\beta}$ **(1)**
- $\boldsymbol{v_i^j(t) = v_i^j(t-1) + [\hat{x}^j - x_i^j(t-1)]f_i}$ **(2)**

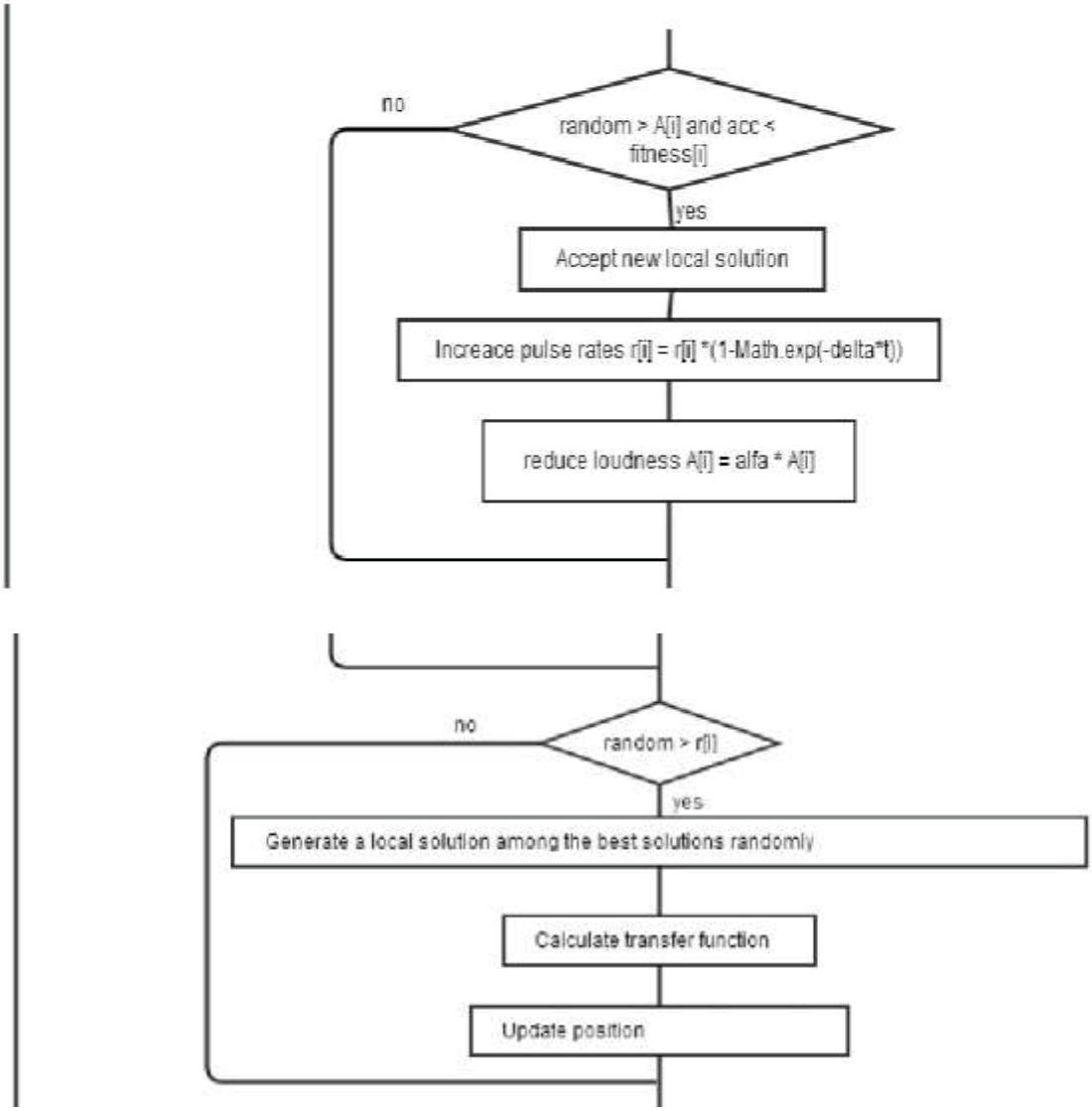
In the original Bat Algorithm three equations are used, the third one is not useful in this Binary case, that step will be replaced by a transfer function that is after transform with a discrete function.

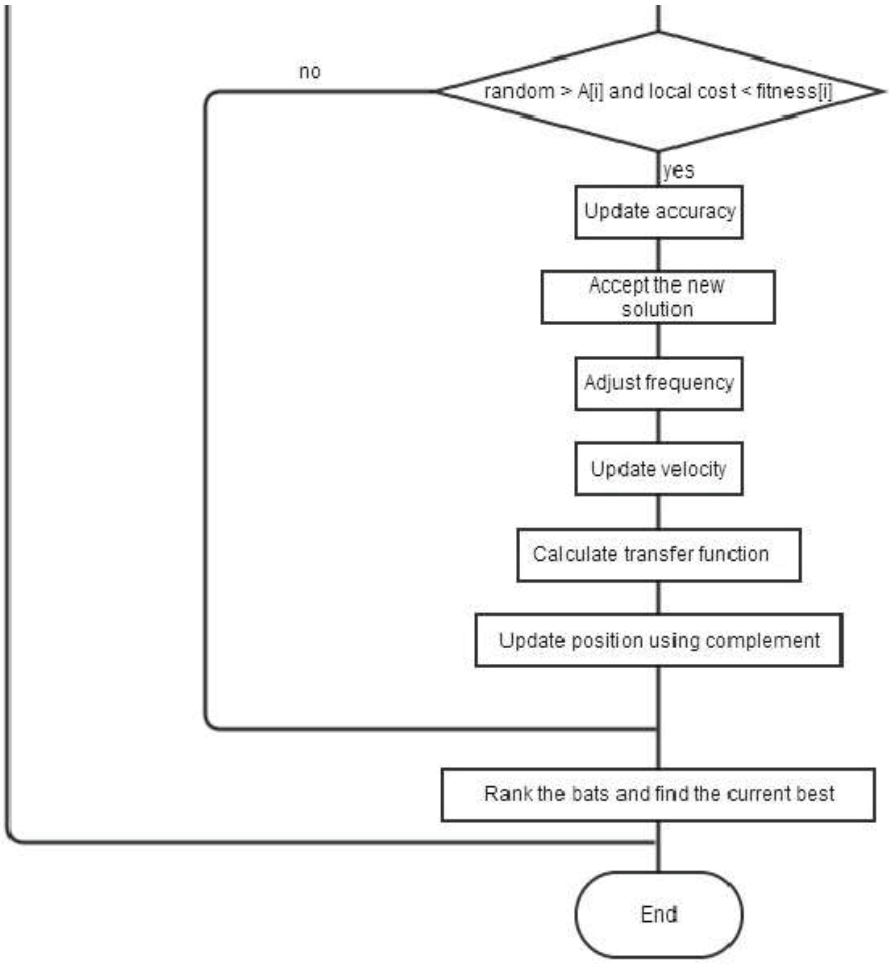
The original proposed Binary Bat Algorithm [6] is:

1. $Initialize\ the\ bat\ population: x_i (i = 1,2, \dots, n) = rand(0\ or\ 1)\ and\ v_i = 0$
2. $Define\ pulse\ frequency\ f_i$
3. $Initialize\ pulse\ rates\ r_i\ and\ loudness\ A_i$
4. $while\ (t < Max\ number\ of\ iterations)$
5. $Adjusting\ frequency\ and\ updating\ velocities$
6. $Calculate\ transfer\ function$
7. $Update\ positions\ using\ equation\ discrete\ function$
8. $if (rand > r_i)$
9. $Select\ a\ solution\ (Gbest) among\ the\ best\ solutions\ randomly$
10. $Change\ some\ dimensions\ of\ position\ vector\ with\ some\ of\ the\ dimensions\ of\ Gbest$
11. $end\ if$
12. $Generate\ a\ new\ solution\ by\ flying\ randomly$
13. $if (rand > A_i\ \&\ f(x_i) < f(Gbest))$
14. $Accept\ the\ new\ solutions$
15. $Increase\ r_i\ and\ reduce\ A_I$
16. $end\ if$
17. $Rank\ the\ bats\ and\ find\ the\ current\ Gbest$
18. $end\ while$

The implementation of this methaheuristic to resolve the problem consider that each row represents a bat position and each bat position will be a solution for the problem. The following flowchart describes the propose solution.


Begin
Initialize bat population
Assign x=0 , when random < 0,5
Initialize velocity v[i][j]=0
Initialize frequency min = 0 and frequency max = 2
define alfa = 0,9 delta = 0,9
Initialize pulse rates r[i] = 0,5 and loudness A[i] = 0,25
no
t<1000
yes
for each bat adjust frequency Q[i] = Qmin + (Qmin-Qmax)*beta
for each bat updatte velocity v(i)[j] = v[i][j]+(x[i][j]- xmejor[i][j])*Q[i])
Calculate transfer function
Update position
random > r[i]
yes
Select a solution Gbest among the best solutions
Generate a local solution amogng the selected solution. "Change some of the dimension of position vector with som of the dimensions of Gbest"


no
random > A[i] and acc < fitness[i]
yes
Accept new local solution
Increace pulse rates r[i] = r[i] *(1-Math.exp(-delta*t))
reduce loudness A[i] = alfa * A[i]
no
random > r[i]
yes
Generate a local solution among the best solutions randomly
Calculate transfer function
Update position



6.1. Genetic Algorithm to Parameter Tunning

The Genetic Algorithm (GA) [4] is a search technique used to find approximate solutions for search and optimization problems. Genetic Algorithm operates by iteratively updating a population of individuals, which are candidate solutions to the problem. The individuals are encoded as binary strings, called chromosomes, and are assigned a fitness value according to their evaluation

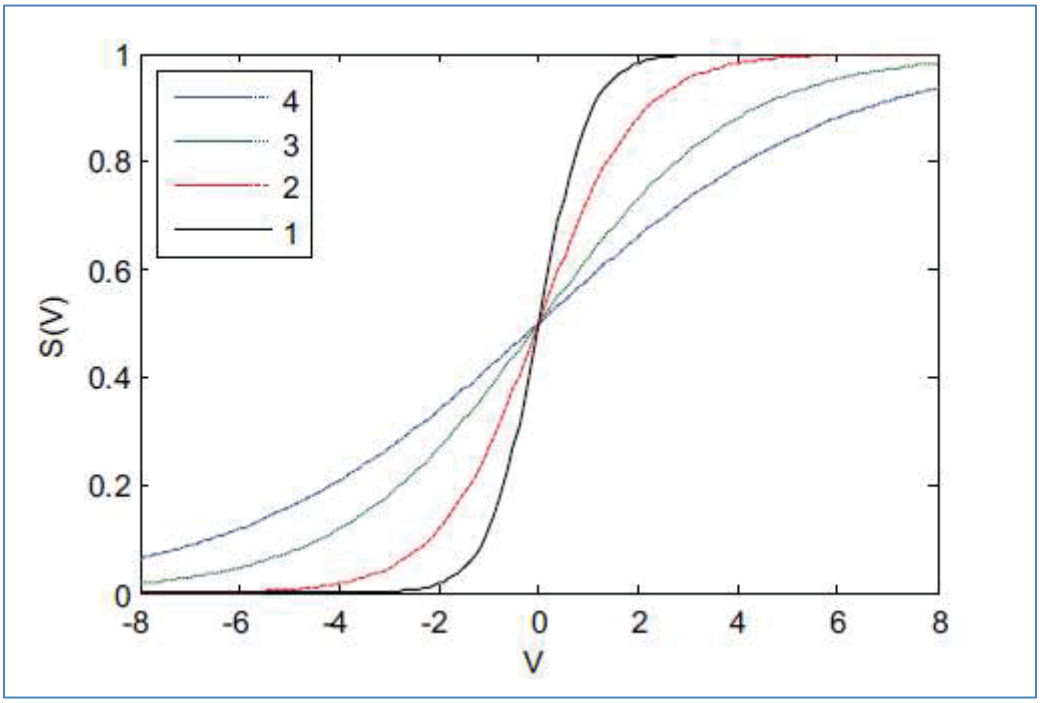
Before running the Bat Algorithm the Genetic Algorithm is used to obtain parameter values that could be useful over a wide range of instances.

Genetic Algorithm will be used as a search technique to find approximate solutions for the configuration parameters of the Binary Bat Algorithm. This means tuning the Bat Algorithm using the Genetic Algorithm to resolve the problem

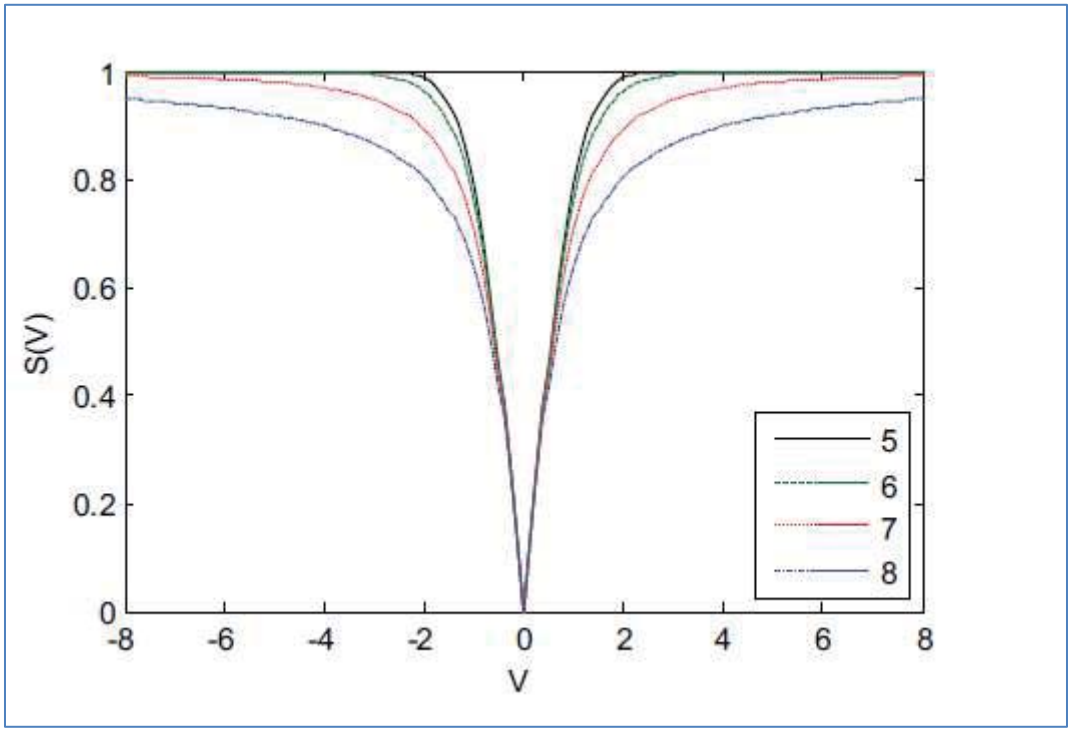
7. Transfer Functions

Eight Transfer functions are going to be proved. Transfer function 1, 2, 3 and 4 are S-shaped family transfer functions [5]. The functions 5, 6, 7 and 8 are from the V-shaped family of functions.

N°	Transfer Function
1	$S(x) = \frac{1}{1+e^{-2x}}$
2	$S(x) = \frac{1}{1+e^{-x}}$
3	$S(x) = \frac{1}{1+e^{-\frac{x}{2}}}$
4	$S(x) = \frac{1}{1+e^{-\frac{x}{3}}}$
5	$S(x) = \left\lvert \text{erf}\,(\frac{\sqrt{\pi}}{2}x) \right\rvert$
6	$S(x) = \lvert \tan(x) \rvert$
7	$S(x) = \left\lvert \frac{x}{\sqrt{1+x^2}} \right\rvert$
8	$S(x) = \left\lvert \frac{\pi}{2} \arctan\,(\frac{\pi}{2}x) \right\rvert$



1: S-shape family transfer functions



2: V-shape family transfer functions

3. Discrete Functions

Five discrete functions are used:

- Standard

$$x_{i,j}^{k}(t+1) = \begin{cases} 0 & If\ rand < S(v_{i,j}^{k}(t+1)) \\ 1 & If\ rand \geq S(v_{i,j}^{k}(t+1)) \end{cases}$$

- Complement

$$x_{i,j}^{k}(t+1) = \begin{cases} complement(x_{i,j}^{k}(t)) & If\ rand < S(v_{i,j}^{k}(t+1)) \\ x_{i,j}^{k}(t) & If\ rand \geq S(v_{i,j}^{k}(t+1)) \end{cases}$$

- Set the best

$$x_{i,j}^{k}(t+1) = \begin{cases} Best\ position & If\ rand < S(v_{i,j}^{k}(t+1)) \\ 0 & If\ rand \geq S(v_{i,j}^{k}(t+1)) \end{cases}$$

- Static Probabilistic

$$t = \frac{1}{(1 + \exp(-D))}$$

$$x_w\ (new) = \begin{cases} 0 & if\ t \leq \alpha \\ x_w & if\ \alpha < t \leq \frac{1}{2}(1+\alpha) \\ 1 & if\ t \geq \frac{1}{2}(1+\alpha) \end{cases}$$

- Roulette Wheel

$$x_{i,j}^{k}(t+1) = \begin{cases} \dfrac{f_i}{\sum_{j=1}^{k} f_i} & If\ rand < S(v_{i,j}^{k}(t+1)) \\ 0 & If\ rand \geq S(v_{i,j}^{k}(t+1)) \end{cases}$$

8. Test Results

The test are the most important part of our investigaction. We need to obtain valid results that could be reproduce executing the algorithm. The main goal of the preliminary test were to validate that the algorithm implementation was a valid one and to observe what kind of results we obtain with it.

The formal test were made executing every instance 30 times.

8.1 Preliminary Test 1

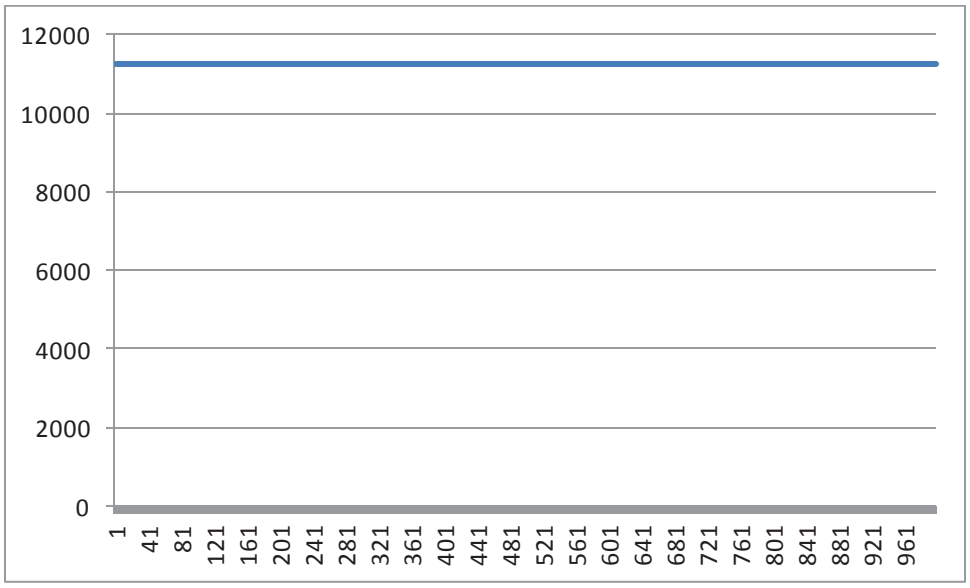
Several test were made with each of the functions, the following graphics are from a test made using the instance scp41 from the Beastly Benchmarks. The best Transfer function for the implementation is 7, as you will see in this case.

Data:

Bats	30
Iterations	1000
Instance	Scp41
Optimum	429

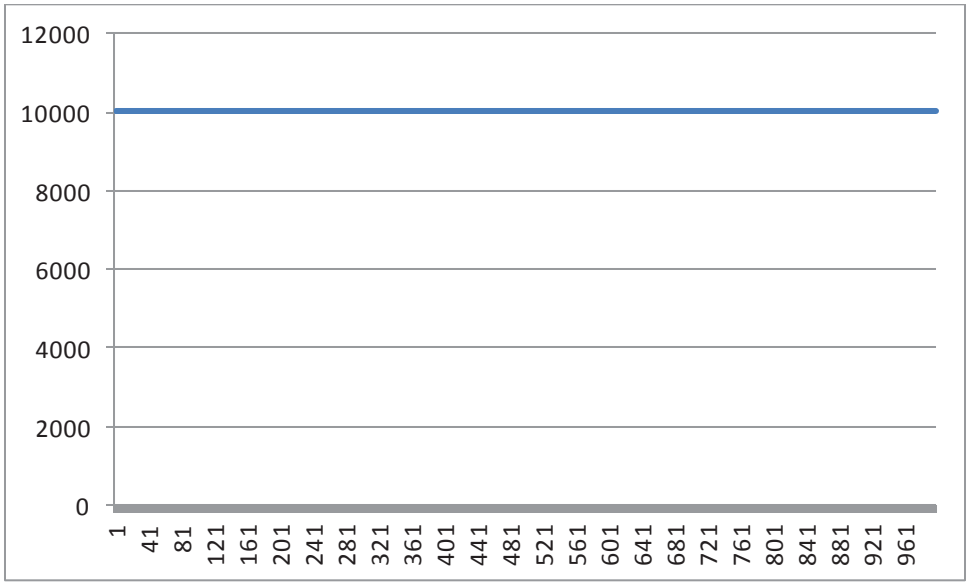
Transfer Function 1

Transfer Function	1
Initial value	11235
Best value	11235
Media	10006



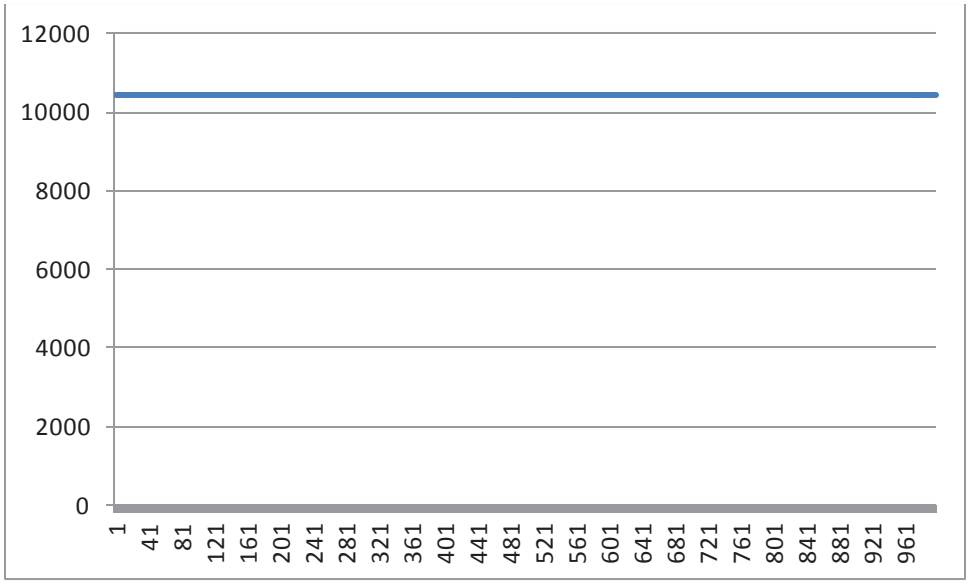
Transfer Function 2:

Transfer Function	2
Initial value	10006
Best value	10006
Media	10006



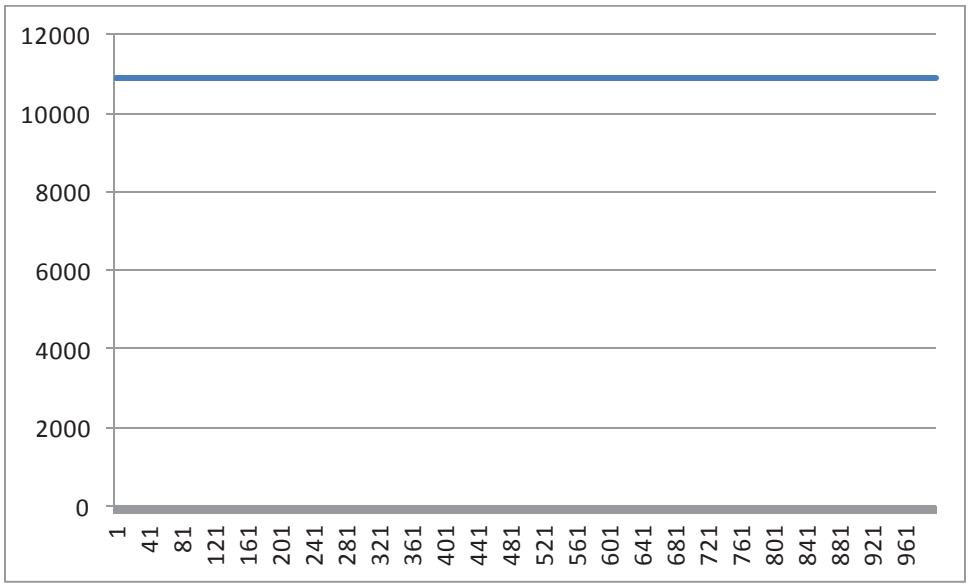
Transfer Function 3

Transfer Function	3
Initial value	10425
Best value	10425
Media	10425



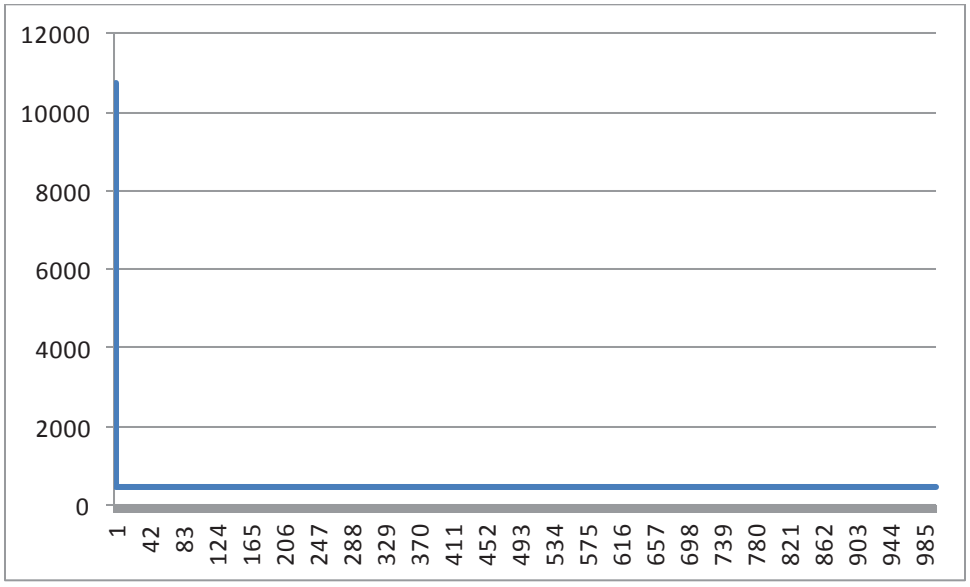
Transfer Function 4

Transfer Function	4
Initial value	10889
Best value	10889
Media	10889



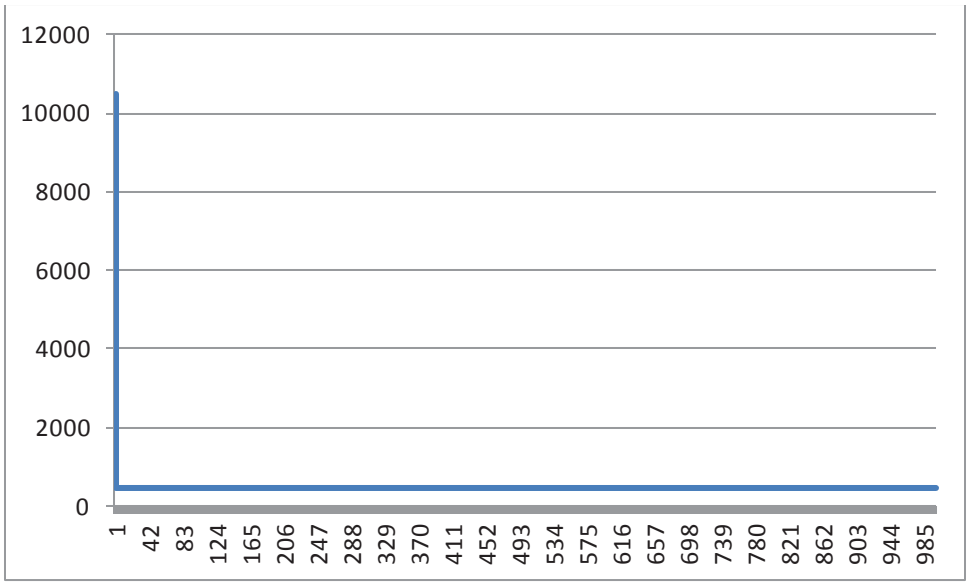
Transfer Function 5

Transfer Function	5
Initial value	10735
Best value	459
Media	461,803



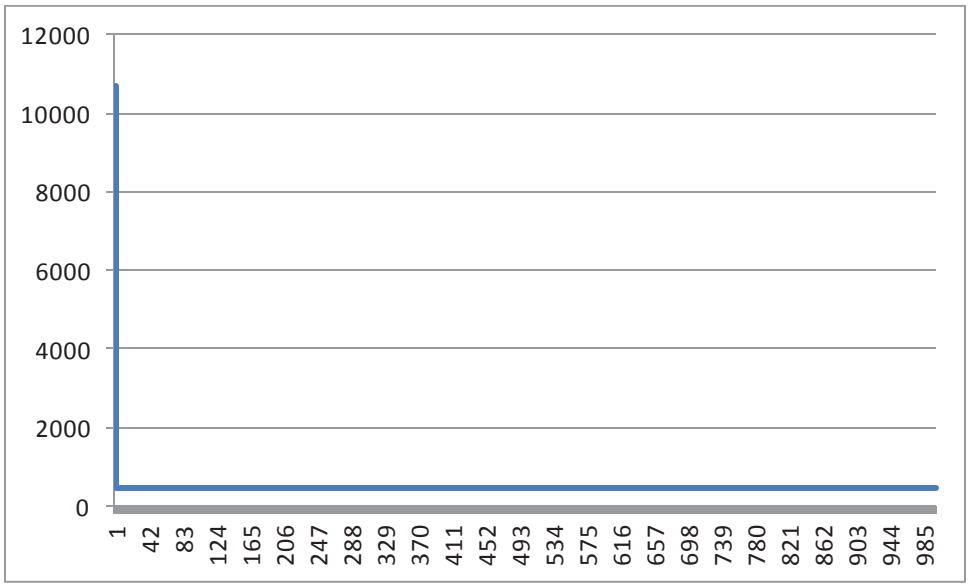
Transfer Function 6

Transfer Function	6
Initial value	10504
Best value	454
Media	455,359



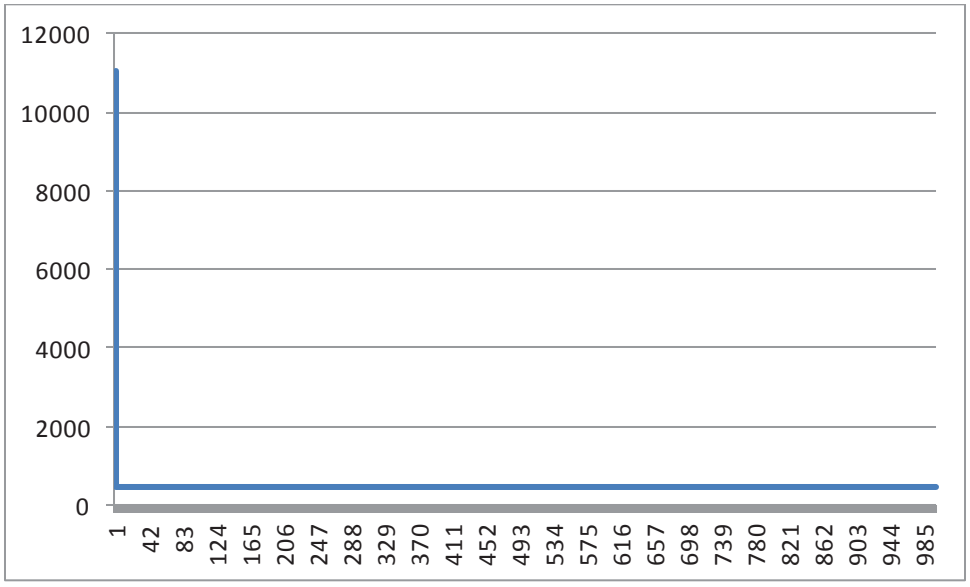
Transfer Function 7

Transfer Function	7
Initial value	10709
Best value	448
Media	448,432



Transfer Function 8

Transfer Function	8
Initial value	11027
Best value	448
Media	448,184

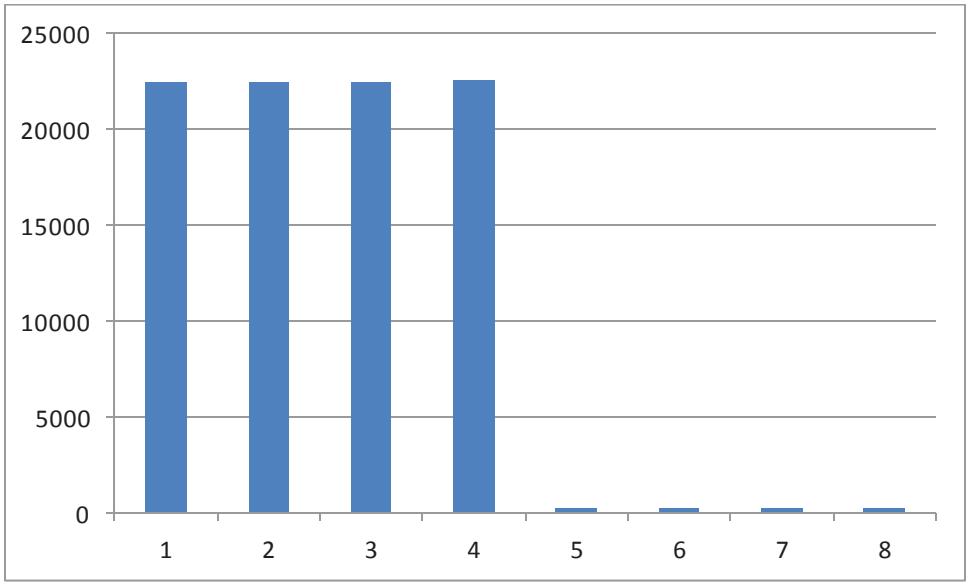


The best transfer function in this test was 7.

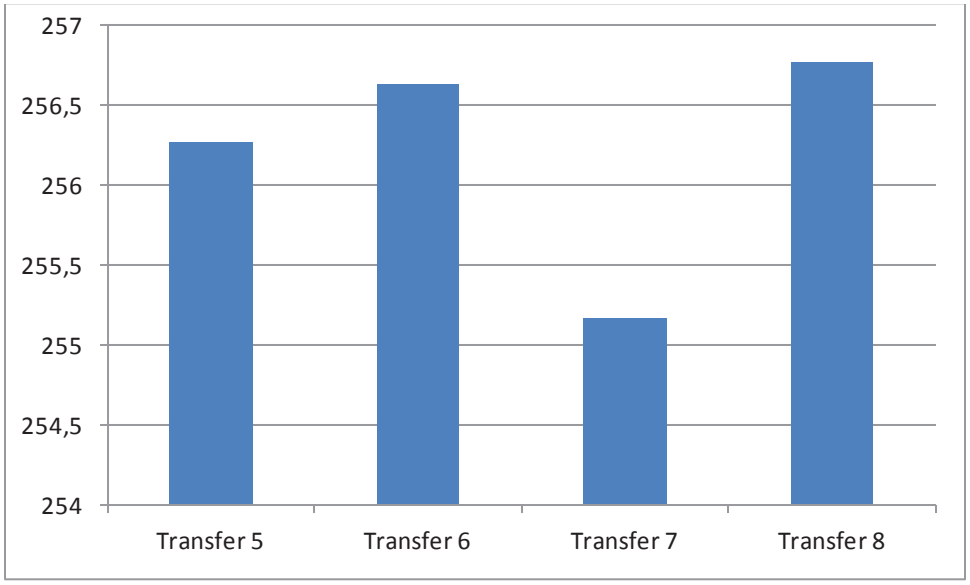
8.2 Preliminary Test 2

Thus test was made thirty times, using every transfer function, we evaluate their performance.

- T=20000
- n=30
- f max=2
- f min=0
- pulse rates=0.5
- loudness=0.25
- Instance= scp 54



Transfer function comparison



Bests transfer functions comparison

Instance	Optimum	Minimum	Maximum	Average

8.3 Preliminary Test 3

With thrifty iterations made of the BBA using transfer function 7 and parameter values used in the original algorithm proposed by Seyedali Mirjalili,· Seyed Mohammad Mirjalili and·Xin-She Yang .

- T=1000
- n=30
- f max=2
- f min=0
- pulse rates=0.5
- loudness=0.25

Instance	Optimum	Minimum	Maximum	Average
Scp 41	429	448	478	452
Scp 42	512	559	616	564
Scp 43	516	537	589	545
Scp 44	494	527	582	532
Scp 45	512	527	624	537
Scp 46	560	607	655	614
Scp 47	430	447	529	458
Scp 48	492	509	560	517
Scp 410	514	571	612	576
Scp 51	253	280	313	285
Scp 52	302	314	387	329
Scp 53	226	247	264	249
Scp 54	242	251	276	254
Scp 55	211	225	260	233
Scp 56	213	247	270	250
Scp 57	293	314	354	319
Scp 58	288	315	363	324
Scp 59	279	315	360	320
Scp 510	265	280	312	283

Instance	Optimum	Minimum	Maximum	Average
Scp 61	138	152	179	154
Scp 62	146	160	177	164
Scp 63	145	160	178	161
Scp 64	131	160	178	163
Scp 65	161	182	208	186
Scp A1	253	261	280	264
Scp A2	252	276	321	285
Scp A3	232	252	267	253
Scp A4	234	249	279	252
Scp B1	69	82	94	86
Scp B2	76	88	101	89
Scp B3	80	85	101	89
Scp B4	79	84	111	86
Scp C1	227	235	272	240
Scp C2	219	236	277	240
Scp C3	243	270	319	276
Scp C4	219	244	269	248
Scp C5	215	222	252	225
Scp D1	60	61	74	63
Scp D2	66	73	82	72
Scp D3	72	79	90	82
Scp D4	62	67	74	68
Scp D5	61	66	73	66
Scp NRE1	29	30	34	30
Scp NRE2	30	34	39	35
Scp NRE3	27	32	39	34
Scp NRE4	28	33	36	33

Scp NRE5	28	30	36	31
Scp NRF1	14	17	19	17
Scp NRF2	15	17	19	18
Scp NRF3	14	17	23	18
Scp NRF4	14	17	21	17
Scp NRF5	13	16	18	16
Scp NRH3	59	69	78	71
Scp NRH4	58	67	78	69
Scp NRH5	55	61	71	62

8.4 Preliminary Test 4

Instancia	Óptimo	Best	Transfer Function	Discrete Method	RPD
scp 41	429	448	5	4	4.42
scp 42	512	559	2, 3 & 4	4	9.17
scp 43	516	537	2, 3, 4 & 5	4	4.06
scp 44	494	527	4	4	6.68

Instancia	Óptimo	worts	Transfer Function	Discrete Method	RPD
scp 41	429	6902	2	2	1508.85
scp 42	512	8218	4	2	1505.07
scp 43	516	9316	4	2	1705.42
scp 44	494	9086	2	2	1739.27

8.5 Formal test

Every instance has been execute thirty times, with the following parameters values:

- T= 1000
- Bats=30
- alfa= 0.9
- delta= 0.9
- beta= 0.000001
- Qmin = 0
- Qmax = 2

Instance	Best	worse
scp410	pending	pending
scp51	285	297
scp52	337	361
scp53	248	253
scp54	256	269
scp55	235	248
scp56	254	264
scp57	324	339
scp58	325	345
scp59	322	343
scp510	386	397
scp61	154	160
scp62	160	164
scp63	160	165
scp64	140	143
scp65	185	194
scpa1	262	268
scpa2	285	296
scpa3	254	260
scpa4	254	266
scpa5	244	258
scpb1	Pending	pending
scpb2	98	92
scpb3	85	86

scpb4	95	93
scpb5	79	84
scpc1	239	251
scpc2	240	252
scpc3	274	294
scpc4	248	255
scpc5	pending	Pending
scpd1	63	68
scpd2	73	76
scpd3	79	81
scpd4	67	69
scpd5	66	68
scpnre1	30	31
scpnre2	pending	Pending
scpnre3	pending	Pending
scpnre4	pending	Pending
scpnre5	pending	Pending

9. Conclusions

Investigation needs to be done; the implementation of this project has different stages. The results of the implementation are strongly related to with the transfer function and the discrete function used in the algorithm., that's why we need to execute several times each test.

.

10.References

[1] Bat Algorithm: Literature Review and Applications
Xin-She Yang School of Science and Technology, Middlesex University,
The Burroughs, London NW4 4BT, United Kingdom

[2] A 2-level Metaheuristic for the Set Covering Problem
C. Valenzuela, B. Crawford, R. Soto, E. Monfroy, F. Paredes

[3] BBA: A Binary Bat Algorithm for Feature Selection
R. Y. M. Nakamura, L. A. M. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa
Department of Computing, Sao Paulo State University, Bauru, Brazil
X.-S. Yang, National Physical Laboratory, London, UK

[4] Automated Parameterisation of a Metaheuristic for the Orienteeng Problem
Wouter Souffriau, Pieter Vansteenwegen, Greet Vanden Bergue and Dick Van Oudheusden
Belgium

[5] A Study of Different Transfer Functions for Binary Version of Particle Swarm Optimization
S. Mirjalili, S.Z. Mohd Hashim, G. Taherzadeh, S.Z. Mirjalili, and S. Salehi. Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia Skudai, Johor Bahru, Malaysia Faculty of Information Technology, Multimedia University, Selangor, Malaysia Faculty of Mathematics, Sharif University of Technology, Tehran, Iran.

[6] “Binary Bat” Seyedali Mirjalili, · Seyed Mohammad Mirjalili and ·Xin-She Yang. Neural Comput & Applic DOI 10.1007/s00521-013-1525-5