Grasshopper Optimisation Algorithm

(GOA)

By
Shrist Das(B417042)
Shubham Sharma(B417043)
Siddarth Kumar(B417044)

INTRODUCTION

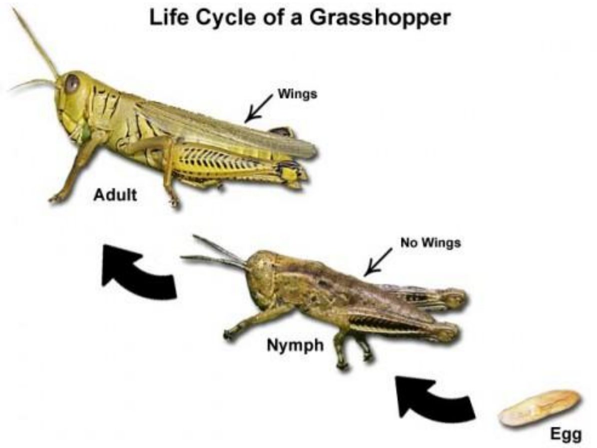
- Recent years brought significant advances in the field of nature-inspired optimization. Several new algorithms have been proposed - aimed at tackling both continuous, combinatorial and multiobjective optimization problems.
- Grasshopper Optimization Algorithm (GOA) is an optimization technique introduced by Saremi, Mirjalili and Lewis in 2017 . It includes both social interaction between ordinary agents (grasshoppers) and the attraction of the best individual.
- GOA is a population based method
- GOA mimicking the behavior of grasshopper swarms and their social interaction.

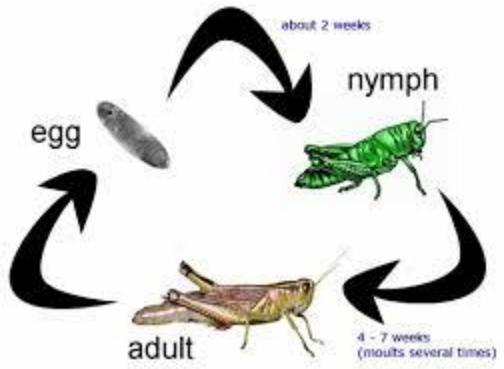
INTRODUCTION

- Nature inspired swarm based algorithms are most popular among stochastic optimization approaches. Creatures in nature uses different techniques, which are used in such optimization techniques.
- The main aim of all creatures in nature is to survive and to achieve this goal they intend to evolve and modify as well as adapt different ways.
- So nature is the best inspiration as it is the best optimizer on the planet. These algorithms are of two types: (a) Single solution based, (b) Multi solution based.
- In single solution based type a single random solution is generated and improvised further while in multi solution based type multiple solutions are generated and modified. Usually multi solution based algorithms are chosen over single solution based algorithm

Nature behavior of grasshopper

- Grasshoppers are destructive insects according to their damage to agriculture.
- They life has two phases, nymph and adulthood.
- The nymph grasshoppers have no wings so they move slowly and eat all vegetation on their path.
- After period of time they grow up and become adult with wings to form a swarm in the air and move fast to large scale region.





Nature behavior of grasshopper

- Although grasshoppers are usually seen individually in nature, they join in one of the largest swarm of all creatures.
- The size of the swarm may be of continental scale and a nightmare for farmers.
- The unique aspect of the grasshopper swarm is that the swarming behavior is found in both nymph and adulthood.

The main characteristic of the grasshopper swarm.

- The main characteristic of the swarm in the larval phase is slow movement and small steps of the grasshoppers.
- Long-range and abrupt movement is the essential feature of the swarm in adulthood.
- Food source seeking is another important characteristic of the swarming of grasshoppers by dividing the search process into two tendencies exploration and exploitation.

The three forces on grasshopper can be shown as

$X_i = S_i + G_i + A_i$

S_i is the social interaction

$$S_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} s(d_{ij})\,\widehat{d_{ij}}$$

Social forces

$$s(r) = f e^{-\frac{r}{l}} - e^{-r}$$

Gravity force

$$G_i = -g\widehat{e_g}$$

Wind advenction

$$A_i = u\widehat{e_w}$$

Then formula changes to

$$X_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} s(|x_j - x_i|)\frac{x_j - x_i}{d_{ij}} - g\widehat{e_g} + u\widehat{e_w}$$

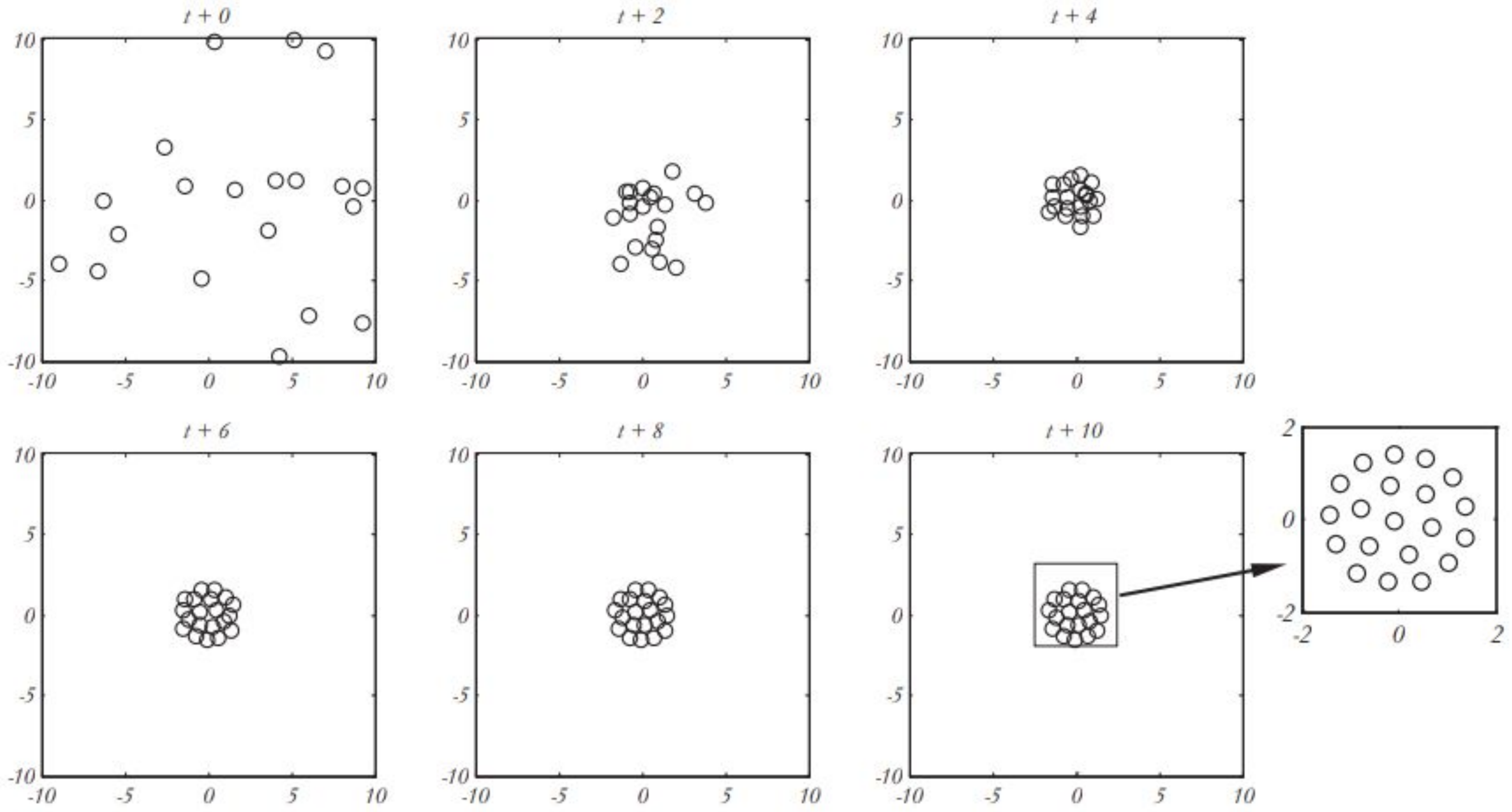


Fig. 5. Behaviour of swarm in a 2D space.

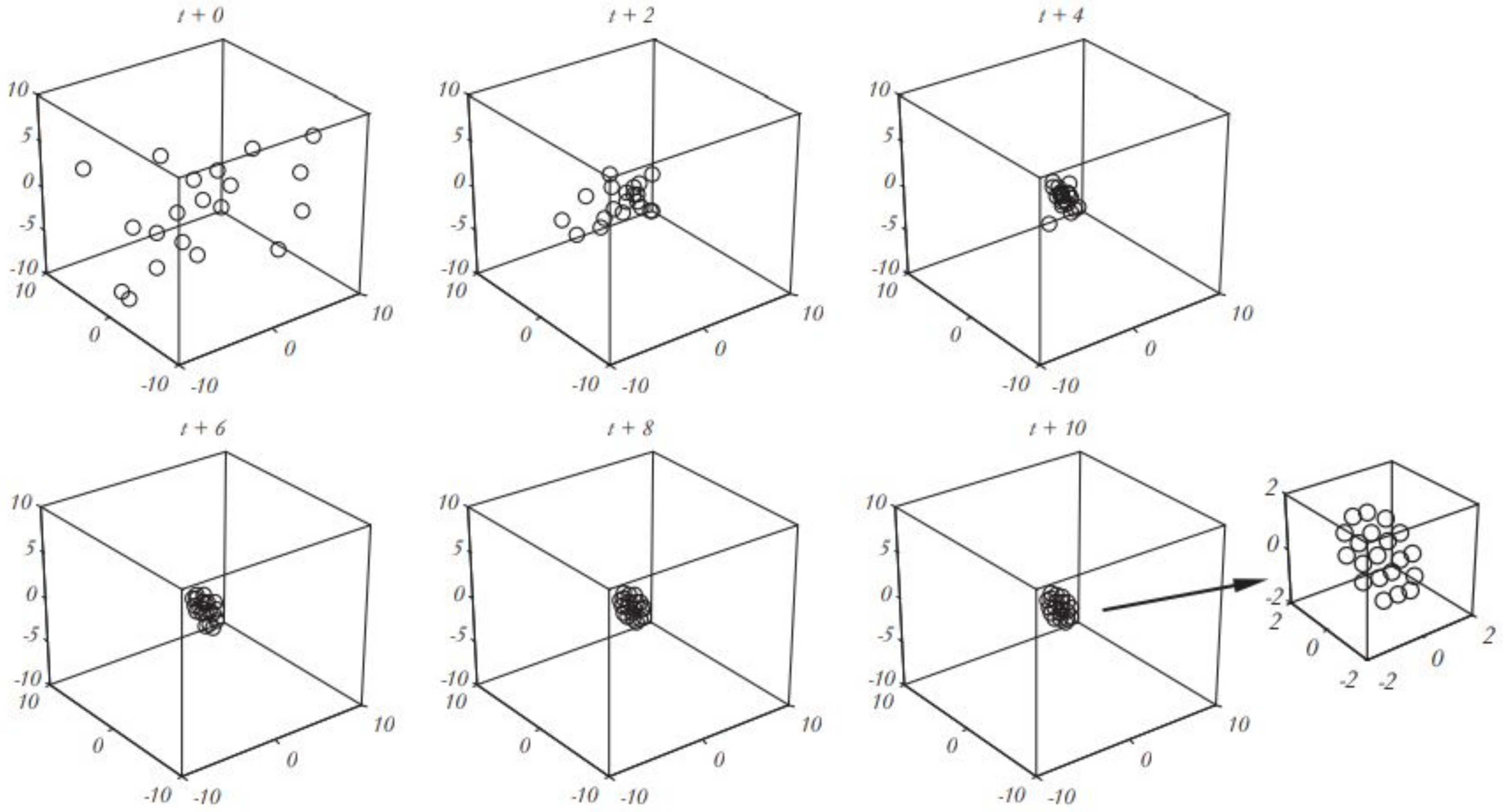


Fig. 6. Behaviour of swarm in a 3D space.

After modification

$$X_i^d = c\left(\sum_{\substack{j=1\\ j\neq i}}^{N} c\frac{ub_d - lb_d}{2} s\left(\left|x_j^d - x_i^d\right|\right)\frac{x_j - x_i}{d_{ij}}\right) + \widehat{T}_d$$

The value of c need to be changed.

$$c = cmax - l\frac{cmax - cmin}{L}$$

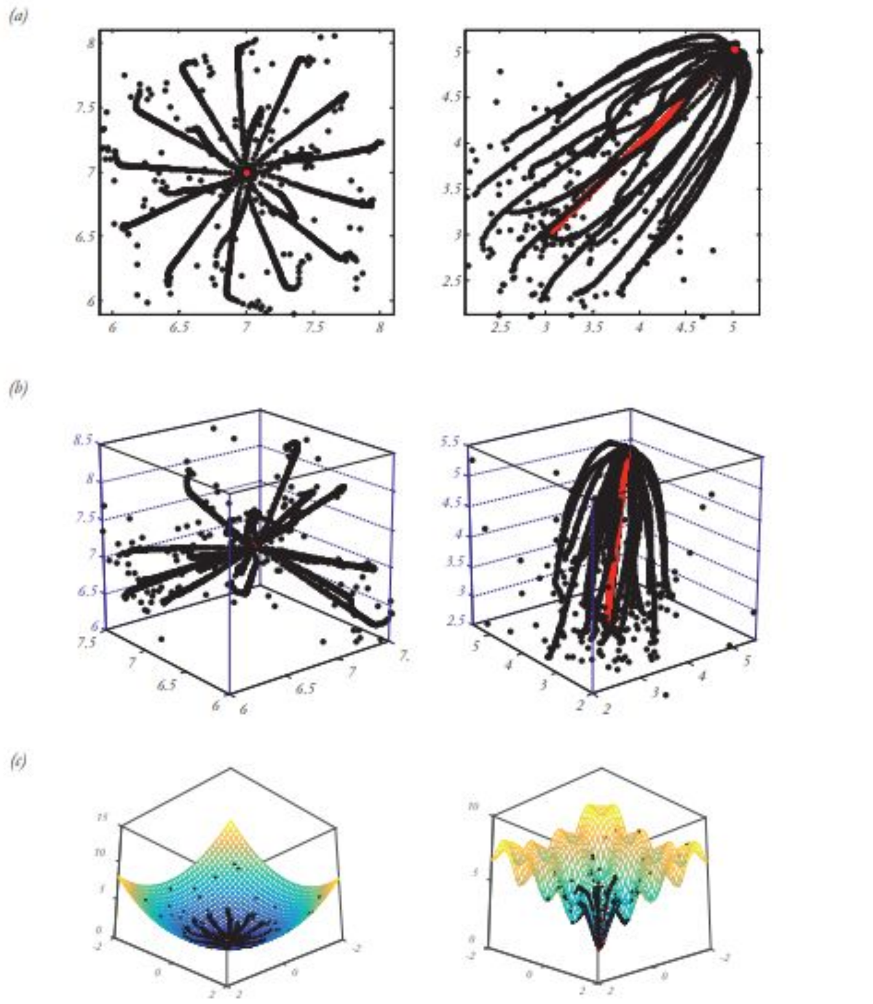
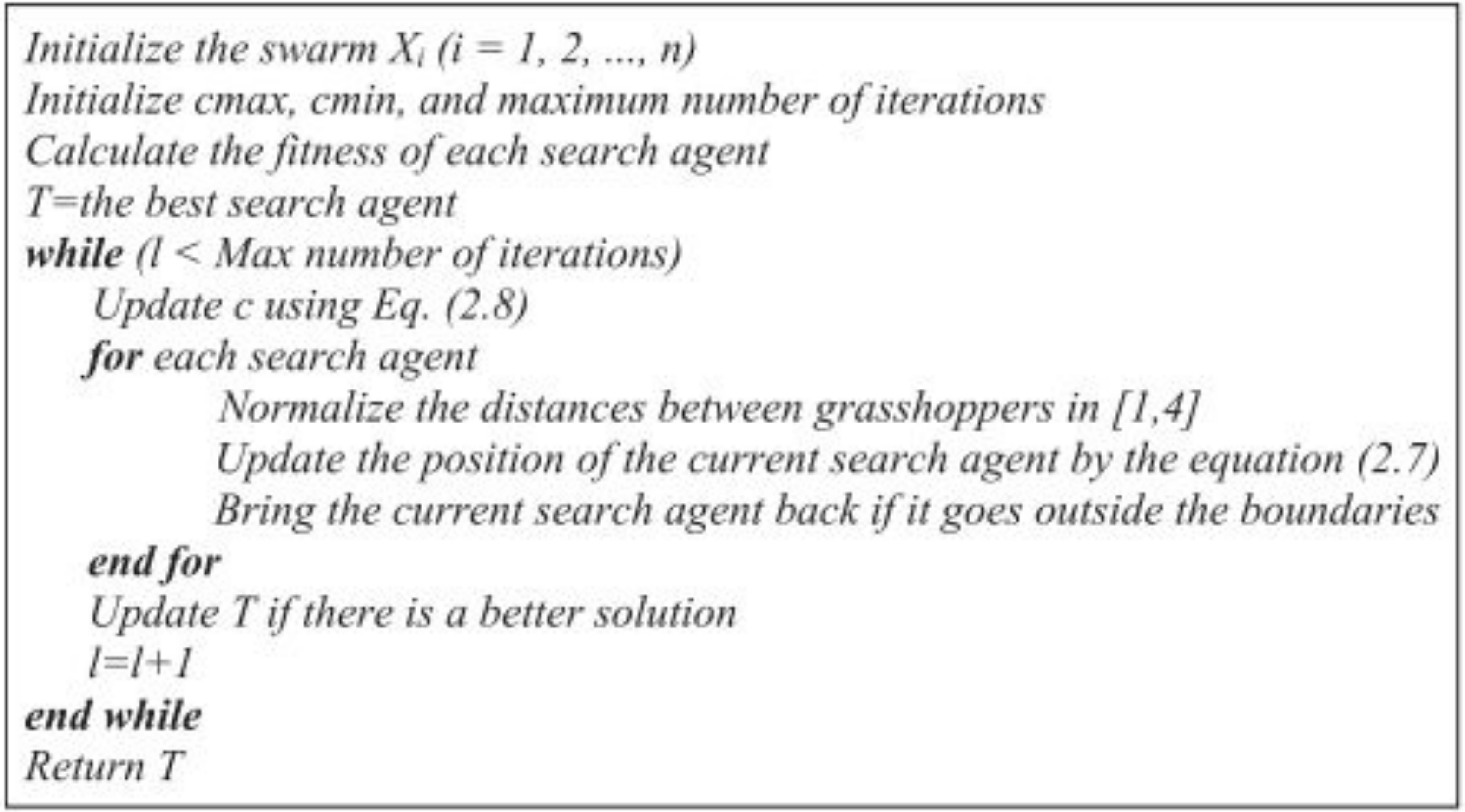


Fig. 7. (a) Behaviour of grasshoppers around a stationary and mobile target in 2D space and (b) 3D space (c) Behaviour of grasshoppers on a unimodal test function and a multi-modal test function.

Algo.



```
Initialize the swarm $X_i$ ($i = 1, 2, ..., n$)
Initialize cmax, cmin, and maximum number of iterations
Calculate the fitness of each search agent
T=the best search agent
while ($l <$ Max number of iterations)
    Update c using Eq. (2.8)
    for each search agent
            Normalize the distances between grasshoppers in [1,4]
            Update the position of the current search agent by the equation (2.7)
            Bring the current search agent back if it goes outside the boundaries
    end for
    Update T if there is a better solution
    $l=l+1$
end while
Return T
```

Fig. 8. Pseudo codes of the GOA algorithm.

RESULTS

1. Experimental setup

-The performance of different algorithms can be measured quantitatively. the characteristics of the test functions should be diverse to be able to draw a mature conclusion. The test functions are

--**Unimodal**

--Multimodal

--Composite.

-For solving the test functions, 30 search agents and 500 iterations were employed. Each of the test functions was solved 30 times to generate the statistical results. Different performance indicators were utilised to quantitatively compare the algorithms: average and standard deviation of the best solutions obtained in the last iterations.

2. Qualitative results and discussion

The first experiment was performed on the 2D version of some of the test functions using only 5 artificial grasshoppers. The main objective for this experiment was to observe the behaviour of the GOA qualitatively.

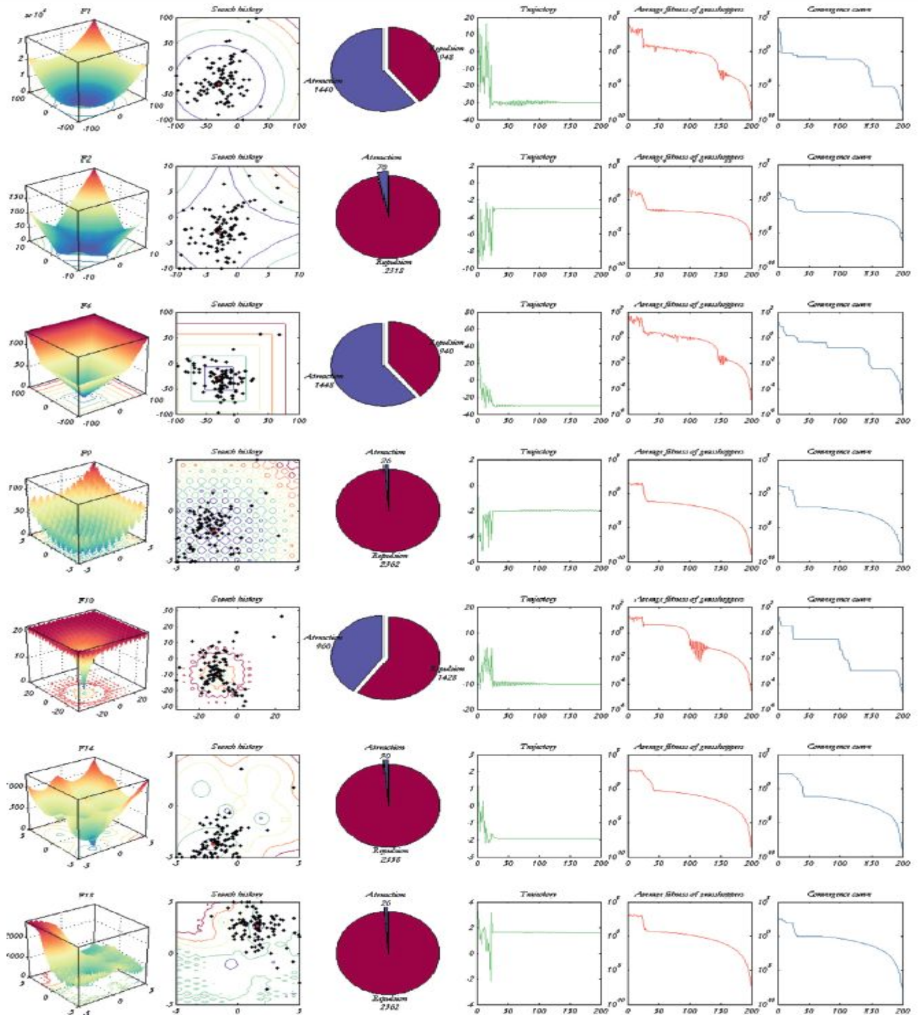
•**Search history**: Shows the location history of the artificial grasshoppers during optimisation.

•**Attraction/repulsion rates**: Shows the number of times that all artificial grasshoppers attracted or repelled each other during optimisation.

•**Trajectory** of the first grasshopper in the first dimension: Shows the value of the first variable of the first grasshopper in each iteration.

•**Average fitness**: Indicates the average objective value of all grasshoppers in each iteration.

•**Convergence curve**: Shows the objective value of the best solutions obtained so far (target) in each iteration.


F1
F2
F6
F9
F10
F14
F15
Search history
Trajectory
Average fitness of grasshoppers
Convergence curve
Attraction 1440
Repulsion 948
Attraction 79
Repulsion 2318
Attraction 1448
Repulsion 940
Attraction 26
Repulsion 2362
Attraction 960
Repulsion 1428
Attraction 50
Repulsion 2338
Attraction 26
Repulsion 2362

The above discussed results qualitatively demonstrated that the GOA is able to solve optimisation problems. However, the test functions were of 2 variables and qualitative results cannot tell us how much better this algorithm is compared to current ones

The GOA algorithm shows the best results when solving Unimodal test functions. The results of this algorithm are substantially better in more than half of the **Unimodal** test functions, showing the high performance of this algorithm. Unimodal test functions have only **one global optimum**,GOA algorithm benefits from **high exploitation** ability.

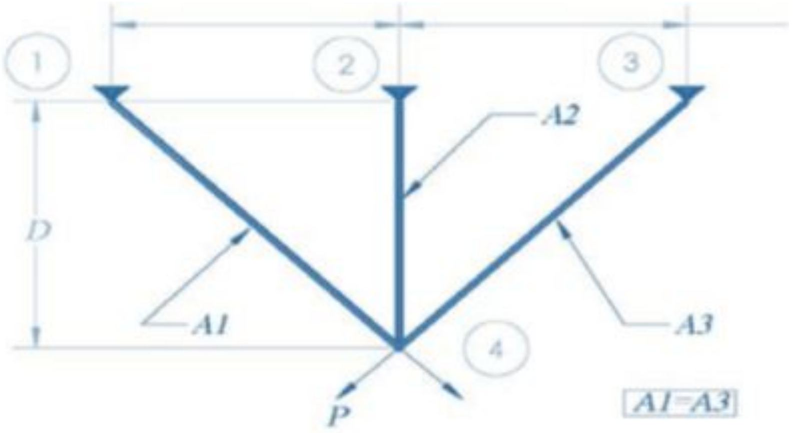
The results are consistent in which the GOA algorithm tends to significantly outperform others in both of the performance metrics. The results of this algorithm are again remarkably superior in the majority of **Multimodal** test functions. Since the multi-modal test functions have a significant number of local solutions, these results quantitatively show the effectiveness of the proposed algorithm in avoiding local solutions during optimisation.

The results of the algorithms on composite test functions. These results show that the GOA algorithm provides very competitive results compared to other algorithms. **Composite** test functions are even more challenging than the multi-modal ones and require a **proper balance between exploration and exploitation**. Therefore, it can be stated that the GOA is able to balance exploration and exploitation properly for solving such challenging problems.

REAL APPLICATIONS

1.*Three-Bar Truss Design Problem*

Consider _ x = [*x* 1 *x* 2] = [*A* 1 *A* 2] ,
Minimise f (_ x) = _2 √ 2 x 1 + x 2 _∗l ,
Subject to g 1 (_ x) = √ 2 *x* 1 + *x* 2 √ 2 *x* 2 1 + 2 *x* 1 *x* 2 P −σ≤0 ,
g 2 (_ x) = *x* 2 √ 2 *x* 2 1 + 2 *x* 1 *x* 2 P −σ≤0 ,
g 3 (_ x) = 1 √ 2 *x* 2 + *x* 1 P −σ≤0 ,
Variable range 0 ≤*x* 1 , *x* 2 ≤1 , where l = 100 cm ,
P = 2 KN / c m 2 , σ= 2 KN / c m 2

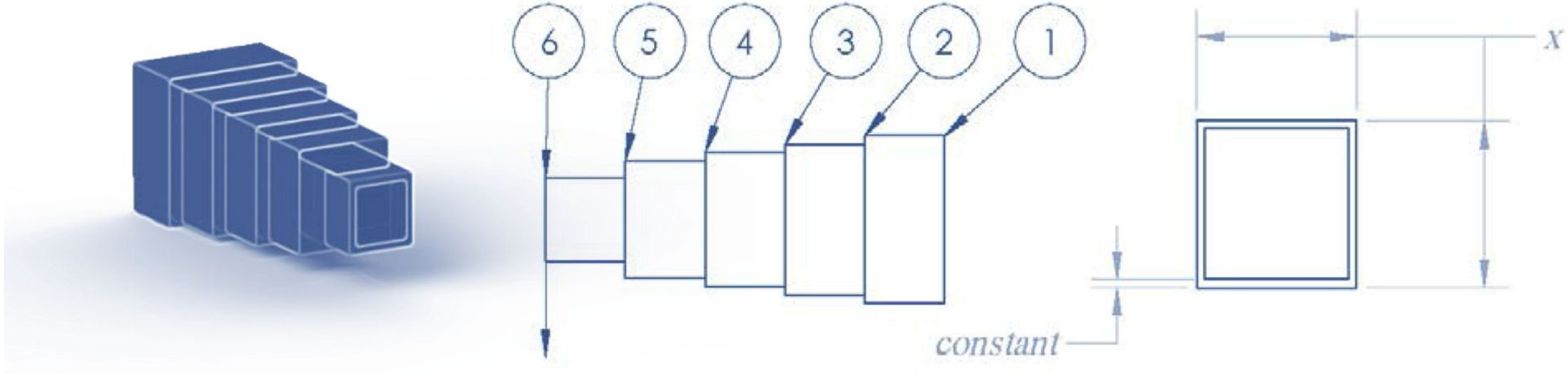


The objective is to **minimise the weight of the truss**. This problem is subject to several constraints as well: stress, deflection, and buckling constraints. The proposed GOA with **20 search agents and 650 iterations** was employed on this problem. Since this problem is a constrained problem, a constraint handling method needed to be integrated with GOA.this algorithm outperforms the rest of the algorithms significantly. These results show that the GOA algorithm is able to handle the difficulties of a constrained search space efficiently.

2. ***Cantilever beam design problem***

Consider _ x = [x 1 x 2 x 3 x 4 x 5]
Minimise f (x) = 0 . 6224 (*x* 1 + *x* 2 + *x* 3 + *x* 4 + *x* 5) ,
Subject to g (x) = 61 x 3 1 + 27 x 3 2 + 19 x 3 3 + 7 x 3 4 + 1 x 3 5 −1 ≤0 ,
Variable range 0 . 01 ≤x 1 , x 2 , x 3 , x 4 , x 5 ≤100 ,

that the cantilever beam is built using five, hollow, square-section, box girders, and the lengths of those girders are the design parameters for this problem. There is also one constraint for this problem. The GOA algorithm with **20 search agents** and a **maximum of 650 iterations** is employed to determine the optimum for this p, This algorithm provides the lowest number maximum function evaluation problem.



3. 52-bar truss design

In this problem, the objective is to **minimise the weight of a truss with 52 bars and 20 nodes**.Four of the nodes are fixed and the bars are classified in 12 groups as follows, which are the main parameters to be optimised for this problem:

•*Group* 1: *A* 1 , *A* 2 , *A* 3 , *A* 4

•*Group* 2: *A* 5 , *A* 6 , *A* 7 , *A* 8 , *A* 9 , *A* 10

•*Group* 3: *A* 11 , *A* 12 , *A* 13

•*Group* 4: *A* 14 , *A* 15 , *A* 16 , *A* 17

•*Group* 5: *A* 18 , *A* 19 , *A* 20 , *A* 21 , *A* 22 , *A* 23

•*Group* 6: *A* 24 , *A* 25 , *A* 26

•*Group* 7: *A* 27 , *A* 28 , *A* 29 , *A* 3

•*Group* 8: *A* 31 , *A* 32 , *A* 33 , *A* 34 , *A* 35 , *A* 36

•*Group* 9: *A* 37 , *A* 38 , *A* 39

•*Group* 10: *A* 40 , *A* 41 , *A* 42 , *A* 43

•*Group* 11: *A* 44 , *A* 45 , *A* 46 , *A* 47 , *A* 48 , *A* 49

•*Group* 12: *A* 50 , *A* 51 , *A* 52

The following list presents other parameters involved in this problem:

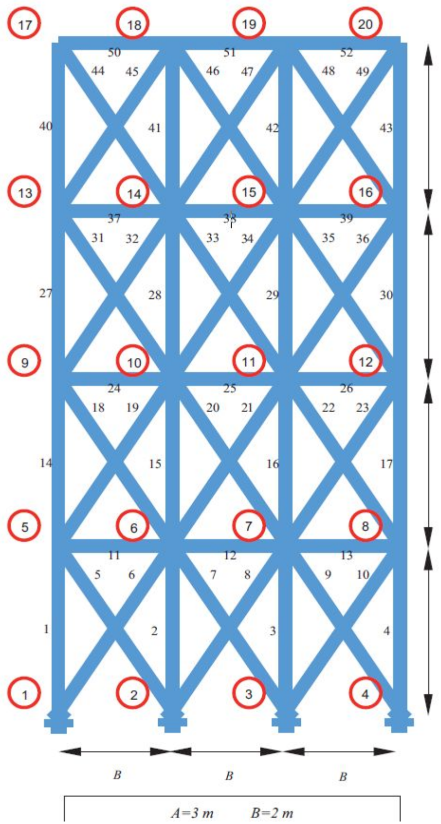
•ρ=7860.0 *kg* / *m* 3

•*E* = 2.07 *e* 5 *MPa*

•*Stress limitation* = 180 *MPa*

•*Maximum stress* = 179.7652 *MPa*

•*Design variable set*•*P k* = 100 *kN* ,*P y* = 200 *kN*



This is a discrete problem,To make GOA discrete, we simply round the search agents to the nearest integer.
-This problem is solved using **30 search agents and 500 iterations**,it is evident that the GOA finds the best optimal value for this problem with the least number of function evaluations. This highlights the performance of GOA in solving real problems with more variables. These results clearly demonstrate the merits of the GOA algorithm in solving real problems with unknown search spaces.
-The **exploration ability** of GOA is high in the initial steps of optimisation, which is due to the large repulsion rate between grasshoppers. This assists GOA to explore the search space broadly and discover its promising regions.
-Then exploitation is high in the last steps of optimisation, which is due to the larger attraction forces between the grasshoppers.

GOA considers a given optimisation problem as a **black box**, so it does not need gradient information of the search space. Therefore, this algorithm can be applied to any optimisation problem in different fields subject to proper problem formulation.

CONCLUSION

•Exploitation of the GOA is satisfactory on problems involving unimodal test functions.
•Exploration of the GOA is intrinsically high for multi-modal test functions.
•GOA properly balances exploration and exploitation when solving challenging problems involving composite test functions.
•GOA has the potential to significantly outperform several current algorithms when solving a range of current or new optimisation problems.

From the real life applications we were able to conclude:

•GOA is able to improve the initial random population for a real problem.
•The target is improved over the course of iterations, so the approximation of the global optimum become more accurate proportional to the number of iterations.
•GOA is able to solve real problems with unknown search spaces.

GOA is only able to solve single-objective problems with contentious variables.