CHAPTER 4

BIRDS FLOCKING

4.1 INTRODUCTION

Birds flocking algorithm, which is a Particle Swarm Optimization (PSO) technique, is used in the second module of the integrated system to optimize the test cases. The optimization of test cases is done by clustering the test cases using objective function.

4.2 PARTICLE SWARM OPTIMIZATION (PSO)

Particle Swarm Optimization (PSO) was invented by Russell Eberhart and James Kennedy in 1995 as they were inspired by the flocking patterns of birds and schooling patterns of fish. PSO is a robust stochastic optimization technique based on the movement and intelligence of swarms. For problem solving, it uses the concept of social interaction. PSO optimizes a problem by iteratively trying to improve the solution with respect to a given objective.

PSO is based on the concept of swarms which move around the search space to select the best solution for the problem. Here the swarm is constituted by a number of agents (particles) and is treated as a point in an N-dimensional search space. While the swarm is moving towards the best solution for the problem it adjusts its movement based on its own knowledge and the knowledge of the other particles in the N-dimensional search space. Usually swarms retain the knowledge in memory.
In case of the search space, the swarms are allowed to share the same position with other particles with the restriction that they have to sustain their individuality meaning the tendency to return to its previous position. Apart from individuality the movement of each particle is the composition of an initial random velocity and the second factor sociality means the tendency to move towards the neighborhood's best previous position.

As an optimization method PSO finds the global optimum of a function (fitness function) defined in a given space (search space). Each swarm may modify their state based on three factors.

- The knowledge of the environment (its fitness value)
- The swarm’s previous history of states (its memory)
- The previous history of states of the swarm's neighborhood (its memory)

PSO has been used in many applications. The PSO techniques are very helpful in showing the relationships between computational techniques and biological phenomena. PSO techniques are used to show how computational problem solving techniques help in studying biological phenomena and how biological techniques help out with computational problem solving techniques.

Apart from PSO, there are already many computational techniques inspired by biological systems are available like artificial neural network simulating the human brain, genetic algorithm are inspired by Darwin's theory about evolution. PSO has also been applied to multi-objective problems, in which the objective function comparison takes pareto dominance into account when moving the PSO particles and non-dominated solutions are stored so as to approximate the pareto front.
4.3 **PSO ALGORITHMS**

The algorithm of PSO emulates from the behavior of animals societies like ants, bees, birds and fish. PSO algorithms are mainly used in optimization. Many algorithms like ant colony, bee colony, fish schooling and birds flocking are available. The below specified types of algorithms are taken from Wikipedia.

### 4.3.1 Altruism Algorithm

Altruism means unselfish. The algorithm is developed based on the unselfish behaviour of the swarm and can evolve over time and result in more effective swarm behaviour.

### 4.3.2 Ant Colony Optimization

Ant colony optimization (ACO) is a population-based metaheuristic optimization algorithm modeled on the actions of an ant colony. Optimization problem is solved by converting the optimization problem into the problem of finding the best path. The ants record their positions and the quality of their solutions by moving on the graph and they incrementally build solutions so that in later iterations more ants locate better solutions.

### 4.3.3 Artificial Bee Colony Algorithm

Artificial bee colony algorithm (ABC) is a meta-heuristic algorithm (Karaboga 2007) and simulates the foraging behaviour of honey bees. The ABC algorithm has three groups of bees. They are employed bee, onlooker bee and scout bee.
4.3.4 Artificial Immune Systems

Artificial immune systems (AIS) is based on biologically inspired computing and natural computation.

4.3.5 Bat Algorithm

Bat algorithm (BA) inspired by the echolocation behavior of microbats is a metaheuristic optimization algorithm (Xin-She Yang 2010) based on the echolocation behaviour of microbats with varying pulse rates of emission and loudness.

4.3.6 Charged System Search

Charged System Search (CSS) is a new multi-agent optimization approach based on the governing laws of Coulomb and Gauss from electrostatics and the Newtonian laws of mechanics. In CSS each agent is a Charged Particle (CP) and is applicable to all optimization fields therefore the CSS is considered as a good global and local optimizer simultaneously.

4.3.7 Cuckoo Search

Cuckoo search (CS) mimics the brooding behaviour of cuckoo species is an optimization algorithm developed by Xin-she Yang and Suash Deb in 2009.

4.3.8 Differential Search Algorithm

Differential search algorithm (DSA) inspired by migration of superorganisms is developed for solution of real-valued numerical optimization problems.
4.3.9 Firefly Algorithm

Firefly algorithm (FA) is especially suitable for a multimodal optimization problem mimics the flashing behaviour of fireflies. Light intensity attraction enables the fireflies with the ability to subdivide into small groups and each subgroup swarm around the local modes.

4.3.10 Glowworm Swarm Optimization

Glowworm swarm optimization (GSO), is a new method of swarm intelligence based algorithm (Krishnanand and Ghose 2005) for simultaneous computation of multiple optima of multimodal functions. The main objective of using this method is to ensure capture of all local maxima of the function.

4.3.11 Gravitational Search Algorithm

Gravitational search algorithm (GSA) is based on the theory of Newtonian physics which is a concept of law of gravity and its searcher agents are the collection of masses. In GSA, the gravitational force is a way of transferring information between different agents (Rashedi et al 2009). The heavy masses correspond to good solutions of the problem and the position of the agent corresponds to a solution of the problem.

4.3.12 Intelligent Water Drops

Intelligent water drops algorithm (IWD) is based on the optimal destination path finding behaviour of natural rivers. As the IWD algorithm is generally a constructive population-based optimization algorithm, the solutions are incrementally constructed by the IWD algorithm where several artificial water drops cooperate to change their environment in such a way that the optimal path is revealed as the one with the lowest soil on its links.
4.3.13 **Krill Herd Algorithm**

Krill herd (KH) is a novel biologically inspired algorithm based on simulating the herding behavior of krill individuals is proposed by Gandomi and Alavi in 2012 for optimization tasks. The minimum distances of each individual krill from food and from highest density of the herd are considered as the objective function for the krill movement.

4.3.14 **Magnetic Optimization Algorithm**

Magnetic Optimization Algorithm (MOA), inspired by the interaction among some magnetic particles with different masses proposed by Tayarani in 2008 is an optimization algorithm. In this algorithm, the possible solutions are some particles with different masses and different magnetic fields. The best solution is identified based on the fitness of the particles.

4.3.15 **Fish Schooling**

It is based on the concept of aggregation of fish. The aggregation is formed usually by the same species and the same age or size.

4.3.16 **River Formation Dynamics**

River formation dynamics (RFD) is a heuristic method based on fact how water forms rivers by eroding the ground and depositing sediments. As water transforms the environment, altitudes of places are dynamically modified, and decreasing gradients are constructed. The gradient orientation of RFD makes it specially suitable for solving these problems and provides a good tradeoff between finding good results and not spending much computational time.
4.3.17 **Self-propelled Particles**

Self-propelled particles (SPP), also referred to as the Vicsek model, were introduced in 1995 by Vicsek as a special case of the boids model introduced in 1986 by Reynolds. A swarm is modeled in SPP by a collection of particles that move with a constant speed and that swarming animals share certain properties at the group level, regardless of the type of animals in the swarm.

4.3.18 **Stochastic Diffusion Search**

Stochastic diffusion search (SDS) is an agent-based probabilistic global search and optimization technique best suited to problems where the objective function can be decomposed into multiple independent partial-functions.

Even though there are so many algorithms available in PSO, this research work is focused on birds flocking algorithm in PSO as it has not been used in testing. Not only that, it has many advantages too when compared with other PSO algorithms. The advantages of flocking are great enough that many different types of birds assemble in small, medium and large groups for different reasons.

The advantages of birds flocking are

- **Foraging:** It is the act of searching for food sources. Birds often form flocks while searching for and exploiting food sources. While feeding, if one from the flock locates the food source, then the other birds are also try to locate food sources which are identified by other flocks. This behavior is much useful in case of clustering the test cases.
- **Protection:** defense or safety is also an important characteristic of birds. They usually form flocks with type of birds with similar characteristics. So, the potential threats are easily identified by the birds. That too the larger group of birds has a better chance of spotting a potential threat than a single bird has.

- **Raising Families:** In case of birds the family care is seen. The parent birds are taking care of their young ones. The parent birds protect their young ones from all the dangers and also they feed their young ones till the young ones are becoming capable enough to get their feed.

- **Aerodynamics:** usually the birds form some shapes when they fly. This formation of shapes helps them in changing the wind patterns so that they use the air in the energy efficient way.

These advantages shown by birds can be very well used for test case optimization. In case test case optimization, there is a need for the test case to find the objective function and also the other test cases with similar objectives and then there is a need to move towards it and then the test cases are clustered based on their objectives so as to provide the strength to the testing. So birds flocking algorithm is used for test case optimization.

### 4.4 TEST CASE OPTIMIZATION USING BIRDS FLOCKING

Bio-inspired computational collective behavior flocking model was first proposed by Craig Reynolds in 1987. Since birds show an exploring behavior, Birds flocking algorithm is preferred for test case optimization. The flock always follows an exploring behavior. Each bird tries to locate the goal by searching the environment. If the bird locates the goal, then all the flock members are moved towards the goal.
4.4.1 Birds Flocking

Birds flocking algorithm is simple and easy to implement and it is much useful for making simple and yet complex decisions. Usually birds are seen in flocks meaning a number of birds feeding, resting, or traveling together. The birds with similar characteristics form a group.

The benefits of aggregating in flocks are varied and flocks will form explicitly for specific purposes. Flocking also has costs, particularly to socially subordinate birds, which are bullied by more dominant birds; birds may also sacrifice feeding efficiency in a flock in order to gain other benefits. The principal benefits are safety in numbers and increased foraging efficiency. Defense against predators is particularly important in closed habitats such as forests where predation is often by ambush and early warning provided by multiple eyes is important, this has led to the development of many mixed-species feeding flocks. These multi-species flocks are usually composed of small numbers of many species, increasing the benefits of numbers but also increasing potential competition for resources.

Group size is a major aspect of the social environment of gregarious animals. However, one has to be careful when using group size measures to characterize animal sociality, because average individuals live in groups larger than mean group size.

4.4.2 Birds Flocking Algorithm

The flocking algorithm imitates the flocking activities of birds on a computer. In birds flocking algorithm, the motion of flocks of birds are characterized as individual behaviors. Each boid (agent) utilize an accurate geometric model for flight. Particularly, in flocking algorithm, there is no
leader meaning no global control. The flocks normally steer toward the
general heading of the rest of the flocks.

A flock of birds is circling over an area where they can smell a
hidden source of food. If one bird finds the food source and if it is closest to
the food, then it communicates the availability of the food source by giving a
signal to other birds and the other birds swing around in its direction. If any
of the other circling birds comes closer to the target than the first, then it gives
signal. The other birds turn over towards him. This reduction pattern
continues until one of the birds happens upon the food. Over a number of
iterations, a group of birds have their values adjusted closer to the member
whose value is closest to the target at any given moment. This process is
repeated until the best conditions or a food source discovered. The process of
test case optimization follows the work of this birds behaviour.

The algorithm keeps track of three global variables

- Optimization condition (Target value)

- Global best (gBest) value indicating which particle's data is
currently closest to the Target

- Stopping value indicating when the algorithm should stop if the
Target isn't found

The different patterns of flock of birds are shown in Figure 4.1
(taken from Google images). Flocks are formed usually when they are
searching for food or for meeting some goal.
Many bird species are sociable, meaning living in flocks or loosely organized communities. Usually birds form flocks for different reasons. There are also many variations in flocking like flocks may be of different sizes, they may occur in different seasons. And usually the flock may be composed of species that can work well together in a group. Flocks are so common in some bird species. A raft of ducks, a charm of finches, and a horde of ravens are some of the special names given to these groups of birds. The birds flock normally in groups. And the group of birds will have similar behavior. It responds only to the flock mates within a certain small radius. he
birds not only flock in different patterns but also they flock in different groups according to their characteristics. The birds with similar characteristics flock together.

Figure 4.2 shows the different flocks of birds based on their size, characteristics, behaviour etc., and it clearly depicts that the flocks of birds differ based on their size, colour, behavior, place etc.,
4.4.3 Test Case Optimization

The birds flocking algorithm is used for test cases optimization. The movement behavior of the birds is utilized as a base idea in optimizing the test cases. In birds flocking algorithm, the motion of flocks of birds are characterized as individual objectives. The algorithm is based on the concepts of flock’s behaviour. The agents search the goals in parallel, and indicate the presence or dearth of significant patterns into the data to other flock members by giving signal.

In the same way, in test case optimization, the test cases are considered as birds. The algorithm is based on objectives of the test cases. Each test case is checked for the presence or the dearth of the objective. If the objective is found, then the test case is selected for the checking of the presence of the next objective and the process is continued till the presence of all the objectives are being checked.

The output from mRMR feature selection is used as input to birds flocking algorithm. Figure 4.3 gives the basic structure of birds flocking algorithm.

The birds flocking algorithm has three stages. They are

- Input
- Applying birds flocking algorithm
- Output clusters
4.4.3.1 Input

The test cases and the objectives are given as the input to birds flocking. The objectives considered for test case optimization are maximization of fault coverage, maximization of coverage, minimization of execution time and minimization of memory usage. Based on the threshold value that is applied on the objectives the test cases are clustered. The remaining test cases are outliers and they are omitted.

Each test case consists of

- Data value for the objectives
- A Velocity value indicating how much the Data can be changed to achieve the target value.
A personal best (pBest) value indicating the closest the test case's Data has ever come to the Target

In the birds flocking, the data would be the W, X, Y, Z coordinates of each bird. Here, four coordinates are used because each test case has four objectives. The individual coordinates of each test case is matched to the coordinates of the test case which is closer to the objective function (gBest). The process is continued till the test case matches the target objective function. In test case optimization, the data value represents input value of the objectives of the test case.

Figure 4.4 gives shows the input of birds flocking algorithm. The test cases from feature selection are given as the input to the birds flocking. Each test case has the data value for the objectives such as the memory required for execution of the test case, the fault covered by that test case, completeness of the test case, the execution time of the test case and the test cases Id along with the calculated information.

Similar to the input given to the input stage of feature selection, each test case and the related information stored inside the notepad are separated by “,”. The function readTestCases() is used for reading the test cases from the notepad. While reading the test cases from the notepad, the symbol “,” indicates the fore coming test case is a new test case. But the difference here is that the input here is the output from feature selection and not the original test suite.
4.4.3.2 Applying birds flocking algorithm

Test case optimization is done by using the three simple basic models of flocking behavior. They are

- Separation - avoid crowding neighbors (short range repulsion)
- Cohesion - steer towards average position of neighbors (long range attraction)
- Alignment - steer towards average heading of neighbors

Test cases are separated from one another in separation. The test cases are pushed separately to keep them away from crashing into each other by maintaining the distance from nearby clusters. The cohesion gives the ability to coalesce with other nearby test cases to form a cluster. The
alignment gives test case the potential to align with other nearby clusters. With these three simple rules, the flock moves in an extremely realistic way, creating complex motion and interaction that would be extremely hard to create otherwise.

4.4.3.2.1 Separation

Separation is mainly collision avoidance with nearby test cases. Separation helps in separating the test cases thus crowding and clashing of test cases is avoided. This is done by maintaining a constant distance between the test case with the neighboring test cases.

With the help of this behavior the test cases are spread out thus maximizing their coverage region. When the coverage region increases they are allowed to scan a wide area.

Figure 4.5 Separation
The algorithm for separation function is given as

- For every test case check whether it is too close to its neighbor
  - For every $t_i$, $i$ varies from 1 to $n$
  - Calculate the distance between $i$ & $i++$
    - If $\text{dist} = 0$
      - Then same test case
    - Else $\text{dist} < \text{specified}$
      - Move test case to a specified distance

In separation function the distance between the test cases are calculated. If the distance is less than the specified distance then they are separated by a specified distance. Here the distance means the value of the objective of the test case.

### 4.4.3.2.2 Cohesion

Cohesion helps in grouping the test cases as shown in Figure 4.6. The test cases are grouped based on their objectives. The test cases with the similar objectives are grouped together. For this, initially the center value is calculated. This helps the test cases to move to the center of mass of its neighbouring test cases. Centering attempts the test cases to stay close to nearby test cases.

In cohesion, test case is moved to the nearby center. So that, the test case starts to move towards each other and also towards the center. Thus the cohesion is calculated by finding the neighbouring test cases of a particular distance and averaging their position velocity. This average gives
the center of mass of neighbours. Once it is calculated all the test cases starts to move to that center of mass thus resulting in formation of group.

![Figure 4.6 Cohesion](image)

The function findclosestdistance() is used where the test cases are given as input to find the distance between them. In this function each test case is read separately and findeuclidean() is used to find the distance. The distance is calculated using the formula

$$ dist = \sqrt{distance + (t1-t2)^2} $$

where t1 is first test case, t2 is second test case and t1!=t2. The calculated Euclidean distance is checked whether it is greater than minimum threshold and lesser than maximum threshold and if it is within the threshold value the test cases along with the distances is stored in the distance vector and birdDist vector.

Now based on the distance all the test cases in birdDist vector is sorted in ascending order. Now the sorted test cases are stored in a new vector called allBirds. The selectrandompoints() function is used for generating random points. The random points are stored in the rpoints vector.

The next step is the grouping of test cases where the test cases are grouped based on their characteristics features. Here each test case and
distance in distance vector is read using the elements in the rpoints vector as index, the difference between distance of each test cases is calculated. A threshold value is set. The threshold value is calculated as the average of all distances. If difference between the distance is less than the threshold, the test case along with the distance is stored in the vector called result. The next step is clustering which is done by alignment.

4.4.3.2.3 Alignment

Alignment as shown in Figure 4.7 involves velocity matching attempt to match velocity with nearby flock mates. Here, the test case objectives are matched to form clusters. In velocity matching, the central bird is located and the remaining are moved towards the central.

The velocity value is calculated as the distance between individual's data to the target. Then the velocity matching is done. In this the velocity are computed by averaging the velocity of the neighbouring flock mates.

This will make the group of boids to travel in the same direction at the same speed. Steering for alignment can be calculated by finding out all the nearby neighborhood agents and averaging their heading vectors. In general, the alignment gives an agent the potential to align with other nearby characters. In a flock of birds that are all the same color, each bird will try to match the direction of the birds around it that it can detect.

After the group of birds are joined together, the next step is to find the closest distance between the birds in order to form the cluster of birds in order to aim a target. Similarly after the test cases are generated and their features are calculated they need to be clustered in order to find the optimized test cases with the matching behavior.
Each testcase's pBest value indicates the closest the objective value has ever come to the objective function.

The gBest value only changes when any testcase's pBest value comes closer to the objective function than gBest. Through each iteration of the algorithm, gBest gradually moves closer and closer to the objective.

The next step is clustering of test cases where startclustering() is used for clustering of test cases. Here each test case and distance in distance vector is read using the elements in the rpoints vector as index, maximum distance is set by adding a threshold to the distance of test case and minimum distance is set by subtracting the threshold to the distance of test case. The remaining test cases from the distance vector are checked whether it is already in the clustered list using isclustered(). The isclustered() function reads the test case in the rescluster vector and checks with the test case distance present in the distance vector and returns true if the test case is in the rescluster vector and false otherwise. If the test case is not clustered and the distance between the test case is within the limit store the test case in the rescluster vector. Similarly the test cases present in the distance vector are clustered.
4.4.3.3 Output

The printcluster() function is used for displaying the resultant clustered test cases as shown in Figure 4.8 in the rescluster vector.

![Figure 4.8 Output of birds flocking](image)

The optimized test cases from birds flocking is given as the input for genetic algorithm for prioritization.

4.5 SUMMARY

The test suite is optimized using birds flocking scheme presented above based on the objective function. But still the effectiveness of the test suite may be increased by prioritizing the test suite. So the prioritizing of the test suite is done in the next chapter.