Chapter 2

Literature Survey

This chapter provides a detailed description of the population based algorithms specially Particle Swarm Optimization, Artificial Bee Colony Algorithm and Animal Migration Optimization algorithm. History, recent modifications and applications of PSO, ABC and AMO discussed in detail in this chapter.

2.1 Intelligent Algorithms

The term intelligent algorithm is intended to unify a collection of interesting and useful computational tools under a consistent and accessible banner. These algorithms are drawn from many sub-fields of artificial intelligence not limited to the scruffy fields of biologically inspired computation, computational intelligence and meta-heuristics. These algorithm may be inspired by intelligent behaviour of individual agent like insects, animals or may simulate some natural phenomenon that happen in nature. An alternative name Population based algorithm may also considered. Classical search and optimization methods make use of one solution in each iteration and outcome is also a single optimized solution shown in Figure 2.1. These classical optimization methods are not able to solve complex optimization problems that are non-differentiable, discrete and multi-model. These classical model includes mathematical optimization algorithms (such as Newton’s method and Gradient Descent that use derivatives to locate a local minimum) and direct search methods (such as the Simplex method and the Nelder-Mead method that use a search pattern to locate optima).

Population-based optimization algorithms find near-optimal solutions to the complex optimization problems by being motivated from nature. These algorithms use a population of solutions in each iteration and outcome is also a population of solutions shown in Figure 2.2. Population-based algorithms share many common concepts.
They could be viewed as an iterative improvement in a population of solutions. First, the population is initialized. Then, a new population of solutions is generated. Finally, this new population is integrated into the current one using some selection procedures. The search process is stopped when a given condition is satisfied that is also known as stopping criterion. Population based methods are appropriate to solve multi-objective optimization problems. Fitness based update of all the potential solutions is a main and common feature of all the population based algorithms. Hence, the population is moved towards better solution areas of the search space. Two important classes of population-based optimization algorithms are evolutionary algorithms [38] and swarm intelligence-based algorithms [39]. A tree architecture of population based algorithms is shown in Figure 2.3.
2.1.1 Evolutionary Algorithms

Evolutionary algorithm is an umbrella term used to describe computer-based problem solving systems which use computational models of evolutionary processes as key elements in their design and implementation. Evolutionary algorithms (EAs) are inspired by the biological model of evolution and natural selection first proposed by Charles Darwin in 1859. In the natural world, evolution helps species adapt to their environments. Environmental factors that influence the survival prospects of an organism include climate, availability of food and the dangers of predators. Evolutionary algorithms are based on a simplified model of this biological evolution. To solve a particular problem we create an environment in which potential solutions can evolve. The environment is shaped by the parameters of the problem and encourages the evolution of good solutions.

Although Genetic Algorithm (GA) [40], Genetic Programming (GP) [41], Evolution Strategy (ES) [42] and Evolutionary Programming (EP) [43] are popular evolutionary algorithms, GA is the most widely used one in the literature. GA is equally useful for continuous and discrete nature of problems. GA is based on genetic science and natural selection and it attempts to simulate the phenomenon of natural evolution at genotype level while ES and EP simulate the phenomenon of natural evolution at phenotype level. One of the evolutionary algorithms which has been introduced recently is Differential Evolution (DE) algorithm [44]. The DE is very efficient for continuous problems. In the basic GA, selection operation is applied to the solutions evaluated by the evaluation unit. At this operation the chance of a solution being selected as a parent depends on the fitness value of that solution. While, at the selection operation of the DE algorithm, all solutions have an equal chance of being selected as parents, i.e. the chance does not depend on their fitness values.

Algorithm 2.1 and Figure 2.4 show the general scheme of an evolutionary algorithm in form of algorithm and figure respectively.

**Algorithm 2.1 The General Scheme of an Evolutionary Algorithm**

- Initialize population with arbitrary candidate solutions;
- Evaluate each candidate solution;
- while Termination criteria is not satisfied do
  - Step 1: Select Parent solution;
  - Step 2: Recombine pair of parents;
  - Step 3: Mutate the resulting offspring;
  - Step 4: Evaluate the new candidate solutions;
  - Step 5: Select individual for next generation;
- end while
2.1.2 Swarm Intelligence based Algorithms

Swarm intelligence is the study of computational systems inspired by the 'collective intelligence'. Collective Intelligence emerges through the cooperation of large numbers of homogeneous agents in the environment. Examples include schools of fish, flocks of birds, and colonies of ants. Such intelligence is decentralized, self-organizing and distributed throughout an environment. In recent years, swarm intelligence has also attracted the interest of many researchers. However, a swarm can be considered as any collection of interacting agents or individuals. These individuals sometimes can solve complex tasks without centralized control. The definition given by Bonabeau for the swarm intelligence is “any attempt to design algorithms or distributed problem-solving devices inspired by the collective behaviour of social insect colonies and other animal societies” [45]. Researchers have analyzed intelligent behaviors of swarm and designed algorithms that can be used to solve nonlinear, nonconvex or combinatorial optimization problems in many science and engineering domains. Previous research [46, 47, 48, 49, 8, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59] have shown that algorithms based on Swarm Intelligence have great potential to find a solution of real world optimization problem. The algorithms based on swarm intelligence that have emerged in recent years include Ant Colony Optimization (ACO) [49], Particle Swarm Optimization (PSO) [8], Bacterial Foraging Optimization (BFO) [60], Artificial Bee Colony Optimization (ABC) [9] etc.

Self organization and division of labour are two most important and significant properties of swarm intelligent behavior [9]. Each of the properties is explained as follows:

1. **Self-organization** is an important feature of a swarm structure, which re-
sults in global level response by means of interactions among its low-level components without a central authority or external element enforcing it through planning. Therefore, the globally coherent pattern appears from the local interaction of the components that build up the structure, thus the organization is achieved in parallel as all the elements act at the same time and distributed as no element is a central coordinator. Bonabeau et al. have defined the following four important characteristics on which self-organization is based [45]:

(a) **Positive feedback** is an information extracted from the output of a system and reapplied to the input to promotes the creations of convenient structures. In the field of swarm intelligence positive feedback provides diversity and accelerate the system to new stable state.

(b) **Negative feedback** compensates the effect of positive feedback and helps to stabilize the collective pattern

(c) **Fluctuations** are the rate or magnitude of random changes in the system. Randomness is often crucial for efflorescent structures since it allows the findings of new solutions. In foraging process, it helps to get-rid of stagnation.

(d) **Multiple interactions** provide the way of learning from the individuals within a society and thus enhance the combined intelligence of the swarm.

2. **Division of labour** is a cooperative labour in specific, circumscribed tasks and like roles. In a group, there are various tasks, which are performed simultaneously by specialized individuals. Simultaneous task performance by cooperating specialized individuals is believed to be more efficient than the sequential task performance by unspecialized individuals [61, 62, 63].

### 2.2 Standard Particle Swarm Optimization Algorithm

PSO is an optimization technique which simulates the birds flocking behavior. PSO is a dynamic population of active, interactive agents with very little in the way of inherent intelligence. In PSO, whole group is called *swarm* and each individual is called *particle* which represents possible candidate’s solution. The swarm finds food for its self through social learning by observing the behavior of
nearby birds who appeared to be near the food source. Initially each particle is initialized within the search space randomly and keeps the information about its personal best position known as \( p_{best} \), swarm best position known as \( g_{best} \) and current velocity \( V \) with which it is moving, in her memory. Based on these three values, each particle updates its position. In this manner, whole swarm moves in better direction while following collaborative trail and error method and converges to single best known solution.

For an \( D \)-dimensional search space, the \( i^{th} \) particle of the swarm is represented by a \( D \)-dimensional vector, \( X_i = (x_{i1}, x_{i2}, ..., x_{iD}) \). The velocity of this particle is represented by another \( D \)-dimensional vector \( V_i = (v_{i1}, v_{i2}, ..., v_{iD}) \). The previously best visited position of the \( i^{th} \) particle is denoted as \( P_i = (p_{i1}, p_{i2}, ..., p_{iD}) \). \( g \) is the index of the best particle in the swarm. PSO swarm uses two equations for movement called velocity update equation and position update equation. The velocity of the \( i^{th} \) particle is updated using the velocity update equation given by equation (2.1) and the position is updated using equation (2.2).

\[
v_{ij} = v_{ij} + c_1 r_1 (p_{ij} - x_{ij}) + c_2 r_2 (p_{gj} - x_{ij}) \tag{2.1}
\]

\[
x_{ij} = x_{ij} + v_{ij} \tag{2.2}
\]

where \( j = 1, 2, ..., D \) represents the dimension and \( i = 1, 2, ..., S \) represents the particle index. \( S \) is the size of the swarm and \( c_1 \) and \( c_2 \) are constants (usually \( c_1 = c_2 \)), called cognitive and social scaling parameters respectively or simply acceleration coefficients. \( r_1 \) and \( r_2 \) are random numbers in the range \([0, 1]\) drawn from a uniform distribution.

The right hand side of velocity update equation (2.1) consists of three terms, the first term \( v_{ij} \) is the memory of the previous direction of movement which can be thought of as a momentum term and prevents the particle from drastically changing direction. The second term \( c_1 r_1 (p_{ij} - x_{ij}) \) is called cognitive component or persistence which draws particle back to their previous best situation and enables the local search in swarm. The last term \( c_2 r_2 (p_{gj} - x_{ij}) \) is known as social component which allows individuals to compare themselves to others in it’s group and is responsible for global search. The Pseudo-code for Particle Swarm Optimization, is described as follows:

Based on the neighborhood size, initially two versions of PSO algorithm were presented in literature namely, global version of PSO which is the original PSO (PSO-G) and the local version of PSO (PSO-L)[64]. The only difference between PSO-G and PSO-L is that the term \( p_g \) in social component in velocity update
**Algorithm 2.2** Particle Swarm Optimization Algorithm:

| Initialize the parameters, \(w\), \(c_1\) and \(c_2\); |
| Initialize the particle positions and their velocities in the search space; |
| Evaluate fitness of individual particles; |
| Store gbest and pbest; |
| **while** stopping condition(s) not true **do** |
| **for** each individual, \(X_i\) **do** |
| **for** each dimension \(j, x_{ij}\) **do** |
| (i) Evaluate the velocity \(v_{ij}\) using (2.1); |
| (ii) Evaluate the position \(x_{ij}\) using (2.2); |
| **end for** |
| **end for** |
| Evaluate fitness of updated particles; |
| Update gbest and pbest; |
| **end while** |
| Return the individual with the best fitness as the solution; |

equation (2.1). For PSO-G, it refers the best particle of whole swarm while for PSO-L it represents the best particle of the individual’s neighborhood. The social network employed by the PSO-G reflects the star topology which offers a faster convergence but it is very likely to converge prematurely. While PSO-L uses a ring social network topology where smaller neighborhoods are defined for each particle. It can be easily observed that due to the less particle inter connectivity in PSO-L, it is less susceptible to be trapped in local minima but at the cost of slow convergence. In general, PSO-G performs better for unimodal problems and PSO-L for multimodal problems.

Velocity update equation in PSO determines the balance between exploration and exploitation capability of PSO. In Basic PSO, no bounds were defined for velocity, due to which in early iterations the particles far from gbest, will take large step size and are very much intended to leave the search space. Thus to control velocity so that particle update step size is balanced, velocity clamping concept was introduced. In velocity clamping, whenever velocity exceeds from its bounds, it is set at its bounds. To avoid the use of velocity clamping and to make balance between exploration and exploitation, a new parameters called inertia weight [65] was introduced in velocity update equation as:

\[
v_{ij} = w \ast v_{ij} + c_1r_1(p_{ij} - x_{ij}) + c_2r_2(p_{gj} - x_{ij})
\]

(2.3)

where inertia weight is denoted by \(w\). In subsequent section, the proposed PSO algorithm is explained in details.
### 2.2.1 Recent Modifications in Particle Swarm Optimization Algorithm

The PSO is now one of the most commonly used optimization techniques. Advances on PSO are endless, here we classify all the modifications in two different classes. The categorization is as follow:

1. Modifications of PSO, including bare-bones PSO, chaotic PSO, fuzzy PSO, quantum-behaved PSO, opposition-based PSO and other significant modifications,

2. Hybridization of PSO with other nature inspired algorithms including ABC, ACO, DE, GA, artificial immune system (AIS), Tabu search (TS), simulated annealing (SA), biogeography-based optimization (BBO), and harmonic search (HS),

#### 2.2.1.1 Modifications in PSO

Jau et al. [66] proposed a modified quantum-behaved particle swarm optimization for parameters estimation of generalized nonlinear multi-regressions model based on Choquet integral with outliers. They modified the popular variant of quantum-behaved PSO and used a high breakdown regression estimator and a least-trimmed-squares method to eliminate the influence caused by observations containing outliers. Jamalipour et al. [67] presented QPSO with differential mutation operator (QPSO-DM) for optimizing WWER-1000 core fuel management. The results showed that QPSO-DM performs better than the others. Bagheri et al. [68] used QPSO to forecast financial time series, especially for the foreign exchange market. Tang et al. [69] proposed an improved QPSO algorithm for continuous nonlinear large-scale problems based on memetic algorithm and memory mechanism. The memetic algorithm was used to make all particles gain some experience through a local search before being involved in the evolutionary process, and the memory mechanism was used to introduce a bird kingdom with memory capacity, both of which can improve the global search ability of the algorithm. Davoodi et al. [70] proposed a new approach, based on a hybrid algorithm combining of improved QPSO and simplex algorithms. QPSO was the main optimizer of algorithm, which can give a good direction to the optimal global region. Nelder-Mead simplex method was used as a local search to fine-tune the obtained solution from QPSO.

J Kennedy proposed a variant of PSO namely bare-bones PSO (BBPSO) [71]. In which the velocity and position update rules are substituted by a procedure that
samples a parametric probability density function. Zhang et al. [72] used both mutation and crossover operators of DE algorithm to modify original BBPSO in order to update certain particles in the population. The performance of the resulting algorithm was tested on 10 benchmark functions and applied to 16 vapor-liquid equilibrium problems. Zhang et al. [73] analyzed the sampling distribution in BBPSO, based on which they propose an adaptive version inspired by the cloud model, which adaptively produced a different standard deviation of the Gaussian sampling for each particle according to the evolutionary state in the swarm, which provided an adaptive balance between exploitation and exploration on different objective functions. Zhang et al. [74] proposed three global optimization algorithms for phase and chemical equilibrium calculations, which played a significant role in the simulation, design, and optimization of separation processes in chemical engineering. The proposed algorithms were unified BBPSO (UBBPSO), integrated DE (IDE), and IDE without Tabu list and radius (IDE\(_N\)).

Chaos theory have been integrated with PSO to improve its performance. This type of PSO variant is called chaotic PSO (CPSO). Chuang et al. [75] introduced chaotic maps into catfish particle swarm optimization. The proposed method increased the search capability via the chaos approach. Zhang and Wu [76] proposed adaptive CPSO (ACPSO) to train the weights/biases of two-hidden-layer forward neural network in order to develop a hybrid crop classifier for polarimetric synthetic aperture radar images. Dai et al. [77] proposed a novel adaptive chaotic embedded PSO (ACEPSO) and applied it in wavelet parameters estimation. ACEPSO embedded chaotic variables in standard PSO and adjusted parameters nonlinearly and adaptively. It also estimated particles whether being focusing or discrete by judging the population fitness variance of particle swarm and average distance amongst points; then chaotic researching was applied to escaping from premature convergence. Li et al. [78] proposed a novel chaotic particle swarm fuzzy clustering (CPSFC) algorithm based on a new CPSO and gradient method. The new CPSO algorithm is the combination of adaptive inertia weight factor (AIWF) and iterative chaotic map with infinite collapses (ICMIC) based chaotic local search. The CPSFC algorithm utilized CPSO to search the fuzzy clustering model, exploiting the searching capability of fuzzy c-means (FCM) and avoiding its major limitation of getting stuck at locally optimal values. Meanwhile, gradient operator is adopted to accelerate convergence of the proposed algorithm.

In order to make PSO more powerful, it was combined with fuzzy sets theory. This type of PSO variant is called fuzzy PSO (FPSO). Juang et al. [79] proposed an adaptive FPSO (AFPSO) algorithm. The proposed AFPSO utilized fuzzy set theory to adjust PSO acceleration coefficients adaptively and was thereby able to
improve the accuracy and efficiency of searches. Incorporating this algorithm with quadratic interpolation and crossover operator further enhanced the global searching capability to form a new variant called AFPSO-Q1. Alfi and Fateh [80] presented a novel improved FPSO (IFPSO) algorithm to the intelligent identification and control of a dynamic system. The proposed algorithm estimated optimally the parameters of system and controlled by minimizing the mean of squared errors. The PSO was enhanced intelligently by using a fuzzy inertia weight to rationally balance the global and local exploitation abilities. Every particle dynamically adjusted inertia weight according to particles best memories using a nonlinear fuzzy model. Yang et al. [81] proposed a novel FPSO algorithm based on fuzzy velocity updating strategy in order to optimize the machining parameters. The proposed FPSO algorithm achieved good results on few benchmark problems and obtained significant improvements on two illustrative case studies of multipass face milling.

Opposition-based learning (OBL) theory was integrated with PSO, and the new variant was dubbed opposition-based PSO (OPSO). Dhahri and Alimi [82] proposed the OPSO using the concept of opposite number to create a new population during the learning process. They combined OPSO with BBFNN. The results showed that the OPSO-BBFNN produced a better generalization performance. Wang et al. [83] presented an enhanced PSO algorithm called GOPSO, which employed generalized OBL (GOBL) and Cauchy mutation. GOBL provided a faster convergence and the Cauchy mutation with a long tail helped trapped particles escape from local optima. Dong et al. [84] proposed an evolutionary circle detection method based on a novel chaotic hybrid algorithm (CHA). The method combined the strengths of PSO, GA, and chaotic dynamics and involved the standard velocity and position update rules of PSOs, with the ideas of selection, crossover, and mutation from GA. The OBL was employed in CHA for population initialization. In addition, the notion of species was introduced into the proposed CHA to enhance its performance in solving multimodal problems. Gao et al. [85] proposed a novel PSO called CSPSO to improve the performance of PSO on complex multimodal problems. Specifically, a stochastic search technique was used to execute the exploration in PSO. In addition, to enhance the global convergence, when producing the initial population, both opposition-based learning method and chaotic maps were employed.

Some researchers make tentative research on improving the optimization performance of PSO by other efficient strategies. For example, Chuang et al. [86] proposed a novel catfish PSO, the mechanism of which is dependent on the incorporation of a catfish particle into the linearly decreasing weight particle swarm optimization. Unlike other ordinary particles, the catfish particles initialized a new
search from the extreme points of the search space when the gbest fitness value had not been changed for a given time, which resulted in further opportunities to find better solutions for the swarm by guiding the whole swarm to promising new regions of the search space and accelerating convergence. Shi and Liu [87] proposed a hybrid improved PSO, in which chaos initialization was introduced to improve the population diversity, and adaptive parameters control strategy was employed to make it independent from specific problem. Besides, novel acceptance policy based on Metropolis rule was taken to guarantee the convergence of the algorithm. Zhang et al. [88] proposed a new adaptive PSO (APSO) that could dynamically follow the frequently changing market demand and supply in each trading interval. A numerical example served to illustrate the essential features of the approach.

2.2.1.2 Hybridization of PSO

PSO was combined with some traditional and evolutionary optimization algorithms in order to take the advantages of both methods and compensate the weaknesses of each other. This type of PSO is called hybridized PSO.

Kuo and Hong [89] presented a two-stage method of investment portfolio based on soft computing techniques. The first stage used data envelopment analysis to select most profitable funds, while hybrid of GA and PSO was proposed to conduct asset allocation in the second stage. Chen and Kurniawan [90] presented a two-stage optimization system to find optimal process parameters of multiple quality characteristics in plastic injection molding. Taguchi method, BPNN, GA, and combination of PSO and GA (PSO-GA) were used in this study to find optimum parameter settings. Nazir et al. [91] extracted facial local features using local binary pattern (LBP) and then fused these features with clothing features, which enhanced the classification accuracy rate remarkably. In the following step, PSO and GA were combined to select the most important features set that more clearly represented the gender and thus the data size dimension was reduced.

Tang et al. [92] presented a novel dynamic PSO algorithm based on improved artificial immune network (IAINPSO). Based on the variance of the populations fitness, a kind of convergence factor was adopted in order to adjust the ability of search. The experimental results showed that not only did the new algorithm satisfy convergence precision, but also the number of iterations was much less than traditional scheme and had much faster convergent speed, with excellent performance in the search of optimal solution to multidimensional function. Zhang et al. [93] proposed a more pragmatic model for stochastic networks, which considered not only determinist variables but also the mean and variances of random
variables. In order to accelerate the solution of the model, they integrated PSO with chaos operator and AIS.

Chen and Chien [94] presented a new method, called the genetic simulated annealing ant colony system with particle swarm optimization techniques, for solving the TSP. The experimental results showed that both the average solution and the percentage deviation of the average solution to the best known solution of the proposed method were better than existing methods. Xiao et al. [95] considered the features of the MRCMPSP problem. They employed ant colony's labor division to establish a task priority-scheduling model firstly. Then, they used improved PSO to find out the optimum scheduling scheme. The approach integrating the above two algorithms had abilities of both local search and global search.

El-Abd [96] tested a hybrid PSO and ABC algorithm on the CEC13 testbed. The hybridization technique was a component-based one, where the PSO algorithm was augmented with an ABC component to improve the personal best of the particles. Sharma et al. [97] proposed a variant called Local Global variant ABC (LGABC) to balance the exploration and exploitation in ABC. The proposal harnessed the local and global variant of PSO into ABC. The proposed variant was investigated on a set of thirteen well-known constrained benchmarks problems and three chemical engineering problems, which showed that the variant can get high-quality solutions efficiently. Kiran and Gndz [98] presented a hybridization of PSO and ABC approaches, based on recombination procedure. The global best solutions obtained by the PSO and ABC were used for recombination, and the solution obtained from this recombination was given to the populations of the PSO and ABC as the global best and neighbor food source for onlooker bees, respectively.

Maione and Punzi [99] proposed a two-step design approach. First, DE determined the fractional integral and derivative actions satisfying the required time-domain performance specifications. Second, PSO determined rational approximations of the irrational fractional operators as low-order, stable, minimum-phase transfer functions with poles interlacing zeros. Extensive time- and frequency-domain simulations validated the efficiency of the proposed approach. Fu et al. [100] presented a hybrid DE with QPSO for the unmanned aerial vehicle (UAV) route planning on the sea. It combined DE algorithm with the QPSO algorithm in an attempt to enhance the performance of both algorithms. Experimental results demonstrated that the proposed method was capable of generating higher quality paths efficiently for UAV than any other tested optimization algorithms.
2.3 Artificial Bee Colony (ABC) Algorithm

Artificial Bee Colony (ABC) algorithm is considered an efficient nature inspired algorithm to solve continuous unconstrained optimization problems. It was developed by taking inspiration from food foraging behavior of honey bee. Bees are classified in three categories namely employed bee, onlooker bee and scout bee. The ABC metaheuristic technique is stimulated through the spontaneous food foraging behavior of the honey bee creature. Honey bee insect most intuitive creation of nature, it shows combined intellectual behavior at the same time as penetrating the food. The honey bee can memorize the ecological circumstances, can accumulate and distribute the information and can decide according to these observations. As per the changes in the surroundings, the bee updates its position, assign the responsibilities dynamically and go on further by means of societal erudition and education. This extraordinary conduct of honey bees motivates research scientists to imitate the intellectual food foraging behavior of the bees. Analogous to the intelligent food foraging behavior of honey bees, algorithm also carry three steps employed bee phase, onlooker bee phase and scout bee phase.

The key steps of ABC algorithm are summarized as follow:

1. Initialization of the swarm:

   \[ x_{ij} = x_{\min j} + \text{rand}[0, 1](x_{\max j} - x_{\min j}), \]  

   Here the \( i^{th} \) solution represented by \( x_i \), the lower and upper bounds of \( x_i \) in \( j^{th} \) dimension are \( x_{\min j} \) and \( x_{\max j} \) respectively and \( \text{rand}[0, 1] \) is an evenly scattered arbitrary number in the range \([0, 1]\).

2. Employed bee phase:

   \[ x_{\text{new}ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), \]  

   This phase update position of \( i^{th} \) candidate solution. Here \( k \in \{1, 2, ..., SN\} \) and \( j \in \{1, 2, ..., D\} \) are arbitrarily selected indicators and \( k \neq i \). \( \phi_{ij} \) is an arbitrary number in the range \([-1,1]\).

3. Onlooker bee phase: Based on fitness of food sources onlooker bees decides that which food sources are most feasible and select a solution with probability \( \text{prob}_i \). Here \( \text{prob}_i \) may be decided with the help of fitness (there may
be some other method).  

\[
prob_i(G) = \frac{0.9 \times \text{fitness}_i}{\text{maxfit}} + 0.1,  
\]

(2.6)

Here \(\text{fitness}_i\) represents the fitness value of the \(i^{th}\) solution and \(\text{maxfit}\) represents the highest fitness among all the solutions. Onlooker bees also select a solution and update it according to their probability of selection.

4. Scout bee phase: In this phase a food source considered as discarded and replaced by arbitrarily generated new solution if its location is not modified for a certain number of cycles.

### 2.3.1 Recent Modifications in Artificial Bee Colony (ABC) Algorithm

The Artificial Bee Colony algorithm is very popular and simple algorithm to solve complex optimization problems. There are lot of complex real world problem that are not solvable by conventional methods. In order to solve this type of problem population base techniques are very helpful. The use of intelligence appeared from collective actions of a swarm that is by and large used to crack multifaceted problems when an individual is not able to find solution of a meticulous problem. G. Yavuz and D. Aydin [101] proposed an Angle modulated Artificial Bee Colony algorithms for feature selection using angle modulation technique. Y. Marinakis et al. proposed A Hybrid Discrete Artificial Bee Colony Algorithm for the Multicast Routing Problem [102] implemented with Variable Neighbourhood Search. N. Sharma et al. [103] proposed a new variant of ABC namely Modified Artificial Bee Colony Algorithm Based on Disruption Operator. This algorithm added a new phase in ABC algorithm namely disruption phase. This algorithm modify all solutions except best one using the distance from current best solution and disrupt or attract new solution. TK Sharma and Millie Pant [104] Shuffled artificial bee colony algorithm. The Shuffled-ABC is hybrid of ABC algorithm and shuffled frog-leaping algorithm. The Shuffled-ABC divides the initial population into two different groups based on their fitness. The ABC applied in best fitted group and SFLA applied on rest population. A. Yurtkuran and Erdal Emel investigated a discrete artificial bee colony algorithm and used it to solve the problem of scheduling on a single machine [105]. X. Li and Guangfei Yang introduced artificial bee colony algorithm with memory [106]. This algorithm memorizes the previous successful foraging experiences. U. Saif et al.[107] proposed a new hybrid of ABC for assembly line balancing with task time variations and named it Hybrid Pareto.
artificial bee colony algorithm. This algorithm make use of Pareto concepts, used diverse neighbours of food sources for each employee bee and also make use of crossover and mutation operation in its structure. C. Caraveo et al. [108] anticipated a new bee colony algorithm with fuzzy dynamic parameter adaptation and applied for optimization of fuzzy controller design. L. Lv et al. [109] developed an artificial bee colony algorithm with accelerating convergence.

S. Kumar et al. [110] modified the onlooker bee phase of ABC algorithm in order to improve performance for function optimization and also introduced improved memetic search in ABC [110]. D. Karaboga and Selcuk Aslan [111] proposed a new emigrant creation strategy for parallel artificial bee colony algorithm. This algorithm presents a new technique for increasing the quality of the distributed source by combining best solutions. C. Ozturk et al. [112] proposed a genetic operator based binary artificial bee colony algorithm. This modification does not follow classical neighbourhood search method. It arbitrarily select two food sources and employ two point crossover between them then select one of them after swapping. S. Kumar et al. introduced two new strategy in basic ABC algorithm by enhancing local search strategy [113] and randomised memetic search [114]. Both algorithms improve balance between exploration and exploitation of local search space. Randomized memetic ABC introduced a new parameter in ABC algorithm. P. K. Singhal et al. [115] introduces a new strategy in ABC algorithm for unit commitment problem which is based on measure of dissimilarity between binary solutions. A crossover based ABC algorithm proposed by S. Kumar et al. [116]. This algorithm combine crossover operator from genetic algorithm into ABC algorithm. S. Pandey and S. Kumar introduced an enhanced version of ABC algorithm [117] and deployed it to get solution of travelling salesman problem. List of new variants of ABC algorithm is endless there are a large number of research paper based on ABC algorithm exits in literature.

2.4 Animal Migration Optimization Algorithm

Animal Migration Optimization (AMO) algorithm is new swarm intelligence algorithm developed by X. Li et al.[118]. It is nature inspired algorithm that mimics that intelligent behavior of animals groups, like birds, mammals, fish, reptiles, amphibians, insects etc. while migrating from one place to other place in search of quality food sources, secure place for habitat and secure place for mating. Animal Migration Optimization algorithm is stimulated by intelligent conduct of animal while migrating from one place to other place in search of food and secure habitat. This algorithm simulate the behavior of animal that how they shift from existing
location to fresh location and how a number of animal depart from the group and other connect with the group. There are mainly two phases namely migration and population update. The AMO algorithm divided into two major phases: first is animal migration and second is population update [118].

2.4.1 Animal Migration Process

This phase states that how individual change their position from current position based on three rules.

- An animal budge in the same direction as its neighbors
- Always stay near to its neighbors
- Do not collide with neighbors

The AMO define local proximity of an individual with the help of topological ring. For the sake of straightforwardness, the basic AMO suggest the length of the locality as five for every individual. Here a static topology considered and contains a vector with set of indices. The animal with index i has animal with index i+2, i+1, i, i-1 and i-2 in its proximity. Similarly animal with index 1 has \( NP - 1^{th}, NP^{th}, 1^{st}, 2^{nd}, 3^{rd} \) animal in its proximity. First we select a neighbor randomly and then update is position using equation 2.7.

\[
X_{i,(G+1)} = X_{i,(G)} + \emptyset \times (X_{\text{neighbourhood},(G)} - X_{i,(G)}) \tag{2.7}
\]

Where

- \( X_{\text{neighbourhood},(G)} \) is the current position of the neighborhood
- \( \emptyset \) is a uniform random number engendered with the help of Gaussian distribution
- \( X_{i,(G)} \) is the existing location of \( i^{th} \) individual, and
- \( X_{i,(G+1)} \) is the new position of \( i^{th} \) individual

2.4.2 Population update

The second phase of AMO is population update phase. This phase suggest that how animals update their position. This phase also imitates that how some animals go away from the group and some new animals connect with the group. This process simulated with the help of probability Pa. There may be different methods
for probability calculation but it must be function of fitness. The fitness of a function indicates about its quality, fitness calculation must include function value. Here all fitness ranked according to their values with the help of equation 2.8. Animal with highest fitness has highest probability of selection for next iteration. The position updating process illustrated in Algorithm 2.3 [118].

**Algorithm 2.3 Population Update**

```plaintext
for i=1 to NP do
  for j=1 to D do
    if rand > Pa then
      \[ X_{i,(G+1)} = X_{r1,(G)} + 0 \times (X_{best,(G)} - X_{i,(G)}) + 0 \times (X_{r2,(G)} - X_{i,(G)}) ; \]
    end if
  end for
end for
```

In this algorithm 2.3 \( \emptyset \) is a uniformly generated arbitrary number. \( r_1, r_2 \in [1, 2, ..., NP] \) such that \( r_1 \neq r_2 \neq i \). \( Pa \) decided by fitness of individual as shown in equation 2.8.

\[
  p_{ai} = \frac{\text{FitnessRank}_i}{NP},
\]  

(2.8)

Where \( \text{FitnessRank}_i \) is rank of \( i^{th} \) solution in terms of fitness and \( NP \) is size of population. A greedy selection approach used to select next generation solution between \( X_{i,(G)} \) \( X_{i,(G+1)} \). Solution with higher fitness promoted for next iteration.

### 2.4.3 Recent Modifications in Animal Migration Optimization Algorithm

Animal Migration Optimization Algorithm is comparatively young swarm intelligence based algorithm. It was proposed in year 2014 by X. Li et al. [118]. The key scheme is implemented by means of concentric zones around each animal. There is very less work done on modifications and performance issues in this algorithm.

Recently Y Zhou et al. proposed animal migration optimization algorithm for constrained engineering optimization problems [119]. It presents a fast convergence animal migration optimization (FCAMO) algorithm to improve standard AMO of the convergence speed and precision, whose key reduce the search space dynamically, which significantly improves the original AMO in solving complex constraint optimization problems. FCAMO was verified using constraint engineering design problems. Lei Hou et al. [120] anticipated an information entropy-based animal migration optimization algorithm and for Data Clustering. The modified AMO algorithm with an entropy-based heuristic strategy used for data clustering.
It calculates the information entropy of each attribute for a given data set and propose an adaptive strategy that can automatically balance convergence speed and global search efforts according to its entropy in both migration and updating steps. Mingzhi Ma et al. [121] improved AMO for Clustering Analysis. Yi Cao in 2013 author present an opposition based animal migration optimization algorithm [122]. For population initialization it use opposition based learning algorithm. It enlarge the search space, accelerate convergence rate and improve search ability.
2.5 Applications of PSO, ABC and AMO

Swarm intelligence based algorithms has wide range of applications in real world. All these algorithm has applications in engineering, management and science steam like in image segmentation, path optimization, optimum solution for scheduling etc. Some real world problems that are not tractable by classical optimization methods are easily solved by using PSO, ABC and AMO algorithms as these algorithms has less number of parameters and applicable for uni-model problem as well as multimodel problems. Non differential problems are also cracked by these algorithms.

Particle swarm optimization is one of the most popular population based stochastic optimization technique [8]. Particle swarm optimization (PSO) is efficient, robust and simple optimization algorithm. Its applications are at large extent. PSO helps in studying particle trajectory as well as mechanical systems. Likewise it also help in parameter identification of scott-russell amplifying mechanism, and optimizing its amplifying mechanism [123]. It has application in optimizing power system for effective power flow with minimal losses as well in solving Heating System Planning Problem. Help in optimization for the placement and orientation of phase array radar systems. PSO has plenty of applications in railway domain like scheduling trains, active controls on trains movement, as well network layout and planning. Image processing applications utilizes discrete wavelet transformation (DWT) optimization technique with integration of PSO for watermark embedding and extraction for image processing [124].

Since its origination, the ABC algorithm has become extraordinarily fashionable as it has less number of control parameters, it is robust and easy to apply. It is successfully applied to the problems from different application areas by many researchers. Applications of ABC algorithms listed by [37] with subject area of use. The ABC algorithm has application in field of computer science, electrical engineering, mechanical engineering and electronics engineering. Artificial bee colony algorithm applied for feature selection, multi-cast routing, single machine scheduling, balancing in assembly line, designing of controller and to find solution of travelling salesman problem. List of applications of ABC algorithm are endless, there are almost every field including engineering, science and management where ABC algorithm is in use with priority over other competitive optimization algorithm. In [125], D. karaboga presented an extended version of ABC for constrained optimization problems and applied it to train neural networks [126] and clustering problems [127] and to solve TSP problems [128]. Banharnsakun et al. [129] applied the Artificial Bee Colony algorithm on Distributed Environ-
Comparative performance analysis of Artificial Bee Colony algorithm for automatic voltage regulator (AVR) system are carried out by Gozd et al. [130].

In [131], A. Singh applied the ABC algorithm on the leaf-constrained minimum spanning tree (LCMST) problem (ABC - LCMST). Further he compared the approach with ACO and GA [131]. In [131], comparison of ABC-LCMST is carried out in terms of the best, average objective function value and computational time. It was observed that ABC-LCMST outperforms the other approaches. Mala et al. [132] applied the ABC optimization-based approach in automated software test optimization framework. In [133], Dahiya et al. show the application of artificial bee colony algorithm in Software Testing. Rao et al. [134] used the artificial bee colony algorithm to solve network reconfiguration problem in a radial distribution system. Furthermore, they compared, the results obtained by ABC algorithm with the other methods and found that ABC outperformed in terms of quality of the objective value and computational efficiency. Nguyen and Nguyen used the ABC for determining the sectionalizing switch to be operated in order to solve the distribution system loss minimization problem [135]. D. Karaboga applied the artificial bee colony algorithm in the signal processing area for designing digital IIR filters [136]. M. Horng and T. Jiang [137] solved the Image vector quantization problem via honey bee mating optimization strategy.

Animal Migration Optimization Algorithm is comparatively new optimization technique that help in optimization of solutions of complex optimization problems. This algorithm is inspired by major animal groups, such as birds, mammals, fish, reptiles, amphibians, insects, and crustaceans. AMO helps in optimization of Robotic control parameters, manipulators as well arms, motion planning and control, it also help in odour source localization, transport robots, and unsupervised robotic learning [118]. Other application of AMO are in design and modelling, electromagnetic case, induction heating cooker design, as well VLSI design, and also in area of power system, like, RF circuit synthesis [118]. AMO helps in various optimization process in field of electrical engineering, also in the field of automobile optimization of internal combustion engines.

The list of applications of PSO, ABC and AMO is endless. It is observed that these algorithms are useful in almost every field of optimization.