Ten Artificial Human Optimization Algorithms

Satish Gajawada and Hassan Mustafa

Ten Artificial Human Optimization Algorithms

Satish Gajawada
Alumnus, Indian Institute of Technology Roorkee
Founder and Father of Artificial Human Optimization Field
satish.gajawada.iit@gmail.com

Hassan M.H. Mustafa
Faculty of Specified Education, Dept. of Educational Technology,
Banha University, Egypt
Grand Father of Artificial Human Optimization Field
prof.dr.hassanmoustafa@gmail.com

Corresponding author: Satish Gajawada

Abstract: The term "Artificial Human Optimization" was first coined by the corresponding author of this work in December 2016 when he published a paper titled "Entrepreneur : Artificial Human Optimization" at Transactions on Machine Learning and Artificial Intelligence (TMLAI) Volume 4, No 6 (December 2016). According to that paper published in 2016, Artificial Human Optimization Field is defined as the collection of all those optimization algorithms which were proposed based on Artificial Humans. In real world we (Humans) solve the problems. In the same way Artificial Humans imitate real Humans in the search space and solve the optimization problems. In Particle Swarm Optimization (PSO) the basic entities in the solution space are Artificial Birds where as in Artificial Human Optimization the basic entities in search space are Artificial Humans. Each Artificial Human corresponds to a point in the solution space. Ten Artificial Human Optimization methods titled "Human Bhagavad Gita Particle Swarm Optimization (HBGPSO)", "Human Poverty Particle Swarm Optimization (HPPSO)", "Human Dedication Particle Swarm Optimization (HuDePSO)", "Human Selection Particle Swarm Optimization (HuSePSO)", "Human Safety Particle Swarm Optimization (HuSaPSO)", "Human Kindness Particle Swarm Optimization (HKPSO)", "Human Relaxation Particle Swarm Optimization (HRPSO)", "Multiple Strategy Human Particle Swarm Optimization (MSHPSO)", "Human Thinking Particle Swarm Optimization (HTPSO)", "Human Disease Particle Swarm Optimization (HDPSO)" are applied on various benchmark functions and results obtained are shown in this work.

Keywords: Computational Intelligence, Evolutionary Computing, Artificial Humans, Artificial Human Optimization, Particle Swarm Optimization, Genetic Algorithms, Hybrid Algorithms, Global Optimization Techniques, Nature Inspired Computing, Bio-Inspired Computing, Artificial Intelligence, Machine Learning

Highlights:

1) World's First Hybrid PSO algorithm based on Human Bhagavad Gita is designed in this work.

2) World's First Hybrid PSO algorithm based on Human Poverty is designed in this work.

3) World's First Hybrid PSO algorithm based on Human Dedication is designed in this work.

4) World's First Hybrid PSO algorithm based on Human Selection is designed in this work.

5) The concept of Money is introduced into Particle Swarm Optimization algorithm for the first time in research industry history to create a new Hybrid PSO algorithm which comes under Artificial Human Optimization Field.

6) Ten Hybrid PSO algorithms which come under Artificial Human Optimization Field are shown in this work.

1 Introduction

The goal of 'Human Optimization' is to increase the performance of real humans through various methods. But 'Artificial Human Optimization' is a new field which took its birth recently in December 2016 as explained in abstract of this paper. This new filed is a sub-field of Evolutionary Computing which in turn is a sub-field of Computational Intelligence field. Hence 'Human Optimization (Real Human Optimization)' is different from Artificial Human Optimization (AHO).

The following is the review obtained from an expert in 2013 for a work under AHO Field. The review is shown below in double quotes as it is:

"The motivation of the paper is interesting. But the paper does not present any evaluation of the proposed algorithm. So we have an idea but we are not able to assess it on the basis of the paper. Next, there seems to be a difference between birds, fishes, ants, bacteria, bees etc. on one side, and human beings on the other side. Birds, fishes, ants, bacteria, bees etc. are more or less the same. People are different. I dare say that taxi drivers are different from politicians, or preschool teachers for example. Some people prefer money or power than love. It is not so difficult to guess which way ants will go but it is not so obvious when we consider people behavior. In my opinion the paper is a very first step to build the algorithm assumed but still lots of work is needed to achieve the goal."

The algorithms under Artificial Human Optimization Field (AHO Field) were proposed in literature starting from year 2003. But from the above review it is clear that the expert felt there are no algorithms under Artificial Human Optimization Field as on 2013 and corresponding author's work is the very first step. Experts are very familiar with Genetic Algorithms, Particle Swarm Optimization, Ant Colony

Optimization etc but according to corresponding author's observation many experts are unaware of the fact that there are algorithms under AHO Field before 2013. Even corresponding author of this work felt that his work submitted for review in 2013 is the beginning of Artificial Human Optimization Field Algorithms. But this was a mistake and it was corrected in later papers. It is also clear from above review shown in double quotes that imitating Humans and creating Evolutionary Computing algorithms is not as easy as imitating beings Birds, fishes, ants, bacteria, bees etc and creating algorithms under Evolutionary Computation domain.

In this work the focus is on creating new AHO Field algorithms by modifying Particle Swarm Optimization (PSO) algorithm. Articles [1-7] give an overview of existing PSO algorithms and other details. Artificial Human Optimization Algorithms that are created by modifying PSO algorithm were shown in [8-12]. Articles [13-25] gives complete details related to Artificial Human Optimization Field and its algorithms. Benchmark Functions used in this paper are taken from [26].

The rest of the article is organized as follows:
Section 2 shows Particle Swarm Optimization algorithm. Section 3 to Section 12 shows "Human Bhagavad Gita Particle Swarm Optimization (HBGPSO)", "Human Poverty Particle Swarm Optimization (HPPSO)", "Human Dedication Particle Swarm Optimization (HuDePSO)", "Human Selection Particle Swarm Optimization (HuSePSO)", "Human Safety Particle Swarm Optimization (HuSaPSO)", "Human Kindness Particle Swarm Optimization (HKPSO)", "Human Relaxation Particle Swarm Optimization (HRPSO)", "Multiple Strategy Human Particle Swarm Optimization (MSHPSO)", "Human Thinking Particle Swarm Optimization (HTPSO)", "Human Disease Particle Swarm Optimization (HDPSO)" respectively. Results are explained in Section 13. Section 14 gives Conclusions.

2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) was proposed by Kennedy and Eberhart in 1995. PSO is based on Artificial Birds. It has been applied to solve complex optimization problems.

In PSO, first we initialize all particles as shown below. Two variables $pbest_i$ and gbest are maintained. $pbest_i$ is the best fitness value achieved by i^{th} particle so far and gbest is the best fitness value achieved by all particles so far. Lines 4 to 11 in the below text helps in maintaining particle best and global best. Then the velocity is updated by rule shown in line no. 14. Line 15 updates position of ith particle. Line 19 increments the number of iterations and then the control goes back to line 4. This process of a particle moving towards its local best and also moving towards global best of particles is continued until termination criteria will be reached.

Procedure: Particle Swarm Optimization (PSO)

1) Initialize all particles
2) iterations = 0
3) **do**
4)   **for** each particle i **do**
5)   **If** (f(x_i) < f($pbest_i$)) **then**

```
6)                          pbest_i = x_i
7)            end if
8)            if ( f( pbest_i ) < f( gbest ) ) then
9)                    gbest = pbest_i
10)           end if
11)    end for
12)    for each particle i do
13)            for each dimension d do
14)                    v_{i,d} = w*v_{i,d} +
                            C_1*Random(0,1)*(pbest_{i,d} − x_{i,d})
                            + C_2*Random(0,1)*(gbest_d − x_{i,d})
15)                    x_{i,d} = x_{i,d} + v_{i,d}
17)            end for
18)    end for
19)    iterations = iterations + 1
20) while ( termination condition is false)
```

3 Human Bhagavad Gita Particle Swarm Optimization

Bhagavad Gita is a Hindu sacred text. There are no Hybrid PSO algorithms based on Bhagavad Gita till date. According to Bhagavad Gita "He who is successful is not ideal. He who failed is not ideal. Only he is ideal and revered who irrespective of success or failure stands steadfast in the pursuit of his mission". Human Bhagavad Gita Particle Swarm Optimization (HBGPSO) is designed based on this fact.

The population consists of ideal and non ideal candidates. Based on random number generated and IdealCandidateProbability, the human is classified into either ideal or non ideal candidate. Ideal candidate is not affected by success or failure and he moves in search space without any halt. So velocity and position are always updated as shown in line number 15 and 16 irrespective of anything. But this is not the case for non ideal candidate. Based on random number generated and SuccessProbability, non-ideal candidate is classified to facing either success or failure. Non ideal candidate will not update velocity and position and moves into halted state when he faces failure as shown in line number 25. He updates velocity and position when he faces success as shown in line number 21 and 22. Hence failure or success is not a matter for ideal candidate. But non ideal candidate will stop progress when he faces failure.

Procedure: Human Bhagavad Gita Particle Swarm Optimization (HBGPSO)

```
1) Initialize all particles
2) iterations = 0
3) do
4)      for each particle i do
5)              If ( f( x_i ) < f( pbest_i ) ) then
6)                      pbest_i = x_i
7)              end if
8)              if ( f( pbest_i ) < f( gbest ) ) then
9)                      gbest = pbest_i
10)             end if
```

11)   **end for**
12)   **for** each particle i **do**
13)   **if** (random(0,1) < IdealCandidateProbability) **then** // ideal candidate
14)   **for** each dimension d **do**
15) $v_{i,d} = w*v_{i,d} +$
                            $C_1*Random(0,1)*(pbest_{i,d} - x_{i,d})$
                            $+ C_2*Random(0,1)*(gbest_d - x_{i,d})$
16) $x_{i,d} = x_{i,d} + v_{i,d}$
17)   **end for**
18)   **else** // non ideal candidate
19)   **if** (random(0,1) < SuccessProbability) **then**
20)   **for** each dimension d **do**
21) $v_{i,d} = w*v_{i,d} +$
                                $C_1*Random(0,1)*(pbest_{i,d} - x_{i,d})$
                                $+ C_2*Random(0,1)*(gbest_d - x_{i,d})$
22) $x_{i,d} = x_{i,d} + v_{i,d}$
23)   **end for**
24)   **else** // non ideal candidate with failure
25)   // non ideal candidate with failure doesnot update position and velocity
26)   **end if**
27)   **end if**
28)   **end for**
29)   iterations = iterations + 1
30) **while** (termination condition is false)

4 Human Poverty Particle Swarm Optimization

There are no Hybrid PSO algorithms based on Human Poverty till date. The population consists of Rich Humans and Poor Humans. Based on random number generated and RichCandidateProbability, the human is classified into either Rich or Poor. Rich Humans have enough money to move in the search space without any halt. So velocity and position are always updated as shown in line number 15 and 16 irrespective of anything. But this is not the case for poor Humans. Based on random number generated and DonationsProbability, Poor Human is classified to having enough money to move in the search space or having insufficient money. Poor Human will not update velocity and position and moves into halted state when he doesn't have enough money as shown in line number 25. He updates velocity and position when he gets donations and has enough money to travel in search space as shown in line number 21 and 22. Hence money is not a matter for Rich Human. But Poor candidate will stop progress when he did not get sufficient money to travel in search space.

Procedure: Human Poverty Particle Swarm Optimization (HPPSO)

1) Initialize all particles
2) iterations = 0
3) **do**
4)   **for** each particle i **do**
5)   **If** (f(x_i) < f($pbest_i$)) **then**

6) $pbest_i = x_i$
7)   **end if**
8)   **if** (f($pbest_i$) < f(gbest)) **then**
9)   gbest = $pbest_i$
10)   **end if**
11)   **end for**
12)   **for** each particle i **do**
13)   **if** (random(0,1) < RichCandidateProbability) **then** // rich candidate
14)   **for** each dimension d **do**
15) $v_{i,d} = w*v_{i,d} +$
                                $C_1*Random(0,1)*(pbest_{i,d} - x_{i,d})$
                                $+ C_2*Random(0,1)*(gbest_d - x_{i,d})$
16) $x_{i,d} = x_{i,d} + v_{i,d}$
17)   **end for**
18)   **else** // poor candidate
19)   **if** (random(0,1) < DonationsProbability) **then** // poor candidate gets donations
20)   **for** each dimension d **do**
21) $v_{i,d} = w*v_{i,d} +$
                                        $C_1*Random(0,1)*(pbest_{i,d} - x_{i,d})$
                                        $+ C_2*Random(0,1)*(gbest_d - x_{i,d})$
22) $x_{i,d} = x_{i,d} + v_{i,d}$
23)   **end for**
24)   **else**
25)   // poor candidate with no donations doesnot update position and velocity
26)   **end if**
27)   **end if**
28)   **end for**
29)   iterations = iterations + 1
30) **while** (termination condition is false)

5 Human Dedication Particle Swarm Optimization

There are no Hybrid PSO algorithms based on Human Dedication till date. Based on random number generated and HumanDedicationProbability, Human is classified into either Dedicated Human or Non Dedicated Human. Dedicated Humans move faster in search space by having a high dedication factor of 0.9 as shown in line number 16. But Non Dedicated Humans have a low dedication factor of 0.1 and move slower in search space than Dedicated Humans as shown in line number 21.

Procedure: Human Dedication Particle Swarm Optimization (HuDePSO)

1) Initialize all particles
2) iterations = 0
3) **do**
4)   **for** each particle i **do**
5)   **If** (f(x_i) < f($pbest_i$)) **then**

```
6)                    pbest_i = x_i
7)          end if
8)          if ( f( pbest_i ) < f( gbest ) ) then
9)                    gbest = pbest_i
10)         end if
11)    end for
12)    for each particle i do
13)         if ( rand(0,1) < HumanDedicationProbability) // dedicated humans
14)              for each dimension d do
15)                   v_{i,d} = w*v_{i,d} +
                           C_1*Random(0,1)*(pbest_{i,d} − x_{i,d})
                           + C_2*Random(0,1)*(gbest_d − x_{i,d})
16)                   x_{i,d} = x_{i,d} + 0.9 * v_{i,d}
17)              end for
18)         else // non-dedicated humans
19)              for each dimension d do
20)                   v_{i,d} = w*v_{i,d} +
                           C_1*Random(0,1)*(pbest_{i,d} − x_{i,d})
                           + C_2*Random(0,1)*(gbest_d − x_{i,d})
21)                   x_{i,d} = x_{i,d} + 0.1 * v_{i,d}
22)              end for
23)         end if
24)    end for
25)    iterations = iterations + 1
26) while ( termination condition is false)
```

6 Human Selection Particle Swarm Optimization

There are no Hybrid PSO algorithms based on Human Selection till date. There are 2 options to select from for Humans. Either Humans move towards local best position or they move towards global best position. Based on random number generated and HumanSelectionProbability, Humans select from 2 options available. If random number generated is less than HumanSelectionProbability then Human move towards local best as shown in line number 15. Otherwise, Human move towards global best position as shown in line number 20.

Procedure: Human Selection Particle Swarm Optimization (HuSePSO)

```
1) Initialize all particles
2) iterations = 0
3) do
4)     for each particle i do
5)          If ( f( x_i ) < f( pbest_i ) ) then
6)                    pbest_i = x_i
7)          end if
```

```
8)                  if ( f( pbest_i ) < f( gbest ) ) then
9)                          gbest = pbest_i
10)                 end if
11)         end for
12)         for each particle i do
13)                 if ( rand(0,1) < HumanSelectionProbability) // moves towards local best
14)                         for each dimension d do
15)                                 v_{i,d} = w*v_{i,d} +
                                            C_1*Random(0,1)*(pbest_{i,d} − x_{i,d})
16)                                 x_{i,d} = x_{i,d} + v_{i,d}
17)                         end for
18)                 else // moves towards global best
19)                         for each dimension d do
20)                                 v_{i,d} = w*v_{i,d} +
                                            C_2*Random(0,1)*(gbest_d − x_{i,d})
21)                                 x_{i,d} = x_{i,d} + v_{i,d}
22)                         end for
23)                 end if
24)         end for
25)         iterations = iterations + 1
26) while ( termination condition is false)
```

7 Human Safety Particle Swarm Optimization

Please see [25], to understand Human Safety Particle Swarm Optimization (HuSaPSO). The code for HuSaPSO is shown below.

Procedure: Human Safety Particle Swarm Optimization (HuSaPSO)

```
1) Initialize all particles
2) iterations = 0
3) do
4)      for each particle i do
5)              If ( f( x_i ) < f( pbest_i ) ) then
6)                      pbest_i = x_i
7)              end if
8)              if ( f( pbest_i ) < f( gbest ) ) then
9)                      gbest = pbest_i
10)             end if
11)     end for
12)     for each particle i do
13)             for each dimension d do
14)                     v_{i,d} = w*v_{i,d} +
                                C_1*Random(0,1)*( x_{i,d} − pworst_{i,d})
                                + C_2*Random(0,1)*( x_{i,d} − gworst_d)
```

15) $x_{i,d} = x_{i,d} + v_{i,d}$
17)   **end for**
18)   **end for**
19)   iterations = iterations + 1
20) **while** (termination condition is false)

8 Human Kindness Particle Swarm Optimization

Please see [25], to understand Human Kindness Particle Swarm Optimization (HKPSO). The code for HKPSO is shown below.

Procedure: Human Kindness Particle Swarm Optimization (HKPSO)

1) Initialize all particles
2) iterations = 0
3) **do**
4)   **for** each particle i **do**
5)   **If** (f(x_i) < f($pbest_i$)) **then**
6) $pbest_i = x_i$
7)   **end if**
8)   **if** (f($pbest_i$) < f(gbest)) **then**
9)   gbest = $pbest_i$
10)   **end if**
11)   **end for**
12)   **for** each particle i **do**
13)   **for** each dimension d **do**
14) $v_{i,d} = w*v_{i,d} +$
                 $C_1*Random(0,1)*(pbest_{i,d} - x_{i,d})$
                 $+ C_2*Random(0,1)*(gbest_d - x_{i,d})$
15) $x_{i,d} = x_{i,d} + KindnessFactor_i * v_{i,d}$
17)   **end for**
18)   **end for**
19)   iterations = iterations + 1
20) **while** (termination condition is false)

9 Human Relaxation Particle Swarm Optimization

Please see [25], to understand Human Relaxation Particle Swarm Optimization (HRPSO). The code for HRPSO is shown below.

Procedure: Human Relaxation Particle Swarm Optimization (HRPSO)

1) Initialize all particles
2) Initialize RelaxationProbability
2) iterations = 0
3) **do**

```
4)        for each particle i do
5)                If ( f( xᵢ ) < f( pbestᵢ ) ) then
6)                        pbestᵢ = xᵢ
7)                end if
8)                if ( f( pbestᵢ ) < f( gbest ) ) then
9)                        gbest = pbestᵢ
10)               end if
11)       end for
12)       for each particle i do
13)               if  Random(0,1) < = RelaxationProbability
14)                       continue   // continues to next particle
15)               end if
16)               for each dimension d do
17)                       vᵢ,d = w*vᵢ,d +
                              C₁*Random(0,1)*(pbestᵢ,d − xᵢ,d)
                              + C₂*Random(0,1)*(gbestd − xᵢ,d)
18)                       xᵢ,d = xᵢ,d + vᵢ,d
19)               end for
20)       end for
21)       iterations = iterations + 1
22) while ( termination condition is false)
```

10 Multiple Strategy Human Particle Swarm Optimization

Please see [25], to understand Multiple Strategy Human Particle Swarm Optimization (MSHPSO). The code for MSHPSO is shown below.

Procedure: Multiple Strategy Human Particle Swarm Optimization (MSHPSO)

```
1) Initialize all particles
2) iterations = 0
3) do
4)        for each particle i do
5)                If ( f( xᵢ ) < f( pbestᵢ ) ) then
6)                        pbestᵢ = xᵢ
7)                end if
8)                if ( f( pbestᵢ ) < f( gbest ) ) then
9)                        gbest = pbestᵢ
10)               end if
11)               If ( f( xᵢ ) > f( pworstᵢ ) ) then
12)                       pworstᵢ = xᵢ
13)               end if
14)               if ( f( pworstᵢ ) > f( gworst ) ) then
15)                       gworst = pworstᵢ
16)               end if
17)       end for
```

18)   **If** ((iterations == 0) || (iterations%2==0)) **then**
              // for starting and even iterations
19)   **for** each particle i **do**
20)   **for** each dimension d **do**
21) $v_{i,d} = w * v_{i,d} +$
                    $C_1 * Random(0,1) * (pbest_{i,d} - x_{i,d})$
                    $+ C_2 * Random(0,1) * (gbest_d - x_{i,d})$
22) $x_{i,d} = x_{i,d} + v_{i,d}$
23)   **end for**
24)   **end for**
25)   **else** // for odd iterations
26)   **for** each particle i **do**
27)   **for** each dimension d **do**
28) $v_{i,d} = w * v_{i,d} +$
                    $C_1 * Random(0,1) * (x_{i,d} - pworst_{i,d})$
                    $+ C_2 * Random(0,1) * (x_{i,d} - gworst_d)$
29) $x_{i,d} = x_{i,d} + v_{i,d}$
30)   **end for**
31)   **end for**
32)   **end if**
33)   iterations = iterations + 1
34) **while** (termination condition is false)

11 Human Thinking Particle Swarm Optimization

Please see [25], to understand Human Thinking Particle Swarm Optimization (HTPSO). The code for HTPSO is shown below.

Procedure: Human Thinking Particle Swarm Optimization (HTPSO)

1) Initialize all particles
2) iterations = 0
3) **do**
4)   **for** each particle i **do**
5)   **If** (f(x_i) < f($pbest_i$)) **then**
6) $pbest_i = x_i$
7)   **end if**
8)   **if** (f($pbest_i$) < f(gbest)) **then**
9)   gbest = $pbest_i$
10)   **end if**
11)   **If** (f(x_i) > f($pworst_i$)) **then**
12) $pworst_i = x_i$
13)   **end if**
14)   **if** (f($pworst_i$) > f(gworst)) **then**

```
15)                    gworst = pworst_i
16)            end if
17)      end for
18)      for each particle i do
19)            for each dimension d do
20)                    v_{i,d} = w*v_{i,d} + Random(0,1)*(pbest_{i,d} − x_{i,d}) + Random(0,1)*(gbest_d − x_{i,d})
21)                    v_{i,d} = v_{i,d} + Random(0,1)*( x_{i,d} - pworst_{i,d} ) + Random(0,1)*( x_{i,d} - gworst_d )
22)                    x_{i,d} = x_{i,d} + v_{i,d}
23)            end for
24)      end for
25)      iterations = iterations + 1
26) while (termination condition is false)
```

12 Human Disease Particle Swarm Optimization

Please see [25], to understand Human Disease Particle Swarm Optimization (HDPSO). The code for HDPSO is shown below.

Procedure: Human Disease Particle Swarm Optimization (HDPSO)

```
1) Initialize all particles
2) iterations = 0
3) do
4)      for each particle i do
5)              If ( f( x_i ) < f( pbest_i ) ) then
6)                      pbest_i = x_i
7)              end if
8)              if ( f( pbest_i ) < f( gbest ) ) then
9)                      gbest = pbest_i
10)             end if
11)     end for

12)     If ((iterations == 0) || (iterations%2==0)) then
                // for starting and even iterations
13)             for each particle i do
14)                     for each dimension d do
15)                             v_{i,d} = w*v_{i,d} +
                                    C_1*Random(0,1)*(pbest_{i,d} − x_{i,d})
                                    +C_2*Random(0,1)*(gbest_d − x_{i,d})
16)                             x_{i,d} = x_{i,d} + v_{i,d}
17)                     end for
18)             end for
19)     else // for odd iterations
20)             for each particle i do
21)                     for each dimension d do
```

22) $v_{i,d} = w*v_{i,d} +$
            $C_1*Random(0,1)*(x_{i,d} - pbest_{i,d})$
            $+ C_2*Random(0,1)*(x_{i,d} - gbest_d)$
23) $x_{i,d} = x_{i,d} + v_{i,d}$
24)   **end for**
25)   **end for**
26)   **end if**
27)   iterations = iterations + 1
28) **while** (termination condition is false)

13 Results

Ten Artificial Human Optimization methods titled "Human Bhagavad Gita Particle Swarm Optimization (HBGPSO)", "Human Poverty Particle Swarm Optimization (HPPSO)", "Human Dedication Particle Swarm Optimization (HuDePSO)", "Human Selection Particle Swarm Optimization (HuSePSO)", "Human Safety Particle Swarm Optimization (HuSaPSO)", "Human Kindness Particle Swarm Optimization (HKPSO)", "Human Relaxation Particle Swarm Optimization (HRPSO)", "Multiple Strategy Human Particle Swarm Optimization (MSHPSO)", "Human Thinking Particle Swarm Optimization (HTPSO)", "Human Disease Particle Swarm Optimization (HDPSO)" are applied on Ackley, Beale, Bohachevsky, Booth and Three-Hump Camel Benchmark Functions and results obtained are shown in this section. The Figures of benchmark functions are taken from [26].
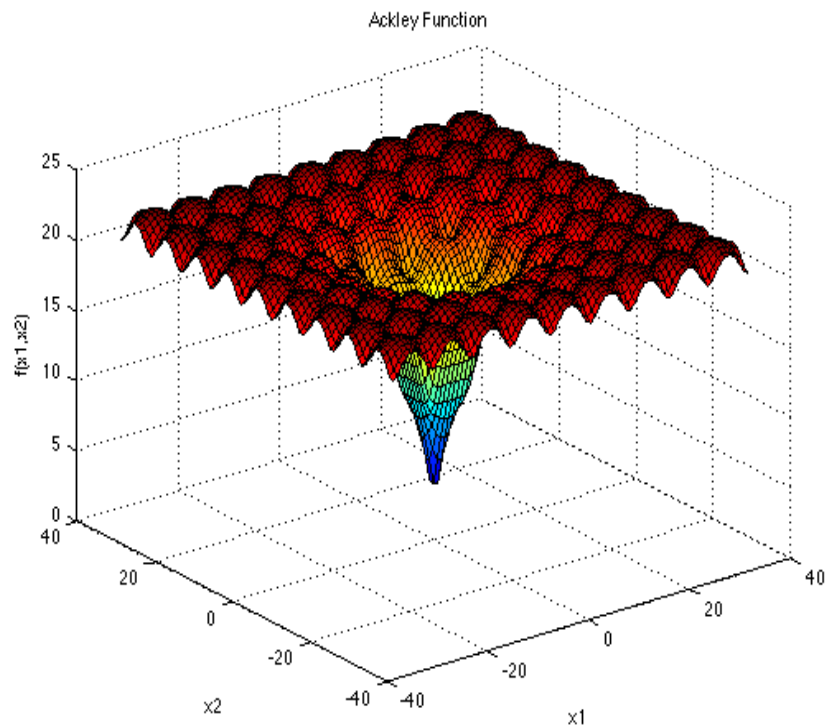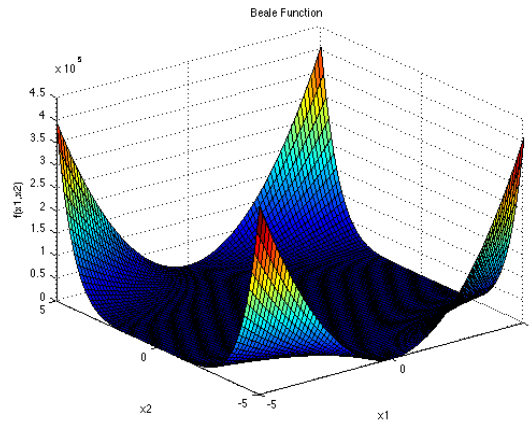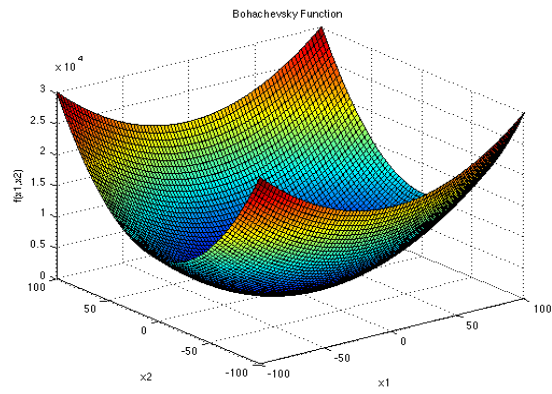

Figure 1. Ackley Function

Figure 2. Beale Function
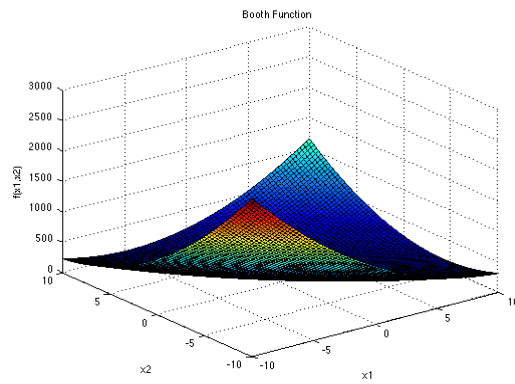

Figure 3. Bohachevsky Function

Figure 4. Booth Function
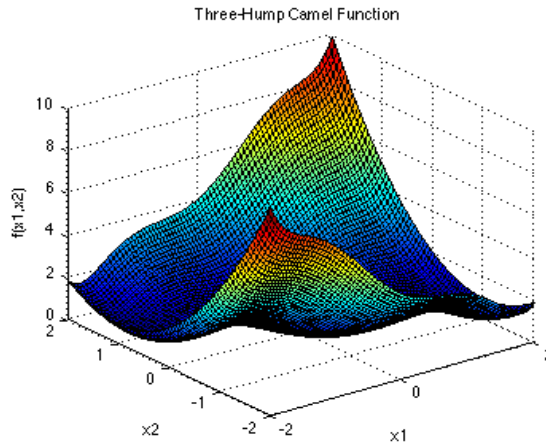
Figure 5. Three-Hump Camel Function

Benchmark Function / AHO Algorithm	PSO	HBGPSO	HPPSO	HuDePSO	HuSePSO	HKPSO	HRPSO
Ackley							
Beale							
Bohachevsky							
Booth							
Three-Hump Camel							

Figure 6. Overall Result Part One

Benchmark Function / AHO Algorithm	HuSaPSO	MSHPSO	HTPSO	HDPSO
Ackley				
Beale				
Bohachevsky				
Booth				
Three-Hump Camel				

Figure 7. Overall Result Part Two

In Figure 6 and Figure 7, first row shows AHO algorithms and first column shows benchmark functions. Green represents "Performed Well". Red represents "Didn't Performed Well". Blue represents "Performed Between Well and Not Well".

From Figure 6 it is clear that HBGPSO, HPPSO, HuDePSO, HuSePSO, HKPSO, HRPSO and PSO Performed Well for all benchmark functions.

From Figure 7 it can be observed that HuSaPSO didn't perform well even on single benchmark function. MSHPSO and HDPSO performed well on three benchmark functions. HTPSO performed well on only single benchmark function.

14 Conclusions

Artificial Human Optimization Algorithms (AHO Algorithms) inspired by Bhagavad Gita (HBGPSO), Human Poverty (HPPSO), Human Dedication (HuDePSO) and Human Selection (HuSePSO) are proposed in this work. Ten AHO algorithms are applied on 5 benchmark functions and results obtained are shown in this work. Six AHO algorithms performed as good as PSO algorithm where as remaining four AHO algorithms didn't performed as good as PSO. HuSaPSO performed worst among all algorithms used in this work. All algorithms designed in this work performed as good as PSO. A general misunderstanding among people is that algorithms inspired by Humans will perform better than other algorithms inspired by other beings. For example, let algorithm A is inspired by Birds and Algorithm B is inspired by Humans. Then because of misunderstanding, it will lead to conclusion that Algorithm B performs better than Algorithm A because Humans are best beings and most intelligent beings on this planet. In this work, we have found that HuSaPSO inspired by Humans did not performed well even on single benchmark function where as PSO inspired by birds performed well on all benchmark functions. Our future work is to design "Human Cricket Particle Swarm Optimization (HCPSO)", "Human Farming Particle Swarm Optimization (HFPSO)" inspired by Human Cricket game and Human Farming respectively. Artificial Human Optimization Algorithms designed from scratch will also be part of our future work.

References

[1] Saptarshi Sengupta, Sanchita Basak, Richard Alan Peters II. Particle Swarm Optimization: A survey of historical and recent developments with hybridization perspectives. https://arxiv.org/abs/1804.05319, 2018.

[2] Yudong Zhang, Shuihua Wang, and Genlin Ji, "A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications," Mathematical Problems in Engineering, vol. 2015, Article ID 931256, 38 pages, 2015. https://doi.org/10.1155/2015/931256.

[3] M. R. AlRashidi, M. E. El-Hawary. A Survey of Particle Swarm Optimization Applications in Electric Power Systems. IEEE Transactions on Evolutionary Computation. Volume 13, Issue 4, August 2009.

[4] Sharandeep Singh. A Review on Particle Swarm Optimization Algorithm. International Journal of Scientific & Engineering Research, Volume 5, Issue 4, April-2014.

[5] T. Saravanan and V. Srinivasan. Overview of Particle Swarm Optimization. Indian Journal of Science and Technology, Vol 8(32), November 2015.

[6] Muhammad Imran, Rathiah Hashim, Noor Elaiza Abd Khalid.An Overview of Particle Swarm Optimization Variants. Procedia Engineering. Elsevier.Volume 53, Pages 491-496, 2013.

[7] Riccardo Poli, James Kennedy, Tim Blackwell. Particle swarm optimization - An overview. Swarm Intelligence. Volume 1, Issue 1, pp 33–57, Springer, 2007.

[8] Liu H, Xu G, Ding GY, Sun YB, "Human behavior-based particle swarm optimization", The Scientific World Journal, 2014.

[9] Ruo-Li Tang, Yan-Jun Fang, "Modification of particle swarm optimization with human simulated property", Neurocomputing, Volume 153, Pages 319–331, 2015.

[10] Muhammad Rizwan Tanweer, Suresh Sundaram, "Human cognition inspired particle swarm optimization algorithm", 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014.

[11] M.R. Tanweer, S. Suresh, N. Sundararajan, "Self regulating particle swarm optimization algorithm", Information Sciences: an International Journal, Volume 294, Issue C, Pages 182-202, 2015.

[12] M. R. Tanweer, S. Suresh, N. Sundararajan, "Improved SRPSO algorithm for solving CEC 2015 computationally expensive numerical optimization problems", 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 1943-1949, 2015.

[13] Satish Gajawada. "POSTDOC : The Human Optimization", Computer Science & Information Technology (CS & IT), CSCP, pp. 183-187, 2013.

[14] Satish Gajawada. "CEO: Different Reviews on PhD in Artificial Intelligence", Global Journal of Advanced Research, vol. 1, no.2, pp. 155-158, 2014.

[15] Satish Gajawada. "Entrepreneur: Artificial Human Optimization". Transactions on Machine Learning and Artificial Intelligence, Volume 4 No 6 December (2016); pp: 64-70

[16] Satish Gajawada. "Artificial Human Optimization – An Introduction", Transactions on Machine Learning and Artificial Intelligence, Volume 6, No 2, pp: 1-9, April 2018

[17] Satish Gajawada. "An Ocean of Opportunities in Artificial Human Optimization Field", Transactions on Machine Learning and Artificial Intelligence, Volume 6, No 3, June 2018

[18] Satish Gajawada. "25 Reviews on Artificial Human Optimization Field for the First Time in Research Industry", International Journal of Research Publications, Vol. 5, no. 2, United Kingdom.

[19] Satish Gajawada and Hassan M. H. Mustafa, "Collection of Abstracts in Artificial Human Optimization Field", International Journal of Research Publications, Volume 7, No 1, United Kingdom, 2018.

[20] Satish Gajawada, Hassan M. H. Mustafa, HIDE : Human Inspired Differential Evolution - An Algorithm under Artificial Human Optimization Field , International Journal of Research Publications (Volume: 7, Issue: 1), http://ijrp.org/paper-detail/264

[21] Satish Gajawada, Hassan M. H. Mustafa , Artificial Human Optimization – An Overview. Transactions on Machine Learning and Artificial Intelligence, Volume 6, No 4, August 2018.

[22] Satish Gajawada, Hassan M. H. Mustafa, Testing Multiple Strategy Human Optimization based Artificial Human Optimization Algorithms, Computer Reviews Journal, vol. 1, no.2, 2018.

[23] Satish Gajawada, Hassan M. H. Mustafa. Hybridization Concepts of Artificial Human Optimization Field Algorithms Incorporated into Particle Swarm Optimization. *International Journal of Computer Applications* 181(19):10-14, September 2018.

[24] Satish Gajawada, Hassan M. H. Mustafa (2018). An Artificial Human Optimization Algorithm Titled Human Thinking Particle Swarm Optimization. *International Journal of Mathematical Research*, 7(1): 18-25. DOI: 10.18488/journal.24.2018.71.18.25

[25] Satish Gajawada, Hassan Mustafa: Novel Artificial Human Optimization Field Algorithms - The Beginning. CoRR abs/1903.12011(2019)

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