IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 26, No. 2, April, 2021, pp. 179-190
Journal homepage: http://iieta.org/journals/isi

Evolutionary Algorithms for Real Time Engineering Problems: A Comprehensive Review

Devineni Gireesh Kumar[1,2], Aman Ganesh[1*], Neerudi Bhoopal[2], Sankaranarayanan Saravanan[2], Madala Prameela[2], Dsnmrao[3], Idamakanti Kasireddy[4]

[1] School of Electronics & Electrical Engineering, Lovely Professional University, Phagwara, Punjab 144411, India
[2] Electrical & Electronics Engineering, B V Raju Institute of Technology, Narsapur 502313, India
[3] Electrical & Electronics Engineering, Gokaraju Rangaraju Institute of Engineering & Technology, Hyderabad 500090, India
[4] Electrical & Electronics Engineering, Vishnu Institute of Technology, Bhimavaram 534201, India

Corresponding Author Email: aman.23332@lpu.co.in

https://doi.org/10.18280/isi.260205

Received: 25 January 2021
Accepted: 29 March 2021

Keywords:
optimization techniques, meta-heuristic algorithms, hybrid optimization techniques, Evolutionary Computation (EVC)

ABSTRACT

This paper presents a variety of contemporary optimization techniques inspired by the real life in nature. Optimization reveals substantial developments in computing systems as well as has come to be the most encouraging strategy for several design applications. The study is conducted on single-objective, multi-objective, and hybrid optimization strategies. These optimization schemes will be of excellent help to organizations to identify optimum criteria and to improve process as well as product high quality. For selected optimization strategies, the process of formulating the objective function/stiffness function for a minimal issue exists. Over the last few years, the most combinatoric problems of all traditional optimization approaches were solved by using metaheuristic algorithms to have optimal solutions for real-time applications. This paper discussed some of the important and feasible optimization scheme and the related algorithms and approaches.

1. INTRODUCTION

In many findings from research, evolutionary computing (EVC) became an important problem-solving approach. A dynamically, self-adaptive, and robust interaction is a major aspect of evolutionary algorithm in contrast with other global optimization methods [1]. Despite being commonly accepted in evolutionary computation for the solution of several major applications, they are only marginally effective sometimes in engineering, commerce, business etc. The unappealing selection of different parameters, visibility, etc., is also blamed. There is an inadequacy of chance that every optimal algorithm will be available to solve all sorts of problems, according to No free lunch theorem [2].

It describes how better performances over a set of problems are specifically compensated by performance over a different set for every algorithm. In exploitation and exploration, the evolutionary nature of the algorithms is determined. These are signs of the need for a hybrid development approach in which the key aspect is optimizing the performance of the immediate evolutionary approach. Evolutionary algorithms have recently gained popularity because they can tackle various real issues such as ambiguity, a noisy environment, incoherence, and misinterpretation [3]. This paper highlights various evolutionary algorithms and then illustrates hybridization of different optimization algorithms for the evolutionary algorithm that have proposed in last few decades. This article also describes the overview of major optimization techniques and their hybrid structures published in the literature for solving the real time engineering problems.

2. EVOLUTIONARY OPTIMIZATION ALGORITHMS

2.1 Genetic Algorithm (GA)

GAs are heuristic search algorithms focused inherently on the natural selection method. Usually this approach is used for designing favorable solutions for wild optimization problems [4, 5]. The cornerstone of GA is a natural selection handling that does not consist of any kind of supplemental features such as computational by-products. The adhering to are some important features of GA that make it a lot more useful in optimization issues:

a) The probability of local minima is reduced.
b) Calculations shift from one state to an additional are declined as well as
c) Fitness evaluation of the search guides for each string.

The benefit of GA approaches is that in certain cases they are contributing to the optimum Pareto frontier globally.

Various GA stages are composed as adheres to:

Step 1: Randomly boot up populace. (This includes initialization for specific population. The designer can determine population size, specific coding and can manage the merging of the algorithm (speed)).
Step 2: Population survey. (The ideal weight may be determined based upon the physical fitness of the populace).
Step 3: Producing children. (Generate brand new people through parental crossover).

Step 4: Impose anomaly to spawn. (Assessing the populace). (Replay steps (3-5) till merging is fulfilled).

Step 5: The size of the population is constant.

The process of Genetic Algorithm is described in Figure 1.

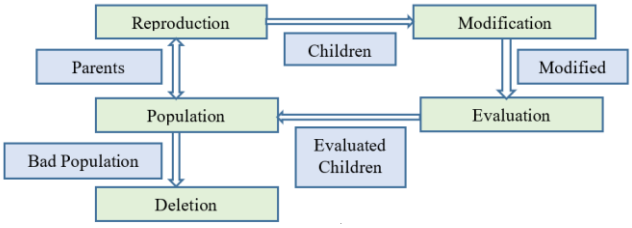

Figure 1. Process of GA

Advantages

1. It can find fit solutions in a very less time. (fit solutions are solutions which are good according to the defined heuristic).
2. The random mutation guarantees to some extent that we see a wide range of solutions.
3. Coding them is easy compared to other algorithms which does the same job.

Disadvantages

1. It is hard for people to come up with a good heuristic which reflects what we want the algorithm to do.
2. It might not find the most optimal solution to the defined problem in all cases.
3. It is also hard to choose parameters like number of generations, population size etc. When we are working even though our heuristic was right, we were not realizing it because we were running for a fewer generation.

The classification of various optimization algorithms for solving engineering problems were represented in Figure 2.

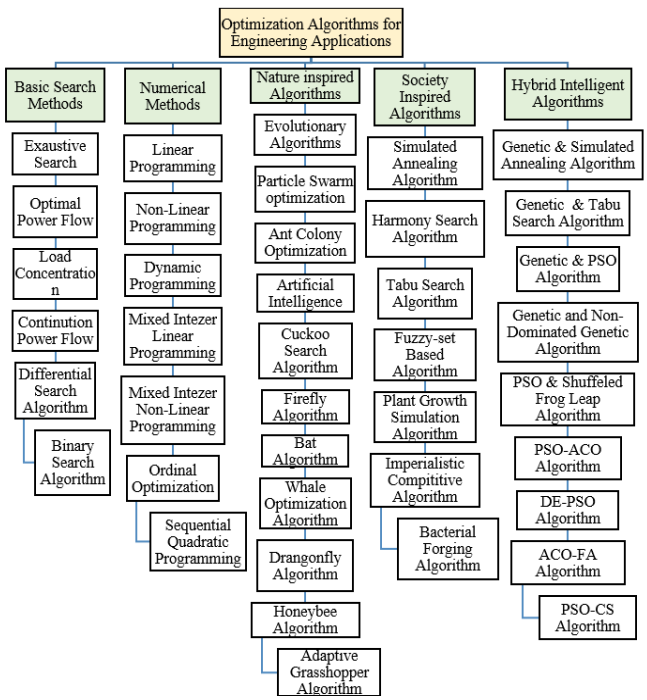

Figure 2. Classification of optimization algorithms for various engineering applications

2.2 Particle Swarm Optimization (PSO)

A bio driven computer search optimization technique called particle swam optimization focuses on birds or fish school social behavior [6-8]. It is analogous to the GA approach in which a populace of arbitrary services at first provided to the machine is a PSO that has no "mutation," "recombination" or "survival of the fittest" operators. Fragments observe individuals traveling around multi-dimensional space. For each bit, the most effective position is acquired with the suitable remedy (fitness) to itself as well as its neighbors. As said, this formula's process starts with a first position as well as speed for every particle, for which the velocities are bound due to non-flying in pointless areas and likewise overrunning prohibitions.

A customized binary particle swarm optimization (BPSO) approach presented a brand-new idea for fixing the OPP issue. BPSO formula is a distinct binary variation of PSO where just 0 as well as 1 value can be drawn from variables [9]. While developing the problem of OPP, the rule ensures the observability of zero injection buses whose neighboring buses knew meaning. Outcomes of the proposed technique as well as different formulations were tested in different situations consisting of standard issue and PMU/branch disturbance. There are lots of PSO versions as well as they can proliferate any time. Figure 3 determines the types of particle swarm optimization techniques.

This technique has major advantages as,

1. Design variables are resistant to scaling.
2. Very limited parameters.
3. It is simple to implement.
4. Non-derivative fitness functions.
5. It is excellent for large-scale optimization.

The drawback is,

1. Multidimensional problems of poor convergence during simple search (Poor capability of local search).

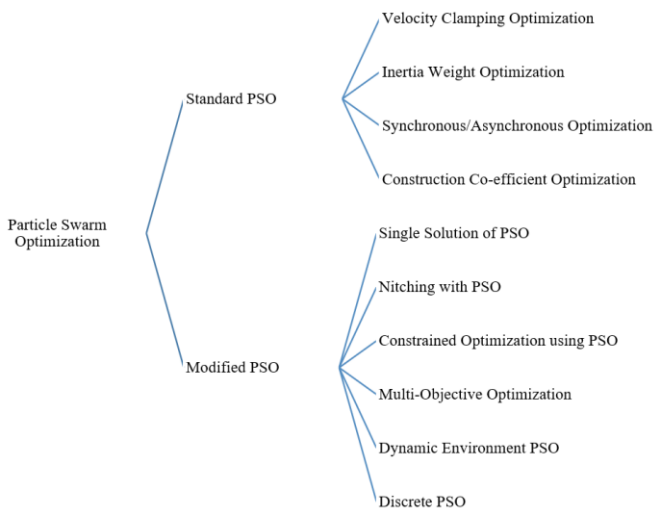

Figure 3. Different forms of PSO

2.3 Simulated annealing (SA)

SA is a complex form of combinatoric optimization in which the current function is randomly updated. The latest service is the worst modification with a reduced chance as the calculation continues [10]. An ideal solution for a complex

problem with combinatory optimization problem to address SA requires a perturbation mechanism in form, price function, operation space, as well as a regeneration schedule [11]. Sufficiency of SA may be achieved where an optimal approach is sought to a significant question of combination optimization to address SA requires a perturbation mechanism in form, price function, operation space, as well as a regeneration schedule comes to finding a near-optimal or optimal service by looking for a massive system and getting great speed. Neighborhood search methods are great at locating regional optima, such as steepest descend method. Troubles take place when the optima international is various from the regional optima is shown in Figure 4. Given that the output worth of all the instant neighboring factors around a regional optimum is worse than it, neighborhood search does not continue while stuck in a regional optima phase.

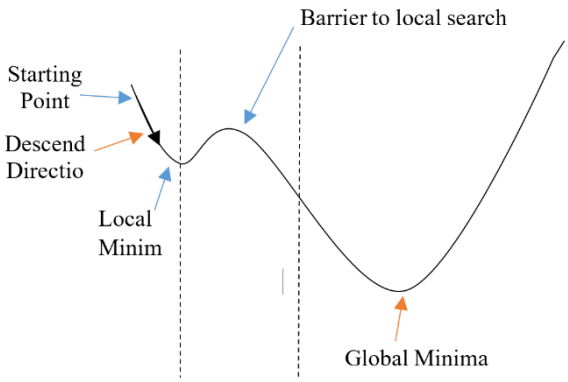

Figure 4. Difficulty in searching global optima

In SA, the different phases are involved which are given as.

Phase 1: Initiate and randomly calculate next position.

Phase 2: Compute different Δ.

Phase 3: If Δ < 0, Assign the current position as next position.

Phase 4: If Δ > 0, Calculate the probability of the position that follows.

phase 5: Select – for the probability of < e (-Δ / temperature), Attribute the next position to the current one.

2.4 Differential evolution (DE)

The concept of differential advancement (DEA) makes use of vectors of the N-dimensional elements to minimize recurring area features. Crossover, selection as well as mutation are the key drivers used to achieve worldwide optimization [12]. This heuristic technique can be utilized thoroughly in different troubles of price feature such as multi-modal, non-linear and non-differentiable functions. Specific advantages of this strategy consist of identical computations, simplicity of use as well as solid convergence properties [13, 14]. This definition considered maximum dimension reliability as well as volunteer PMU failing to reach a completely quantifiable network as well as addressing the OPP problem. Making use of DE formulas obtained from GA caused the idea of NSDE algorithms.

The achievement of unique and full Pareto front and the discovery of several Pareto-optimal remedies were noted as an improvement of this method. The attainment of the minimum variety of PMUs needed for observability of equipment was approached using linear integer programming (ILP), which provided an optimum solution through DE approach. In this concept, 3 drivers were operated that included the option, recombination, and mutation procedure until the stop requirements were reached. Finally, the OPP issue was tackled with and without the injection of null injection considering the validity of faults. Besides making use of PMU in the scheme, traditional measurements are intended to achieve lower prices as well as to obtain a more accurate quote of the company. The algorithm was defined as a global remedy in test systems that were referenced by state estimates. Furthermore, using the generated medication, the most appropriate option has been selected. The recommended DE procedure offered a device for assessing the device's observability for marginal PMUs and their installation within the power system [15]. In contrast to numerous other techniques the results of the design proposed show the small number of PMUs that have been obtained relative to all others.

2.5 Tabu search (TS)

Tabu search is a robust method utilizing several methods, such as linear programming algorithms and heuristic theories. An approach is applied to resolve the coordination and reporting of issues relating to combination optimization [16]. Tabu listing, and it's one of the major components of TS, contains the number of recent states, plus several undesirable states. Some crucial elements of TS are the ambition, diversity as well as summary of a State and its environments. When it is not assembling, there is a reset in TS.

An OPP issue strategy remedy was given regarding accomplishing a totally measurable power system and enough integrity making use of TS based upon a system's linear state estimator design. This easy method of topologically observable evaluation called for loss computer feature based on incidence matrix to resolve the trouble of OPP as well as was highly durable [17]. This approach is also used to change the integer numbers for both convenience and broadband connections to a transparent power system. Some of the observability analyzes using topological approaches while the application of a computational method with TS called Recursive Tabu Search (RTS) was suggested to achieve the completely measurable maximum redundancy relationship [18]. This optimization process has found the best way to use the original solution, mainly through greedy algorithms as residually implemented. This procedure investigated strategies for obtaining minimum numbers of PMUs to resolve the issue of OPP, such as Modified Tabu Quest (MTS). The standard process for TS algorithm is shown in Figure 5.

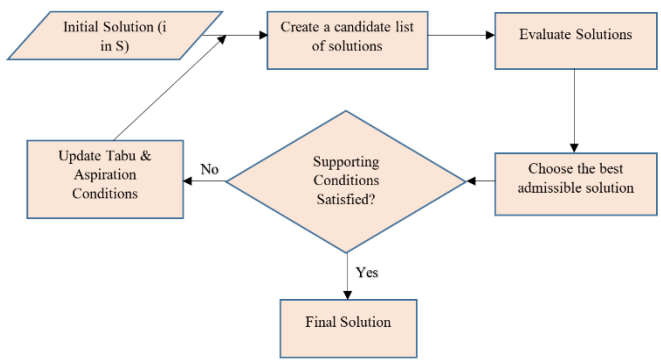

Figure 5. Standard TS algorithm

Specific phases of TS are:

Phase 1: Initial solution generation for X
Phase 2: Develop tabu list
Phase 3: Search for a solution to resolve X

When X * isn't a solution, establish the same X solution x* (non-Tabu list neighbors of X). If x* is not tabu, add x* to X* (at least the aspiration condition is met).

Phase (4): Choose the right one among all solutions (x* in X)

Phase (5): If F(x*) > F(x), the fitness of x is implied by x = x* (F(x)).

Phase (6): Tabu list update (suction criteria).

Phase (7): If the end is reached, halt or loop to Phase (iii).

2.6 Ant colony optimization (ACO)

Ant swarm optimization (ACO) Another principle made use of to offer an optimization issue solution is ant nest optimization (ACO), which originally uses ant populace. The aim of the Ant-colony is to get through the neighboring states through a stochastic, local judgment-optimized controller (plan). Other procedures at ACO are pheromone trail dissipation as well as daemon activities. Using ACO to discover good paths via chart will could computational issues. Most reports given during the Asia-Pacific Power Conference as well as the alternative energy Conference show the optimum issue in placing with the usage of an updated ACO for achieving a visible power network with the least number of PMUs and maximum calculation accuracy [19]. The preliminary search as a graphical academic method for constructing a measuring tree to determine the observability of the network.

Reliable computation as well as equivalence have been noted as attributes of ant swarm system (ACS) between the explorations of brand-new remedy which of aggregated problem discovered. In this article, the development of ACS through versatile stochastic provocation ACS (ASPACS) was proposed to adjust the coefficient of perseverance of the pheromone route (PTPR) and stochastic provocation progress (SPP) [20, 21]. The scope of feasible solutions was propagated by presenting a method that streamlined the accessibility for designers to an expanded plan. There was a comparison in between the impacts of this strategy and the flexible GA and SGA.

Algorithms for optimization may lead to appropriate real-time applications. To fix SA, an ideal service for a wide combinatorial optimization issue requires an in-shape perturbation mechanism, price feature, solution area, and refrigeration routine. Ant Nest Optimization (ACO) Another concept used to present an ant colony optimization system (ACO), that initially uses the ant population, is ant optimization service. The feature of the ant swarm is to pass via adjacent states of the issue by using an ideal controller (policy) for stochastic regional decision, which results in the option for the issue of optimization. Offering OPP services with estimated options and regional remedies considering optimum calculating reliability in power systems.

The merits of ACO is as follows,

1. Efficient travel method with minimal number of nodes for salesmen Problem (TSP).
2. In the early phases of the search, the scornful heuristic helps to locate a relevant solution.
3. The method performs better for TSP than other global methods of optimization.
4. Non-derived algorithm (Healthy choice for different problems).

The demerits of ACO is as follows,

1. It is difficult for theoretical analysis, for instance sequences of random decisions (not independent); Iteration shifts the distribution of probability; work is not theoretical, it is quantitative; convergence is guaranteed, but convergence time is unknown.
2. TSP hard to solve with many nodes (more than 75 cities).

2.7 Whale optimization algorithm

In 2016, WOA is a modern population-based algorithm [22]. This algorithm simulates humpback whales' social behavior. WOA uses a random solutions (people) as well as three rules to update and develop the candidate solutions place in each stage that encircle the prey, spiral updating location, and check for prey, like other population-based algorithms [22, 23].

a) Exploitation Phase: Bubble net attacking

There are two approaches for modeling the behavior of humpback whales in bubble sea, which is called mathematical exploitation.

(1) Encircling Prey: They encircle them after learning the location of the prey. The WOA algorithm thus implies that the present leading candidate solution is or near the optimal target pray the location of optimum design is not identified in the search area. The other search agents then seek to switch their location to the better search agents. The following equations describe this behavior:

$$\vec{X}(t+1) = \vec{X^*}(t) - \vec{A}.\vec{D} \tag{1}$$

$$\vec{D} = \left|\vec{C}.\vec{X^*}(t) - \vec{X}(t)\right| \tag{2}$$

where, $\vec{X^*}(t)$ is the earlier best location for the whale in iteration t. $\vec{X}(t+1)$ is present position of the whale, $\vec{D}$ is a vector distance between pray and whale, and | | indicates absolute value. The coefficient vectors C and A are calculated according to the following:

$$\vec{A} = 2.\vec{a}.\vec{r} + \vec{a} \tag{3}$$

$$\vec{C} = 2.\vec{r} \tag{4}$$

The value of $\vec{a}$ is decreased to apply shrinking in Eq. (3); therefore, the range of oscillation of $\vec{A}$ is also decreased by $\vec{a}$. The $\vec{A}$ value could be lies in $(-a, a)$ interval, where a value is reduced by iterations from 2 to 0. By choosing random values of $\vec{A}$ between (−1, 1), Any search agent may decide the new position somewhere between the agent's original location and the existing best agent location.

(2) Spiral position Updating: After measuring the distance between the whale at (X, Y) and the prey is placed at (X*, Y*). In this case, a spiral approximation is created between the whale's position and the prey in order to trace the loop moving of the humpback whales as follows:

$$\vec{X}(t+1) = e^{bk}.\cos(2\pi k).\vec{D^*} - \vec{X^*}(t) \tag{5}$$

$$\vec{D^*} = \left|\vec{X^*}(t) - \vec{X}(t)\right| \tag{6}$$

where, b is the scalar quantity for the logarithmic spiral and k is a random number inside the range [-1,1]. In the WOA this action affects the role of whales while optimizing. There is a 50% probability of choosing between the shrinking circular system and the spiral pattern, and the following are their elements:

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \vec{A}.\vec{D} & \text{if } p < 0.5 \\ e^{bk}.\cos(2\pi k).\overrightarrow{D^*} - \overrightarrow{X^*}(t) & \text{if } p > 0.5 \end{cases} \quad (7)$$

where, p is an arbitrary number in the range (0, 1).

b) Exploration Phase: Searching Pray

A specific approach based on the vector $\vec{A}$ variances may be used in the search process for the presa called the exploration phase. The whales actively hunt at random to locate their food according to one another's location. Therefore, WOA uses the vector $\vec{A}$ with random values greater or smaller than 1 to force the search agents to move away from the local whale. The location of the search agent is randomly selected by search agent instead of best search agent reorganized throughout the exploration process.

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A}.\vec{D} \quad (8)$$

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_{rand}} - \vec{X}\right| \quad (9)$$

2.8 Honeybee algorithm

Honeybees are one of the social insects most well-studied. In the early years, numbers and combination problems were solved through several experiments aimed at different bee behavior [24]. Bee behavior in nature Community insect colonies may be viewed as a complex framework for capturing and adjusting environmental information. Owing to their specialization, individual insects do not do all the activities during information gathering and adjustment. Basically, all individual insect colonies perform their own morphological labor division [25].

The search algorithm for bees was first developed in 2011 based on the population with a natural forages actions for bees. The simplest version of the algorithm begins with the positioning of scout bees in the search field randomly. Instead it measures the fitness of sites visited by the Scout Bees and selects "chosen bees" and locations visited by bees with the lowest ranking. Near the strategic location, the algorithm then conducts a search, assigning additional bees to the best sites. More bees are recruited at the best e sites more extensely than the other bees. The hierarchical recruiting of bees along with the scouting character is essential. The remaining part of the population are allocated to new solutions around the search area. Such measures are replicated prior to the stop phase. The colony would have two pieces at the end of each replication, the fittest of the patch and the most randomly sent. A neighborhood search is done in conjunction with random searches are used to improve the combination of results and functions [25-27].

Behavior of Bees:

1) Foraging behavior:

a) Searching *for nest site:* Swarming reproduces the most productive colonies. Some queen cells were created to create a new queen at the beginning of the season. The ancient queen abandoned the community for a new colony with half its components until it was found. They are looking for a new site. Scouts are searching for around 12 nest locations. Through dancing, they reveal the different locations of the new nests. The dance standard is related to the nest quality. Thus, over time, until a single site is found, the selected sites are reduced.

b) Searching for food source: First, several "scouts" bees travel and find a source of food in the area. When they do, they come to the hive called the 'dance floor' and talk to others about a dance language. These bees are recruited and forged later. The number of scouts is equal to the detail about the volume of food they receive. This period of discovery is followed by the activity stage. Bee extracts food and measures the volume for a new decision. This chooses from the correct location or leaves the source and returns to hives as ordinary bees.

2) Marriage behavior*:* The Queen is likely to reproduce the behavior of a bee colony. The young Queen will fly on bridal flights after its birth. It will be joined by several drones in a meeting place. In full flight, the queen meets many males, before its sperm theca is finished and it lays eggs after three days. The unidentified egg creates a drone, and the labor or queen makes the fertilized egg based on the consistency of the food provided to the larvae [24].

2.9 Differential search algorithm

DSA is an adaptive meta-heuristic algorithm published in 2012 by Ciicioglu [28]. DSA focuses on the migratory behavior of a living organism, seen in a Brownian random walking model [29]. The availability or capacity of natural food resources can vary in the context of periodic climate change. The living being migrates then to a new location seasonally where essential food supplies are accessible to cope with hunger. DSA has other benefits, such as quick integration and the probability of a minimum global valuation.

By using the following steps, DSA can be described with evolutionary algorithm.

Initialization: Each artificial organism in DSA (i.e. Xa, a = {1, 2, 3… S}) has many participants as the question size (i.e. Xa, b, b = {1, 2, 3… T}). These artificial organisms are the superorganisms given by;

superorganismg = Xa. g = {1, 2, 3…, G}

Here, S represents the individuals in superorganism, T determines the optimization problem dimension and G represents the maximum number of generations. The random method may be used to create every artificial organism member $[p_{(a,\,b)}]$;

$$p_{(a,\,b)} = Low_b + rand \times Up_b - - Low_b$$

Here, rand*n* is the preferred random number between [0, 1] and Low_b, then Up_b is the lower and top limits of the solutions have been established.

Stop site computation: A brownian-like random walking model can explain the mechanism for obtaining a field between artificial organisms [29]. Each single artificial species has been randomly selected to travel to the target contributor to locate the stop site = $[X_{rands,\ shiffling(i)}]$. For a productive migration, it is a very necessary step towards the global

minimum. The location of the stop can be calculated using the following expression:

$$\text{Stopover site} = \text{superorganism} + \text{scale (donor} - \text{superorganism)}$$

Each artificial organism is controlled by its scale value, by its step changes in its position. It can be measured by;

$$\text{scale} = \text{rand g}(2 \times \text{rand1}) \times (\text{rand2} - \text{rand3})$$

The selected random numbers under [0, 1] with rand1, rand2, rand3 and randg is random value within [0, 1], using a uniform distribution.

Computing the participants involved: A stochastic process is used to select members of the artificial organizations of the superorganism who are in the process of finding the stopping location. In the stochastic method the values of parameters $p_1 = 0.3$ x rand4 and $p_2 = 0.3$ x rand5 are used. Here, rand4 and rand5 are the random variables within [0, 1].

Boundary condition: In this cycle the probability occurs that each individual who is part of a stopping place is outside artificial organisms' upper and lower limits. So, the following equation is used to keep such members of the stop site within the boundary search space:

$$\text{stopover site}_{(a,\, b)} = \text{Low}_b + \text{rand}n * (\text{Up}_b - \text{Low}_b)$$

Termination criteria: This stopover substitutes the individual of the artificial organism when the fitness value of the stop individual becomes more productive than the fitness value of the artificial organism. This transforms the site and resumes its movement to the global lowest minimum Superorganism that includes artificial species.

2.10 Cuckoo search algorithm

CS is a new, efficient populational heuristic evolutionary algorithm to solve problems with optimization. CS provides the advantages of fast execution and several control parameters. This model is based on compulsory reproduction in tandem with the variability of other fruit and bird populations, contributing to life loss. A variety of real optimization issues have been tackled with Cuckoo Search algorithm [30].

The following are the search approximation rules.

1) Each cuckoo place one egg at a time and throws it into a randomly selected nest.
2) The best nests can be supplied with high quality eggs to the next generation.
3) A variety of host nests are accessible. A host bird will most likely find an alien egg with the probability P_a [0, 1]. The host bird can take away or leave the nest and construct a new nest.

From the point of view of execution, each nest egg is a solution, and every cuckoo are place only one egg. In this case, an egg, a nest, or a cuckoo will not vary because each nest has an egg which is also one cuckoo. In CS, every nest or egg position is a solution since every nest is equal to one egg. Every solution is randomly generated during the initial process in the $(i+1)^{th}$ generation.. Updated position with the help of,

$$x_i(k+1) = x_i(k) + \alpha \otimes L(\boldsymbol{\lambda})$$

Here $x_i\ (k+1)$ is K+1 generation nest in the population; $x_i(k)$ is the ith population development nest; α is a relevant number showing the behavior proportionate to the scale of the optimization problem; ⊗ represents multiplication in the entry direction; and $L(\boldsymbol{\lambda})$ is the Lévy random search vector distribution.

Lévy flights [31] is an integral part of CS [30] for local and global searches.

$$L(\lambda) \sim u = t^{-\lambda}\ \ (1 < \lambda < 1)$$

In this case, the successive steps of the cuckoo basically constitute a random process following a heavy dose step-long distribution of power law. Lévy can work around the right approach, speeding local scanning, to produce some of the latest solutions. But an extensive randomization, the locations of which should be sufficiently distant from the best solution available today, should generate a significant proportion of new solutions. This strategy means that the system is not captured local optima.

3. MULTI-OBJECTIVE OPTIMIZATION USING A PARETO FRONTIER

A multi - objective optimization system may be used to define a topology for the RP-MII nominee, not necessarily for a design but for a Pareto boundary that exposes design points that are not occupied by certain design points, and thus Pareto is optimal. The explanation for choosing a design point category is that the meanings of the word "largest single feasible" that differ depending on user tastes and specifications.

A feasible x solution is considered optimally Pareto only if there is no other alternative and if it is better objective (within or above the feature definition) than in other objective functions [32]. If there is such a solution, x isn't ideal for Pareto anymore. In certain words, in a certain critical function no changes can be rendered without more regression if Pareto is the optimum solution. The Pareto boundary indicates the converter performance limits and simplifies relations between the various topologies of candidates.

3.1 Non-dominated Sorting Genetic Algorithm-II

Deb et al. [33] developed the 2002 NSGA-II system among the most common heuristic search multi-objective methods. One big difficulty with using multi-objective optimization algorithms like the genetic algorithm is that if no one is influenced by the other. There is no simple means of announcing a stronger or bad option than the other. Therefore, a multi - objective optimization algorithm will consider a target response, which may be similarly successful from Pareto. The Pareto front always needs to be dispersed across the entire field and not limited to a specific one. A Pareto front with such characteristics is found with the NSGA-II algorithm.

Though NSGA-II is using genetic algorithms for its significant reasons, a strong multi-objective optimization requires two different concepts. The following describes these two concepts.

1. Non-dominated Sorting: It uses a ranking concept for assigning each solution a fitness value to simplify the genetic

algorithm selection procedure. First introduced by [34], each solution is essentially ranked by its dominant status. All non-dominated structures are first marked and excluded from the solution pool temporarily. In the remaining solution set, the non-dominated solutions rank 2 and this process continues until no alternative has been discovered.

In NSGA-II the classification every solution is calculated faster than this algorithm (or non-dominated sorting). All non-dominated solutions begin by rank 0, in this method. For each solution, two individuals are then described: Number of solutions n_p that control the p and S_p solutions that exist. Each member of their S_p set (q) is visited for each solution with n_p=0 and their dominion count(n_q) are decreased by 1. Unless the Q is nil, the Q is put in another Q column and is the second unregulated end. It goes so far as to identify all fronts and assign all solutions to each level. Although the NSGA-II creators do not call this classification, the concept as outlined in [34] is essentially the same.

2. Crowding distance: The total crowding difference between two solutions is calculated for all sides of the existing solution. The difference between two solutions for the present solution acts as an approximation for the perimeter of the cuboid, according to [33]. Figure 6 shows the smashed rectangles in a two-target problem.

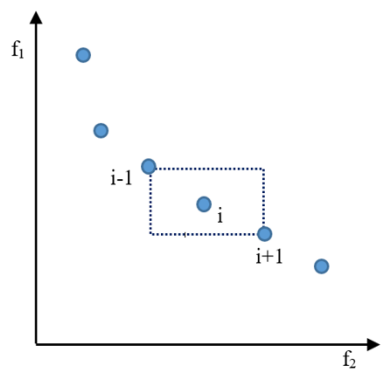

Figure 6. Crowding distance concept [33]

To measure overcrowding frequency, solutions should be separately sorted for each subpopulation. Consider F_i as the sub-population i^{th} sorted. Distance d_{jk} is the distance from j-1 to j+1 on target k. d_{jk} is defined as infinity for F_i solutions with the smallest and largest objective function.

The definition of crowding distance is then described as:

$$d_j = \sum_{k=1}^{m} \frac{d_{jk}}{f_k^{\max} - f_k^{\min}}$$

4. HYBRID OPTIMIZATION ALGORITHMS

4.1 PSO-CS hybrid algorithm

The PSO offers benefits like easy comprehension, simple operation, and quick search. But PSO is easily stuck in the local optimum to solve a large, complex problem. To extend the viability of PSO, this weakness must be overcome [35].

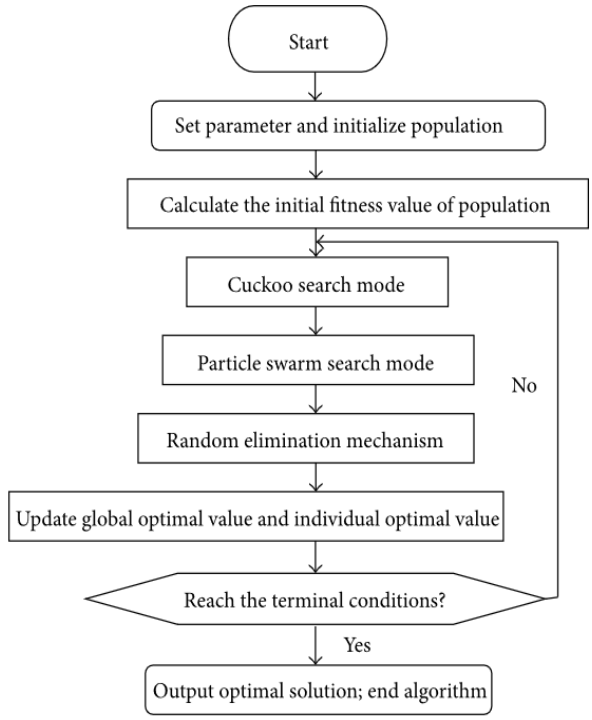

Figure 7. Flowchart of PSO-CS

CS has benefits including little power and good performance, but there are other drawbacks, including sluggish integration and poor accuracy. The high randomness of the Levy flight at CS makes it easy to move from one area to another. Therefore, the algorithm has a strong global search capacity. The algorithm, though, initiates a blind search mechanism that takes into consideration the extreme randomness of flight Lévy and delays the pace of convergence and decreases search performance near the optimum solution substantially. In the CS update process, PSO is introduced to increase the performance of CS. A hybrid PSO-CS algorithm is therefore formed. First PSO-CS uses the search space of Lévy flights and then uses PSO update mode position to speed the particles up to optimally convergence solutions. Simultaneously, CS can escape locally with the random removal process successfully, thus improving the search performance for the optimal solution. The Figure 7 shows a flowchart of PSO-CS. It implements the method.

Terms of the algorithm are as follows.

(1) *Size of the population (size-pop):* There are several persons in the population; the total number of persons with size is the population.

(2) *Fitness:* Fitness is a quality index for individuals. A high fitness value is generally consistent with best solution and vice versa.

(3) *Upper Bound Search Space (U_b) and Lower Bound Search Space (L_b):* U_b and L_b are the top and bottom of the search space for the problem of optimization, respectively.

(4) *Maximum Search Velocity (v_{max}) and Minimum Search Velocity (v_{min}):* The algorithm performs a search and speed is reduced. Let, $v_{max} = a * U_b$ where 'a' is a coefficient of adjustment in the range of (0, 1). In this case $v_{min} = b*L_b$, here 'b' also has the coefficient of adjustment within the range of (0, 1).

(5) *PSO Search Mode:* An individual uses PSO process to update its position and speed in this mode.

(6) *Cuckoo Search Mode:* The CS process allows the

individual to update their position. An individual in CS does not have a formula for improvements in speed and distance when the individual in the PSO search mode has position and pace. In cuckoo search mode, the individual speed is not modified, and the existing velocity is modified with PSO in search mode.

(7) *Discovery Probability:* The host is likely to find foreign eggs through the random removal mechanism in cuckoo search.

4.2 PSO-GA hybrid algorithm

Kuo and Hong [36] have suggested a two-stage soft computing investment strategy. In the first step data envelopment review was used to identify the most efficient assets, while substitutes for the asset classification GA and PSO were introduced in the second phase. Chen and Kurniawan [37] also implemented a two-stage optimizing method to determine optimal process parameters for many consistency features of injection plastic moldings. This study used Taguchi methods, BPNN, GA and PSO-GA configurations for optimized design of the parameter. Nazir et al. [38] have excluded the facial features from the local binary pattern (LBP) and have merged them with the properties of clothing that greatly increased classification accuracy. Throughout the subsequent step, PSO and GA were merged to pick the most appropriate features that accurately reflected the individual and minimized the size of the data. Vidhya and Kumar [39] developed a new channel estimates for PSO and GA multiplexing systems (MIMO-OFDM) MIMO orthogonal frequency division. In all replication, crossover values and measured iterations, studies demonstrate that it is easier than the LS and MMSE approaches the solution being suggested. Xiao et al. [40] had three different types of network-based neural models developed: Elman network; GRNN; and WNN network, made up of three uncross lapping training sets. The analytical findings of the ANNs-PSO-GA method indicated that the forecast efficiency was substantially improved over other regression and linear combo regression. The new selection method, which has the GA and PSO integration, is introduced by Ghamisi and Benediktsson [41]. As a fitness value, the general accuracy of a validation samples support vector (SVM) classification was used. The new method was applied on the popular hyper-spectral data collection of Indian Pines. Results reported that the latest method has optimized the classification accuracy of most useful apps within an appropriate processing period, without allowing users to determine a priori the number of required functionalities.

4.3 PSO-TS hybrid algorithm

The Nonlinear Simplex Method (NSM) implemented into PSO to speed up its transition [42]. They incorporated TS in PSO to provide local solutions to the Tabu object in the region. The organic PSO, NSM and TS algorithm were composed by the PSO hybrid algorithm. A new PSO form based on the TS concept was presented by Nakano et al. [43]. A method of combining PSO and TS points is suggested for the Tabu list PSO (TL-PSO). This method saved p*best* history in a Tabu list. If a particle had a decreased searching capability, a p*best* of history was selected for updating from the historical values. This activated each particle and the swarm 's search capability advanced. Zhang et al. [44] also introduced the manufacturing iron-steel design model focused on the concepts of the manufacture of goods or materials. To solve this nonlinear integer problem the author developed a hybrid PSO and TS algorithm, suggest new hurtful laws to correct unfeasible solutions. Ktari and Chabchoub [45] suggested a method, to obtain a better discrete version of the PSO, several features inspired by TS have been incorporated into the Critical Particle Swarm Optimization Queen (EPSOq). Wang et al. [46] concentrated on preserving the distribution system in a long-term approach by presenting operating information. In specific, Wang et al. [46] concentrated on preserving the distribution system for the long term by supplying organizational information. Numerical findings showed that the proposed approach will prepare for the long-term maintenance of smart grid delivery networks on economic and effective terms.

4.4 PSO-ACO hybrid algorithm

Chen and Chien [47] introduced the latest approach called the genetic model ant colony scheme with particle swarm optimization strategies for TSP resolution. The test findings revealed that the overall solution and the average variance in percentage of the solution of the system suggested were greater than the actual approach. The MRCMPSP features were considered in Xiao et al. [48]. They used the division of work of the colony to first create a layout of the priority task. So, the optimum preparation has been enhanced with strengthened PSO. Both local and global search abilities were integrated in the above two algorithms. The hybrid model to assess the demand in Turkey for energy by PSO and ACO is established by Kiran et al. [49]. PSO has been designed to address persistent problems of optimization; ACO has been used to automate discreetly. The hybrid system PSO and ACO were developed for the purpose of estimating energy needs by using gross domestic products, individuals, imports and exports. To enhance search capabilities, Huang et al. [50] have used ACOR in their PSO work, which has explored four forms of hybridization as follows: sequential approach, parallel approach, phromone-particle approach, and best global exchange approach. The series method with the expanded pheromone table was superior to the other approximations between the four hybridization approaches, because the extended table diversified the development of new solutions for ACOR and PSO that avoided trapping into the optimum location. The new Hybrid Swarm procedure (HAP) was used by Rahmani et al. [51] to estimate energy production from the actual Binaloud, Iran wind farm. The approach was hybridisation of the Swarm Intelligency community ACO and PSO, two metaheuristic approaches. The two algorithms' hybridization to optimize the expected model led to an improved output outcome with a high convergence rate. The new approach to optimization based on the multi-objective PSO and Fuzzy ACO was illustrated by Elloume et al. [52]. These two strategies must be paired with the highest particle in the Fuzzy Ant algorithm to form a new system called hybrid MOPSO with FACO as the best local PSO. This hybridization addressed the multi - objective problem based on the shortest time and trajectory parameters.

4.5 PSO-SA hybrid algorithm

For cell assigning solution in cnnosnanowireMOLecular Hybrid (CMOL), The fusion of PSO and SA was proposed by Sait et al. [53]. Tests have shown that the hybrid method introduced is a safer alternative for buffer count in acceptable

time. In the current development of SVMs, through altering the exercise feature Jiang and Zou [54] suggested an enhanced optimization parameter strategy focused on traditional PSO algorithms. This process was then combined with the Global Search Algorithm of SA in order to prevent local convergence of PSO algorithms. This technique provided excellent results in the interpretation of medical images and obtained significant precision in the identification of clinical diseases in ROC curves. Niknam et al. [55] suggested the hybrid model combining PSO and SA for addressing the problem of DOPF when considering the restricted zones, valve points and ramp rate limits. The hybrid PSO-SA algorithm will effectively search and test the space for solutions by utilizing PSO and SA algorithms. In order to optimize the fuel management, Khoshahval et al. [56] built the latest P-PSOSA parallel algorithm, which defines two separate fitness functions to optimize the multiplication factor and at the same time decrease the power peak factor. P-PSOSA numerical results indicated that the algorithm proposed had tremendous power to achieve almost global core functional pattern during acceptable use. Du et al. [57] applied a hybrid algorithm that uses stronger PSO and SA algorithms to address the issue of resource constraints. The adaptive weight and synchronous reduction approach and SA of the Hybrid algorithms have been used to overcome shortcomings of premature PSO convergence. Zhang et al. [58] implemented a better technique for the decomposition of arbitrary structural elements. In the reformatted, SA-PSO method, they introduced a new SA and PSO combination. Geng et al. [59] implemented the Robust VSVR model to predict port performance. A chaotic virtual annealing PSO algorithm was proposed to look for the parameters most suitable for the RSVR model.

4.6 PSO-DE hybrid algorithm

The two-phase modeling approach was proposed by Maione and Punzi [60]. Firstly, DE defined the integral and derivative fractional actions that meet the necessary performative time-domain requirements. Furthermore, PSO has defined irrational marginal drivers with pole to zero interconnections as low natural, stable, minimal phase functions. Comprehensive time and frequency calculations demonstrate the efficacy of the suggested solution. The path preparation of unmanned aircraft (uAV) at sea is Fu et al. [61] proposed the QPSO hybrid DE. Experimental findings have shown that UAV can efficiently generate higher quality pathways than other tested algorithms of optimization. The efficient design of Lowpass- and Highpass-FIR filters using a new ADESO algorithm was presented by Vasundhara et al. [62], a hybrid fitness-based adaptive DE/PSO fitness algorithm. ADEPSO overcame all individual drawbacks of algorithms and has been used to build linear FIR filters with low pass and high pass. The results of the simulation have shown that in combination with PSO, ADE and DE have outperformed ADEPSO not in terms of magnitude response, but even as regards the pace of convergence. By improving the PSO-DE equilibrium (HPSO-DE), The new adaptive algorithm based on the DE and PSO was developed by Yu et al. [63]. The community was distributed around the local optima through an acceptable mutation in current populations. The HPSO-DE gained from PSO and DE and preserved population diversity. The output of HPSO-DE was competitive compared to PSO, DE and their variants. Wang et al. [64] presented the stable hybrid metaheuristic optimization method to address numerical optimization problems by introducing DE mutation operators to the Adaptive PSO (APSO) algorithms. The latest CSHPSO, suggested by Yadav and Deep [65] for restricted optimization issues, was the result of the hybridized DE approach to the PSO shrinking hypersphere (SHPSO). To use SHPSO, the first subwarms and DE the second subwarm were subdivided into two sub-wares. Experiments have shown CSHPSO to be a promising new co-swarm PSO to solve any truly restricted problems of optimization.

4.7 ACO-FA algorithm

A modern program that incorporates the functionality of two smart algorithms of natural significance, the ant colony (ACO) and firefly algorithms (FA) [66]. ACO's main attribute is how colonies of actual ants locate a nutritional supply. Ant colony algorithms have demonstrated their utility to look for alternatives in isolated and continuing instances globally. FA's key function is the flight and contact of real fireflies with one another. Swarm-based algorithms are very efficient when it comes in complex continuous areas to find almost optimal solutions. Normal methodologies, generally, provide an outstanding search capacity. Thus, two methodologies for finding complex solutions are expected to be combined with good heuristic rules. Secondly, other basic functions of these algorithms are studied that deal with the search capabilities. In fact, comparative findings are provided using alternate mechanisms. The application area is concerned with a complex portfolio management issue whose objective is to maximize the financial ratio according to the index tracking capability of the portfolio created. This research primarily aims at stressing the significance and reliability of hybrid natural intelligent systems and offering deeper visibility into the functionalities of the major algorithms mechanisms.

4.8 GA-SA algorithm

Ganesh and Pünniya Moorthy proposed the hybrid GA – Simulation Annealing (SA) algorithm for ongoing joint performance problems [67]. GA's ability to build the global solution and to allow SA to refine each specific solution locally is the inspiration behind the GA–SA combination [67]. In two phases, the hybrid algorithm runs: The GA creates the initial solutions on a random basis in the first step. To obtain new and potentially optimal solutions, GA operates the solution with restricted, crossover and mutation operators. Every solution is sent by GA to be improved after each generation in the second phase of SA. The community development scheme used in SA is a simple incorporation process. If the SA is done with a GA solution, that GA solution would be transferred to SA. This pattern repeats until the completion of all GA implementations in one generation. When all approaches are applied by SA in a GA process, GA approaches are the strongest solutions in population size from SA for the next time frame. The GA-SA exchange is pursued until the end of the all generations [67].

5. CONCLUSIONS

This paper presented different single-objective, multi-objective optimization schemes to solve engineering problems. As per no free lunch theorem, each of these are suffering with some drawbacks like convergence rate, no of iterations, initial

guess, local and global optima problems. Therefore, it essential to hybridize by combing the advantages of any two algorithms for better optimal solution. Several combinational hybrid optimization methods are described and the process to adopt the hybridization is also presented in this paper. This literature concludes that the effective optimal solution of engineering problems can be obtained with the hybrid optimization schemes.

REFERENCES

[1] Grosan, C., Abraham, A. (2007). Hybrid evolutionary algorithms: Methodologies, architectures, and reviews. Studies in Computational Intelligence, 75: 1-17. https://doi.org/10.1007/978-3-540-73297-6_1

[2] Adam, S.P., Alexandropoulos, S., N., Pardalos, P.M., Vrahatis, M.N. (2019). No free lunch theorem: A review. Springer Optimization and Its Applications, 145: 57-82. https://doi.org/10.1007/978-3-030-12767-1_5

[3] Estudillo, A.C.M., Martínez, C.H., Estudillo, F.J.M., Pedrajas, G.C. (2006). Hybridization of evolutionary algorithms and local search by means of a clustering method. IEEE Transactions on Systems, Man and Cybernetics, Part B, 36(3): 534-545. https://doi.org/10.1109/tsmcb.2005.860138

[4] Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization & Machine Learning. Reading, MA: Addison-Wesley. https://doi.org/10.5860/choice.27-0936

[5] Man, K.F., Tang, K.S., Kwong, S. (1996). Genetic algorithms: Concepts and applications [in engineering design]. In IEEE Transactions on Industrial Electronics, 43(5): 519-534. https://doi.org/10.1109/41.538609

[6] Parsopoulos, K.E., Vrahatis, M.N. (2002). Recent approaches to global optimization problems through particle swarm optimization. Natural Computing, 1: 235-306.

[7] Kennedy, J., Eberha, R.C. (1995). Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks, pp. 1942-1948. https://doi.org/10.1023/A:1016568309421

[8] Clerc, M., Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. In IEEE Transactions on Evolutionary Computation, 6(1): 58-73. https://doi.org/10.1109/4235.985692

[9] Hajian, M., Ranjbar, A.M., Amraee, T., Mozafari, B. (2011). Optimal placement of PMUs to maintain network observability using a modified BPSO algorithm. International Journal of Electric Power and Energy Systems, 33: 28-34. https://doi.org/10.1016/j.ijepes.2010.08.007

[10] Abramson, D., Krishnamoorthy, M., Dang, H. (1999). Simulated annealing cooling schedules for the school timetabling problem. Asia-Pacific Journal of Operational Research, 16: 1-22.

[11] Yu, X.G., Zhan, D.C. (2009). Genetic simulated annealing algorithm for resource-constrained project scheduling problem. Computer Engineering and Applications, 45(24): 17-20.

[12] Ali, M.M. (2011). Differential evolution with generalized differentials. Journal of Computational and Applied Mathematics, 235: 2205-2216. https://doi.org/10.1016/j.cam.2010.10.018

[13] Das, S., Abraham, A., Chakraborty, U.K., Konar, A. (2009). Differential evolution using a neighborhood-based mutation operator. In IEEE Transactions on Evolutionary Computation, 13(3): 526-553. https://doi.org/10.1109/TEVC.2008.2009457

[14] Kaelo, P., Ali, M.M. (2006). A numerical study of some modified differential evolution algorithms. European Journal of Operational Research, 169(3): 1176-1184. https://doi.org/10.1016/j.ejor.2004.08.047

[15] Adeyemo, J.A., Otieno, F.A.O. (2009). Multi-objective differential evolution algorithm for solving engineering problems. Journal of Applied Sciences, 9(20): 3652-3661. https://doi.org/10.3923/jas.2009.3652.3661

[16] Brandãoa, J., Egleseb R. (2008). A deterministic tabu search algorithm for the capacitated arc routing problem. Computers & Operations Research, 35: 1112-1126. https://doi.org/10.1016/j.cor.2006.07.007

[17] Mittal, M.L., Soni, G., Joshi, D. (2018). A tabu search algorithm for simultaneous selection and scheduling of projects. Advances in Intelligent Systems and Computing, 714. https://doi.org/10.1080/00207543.2014.910628

[18] Koutsoukis, N.C., Manousakis, N.M., Georgilakis, P.S., Korres, G.N. (2013). Numerical observability method for optimal phasor measurement units placement using recursive Tabu search method. The Institution of Engineering and Technology, 7(4): 347-356. https://doi.org/10.1049/iet-gtd.2012.0377

[19] Colorni, A., Dorigo, M., Maniezzo, V. (1991). Distributed optimization by ant colonies. Proceedings of Proceedings of ECAL91 - European Conference on Artificial Life, 132: 132-142.

[20] Dorigo, M., Maniezzo, V., Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. In IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26(1): 29-41. https://doi.org/10.1109/3477.484436

[21] Dorigo, M., Stützle, T. (2019). Ant colony optimization: overview and recent advances. In Handbook of Metaheuristics, M. Gendreau and J.-Y. Potvin, Eds., 272: 227-263. https://doi.org/10.1007/978-1-4419-1665-5_8

[22] Mirjalili, S., Lewis, A. (2016). The whale optimization algorithm. Advances in Engineering Software, 95: 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008

[23] Trivedi, I.N., Pradeep, J., Narottam, J., Arvind, K., Dilip, L. (2016). Novel adaptive whale optimization algorithm for global optimization. Indian Journal of Science and Technology, 9(38). https://doi.org/10.17485/ijst/2016/v9i38/101939

[24] Bitam, S., Batouche, M., Talbi, E. (2010). A survey on bee colony algorithms. 2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), Atlanta, GA, USA, pp. 1-8. https://doi.org/10.1109/IPDPSW.2010.5470701

[25] Aghazadeh, F., Meybodi, M.R. (2011). Learning Bees Algorithm For optimization. IACSIT Press, Singapore, vol. 18.

[26] Pham, D.T., Kog, E., Ghanbarzadeh, A., Otri, S., Rahim, S., Zaidi, M. (2006). The bees algorithm–a novel tool for complex optimization problems. Proceedings of the Intelligent Production Machines and Systems (IPROMS)

Conference, pp. 454-461. https://doi.org/10.1016/B978-008045157-2/50081-X
[27] Civicioglu, P. (2013). Backtracking search optimization algorithm for numerical optimization problems. Appl. Math. Comput., 219(15): 8121-8144. https://doi.org/10.1016/j.amc.2013.02.017
[28] Civicioglu, P. (2012). Transforming geocentric Cartesian coordinates to geodetic coordinates by using differential search algorithm. Comput. Geosci., 46(15): 229-247. https://doi.org/10.1016/j.cageo.2011.12.011
[29] Trianni, V., Tuci, E., Passino, K.M. (2011). Swarm cognition: An interdisciplinary approach to the study of self-organising biological collectives. Swarm Intell., 5(1): 3-18. https://doi.org/10.1007/s11721-010-0050-8
[30] Yang, X.S., Deb, S. (2010). Engineering optimization by Cuckoo search. Int. J. Math. Modell. Neumeric Opt., 1(4): 330-343. https://doi.org/10.1504/IJMMNO.2010.035430
[31] Cui, Z.H., Sun, B., Wang, G.G., Xue, Y., Chen, J.J. (2017). A novel oriented cuckoo search algorithm to improve DV-Hop performance for cyber–physical systems. Journal of Parallel and Distributed Computing, 103: 42-52. https://doi.org/10.1016/j.jpdc.2016.10.011
[32] Coelle, C.A.C., Dhaenens, C., Jourdan, L. (2010). Advances in Multi-Objective Nature Inspired Computing. Berlin; Heidelberg: Springer-Verlag. https://doi.org/10.1007/978-3-642-11218-8
[33] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. In IEEE Transactions on Evolutionary Computation, 6(2): 182-197. https://doi.org/10.1109/4235.996017
[34] Srinivas, N., Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation, 2(3): 221-248. https://doi.org/10.1162/evco.1994.2.3.221
[35] Ding, J.J., Wang, Q.J., Zhang, Q., Ye, Q.B., Ma, Y. (2019). A hybrid particle swarm optimization-cuckoo search algorithm and its engineering applications. Mathematical Problems in Engineering, 2019: 5213759. https://doi.org/10.1155/2019/5213759
[36] Kuo, R.J., Hong, C.W. (2013). Integration of genetic algorithm and particle swarm optimization for investment portfolio optimization. Applied Mathematics and Information Sciences, 7(6): 2397-2408.
[37] Chen, W.C., Kurniawan, D. (2014). Process parameters optimization for multiple quality characteristics in plastic injection molding using Taguchi method, BPNN, GA, and hybrid PSO-GA. International Journal of Precision Engineering and Manufacturing, 15(8): 1583-1593. https://doi.org/10.1007/s12541-014-0507-6
[38] Nazir, M., Majid-Mirza, A., Ali-Khan, S. (2014). PSO-GA based optimized feature selection using facial and clothing information for gender classification. Journal of Applied Research and Technology, 12(1): 145-152. https://doi.org/10.1016/S1665-6423(14)71614-1
[39] Vidhya, K., Kumar, K.R.S. (2014). Channel estimation of MIMO-OFDM system using PSO and GA. Arabian Journal for Science and Engineering, 39(5): 4047-4056. https://doi.org/10.1007/s13369-014-0988-8
[40] Xiao, Y., Xiao, J., Lu, F., Wang, S. (2014). Ensemble ANNs-PSO-GA approach for day-ahead stock E-exchange prices forecasting. International Journal of Computational Intelligence Systems, 7(2): 272-290. https://doi.org/10.1080/18756891.2013.756227
[41] Ghamisi, P., Benediktsson, J.A. (2015). Feature selection based on hybridization of genetic algorithm and particle swarm optimization. IEEE Geoscience and Remote Sensing Letters, 12(2): 309-313. https://doi.org/10.1109/LGRS.2014.2337320
[42] Li, Z.C., Zheng, D.J., Hou, H.J. (2010). A hybrid particle swarm optimization algorithm based on nonlinear simplex method and tabu search. In Advances in Neural Networks, 126-135. https://doi.org/10.1007/978-3-642-13278-0_17
[43] Nakano, S., Ishigame, A., Yasuda, K. (2010). Consideration of particle swarm optimization combined with tabu search. Electrical Engineering in Japan, 172(4): 31-37. https://doi.org/10.1541/ieejeiss.128.1162
[44] Zhang, T., Zhang, Y.J., Zheng, Q.P., Pardalos, P.M. (2011). A hybrid particle swarm optimization and tabu seach algorithm for order planning problems of steel factories based on the make-to-stock and make-to-order management architecture. Journal of Industrial and Management Optimization, 7(1): 31-51. https://doi.org/10.3934/jimo.2011.7.31
[45] Ktari, R., Chabchoub, H. (2014). Essential particle swarm optimization queen with tabu search for MKP resolution. Computing, 95(9): 897-921. https://doi.org/10.1007/s00607-013-0316-2
[46] Wang, J., Lu, J., Bie, Z., You, S., Cao, X. (2014). Long-term maintenance scheduling of smart distribution system through a PSO-TS algorithm. Journal of Applied Mathematics, 2014: 694086. https://doi.org/10.1155/2014/694086
[47] Chen, S.M., Chien, C.Y. (2011). Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques. Expert Systems with Applications, 38(12): 14439-14450. https://doi.org/10.1016/j.eswa.2011.04.163
[48] Xiao, R., Chen, W., Chen, T. (2012). Modeling of ant colony's labor division for the multi-project scheduling problem and its solution by pso. Journal of Computational and Theoretical Nanoscience, 9(2): 223-232. https://doi.org/10.1166/jctn.2012.2016
[49] Kiran, M.S., Özceylan, E., Gündüz, M., Paksoy, T. (2012). A novel hybrid approach based on Particle Swarm Optimization and Ant Colony Algorithm to forecast energy demand of Turkey. Energy Conversion and Management, 53(1): 75-83. https://doi.org/10.1016/j.enconman.2011.08.004
[50] Huang, C.L., Huang, W.C., Chang, H.Y., Yeh, Y.C., Tsai, C.Y. (2013). Hybridization strategies for continuous ant colony optimization and particle swarm optimization applied to data clustering. Applied Soft Computing Journal, 13(9): 3864-3872. https://doi.org/10.1016/j.asoc.2013.05.003
[51] Rahmani, R., Yusof, R., Seyedmahmoudian, M., Mekhilef, S. (2013). Hybrid technique of ant colony and particle swarm optimization for short term wind energy forecasting. Journal of Wind Engineering and Industrial Aerodynamics, 123: 163-170. https://doi.org/10.1016/j.jweia.2013.10.004
[52] Elloumi, W., Baklouti, N., Abraham, A., Alimi, A.M. (2014). The multi-objective hybridization of particle swarm optimization and fuzzy ant colony optimization.

Journal of Intelligent and Fuzzy Systems, 27(1): 515-525. https://doi.org/10.3233/IFS-131020

[53] Sait, S.M., Sheikh, A.T., El-Maleh, A.H. (2013). Cell assignment in hybrid CMOS/nanodevices architecture using a PSO/SA hybrid algorithm. Journal of Applied Research and Technology, 11(5): 653-664. https://doi.org/10.1016/S1665-6423(13)71573-6

[54] Shah, U. (2018). A hybrid approach of ANN with PSO for classification problems. TENCON 2018 - 2018 IEEE Region 10 Conference, Jeju, Korea. https://doi.org/10.1109/TENCON.2018.8650351

[55] Niknam, T., Narimani, M.R., Jabbari, M. (2013). Dynamic optimal power flow using hybrid particle swarm optimization and simulated annealing. International Transactions on Electrical Energy Systems, 23(7): 975-1001. https://doi.org/10.1002/etep.1633

[56] Khoshahval, F., Zolfaghari, A., Minuchehr, H., Abbasi, M.R. (2014). A new hybrid method for multi-objective fuel management optimization using parallel PSO-SA. Progress in Nuclear Energy, 76: 112-121. https://doi.org/10.1016/j.pnucene.2014.05.014

[57] Shou, Y.Y., Li, Y., Lai, C.T. (2015). Hybrid particle swarm optimization for preemptive resource-constrained project scheduling. Neurocomputing, 148: 122-128. https://doi.org/10.1016/j.neucom.2012.07.059

[58] Zhang, Y., Wang, S., Sun, Y., Ji, G., Phillips, P., Dong, Z. (2014). Binary structuring elements decomposition based on an improved recursive dilation-union model and RSAPSO method. Mathematical Problems in Engineering, 2014: 272496. https://doi.org/10.1155/2014/272496

[59] Geng, J., Li, M.W., Dong, Z.H., Liao, Y.S. (2015). Port throughput forecasting by MARS-RSVR with chaotic simulated annealing particle swarm optimization algorithm. Neurocomputing, 147: 239-250. https://doi.org/10.1016/j.neucom.2014.06.070

[60] Maione, G., Punzi, A. (2013). Combining differential evolution and particle swarm optimization to tune and realize fractional-order controllers. Mathematical and Computer Modelling of Dynamical Systems, 19(3): 277-299. https://doi.org/10.1080/13873954.2012.745006

[61] Fu, Y., Ding, M., Zhou, C., Hu, H. (2013). Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization. In IEEE Transactions on Systems, Man, and Cybernetics: Systems, 43(6): 1451-1465. https://doi.org/10.1109/TSMC.2013.2248146

[62] Mandal, V.D., Kar, R., Ghoshal, S.P. (2014). Digital FIR filter design using fitness-based hybrid adaptive differential evolution with particle swarm optimization. Natural Computing, 13(1): 55-64. https://doi.org/10.1007/s11047-013-9381-x

[63] Yu, X., Cao, J., Shan, H., Zhu, L., Guo, J. (2014). An adaptive hybrid algorithm based on particle swarm optimization and differential evolution for global optimization. The Scientific World Journal, 2014: 215472. https://doi.org/10.1155/2014/215472

[64] Wang, G.G., Hossein Gandomi, A., Yang, X.S., Hossein Alavi, A. (2014). A novel improved accelerated particle swarm optimization algorithm for global numerical optimization. Engineering Computations, 31(7): 1198-1220. https://doi.org/10.1108/EC-10-2012-0232

[65] Yadav, A., Deep, K. (2014). An efficient co-swarm particle swarm optimization for non-linear constrained optimization. Journal of Computational Science, 5(2): 258-268. https://doi.org/10.1016/j.jocs.2013.05.011

[66] Rizk-Allah, R.M., Zaki, E.M. (2013). Hybridizing ant colony optimization with firefly algorithm for unconstrained optimization problems. Applied Mathematics and Computation, 224: 473-483. https://doi.org/10.1016/j.amc.2013.07.092

[67] Ganesh, K., Punniyamoorthy, M. (2014). Optimization of continuous-time production planning using hybrid genetic algorithms-simulated annealing. International Journal of Advanced Manufacturing Technology, 26(1): 148-154. https://doi.org/10.1007/s00170-003-1976-4