# EVALUATION OF A NEW BIO-INSPIRED ALGORITHM: KRILL HERD

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Devi Madamanchi

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

November 2014

Fargo, North Dakota

# North Dakota State University

Graduate School

**Title**

EVALUATION OF A NEW BIO-INSPIRED ALGORITHM: KRILL HERD

**By**

Devi Madamanchi

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

**MASTER OF SCIENCE**

SUPERVISORY COMMITTEE:

Dr. Simone Ludwig
Advisor (typed)

Dr. Wei Jin

Dr. María de los Ángeles Alfonseca-Cubero

Approved by Department Chair:

| | |
|---|---|
| 11/07/2014 | Dr. Brian M. Slator |
| Date | Signature |

# ABSTRACT

Number of nature inspired algorithms is proposed to solve complex optimization problems. The Krill Herd algorithm is one such biologically-inspired algorithm, proposed to solve optimization problems in response to biological and environmental processes. It is mainly based on the simulation technique of the herding behavior of the krill swarms. The objective function is defined as the combination of minimum distance of the krill individual from food and from the highest density of the swarm. The position of the krill individual is mainly influenced by three important factors, (i) movement induced by other krill individuals, (ii) foraging activity, and (iii) random diffusion. The process is mimicked to find optimum solution of the algorithm. In this paper, I implemented and evaluated the algorithm using five different benchmark functions, namely Alpine, Ackley, Griewank, Rastrigin and Sphere. The results obtained are satisfactory and proves the Krill herd algorithm’s efficiency in solving the optimization problems.

# ACKNOWLEDGEMENTS

I would like to take the opportunity to thank my advisor Dr. Simone Ludwig for the immense support, motivation, valuable ideas and continuous mentoring throughout my research.

I would also like to thank Ibrahim Aljarah, for helping me with my research and for the valuable suggestions and work done.

I would also like to thank my committee members for being a part of my research and for their support throughout.

Finally I would like to thank my parents and my brother Manoj Babu Madamanchi, friends and others who have been a great support throughout my career and always had belief in me that I could do it.

# DEDICATION

I would like to dedicate my work to my parents and my brother Manoj Babu Madamanchi for their continuous support throughout my life and career and helping me achieve all the success in my life.

# TABLE OF CONTENTS

# LIST OF TABLES

Table Page

# LIST OF FIGURES

# 1. INTRODUCTION

Optimization can be defined as the act of achieving the best possible solution to a problem under given circumstance. The interest in the field of optimization has been enormously increased since the late fifties and various theories have been proposed to provide an optimum solution. The optimization problems in the real world are very challenging and hard to solve. Solving such hard problems requires using optimization tools. Also, the optimization tools may or may not provide the optimum solution depending on the degree of difficulty of the problem. As a result, various optimization techniques came into practice. Additionally, many new algorithms are proposed to solve these kinds of challenging optimization problems.

In the process of optimization, there are many decisions that are to be taken according to the constraints. The goal of these decisions is to either minimize or maximize the solution to constrained optimization problems. Here, the solution can be referred to as a function of certain design variables. Hence, we can say optimization is the process of finding the conditions that give the maximum or the minimum value of a function. The word optimum can mean either ‘maximum’ or ‘minimum’ based on the circumstances.

Optimization problems are always encountered in different fields such as in Engineering, Mathematics, and Sciences, etc. And there is no one single method that is available to solve all kinds of optimization problems. Hence, various numbers of theories have come into existence and are still undergoing continuous development to provide efficient solutions to the complex optimization problems based on some criterion.

Metaheuristic algorithms have become a prominent source in providing solutions to the optimization problems, and are proven to be efficient when compared to the other conventional methods followed based on the Linear and Non-Linear programming [1]. It has mainly two phases, firstly it searches for the current best solution and the best individuals, and secondly, the search space is explored thoroughly keeping in mind all the constraints [2]. This procedure ensures the algorithm to be faster and also makes it capable of solving larger search spaces [3].

Always, nature has been one of the inspirations to many of the research areas. As such, the nature inspired algorithms came into existence. All the existing algorithms are divided into four categories depending upon the source of inspiration namely, Swarm intelligence based, Bio-inspired based, Physics/Chemistry based, and others [4]. This division is emphasized based upon the focus and perspective of the search criteria. For example: population-based and trajectory-based in case of Particle Swarm Optimization [5] and Simulated Annealing [6]. If the focus is upon the interaction between the individuals, they can be categorized as attractive based and non-attractive-based and so on. Here, the classifications are done mainly considering the actual perspective and the motivations behind it along with the sources of inspiration.

Among all, Particle Swarm Optimization, Firefly algorithm [7] etc., have gained importance due to their efficiency. Currently, about 40 different algorithms are used in practice. Classification of these algorithms systematically is a tedious task.

### 1.1. Swarm Intelligence Algorithms

The Swarm Intelligence (SI) algorithms are based upon the simulation of the collective, emerging behavior of individuals following some simple rules. Each individual in the solution space is considered to be not very intelligent, the whole system of multiple individuals exhibit self-organization behavior, thus showing some collective intelligence.

The SI based algorithms are inspired from the collective behavior of the insects, like ants, bees, wasps, etc., and also from different animal categories like flock of birds, or fish, etc. Few popular algorithms of Swarm Intelligence are: (i) Particle Swarm Optimization (PSO), proposed by Eberhart and Kennedy, where the algorithm uses the swarming behavior of fish or birds etc. (ii) Firefly Algorithm (FA), which uses the flashing behavior of swarming fireflies. (iii) Ant Colony Optimization [8], which uses the interaction of the social insects like ants. Similarly, cuckoo search [2] uses the brooding parasitism of cuckoos; bat algorithm [9] uses the echolocation of foraging bats; bee algorithm [10] uses the foraging behavior of the honey bees.

The Swarm Intelligence algorithms are popular and widely used in practice, mainly the reason being the sharing of information between multiple individuals making self-organization over the iterations highly efficient. Also, Swarm intelligence can be parallelized so that large-scale optimization implementations appear more practical in nature.

### 1.2. Bio-Inspired Optimization Algorithms

Among all the Nature Inspired Algorithms, the Bio Inspired Algorithms is one of the main categories of Nature Inspired Meta-heuristic Algorithms. These algorithms also can be said as a super class to the Swarm Intelligence based algorithms.

The Swarm Intelligence algorithms follow the swarming behavior of the individuals whereas the Bio Inspired algorithms indirectly follow the swarming behavior of the individuals. For example, Genetic Algorithms [11] fall under Bio-inspired algorithms rather than Swarm Intelligence based. Similarly, the Flower Algorithm [12], where it is a culmination of the pollination characteristics of flowering plants and associated flower consistency of pollinating insects.

The Bio-inspired Algorithms in turn can be categorized into

(i) Evolutionary Algorithms,

(ii) Bacterial Foraging Algorithms

### 1.2.1. Evolutionary Algorithms

The emphasis of the Evolutionary algorithms is on the genetic evolution process. They are also said to be inspired by natural selection and biological modeling of evolution. The Evolution helps the species adapt to their environments. The factors that influence the survival of the individuals may include climate, the availability of the food, presence of predators, etc. It is modeled on natural processes, such as Selection, Mutation, Crossover, and Neighborhood, etc. The environment is constrained around the parameters and in return improves the generation of potential solutions over iterations [13].

These algorithms work on a population of solutions rather than a single solution. The algorithm applies the principle of survival over iterations to produce better and better results to a solution. The algorithm randomly initializes the population and the objective function is evaluated on them. The individuals are selected for the production of the next generation depending on their fitness values in the current generation. The individuals are recombined to produce the offspring, and then the fitness of the offspring is evaluated. The offspring replaces the parents producing the new generation, and the process continues until the optimization criterion is reached.

Evolutionary algorithms are capable of providing a number of potential solutions for a given problem. It generally uses the probabilistic methods rather than the deterministic methods. Stud Genetic Algorithm (SGA) [14] is one such evolutionary algorithm where it uses only the best individual at each of the iteration for the crossover.

Following the Evolutionary Genetic Algorithms, many approaches have been proposed as an improvement of the Genetic Programming [15] for non-linear system modelling. Some of them are Multi-stage GP [16], where the simulations are based on the effect of the predictor variable as well as the interactions between the variables, this algorithm is proposed by Gandomi and Alavi; and Biogeography based optimization (BBO) [17] proposed by Simon.

### 1.2.2. Bacterial Foraging Algorithms

The Bacterial Foraging Optimization [18] technique is introduced to provide approximate solutions to extremely difficult or impossible numeric problems. The algorithm follows a probabilistic approach and simulation is based upon the food seeking and the reproductive behavior of the bacteria. Computing Systems of Microbial Interactions and Communications (COSMIC) [19] and Rule Based Bacterial Modelling (RUBAM) [20] are few famous Bacterial Foraging Algorithms in the current generation and wide research is taking place in this area.

The Krill Herd (KH) [21] algorithm is a meta-heuristic algorithm that is inspired by the bio-based swarm intelligence algorithm. The algorithm is modelled on the simulation of the herding behavior of the Krill Swarms. The distance from the food and from the highest density of the swarm forms the objective function of the krill individuals. The time-dependent position of the krill individuals is mainly influenced by the following three factors:

(i) Movement induced by other krill individuals

(ii) Foraging Activity

(iii)Random Diffusion

The objective function of the krill individuals is evaluated against the above factors and the best krill and the best position is determined over the iterations until the optimization criteria are reached. The algorithm is evaluated a number of times with respect to the changes in the number of iterations per run, population size, problem dimension, and number of runs.

Here, the algorithm is implemented in Eclipse environment using Java language and tested against five different benchmark functions. Various test cases are considered to validate the performance of the algorithm. The results are tabulated and shown in graphical representations. Comparisons were made to study the performance of the algorithm with respect to others.

The paper is organized as follows: Chapter 2 describes the Krill Herd Algorithm, Lagrangian model, and characteristics of the algorithm, formulation and the herding behavior of the krill individuals. Chapter 3 describes the experimental set up and the results obtained during the evaluation. Chapter 4 concludes the study with ideas for future work.

# 2. KRILL HERD ALGORITHM

The Krill Herd Algorithm was first proposed by Gandomi and Alavi in 2012 [21]. This is a Bio-Inspired Optimization Algorithm. This algorithm is proposed to solve complex engineering optimization problems. This algorithm studies the herding behavior of the krill swarms considering the factors influencing the position of the krill individually as well as in the herd. The algorithm is evaluated using various benchmark functions to test its efficiency. Also, it has been compared with other eight well know methods for the best optimum solution obtained.

## 2.1. Herding Behavior of Krill Swarms

Nature has always been an interesting topic in the area of research. One of the topics is identifying the grouping mechanism of various animal species. The formations are usually non-organized and not regular in manner. Many studies have taken place in order to understand the technique behind it, how the grouping mechanism is done, and how it is affected. The researchers came up with various observations, which suggest the mechanisms are mainly influenced by the (i) feeding ability, (ii) reproduction capability, (iii) predators effect, and (iv) environmental conditions. The effect of these factors varies from species to species in governing the grouping mechanism.

This paper describes one such observation made on the Antarctic Krill species [22]. This is considered as one of the best studied marine animal species. These krills are capable of forming large groups of swarms with no parallel orientation existing between them. Since decades, studies are carried out to understand the herding mechanism of the krills and the individual distribution of the krills among the herds. The studies made clear some of the factors in spite of some uncertainties observed. It says, the basic unit of the organization is the ‘krill swarm’.

As mentioned earlier, one important factor considered here is the presence of the predator. When one krill is affected by the predator, that krill is abandoned from the herd, which in turn reduces the density of the swarm. The increase in the krill density and distance from the food is always the phenomenon or the criteria to be reached during the formation of the herds. Hence, it can be called as one multi-objective mechanism.

The optimum solution or the global minima is attained considering the two main objectives of density-dependent attraction of the krill and areas of high food concentration. The relation between the objective function and the krill position is proportionate in nature. The global minimum, or less objective function value, implies the minimum distance of a krill from the highest density and food in the solution. The objective function of the algorithm is determined with some coefficients determined after a brief empirical study. The krill individuals always try to move towards the best solution in this process.

## 2.2. Lagrangian Model of Krill Herding

The Lagrangian model [23] states the objective function of the Krill Herd algorithm. The fitness function of a krill is a combination of the highest density of the krill and the distance of food from the krill. The fitness value gives the imaginary distances of the krill from the herd density and food.

In a two dimensional surface, the time-dependent position of a krill is mainly influenced/controlled by the following three factors:

(i) Movement induced by other krill individuals

(ii) Foraging activity

(iii)Random diffusion

Extending the algorithm to an n-dimensional space, the fitness function of the algorithm (for i$^{th}$ krill individual) is determined as below:

$$\frac{dX_i}{dt} = N_i + F_i + D_i \tag{1}$$

Where $N_i$: Motion induced on i$^{th}$ krill individual due to the other krill individuals

$F_i$: Foraging motion

$D_i$: Random diffusion

### 2.2.1. Motion Induced by other Krill Individuals

The fitness function of the algorithm mainly depends on the density of the krills in the search space. So, it is essential to maintain a high krill density in order to achieve an optimum solution. The individuals keep on rebuilding the system maintaining this high density under the influence of the other individuals. The movement of a krill individual is mainly dependent on the neighboring krill individuals and the mutual effects between them.

The direction of the krill movement, $\boldsymbol{\alpha_i}$ , is calculated based upon the different swarm densities i.e., i) 'local effect' provided by local krill density, ii) 'target effect' provided by target krill density, and iii) 'repulsive effect' provided by repulsive swarm density. Considering the influence of all these effects, the movement of the i$^{th}$ krill individual $N_i$ can be defined as:

$$N_i^{new} = N^{max}\alpha_i + \omega_n N_i^{old} \tag{2}$$

Where, $\alpha_i = \alpha_i^{local} + \alpha_i^{target}$

$N^{max}$: Maximum induced speed

$\omega_n$: Inertia weight

$N_i^{old}$: Previous motion induced

$\alpha_i^{local}$: Local effect provided by the neighboring krill individuals

$\alpha_i^{target}$: Target effect provided by the best krill individual

Calculation of Neighbors Krill Effect:

From the empirical study, the inertial weight of the motion induced $\omega_n$, is said to be in the range of [0, 1], and the $N^{max}$ the maximum induced speed is calculated to be around 0.01 $ms^{-1}$. The neighbors of a krill individual have either an attractive or a repulsive effect on the movement of the krill. The effect of the neighbors on the movement of a krill individual is generally calculated by the following equations:

$$\alpha_i^{local} = \sum_{j=1}^{NN} \widehat{K}_{ij}\,\widehat{X}_{ij} \tag{3}$$

$$\widehat{X}_{ij} = \frac{X_j - X_i}{\|X_j - X_i\| + \varepsilon} \tag{4}$$

$$\widehat{K}_{ij} = \frac{K_i - K_j}{K^{worst} - K^{best}} \tag{5}$$

Here, $K^{best}$ and $K^{worst}$ represent the best fitness and worst fitness values of the krill individuals so far.

- $K_i$: Fitness value of the $i^{th}$ krill individual
- $K_j$: Fitness value of the $j^{th}$ neighbor (j = 1, 2, …,NN) NN: total number of neighbors
- X: Relative position of the krills
- ε: Small positive number

The effect induced by the neighbors is obtained from the above equations. Each krill individual 'i' is taken into consideration, and the $\alpha_i^{local}$ value is calculated. The $K_{i,j}$ is the fitness value of the $i^{th}$ individual under the influence of $j^{th}$ neighbor. All the neighbor pair's fitness values are calculated. Similarly, the relative position vector $X_{i,j}$ is determined. The $\alpha_i^{local}$ , that is the cumulative sum of the local effect of the neighbors (j=1,2,..,NN) is calculated from the relative position $X_{i,j}$ and relative fitness $K_{i,j}$ values. The alpha vector gives the induced directions by the different neighbors and each value represents the effect of a neighbor on a krill individual. The positive value determines the attractive effect while the negative value indicates the repulsive effect on the krill individual from its neighbors.

The important part in the evaluation of the Krill movement is the identification of the neighbors and their consideration. From all the individuals in the local search space, only some krills are said to be a neighbor of $i^{th}$ krill individual and vice versa. The identification of the neighbors is done using different strategies present in the real time scenario. One among them is from the calculation of the "Neighborhood ratio" value. It is defined in order to determine the number of nearest krill individuals/neighbors present in the search space around the $i^{th}$ krill individual located at position $X_i$. The sensing distance '$d_s$' is calculated from this individual and all the neighbors located within this distance are said to be neighbors of krill individual 'i'.

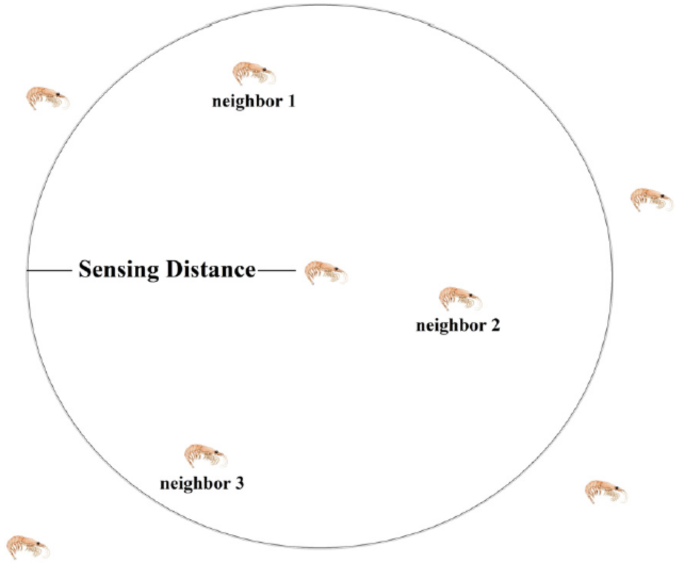

**Figure 1. Schematic representation of Sensing Distance around a krill individual 'i'**

From Figure 1, we can say the krill individual has three neighbors namely neighbor 1, 2, 3 that fall inside the sensing ambit.

The sensing distance of i$^{th}$ krill individual $d_{s,i}$ is determined from the following equation using one of the heuristic methods.

$$d_{s,i} = \frac{1}{5N} \sum_{j=1}^{N} \|X_i - X_j\| \tag{6}$$

Where N represents the number of krill individuals in the search space.

According to the study, the distance between the krill individuals is compared against the sensing distance and if it is encountered to be less than $d_{s,i}$, then the two krill individuals are said to be neighbors.

Calculation of Best Krill Effect:

The direction of motion induced due to the best krill individual in the herd is said to be referred to the target effect. The other krill individuals in the herd try to reach the global optima. In general, a krill individual with the lowest fitness value is said to be the best krill individual in the herd. All the other individuals try to reach this target vector of lowest fitness. The effect of this best fitness and best fitness krill individual on the i$^{th}$ krill individual is calculated by the following equation.

$$\alpha_i^{target} = C^{best} \hat{K}_{i,best} \hat{X}_{i,best} \tag{7}$$

Where $C_{best}$: Effective coefficient of best fitness krill individual to the i$^{th}$ krill individual.

Iterating over this equation, $\alpha_i^{target}$ value leads us in achieving the global optimum solution.

The $C_{best}$ value should be calculated efficiently, as it has significant importance in achieving the global optima, when compared to the local effect due to the neighbors. The $C_{best}$ value is calculated using the following equation.

$$C^{best} = 2\left(rand + \frac{I}{I_{max}}\right) \tag{8}$$

Where rand: a random variable in the range of [0,1].

I: current iteration index

$I_{max}$: total/maximum number of iterations

### 2.2.2. Foraging Motion

The Krill herding mechanism is influenced by the presence of food around the individuals. The minimum distances of the krill individual from the food is said to be considered as one of the objective functions for the movement of the krill's in the solution space. The less the distance to the food as well as to the highest density of the herd, the less the objective function value which in turn implies the best fitness for that krill individual. Iterating over the process, it eventually leads to the global optima.

Here, the foraging motion is nothing but the motion induced to a krill individual due to the presence of food and its locations in the solution. The foraging motion considers two prominent factors, namely 1) the food location, and 2) the previous experience about the food location.

The foraging motion value for the $i^{th}$ krill individual is calculates by the following equation:

$$F_i = V_f \beta_i + \omega_f F_i^{old} \tag{9}$$

Where $V_f$ : Foraging speed

$\omega_f$ : inertia weight of the foraging motion

$F_i^{old}$ : Last foraging motion value

From the study, the inertia weight is calculated to be in the range of [0,1] and the foraging speed $V_f$ is taken to be 0.02 $ms^{-1}$.

The value of $\beta_i$ is calculated as follows:

$$\beta_i = \beta_i^{food} + \beta_i^{best}$$

Where $\beta_i^{food}$ : Effect due to the presence of food

$\beta_i^{best}$ : Effect due to the current krill's best fitness value recorded

The effect of food on the herding mechanism is defined depending on the food's location. Here, we are calculating the center of the food location, and then the food attraction value $\beta_i^{food}$ is calculated. The center of the food location is estimated from the krills fitness distribution values. We need to calculate the center of food location in the iteration through the krill size N, and it gets updated through the iterations. The equation to calculate $X^{food}$, center of food location is as follows:

$$X^{food} = \frac{\sum_{i=1}^{N} \frac{1}{K_i} X_i}{\sum_{i=1}^{N} \frac{1}{K_i}} \tag{10}$$

Where, $X_i$ and $K_i$ are the krills current position and fitness values.

Calculation of Food Attraction:

The attractive effect of food on the ith krill individual is obtained by the following equation:

$$\beta_i^{food} = C^{food} \hat{K}_{i,food} \hat{X}_{i,food} \tag{11}$$

Where $C^{food}$ : Effective food coefficient

The $C^{food}$ is calculates as follows:

$$C^{food} = 2\left(1 - \frac{I}{I_{max}}\right) \tag{12}$$

Where $I_{max}$ : Total/Maximum number of iterations.

From the above equation we can say that the effect of food on the Krill herding mechanism decreases over time.

Calculation of Best Position Attraction:

The ultimate goal of the algorithm is to achieve the global optima; hence the effect of food attraction on the krill individuals decreases after some iteration once the global optima criterion has been reached. This is considered to be an efficient global optimization strategy that improves the globality of the Krill Herd algorithm.

The effect of the i$^{th}$ krill individual’s best fitness so far, is determined by the following equation:

$$\beta_i^{best} = \widehat{K}_{i,ibest}\hat{X}_{i,ibest} \tag{13}$$

Where $K_{ibest}$: fitness value of the previously visited position

$X_{ibest}$: previous position of i$^{th}$ krill individual

Calculating the food attraction and best position effect, the Food fitness values are obtained and the value $K_{food}$, is made sure to be the best food location (minimum fitness) by comparing it against the previous food location in the last iteration and is updated accordingly.

### 2.2.3. Physical Diffusion

The third factor that influences the herding mechanism of the Krill individuals is the Physical Diffusion process. This is nothing but the random diffusion of the krill individuals in the solution space.

The krill movement is calculated from the speed of the krill movement and a randomly generated directional vector. The equation to calculate this diffusion effect is as follows:

$$D_i = D^{max}\delta \tag{14}$$

Where $D^{max}$: Maximum diffusion speed

$\delta$: Random directional vector

The random directional vector, $\delta$ with values ramping in between [-1, 1].

From the studies of Wolpert and Macready, it is observed that for a krill individual, the range of the maximum diffusion speed lies in between [0.002, 0.0010] ($ms^{-1}$).

The effect of this random diffusion is inversely proportional to the best position of the krill, that implies when the krill individual is in its best position, then the random diffusion of that krill individual is also less. Thus, another term is added to take this effect into consideration. Also, the physical diffusion is a random process and this random vector does not steadily reduce with an increase in the iteration index. To avoid this effect, an additional term is added.

Hence, the above equation is modified to the following:

$$D_i = D^{max}\left(1 - \frac{I}{I_{max}}\right)\delta \qquad (15)$$

The above added term makes the random diffusion speed to decrease linearly over the time. Here, the random diffusion speed of every krill individual (to the maximum of N) is calculated over all the iterations.

### 2.2.4. Motion Process of the KH Algorithm

The KH algorithm is developed to achieve the global optima for the whole solution. Also, in this process, each krill individual moves towards the best fitness position as the direction of motion changes due to the effects of neighbor krills and the presence of food. The two factors i) motion induced by other krill individuals and the foraging motion always have global and local effects on the krill individuals, i.e, Local Effect: Neighbor krill's effect ($\alpha_i^{local}$), and Global Effect: Best Krill effect ($\alpha_i^{target}$) in the case of motion induced by other krill individuals, and secondly Local Effect: Presence of Food ($\beta_i^{food}$), and Global Effect: Best fitness position ($\beta_i^{best}$). These two strategies work in parallel to improve the efficiency of the Krill Herd Algorithm.

From the study, considering the respective fitness values of the above mentioned global and local effects, i.e., $K_j$, $K^{best}$, $K^{food}$, $K_i^{best}$ of an i$^{th}$ krill individual, if any of this fitness values is better i.e., less than the previous positions fitness value of the i$^{th}$ krill individual, then it is said to be having an attractive effect on the krill individual. Otherwise, if the fitness values are worse, that is greater than the previous fitness value, then it is said to be having a repulsive effect. From the above equations, it is observed that the effect of the best fitness on the krill individual is having much impact on the krill movement when compared to the other effects. The physical diffusion/random diffusion process performs a random search in the solution space.

Using all the effective parameters ($N_i$, $F_i$, $D_i$) of the motion obtained over time, the random position of the krill individual during the interval t and Δt can be obtained by the following equation and the position of the each krill individual is updated,

$$X_i(t + \Delta t) = X_i(t) + \Delta t \frac{dX_i}{dt} \tag{16}$$

and dXi = N(i) + F(i) + D(i)

Where Δt, is referred to as Scale factor, and it should be set carefully as it is considered to be one of the most important constants in solving the Krill herd optimization problem. This constant will be working as a scale factor of the speed vector.

The scale factor, Δt, can be obtained by the following equation and it also depends on the solution search space and its dimensions.

$$\Delta t = C_t \sum_{j=1}^{NV} (UB_j - LB_j) \tag{17}$$

Where $UB_j$, $LB_j$: Upper and lower bounds of the variable where j varies from 1 to number of variables.

NV: Number of Variables.

$C_t$: Empirical constant that lies between [0, 2].

Considering the lower and upper bounds, the absolute difference between these bounds gives the information about the solution/search space. And from the studies [21], it is observed that the values of $C_t$ are directly related to the search space. The lesser the value of $C_t$ is, the more efficiently the individuals in the krill herd can search through the space.

Once the new positions of the each krill individual are obtained, the positions are updated and the process is repeated, by evaluating the each krill individual with the updated positions and calculating the induced motion from the effects of neighbors, food and physical diffusion and so on, until the global optima or the criterion has been reached.

## 2.3. Methodology of the Krill Herd Algorithm

The Krill Herd Optimization algorithm usually starts with a set of krill individuals located in the search space or distributed in the search space randomly. And at each generation of optimization, the movement of the krill individuals, foraging activity and random diffusion parameters are calculated from the above mentioned equations, respectively. Finally, the displacements of the krill individuals from the previous positions due to these effects are calculated by Equation 16. The position of each krill individual is updated by this equation and the evaluation process takes place. Eventually, the Krill herd optimization algorithm is terminated when the satisfaction criterion of global maxima or global minima is obtained; here the criterion is global minima which implies the best fitness value for the overall solution and as well as for the each krill individual.

Summarizing the Krill herd algorithm and laying out a step by step procedure, the Krill herd algorithm can be described by the following steps:

**I. Data Structures**:

Define the simple bounds, determination of algorithm parameters, etc. Some of the parameters are defined in the algorithm as below:

numberOfRuns -------- Number of Runs

numberOfIterations -------- Number of Iterations

krillSize --------- Number of Krill’s

dimension ---------- Number of Dimensions

Nmax -------- Maximum Induced Speed

Vf --------- Foraging Speed

Dmax --------- Maximum Diffusion Speed

rand -------- Random variable

epsilon ------- Small positive number

**II**. **Initialization**: Randomly create the initial population in the search space.

**III**. **Fitness Evaluation**: Evaluation of each krill individual according to its position

```
A- For i = 1: krillSize do
    Xvector (i) = random number (between the maxPosition and minPosition)
    Kvector (i) = evaluate (Xvector(i))
B- $K_{ib}$=K; best fitness values for whole swarm so far
    $X_{ib}$=X; best position for whole swarm so far
C- Find the krill that has the best fitness and its position
    Kgbest = min(K)
    Xgbest= return the position of Kgbest
```

**IV**. **Motion Calculation**:

- Motion induced by the presence of other individuals
- Foraging motion
- Physical Diffusion

**V. Updating**: updating the krill individual position in the search space

$dX_i$ = delta_t*(N(i)+F(i)+D(i));

X(i)=X(i)+$dX_i$; Update the position

**VI. Repeating**: go to step III until the stop criteria is reached

K(i)=evaluate(X(i)) and update the current best Kgbest

**VII. End**

**Table 1. Parameters and definitions**

| **Parameter** | **Notation in Algorithm** |
|---|---|
| Number of generations | numberOfRuns |
| Number of Neighbors | NN |
| Number of Dimensions | dimension |

The above table gives the mapping of the parameters and corresponding notations used in the algorithm below.

**Algorithm**:

```
Initialize the benchmark functions f(x)
Initialize the parameters
  for p=1:numberOfRuns
     Randomly generate the krill population x(i=1, 2,…,krillSize)
       for q=1:numberOfIterations
          for k=1:krillSize
            Evaluate the krills fitness values
```

```
              Find bestFitness, worstFitness, bestKrillIndex, bestPosition
            end
            Find the virtual Food
                  for i=1:dimension
                      for j=1:krillSize
                         vFood(i) = $\sum \left(\frac{X_i(1:krillSize)}{K}\right)$ ;
                      end
                      Calculate xFood, kFood
                  end
Update inertia
   for i=1:krillSize
     for j=1:dimension
        A- Motion induced by other Krill Individuals
                 N[i][j] = inertia*N[i][j] + Nmax*(alpha_local[i][j]+alpha_target[i][j]);
       B- Foraging Motion
                  F[i][j] = inertia*F[i][j] + $V_f$*(beta_best[i][j]+beta_food[i][j]);
       C- Random Diffusion
                  D[i][j] = Dmax*(1-q/numberOfIterations)*segma;
       Calculate the scale factor
       Update the krill positions
                $dX_i$[i][j] = scaleFactor*(N[i][j]+F[i][j]+D[i][j]);
                $X_i$(j)=$X_i$(j)+$dX_i$[i][j]; Update the position
      end
```

```
    end
 end
Plot the results
end
```

Representing the Krill Herd Algorithm diagrammatically, we have

Krill Herd Algorithm Flow Chart:

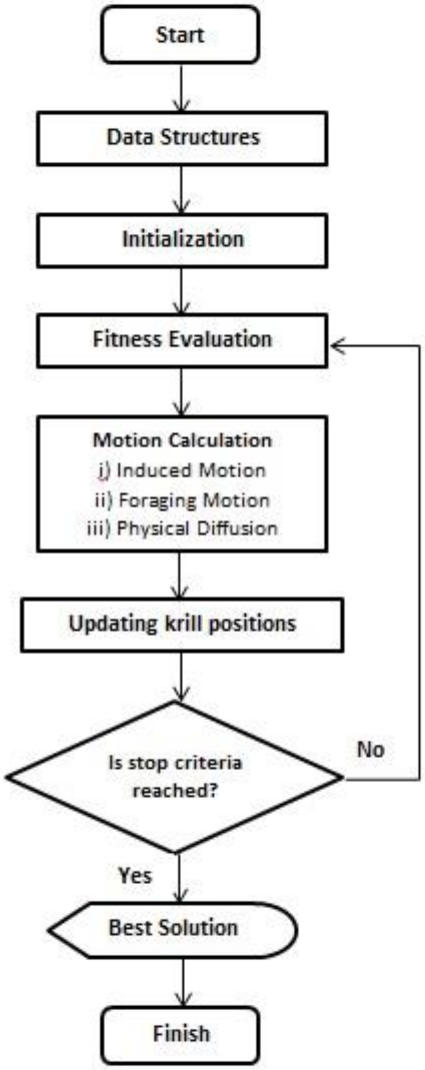

**Figure 2. Simplified flowchart of the krill herd algorithm**

# 3. BENCHMARK PROBLEMS

In the real world, there is a need of various test functions in order to compare and validate the optimization algorithms for its performance and reliability. These test functions which are also referred as benchmark functions are tested against many new optimization algorithms that come into existence to test their reliability, efficiency and validity of the algorithm [24]. These are nothing but a set of common standard functions that have diverse properties to judge the new optimization problems in an unbiased way.

Many researchers follow the process of comparing different algorithms on a large test set, whenever there is the involvement of optimization techniques. Identification of the kind of problems where it performs the best when compared to others, helps in the categorization of the problems for which the algorithm suits best. Therefore, the benchmark functions are categorized in to variety of problems [25], such as

a) Unimodal and Multimodal Functions,

b) Regular and Irregular Functions,

c) Separable and non-separable Functions, and

d) Multi-dimensional Functions.

In this paper, the Krill Herd Optimization algorithm is tested against five benchmark problems that are defined below with their mathematical notation. These functions are implemented in order to evaluate the performance of the algorithm, in terms of efficiency of obtaining the optimum solution and the rate of convergence to the optimum solution.

## 3.1. Ackley Function

Ackley is a multimodal non-separable function [25]. It is widely used for validation and testing purposes of algorithms.

$$\mathrm{f(x)} = -\mathrm{a.exp}\left(-0.02.\sqrt{\frac{\sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{x_i^2}}{\mathrm{n}}}\right) - \exp\left(\frac{\sum_{\mathrm{i=1}}^{\mathrm{n}} \cos(\mathrm{cx_i})}{\mathrm{n}}\right) + \mathrm{a} + \exp(1) \tag{18}$$

where, a = 20, n is the dimension of the problem and the test area lies between $-32 \leq x_i \leq 32$, i=1,2,…,n. Its global minimum f(x) = 0 is obtainable for $x_i$=0, i= 1,…,n.

### 3.2. Alpine Function

The Alpine function is a non-separable function f(x), and is called an m-nonseparable function, if at most m of its parameters $x_i$ is not independent. A non-separable function f(x) is called fully non-separable function if any two of its parameters $x_i$ are not independent.

$$f(x) = \sum_{i=1}^{n} |x_i sin(x_i) \ + \ 0.1x_i| \tag{19}$$

where, n ≥2 is the dimension and x = ($x_1$, $x_2$, …, $x_d$) is a n-dimensional row vector (i.e., a 1×n matrix). Test area is usually restricted to -10 ≤ $x_i$ ≤ 10, i=1,…,n. Its global minimum F(x) = 0 is obtainable for $x_i$, i=1,…,n.

### 3.3. Griewank Function

The Griewank function is a continuous and a non-separable multimodal function. The product term in the equation introduces the interdependence among the variables. The optimization techniques optimize each variable independently. The global optima of Griewank function resemble a regularly distributed plot.

$$f(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{20}$$

where, n ≥2 is the dimension and the test area is restricted to $-600 \leq x_i \leq 600$, and i=1, 2, …,n. Its global minimum f(x) = 0 is obtainable for $x_i$, i=1,2,…,n.

### 3.4. Rastrigin Function

The Rastrigin function is based on the function of De Jong [26] with the addition of Cosine modulation in order to produce frequent local minima and is defined as follows:

$$f(x) = 10n + \sum_{i=1}^{n} \left( x_i^2 - 10cos(2\pi x_i) \right) \tag{21}$$

where, n is the dimension and x = (x1, x2, … , xd) is a n-dimensional row vector (i.e., a 1×n matrix). Similarly, to make it nonseparable, an orthogonal matrix is also used for coordinate rotation. Rastrigin's function is a classical multimodal problem. It is difficult since the number of local optima grows exponentially with the increase of dimensionality. Test area is usually restricted to a hypercube $-5.12 \leq x_i \leq 5.12$, i=1,…,n. Its global minimum f(x)=0 is obtainable for $x_i$=0, i= 1,…,n.

### 3.5. Sphere Function

The Sphere function is a simple and a continuous benchmark function [24] and belongs to the unimodal, multi-dimensional benchmark function. It is mathematically defined as:

$$\mathrm{f(x)} = \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x}_\mathrm{i}^2 \tag{22}$$

where n is the dimension of the problem and the test area lies between $-5.12 \leq x_i \leq 5.12$, i=1,2,…,n. Its global minimum f(x)=0 is obtainable for $x_i$=0, i= 1,…,n.

**Table 2. Benchmark problems**

| **Function Name** | **Search Space** |
|---|---|
| Ackley | -32 to +32 |
| Alpine | -10 to +10 |
| Griewank | -600 to +600 |
| Rastrigin | -5.12 to +5.12 |
| Sphere | -5.12 to +5.12 |

The krill herd optimization problem is tested using the above five benchmark functions mentioned in Table 2 for its performance and efficiency of the technique.

# 4. EXPERIMENTAL SETUP AND RESULTS

## 4.1. Experimental Setup

As said in the earlier section, the Krill Herd algorithm is evaluated to test the validity and performance of the optimization technique using some well-known benchmark functions that are mentioned, such as Alpine function, Ackley function, Griewank function, Rastrigin function and finally the Sphere function. These benchmark functions are tested against different dimensions such as both low and high dimension problems, and the algorithm is evaluated. Similarly, the algorithm is evaluated with different configurations by varying the krillSize, number of iterations, and number of runs. The results obtained are observed and comparisons are made to evaluate the behavior and efficiency of the algorithm in reaching the global minima with respect to the time required.

As the Krill Herd algorithm deals with the random behavior of the krill individuals and is not easy to determine the end results, the performance of the algorithm needs to be judged by running the algorithm in multiple runs with different number of iterations. Similarly, different attempts with independent krill population initializations are made to obtain the best conclusion about the Krill Herd algorithm in terms of performance.

### 4.1.1. General Settings

The algorithm has some of the general settings with some constant parameters such as:

a) Inertia Weight ω: is taken as 0.9 at the beginning of the search and it gets updated for every iteration according to the following equation: (here ω refers to both $\omega_n$, $\omega_f$)

$$\omega = (0.1 + 0.8 * (i/maxIterations)) \tag{23}$$

where i: Current iteration number

maxIterations: Total number of iterations in a single run.

b) Maximum Foraging Speed, $V_f$ : is a constant value throughout the experiment and is equal to 0.02 ($ms^{-1}$).

c) Maximum Induced Speed, $N_{max}$ : is a constant value throughout the experiment and is equal to 0.01 ($ms^{-1}$).

d) Maximum Diffusion Speed, $D_{max}$ : is a constant value throughout the experiment and is equal to 0.005 ($ms^{-1}$).

e) Random number, rand = 0.9.

f) Small positive number, epsilon $\varepsilon = 0.00001$.

The Search space for the Krill Herd algorithm is different for each of the benchmark functions. Once the termination criterion, that is the maximum number of fitness/objective function evaluations are performed, the algorithm is terminated.

The global optimum value for the above mentioned five benchmark functions is 0.0.

The number of simulations/runs = 25, are made for each of the five benchmark functions with varying parameters of krillSize, numberOfIterations, and dimensions.

### 4.1.2. Test Cases

Case (i):

Number of Runs: 25, Number of Iterations: 1000, Problem Dimension: 4, Krill Size 50.

For the 25 simulations, all the parameters are kept fixed and the best fitness, average fitness, standard deviation and the execution time are observed.

Case (ii):

Number of Runs: 25, Number of Iterations: 1000, Krill Size: 50.

For the constant 25 simulations, the problem dimension is changed for all the five benchmark functions; dimension is varied from 2, 4, 6, 8 and the effect of the change in Fitness values is observed.

<u>Case (iii)</u>:

Number of Runs: 25, Krill Size: 50, Problem Dimension: 4.

For the constant 25 simulations, the maximum number of iterations or the number of function evaluations is changed; noOfIterations varying from 100 to 1000, 10000 and to 100000, and the effect of the change in Fitness Evaluations is observed.

<u>Case (iv)</u>:

Number of Iterations: 50, Krill Size: 50, Dimension: 4 and Number of Runs: 1

This test case involves plotting the graph of Best fitness over Number of Iterations and the behavior of the algorithm over five different benchmark functions is observed.

**<u>Observations</u>**:

From the simulations performed, on all the five benchmark functions mentioned above, the following results have been collected:

1. The best fitness value
2. The average fitness value
3. The standard deviation value
4. The time elapsed in executing the simulation

### 4.2. Results

The Krill Herd algorithm is tested against five different benchmark functions via different test cases, as mentioned earlier. This section gives a clear explanation of the test cases that are performed to evaluate the performance of the Krill Herd algorithm. The test cases have a varying parameter and the effect of the parameter on the best fitness, average fitness and the standard deviation is evaluated.

#### 4.2.1. Test Case 1: Fixed Number of Iterations = 1000

Settings: Krill Size=50 & Dimension 4 & Runs 25.

Table 3 shows the average results of 25 independent runs on the benchmark function Ackley. The best fitness value of the Krill Herd algorithm is 6.88E-06, the mean of the 25 independent runs was 0.001147 and the value of the Standard deviation is 0.005504. The time elapsed for performing 25 independent runs and to collect the results was 644 ms.

**Table 3. Ackley benchmark function outputs**

| Best | Average | Standard Deviation | Time (in ms.) |
|---|---|---|---|
| 6.88E-06 | 0.001147 | 0.005504 | 644 |

Table 4 shows the results of 25 independent runs on the benchmark function Alpine. The best fitness value of the Krill Herd algorithm is 4.83E-11, the mean of the 25 independent runs was 4.04E-07 and the value of the Standard deviation is 1.94E-06. The time elapsed for performing 25 independent runs and to collect the results was 643 ms.

**Table 4. Alpine benchmark function outputs**

| Best | Average | Standard Deviation | Time (in ms.) |
|---|---|---|---|
| 4.83E-11 | 4.04E-07 | 1.94E-06 | 643 |

Table 5 shows the results of 25 independent runs on the benchmark function Griewank. The best fitness value of the Krill Herd algorithm is 0.006416, the mean of the 25 independent runs was 0.472084 and the value of the Standard deviation is 2.266001. The time elapsed for performing 25 independent runs and to collect the results was 381 ms.

**Table 5. Griewank benchmark function outputs**

| Best | Average | Standard Deviation | Time (in ms.) |
|---|---|---|---|
| 0.006416 | 0.472084 | 2.266001 | 381 |

Table 6 shows the results of 25 independent runs on the benchmark function Rastrigin. The best fitness value of the Krill Herd algorithm is 0.000679, the mean of the 25 independent runs was 0.158179 and the value of the Standard deviation is 0.759258. The time elapsed for performing 25 independent runs and to collect the results was 380 ms.

**Table 6. Rastrigin benchmark function outputs**

| Best | Average | Standard Deviation | Time (in ms.) |
|---|---|---|---|
| 0.000679 | 0.158179 | 0.759258 | 380 |

Table 7 shows the results of 25 independent runs on the benchmark function Sphere. The best fitness value of the Krill Herd algorithm is 4.58E-07, the mean of the 25 independent runs was 4.76E-05 and the value of the Standard deviation is 0.000229. The time elapsed for performing 25 independent runs and to collect the results was 617 ms.

**Table 7. Sphere benchmark function outputs**

| Best | Average | Standard Deviation | Time (in ms.) |
|---|---|---|---|
| 4.58E-07 | 4.76E-05 | 0.000229 | 617 |

From the above results we can say that, the Alpine benchmark function achieved best results in terms of best fitness value (4.83E-11), average value (4.04E-07) and standard deviation with a very low value of 1.94E-06. The Griewank function performed poorly when compared to other benchmark functions with a high standard deviation value of 2.266001. The Rastrigin function is the least complex function to evaluate in terms of evaluation time. The average time taken to execute a single run is 380 ms and this is closely followed by Griewank function with an execution time of 381ms. All the others functions performed moderately.

### 4.2.2. Test Case 2: Varying Problem Dimension = 2, 4, 6, 8

The following section discusses the results obtained by varying the problem dimension over a constant number of iterations. 25 independent repetitions have been performed for five functions and the best, average, and standard deviation of all the 25 independent runs has been reported. Table 8 to Table 10 shows the collected results.

Effect of the varying dimension on the Best Fitness value: (Krill Size 50, Iterations 1000)

**Table 8. Best fitness values for function on varying dimension**

| Function Name | Dim = 2 | Dim = 4 | Dim = 6 | Dim = 8 |
|---|---|---|---|---|
| Ackley | 3.43E-07 | 2.44E-05 | 4.95E-05 | 0.000137 |
| Alpine | 3.52E-10 | 7.18E-10 | 1.07E-09 | 4.57E-09 |
| Griewank | 0.000392 | 0.002856 | 0.018128 | 0.031813 |
| Rastrigin | 7.45E-07 | 0.000887 | 0.004092 | 0.004916 |
| Sphere | 1.1E-09 | 4.88E-07 | 6.24E-06 | 8.65E-06 |

As the dimension of the Krill algorithm increased, the best fitness values obtained from the algorithm are increasing as expected; since it requires more iteration to reach the best values.

Effect of the varying dimension on the Standard Dev: (Krill Size 50, Iterations 1000)

**Table 9. Standard deviation values for function on varying dimension**

| Function Name | Dim = 2 | Dim = 4 | Dim = 6 | Dim = 8 |
|---|---|---|---|---|
| Ackley | 0.000424 | 0.005438 | 0.01657 | 0.030205 |
| Alpine | 7.8E-07 | 2.01E-06 | 3.23E-06 | 7.47E-06 |
| Griewank | 0.417563 | 2.782524 | 12.5561 | 10.38485 |
| Rastrigin | 0.054719 | 3.15004 | 3.251203 | 4.118589 |
| Sphere | 9.35E-07 | 0.000207 | 0.003084 | 0.012852 |

As the dimension of the krill algorithm increased, the standard deviation is increased as expected except for the Griewank function.

Effect of the varying dimension on the Average Fitness Value: (Krill Size 50, Iterations 1000)

**Table 10. Average results for function on varying dimension**

| Function Name | Dim = 2 | Dim = 4 | Dim = 6 | Dim = 8 |
|---|---|---|---|---|
| Ackley | 8.83E-05 | 0.001133 | 0.003452 | 0.006293 |
| Alpine | 1.63E-07 | 4.18E-07 | 6.72E-07 | 1.56E-06 |
| Griewank | 0.086992 | 0.579693 | 2.615854 | 2.163511 |
| Rastrigin | 0.0114 | 0.656258 | 0.677334 | 0.858039 |
| Sphere | 1.95E-07 | 4.31E-05 | 0.000643 | 0.002678 |

Presented below are the individual graphs for each function displaying the variation of mean value over 25 independent runs with increase in the problem dimension value.

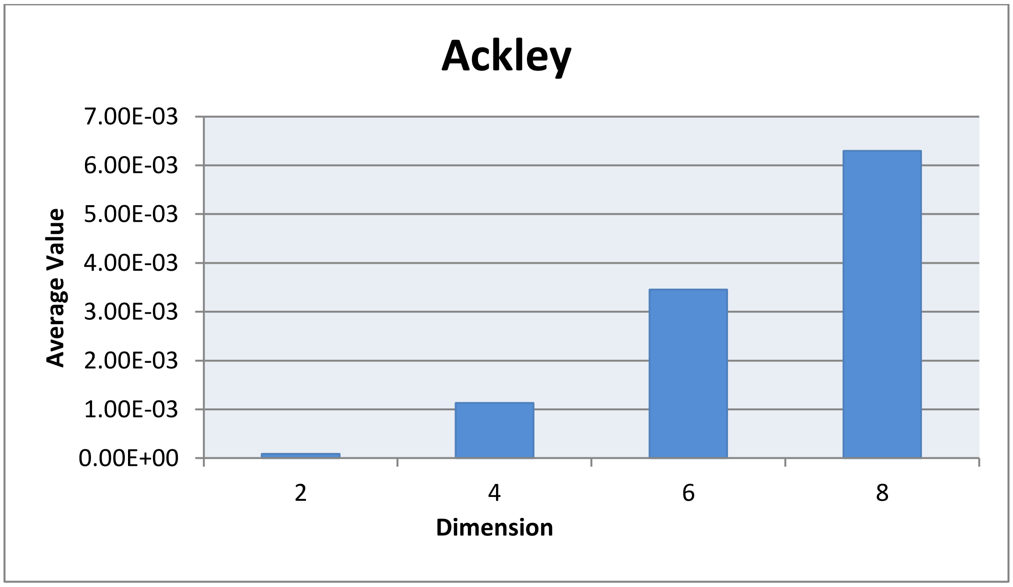

**Figure 3. Average values for Ackley against dimension**

The variation of the mean versus augmenting dimension value for Ackley is shown in Figure 3. The lowest mean value 8.83E-05 was for dimension 2, and the highest mean value 0.006283 was obtained for dimension 8.

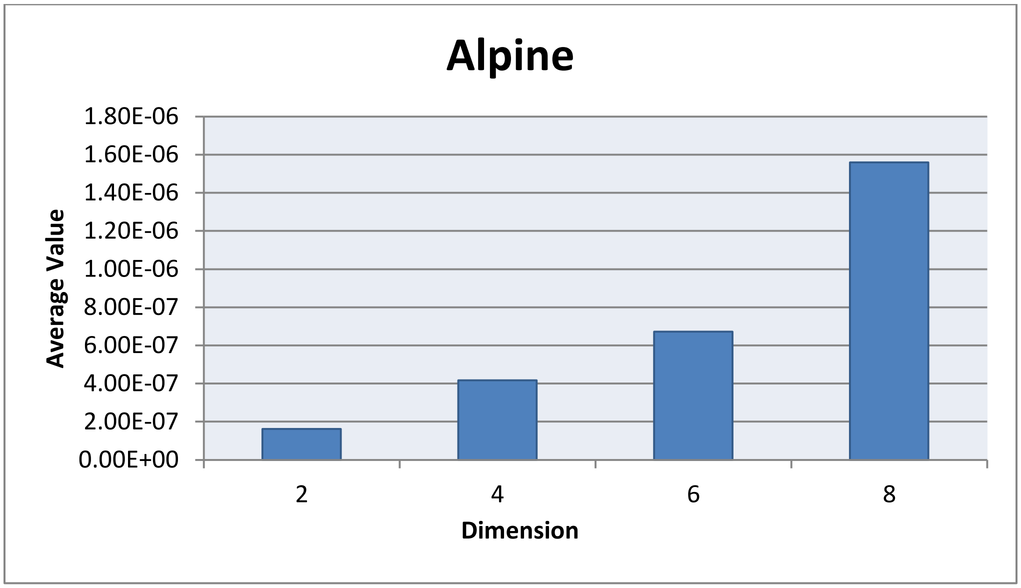

**Figure 4. Average values for Alpine against dimension**

The variation of the mean versus augmenting dimension value for Alpine function is shown in Figure 4. The lowest mean value 1.63E-07 was achieved for dimension 2, and the highest mean value 1.56E-06 was obtained for dimension 8.

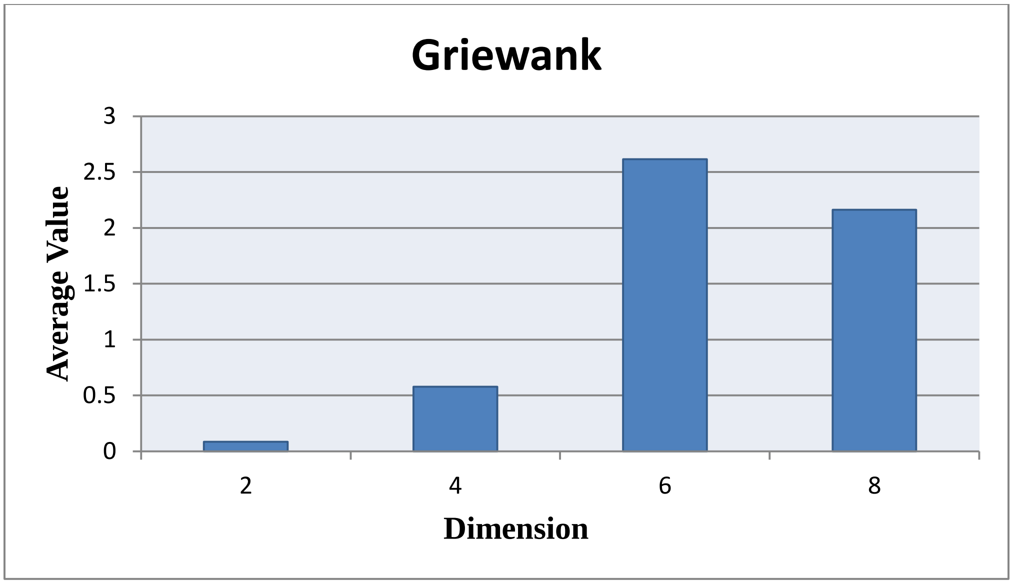

**Figure 5. Average values for Griewank against dimension**

The variation of the mean versus augmenting dimension value for Griewank is shown in Figure 5. The lowest mean value 0.086992 was at dimension 2 and the highest mean value 2.615854 was at dimension 6.

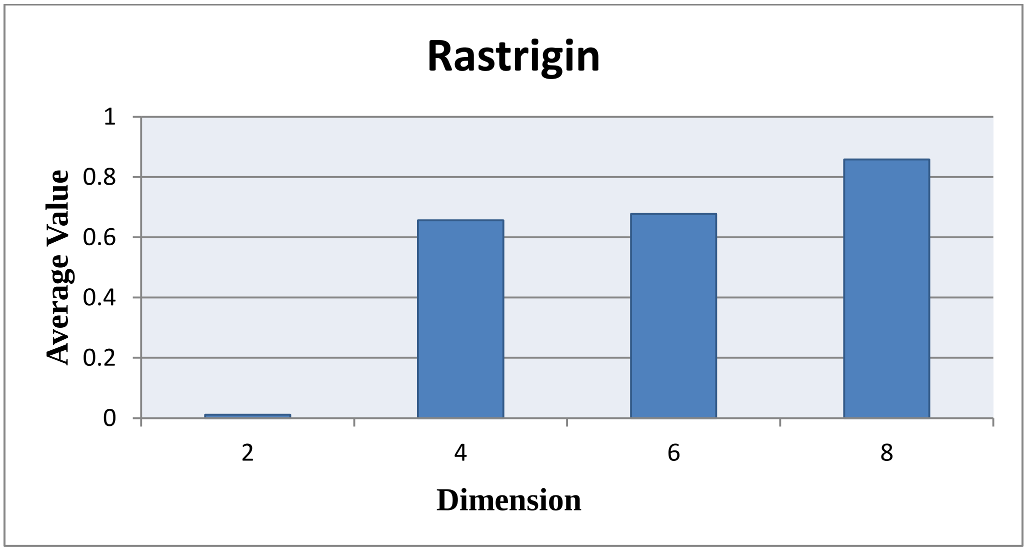

**Figure 6. Average values for Rastrigin against dimension**

The variation of the mean versus augmenting dimension value for Rastrigin is shown in Figure 6. The lowest mean value 0.0114 was at dimension 2 and the highest mean value 0.858039 was at dimension 8.

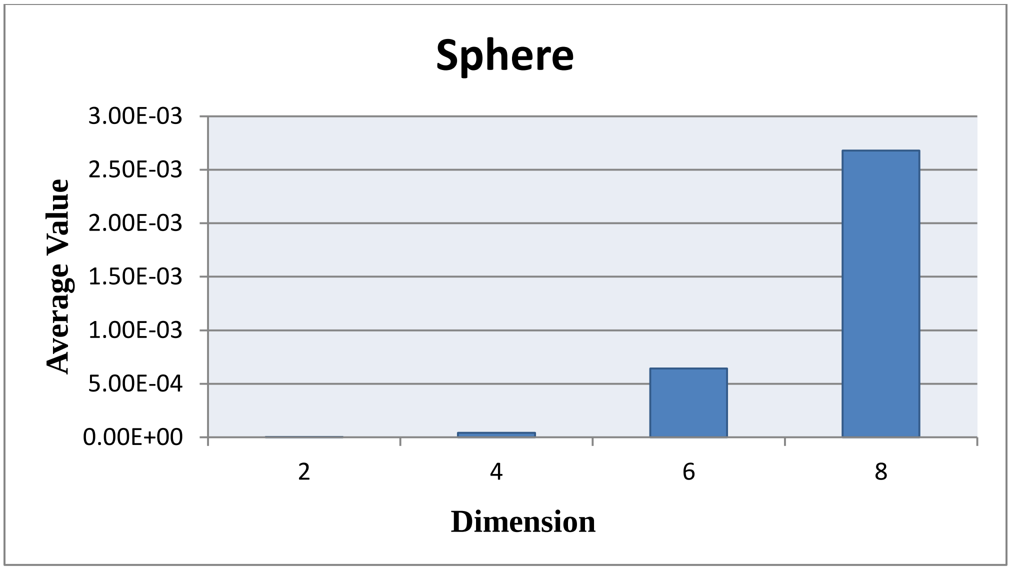

**Figure 7. Average values for Sphere against dimension**

The variation of the mean versus augmenting dimension value for Sphere is shown in Figure 7. The lowest mean value 1.95E-07 was at dimension 2 and the highest mean value 0.002678 was at dimension 8.

The tabular values and the graphs presented illustrate that the lower the dimension, the better the performance of the algorithm for a fixed number of iterations. This is a well-known behavior of optimization algorithms in general.

### 4.2.3. Test Case 3: Varying Number of Iterations = 100, 1000, 10000, 100000

Five functions were evaluated by the krill herd algorithm on a constant number of dimensions, by varying the number of iterations. Twenty five independent runs for each function have been made, the best, average and standard deviation over the 25 runs are calculated and the results have been tabulated in Table 11 to Table 13.

Effect of varying Iterations on the Best Fitness: (Krill Size 50, Dimension 4)

**Table 11. Varying number of iterations and calculating best fitness value**

| **Function Name** | **Iterations = 100** | **Iterations = 1000** | **Iterations = 10000** | **Iterations = 100000** |
|---|---|---|---|---|
| Ackley | 2.9764E-05 | 1.01454E-05 | 7.4485E-06 | 4.4229E-06 |
| Alpine | 2.49887E-09 | 7.79325E-10 | 4.8406E-11 | 1.2738E-12 |
| Griewank | 0.001174387 | 0.008668097 | 0.00795148 | 0.00129299 |
| Rastrigin | 0.000813417 | 0.000992133 | 0.00094699 | 0.00014526 |
| Sphere | 1.01592E-05 | 2.82841E-07 | 4.4668E-08 | 2.697E-08 |

As the number of iterations got increased, the Best Fitness value got improved except for the Griewank and Rastrigin Functions.

Effect of varying Iterations on the Standard Dev: (Krill Size 50, Dimension 4)

**Table 12. Varying number of iterations and calculating standard deviation**

| **Function Name** | **Iterations = 100** | **Iterations = 1000** | **Iterations = 10000** | **Iterations = 100000** |
|---|---|---|---|---|
| Ackley | 0.0111448 | 0.00537887 | 0.003084698 | 0.001557194 |
| Alpine | 1.909E-05 | 1.9717E-06 | 2.36943E-07 | 1.97986E-08 |
| Griewank | 2.6525683 | 5.04392968 | 2.568646693 | 3.474356292 |
| Rastrigin | 1.3352213 | 3.21568529 | 5.922939286 | 2.212308028 |
| Sphere | 0.0144992 | 0.00022608 | 6.2379E-05 | 1.28587E-05 |

As the number of iterations got increased, the Standard Deviation got improved except for the Griewank and Rastrigin Functions.

Effect of varying Iterations on the Average Fitness Value: (Krill Size 50, Dimension 4)

**Table 13. Varying number of iterations and calculating average value**

| **Function Name** | **Average Fitness** | | | |
|---|---|---|---|---|
| | **Iterations =100** | **Iterations =1,000** | **Iterations =10,000** | **Iterations =100,000** |
| Ackley | 0.0023218 | 0.001120599 | 0.000642645 | 0.00032442 |
| Alpine | 3.977E-06 | 4.10781E-07 | 4.93632E-08 | 4.1247E-09 |
| Griewank | 0.9867384 | 1.095340352 | 0.661511676 | 0.504567335 |
| Rastrigin | 0.8826019 | 0.405636546 | 0.354945234 | 0.151437425 |
| Sphere | 0.0030207 | 4.70998E-05 | 1.29956E-05 | 2.6789E-06 |

The results tabulated have been plotted and have shown the following behavior:

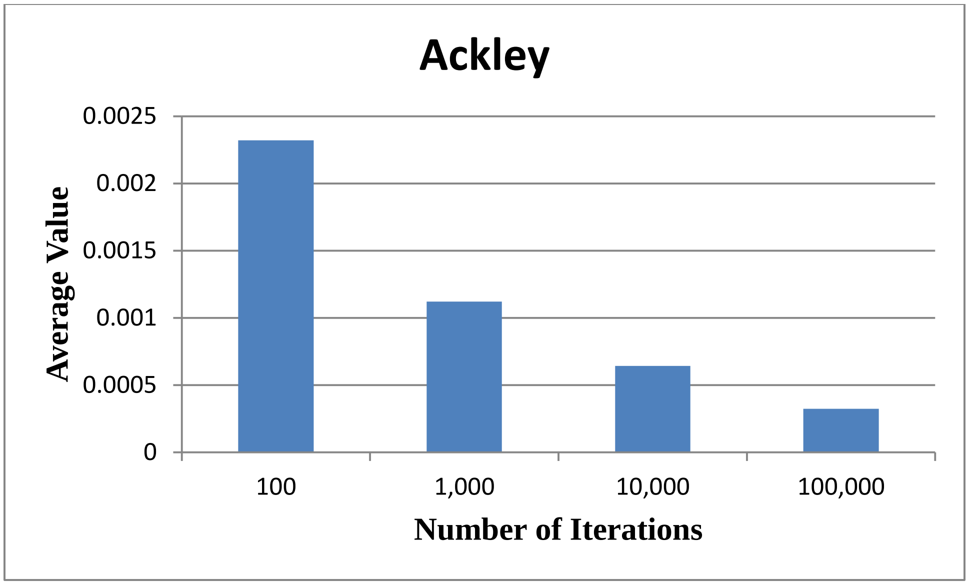

**Figure 8. Variation of average value for Ackley against increasing iterations**

The variation of average value versus the number of iterations for Ackley function is illustrated in Figure 8. The average value has varied between 0.0023 and 0.00032.

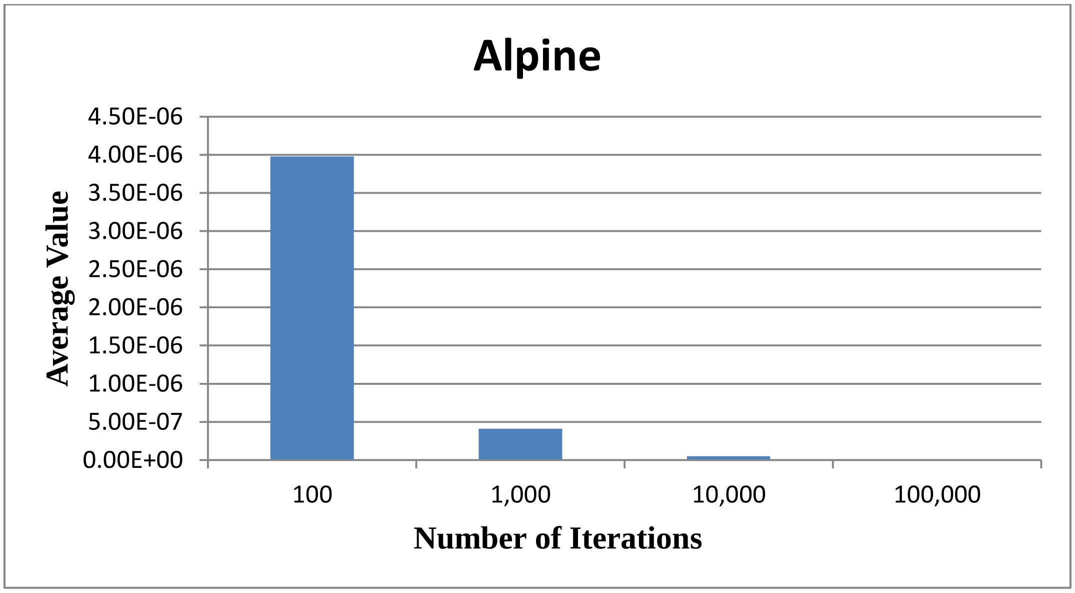

**Figure 9. Variation of average value for Alpine against increasing iterations**

The variation of average value versus the number of iterations for Alpine function is illustrated in Figure 9. The average value has been calculated at 4 different points between 100 and 100,000. The average value has varied between 3.98E-06 and 4.12E-09. The average value almost remained constant from 10,000 to 100,000 iterations.

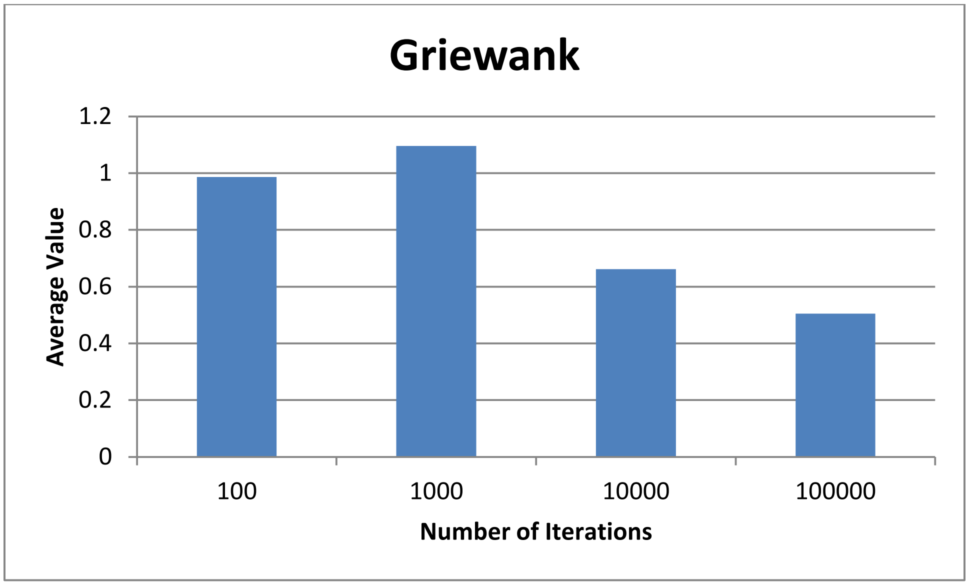

**Figure 10. Variation of average value for Griewank against increasing iterations**

The variation of average value versus the number of iterations for Griewank function is illustrated in Figure 10. The average value has been calculated at 4 different points between 100 and 100,000. The average value has varied between 0.9867 and 0.5046.

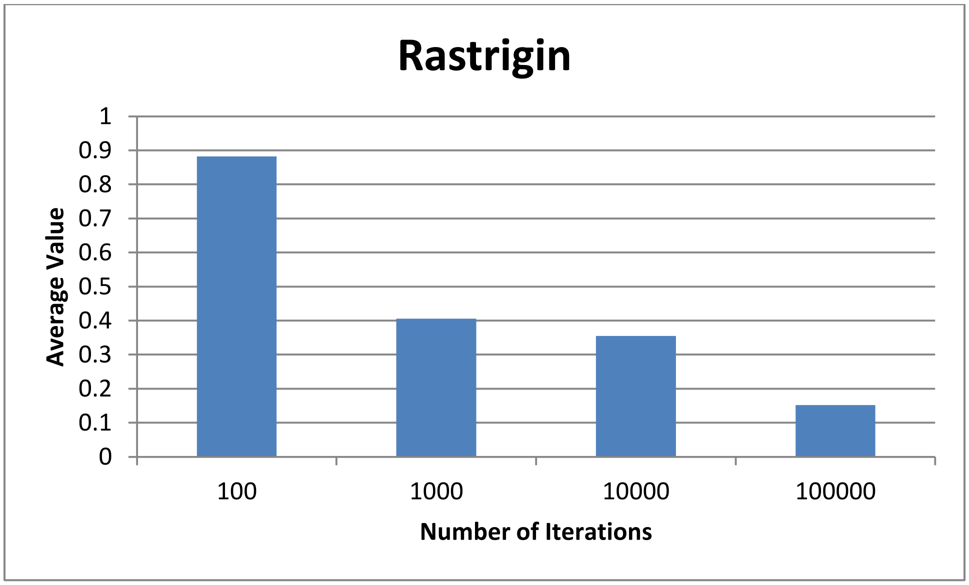

**Figure 11. Variation of average value for Rastrigin against increasing iterations**

The variation of average value versus the number of iterations for Rastrigin function is illustrated in Figure 11. The average value has been calculated at 4 different points between 100 and 100,000. The average value has varied between 0.8826 and 0.151.

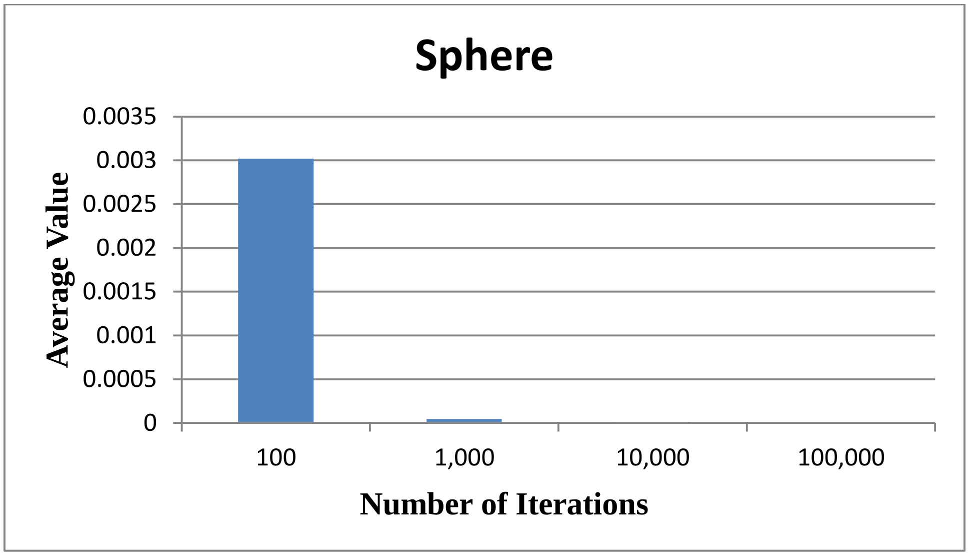

**Figure 12. Variation of average value for Sphere against increasing iterations**

The variation of average value versus the number of iterations for Sphere function is illustrated in Figure 12. The average value has been calculated at 4 different points between 100 and 100,000. The average value has varied between 0.003 and 2.68E-06. The average value almost remained constant from 1,000 to 100,000 iterations.

There is an improvement in performance of the algorithm with the increase in the number of iterations for all of the benchmark functions except Griewank and Rastrigin. Also, the increased performance is larger in some functions than in other functions.

### 4.2.4. Test Case 4: Plot Graph Number of Iterations = 50

The following section describes the results obtained by considering the number of iterations equal to 50 and recording the best value for one single independent run with the dimension=4. The simulations have been performed on five different functions and the results obtained are illustrated in Figure 13.

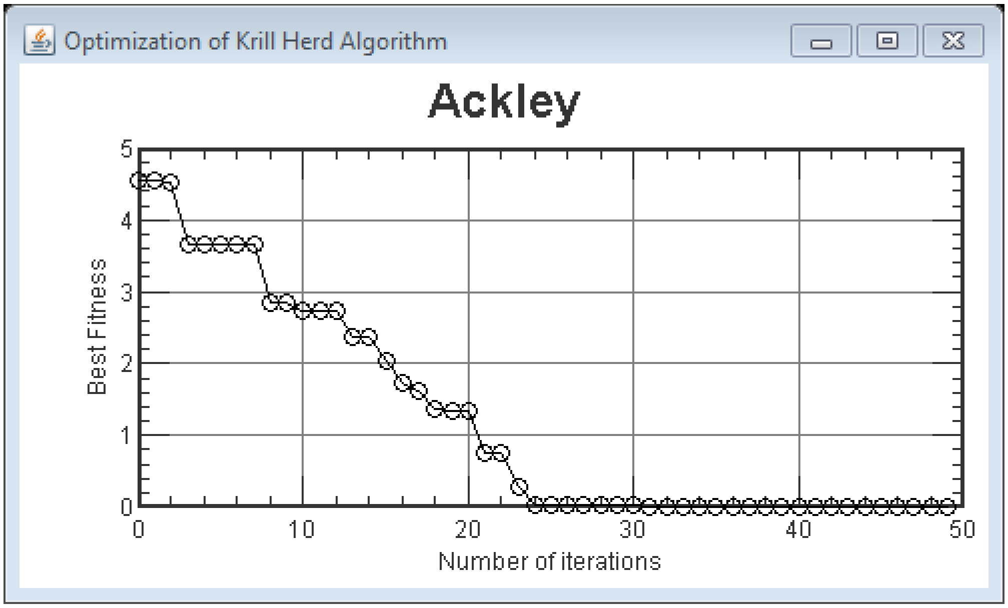

**Figure 13. Variation of best value with number of iterations for Ackley**

The variation of best objective value versus number of iterations for Ackley function is illustrated in Figure 13. The number of iterations has been varied between 0 and 50. The best objective value obtained is 0.0030515.

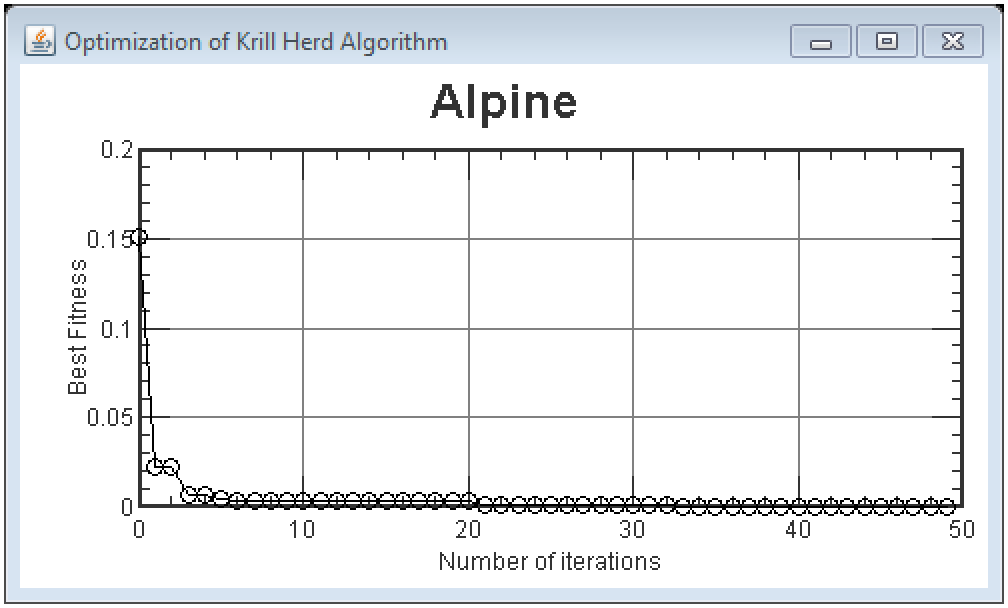

**Figure 14. Variation of best value against number of iterations for Alpine**

The variation of best objective value versus number of iterations for Alpine function is illustrated in Figure 14. The number of iterations has been varied between 0 and 50. The best objective value obtained is 7.93E-05.

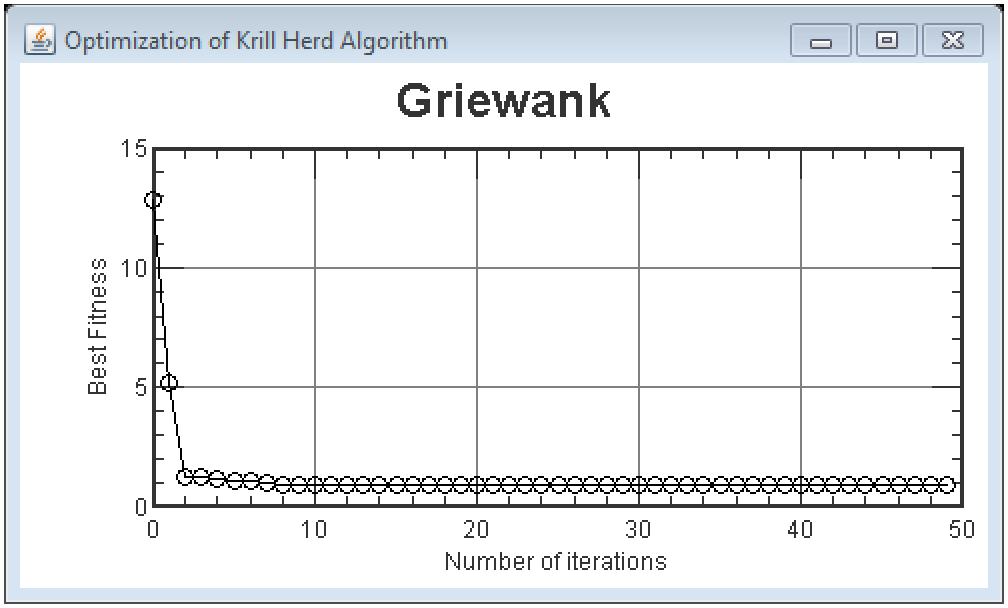

**Figure 15. Variation of best value with number of iterations for Griewank**

The variation of best objective value versus number of iterations for Griewank function is illustrated in Figure 15. The number of iterations has been varied in between 0 and 50. The best objective value obtained is 0.951116.

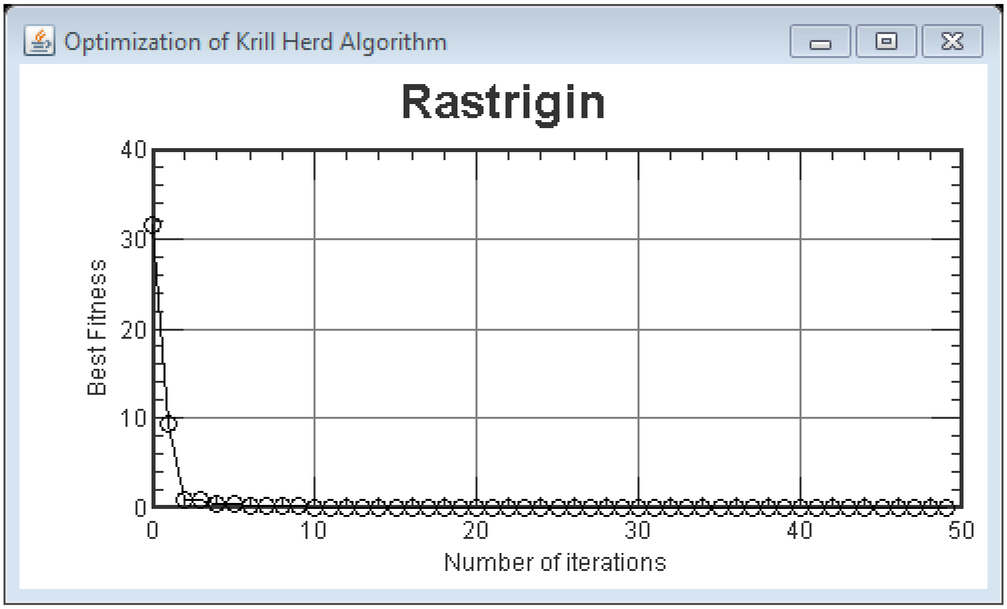

**Figure 16. Variation of best value with number of iterations for Rastrigin**

The variation of best objective value versus number of iterations for Rastrigin function is illustrated in Figure 16. The number of iterations has been varied in between 0 and 50. The best objective obtained is 0.095465.

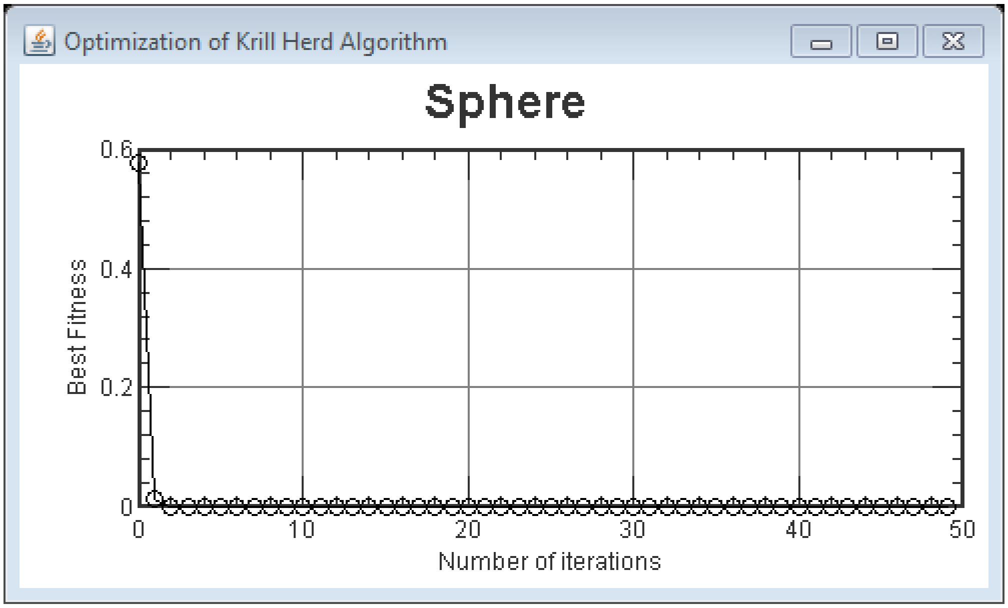

**Figure 17. Variation of best value with number of iterations for Sphere**

The variation of best objective value versus number of iterations for the Sphere function is illustrated in Figure 17. The number of iterations has been varied between 0 and 50. The best objective value obtained is 7.09E-04.

The plots clearly show that the best value decreases, that is, the performance of the algorithm increases with increasing number of iterations.

From the experiments performed and the results obtained, it is observed that the Krill Herd Algorithm is very efficient in finding the global minima under the influence of all the factors such as Neighbors, Food, Random diffusion, in all the possible cases.

## Comparisons:

The Krill Herd algorithm is compared with the basic PSO [27]. Both algorithms are evaluated over 25 simulations for 1 million iterations, and two dimensions. The average values are recorded and are tabulated for comparison as shown in

Table 14.

**Table 14. Average Fitness value comparisons made for 1 million iterations and 2 dimension**

| | Krill Herd | Basic PSO |
|---|---|---|
| Alpine | 3.49E-09 | 0.0 |
| Rastrigin | 0.0172 | 0.0254 |
| Sphere | 2.59E-09 | 9.1E-15 |

From the table, it is observed that the performance of the Krill Herd algorithm is better when compared to the PSO algorithm, when evaluated with the Rastrigin benchmark function. The PSO algorithm outperformed the Krill Herd Algorithm when evaluated with the Sphere and Alpine benchmark functions. However, this is a small set of benchmark functions to compare both algorithms with, and therefore, more benchmark functions should be tested for a meaningful comparison.

# 5. CONCLUSIONS AND FUTURE WORK

The Krill Herd Algorithm uses the krill herding mechanism in order to solve the optimization problems. The algorithm optimizes the objective function in an efficient way. This makes the algorithm useful in many areas and is extended to solve real world optimization problems.

The algorithm mainly considers the three essential activities, namely (i) Motion induced by other krill individuals, (ii) Foraging Activity, and (iii) Random Diffusion process. These factors determine the time dependent position of a krill individual. The first two activities contain both a global and a local optimizer that is having a local effect due to the presence of neighbor krills, and the global effect due to the best krill individual. Similarly in the case of the foraging activity, the local effect is due to the food attraction, and the global effect is due to the best position attraction of a krill individual found so far. Also, the third action, random movement is added to the krill herd methodology.

The Krill herd algorithm simulations are done carefully by considering the coefficients obtained from real world empirical studies done on the krill and its herding mechanism. Hence, the algorithm requires only the parameter $C_t$ (time interval) to be fine-tuned. Using the real world coefficients and its requirement of less number of parameters tuning, makes the Krill herd algorithm a good algorithm when compared to the other nature-inspired algorithms.

There are many advantages of the Lagrangian model as it considers the effect of each and every krill individual’s fitness value for the moving process in the search space. Also, the neighbors influence the movement of the krill individuals and the krills always try to maintain a high density in the herd. Finally, finding the center of food can also be considered as obtaining the global best fitness value of the krill herd algorithm. The paper mainly discusses about the proposed krill herd algorithm and its methodology.

In this paper, the Krill Herd algorithm was implemented using Eclipse environment in Java language with five different benchmark functions. Various test case conditions are evaluated and tested against the Krill herd algorithm to validate its method.

The test case 1, with the fixed number of iteration provides the best, average and standard deviation values. In test case 2, with the varying dimension, the function objective values are calculated and plotted. It was observed as the dimension increases, the performance of the algorithm decreases. In the test case 3 and 4, with varying number of iterations for 25 independent runs and one independent run, the function objective values are calculated and plotted. And as expected, the performance of the algorithm increased with the increase in number of iterations. The results from the experiments performed show that the algorithm efficiently solves the optimization problems.

In the future, we plan to incorporate the crossover and mutation genetic operators to make the existing krill herd algorithm more effective. Also, the algorithm can be tested by changing various KH parameters like $\Delta t$, $C_t$, etc. Additionally, the algorithm can be applied to various real world optimization problems.

# 6. REFERENCES

[1] Yang X-S, "Nature inspired metaheuristic algorithm," *Luniver Press,* 2008.

[2] Gandomi AH, Yang X-S, and Alavi AH, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems", Engineering with Computers in press.

[3] Talbi EG, "Metaheuristics: from design to implementation," Hoboken, New Jersey, 2009.

[4] J. Iztok Fister, X.-S. Yang, B. Janez and F. Dusan, "A brief review of nature inspired algorithms for optimization".

[5] Eberhart, J. Kennedy and Russell, "Particle swarm optimization," *IEEE International Conference,* vol. 4, pp. 1942-1948, 1995.

[6] S. Kirkpatrick, J. Gelatt.D and M. P. Vecchi, "Optimization by simulated annealing," vol. 220, no. 4598, pp. 671-680, 1983.

[7] F. Iztok, J. Iztok Fister, Y. Xin-She and B. Janez, "A comprehensive review of firefly algorithms," 2013.

[8] Dorigo M, Maniezzo V and Colorni A, "The ant system: optimization by a colony of cooperating agents," *IEEE Trans Syst Man Cybern B,* vol. 26, no. 1, pp. 29-41, 1996.

[9] Yang X-S and Gandomi AH, "Bat algorithm: a novel approach for global engineering optimization," *Eng Comput,* vol. 29, no. 5, 2010.

[10] Yang X-S, "Engineering optimizations via nature-inspired virtual bee algorithms," vol. 3562, pp. 317-323, 2005.

[11] Goldberg DE, "Genetic algorithms in search, optimization and machine learning," 1989.

[12] Karamanoglu M, He X-S, Yang X-S, "Multi-objective flower algorithm for optimization,"

vol. 18, pp. 861-868, 2013.

[13] "Evolutionary Algorithms," [Online]. Available: http://www.geatbox.com/docu/algindex-01.html.

[14] Khatib W and Fleming P, *The stud GA: A mini revolution?,* New York, Springer, 1988.

[15] Koza JR, "Genetic programming: on the programming of computers by natural selection," *MIT Press,* 1992.

[16] A. Alavi and A. Gandomi, "Multi-stage genetic programming: a new strategy to nonlinear system modeling," *Inf Sci,* vol. 181, no. 23, pp. 5227-39, 2011.

[17] Simon D, "Biogeography-based optimization," no. 12, p. 702–13, 2008.

[18] Passino KM., "Biomimicry of bacterial foraging for distributed optimization and control," no. 22, pp. 53-67, 2002.

[19] Gregory R, Paton RC, Saunders JR and Wu QH, "Parallelising a model of bacterial interaction and evolution," vol. 76, pp. 121-31, 2004.

[20] Vlachos C, Paton RC, Saunders JR and Wu QH, "A rule-based approach to the modelling of bacterial ecosystems," vol. 84, pp. 49-72, 2005.

[21] Gandomi AH and Alavi AH, "Krill Herd: A new bio inspired optimization algorithm," vol. 17, no. 12, p. 4831–4845, 2012.

[22] Marr JWS, "The natural history and geography of the Antarctic krill (Euphausia superba Dana).," no. 32, pp. 33-464, 1962.

[23] Hofmann EE, Haskell AGE, Klinck JM and Lascara CM, "Lagrangian modelling studies of Antarctic krill (Euphasia superba) swarm formation," no. 61, pp. 617-31, 2004.

[24] J. Momin and Yang X-S, "A literature survey of benchmark functions for global

optimization problems," 2013.

[25] "Benchmark Problems," [Online]. Available: http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume24/ortizboyer05a-html/node6.html.

[26] "Wiki-Rastrigin," [Online]. Available: http://en.wikipedia.org/wiki/Rastrigin_function.

[27] S.Daggubati. [Online]. Available: http://library.ndsu.edu/repository/handle/10365/22285.