A Neighborhood Centroid Opposition-Based Grasshopper Optimization Algorithm

Ling Liao and Yongquan Zhou*
College of Information Science and Engineering, Guangxi University for Nationalities, Nanning, China

*Corresponding author e-mail: yongquanzhou@126.com

Abstract: Grasshopper Optimization algorithm (GOA) is proposed in recent years as an intelligent optimization algorithm. GOA has the advantages of simple structure, less parameters, the characteristics of rapid convergence, but it's also need to further improve the population diversity and convergence precision. Therefore, for improve the development and exploration and exploitation ability of my algorithm, my paper proposed a Neighborhood Centroid Opposition-Based Grasshopper Optimization algorithm (NCOGOA). In NCOGOA, population is divided into multiple areas, direct interaction between the individual and to calculate reference point reverse point, for sufficient the group of search experience while maintaining the diversity of population. In this paper, NCOGOA is verified by 3 reference functions. Compared with five different classical algorithms, the experimental results find that GOA can get more accurate solutions than other group algorithms and with fast convergence and good stability.

1. Introduction
Nowadays, so many different heuristic optimization methods have been established. Compared with the traditional method, it has better problem solving and different problems. For instance, in the field of optimization, the Netherlands proposed a revolutionary idea to simulate the concept of evolution in the computer to solve the fact problem [1]. That is the most respected heuristic algorithm known as Genetic Algorithm (GA) [2]. Inspired by this way, so many optimization methods has been proposed including the PSO( particle swarm optimization) [3] and others are applied widely. In last two decades, some new population based algorithms were proposed, such as CS( cuckoo search) [4], BA( bat algorithm)[5] and so on.

2. Neighborhood Centroid Opposition-Based Grasshopper Optimization algorithm

2.1. Grasshopper Optimization algorithm (GOA)
The grasshopper optimization algorithm imitate the swarm feeding behavior of grasshopper under field conditions for solve the practical problems in life or some function optimization issues by establishing mathematical model. The network can constraint the grasshopper that enable each grasshopper location to update to the best of the target location, and the grasshopper can through the network of group consciousness to decide the direction of the position update. Fig.1 depicts the interaction between individual grasshopper. There are repulsion and attraction force in the grasshopper swarm. The repulsive force is mainly to let this grasshoppers increase the detection capability, while
attraction encourages them to develop the optimal value of the current region to increase the mining capacity. These two forces are equal when the repulsive force and the attraction are offset. Finally, this grasshoppers gathered together to find the best food source. This process is mathematically described as follows.

\[ X_i = S_i + G_i + A_i \]  \hspace{1cm} (1)

where \( X_i \) is the place of the \( i-th \) grasshopper, \( S_i \) is the social attraction between different grasshoppers, \( G_i \) is the earth gravity of the \( i-th \) grasshopper, and \( A_i \) is the power of the wind with the \( i-th \) grasshopper.

\[ S_i = \sum_{j=1, j \neq i}^{N} s(d_{ij}) \vec{d}_{ij} \]  \hspace{1cm} (2)

where \( d_{ij} \) is the interval between \( i-th \) and \( j-th \) grasshopper and it is counted with \( d_{ij} = |X_j - X_i| \), and \( s \) is the function defining the strength or forces with different grasshoppers as exhibition in the formula (2.3), and the vector \( \vec{d}_{ij} = \frac{x_j - x_i}{d_{ij}} \) is defined a vector between \( i-th \) grasshopper and \( j-th \) grasshopper.

This function of \( S \) is consider to the social power and the value of this function varies with distance , and it is calculated as this:

\[ s(d) = f e^{-d/l} - e^{-d} \]  \hspace{1cm} (3)

Where, \( f \) represents the coefficient of gravitational strength, and \( l \) represents the scale of the cited length. The function \( s \), shown in Fig.1, shows how it affects the social interaction (attraction and exclusion) of grasshoppers.Fig.2 shows that the repulsive force is encouraged within the distance of [0,2.079].

Considering the gravity effect of the grasshoppers and assuming that the wind or the direction of the air flow is always in the direction of the target position. The improved mathematical model is as follows:

\[ X_d^i (t + 1) = c \left\{ c \cdot \frac{ub_d - lb_d}{2} \cdot s \left[ X_j^d (t) - X_i^d (t) \right] \cdot \frac{x_j(t) - x_i(t)}{d_{ij}} \right\} + \bar{T}_d \]  \hspace{1cm} (4)

where \( t \) is the count of the iteration, and the parameter \( c \) is the decreasing coefficient, which is used to diminution the comfort space, repulsion space and the attraction space, \( ub_d \) is the upper boundary, \( lb_d \) is the lower boundary, \( \bar{T}_d \) is the best solution at present. \( s \) is consider to the power of social forces including reputation and attraction with different grasshoppers. In this algorithm, the parameter \( c \) plays a critical role, when the \( c \) is very huge, exploration ability has played a leading role to increase the local search ability. When \( c \) is small, exploitation capacity has played a leading role to increase the global searching ability. Therefore, the selection of parameter \( c \) is very important in this algorithm. The parameter \( c \) is defined as the following:

\[ c(t) = C_{\text{max}} - t \cdot \frac{C_{\text{max}} - C_{\text{min}}}{t_{\text{max}}} \]  \hspace{1cm} (5)

where \( C_{\text{max}} \) and \( C_{\text{min}} \) are the numerical of the coefficient \( c \) with maximum and the minimum values, and the \( t_{\text{max}} \) is the high number of iteration.

More details about the algorithm of GOA could be defined in [6]. The pseudo code of GOA is described in algorithm 1.
Figure 1. Model of interaction between grasshoppers groups.

Algorithm 1. Pseudo codes of the GOA

1. Initialize grasshoppers $X_i$
2. Initialize coefficients $C_{\text{max}}, C_{\text{min}}, \text{iterations}$
3. Count the value of the search agent $B$ equal the first-rate value
   \begin{algorithmic}
   \STATE \textbf{while}(i<\text{Max iterations})
   \STATE \quad \text{Update coefficient c using the equation(5)}
   \STATE \quad \quad \textbf{for} different agent
   \STATE \quad \quad \quad Standardize the distances with different grasshopper in $[1, 4]$
   \STATE \quad \quad \quad According to formula (4) update t location
   \STATE \quad \quad \quad If the current search agent exceeds the bounds, update the bounds again
   \STATE \quad \textbf{end for}
   \STATE \quad \text{Update } B \text{ according the better solution at present}
   \STATE \quad \quad \quad l = l + 1
   \STATE \textbf{end while}
   \STATE \text{Return } B
   \end{algorithmic}

2.2. Neighborhood Centroid Opposition-Based Learning (NCOL)

The Opposite-Based Learning (OBL) is a kind of intellectual technique that Tizhoosh has proposed. The master wants to think about the same time when the previous solution and its reverse solution are used, and the selection is used to add speed to the search [12]. The traditional Opposition-based Learning method is to use the maximum and minimum boundary to calculate the opposition point. The Opposite-Based Learning has obtained better results in some algorithms. But the OBL does not take advantage of the whole population's search information. So, Rahnamayan and others proposed the Centroid Opposition-Based Learning (COBL) strategies, The opposition point is calculated with the focus of the whole group as reference point, so that the reverse point contains group search experience. COBL advantage mainly lies in the calculation of the opposition point of reference point is the center of gravity, it carries the group search experience, and overcome the disadvantages of with,
however, COBL used to calculate the opposition solution of the reference point only a center of gravity, did not consider promoting the diversity of reverse solution. Therefore, COBL has further room for improvement.

If multiple center of gravity can be used to calculate the opposition solution, then search experience in the use of a group simultaneously also can maintain the diversity of the group, in the local PSO[14], agent according to the topological structure is divided into many fields, in the field of individual directly influence each other, and then this influence by adjacent individual spread to other areas. The advantage of this approach is that small groups of different areas can search different areas of the problem space. Reverse calculation, therefore, in the field of gravity field calculation a center of gravity, individual areas of center of gravity as the reference point in the field of computational opposition solution, so that you can make full use of the small group search experience, at the same time, there are multiple reference points, calculate the opposition solution can search more search space. Searching the center of gravity and the opposition center of gravity is consider to following:

**Definition 1:** the center of gravity, set \((X_1, X_2, \ldots, X_n)\) is the n point with unit mass in dimension \(D\) search space, then the whole center of gravity is defined as:

\[
M = \frac{X_1 + X_2 + \cdots + X_n}{n}
\]  

(6)

We can also represent this formula in this form as:

\[
M_j = \frac{\sum_{j=1}^{D} x_j}{n}, i = 1, 2, \ldots, n
\]

(7)

where \(D\) is the dimension, \(M_j\) is the whole center of gravity in \(D\) dimension search space.

**Definition 2:** the opposition center of gravity, if the center of mass of a discrete uniform is \(M\), then the opposition point of \(X_i\) in the whole is defined as:

\[
\tilde{X}_i = 2 \times M - X_i, i = 1, 2, \ldots, n
\]

(8)

The inverted point is located in a search space with dynamic boundaries, the dynamic boundaries is \(x_j \in [a_j, b_j]\), and the dynamic boundary can allow the opposition point to be located in a shrinking search space. Dynamic boundary is calculated by equation (9).

\[
a_j = \min(x_j)
\]

\[
b_j = \max(x_j)
\]

(9)

When the opposition point exceeds the boundary, the reverse point is recomputed by equation

\[
\tilde{x}_{i,j} = \begin{cases} 
  a_j + \text{rand}(0,1) \times (M_j - a_j), & \text{if } \tilde{x}_{i,j} < a_j \\
  M_j + \text{rand}(0,1) \times (b_j - M_j), & \text{if } \tilde{x}_{i,j} > b_j
\end{cases}
\]

(10)

**Definition 3:** Neighborhood Centroid Opposition-Based Point, \(X_i\) is the \(i\)-th individual in the group, and \(M_i\) is the center of gravity in its field, and the neighborhood centroid opposition Point is:

\[
\bar{X}^*_i = 2 \times k \times M_i - \bar{X}_i, i = 1, 2, \ldots, n
\]

(11)

Where \(M_j\) is the number of agent in the neighborhood, \(k\) is A random digit that follows a uniform distribution from the \([0, 1]\). The code in algorithm 2 shows the pseudo code of NCOGOA.

|| Algorithm 2. Pseudo codes of the NCOGOA algorithm |
|-----------------|-----------------|
| Initialize grasshoppers \(X_i\) |
| Initialize coefficients \(C_{\text{max}}, C_{\text{min}}, \text{iterations}\) |
| Count the value of the search agent |
while (i < Max iterations)
   for different agent
      Update coefficient c using the equation (5)
      Standardize the distances with different grasshopper in $[1, 4]$.
      According to formula (4), update $t$ location
      If the current search agent exceeds the bounds, update the bounds again
   end for
   Update $B$ according to the better solution at present
   Calculate the neighborhood centroid $M$
   Calculate opposite position $OX$ according to (8)
   Update the dynamic interval boundaries according to (9)
   Check for boundary constraint violations according to (10)
   Calculate the fitness of opposite position $f(OX)$
   for $i = 1$ to $N$
      if $f(OX_i) < f(X_i)$ then $X_i = OX_i$
   end if
   $l = l + 1$
end while
Return $B$

3. Simulation Experiments and Result Analysis

At this part, 4 standard trial functions [9, 10] are used to examine the property of NCOGOA.

| Benchmark Test Functions | Dimension | Range     | $f_{min}$ |
|-------------------------|-----------|-----------|-----------|
| $f_{01} = \sum_{i=1}^{n} x_i^2$ | 30        | $[100,100]$ | 0         |
| $f_{02}(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$ | 30        | $[-10,10]$  | 0         |
| $f_{03}(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$ | 30        | $[100,100]$ | 0         |

3.1. Experimental setup
The experimental part of the NCOGOA was tested in MATLAB R2012a, and experiments environment were conducted on 3.30ghz Intel(R) Core(TM) i5-4590 processor and 4GB RAM.

3.2. Algorithm performance comparison
The NCOGOA contrast with different swarm intelligence optimization algorithms including FPA [11], CS [4], PSO [10], GWO [12], mean and standard deviation were used to compare the optimal performance of different algorithms. The population number is set to 30, the maximum iterations is 500, and 30 independent runs. The parameters of different algorithms are set as follows:
- FPA: $p = 0.8$; CS: $\beta = 1.5, \rho_0 = 1.5$
- PSO: weight factor $\omega = 0.6, c_1 = c_2 = 1.5$
- GWO: Linearly $\alpha$ decreased from 2 to 0. GOA: $c = 0.0004$

Table 2 shows the comparison of test results of standard test functions.
Table 2. The test functions consequence $f_i, i = 1,2,3,4,5,6$.

| Benchmark functions | Method | Best       | Mean       | Worst       | Std.       | Rank |
|----------------------|--------|------------|------------|-------------|------------|------|
| $f_{01}$ (D=30)      | FPA    | 4.5000E-01 | 1.3600E+00 | 7.6700E-01 | 1.9800E-01 | 3    |
|                     | CS     | 9.7600E-02 | 3.5500E-01 | 1.9100E-01 | 5.5800E-02 | 2    |
|                     | PSO    | 2.3300E-02 | 2.7400E+01 | 4.6500E+00 | 5.9500E+00 | 4    |
|                     | GWO    | 2.6087E+02 | 4.6019E+02 | 6.9562E+02 | 1.4430E+02 | 6    |
|                     | GOA    | 1.1164E+01 | 9.3242E+01 | 3.8235E+01 | 2.2509E+01 | 5    |
|                     | NCOGOA | 0          | 0          | 0           | 0          | 1    |
| $f_{02}$ (D=30)      | FPA    | 2.7900E+00 | 4.6000E+00 | 3.7000E+00 | 4.3100E-01 | 3    |
|                     | CS     | 1.1150E-03 | 2.4370E-03 | 4.6290E-03 | 1.0600E-03 | 2    |
|                     | PSO    | 1.9600E-02 | 2.8200E+01 | 2.5100E+00 | 5.9500E+00 | 5    |
|                     | GWO    | 1.9271E+00 | 3.4197E+00 | 7.8357E+00 | 1.3257E+00 | 4    |
|                     | GOA    | 4.6795E+00 | 1.1282E+02 | 1.9694E+01 | 2.6829E+01 | 6    |
|                     | NCOGOA | 0          | 0          | 0           | 0          | 1    |
| $f_{03}$ (D=30)      | FPA    | 2.6500E-01 | 9.3500E-01 | 6.0900E-01 | 1.8500E-01 | 2    |
|                     | CS     | 3.3300E+00 | 1.0400E+01 | 5.5700E+00 | 1.8600E+00 | 3    |
|                     | PSO    | 3.8311E+03 | 5.7758E+03 | 7.6849E+03 | 1.3714E+03 | 5    |
|                     | GWO    | 1.0031E+03 | 2.3714E+03 | 4.1487E+03 | 7.3049E+02 | 4    |
|                     | GOA    | 3.6995E+02 | 4.1232E+04 | 7.0364E+03 | 8.4214E+03 | 6    |
|                     | NCOGOA | 0          | 0          | 0           | 0          | 1    |

Fig 3. $dim = 30$, The fitness value converges of $f_{01}$

Fig 4. $dim = 30$, The fitness value converges of $f_{02}$

Fig 5. $dim = 30$, The fitness value converges of $f_{03}$

From the Figs.3-5, we can quickly find that NCOGOA converge is very fast compare other algorithms.

4. Conclusions and Future Works

This paper that a Neighborhood Centroid Opposition-Based Grasshopper Optimization algorithm (NCOGOA) was proposed. Compared the results from different benchmark functions, Compared with other classical algorithms in this paper, NCOGOA perform a relatively good performance. NCOGOA
has fast convergence speed and with a relatively high stability. It is also more accurate than others. For the proposed NCOGOA, some practical problems in life can be solved, such as travel salesman problem, vehicle scheduling problem, and orbital allocation problem and others. For assess the property of NCOGOA, a classical problem which NP-hard problems should be solved tomorrow.

References
[1]. Holland, John H, Reitman, Judith S. Cognitive systems based on adaptive algorithms. Pattern-Directed Inference Systems, 1978, 63(63):313-329.
[2]. Tsai CF, Eberle W, Chu C Y. Genetic algorithms in feature and instance selection. Elsevier Science Publishers B. V. 2013, 39(2):240-247.
[3]. Kennedy J, Eberhart R. Particle Swarm Optimization. IEEE International Conference on Neural Networks, 1995, 4: 1942-1948.
[4]. Yang X S, Deb S. Cuckoo Search via Levy Flights. Mathematics, 2010:210 – 214.
[5]. Yang X S. A New Metaheuristic Bat-Inspired Algorithm. Computer Knowledge & Technology, 2010, 284:65-74.
[6]. Tizhoosh H R. Opposition-Based Learning: A New Scheme for Machine Intelligence. International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce Vol. IEEE Computer Society, 2005,1:695-701.
[7]. Rahnamayan S, Jesuthasan J, Bourennani F, et al. Computing opposition by involving entire population. Evolutionary Computation. IEEE, 2014:1800-1807.
[8]. Kennedy J, Mendes R, Population structure and particle swarm performance. Congress on Evolutionary Computation, 2012, 1671-1676.
[9]. M. Fesanghary, M. Mahdavi, M. Minary-Jolandan and Y. Alizadeh, Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems, Computer Methods in Applied Mechanics and Engineering, 2008,197 (33-34):3080-3091.
[10]. Kaveh A, Talatahari S. Engineering optimization withhybrid particle swarm and ant colony optimization. Asian Journal of Civil Engineering, 2009, 10:611-628.
[11]. Yang X S. Flower Pollination Algorithm for Global Optimization. International Conference on Unconventional Computation and Natural Computation. Springer-Verlag, 2012, 7445: 240-249.
[12]. Mirjalili S, Mirjalili SM, Lewis A. Grey Wolf Optimizer. Advances in Engineering Software, 2014, 69(3):46-61.