Aswan University Journal of Science and Technology Volume 1, issue 1

Nature-Inspired Algorithms Applications to Power System Optimization

Tomonobu Senjyu[1], Salem Alkhalaf[2], Al-Attar Mohamed[3]
[1] Graduate School of Science and Engineering, University of the Ryukyus, Okinawa 903-0213, Japan
[2] Department of Computer, College of Science and Arts in Ar-Rass, Qassim University, Ar Rass, Saudi Arabia
[3] Electrical Engineering Department, Faculty of Engineering, Aswan University, Aswan, Egypt,
s.alkhalaf@qu.edu.sa

Abstract

Nature has been evolving for several hundred million years, and it has found various ingenious solutions to problem-solving and adaption to ever-changing environments. From Darwinian evolution point of view, survival of the fittest will result in the variations and success of species, which can survive and optimally adapt to environments.

Nature-inspired algorithms are still at a very early stage with a relatively short history, it has come up as a new era in comparing with many traditional methods; however, nature-inspire algorithms have already shown their great potential, flexibility and efficiency with ever-increasing diverse ranges of applications.

Nature is the best tutor and its designs and strengths are extremely massive and strange that it gives inspiration to us to imitate nature to solve hard and complex problems in power system. Different Algorithms are presented for solving power system optimization problems.

I OPTIMIZATION

Optimization is the act of obtaining the best result under given circumstances. In design, construction, and maintenance of any engineering system, engineers have to take many technological and managerial decisions at several stages. The ultimate goal of all such decisions is either to minimize the effort required or to maximize the desired benefit. Since the effort required or the benefit desired in any practical situation can be expressed as a function of certain decision variables, optimization can be defined as the process of finding the conditions that give the maximum or minimum value of a function [1].

Mathematically speaking, it is possible to write most optimization problems in the generic form:

$$\text{minimize}_{x \in \mathbb{R}^n} \quad f_i(x) \qquad (i = 1,2,\ldots,M) \tag{1}$$

$$\text{subject to} \quad h_j(x) = 0 \qquad (j = 1,2,\ldots,J) \tag{2}$$

$$g_k(x) \leq 0 \qquad (k = 1,2,\ldots,K) \tag{3}$$

where $f_i(x)$, $h_j(x)$ and $g_k(x)$ are functions of the design vector $x = (x1, x2, \ldots, xn)^T$ Here the components xi of x are called design or decision variables, and they can be real continuous, discrete or the mixed of these two. The functions $f_i(x)$ where i= 1,2,...,M are called the objective functions or simply cost functions, and in the case of M = 1, there is only a single objective. The space spanned by the decision variables is called the design space or search space $\mathbb{R}^n$, while the space formed by the objectivefunction values is called the solution space or response space. The equalities for hj and inequalities for g_k are called constraints. It is worth pointing out that we can also write the inequalities in the other way≥ 0, and we can also formulate the objectives as a maximization problem [22].

Global Optimization

The goal of global optimization is to find the best possible elements x* from a set X according to a set of criteria F = {f1, f2, .., fn}. These criteria are expressed as mathematical functions, the so-called objective functions. A mathematical optimization model consists mainly three basic sets of elements [6,7,8]:

- Objective function: quantity to be optimized (minimized or maximized).
- Variables: inputs to the objective function.
- Constraints: limitations assigned to the variables.

Classification of Optimization

Optimization algorithms divides into six categories. For instance, a dynamic optimization problem could be either constrained or unconstrained. In addition some of the variables may be discrete and others continuous [4]. Six categories of optimization is given in Fig. 1.

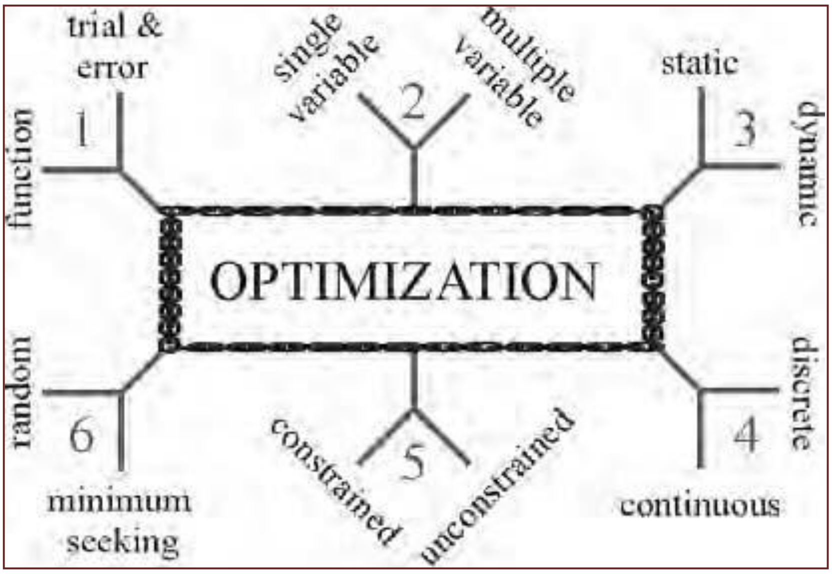


Fig. 1. Six categories of optimization algorithms.

There is no single method available for solving all optimization problems efficiently. Hence the optimum seeking methods are also known as mathematical programming techniques as a part of operations research. Operations research is concerned with the application of scientific techniques to decision making problems and with establishing optimal solutions. Mathematical programming techniques are useful in finding the minimum of a function of several variables under a prescribed set of constraints. Stochastic process techniques can be used to analyze problems described by a set of random variables having known probability distributions. Statistical methods enable one to analyze the experimental data and build empirical models to obtain the most accurate representation of the physical situation [1]. Applied methods of operation research is given in table 1.

TABLE 1 Methods of Operations Research

Mathematical Programming Techniques	Stochastic Process Techniques	Statistical Methods
Calculus methods Calculus of variations Nonlinear programming Geometric programming Quadratic programming Linear programming Dynamic programming Integer programming Stochastic programming Separable programming Multiobjective programming Network methods: CPM and PERT Game theory Simulated annealing Genetic algorithms Neural networks	Statistical decision theory Markov processes Queueing theory Renewal theory Simulation methods Reliability theory	Regression analysis Cluster analysis, pattern recognition Design of experiments Discriminate analysis (factor analysis)

Generally, optimization algorithms can be divided in two basic classes: deterministic and probabilistic algorithms. *Deterministic algorithms* are most often used if a clear relation between the characteristics of the possible solutions and their utility for a given problem exists. But If the relation is not so obvious or too complicated, or the dimensionality of the search space is very high.

Then, *probabilistic algorithms* come into play. This combination is often performed stochastically by utilizing statistics obtained from samples of the search space or based on a model of some natural phenomenon or physical process. An important class of probabilistic, Monte Carlo metaheuristics is Evolutionary Computation [Heitkötter & Beasley, 1998]. It encompasses all algorithms that are based on a set of multiple solution candidates, called population, which are iteratively refined. This field of optimization is also a class of Soft Computing [Zadeh, 1994] as well as a part of the artificial intelligence [Buchanan, 2005] area. Some of its most important members are evolutionary algorithms and Swarm Intelligence. Besides these nature-inspired and evolutionary approaches, there exist also methods that copy physical processes like the Simulated Annealing [6,7]. The taxonomy of global optimization algorithms is given in Fig. 2.

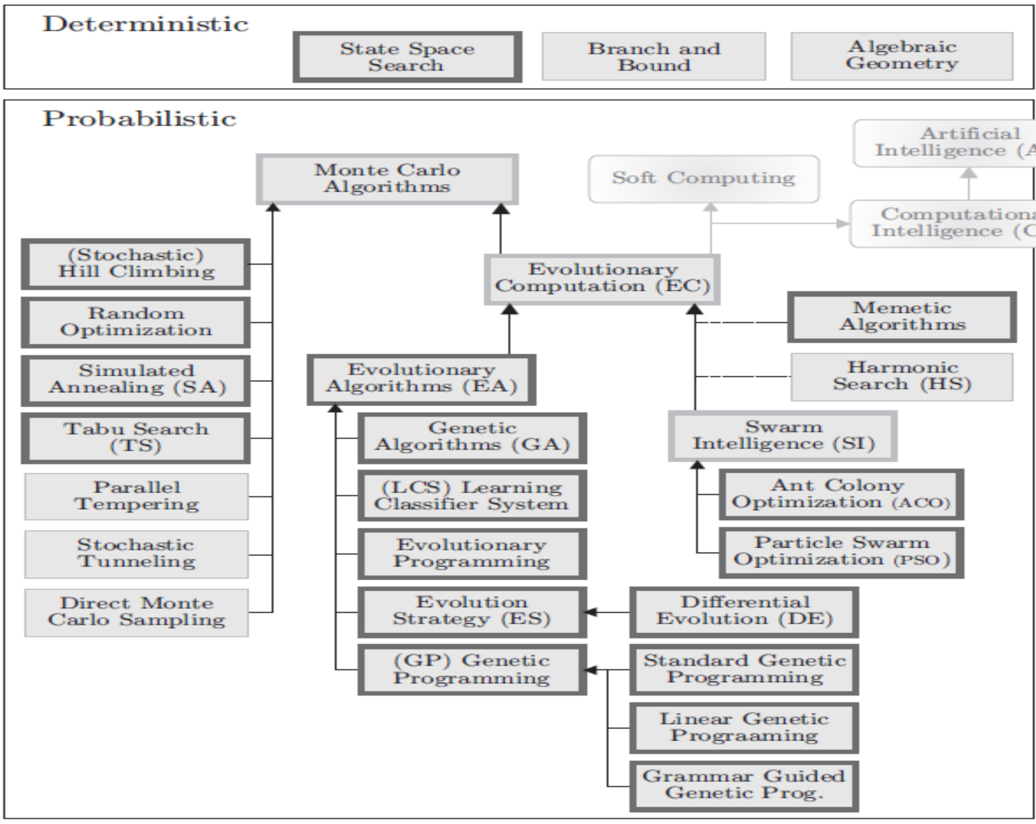


Fig. 2 The taxonomy of global optimization algorithms

II. OPTIMIZATION IN POWER SYSTEM

The electric power generation are more than a century old technology but each element of the system contains high-tech solutions (power plant technology, generators, transformers, power lines, power electronic devices, Supervisory Control and Data Acquisition, etc.) The controlled elements are several millions so the system operation, stability, control, balance, optimization, settling is a really complex and distributed task [61].

Power system optimization fields like: economic dispatch and optimal power flow, reactive power and voltage control, optimal design of power system stabilizer, harmonics, state estimation, unit commitment problem, Maximum power point tracking, dynamic security border identification, Regional energy trade, Transmission Network Expansion and Planning, short-term hydrothermal scheduling, security assessment, Maintenance scheduling, Load forecasting and types of power system balance, operation, quality, reliability and stability.

The difficulties associated with using mathematical optimization on large-scale engineering problems have contributed to the development of alternative solutions, during the last two decades; the interest in applying new metaheuristic optimization methods in power system field has grown rapidly.

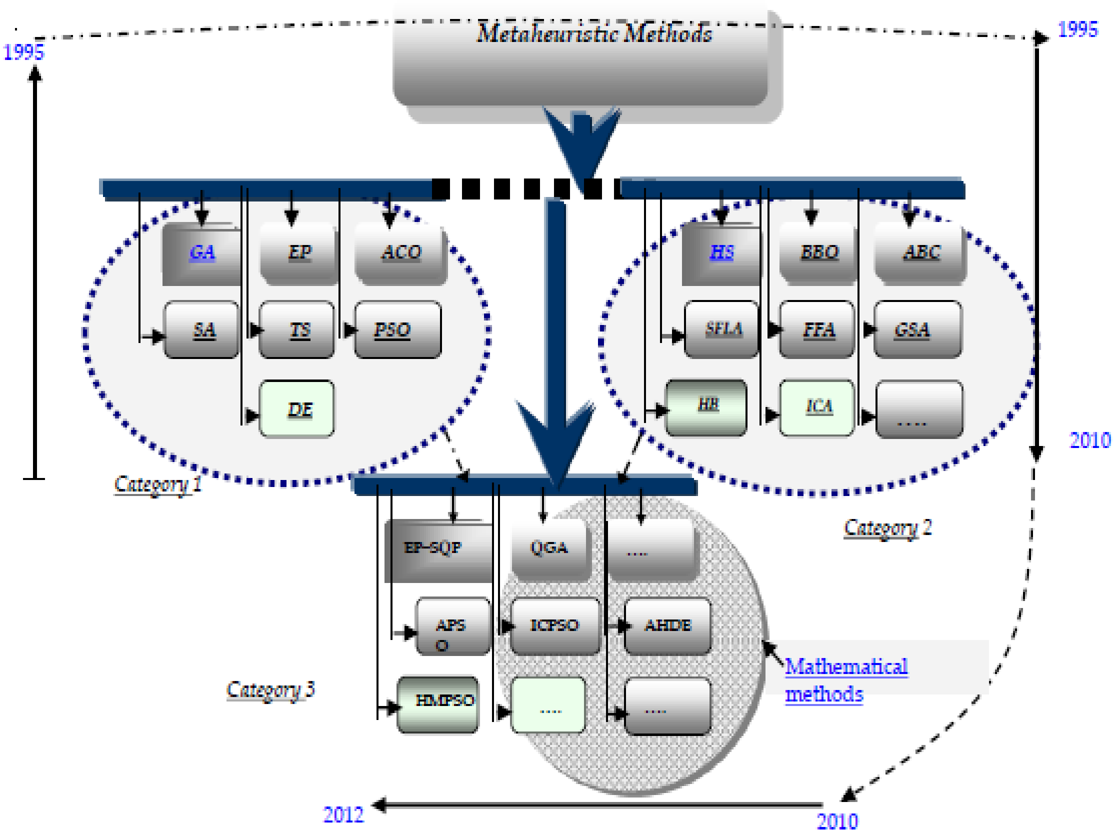


Fig. 3 Presentation of metaheuristic optimization methods[62]

Generally, metaheuristic optimization methods classified into three categories[62] as given in Fig. 3.

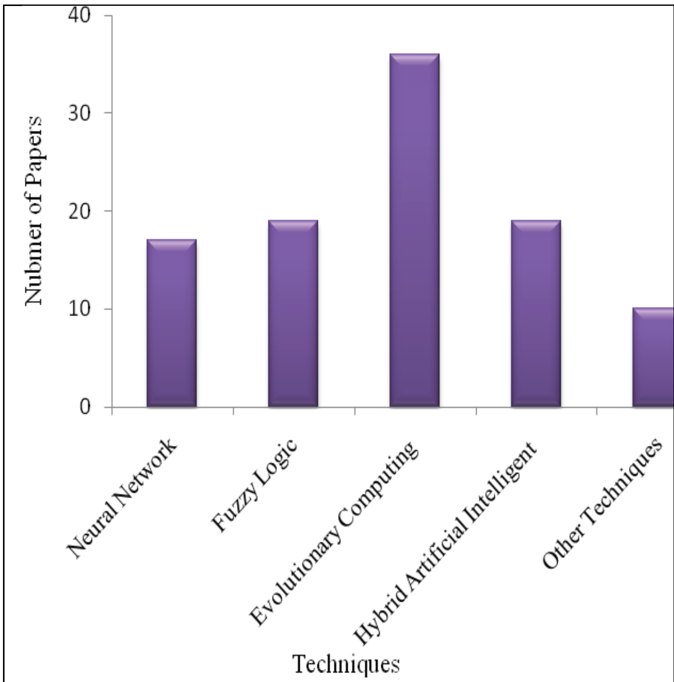
1. The first category such as: GA, TS, SA, ACO, PSO, DE and theirs successive developed variants.

2. The second category includes: Harmony search (HS), Biogeography based optimization

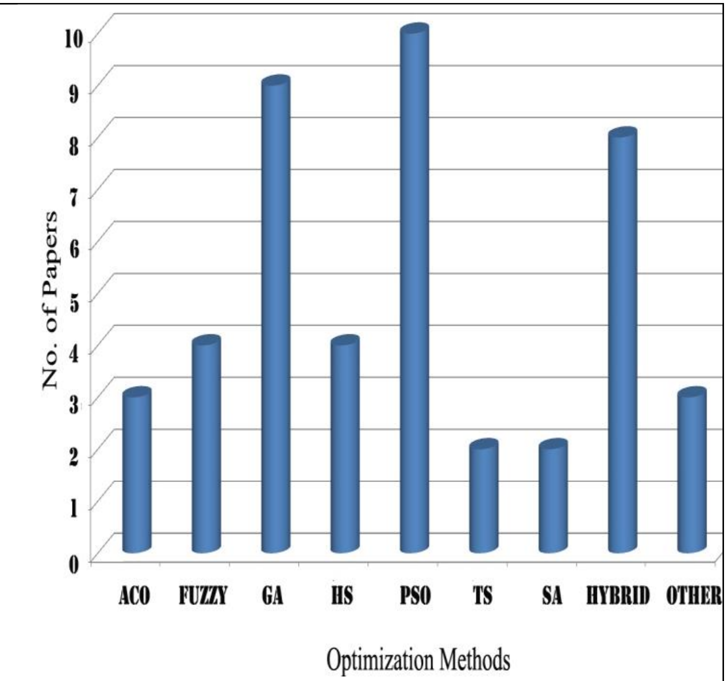
(BBO), Artificial bee colony (ABC), Honey bee (HB), Shuffled frog leaping algorithm (SFLA), Firefly algorithm (FFA), Gravitational search algorithm (GSA), Imperialist competition algorithm (ICA) and many other variants.

3. The third category called hybrid optimization methods includes a combination between metaheuristics and conventional methods.

Details about their performances and the application of metaheuristic optimization techniques for solving many practical problem related to power system field particularly multi ob-jective OPF can be found in a recent state of the art of non-deterministic optimization and hybrid methods presented in [63]. Papers published IEEE/IET/Elsevier/Springer Databases (1995-2010) in Power system stabilizers, and papers published IEEE/IET/Elsevier/Springer Databases (2001-2011) in optimal capacitor placement is given in Fig. 4.



papers published IEEE/IET/Elsevier/Springer Databases (1995-2010) in Power system stabilizers [64]



papers published IEEE/IET/Elsevier/Springer Databases (2001-2011) in optimal capacitor placement [65]

Fig. 4 published papers in PSS and optimal capacitor placement

IV. NATURE-INSPIRED

During the last decade nature inspired intelligence becomes increasingly popular through the development and utilization of intelligent paradigms in advanced systems design. Among the most popular nature inspired approaches, when the task is optimization within complex domains, are those methods representing successful animal and micro-organism team behaviour, such as swarm or flocking intelligence, artificial immune systems, ant colonies, or optimized performance of bees, etc. The Honey Bees Mating Optimization Algorithm was first presented in[9,10], and since then it was used on a number of different applications[11,12,14].

In recent years, the Nature-Inspired Computation (NIC) methods have gained considerable attention from different communities. Compared with conventional optimization schemes, NIC methods offer more suitable candidates for dealing with the demanding problems faced by industry, and can thus offer us robust, competitive solutions. As a matter of fact, for optimization applications NIC methods are capable of outperforming the classical techniques by providing better and more flexible solution choices[8].

Swarm intelligence and bio-inspired algorithms form a hot topic in the developments of new algorithms inspired by nature. These nature-inspired metaheuristic algorithms can be based on *swarm intelligence*, *biological systems*, *physical and chemical systems*, depending on the sources of inspiration. Though not all of them are efficient, a few algorithms have proved to be very efficient and thus have become popular tools for solving real-world problems[38].

A. SWARM INTELLIGENCE

Swarm intelligence (SI) describes the collective behaviour of decentralized, self-organized systems, natural or artificial. The concept is employed in work on artificial intelligence. The expression was introduced by Gerardo Beni and Jing Wang in 1989, in the context of cellular robotic systems [3]. SI systems are typically made up of a population of simple agents or boids interacting locally with one another and with their environment. The agents follow very simple rules, and although there is no centralized control structure dictating how individual agents should behave, local, and to a certain degree random, interactions between such agents lead to the emergence of "intelligent" global behaviour, unknown to the individual agents. Natural examples of SI include ant colonies, bird flocking, animal herding, bacterial growth, and fish schooling. The application of swarm principles to robots is called swarm robotics, while 'swarm intelligence' refers to the more general set of algorithms. 'Swarm prediction' has been used in the context of forecasting problems [2].

Swarm describes behaviour of an aggregate of animals of similar size and body orientation, often moving en masse or migrating in the same direction. Swarming is a general term that can be applied to any animal that swarms. The term is applied particularly to insects, but can also be applied to birds, fish, various microorganisms such as bacteria, and people. The term flocking is usually used to refer to swarming behaviour in birds, while the terms shoaling or schooling are used to refer to swarming behaviour in fish. The swarm size is a major parameter of a swarm.

1. Properties Of Swarm Intelligence

The typical swarm intelligence system has the following properties [48]:

- It is composed of many individuals
- The individuals are relatively homogeneous
- The interactions among the individuals are based on simple behavioral rules that exploit only local information that the individuals exchange directly or via the environment
- The overall behaviour of the system results from the interactions of individuals with each other and with their environment, that is, the group behaviour self-organizes.

2. Modelling Swarm Behaviour

The simplest mathematical models of animal swarms generally represent individual animals as following three rules [2,49]:

1. Move in the same direction as your neighbour
2. Remain close to your neighbours
3. Avoid collisions with your neighbours

Many current models use variations on these rules, often implementing them by means of concentric "zones" around each animal. In the zone of repulsion, very close to the animal, the focal animal will seek to distance itself from its neighbours to avoid collision. Slightly further away, in the zone of alignment, the focal animal will seek to align its direction of motion with its neighbours. In the outermost zone of attraction, which extends as far away from the focal animal as it is able to sense, the focal animal will seek to move towards a neighbour[2].

The shape of these zones will necessarily be affected by the sensory capabilities of the given animal. For example the visual field of a bird does not extend behind its body. Fish rely on both vision and on hydrodynamic perceptions relayed through their lateral line, while Antarctic krill rely both on vision and hydrodynamic signals relayed through antennae. Some of the animals that exhibit swarm behaviour are:

Insects – Ants, bees, locusts, termites, mosquitoes and insects migration, Bacteria, Birds, Land animals, Aquatic animals – fish, krill and other aquatic animals and People

3. Example Algorithms of Swarm Intelligence:

- Particle swarm optimization Algorithm
- Ant colony optimization Algorithm
- Honey Bees Optimization Algorithm

- Cuckoo Optimization Algorithm
- Firefly Optimization Algorithm
- Bat Optimization Algorithm
- Bacterial Foraging Optimization Algorithm
- Wolf Pack Search Optimization Algorithm

a. Particle Swarm Optimization(PSO)

Kennedy and Eberhart in 1995 developed the particle swarm optimization (PSO) algorithm by studying social and cognitive behavior of ants[32]. The movement of the particles is influenced by two factors using the global particle-to-particle best solution and the local particle's iteration-to-iteration best solution[21]. As a result of iteration-to-iteration information, the particle stores in its memory the best solution it has visited so far, called "*pbest*", and experiences an attraction towards this solution as it traverses through the solution search space. This attraction is stronger if the best solution is farther from the current particle's location and not related to its performance. As a result of the particle-to-particle information, the particle stores in its memory the best solution visited by any particle, and an attraction towards this solution, called "*gbest*", results as well. The first, *pbest*, and second, *gbest*, factors are called the cognitive and social components, respectively. After each iteration the *pbest* and *gbest* are updated for each particle if better, more dominating solutions (in terms of performance or fitness), is found. This process continues, iteratively, until either the algorithm achieves the desired result, or it's determined that an acceptable solution cannot be found within computational limits determined by the application. These two factors determine the direction and amount of movement resulting from the particle's velocity. Interestingly, the performance of the two solution points does not affect the direction or amount of motion in traditional PSOs but completely controls the choice of the global and local best solution. A modified PSO, Fitness Distance Ratio PSO, incorporates the solution's performance or fitness into the velocity and does have faster convergence to the globally best answer.

The PSO defines each particle in the D-dimensional space as $X_i = (x_{i1}, x_{i2}, ..., x_{iD})$, where the subscript 'i' represents the particle number and the second subscript is the dimension, number of parameters defining the solution. The memory of the previous best position is represented as $P_i = (p_{i1}, p_{i2}, ..., p_{iD})$, and a velocity for each dimension is independently established as $V_i = (v_{i1}, v_{i2}, ..., v_{iD})$. After each iteration, the velocity term is updated, and the particle is moved with some randomness in the direction of its own best position, *pbest*, and the global best position, *gbest*. This is apparent in the velocity update equation, given by[34]

$$V_{id}^{(t+1)} = \begin{array}{l}\omega \times V_{id}^{(t)} + U[0,1] \times \psi_1 \times (p_{id}^{(t)} - x_{id}^{(t)}) + \\ U[0,1] \times \psi_2 \times (p_{gd}^{(t)} - x_{id}^{(t)})\end{array} \tag{4}$$

The position is updated using this velocity and

$$X_{id}^{(t+1)} = X_{id}^{(t)} + V_{id}^{(t+1)} \tag{5}$$

where U [0,1] samples a uniform random distribution, t is a relative time index, ψ_1 and ψ_2 are weights trading off the impact of the local best and global best solutions' on the particle's total velocity. The particle swarm optimization algorithm is highly efficient in searching complex, continuous solution landscapes. The particle swarm can also be implemented as a parallel algorithm improving its efficiency for real-time applications. The particles can be split up among multiple processors and then the global best solution is shared among the particles. The flowchart of PSO optimization is given in Fig. 5.

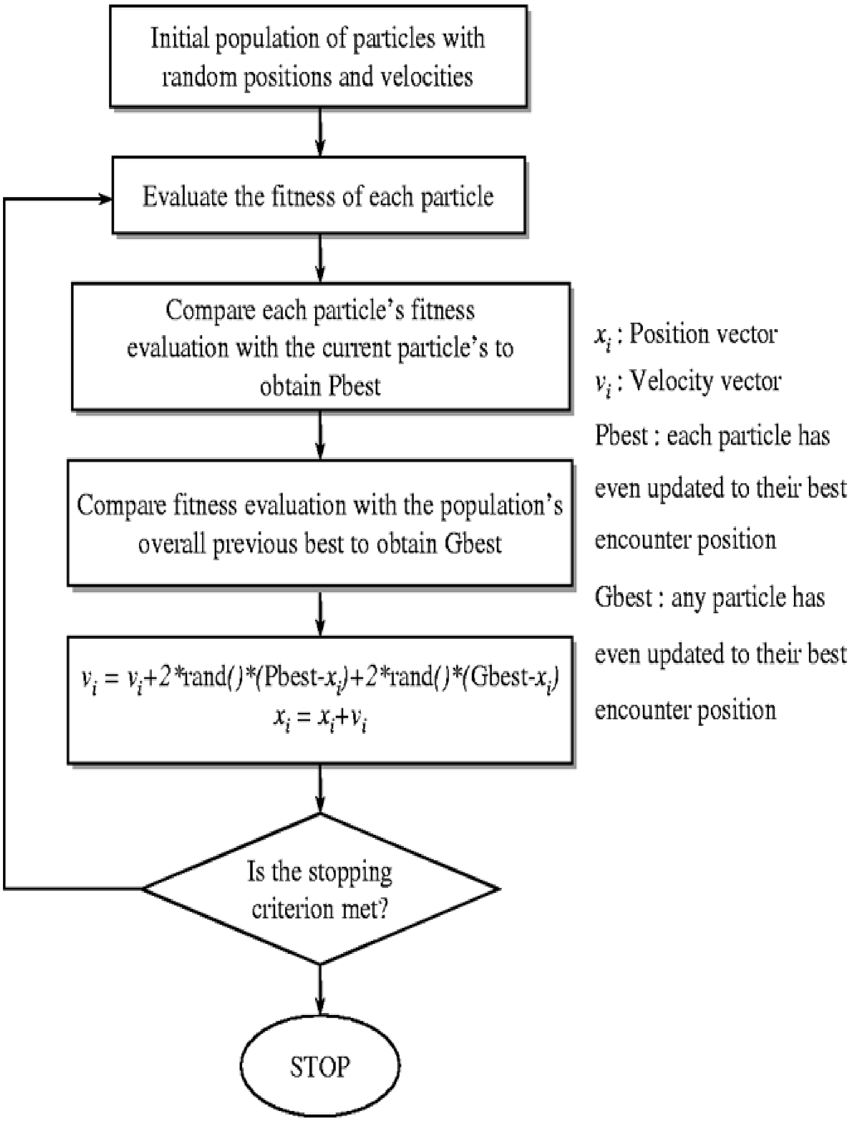


Fig. 5 Flowchart of PSO [33]

b. Ant Colony Optimization Algorithm

The Conventional Ant Colony Optimization (ACO) algorithm is a meta-heuristic that has a combination of distributed computation, *autocatalysis* (positive feedback), and constructive greediness to find an optimal solution for combinatorial optimization problems. This algorithm tries to mimic the ant's behavior in the real world. Since its introduction, the ACO algorithm has received much attention and has been incorporated in many optimization problems, namely the network routing, traveling salesman, quadratic assignment, and resource allocation problems [18].

The ACO algorithm has been inspired by the experiments run by Goss et al. [19] using a colony of real ants. They observed that real ants were able to select the shortest path between their nest and food resource, in the existence of alternate paths between the two. The search is made possible by an indirect communication known as *stigmergy* amongst the ants. While traveling their way, ants deposit a chemical substance, called *pheromone*, on the ground. When they arrive at a decision point, they make a probabilistic choice, biased by the intensity of pheromone they smell. This behavior has an autocatalytic effect because of the very fact that an ant choosing a path will increase the probability that the corresponding path will be chosen again by other ants in the future. When they return back, the probability of choosing the same path is higher (due to the increase of pheromone). New pheromone will be released on the chosen path, which makes it more attractive for future ants. Shortly, all ants will select the shortest path.

In this case, because of the same pheromone laying mechanism, the shortest branch is most often selected. The first ants to arrive at the food source are those that took the two shortest branches. When these ants start their return trip, more pheromone is present on the short branch than the one on the long branch. This will stimulate successive ants to choose the short branch. Although a single ant is in principle capable of building a solution (i.e., of finding a path between nest and food resource), it is only the colony of ants that presents the "shortest path finding" behaviour. In a sense, this behaviour is an emergent property of the ant colony.

This behaviour was formulated as Ant System (AS) by Dorigo et al. [18]. Based on the AS algorithm, the Ant Colony Optimization (ACO) algorithm was proposed [20]. In ACO algorithm, the optimization problem is formulated as a graph $G = (C; L)$, where C is the set of components of the problem, and L is the set of possible connections or transitions among the elements of C. The solution is expressed in terms of feasible paths on the graph G, with respect to a set of given constraints. The population of agents (ants) collectively solves the problem under consideration using the graph representation. Though each ant is capable of finding a (probably poor) solution, good quality solutions can emerge as a result of collective interaction amongst ants. Pheromone trails encode a long-term memory about the whole ant search process. Its value depends on the problem representation and the optimization objective. A general outline of the ACO algorithm is presented in Figure 6 [20].

```
Algorithm ACO meta heuristic();
    while (termination criterion not satisfied)
        ant generation and activity();
        pheromone evaporation();
        daemon actions(); "optional"
    end while
end Algorithm
```

Fig 6 Ant Colony Algorithm.

Informally, the behaviour of ants in ACO algorithm can be summarized as follows. A colony of ants concurrently and asynchronously moves through adjacent states of the problem by moving through neighbour nodes of *G*. They move by applying a stochastic local decision policy which makes use of the information contained in the local node and ant's routing table. By moving, ants incrementally build solutions to the optimization problem. When the solution is being built, every ant evaluates the solution and puts the information about its goodness on the pheromone trails of the connection used. This pheromone information will direct the search of future ants, until a feasible solution is found.

The ants in ACO algorithm have the following properties [18]:

1. Each ant searches for a minimum cost feasible partial solution.
2. An ant k has a memory M^k that it can use to store information on the path it followed so far. The stored information can be used to build feasible solutions, evaluate solutions and retrace the path backward.
3. An ant k can be assigned a start state s^k_s and more than one termination conditions e^k.
4. Ants start from a start state and move to feasible neighbour states, building the solution in an incremental way. The procedure stops when at least one termination condition e^k for ant k is satisfied.
5. An ant k located in node i can move to node j chosen in a feasible neighbourhood N^k_i through probabilistic decision rules. This can be formulated as follows:

An ant k in state $sr =< s_{r-1}; i >$ can move to any node j in its feasible neighbourhood N^k_i , defined as $N^k_i = \{j \mid (j \in Ni) \wedge (< sr, j > \in S)\}\ sr \in S$, with S is a set of all states.

6. A probabilistic rule is a function of the following.
   a) The values stored in a node local data structure $Ai = [a_{ij}]$ called *ant routing table* obtained from pheromone trails and heuristic values,
   b) The ant's own memory from previous iteration, and

c) The problem constraints.

7. When moving from node i to neighbour node j, the ant can update the pheromone trails τ_{ij} on the edge (i, j).

8. Once it has built a solution, an ant can retrace the same path backward, update the pheromone trails and die.

c. Honey Bees Optimization Algorithm

The Honey Bees Mating Optimization algorithm simulates the mating process of the queen of the hive. The mating process of the queen begins when the queen flights away from the nest performing the mating flight during which the drones follow the queen and mate with her in the air [9,11]. The Honey Bees Mating Optimization (HBMO) can be used in order to give very good results in the Vehicle Routing Problem (VRP) and in the Travelling Salesman Problem (TSP). The *Travelling Salesman Problem (TSP)* is the problem of finding the shortest tour through all the cities that a salesman has to visit. The TSP is probably the most famous and extensively studied problem in the field of Combinatorial Optimization [13,15]. The *Vehicle Routing Problem (VRP)* is often described as the problem in which vehicles based on a central depot are required to visit geographically dispersed customers in order to fulfil known customer demands. The problem is to construct a low cost, feasible set of routes - one for each vehicle. A route is a sequence of locations that a vehicle must visit along with the indication of the service it provides. The vehicle must start and finish its tour at the depot [16].

Honey-Bees Modelling

The mating-flight may be considered as a set of transitions in possible solutions where the queen moves between the different states in some speed and mates with the drone encountered at each state probabilistically. At the beginning of flight, each queen is initialized by an amount of energy and if this amount reaches a threshold or Zero, or even spermatheca has been filled, the queens will return to the nest. In this algorithm, workers' task is watching broods. In developed algorithm, workers are implemented as heuristic functions which cause fitness of broods to be increased. A drone mates with a queen probabilistically using below function as [35,36]:

$$Prob(Q, D) = e^{-\frac{\Delta(f)}{S(t)}} \qquad (6)$$

where Prob(Q, D) is the probability of adding the sperm of drone D to the spermatheca of queen Q (that is, the probability of a successful mating); Δ(f) is the absolute difference between the fitness of D (i.e. f(D)) and the fitness of Q (i.e. f(Q)); and S(t) is the speed of the queen at time t. It is apparent that this function acts as an annealing function, where the probability of mating is high when either the queen is still in the start of her mating-flight and therefore her speed is high, or when the fitness of the drone is as good as the queen's. After each transition in space, the queen's speed, S(t), and energy, E(t), decay using the following equations:

$$S(t+1) = \alpha \times S(t) \quad (7)$$

$$E(t+1) = E(t) - \gamma \quad (8)$$

where α is a factor ϵ [0, 1] and γ is the amount of energy reduction after each transition. Thus, Honey Bees Mating Optimization (HBMO) algorithm may be constructed with the following five main stages[37]:

1) *The algorithm starts with the mating-flight, where a queen (best solution) selects drones probabilistically to form the spermatheca (list of drones). A drone is then selected from the list at random for the creation of broods.*
2) *Creation of new broods (trial solutions) by crossoverring the drones' genotypes with the queen's.*
3) *Use of workers (heuristics) to conduct local search on broods (trial solutions).*
4) *Adaptation of workers' fitness based on the amount of improvement achieved on broods.*
5) *Replacement of weaker queens by fitter broods.*

d. Cuckoo Optimization Algorithm

a new meta-heuristic algorithm, called Cuckoo Search (CS), for solving optimization problems. This algorithm is based on the obligate brood parasitic behaviour of some cuckoo species in combination with the L´evy flight behaviour of some birds and fruit flies[23]. The pseudo code of cuckoo search is given in Fig. 7.

Cuckoo Search via L´evy Flights
begin
Objective function f(x), x = (x1, ..., xd)T
Generate initial population of
n *host nests* xi (i = 1, 2, ..., n)
while *(*t *<MaxGeneration) or (stop criterion)*
Get a cuckoo randomly by L´evy flights
evaluate its quality/fitness Fi
Choose a nest among n *(say,* j*) randomly*
if *(*Fi > Fj*),*
replace j *by the new solution;*
end
A fraction (pa) *of 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
Postprocess results and visualization
end

Fig 7 Pseudo code of the Cuckoo Search (CS).

For simplicity in describing Cuckoo Search, use the following three idealized rules: 1) Each cuckoo lays one egg at a time, and dump its egg in randomly chosen nest; 2) The best nests with high quality of eggs will carry over to the next gener- ations; 3) The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with pa 2 [0, 1] a probability [24].

e. Firefly Optimization Algorithm

Following three idealized rules: 1) all fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex; 2) Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less brighter one will move towards the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If there is no brighter one than a particular firefly، it will move randomly; 3) The brightness of a firefly is affected or determined by the landscape of the objective function[25,26]. The matlab code is given in Fig. 8.

Objective function f(x), x = $(x1, \dots, xd)^T$
Generate initial population of fireflies xi (i = 1, 2, ..., n)
Light intensity Ii at xi is determined by f(xi)
Define light absorption coefficient γ
while *(t <MaxGeneration)*
for *i = 1 : n all n fireflies*
for *j = 1 : i all n fireflies*
if *(Ij > Ii), Move firefly i towards j in d-dimension;* ***end if***
Attractiveness varies with distance r via exp[−γr]
Evaluate new solutions and update light intensity
end for *j*
end for *i*
Rank the fireflies and find the current best
end while
Postprocess results and visualization

Fig. 8 Pseudo code of the firefly algorithm [25].

f. Bat Optimization Algorithm

Bat-inspired algorithm is a metaheuristic optimization algorithm developed by Xin-She Yang in 2010[27]. This bat algorithm is based on the echolocation behaviour of microbats with varying pulse rates of emission and loudness[28,29]. The idealization of the echolocation of microbats can be summarized as follows: Each virtual bat flies randomly with a velocity v_i at position (solution) x_i with a varying frequency or wavelength and loudness A_i. As it searches and finds its prey, it changes frequency, loudness and pulse emission rate r. Search is intensified by a local random walk. Selection of the best continues until certain stop criteria are met. This essentially uses a frequency-tuning technique to control the dynamic behaviour of a swarm of bats, and the balance between exploration and exploitation can be controlled by tuning algorithm-dependent parameters in bat algorithm[30,31].

g. Bacterial Foraging

Bacteria Foraging Optimization (BFO) algorithm is a new class of biologically inspired stochastic global search technique based on mimicking the foraging (methods for locating, handling, and ingesting food) behavior of (E. Coli) bacteria. During foraging, a bacterium can exhibit two different actions: tumbling or swimming. The tumble action modifies the orientation of the bacterium. During swimming (chemotactic step), the bacterium will move in its current direction. Chemotactic movement is continued until a bacterium goes in the direction of positive-nutrient gradient. After a certain number of complete swims, the best half of the population undergoes reproduction, eliminating the rest of the population. In order to escape local optima, an elimination-dispersion event is carried out where some bacteria are liquidated at random with a very small probability and the new replacements are initialized at random locations of the search space [51,52]. The flowchart is given in Fig. 9.

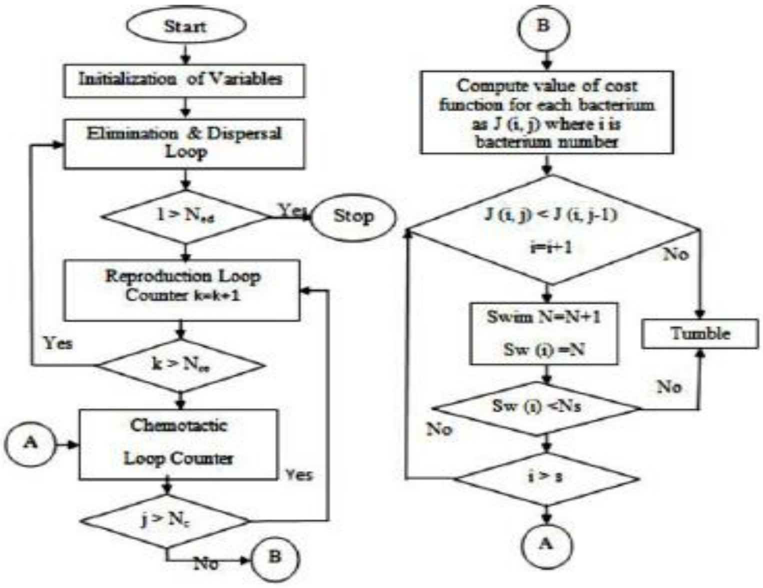


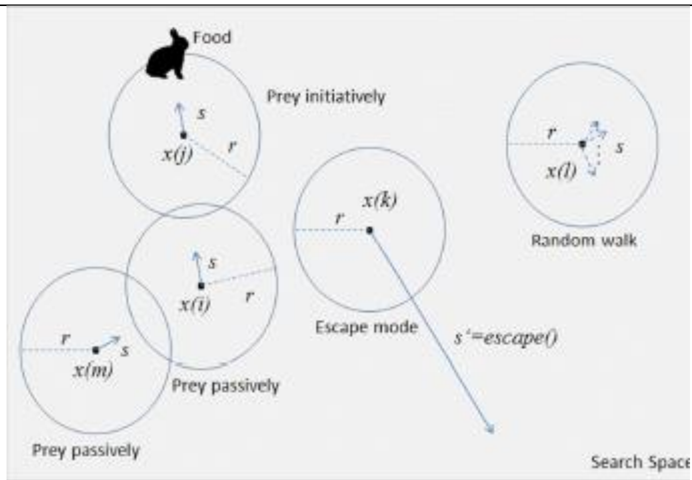
Fig. 9 Flowchart of the Bacterial Foraging Algorithm

h. Wolf Search

Most of the time, young wolves (not pups and old ones) instinctively separate from their initial packs in order to reproduction and seek their related pair and territory.As soon as two alone wolves find each other, they move together and start to seek the territory. This correlation between two wolves will continue until one of them dies. The emerging theory suggests that the wolves group would work i.e. group concentration is usually more successful in the reproduction than hunting. The packs are managed by two wolves that have higher social position and practically more freedom in comparison to the other wolves of the pack. These two wolves (two first standing among the population) that are called alpha gain more food and also have exclusive lows for mating. Most of the time the group chief (the best members) mate with each other, but in case of losing (death or injury) its counterpart mate, alpha wolf can also mate with one of the lower rank wolves. Even losing a sibling mate does not influence the chief and the alone wolf finds another mate for itself quickly.

Usually alpha pair is successful in raising its pups. The other wolves of the pack can mate but when they're in lack of resources such as food and time, the existing resources are devoted to the alpha pair children. The third wolf after the alpha pair is called Beta that more cares the alpha pair's children in comparison to the other wolves. Also Beta wolf wants to obtain mastership position from the alpha wolf, but some of them, depending on the condition, prefer to hold the same third position [68,66] . GWO schematic diagram is given in Fig. 10.

Fig. 10 Grey wolf optimization[69]



illustrates the wolves' movement. $x(i)$ is in range with $x(j)$ and $x(m)$. $x(i)$, however, is moving toward $x(j)$because it has a better terrain or is closer to the food (as represented by the better fitness function value). $x(m)$ will follow $x(i)$ and $x(k)$ will leap a large distance away when the threat alert is trigged by the probability. Otherwise, it is probable that $x(k)$ may sense $x(j)$ in its range and merge if $f(x(j))>f(x(k))$. $x(j)$ will ignore its nearby peers and move toward the food, which has the highest fitness value. $x(l)$ is roaming in Brownian motion some distance away[68].

The algorithm is as follows [66]:

1. Generate initial population in the search domain randomly.

2. Evaluate each wolf.
3. Divide the search domain into some territories. (here 2 territories)
4. Randomly distribute the wolfs in the territories.
5. Apply a random move on each wolf in its territory.
6. Evaluate each neighborhood (wolfs in each territory) and determine the two ones with best improvement in comparison to the last generation. Name them Leader1 and Leader2.
7. In each territory Leader1 and Leader2 mate to generate two offspring which are exposed to a local search by the beta wolf. Then these offspring replace the two worst ones in each territory.
8. If Leader1 is changed in the last q iterations, then go back to step 4, else continue to step 9.
9. In the corresponding territory Leader1 is substituted by Leader2 and a random wolf is generated in that territory. Go back to 5
10. If by any chance leader1 and Leader2 are the same, Leader2 is replaced by a wolf of lower rank in order to avoid twins.

4. Advantages Of Swarm Intelligence

There are several advantages of swarm Intelligence. Some are [50]:

• Agents are not goal directed; they react rather than plan extensively.
• Agents are simple, with minimal behaviour and memory.
• Control is decentralized; there is no global information in the system.
• Failure of individual agents is tolerated; emergent behaviour is robust with respect to Individual failure.
• Agents can react to dynamically changing environments.
• Direct agent interaction is not required.

Different types of swarm intelligence algorithms are given in table2.

Table 2 SOME KINDS OF SWARM INTELLIGENCE ALGORITHMS

Algorithm	Author, Reference	Algorithm	Author, Reference
Bacterial-GA Foraging	Tai-Chen Chen, "A novel optimization approach: bacterial-ga foraging", IEEE, 2007.	Glow- worm swarm	Krishnanand, "Glowworm swarm optimisation multi-modal functions" *International Journal of Computational Intelligence* 2009
Good lattice swarm	Shoubao Su, "Goodlattice swarm algorithm for constrained engineering design optimization", IEEE,*WiCom 2007.*	Hier- archical swarm model	Ma, Lianbo, et al. "Discrete and Continuous Optimization Based on Hierarchical Artificial Bee Colony Optimizer." *Journal of Applied Mathematics* 2014
BeeHive	Tam´as Vicsek, " Novel type of phase transition in a system of self-driven particles",*Physical Review Letters*, 5(6):1226–1229,1995.	Krill Herd	Mandal, Barun, "Economic load dispatch using krill herd algorithm" *International Journal of Electrical Power & Energy Systems* (2014): 1-10.

Bumble bees	Lihoreau, "Travel optimization by foraging bumblebees through readjustments of traplines after discovery of new feeding locations." *The American Naturalist* 176.6 (2010): 744-757.	Monkey search	A. Mucherino "Monkey search: a novel metaheuristic search for global optimization", *Data Mining,Systems Analysis and Optimization*, vol 953, 2007.
Cat swarm	Chu, Shu-Chuan, "Cat swarm optimization." *PRICAI 2006: Trends in Artificial Intelligence*. Springer Berlin Heidelberg, 2006. 854-858.	Eagle strategy	Yang, Xin-She, "Eagle strategy using lévy walk and firefly algorithms for stochastic optimization.", Springer Berlin Heidelberg, 2010. 101-111.
Consultant-guided search	Iordache, Serban. *Consultant-guided search algorithms for the quadratic assignment problem*. Springer Berlin Heidelberg, 2010.	Fish swarm-school	Cheng, Yongming, "Novel clustering algorithms based on improved artificial fish swarm algorithm." *FSKD'09* IEEE, 2009.

B. Biological -Inspired Algorithms (*But Not Si Based)*

Obviously, SI-based algorithms belong to a wider class of algorithms, called biological-inspired (bio-inspired) algorithms. In fact, bio-inspired algorithms form a majority of all nature-inspired algorithms. From the set theory point of view, SI-based algorithms are a subset of bio-inspired algorithms, while bio-inspired algorithms are a subset of nature-inspired algorithms. That is SI-based ⊂ bio-inspired ⊂ nature-inspired.

Many bio-inspired algorithms do not use directly the swarming behaviour. Therefore, it is better to call them bio-inspired, but not SI-based. For example, genetic algorithms are bio-inspired, but not SI-based [38].

1. Genetic Optimization Algorithm (GA)

Genetic Algorithms as (biologically inspired approaches) [44,45], are search algorithms based on natural genetics that provide robust search capabilities in complex spaces, and thereby offer a valid approach to problems requiring efficient and effective search processes[39,40].

Some of the advantages of a GA include that it [4].

• Optimizes with continuous or discrete variables.
• Doesn’t require derivative information.
• Simultaneously searches from a wide sampling of the cost surface.
• Deals with a large number of variables.
• Is well suited for parallel computers.
• Optimizes variables with extremely complex cost surfaces (they can jump out of a local minimum).

• Provides a list of optimum variables, not just a single solution.
• May encode the variables so that the optimization is done with the encoded variables.
• Works with numerically generated data, experimental data, or analytical functions.

GAs are population-based search techniques that maintain populations of potential solutions during searches. A string with a fixed bit-length usually represents a potential solution. In order to evaluate each potential solution, GAs need a payoff (or reward, objective) function that assigns scalar payoff to any particular solution. Once the representation scheme and evaluation function is determined, a GA can start searching. Initially, often at random, GAs create a certain number, called the population size, of strings to form the first generation. Next, the payoff function is used to evaluate each solution in this first generation. Better solutions obtain higher payoffs. Then, on the basis of these evaluations, some genetic operations are employed to generate the next generation. The procedures of evaluation and generation are iteratively performed until the optimal solution is found or the time allotted for computation ends [41,42]. The evolution flow of genetic algorithm schematic diagram is given in Fig. 11.

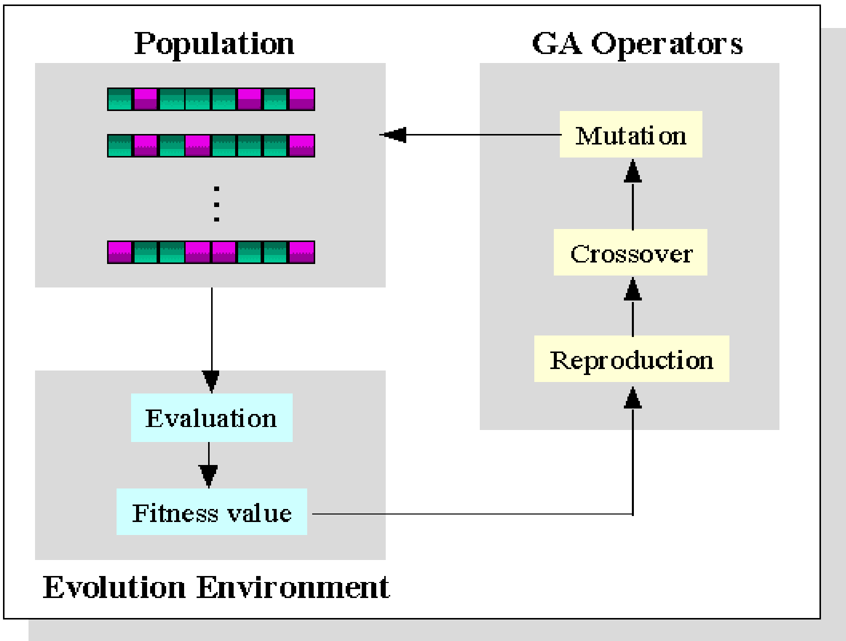


Fig. 11 Evolution flow of genetic algorithm[42]

To set the values for the various parameters, such as population size, crossover rate, and mutation rate. These parameters typically interact with
one another nonlinearly, so they cannot be optimized one at a time[43]. There are two main types of GA, The two types of GA binary GA and continuous GA (real-valued) are discuss in details [4], the flowchart is given in Fig. 12.

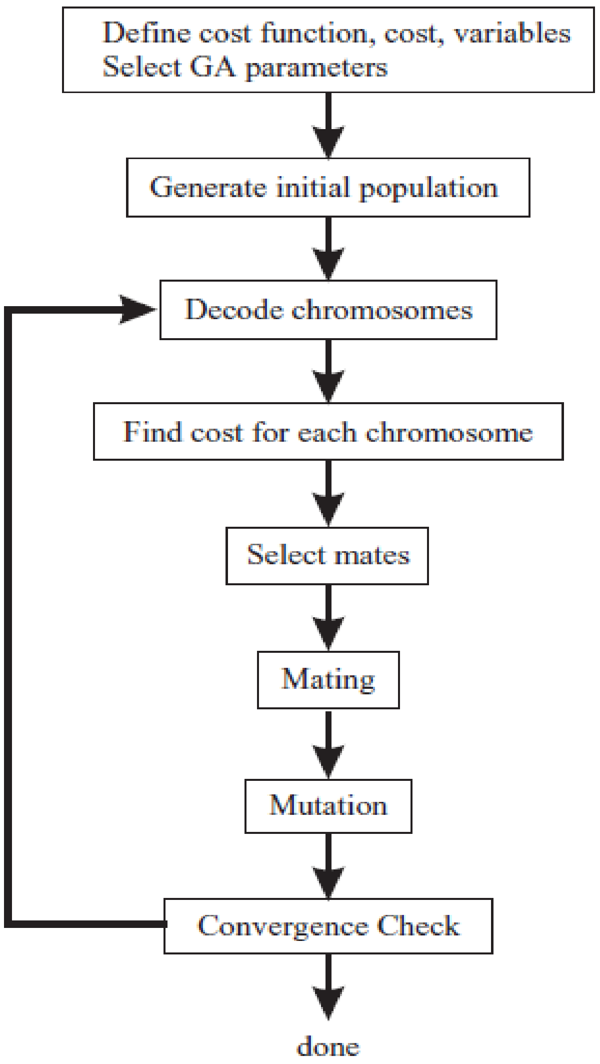


Fig. 12-a Flowchart of a binary GA[4]

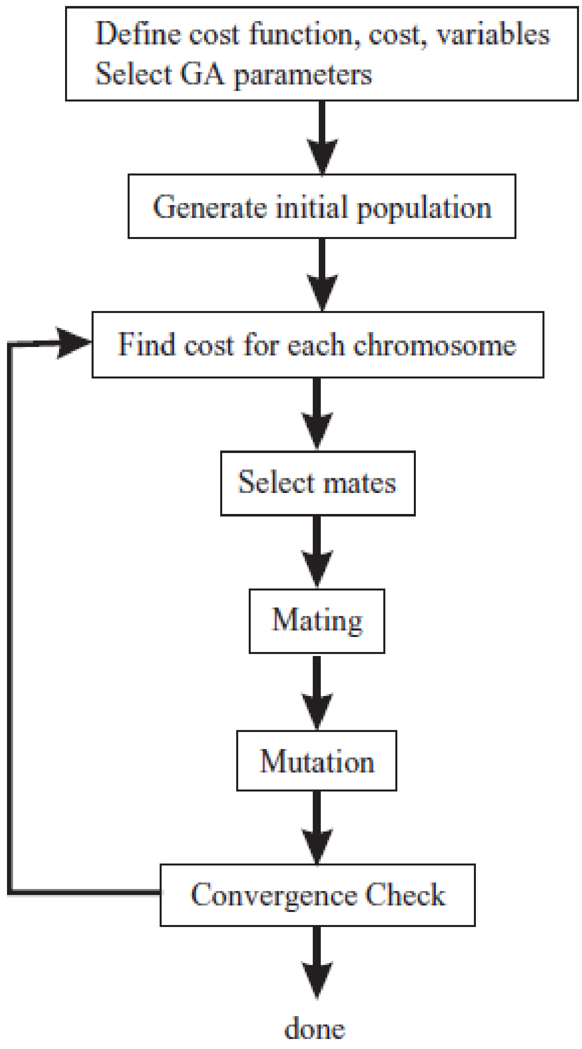


Fig. 12-b Flowchart of a continuous GA[4]

2. Artificial Neural Network Optimization Algorithm

An Artificial neural network (ANN) is a biologically inspired computational model formed from several of single units, artificial neurons, connected with coefficients (weights) which constitute the neural structure. Each processing elements (PE) has weighted inputs, transfer function and one output. PE is essentially an equation which balances inputs and outputs. There are many types of (ANN) designed, all can be described by the transfer functions of their neurons, by the training or learning algorithm (rule), and by the connection formula[46]. In the ANN, simple artificial nodes (called neurons) are connected together to form a network of nodes mimicking the biological neural network. There are various kinds of neural network mechanisms are explored, feed-forward neural networks, recurrent neural networks, time-delayed neural networks, real-time recurrent neural networks, etc. [47]. The Basic structure of an Artificial Neural Network is given in Fig. 13.

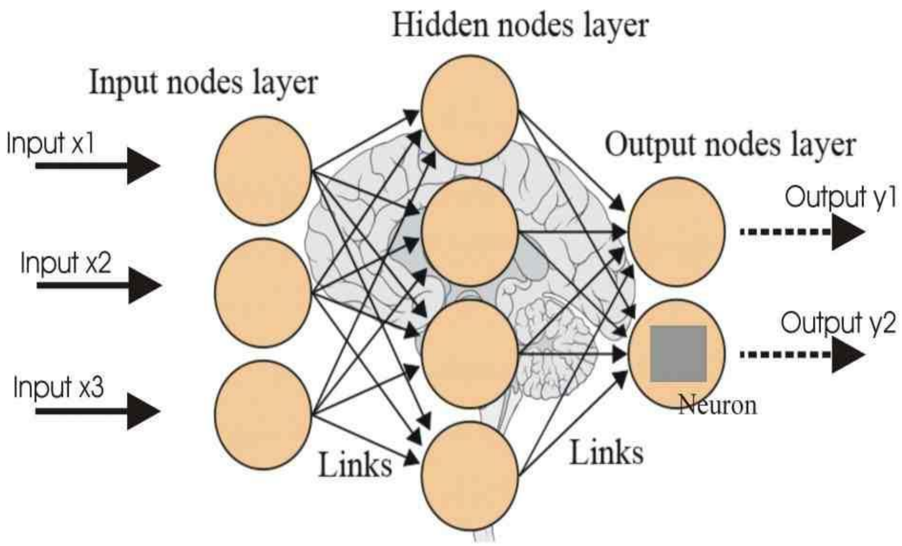


Fig. 13 Basic structure of an Artificial Neural Network (ANN)

3. Differential Evolution Optimization Algorithm

for minimizing possibly nonlinear and non-differentiable continuous space functions, by means of an extensive testbed it is demonstrated that converges faster and with more certainty than many other acclaimed global optimization methods. It requires few control variables, is robust, easy to use, and lends itself very well to parallel computation[53].

DE pseudo code[54]:

Step 1: random initialization of the parent population.Randomly generate a population of (say) NP vectors, each of n dimensions: xi,j= xmin,j + rand(0, 1)(xmax,j-xmin,j), where xmin,j and xmax are lower and upper bounds for jth component respectively, rand(0,1) is auniform random number between 0 and 1.
Step 2: Calculate the objective function value f(Xi) for all Xi.
Step 3: Select three points from population and generate perturbed individual Vi .
Step 4: Recombine the each target vector xi with perturbed individual generated in step 3 to generate a trial vector Ui .
Step 5: Check variables of the trial vector is within range. If yes, then go to step 6 else make it within range using ui,j =2 xmin,j - ui,j ,if ui,j < xmin,j and ui,j =2* xmax,j - ui,j , if ui,j> xmax,j, and go to step 6.*
Step 6: Calculate the objective function value for vector Ui.
Step 7: Choose better of the two (function value at target and trial point) for next generation.
Step 8: Check whether convergence criterion is met if yes then stop; otherwise go to step 3.

Table 3 SOME OTHER KIND OF BIO-INSPIRED ALGORITHMS

Algorithm	Author, Reference	Algorithm	Author, Reference
Brain Storm Optimization	Shi, Yuhui. "Brain storm optimization algorithm." *Advances in Swarm Intelligence*. Springer Berlin Heidelberg, 2011. 303-309.	Invasive weed optimization	Rad, Hoda Sepehri, and Caro Lucas. "A recommender system based on invasive weed optimization algorithm" *CEC 2007*. IEEE, 2007.
Flower pollination algorithm	Yang, Xin-She. "Flower pollination algorithm for global optimization" *Unconventional Computation and Natural Computation*. Springer Berlin Heidelberg, 2012. 240-249.	Paddy Field Algorithm	Wang, Sheng, et al. "RBF neural network parameters optimize based on paddy field algorithm" *(ICIA),IEEE*, 2011.
Group search optimizer	He, Shan, "Group search optimizer: an optimization algorithm inspired by animal searching behavior" *EC, IEEE Transactions on* 13.5 (2009): 973-990.	Roach infestation algorithm	Havens, Timothy C., et al. "Roach infestation optimization." *Swarm Intelligence Symposium, 2008. SIS 2008. IEEE*. IEEE, 2008.
Human-Inspired Algorithm	Zhang, Luna Mingyi,"Human-inspired algorithms for continuous function optimization." *ICIS, IEEE*, 2009.	Termite colony optimization	Hedayatzadeh, Ramin, et al. "Termite colony optimization: A novel approach for optimizing continuous problems" *(ICEE)*, IEEE, 2010.

C. Physics And Chemistry Algorithms

For the algorithms that are not bio-inspired, most have been developed by mimicking certain physical and/or chemical laws, including electrical charges, gravity, river systems, etc. As different natural systems are relevant to this category, we can even subdivide these into many subcategories which is not necessary[38] . Some other types of bio-inspired algorithms are given in table 3. Different types of physics and chemistry algorithms are given in table 4.

1. Simulated Annealing Optimization Algorithm

The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one[70]. There is a deep and useful connection between statistical mechanics and multivariate or combinatorial optimization (finding the minimum of a given function depending on many parameters). A detailed analogy with annealing in solids provides a framework for optimization of the properties of very large and complex systems. This connection to statistical mechanics exposes new information and provides an unfamiliar perspective on traditional optimization problems and methods[55]. Fig 14 depicts the simulated annealing algorithm.

1	Initializing: Initial solution S_i Initial and final temperature: T_i and T_f $T = T_i$
2	Temperatures cycle:
3	Metropolis cycle: Generating S_j from S_i $dif = J(S_j) - J(S_i)$ If $dif < 0$ then $S_i = S_j$ else if $e^{\frac{-dif}{T}} > \text{rnd}(0,1)$ $S_i = S_j$
4	Metropolis condition: If thermal equilibrium is reached goto 5 Else goto 3
5	Stop criterion: If the final temperature T_f is reached End Else Update T goto 2

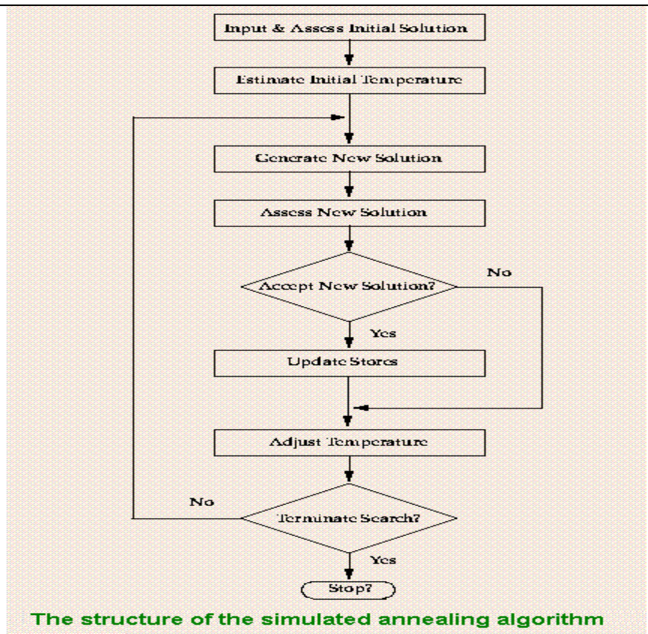


Fig. 14 Simulated annealing algorithm

2. Harmony Search Optimization Algorithm

When listening to a beautiful piece of classical music, we wondered if there is any connection between music and finding an optimal solution. Harmony search is a music-based metaheuristic optimization algorithm. It was inspired by the observation that the aim of music is to search for a perfect state of harmony. This harmony in music is analogous to find the optimality in an optimization process. On the other hand, an optimal solution to an optimization problem should be the best solution available to the problem under the given objectives and limited by constraints. Both processes intend to produce the best or optimum[56,22]. describes a new harmony search with continuous design variables, for a perfect state of harmony as given in Fig. 15 [57].

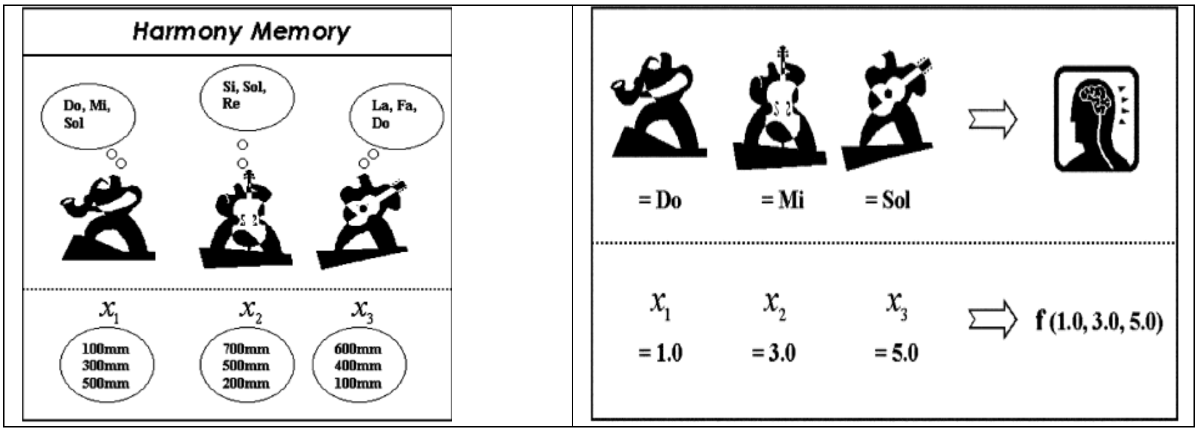


Fig. 15 Analogy between music improvisation and engineering optimization [57]

each player sounds any pitch within the possible range, together making one harmony vector. If all the pitches make a good harmony, that experience is stored in each players memory, and the possibility to make a good harmony is increased next time. Similarly in engineering optimization, each decision variable initially chooses any value within the possible range, together making one solution vector. If all the values of decision variables make a good solution, that experience is stored in each variable_s memory, and the possibility to make a good solution is also increased next time as given in Fig. 16 [57].

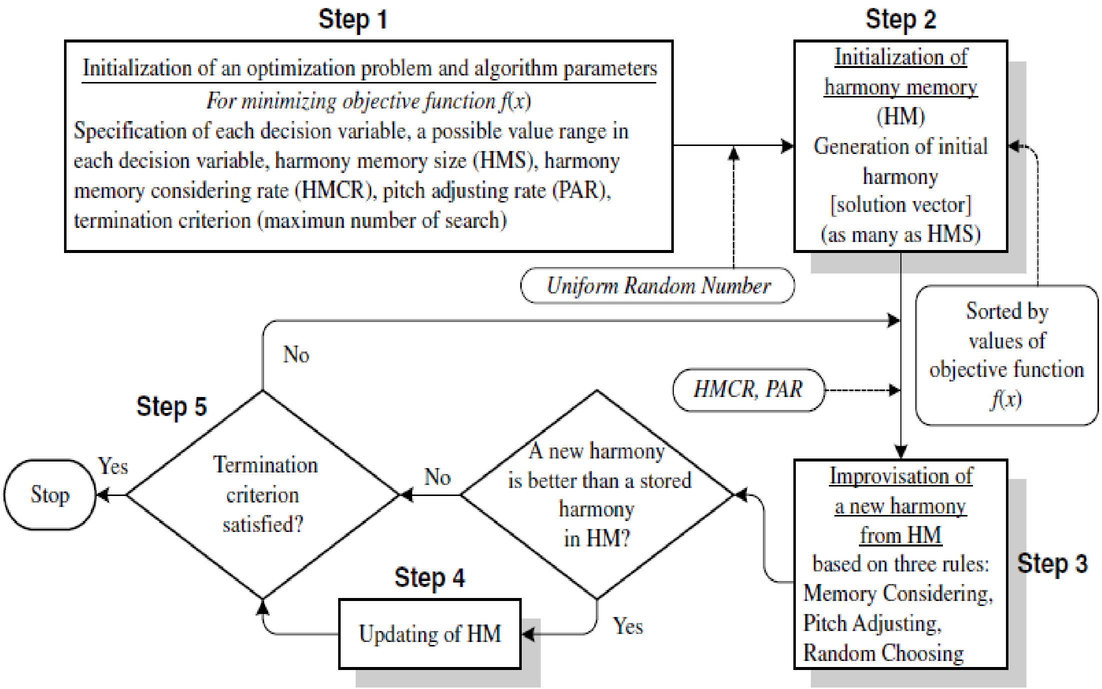


Fig. 16 Optimization procedure of the harmony search algorithm [57].

3. Gravitational Search Algorithm

Gravitational Search Algorithm (GSA) is a recent algorithm that has been inspired by the Newtonian's law of gravity and motion. Since its introduction in 2009, GSA has undergone a lot of changes to the algorithm itself and has been applied in various applications. At present, there are various variants of GSA which have been developed to enhance and improve the original version. The algorithm has also been explored in many areas[58].

The GSA could be considered as an isolated system of masses. It is like a small artificial world of masses obeying the Newtonian laws of gravitation and motion. More precisely, masses obey the following laws. The flowchart is given in Fig. 17[59]:

Law of gravity: the gravitational force between two particles is directly proportional to the product of their masses and inversely proportional to the distance between them.

Law of motion: acceleration of any mass is equal to the force acted on the system divided by mass of inertia.

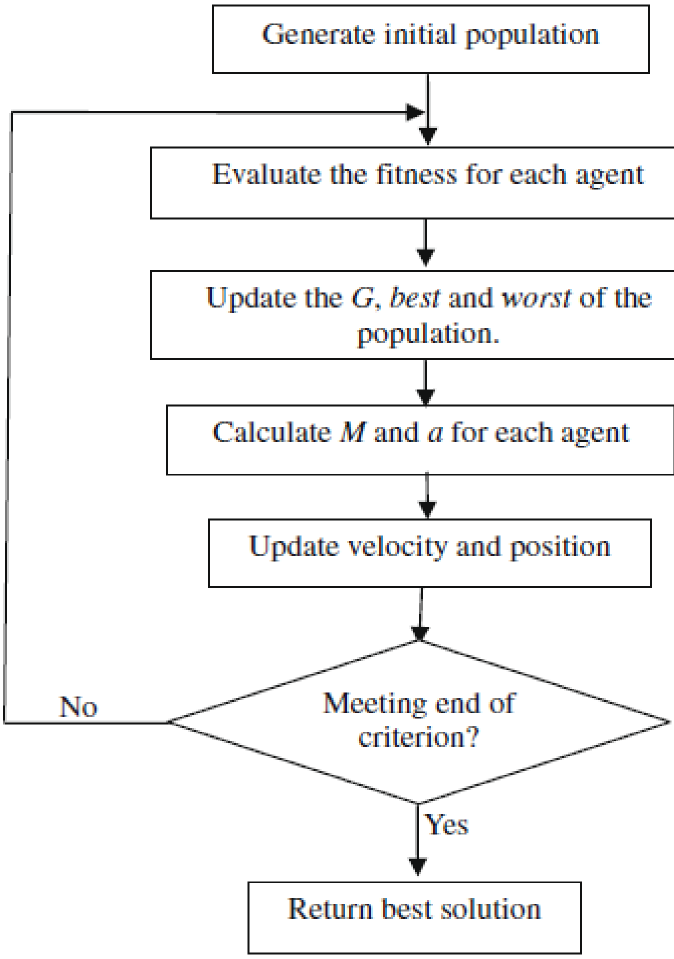


Fig. 17 General principle of GSA [58,59].

Table 4 SOME KINDS OF PHYSICS AND CHEMISTRY ALGORITHMS

Algorithm	Author, Reference	Algorithm	Author, Reference
Black hole	Hatamlou, "Black hole: A new heuristic optimization approach for data clustering." *Information Sciences* 222 (2013): 175-184.	River formation dynamics	Rabanal, Pablo, "Using river formation dynamics to design heuristic algorithms", Springer Berlin, 2007. 163-177.
Central force	Formato, "Parameter-free deterministic global search with simplified central force optimization" *AICTA* Springer Berlin, 2010.	Stochastic diffusion search	al-Rifaie, M. Majid,"The mining game: a brief introduction to the stochastic diffusion search metaheuristic." *Q: The magazine of AISB* 130 (2010): 8-9.
charged system search	Kaveh, "A novel heuristic optimization method charged system search" *Acta*,2010	Spiral search	Tamura, Kenichi,"Spiral multipoint search for global optimization" *Machine Learning ApplicationsWorkshops(ICMLA)*,IEEE, 2011.

galaxy-based search algorithm	S.Hosseini, "Principal components analysis by the galaxy-based search algorithm"*International Journal of Computational Science and Engineering* (2011)	Water cycle algorithm	Eskandar, Hadi, et al. "Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems" *Computers & Structures* 110 (2012): 151-166.

IV. HYBRID OPTIMIZATION TECHNIQUES

In hybrid intelligent techniques, two or more artificial intelligent techniques are integrated to into one algorithm to obtain successful results [67]. If done right, then the advantages of each method can be used to overcome the disadvantages of the others, leading to a very powerful algorithm. Typically, hybrid methods can achieve significant improvements (e.g. in computation time, convergence properties, solution quality, or parameter robustness) over each of the individual methods. Hybrid methods have gained popularity in the last decade for various applications [63]. Some types of physics and chemistry algorithm is given in table 3.

1. Particle Swarm combined with Gravitational Search (PSOGSA).

hybrid population-based algorithm (PSOGSA) is proposed with the combination of Particle Swarm Optimization (PSO) and Gravitational Search Algorithm (GSA). The main idea is to integrate the ability of exploitation in PSO with the ability of exploration in GSA to synthesize both algorithms' strength. The results in[71], show that the hybrid algorithm possesses a better capability to escape from local optimums with faster convergence than the standard PSO and GSA. ...

2. Cuckoo Search Inspired Hybridization of the Nelder-Mead

Hybridization of the Cuckoo Search (CS) with the Nelder-Mead method. More precisely, instead of using single solutions as nests for the CS, use the concept of a simplex which is used in the Nelder-Mead algorithm. This makes it possible to use the flip operation introduces in the Nelder-Mead algorithm instead of the Levy flight which is a standard part of the CS. In this way, the hybridized algorithm becomes more robust and less sensitive to parameter tuning which exists in CS. The results in[71], show that the new method has a better performance when compared to similar but more complex hybridizations of Nelder-Mead algorithm using genetic algorithms or particle swarm optimization on standard benchmark functions. Finally, we show that the new method outperforms some standard meta-heuristics for the problem of interest.

Conclusions:

Different optimization techniques were applied for improvement the network performances, improving the data classification, and unit commitment [72- 91]. These techniques is implemented to determine the accurate locations of DGs, determining the best data classification, find the optimum size of controllers. The optimization techniques have wide varieties of applications in power systems, including drives, micro-grid, data classifications, image processing and controller parameters design [92-123].

REFERENCES

1. Singiresu S. Rao, "Engineering Optimization Theory And Practice", Fourth Edition, 1996 by John Wiley & Sons, Inc.,USA
2. A.V.N. Purna Chandra Rao, B.R. Jyoshna, "Swarm intelligence", Anil Neerukonda Institute of Technology and Sciences, 2010
3. R. Baskarane, T. Sendhil kumar, "Hybrid Optimization for Multiobjective Multicast Routing", International Journal of Research in Advent Technology, Vol.2, March 2014
4. Randy L. Haupt, Sue Ellen Haupt, "Practical Genetic Algorithms", Second Edition, 2004, A JOHN WILEY & SONS, INC.
5. Panos M. Pardalos, H. Edwin Romeijn, "Handbook Of Global Optimization", Kluwer Academic Publishers
6. Thomas Weise, "Global Optimization Algorithms Theory and Application", Book, Second Edition, 2009.
7. Angel Udías, Roman Efremov, Javier Cano, Andres Redchuk, "Methodological review of multicriteria optimization techniques", Technical Report 2012, Analisis Deriesgos
8. Xiaolei Wang, "Hybrid Nature-Inspired Computation Methods For Optimization", Doctoral Dissertation, Helsinki Of Technology, 2009
9. Abbass H A, “A monogenous MBO approach to satisfiability”. In: Proceeding of the International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA'2001. Las Vegas, NV, USA
10. Abbass H A, “Marriage in honey-bee optimization (MBO): a haplometrosis polygynous swarming approach.” In: The Congress on Evolutionary Computation (CEC2001). Seoul, Korea, May 2001: 207-214
11. Afshar A, Bozog Haddad O, Marino M A, Adams B J, “Honey-bee mating optimization (HBMO) algorithm for optimal reservoir operation.”, Journal of the Franklin Institute 344: 452-462, 2007
12. Fathian M, Amiri B, Maroosi A, “Application of honey bee mating optimization algorithm on clustering”. Applied Mathematics and Computation, doi: 10.1016/j.amc. 2007
13. Gutin, G. and A. Punnen, “The Traveling Salesman Problem And Its Variations”, Kluwer Academic Publishers, Dordrecht.,2002
14. Haddad O B, Afshar A, Marino M A, “Honey-bees mating optimization (HBMO) algorithm: A new heuristic approach for water resources optimization.” Water Resources Management 20: 661-680, 2006
15. Lawer, E.L., Lenstra, J.K., Rinnoy Kan, A.H.G., D.B. Shmoys,“The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization”, Wiley and Sons., 1985
16. Toth P, Vigo D, “The vehicle routing problem”. Monographs on Discrete Mathematics and Applications, SIAM, 2002

17.M. Dorigo, Gianni Di Caro, and Luca M. Gambardella. Ant Algorithms for Discrete Optimization. Technical Report Tech. Rep. IRIDIA/98-10, IRIDIA, Universite Libre de Bruxelles, Brussels, Belgium, 1998.

18.M. Dorigo and M. Maniezzo and A. Colorni. The Ant Systems: An Autocatalytic Optimizing Process. Revised 91-016, Dept. of Electronica, Milan Polytechnic, 1991.

19.S. Goss, S. Aron, J. L. Deneubourg, and J. M. Pasteels. Self-organized Shorcuts in the Argentine Ant. Naturwissenschaften, 76:579–581, 1989.

20.M. Dorigo and G. Di Caro. New Ideas in Optimisation. McGraw Hill, London, UK, 1999.

21.Stefan Janson, Daniel Merkle, Martin Middendorf, "A Decentralization Approach For Swarm Intelligence Algorithms In Networks Applied To Multi Swarm Pso", International Journal of Intelligent Computing and Cybernetics, 2008

22.Xin-She Yang, "Nature-InspiredMetaheuristic Algorithms", Second Edition, 2010 by Luniver Press

23. X.-S. Yang, S. Deb, “Cuckoo search via L´evy flights”, World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), December 2009, India. IEEE Publications, USA, pp. 210-214 (2009).

24.Ehsan Valian, Shahram Mohanna and Saeed Tavakoli , "Improved Cuckoo Search Algorithm for Global Optimization", International Journal of Communications and Information Technology, IJCIT, Dec 2011

25.Xin-She Yang, "Firefly Algorithms for Multimodal Optimization", 5th international conference on Stochastic algorithms, Pages 169-178 Berlin, Heidelberg,2009

26.J. Kwiecień, B. Filipowicz, "Firefly Algorithm In Optimization Of Queueing Systems", Bulletin Of The Polish Academy Of Sciences Technical Sciences, Vol. 60, No. 2, 2012

27. X. S. Yang, "A New Metaheuristic Bat-Inspired Algorithm", Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)

28.J. D. Altringham, "Bats: Biology and Behaviour", Oxford University Press, 1996.

29. P. Richardson, "Natural History Museum", London, 2008

30. Parpinelli, R. S., and Lopes, H. S., New inspirations in swarm intelligence: a survey,Int. J. Bio-Inspired Computation, Vol. 3, 1-16 (2011).

31.T. C. Bora, L. S. Coelho, L. Lebensztajn, Bat-inspired optimization approach for the brushless DC wheel motor problem, IEEE Trans. Magnetics, Vol. 48 (2), 947-950 (2012).

32. Kennedy, J. and Eberhart, R. C. (1995) ‘Particle swarm optimization’, Proc. of IEEE International Conference on Neural Networks, Piscataway, NJ. pp. 1942-1948.

33.Dah-Jing Jwo, Shun-Chieh Chang, (2009) "Particle swarm optimization for GPS navigation Kalman filter adaptation", Aircraft Engineering and Aerospace Technology, Vol. 81 Iss: 4, pp.343 – 352

34. Anu Bala, "A Survey On Different Techniques Of Spectrum Allocation In Congnitive Radio Networks". IJREAS Volume 2, Issue 2, February 2012
35. H. Abbass, “A Monogenous MBO Approach to Satisfiability,” Proceeding of the International Conference on Computational Intelligence for Modelling, Control and Automation, Las Vegas, 2001.
36. H. Abbass, “Marriage in Honey-Bee Optimization (MBO): A Haplometrosispolygynous Swarming Approach,” The Congress on Evolutionary Computation (CEC2001), Seoul, 27-30 May 2001, pp. 207-214.
37. A. Afshar, O. Bozog Haddad,“Honey-Bee Mating Optimization (HBMO) Algorithm for Optimal Reservoir Operation,” Journal of the Franklin Institute, Vol. 344, No. 5, 2007, pp. 452-462.
38. I. Fister Jr., Xin-She Yang, I. Fister, J. Brest, Dušan, "A Brief Review of Nature-Inspired Algorithms for Optimization", Elektrotehni\vški vestnik, 80(3), 2013
39. J.H. Holland, "Adaptation in natural and artificial systems", MIT Press, Michigan ، 1992
40. Dharminder Kumar, Sunita Beniwal, "Genetic Algorithm And Programming Based Classification: A Survey ", Journal of Theoretical and Applied Information Technology, August 2013. Vol. 54 No.1
41. J. J. Grefenstette. "Optimization of control parameters for genetic algorithms". IEEE Transactions on System, Man, and Cybernetics, SMC-16(1):122-128, 1986.
42. Ying Liao, Chuen Sun, "An Educational Genetic Algorithms Learning Tool", IEEE, 2001.
43. Mitchell Melanie, "An Introduction to Genetic Algorithms",A Bradford Book The MIT Press, Cambridge, Fifth printing, 1999
44. Noraini Mohd and Geraghty, "*Biologically inspired genetic algorithm to minimize idle time of the assembly line balancing*" 3rd World Congress on Nature and Biologically Inspired Computing (NaBIC2011), Oct 2011, Spain. ISBN 978-1-4577-1122-0
45. Gerken W, Purvis L, Butera R., "Genetic algorithm for optimization and specification of a neuron model", Conf Proc IEEE Eng Med Biol Soc. 2005;4:4321-3.
46. A. Ghaffari, H. Abdollahi, M.R. Khoshayand,A. Dadgar, "Performance Comparison Of Neural Network Training Algorithms In Modeling Of Bimodal Drug Delivery", International Journal of Pharmaceutics (2006) 126–138
47. M. Janga Reddy, D. Nagesh Kumar, "Computational algorithms inspired by biological processes and evolution", CURRENT SCIENCE, Vol. 103, No. 4, 25 August (2012)
48. Dario Floreano and Claudio Mattiussi, "Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies", MIT Press, 2008
49. Irene Giardina, "Collective behavior in animal groups: theoretical models and empirical studies", HFSP J. Aug 2008; 2(4): 205–219.

50. Utkarsh Jaiswal, Shweta Aggarwal, "Ant Colony Optimization", International Journal of Scientific & Engineering Research Vol 2, Issue 7, July-2011
51. K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," IEEE Control Systems Magazine, vol. 22, no. 3, pp. 52–67, 2002.
52. A. Biswas, S. Dasgupta, S. Das, and A. Abraham, "Synergy of PSO and bacterial foraging optimization—a comparative study on numerical benchmarks," in Advances in Soft Computing, vol. 44 of Innovations in Hybrid Intelligent Systems, pp. 255–263, 2007
53. Rainer Storn, Kenneth Price, "Differential Evolution – A Simple And Efficient Heuristic For Global Optimization Over Continuous Spaces",1997, Journal Of Global Optimization
54. Musrrat Ali, Millie Pant, Ajith Abraham, "Simplex Differential Evolution", Acta Polytechnica Hungarica Vol. 6, No. 5, 2009.
55. S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi, "Optimization by Simulated Annealing",Science,2007
56. X.-S. Yang, "Harmony Search as a Metaheuristic Algorithm", Studies in Computational Intelligence, Springer Berlin, vol. 191, pp. 1-14 (2009)
57. Kang Seok Lee, Zong Woo Geem, "A new meta-heuristic algorithm for continuous engineering optimization", ELSEVIER , Comp. Methods Appl. Mech. Engg. 194 (2005)
58. N. Sabri, M. Puteh, M. Mahmood, "A Review of Gravitational Search Algorithm", International Journal Advance. Soft Comput. Appl., Vol. 5, No. 3, November 2013
59. E. Rashedi, Nezamabadi-pour, S. Saryazdi, "GSA: A Gravitational Search Algorithm", Elsevier, Information Sciences 179 (2009) 2232–2248
60. Yang, Xin-She. "Nature-inspired mateheuristic algorithms: success and new challenges." arXiv preprint arXiv:1211.6658 (2012).
61. Péter Kádár, " Application of Optimization Techniques in the Power System Control" , Acta Polytechnica Hungarica Vol. 10, No. 5, 2013
62. Belkacem Mahdad, "Power Quality Management Based Security OPF Considering FACTS Using Metaheuristic Optimization Methods", book ISBN: 978-953-51-1068-2, April 2013
63. S. Frank, I. Steponavice, S. Rebennak, "Optimal power flow: a bibliographic survey II, non-deterministic and hybrid methods," Int. J. Energy.System (Springer-Verlag), 2012.
64. M. Eslami, H. Shareef, A. Mohamed, "Application Of artificial intelligent techniques in PSS design: a survey of the state-of-the-art methods", Przegląd Elektrotechniczny (Electrical Review), ISSN 0033-2097, R. 87 NR 4/2011.
65. R. Sirjani, et al, "Heuristic optimization techniques to determine optimal capacitor placement and sizing in radial distribution networks: A comprehensive review," Przegląd Elektrotechniczny (Electrical Review), pp. 1-7, 2012.
66. Shoghian, Shahla, "A Comparison among Wolf Pack Search and Four other Optimization Algorithms", World Academy of Science, Engineering and Technology Vol:6 2012

67. O. Mahela, D. Mittal, L. Goyal, "Optimal Capacitor Placement Techniques in Transmission and Distribution Networks to Reduce Line Losses and Voltage Stability Enhancement: A Review", IOSR Journal of Electrical and Electronics Engineering ISSN: 2278-1676Vol 3, Dec. 2012.
68. Tang, Rui, et al. "Wolf search algorithm with ephemeral memory." Digital Information Management (ICDIM), 2012 Seventh International Conference on. IEEE, 2012.
69. S. Mirjalili, S. M. Mirjalili, A. Lewis, "Grey Wolf Optimizer", Advances in Engineering Software, Volume 69, March 2014, Pages 46-61, ISSN 0965-9978.
70. El-Hosseini, Mostafa A., A. Abraham, H. Al-Qaheri,"Genetic Annealing Optimization: Design and Real World Applications." Intelligent Systems Design and Applications, 2008. ISDA'08. Eighth International Conference on. Vol. 1. IEEE, 2008.
71. Mirjalili, Seyedali, S. Zaiton, "A new hybrid PSOGSA algorithm for function optimization", Computer and information application (ICCIA), 2010 international conference on. IEEE, 2010.
72. Mahmoud G. Hemeida, Salem Alkhalaf, Al-Attar A. Mohamed, Abdalla Ahmed Ibrahim and Tomonobu Senjyu, "Distributed Generators Optimization Based on Multi-Objective Functions Using Manta Rays Foraging Optimization Algorithm (MRFO)", Energies 2020, 13, 3847; doi:10.3390/en13153847.
73. Mahmoud G. Hemeida, Abdalla Ahmed Ibrahim, Al-Attar A. Mohamed, Salem Alkhalaf, Ayman M. Bahaa El-Dine, "Optimal allocation of distributed generators DG based Manta Ray Foraging Optimization algorithm (MRFO)", Ain Shams Engineering Journal 12 (2021) 609-619.
74. Mahmoud G Hemeida, Salem Alkhalaf, Tomonobu Senjyu, Abdalla Ibrahim, Mahrous Ahmed, Ayman M. Bahaa-Eldin, "Optimal probabilistic location of DGs using Monte Carlo simulation based different bio-inspired algorithms", Ain Shams Engineering Journal.
75. Yongyi Huang, Atsushi Yona, Hiroshi Takahashi, Ashraf Mohamed Hemeida, Paras Mandal, Alexey Mikhaylov, Tomonobu Senjyu and Mohammed Elsayed Lotfy, "Energy Management System Optimization of Drug Store Electric Vehicles Charging Station Operation", Sustainability 2021, 13, 6163. https://doi.org/10.3390/su13116163.
76. Akito Nakadomari, Ryuto Shigenobu, Takeyoshi Kato, Narayanan Krishnan, Ashraf Mohamed Hemeida, Hiroshi Takahashi and Tomonobu Senjyu, "Unbalanced Voltage Compensation with Optimal Voltage Controlled Regulators and Load Ratio Control Transformer", Energies 2021, 14, 2997. https://doi.org/10.3390/en14112997.
77. Yongyi Huang, Hasan Masrur, Ryuto Shigenobu, Ashraf Mohamed Hemeida, Alexey Mikhaylov and Tomonobu Senjyu, "A Comparative Design of a Campus Microgrid Considering a Multi-Scenario and Multi-Objective Approach", Energies 2021, 14, 2853. https://doi.org/10.3390/en14112853.
78. Gul Ahmad Ludin, Mohammad Amin Amin, Hidehito Matayoshi, Shriram S. Rangarajan, Ashraf M. Hemeida, Hiroshi Takahashi and Tomonobu Senjyu, "Solid-State DC Circuit Breakers and their Comparison in Modular Multilevel Converter Based-HVDC Transmission System", Electronics 2021, 10, 1204. https://doi.org/10.3390/electronics10101204.
79. Mahmoud M.Gamil, Mohammed Elsayed Lotfy, Ashraf M.Hemeida, Paras Mandal, Hiroshi Takahashi and Tomonobu Senjyu, "Optimal sizing of a residential microgrid in Egypt under deterministic and stochastic conditions with PV/WG/Biomass Energy integration", AIMS Energy (2021), 9(3): 483–515.

80. Ashraf M. Hemeida, Somaia Awad, Mahmoud Ali Saber, Salem Alkhalaf, Mountasser Mohamed, Abdalla H. Alaied,'Optimization of matrix-matrix multiplications on Intel's Advanced Vector Extensions (AVX) Processor', Ain Shams Engineering Journal, Vol. 11, Issue 4, December, 2020, pp. 1179-1190.

81. Ayat Ali Saleh, Tomonobu Senjyu, Salem Alkhalaf, Majed A. Alotaibi, and Ashraf M. Hemeida, "Water Cycle Algorithm for Probabilistic Planning of Renewable Energy Resource, Considering Different Load Models", Energies 2020, 13, 5800; doi:10.3390/en13215800.

82.**Ashraf Hemeida**, Shimaa Mohamed, Mountasser Mahmoud, "Load Frequency Control Using Optimized Techniques", Journal of Engineering Sciences, Assiut University, Faculty of Engineering, Vol. 48, Issue 6, November 2020, pp. 1119-1136.

83.Somaia Hassan, Ashraf M. Hemeida, Salem Alkhalaf, Al-Attar Mohamed and Tomonobu Senjyu**,** **"**Multi-variant diferential evolution algorithm for feature selection", Scientific Reports (2020) 10:17261, https://doi.org/10.1038/s41598-020-74228-0.

84.Mahmoud M. Gamil, Makoto Sugimura, Akito Nakadomari, Tomonobu Senjyu, Harun Or Rashid Howlader, Hiroshi Takahashi and Ashraf M. Hemeida, "Optimal Sizing of a Real Remote Japanese Microgrid with Sea Water Electrolysis Plant Under Time-Based Demand Response Programs", Energies 2020, 13, 3666; doi:10.3390/en13143666.

85.Ashraf Mohamed Hemeida, Somaia Awad Hassan, Al-Attar Ali Mohamed, Salem Alkhalaf, Mountasser Mohamed Mahmoud, Tomonobu Senjyu, Ayman Bahaa El-Din, "Nature-inspired algorithms for feed-forward neural network classifiers: A survey of one decade of research", Ain Shams Engineering Journal 11 (2020) 659-675.

86.Hidehito Matayoshia, Mitsunaga Kinjob, Shriram S. Rangarajanc, Girish Ganesan R, Ashraf M. Hemeida , Tomonobu Senjyu, "Islanding operation scheme for DC microgrid utilizing pseudo droop control of photovoltaic system", Energy for Sustainable Development 55 (2020) 95-104.

87.Al-Attar A. Mohamed, S. A. Hassan, A. M. Hemeida, Salem Alkhalaf, Mountasser M. Mahmoud, Ayman M. Baha Eldin, "Parasitism – Predation algorithm (PPA): A novel approach for feature selection", Ain Shams Engineering Journal 11 (2020) 293-308.

88. A. M. Hemeida, et al. "Multi-Objective Optimization of A Stand Alone Hybrid PV/Wind/Battery/Diesel Micro-Grid", Accepted for oral presentation at the 2019 IEEE Conference on Power Electronics and Renewable Energy (CPERE), October 21-25, Aswan, Egypt.

89.Shimaa Mohamed Ali, Al-Attar Ali Mohamed, A. M. Hemeida, "A Pareto Strategy based on Multi-Objective for Optimal Placement of Distributed Generation Considering Voltage Stability", 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), 2019.

90.Al-Attar Ali Mohamed, Shimaa Mohamed Ali, A. M. Hemeida, Abdalla Ahmed Ibrahim. "Optimal Placement of Distributed Energy Resources Including Different Load Models Using Different Optimization Techniques", 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), 2019.

91.Ayat Ali Saleh, Al-Attar Ali Mohamed, A. M. Hemeida. "Optimal Allocation of Distributed Generations and Capacitor Using Multi-Objective Different Optimization Techniques", 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), 2019

92.Ayat Ali Saleh, Al-Attar Ali Mohamed, Ashraf Mohamed Hemeida, Abdalla Ahmed Ibrahim. "Comparison of different optimization techniques for optimal allocation of multiple distribution generation", 2018 International Conference on Innovative Trends in Computer Engineering (ITCE), 2018

93. Ayat Ali Saleh, Al-Attar Ali Mohamed, A. M. Hemeida, Abdalla Ahmed Ibrahim. "Multi-Objective Whale Optimization Algorithm for Optimal Allocation of Distributed Generation and Capacitor Bank", 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), 2019.
94. Shimaa Mohamed Ali ; Al-Attar Ali Mohamed ; A.M. Hemeida, "A Pareto Strategy based on Multi-Objective for Optimal Placement of Distributed Generation Considering Voltage Stability", 2019 International Conference on Innovative Trends in Computer Engineering (ITCE).
95. Ayat Ali Saleh ; Al-Attar Ali Mohamed ; A. M. Hemeida, "Optimal Allocation of Distributed Generations and Capacitor Using Multi-Objective Different Optimization Techniques", 2019 International Conference on Innovative Trends in Computer Engineering (ITCE).
96. Al-Attar Ali Mohamed ; Ahmed Lotfy Haridy ; A.M. Hemeida, "The Whale Optimization Algorithm based controller for PMSG wind energy generation system", 2019 International Conference on Innovative Trends in Computer Engineering (ITCE).
97. Shimaa Mohamed Ali ; Al-Attar Ali Mohamed ; A.M. Hemeida, "Impact of Optimal Allocation of Renewable Distributed Generation in Radial Distribution Systems Based on Different Optimization Algorithms", 2019 International Conference on Innovative Trends in Computer Engineering (ITCE).
98. Al-Attar Ali Mohamed ; Shimaa Mohamed Ali ; A.M. Hemeida ; Abdalla Ahmed Ibrahim, "Optimal Placement of Distributed Energy Resources Including Different Load Models Using Different Optimization Techniques", 2019 International Conference on Innovative Trends in Computer Engineering (ITCE).
99. Ayat Ali Saleh ; Al-Attar Ali Mohamed ; A. M. Hemeida, "Impact of Optimum Allocation of Distributed Generations on Distribution Networks Based on Multi-Objective Different Optimization Techniques", 2019 International Conference on Innovative Trends in Computer Engineering (ITCE).
100. Ayat Ali Saleh ; Al-Attar Ali Mohamed ; A. M. Hemeida ; Abdalla Ahmed Ibrahim, "Multi-Objective Whale Optimization Algorithm for Optimal Allocation of Distributed Generation and Capacitor Bank", 2019 International Conference on Innovative Trends in Computer Engineering (ITCE)
101. Quazi Nafees Ul Islam, Ashik Ahmed, Saad Mohammad Abdullah, "Optimized controller design for islanded microgrid using non-dominated sorting whale optimization algorithm (NSWOA)", Ain Shams Engineering Journal.
102. Ziad M. Ali, Ibrahim Mohamed Diaaeldin, Ahmed El-Rafei, Hany M. Hasanien, Shady H.E. Abdel Aleem, Almoataz Y. Abdelaziz, "A novel distributed generation planning algorithm via graphically-based network reconfiguration and soft open points placement using Archimedes optimization algorithm", Ain Shams Engineering Journal.
103. Mohammad H. Soliman, Hossam E.A. Talaat, Mahmoud A. Attia, "Power system frequency control enhancement by optimization of wind energy control system", Ain Shams Engineering Journal.
104. Berge Djebedjian, Hossam A.A. Abdel-Gawad, Riham M. Ezzeldin, "Global performance of metaheuristic optimization tools for water distribution networks", Ain Shams Engineering Journal 12 (2021) 223-239.
105. A.M. Hussien, Hany M. Hasanien, S.F. Mekhamer, "Sunflower optimization algorithm-based optimal PI control for enhancing the performance of an autonomous operation of a microgrid", Ain Shams Engineering Journal.
106. Mohamed A.M. Shaheen, Hany M. Hasanien, Abdulaziz Alkuhayli, "A novel hybrid GWO-PSO optimization technique for optimal reactive power dispatch problem solution", Ain Shams Engineering Journal 12 (2021) 621-630.
107. Mihailo Micev, Martin C´ alasan, Ziad M. Ali, Hany M. Hasanien, Shady H.E. Abdel Aleem, "Optimal design of automatic voltage regulation controller using hybrid simulated annealing – Manta ray foraging optimization algorithm", Ain Shams Engineering Journal 12 (2021) 641-657.

108. Naggar H. Saad, Ahmed A. El-Sattar, Mohamed E. Marei, “Improved bacterial foraging optimization for grid connected wind energy conversion system based PMSG with matrix converter”, Ain Shams Engineering Journal 9 (2018) 2183-2193.
109. Narges Daryani, Kazem Zare, “Multiobjective power and emission dispatch using modified group search optimization method”, Ain Shams Engineering Journal 9 (2018) 319-328.
110. Moumita Pradhan, Provas Kumar Roy, Tandra Pal, “Oppositional based grey wolf optimization algorithm for economic dispatch problem of power system”, Ain Shams Engineering Journal 9 (2018) 2015-2025.
111. Ashraf Mohamed Hemeida, Mohamed M. Hamada, Youssef A. Mobarak , A. El-Bahnasawy , Mohamed G Ashmawy , Tomonobu Senjyu, “TCSC with Auxiliary Controls Based Voltage and Reactive Power Controls on Grid Power System”, Ain Shams Engineering Journal 11 (2020) 587- 609.
112. A. M. Hemeida, Salem Alkhalaf, A. Mady, E. A. Mahmoud, Mohamed E. Hussain, Ayman M. Baha Eldin, “Implementation of nature-inspired optimization algorithms in some data mining tasks” Ain Shams Engineering Journal 11 (2020) 309-318.
113. A. M. Hemeida, M. H. El-Ahmar, A. M. El-Sayed, Hany M. Hasanien, T. Senjyu, “Optimum design of hybrid wind/PV energy system for remote area”, Ain Shams Engineering Journal 11 (2020) 11-23.
114. Harun Or Rashid Howlader, Oludamilare Bode Adewuyi, Ying-Yi Hong, Paras Mandal, Ashraf Mohamed Hemeida and Tomonobu Senjyu, “Energy Storage System Analysis Review for Optimal Unit Commitment”, *Energies* **2020**, *13*(1), 158; doi:10.3390/en13010158
115. Yuta Susowake, Tetsuya Yabiku, Hasan Masrur, Tomonobu Senjyu, Abdul Motin Howlader, Mamdouh Abdel-Akher, Ashraf M. Hemeida, “A Multi - objective Optimization Approach Towards a Proposed Smart Apartment with Demand Response in Japan”, *Energies* 2020, *13*(1), 127; https://doi.org/10.3390/en13010127.
116. Al-Attar Ali Mohamed, Shimaa Ali, Salem Alkhalaf, Tomonobu Senjyu, Ashraf M. Hemeida, “Optimal Allocation of Hybrid Renewable Energy System by Multi-Objective Water Cycle Algorithm”, *Sustainability* 2019, *11*(23), 6550; https://doi.org/10.3390/su11236550.
117. Salem Alkhalaf, Tomonobu Senjyu, Ayat Ali Saleh, A. M. Hemeida, and Al-Attar Ali Mohamed, “A MODA and MODE Comparison for Optimal Allocation of Distributed Generations with Different Load Levels”, *Sustainability* 2019, *11*(19), 5323; https://doi.org/10.3390/su11195323
118. Sayed Mir Shah Danish, Ryuto Shigenobu, Mitsunaga Kingo, Paras Mandal, Narayanan K, Hemeida A. M. and Tomonobu Senjyu, “A Real Distribution Network Voltage Regulation Incorporating Auto-Tap-Changer Pole Transformer Multiobjective Optimization”, *Appl. Sci.* 2019, *9*, 2813; doi:10.3390/app9142813.
119. Al-Attar Ali Mohamed, Mohamed, Y. Mohamed, A. El-Gaafary, A. M. Hemeida, “Optimal power flow using moth swarm algorithm”, Electric Power Systems Research 142 (2017) 190–206.
120. Al-Attar A. M., A. A. El-Gaafary, Y.S. Mohamed, A. M. Hemeida, " Multi-Objective States of Matter Search Algorithm for TCSC-based Smart Controller Design" Electric Power Systems Research, Volume 140, November 2016, Pages 874-885.
121. A. A. El-Gaafary, Y.S. Mohamed, A. M. Hemeida, and E. A. Mohamed, "Grey Wolf Optimization for Multi-Input Multi-Output System" **Universal Journal of Communications and Network Vol. 3(1), 2015, pp. 1 – 6**.
122. Makoto Sugimura, T. Senjyu, Narayanan K, P. Mandel, M. Abdel Akher, Ashraf Hemeida, “*Sizing and Operation Optimization for Renewable Energy Facilities with Demand Response in Micro-grid”,* Accepted for oral presentation at the 20th International Conference on Intelligent System Application to Power Systems (ISAP), December, 10-14, 2019, New Delhi, India.

123. Lei Liu;Tomonobu Senjyu;Takeyoshi Kato;Paras Mandal;Ashraf M. Hemeida;Abdul Motin Howlader, "Renewable Energy Power System Frequency Control by using PID controller and Genetic Algorithm", 2020 12th IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC).