

A Review on Nature-based Swarm Intelligence Optimization Techniques and its Current Research Directions

K. Kanaka Vardhini¹* and T. Sitamahalakshmi²

¹Department of CSE, ASCET, Gudur – 524101, Andhra Pradesh, India;
kanakavardhini@gmail.com

²Department of CSE, GITAM University, Visakhapatnam – 530045, Andhra Pradesh, India;
tsm@gitam.edu

Abstract

Background /Objectives: In today's world, finding a feasible solution for combinatorial problems becoming a crucial task. The main objective of this paper is to analyze and comprehend different nature based algorithms enabling to find optimal solution. **Methods/statistical analysis:** Bacterial Foraging Algorithm (BFOA), firefly algorithm, Ant Colony Optimization (ACO), bee colony optimization, cuckoo optimization etc. Which have been used in power load balancing, cost estimating, optimal routing, color segmentation were discussed. This paper also highlights the constraints and convergence properties of each algorithm to solve certain problems encountered in various fields of application. **Findings:** Ant colony algorithms were successful in finding solutions within 1% of known optimal solutions. Optimal solution was found in BFOA by adjusting chemo taxis step size. Also, this paper analyzes results of various research works done in numerous fields using the swarm intelligence techniques.

Keywords: Combinatorial Problems, Optimization, Swarm Intelligence

1. Introduction

Optimization is the process of finding the best inputs in obtaining the maximum or minimum output with minimum cost¹. The procedure for optimization problem begins by choosing the design variable, and by formulating the constraints and objective functions. The main goal of objective function is to set a value to chosen variables according to given constraint which will produce the optimum response. These optimization problems can be classified as continuous and combinatorial optimization problems depending on number of available solutions for a given problem. The continuous problems have infinite solutions whereas combinatorial have finite number of solutions. In case of combinatorial, to find the best from available finite solution different optimization paradigms are to be utilized in handling various computation problems.

In this paper the authors mainly review the different nature based optimization techniques and their current research direction. In the section-2, the authors have given an overview on basic concept of optimization technique. Section-3 gives an overall view of the existing swarm intelligence based optimization techniques.

2. Basic Concepts of Optimization Technique

Optimization techniques find the best or optimum solutions from the feasible solutions for a given problem. There are many types of optimization techniques available to solve single, multi variable functions with or without constraints. The optimization techniques can be developed through many methods like linear, non-linear programming, simulated annealing which is inspired from annealing process of metallurgy, genetic algorithms

* Author for correspondence

which finds the best local solution from given population or through swarm intelligence. Optimum solution can be achieved either through traditional or non-traditional methods.

Differential calculus techniques¹ used by the traditional methods in finding out the optimal solution of a given problems. In general the traditional methods are classified as direct or gradient methods based on the search technique it used to find best one. The direct methods use value of objective function whereas a gradient method uses derivative function information to guide the search. These traditional methods are more suitable for uni modal, simple objective function. But these methods are not effective for multimodal, complex problems.

In order to get the efficient result in above discussed case nontraditional/Heuristic based techniques are developed. These non-traditional methods can be classified as evolutionary, nature and logical based techniques.

Evolutionary computation is a computational paradigm inspired by Darwinian evolution. An artificial evolutionary system is mainly based on a computational system with the notion of simulated evolution. This type of evolutionary system consists of a constant - or variable-size population of individuals, a fitness criterion, and genetically inspired operators which develop the next generation from the present system. Examples are like genetic programming, evolutionary programming etc.

The nature based methods or swarm intelligence techniques are developed based on collective behavior of individual agents which can communicate directly or indirectly to perform the complex task in nature. It addresses the study of the collective behaviors of systems made by many components that coordinate using decentralized controls and self-organization process as they are exhibited in the working culture of social creatures like flocks of birds, ants, fishes etc. where individuals are simple, homogeneous and autonomous in nature.

By combining mathematical programming and logic programming logical based approach is developed². The mathematical programming was used for extracting the structure whereas logical programming was used to extract the information to find the optimum solutions for a given problem. Tabu search cross optimization methods are the best example for logical based optimization techniques. The classification³ of optimization techniques depends on above discussed methods are shown in the following Figure 1.

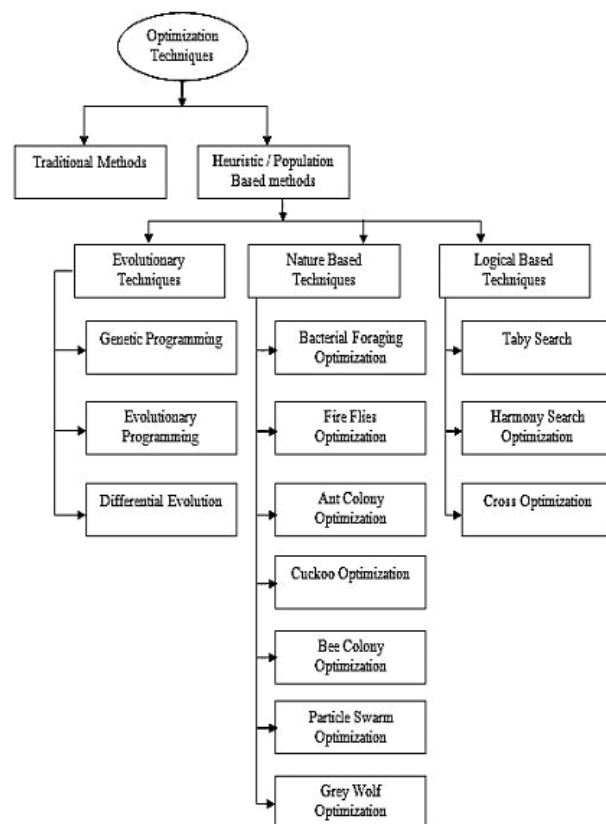


Figure 1. Classification of optimization techniques

3. Study on Nature based Optimization Techniques and its Applications

Swarm intelligence³ is a discipline of self-organization in which structure appears at global level from the coordination of individual agents'. It is a contemporary artificial intelligence discipline that primarily focuses on the design of multi-agent systems with applications, such as optimization, robotics etc. It addresses the study of the collective behaviors of systems made by many components that coordinate using decentralized controls and self-organization process as in the working culture of social creatures. There are many swarm intelligence based optimization evolutionary algorithms which are inspired from social insect's behavior like ant stigmergy, birds flocking, fish schooling and bees waggle behavior. In this paper, the authors discussed some of the swarm intelligence optimization techniques, which are more effective for solving linear/nonlinear problem compared to other techniques.

3.1 Bacterial Foraging Algorithm

The bacterial foraging optimization algorithm⁴ is a computational intelligence and Meta heuristics type. It was first proposed by Passino⁵. Like another swarm algorithms such as artificial bee colony and firefly algorithm, it is also applied to get solutions for many engineering problems. The inspiration for this algorithm is taken from chemotaxis behavior of bacteria *E. coli* and *MX anthus*.

According to the gradient of chemicals in the environment, bacteria seek direction for food. Bacteria can tumble or swim according to its flagella⁵. Using its swimming behavior bacteria moves in different directions in order to get food. These Bacteria cells are treated like agents. Depending on the agent to agent interactions, agents may form a group or it may ignore each other.

```

Input:  $P_s, C_{no}, N_{ed}, N_{rp}, N_{ch}, N_{sc}, RDVs, Dattr, Wattr,$ 
 $H_{rep}, W_{rep}, PR_{ced}$ 
Output:  $C_{best}$ 
POP  $\leftarrow$  Initialize population (Cno, Ps)
For (a=0 to  $N_{ed}$ )
For (b=0 to  $N_{rp}$ )
For(c=0 to Nch)
Chemo taxis and Swim (pop, Ps, Cno, Nsc,
RDVs, Dattr, Wattr, Hrep, Wrep)
For( $C \in$  pop)
If ( $cost(C) \leq cost(C_{best})$ )
 $C_{best} \leftarrow C$ 
End
Sort by cell health (Pop) Sel $\leftarrow$  Select by cell
health (Pop, Cs/2)Pop $\leftarrow$ Sel
End
For( $C \in$  pop)
If ( $rand \leq PR_{ced}$ )
 $C \leftarrow$  Create Cell At Random Location ( )
End
End
End
Return ( $C_{best}$ )

```

Pseudo-code 1. pseudo-code of Bacterial foraging algorithm.techniques

The optimization can be achieved through three populated simulated cells 1. Chemo taxis 2. Reproduction 3. Elimination-dispersal. By taking this as an inspiration, Passino simulated the following algorithm as artificial bacterial foraging to find its best solution.

The parameters of the algorithm are as follows Cno represents the total cells of the population, Ned is the number of elimination-dispersal steps, Nrp is the number of reproduction steps, Nch is the number of chemo taxis steps, Nsc is the swim steps for a given cell, RDV is vector with the same size of problem space and Ped is the probability to decide chance of cell elimination and dispersal., Dattr, Wattr are the attraction coefficients. The pseudo code for bacterial foraging behavior is discussed with pseudo-code 1

3.1.1 Research Scenario

A general and strong model for classification and identification of diseases in a person using BFOA proposed by Bensujin B et al.⁶, The proposed system finds the best position with a minimum chemotactic step and reproduction stages. The result show improvement in accuracy of ST segment elevation myocardial infraction prediction.

Tan et al. proposed an Adaptive Comprehensive Learning Bacterial Foraging Optimization (ACLBFO) to solve vehicle routing problem with time windows⁷. They used non-linearly decreasing model to maintain the balance between exploitation and exploration of ACLBFO to realize that it gives better performance in solving multi model problems.

Kim and Abraham⁸ used bacterial foraging with genetic algorithm to tune a PID controller of an Automatic Voltage Regulator (AVR). Initially they studied the performance of proposed algorithm with mutation, crossover, variation of step sizes, chemotactic steps as significant aspects.

Naveen⁹ et al. applied modified bacterial foraging optimization algorithm on Distribution system reconfiguration for loss minimization .The modified algorithm tested on 16 bus, 33 bus and 69 bus test systems to prove the efficiency of modified algorithm and they proved that it gave better performance than existing methodology.

B. K. Panigrahi, V. R. Pandi¹⁰ used bacterial foraging algorithm for multi objective optimization. Chemo taxis process and Fuzzy dominance based sorting procedure are used to find most recent bacterial locations, Pareto Optimal Front (POF). The researcher studied the algorithm on economic dispatch to test the applicability of the algorithm.

Several adaption techniques are applied by researchers to improve BFA in solving many engineering problems

such as economic emission load dispatch¹¹, load frequency control¹², portfolio optimization¹³, optimal power flow¹⁴, automated experimental control drives¹⁵ and color segmentation¹⁶

3.2 Firefly Algorithm

In today's world, more promising swarm optimization technique is Firefly algorithm¹⁷, which was introduced by yang in 2008. It uses a randomization by searching for a set of solution so it belongs to a stochastic algorithm. In this algorithm, he incorporated the nature of real fireflies. The fireflies¹⁷ are one type of insects, which produces flash light with different color. Each firefly produces flash light with different intensity. They use flashing for communication, and to attract other flies for mating. By taking this as a inspiration Xin-She Yang¹⁶ developed an artificial Firefly Algorithm (FA). He developed the FA based on attractiveness between fireflies and negative correlation between the distance and brightness of fireflies flash. The brightness can be calculated using landscape of objective function. The main advantages of fireflies algorithm are its subdivision of population automatically and finding the best global solutions when the size of subdivision population has larger size compare to number of modes. Classical firefly algorithm is suitable for multimodal optimization problems. The above discussed procedure of FA was explained with the Figure 2

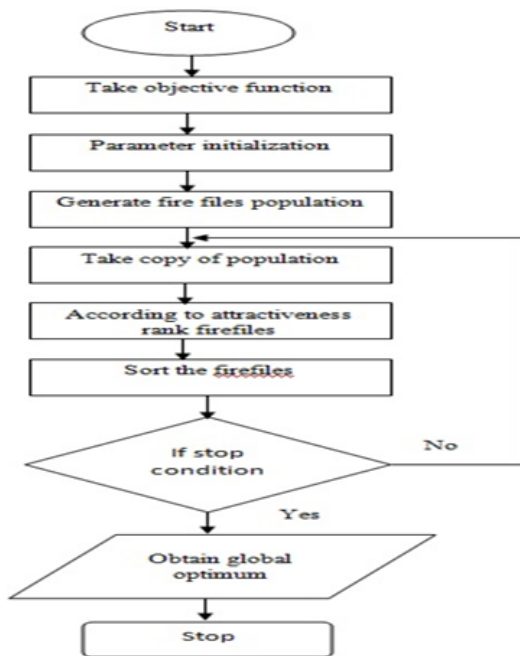


Figure 2. flow chart for Firefly algorithm.

The algorithm is population based algorithm and it has mainly four steps¹⁸ namely initialize the swarm, calculate the light intensity, calculate the attractiveness and generation of new swarm. The procedure for above four steps is discussed with the following pseudo-code 2. Start

```

Set the goal function  $g(Y)$ ,  $Y=(y_1, y_2, \dots, y_n)$ 
Generate initial population  $Y_k$  ( $k=1, 2, \dots, d$ )
Define the light intensity (LI) for  $y_k$  by calculating  $g(y_k)$ 
Define absorption coefficient  $\mu$ 
While ( $k < MG$ )
Make a copy of the generated firefly population for move function
For  $a=1$  to  $n$ 
For  $b=1$  to  $n$ 
If ( $LI_b > LI_a$ )
 $FF(b) \leftarrow FF(a)$  according to attractiveness
End if
Attractiveness changes through  $\exp(\mu, r)$ 
Updated LI and evaluate new solutions
End
End
Find the best FF
End
Show the result after pre-processing
End
    
```

Pseudo-code 2. Pseudo-code of Firefly optimization techniques

Many variants (binary, real-valued, non-elitism, elitism, uniform, chaos distribution) of existence firefly algorithms e depend on their strategy parameter settings. The settings¹⁹ reflects the performance of the algorithm. The Figure 2 explains each step of firefly in brief manner

3.2.1 Research Scenario

The research scenario of firefly optimization is discussed below

K. C. Udaiya kumar et al. used firefly algorithm on job scheduling problem²⁰ to find the single objective of JSSP (i.e. make span minimization using 1-25 Lawrence problems as a bench marking from a classical OR- library).

Sayadia et al. used Firefly algorithm for solving discrete optimization and cell formation²¹. They also proved that FA discrete version solve the NP hard problem more

effective compare to PSO.

Yafei Tian , Weiming Gao and Shi Yan proposed inertia weight firefly to avoid repeated oscillation on the position of local or global extreme value point²², due to optimizing accuracy is reduced.

Xin-She Yang used Firefly for multimodal optimization problem²³ and he proved particle swarm optimization (PSO) is one type of firefly algorithms.

Senthil et al. used firefly algorithm for clustering²⁴. They proved the performance of FA by comparing with eleven different algorithms of clustering. They also used Firefly to train neural network to get effective results.

3.3 Ant Colony Optimization

Ant colony optimization is introduced²⁰ by Dorigo et al. He takes the inspiration from real ants' stigmergy behavior and targets on discrete and continuous optimization problems. The real ants find the shortest path between its food and nest without having direct communication between them. Initially ants choose random paths from their nest to food. During their traveling, it leaves pheromone (one type of chemical) along the path. Randomly the ants which choose the shortest path will travel on the same path more times comparing to ants which travels on the longest path. For time being the pheromone will be evaporated. Because of this, the shortest path has highest pheromone concentration which attracts remaining ants which is shown in following Figure 3. Based on this behavior Marco Dorigo and Thomas stutzle developed an artificial Ant Colony Optimization (ACO) metaheuristic. The ACO is probabilistic technique which can be found the best path through graphs. Each path followed by an ant shows candidate solutions for the given problem. The amount of pheromone is increased depends on quality of candidate solution. The remaining ants choose a path that depends on pheromone amount on the available paths. Due to this, the shortest path has more pheromone amount on its and it becomes the best path among all the paths. The following pseudo code 3 explained the above concept.

The above ant colony pseudo code is initially proposed by Marco Dorigo, Thomas stutzle in the year 1991. Later on some modification is made on the algorithm to solve in different problems in an effective way. Table 1 list out the people which worked on ACO with small modifications in chronological order²⁵.



Figure 3. Path selection of ants.

```

Input: Colony size( $C_s$ ), number of ants( $NU_{ant}$ ),  $n$ ,
 $\beta$ ,  $\sigma$ ,  $\rho$ ,  $Q_0$ 
Output: Best path ( $P_{best}$ )
 $P_{best}$  Create Heuristic solution ( $C_s$ )
Pheromone Initialize pheromone
While (Stop condition)
For ( $j=1$  to  $n$ )
 $Sol_i$  construct solution (pheromone,  $C_s$ ,  $\beta$ ,  $Q_0$ )
 $Sol_i$  cost ( $Sol_i$ )
If ( $soli_{cost} \leq Pbest_{cost}$ )
 $Pbest_{cost} = soli_{cost}$ 
 $P_{best} = Sol_i$ 
End If
Local Update and decay pheromone
End
Global update and decay pheromone
End While Return ( $P_{best}$ )

```

Pseudo-code 3. Pseudo-code of Ant colony optimization.

Table 1. Ant miner versions

Algorithm	Authors	Year
Ant system (as) 26	Dorigo et al.	1991
Elitist as 27	Dorigo et al	1992
Ant-q28	Gambardella and dorigo	1995
Ant colony system 29-30	Dorigo and gambardella	1996
Max-min as 31-32	Stutzle and hoos	1996
Rank-based as 33-34	Bullnheimer et al	1997
Ants35	Maniezzo	1999
Bwas36	Cordon et al	2000
Hyper-cube as37	Blum et al	2001

Initially the Dorigo et al. simulated the real ant behavior as an ant system and applied this for solving many NP-hard problem like travelling sales man ,job shop scheduling. The ACO has only global level pheromone optimization which is reducing the efficiency of solution. To avoid this, they made some changes on pheromone updating and called it as ELITIST AS in the year 1992.

In the year 1995 the Gambardella and Dorigo finds antecedent of ant system as ANT-Q. The inspiration is taken from Q-learning algorithm and Ant systems. Ants have memory and they exchange the information between

them to get optimum solutions in this technique.

Later stutzle and Hoos get best solution by keeping the range for pheromone updating and called it as a mas-min ant colony optimization.

3.3.1 Applications of ACO

The ant colony optimization is graph based technique which was used to get the optimum solutions in various applications. Initially it was used to solve NP hard problems like traveling salesman, job shop scheduling problems etc.

Table 2. Applications of ACO

Problem type	Problem Name	Procedure	Author	Year
NP hard	Job shop Scheduling ³⁹	Elitist Ant system	Edson Flórez1 ,Lola Bautista	2013
Designing	Micro Wave Filter Design ³⁸	iterative procedure Of ACO with electromagnetic simulator	Ivan A. Mantilla-Gaviria Alejandro Díaz-Morcillo, Juan V. Balbastre-Tejedor	2013
Preprocessing	Feature selection ⁴⁰	ACOFs with hybrid Search technique	MonirulKabir, MdShahjahan, Kazuyuki Muras	2013
Clustering	Clustering ⁴¹	ACO with dynamic Pheromone updating	Xiaoyong Liu	2014
Optimization	Ship steering Control ⁴²	Ant colony algorithm	MirosławTomera	2014
Image processing	Face identification systems ⁴³	ACO+ genetic algorithm	S.Venkatesan et al.	2015
Routing	Vehicle routing Problem ⁴⁴	Max–Min Ant System	JiafuTanga et al.	2013
Security	Detection of black hole attack in MANET ⁴⁵	ACO	V.Anchugam et al.	2015
Cost estimation	S/W cost estimation ⁴⁶	ACO + COA	Zahra Asheghi Dizaji	2015
Routing	detecting optimal path for routing in WSN ⁴⁷	Parallel ACO	K. V. Janaki Raam	2015

Later on it was used to solve many engineering problems which are discussed in Table 2.

3.4 Cuckoo Optimization

Most of the birds care a lot until the development of young ones to the bird right from laying eggs. However, there are birds so as certain species, like cuckoo that depends on other birds for raising their young ones. This process of relying on other organism to raise their young ones is called brood parasitism.

Brood parasitism of three types namely Intra specific, Cooperative breeding and nest takeover. Few species of cuckoo like Ani and Guira lay eggs in communal nests

while others lay in nest of birds of other species. However, if the host bird identifies the presence of eggs which are not its own, it may directly raise a conflict with the cuckoo or leave the nest abandoned. However, a new species of cuckoo, namely Tapera has developed which is specialized in mimicry in colors and patterns of eggs of host birds, thus it reduces chance for their eggs to abandon. In addition to this, cuckoo even takes at most care so that all its young ones are properly raised by the host. Soon after expelling eggs of host bird the first cuckoo chick comes out. Thus it ensures greater share of food to the cuckoo chicks by host mother birds. The other most interesting feature of a cuckoo chick is that it can mimic

the call of host chicks. The brood parasitism of cuckoo which is used to find the best nest for his younger ones to be bring up as bird is explained thorough below Figure 4.

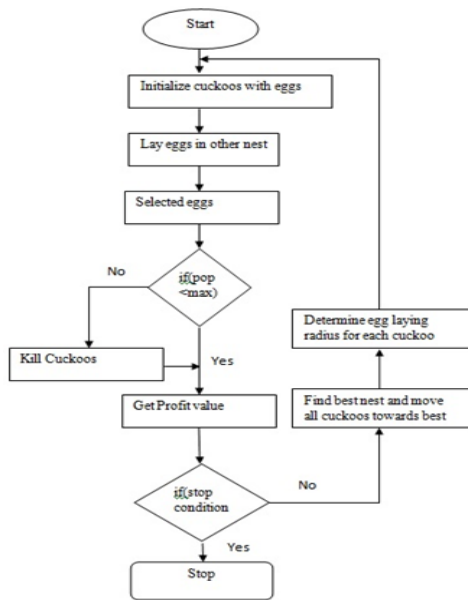


Figure 4. Cuckoo optimization.

Xin-She yang and SuashDeb⁴⁸ developed Cuckoo optimization in 2009 by taking Cuckoo brood parasitism as an inspiration. Initially they generate the population of cuckoo, after that select the best nest which depends on fitness value to lay the eggs. The nest with the worst fitness is abandoned and the remaining is ranked to find the optimum solution. The pseudo code of the Cuckoo optimization has discussed in pseudo-code 4.

```

Begin
Objective function  $O(Y)$ ,  $Y = (Y_1, \dots, Y_d)$  T
Generate initial population of cuckoos  $Y_i$  ( $i = 1, 2, \dots, n$ )
While (stop criterion)
Select a nest randomly
Calculate fitness  $F_m$ 
Select a nest among  $n$  (say  $n$ ) randomly
if ( $F_m > F_n$ ),
Replace  $n$  by the new solution;
End if
Worse nests are abandoned and new ones are built
Keep the best solutions (or nests with quality solutions)
Rank the solutions and find the current best
End while
Post process results and visualization
End
  
```

Pseudo-code 4. Pseudo-code of Cuckoo search.

3.4.1 Applications of Cuckoo Search

Thuan Thanh Nguyen, Anh Viet Truong⁴⁹ have proposed a cuckoo optimization based reconfiguration methodology for minimizing active power loss and the maximizing voltage magnitude. They have also proved, cuckoo optimization is efficient method for network reconfiguration problems.

Ehsan Valian⁵⁰ et al. proved that tuned cuckoo search parameters enhance the accuracy of cuckoo search comparing to fixed parameters.

Carlos Cobos et al.⁵¹ used cuckoo search in description-centric algorithm for the clustering of web results especially for global or local search, which is used to find the best nest.

A comparison⁵² made by Pinar Civicioglu, Erkan Besdok among Differential Evolution (DE), artificial bee colony algorithm, Particle Swarm Optimization (PSO) and cuckoo optimization algorithms. Their analysis revealed that cuckoo, DE is closer and better than PSO.

In⁵³ Ashok K. Bhateja et al used cuckoo search to find key of a vigenere cipher. They combined Levy flights behavior and Markov chain random walk with cuckoo search for obtaining the better global optimal solution at the time of generating key of vigenere cipher.

Xiangtao Li and Minghao Yin⁵⁴ introduced self-adaptive parameter algorithm as a uniform random value to enhance the diversity of the population. The proposed method uses two new mutation rules based on the rand and best individuals among the entire population. In order to balance the exploitation and exploration of the algorithm, the new rules are combined through a linear decreasing probability rule.

3.5 Bee Colony Optimization (BCO)

Bee Colony Optimization is a bottom up approach, which is introduced by Teodorovic and Lucic to solve combinatorial optimization problems. In nature, Bees are classified as employed, onlooker, scout bees. Initially employed bees starts search for food at the nearest places of their hive. According to the nectar amount, employed bee perform the waggle dance. According to the waggle dance, the onlooker selects the hive which has highest amount of nectar. It transfers the nectar/food to the hive by relinquish the food source to employee. Later employee bee searches for another food source. Collection and process of food/nectar by bees simulated as bee colony. The nature based Bee colony optimization

starts with initialization of parameters like number of bees (Solutions), number of iteration and stopping criteria. It has two passes namely forward, backward passes

```

Initialize  $B_n$ : Number of bees
NP: Number of passes
SC: Stopping criteria
Read data(sample data set)
Do
Assign b(n) solutions to each bee
For (k=0;k<NP;k++)
For (j=0;j< $B_n$ ;j++)
For (i=0;i<f(NP);i++)
Assess possible moves
Using roulette move select one move
For (m=0;m< $B_n$ ,m++)
Assess the solution of bee
For (m=0;m< $B_n$ ,m++)
Find loyalty decision of bee
For (m=0;m< $B_n$ ,m++)
If m is uncommitted choose a recruiter
Find best one other wise update  $X_b$ ,  $f(x_b)$ 
Return ( $X_b$ ,  $f(X_b)$ )

```

Pseudo-code 5. pseudo-code of bee colony optimization.

In forward pass, set of solutions are chosen from the available solutions randomly. The quality of solutions is found in backward pass. According to quality of selected solution, it can keep it as the best or abandon the solution. Iterations are continued till either of the best solution or stopping criteria could meet. The pseudo code 5 explains the procedure of BCO.

3.5.1 Research Scenario

R. J. Kuo et al.⁵⁵ proposed Automatic Kernel Clustering With Bee Colony Optimization (AKC-BCO) to perform cluster analysis on given dataset without knowing the number of clusters to be made in prior. The kernel function is used in this algorithm to perform automatic clustering which gives better and more stable results than existing ones.

M. Alzaqebaha, S. Abdullahb proposed a hybrid bee colony optimization⁵⁶ for examination timetabling problems. Different selection strategies, self-adaptive mechanism are used in selecting a neighborhood structure during the search process and local search algorithms are

hybridized to get the best solution.

Rajesh Kumar⁵⁷ proposed a Direct Bee Colony (DBC) optimization and tested on various optimization problems to prove the robustness of DBC algorithm. He also proved that DBC generates the unique solution on different runs, whereas other evolutionary algorithm produces different optimum solution during different runs. So DBC gives the better optimum solution for real time problems

Teodorovic⁵⁸ et al. illustrate the Bee colony optimization efficiency and characteristics with various examples. They proved that BCO is capable of solving combinatorial optimization problems which are characterized by its uncertainty. They also applied fuzzy based BCO to ride-sharing problems.

Mohammad Khalilzadeh⁵⁹ used bee optimization to minimize the total costs of both renewable and nonrenewable resource usage. Prioritization rule was used for activities along with small modifications in local search of bees which was increasing the accuracy of the process.

Bee Colony Optimization (BCO) metaheuristic approach used to optimize the parameters of COCOMO model for improving software cost estimation by Sherry Chalotra et al.⁶⁰. Interactive Voice Response software project dataset was used for validations and best solutions are selected using mean magnitude of relative error.

3.6 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique modelled on the social behaviors found in animals or insects which was first proposed by Kennedy and Russell Eberhart in 1995. In this optimization technique, individual *particles* of a swarm is represented as potential solutions, which budge through the problem search space for finding an optimal or the best solution for the problem. The particles transmit their current positions to neighboring particles. The mechanics used here is that the position of each particle is attuned according to its velocity (i.e., rate of change) and the difference between its current positions, correspondingly the best position found by its neighbors, and the best position it has found yet. With the iteration of the model, the swarm prioritized more of the search space yielding high-quality solutions.

3.6.1 Research Scenario

The researchers have been used particle swarm optimization to predict scour depths by obtaining

appropriate parameters for the neural network model and fuzzy inference system, where set of 188 scour depth data supplied by USGS were used for training and testing⁶¹. The proposed model generated lower RMSE and MSE values compared to existing ones.

H. Izadi et al⁶². has been proposed a new stabilizer system to increase the stability and synchronization of multi-machine power system through power system stabilizers 3 band (pss3b) by using particle swarm optimization. The results have been compared by applying three-phase fault at different working conditions with existing ones.

In⁶³, the researchers have been proposed a approach for resource allocation in a hybrid cloud. The local search problem of PSO was avoided with the combination of cuckoo search. The results of the proposed method increase the IaaS providers benefit compare to SLPSO.

In⁶⁴, an optimization model was proposed for detection and separation of brain tumor through Magnetic Resonance Imaging (MRI) medical images using Particle Swarm Optimization (PSO) with four stages. The approach proved that segmenting the MRI brain image by PSO algorithm is better for the coronal plane than the axial plane.

In⁶⁵, a weather forecasting approach based on Fuzzy Neural Network (FNN) and Hierarchical Particle Swarm Optimization (HPSO) has been presented. The proposed FNN-HPSO provided the most optimal forecasting results in comparison against other conventional techniques.

In⁶⁶, Particle swarm optimization has been used to minimize the production cost for designing the concrete gravity Dam. PSO values apply a function instead of a number for measuring bottom width which will reduce the dam cost compare to existing ones.

In⁶⁷, Comprehensive Learning Particle Swarm Optimization (CLPSO) algorithm was presented for multi objective optimal power flow. IEEE 30-bus 6-generator system was used to show the potential and efficiency of algorithm in terms of the diversity and quality of the obtained optimal solutions.

Reza Effatnejad et al. proposed a hybrid method by combining dynamic programming (DP), genetic algorithm and particle swarm optimization to solve the unit commitment of power systems by considering ring system security constraints⁶⁸. They avoided repetitive calculations by using DP in solving SCUC. PSO was used to handle binary and real world variable and local optima

avoided with the help of GA. The proposed method showed optimal response in solving problem.

3.7 Grey Wolf Optimization (GWO)

By taking the inspiration from hunting mechanism of grey wolves, Mirjalili et al. proposed a Grey Wolf Optimizer (GWO) in 2014⁶⁹. By nature, grey wolves are placed at the top of the food chain. They live in a group. Each group contains 5-12 members. These members follow the social dominant hierarchy which was shows in following Figure 5.

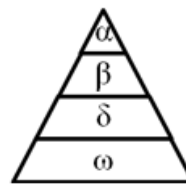


Figure 5. Dominant Hierarchy of grey wolf

The top level of wolf can be called as alpha wolf which can act like a leader. It takes the decision for searching, encircling, attacking and hunting the prey. The orders of alpha wolf are reinforced by the beta wolves which are placed at second level in dominant hierarchy of grey wolf. They take the feedback about the decision of alpha wolves from it low level wolves (delta, omega) and tell to the alpha. These are the best candidates to become a leader when alpha passes away. The next level wolfs are delta which are called as subordinate. These are the responsible for providing food to pack, protecting its members of pack. Delta wolves give support to alpha and beta during hunting.

The omega wolves are the lowest level wolves and they are not considering as much as important. The chasing, pursuing, encircling, hunting processes of grey as shown in following Figure 6.

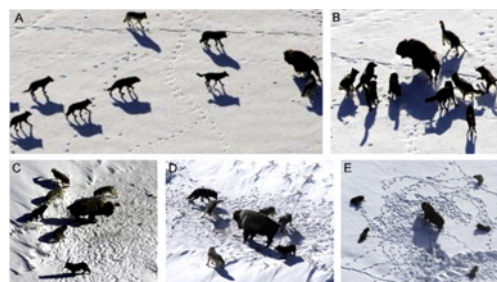


Figure 6. A) Chasing B) pursuing C) harassing D) encircling E) attacking

Mirjalili et al. simulates the above wolf nature as grey wolf optimizer. The total concept of GWO is given in pseudo-code 6

```

Begin
Initialize parameters a, A and C, population size N
For I = 1:N
Initialize population
Calculate the fitness for each searchagent f(Wi)
End for
Assign the best three solutions (W $\alpha$ , W $\beta$ , W $\delta$ )
For j=1:N
Update the position of each search agent
Update the A, C, a
Calculate the fitness function of search agent
End
Update W $\alpha$ , W $\beta$ , W $\delta$  vectors
Produce best solution
End

```

Pseudo-code 6. Pseudo-code of GWO.

3.7.1 Applications

G. M. Komaki and Vahid Kayvanfar⁷⁰ use grey wolf optimizer to find solution for assembly flow shop scheduling problem which reduces the processing time. In this they proposed two stages namely fabrication, assembly stages for job scheduling. They also proved that the proposed GWO with lower bound, dispatching rules were giving efficient results compared to other techniques

A Comparison made by Shahla Shoghian, Maryam Kouzehgar⁷¹ among wolf pack search and particle swarm optimization, genetic algorithm and Shuffled Frog Leaping(SFL). They suggest that solving a problem in less time SFL is recommended. Otherwise in case of attaing accuracy wolf pack search is recommended among algorithm specified

Ali Madadi and Mahmood Mohseni Motlagh used GWO⁷² for adjusting f Proportional-Integral-Derivative (PID) controller parameters in DC motors. They used GWO to find global optimization problem in controlling DC motor speed by tuning the PID control parameters. They proved that GWO-PID gave good results compare to PSO-PID controller

Mohd Herwan Sulaiman et al. used GWO⁷³ to solve Optimal Reactive Power Dispatch (ORPD) problems. They utilized GWO to find the best combinations of

generator voltages and tap changing transformation ratios in order to reduce voltage deviations and power loss. They used IEEE 30-bus system and IEEE 118-bus system to show the effectiveness of GWO technique compared to other techniques available. They also proved that GWO gave better result comparing to other techniques.

Unimodal, multimodal, fixed-dimension multimodal and composite functions are used by Mirjalili et al.⁶⁹ to prove the GWO performance in term of exploration, exploitation, local optima avoidance, and convergence.

E. Emary et al. used Grey wolf optimization⁷⁴ for reducing the features of a search space through interactions of individuals in population. They proposed classification accuracy based fitness function to find the optimal features. To prove the benchmark of GWO, they compared the results with the other available optimization techniques

Similarly GWO used for many problems like, parameter estimation in surface waves⁷⁵, economic dispatch problem⁷⁶.

4. Conclusions

This paper draws from the observation of various algorithms that each algorithm is efficient for a particular application based on its type. ACO as graph based algorithm suits to find the optimal path in determining the cost of the route compared to other algorithm in discrete environment. Bacterial foraging optimization algorithm is easy to implement and solve multi objective optimization problems. In this paper, we reveal that all these algorithms have some special features in finding the best solutions like either their high convergence speed capability or their ability to find the better solution. In spite of these advantages, they again suffer from other various problems. So researchers suggest that, the concept of hybrid algorithms to prevail the shortcomings which occur in various individual algorithms. Hybrid algorithms take the best features of the individual algorithms, combine with each other and thus be apt to be more effective than the individual algorithms.

5. References

1. Ghose T. Optimization techniques and an introduction to genetic algorithms and simulated annealing. Dept EEE, BIT, Mishra. 2002 Aug. p. 1–19.

2. Hooker JN. Logic based methods for optimization, principles and practice of constrained programming lecture notes in computer science. Springer; 1994; 874(1):336–49
3. Debanth: Research gate article. National Institute of Technology, Agartala
4. Das S, Biswas A, Dasgupta S, Abraham A. Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. 2009; 203(1):23–55.
5. Thomas. Survey of Bacterial Foraging Optimization Algorithm. IJISME. 2013 Mar; 1(4):11–2. ISSN: 2319-6386.
6. Bensujin BC, Kezi Selva Vijila, Jude Hemanth, Cynthia Hubert. Frequency dependent adaptive chemotaxis in bacterial foraging optimization for ST segment elevation myocardial infarction prediction. Indian Journal of Science and Technology. 2016 Jan; 9(1). Available from:10.17485/ijst/2016/v9i1.71685
7. Tana L, Lina FH. Adaptive comprehensive learning bacterial foraging optimization and its application on vehicle routing problem with time windows. Elsevier Neuro computing; 2015 Mar 3; 151(3):1208–15.
8. Kim DH, Cho AA. A hybrid genetic algorithm and bacterial foraging approach for global optimization. Elsevier, Information Sciences; 2007 Sep 15; 177(18): 3918–37.
9. Naveen S, Kumar SK, Rajalakshmi K. Distribution system reconfiguration for loss minimization using modified bacterial foraging optimization algorithm. Systems. 2015 July; 69(9):90–7.
10. Panigrahi BK, Pandi VR. Bacterial foraging optimization: Nelder–Mead hybrid algorithm for economic load dispatch. IET Generation, Transmission and Distribution. 2008; 2(4):556–65.
11. Hota PK, Barisal AK, Chakrabarti R. Economic emission load dispatch through fuzzy based bacterial foraging algorithm. International Journal of Electrical Power and Energy Systems. 2010; 32(7):794–803.
12. AliE S, Elazim SM. Bacteria foraging optimization algorithm based load frequency controller for interconnected power system. International Journal of Electrical Power and Energy Systems. 2011 Mar; 33(3):633–8.
13. Niu B, Fan Y, Xiao H, Xue B. Bacterial foraging based approaches to portfolio optimization with liquidity risk. Neuro computing. 2012 Dec 3; 98(8):90–100.
14. Tang WJ, Li MS, He S, Wu QH, Saunders JR. Optimal power flow with dynamic loads using bacterial foraging algorithm. International Conference on Power System Technology. 2006 Oct. p. 1–5.
15. Okaeme NA, Zanchetta P. Hybrid bacterial foraging optimization strategy for automated experimental in control design. IEEE Transactions on Industrial Informatics. 2013 May; 9(2):668–78.
16. Hanmandlu M, Verma OP, Susan S, Madasu VK. Color segmentation by fuzzy co-clustering of chrominance color features. Neurocomputing. 2013 Nov 23; 120(10):235–49.
17. Fister I, Fister IJR, Yang XS, Brest J. A comprehensive review of firefly algorithms swarm and evolutionary computation. Elsevier; 2013 Dec; 13(13):34–46.
18. Pal SK, Dr Rai CS, Dr. Sing AP. Comparative study of firefly algorithm and particle swarm optimization for noisy nonlinear optimization problems. International Journal of Intelligent Systems and Applications. 2012. p. 50–7.
19. Basu B, Mahanti GK. Thinning of concentric two-ring circular array antenna using fire fly algorithm. Scientia Iranica. 2012 Dec; 19(6): 1802–9.
20. Udaiyakumara KC, Chandrasekaran M. Application of firefly algorithm in job shop scheduling problem for minimization of makespan. 12th Global Congress on Manufacturing and Management. Elsevier. 2012; 97: 1798–807.
21. Sayadia MK, Hafezalkotob A, Nainia SGJ. Firefly-inspired algorithm for discrete optimization problems: An application to manufacturing cell formation. Journal of Manufacturing Systems. Elsevier. 2013 Jan; 32(1):78–84.
22. Tian Y, Shiyang WG. An improved inertia weight firefly optimization algorithm and application. International Conference on Control Engineering and Communication Technology (ICCECT). IEEE; 2012 Dec. p. 64–8.
23. Yang XS. Nature-Inspired Metaheuristic Algorithms, Luni-ver Press: UK: 2008.
24. Senthilnath J, Omkar SN, Mani V. Clustering using firefly algorithm: performance study, Swarm and Evolutionary Computation. 2011 Sep; 1(3):164–71.
25. Blum C. Ant colony optimization: Introduction and recent trends. Elsevier C. Blum / Physics of Life Reviews. 2005 Dec; 2(4):353–73.
26. Dorigo M, Maniezzo V, Coloni A. Positive feedback as a search strategy. Dipartimento di Elettronica, Politecnico di Milano, Italy. Tech. Rep. 91-016. 1991.
27. Dorigo M. Optimization, learning and natural algorithms (in italian). Ph.D. dissertation, Dipartimento di Elettronica, Politecnico di Milano, Italy. 1992.
28. Gambardella LM, Dorigo M. Ant-Q: A reinforcement learning approach to the traveling salesman problem. In: A. Prieditis and S. Russell, editors. 12th International Conference on Machine Learning (ML-95). Morgan Kaufmann Publishers; 1995. p. 252–60.
29. Dorigo M, Gambardella LM. Ant colonies for the traveling salesman problem. Bio Systems. 1997; 43(2):73–81.
30. Gambardella LM, Dorigo M. Solving symmetric and asymmetric TSPs by ant colonies. IEEE International Conference on Evolutionary Computation (ICEC'96). 1996. p. 622–7.
31. Stutzle T, Hoos HH. Improving the Ant System: A detailed report on the MAX–MIN Ant system. FG Intellektik, FB Informatik, TU Darmstadt. Germany. Tech. Rep. AIDA–96–12. 1996.
32. Stutzle T, Hoos HH. MAX–MIN Ant System. Future Generation Computer Systems. 2000 June; 16(8):889–914.
33. Bullnheimer B, Hartl RF, Strauss C. A new rank based version of the Ant System— a computational study. Institute of Management Science, University of Vienna. Tech. Rep. 1997.
34. Bullnheimer B, Hartl RF, Strau BC. A new rank-based version of the Ant System: A computational study. Central European Journal for Operations Research and Economics. 1999; 7(1):25–38
35. Maniezzo. Exact and approximate nondeterministic tree-

- search procedures for the quadratic assignment problem. *Inform Journal on Computing*. 1999;11(4):358–69.
36. Cordon O, Viana DIF, Herrera F, Moreno L. A new ACO model integrating evolutionary computation concepts: The best-worst Ant System. In *Proceedings of ANTS 2000 - From Ant Colonies to Artificial Ants: Third International Workshop on Ant Algorithms Key: citeulike: 6592051*. 2000. p. 22–9.
 37. Blum C, Dorigo M. The hyper-cube framework for Ant Colony Optimization. *IEEE Transactions on systems, man, and cybernetics—Part B: Cybernetics*. 2004 Apr; 34(2).
 38. Gaviria MVI, Morcillo DA, Tejedor BVJ. Ant Colony Optimization Algorithm for Microwave Corrugated Filters Design. *Journal of Computational Engineering*. 2013. p. 1–8.
 39. Florez E, Gomez W, Bautista L. An ant colony optimization algorithm for job shop scheduling problem. *IJAIA*; 2013 July; 4(4):53–66.
 40. Kabir M, Shahjahan MD, Murase. Ant Colony Optimization toward Feature Selection. 2013. Available from: <http://dx.doi.org/10.5772/51707>
 41. Liu XY. Ant colony optimization algorithm based on dynamical pheromones for clustering analysis. *International Journal of Hybrid Information Technology*. 2014; 7(2):29–38.
 42. Tomera M. Ant colony optimization algorithm applied to ship steering control. *Procedia Computer Science*. 2014;(35):83–92.
 43. Venkatesan S, Srinivasa Rao Madane S. Study on identification of face in versatile circumstances through genetic and ant colony optimization algorithms. *Indian journal of Science and Technology*. 2015 Nov; 8(30). Available from:10.17485/ijst/2015/v8il/75405
 44. Tang J, Maa Y, Guanb J, Yan C. A max–min ant system for the split delivery weighted vehicle routing problem. *Expert Systems with Applications*; 2013 Dec 15; 40(18):7468–77.
 45. Vangili A, Thangadurai K. Detection of black hole attack in mobile ad-hoc networks using ant colony optimization-simulation analysis. *Indian Journal of Science and Technology*. 2015 July; 8(13). Available from: 10.17485/ijst/2015/v8il3/58200
 46. Dizaji ZA, Gharehchopogh FS. A hybrid of ant colony optimization and chaos optimization algorithms approach for software cost estimation. *Indian Journal of Science and Technology*. 2015 Jan; 8(2). Available from:10.17485/ijst/2015/V8i2/57776
 47. Janaki Raam KV, Rajkumar K. A novel approach using parallel ant colony optimization algorithm for detecting routing path based on cluster head in wireless sensor network. *Indian Journal of Science and Technology*, 2015 July;8(16). Available from:10.17485/ijst/2015/V8i16/61986
 48. Yang XS. Cuckoo search and firefly algorithm. *Studies in computational intelligence*. Switzerland: Springer; 2014; 516. Available from: 10.1007/978-3-319-02141-6_1
 49. Nguyen TT, Truong AV. Distribution network reconfiguration for power loss minimization and voltage profile improvement using cuckoo search algorithm. *Electrical Power and Energy Systems*. 2015 June; (68):233–42.
 50. Valian E, Tawakoni S, Mohanna S, Haghi A. Improved cuckoo search for reliability optimization problems. *Computers and Industrial Engineering*. 2013 Jan; 64(1):459–68.
 51. Cobos C, Collazos HM, Collazo RU, Mendoza M, Leon E, Viedma EH. Clustering of web search results based on the cuckoo search algorithm and Balanced Bayesian Information Criterion. *Science Direct Information Sciences*. 2014 Oct 10; 281(1):248–64.
 52. Civicioglu P, Besdok E. A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. *Springer Science, Business Media*; 2013 Apr; 39(4):315–46. Available from: 10.1007/s10462-011-9276-0.
 53. Ashok KB, Aditi B, Santanu C, Saxena PK. Cryptanalysis of vigenere cipher using cuckoo search. *Science Direct, Applied Soft Computing*. 2015 Jan; 26(6): 315–24
 54. Li X, Yin M. Modified cuckoo search algorithm with self-adaptive parameter method. *Science Direct, Information Sciences*. 2015; 298(8):80–97.
 55. Kuo RJ, Huang YD, Lin CC, HungYL, Zulvia FE. Automatic kernel clustering with bee colony optimization Algorithm. *Information Sciences*. 2014 Nov 1; 283(3):107–22
 56. Alzaqebah M, Abdullah S. Hybrid bee colony optimization for examination timetabling problems. In *Computer and Operations Research*. 2015 Feb; 54(4):142–54.
 57. Kumar R. Directed bee colony optimization algorithm. *Swarm and Evolutionary Computation*. Elsevier; 2014 Aug; 17(7): 60-73.
 58. Teodorovic D, Dell'Orco M. Bee colony optimization-A cooperative learning approach to complex transportation problems. *Advanced OR and AI Methods in Transportation*. 2005.
 59. Khalilzadeh M. A honey bee swarm optimization algorithm for minimizing the total costs of resources in MRCPS. *Indian Journal of Science and Technology*. 2015 June; 8(11). Available from:10.17485/ijst/2015/V8i11/71804
 60. Chalotra S, Sehra SK, Brar YS, Kaur N. Tuning of COCO-MO model parameters by using bee colony optimization. *Indian Journal of Science and Technology*. 2015 June ;8(14). Available from:10.17485/ijst/2015/V8i14/70010
 61. Jannaty MH, Eghbalzadeh A, Hosseini SA. Hybrid ANFIS model for predicting scour depth using particle swarm optimization. *Indian Journal of Science and Technology*. 2015 sep ;8(22). Available from: 10.17485/ijst/2015/V8i22/79321
 62. Izadi H, Effatnejad R. Power system stabilizers and control of reactive power compensator in multi-machine by particle swarm optimization algorithm. *Indian Journal of Science and Technology*. 2015 Feb; 8(3). Available from: 10.17485/ijst/2015/V8i3/59595
 63. Shahina Banu A, Helen WR. Scheduling deadline constrained task in hybrid iaas cloud using cuckoo driven par-

- ticle swarm optimization. *Indian Journal of Science and Technology*; 2015 July; 8(16). Available from: 10.17485/ijst/2015/V8i16/61830
64. Mahalakshmi S, Velumurugan T. Detection of brain tumor by particle swarm optimization using image segmentation. *Indian Journal of Science and Technology*. 2015 Sep; 8(22). Available from: 10.17485/ijst/2015/V8i22/79092
 65. Rajkumar R, James Albert A, Chandrakala D. Weather Forecasting using Fuzzy Neural Network(FNN) and Hierarchy Particle Swarm Optimization Algorithm(HPSO). *Indian Journal of Science and Technology*. 2015 June; 8(12). Available from: 10.17485/ijst/2015/V8i12/52708
 66. Yazd HGH, Arabshahi SJ, Tavousi M, Alvani A. Optimal designing of concrete gravity dam using Particle Swarm Optimization Algorithm(PSO). *Indian Journal of Science and Technology*. 2015 June; 8(12). Available from:10.17485/ijst/2015/v8i12/70714
 67. Rahmati M, Effatnejad R, Safari A. Comprehensive Learning Particle Swarm Optimization (CLPSO) for multi-objective optimal Power Flow. *Indian Journal of Science and Technology*. 2014 Jan; 7(3). Available from:10.17485/ijst/2014 /v7i3/47643
 68. Effatnejad R, Rouhi F. Unit commitment in power system by combination of Dynamic Programming(DP), Genetic Alogirhm(GA) and Particle Swarm Optimization(PSO). *Indian Journal of Science and Technology*. 2015 Jan; 8(2). Available from:10.17485/ijst/2015 /v8i2/57782
 69. Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Advances in Engineering Software*. 2014 Mar; 69(69):46–61.
 70. Komaki GM, Kayvanfar. Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time. *Journal of Computational Science*. 2015 May; 8(8):109–20.
 71. Shoghian S, Kouzehgar M. A comparison among wolf pack search and four other optimization algorithms. *World Academy of Science, Engineering and Technology*. 2012; 6(6):12–23.
 72. Madadi A, Motlagh MM .Optimal control of DCmotor using grey wolf optimizer algorithm. *TJEAS Journal-2014-4-04/373-379*. 2014; 4(4):373–79. ISSN: 2051-0853.
 73. Sulaiman MHZ, Mohamed MR, Aliman O. Using the gray wolf optimizer for solving optimal reactive power dispatch problem. *Applied Soft Computing* . 2015 July; 32(2):286–92.
 74. Emary E, Zawbaa MH, Grosan C, Hassenian AE. Feature subset selection approach by gray-wolf optimization. *Afro-European Conference for Industrial Advancement Advances in Intelligent Systems and Computing*. 2015; 334(2):1–13.
 75. Song X, Tang L, Zhao S, Zhang X, Li L, Huang J, Cai W. Grey Wolf Optimizer for parameter estimation in surface waves. *Soil Dynamics and Earthquake Engineering*. 2015 Aug; 75(5):147–57.
 76. Song HM, Sulaiman MH, Mohamed MR. An application of Grey wolf optimizer for solving combined economic emission dispatch problems. *International Review on Modelling and Simulations*. 2014; 7(5):838–44.