

Received July 28, 2020, accepted August 5, 2020, date of publication August 10, 2020, date of current version August 21, 2020. *Digital Object Identifier* 10.1109/ACCESS.2020.3015206

# An Enhanced Firefly Algorithm Using Pattern Search for Solving Optimization Problems

FAZLI WAHID<sup>1</sup>, M. SULTAN ZIA<sup>2</sup>, RAO NAVEED BIN RAIS<sup>3</sup>, MUHAMMAD AAMIR<sup>®4</sup>, UMAIR MUNEER BUTT<sup>®5,2</sup>, MUBASHIR ALI<sup>®6,2</sup>, ADEEL AHMED<sup>®7</sup>, (Graduate Student Member, IEEE), IMRAN ALI KHAN<sup>®8</sup>, AND OSMAN KHALID<sup>®8</sup>

<sup>1</sup>Department of Information Technology, The University of Haripur, Haripur 22620, Pakistan

<sup>2</sup>Faculty of Computer Science and Information Technology, The University of Lahore, Gujrat Campus, Lahore 50700, Pakistan

<sup>3</sup>Department of Electrical and Computer Engineering, College of Engineering and Information Technology, Ajman University, Ajman 34, UAE

<sup>4</sup>Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Johor 86400, Malaysia

<sup>5</sup>School of Computer Science, Universiti Sains Malaysia George Town 11800, Malaysia

<sup>6</sup>School of Engineering and Applied Sciences, The University of Bergamo, 24129 Bergamo, Italy

<sup>7</sup>Department of Computer Science, Quaid-i-Azam University, Islamabad 15320, Pakistan

<sup>8</sup>Department of Computer Science, COMSATS University Islamabad, Abbottabad Campus, Khyber Pakhtunkhwa 22060, Pakistan

Corresponding author: Muhammad Aamir (aamir@uthm.edu.my)

**ABSTRACT** Firefly Algorithm (FA) is one of the most recently introduced stochastic, nature-inspired, meta-heuristic approaches used for solving optimization problems. The conventional FA use randomization factor during generation of solution search space and fireflies position changing, which results in imbalanced relationship between exploration and exploitation. This imbalanced relationship causes in incapability of FA to find the most optimum values at termination stage. In the proposed model, this issue has been resolved by incorporating PS at the termination stage of standard FA. The optimized values obtained from the FA are set as the initial starting points for the PS algorithm and the values are further optimized by PS to get the most optimal values or at least better values than the values obtained by conventional FA during its maximum number of iterations. The performance of the newly developed FA-PS model has been tested on eight minimization functions and six maximization functions by considering various performance evaluation parameters. The results obtained have been compared with other optimization algorithms namely genetic algorithm (GA), standard FA, artificial bee colony (ABC), ant colony optimization (ACO), differential equations (DE), bat algorithm (BA), grey wolf optimization (GWO), Self-Adaptive Step Firefly Algorithm (SASFA), and FA-Cross algorithm in terms of convergence rate and various numerical performance evaluation parameters. A significant improvement has been observed in the solution quality by embedding PS in the standard FA at the termination stage. The result behind this improvement is the better exploration and exploitation of the solution search space at this stage.

**INDEX TERMS** Firefly algorithm, pattern search, rich exploration and exploitation, optimization problems, fast convergence, minimization and maximization functions.

#### I. INTRODUCTION

Optimization is the process of finding the solution of a problem with the most cost-effective or highest possible achievable performance using the resources in hand by minimizing the undesired factors and maximizing the desired ones. The optimization can be as simple process as taking a dinner at a restaurant with all the basic nutrients required for a human body with the least possible cost to the investment of billions of dollars for carrying out space research. The solution found

The associate editor coordinating the review of this manuscript and approving it for publication was Huiling Chen<sup>(D)</sup>.

with the highest possible benefit and least possible cost is called an optimal solution. Finding the most optimal solution from few alternatives is considered to be one of the most complicated processes [1], [2]. In the modern fast growing information and communication era, the development of algorithms, techniques and approaches for solving various types of optimization problems is one of the most focused and attractive areas among researchers' community.

The phenomenon behind this remarkable attraction is the indefensible hype of increase both in the solution cost and decrease in the resources for solving these complicated problems prevailing in academia and industries. Therefore,

many artificial intelligence (AI) techniques [3], evolutionary approaches [4] and swarm intelligence (SI) algorithms [5] have been developed for solving various types of optimization algorithms. Based on the inspirations taken from behaviors, characteristics, inter-coordination and working mechanisms of different natural creations, various types of AI techniques have been developed targeting numerous areas according to the problem nature. The major AI techniques include expert systems (ES) [6], artificial neural network (ANN) [7]-[9], fuzzy logic (FL) [10], artificial immune system (AIS) [11], generalized regression neural network (GRNN) [12], genetic algorithms (GA) [13], Henry gas solubility Optimization (HGSO) [14], Slime mould algorithm (SMA) [15], Equilibrium optimizer (EO) [16] and genetic programming (GP) [17] which can be used for solving optimization problems or assist other optimization algorithms in solving these problems.

Moving forward, a subfield of AI known as evolutionary intelligence (EI) is another family of techniques, algorithms and procedures taking motivation from biological evolutionary stages of living organisms mainly human beings. Technically, these are population based complex problem solving techniques for solving various types of optimization problems and work on trial and error mechanism [18]. In operational procedures of these algorithms, a solution search space in initially generated and then updated iteratively. During each iteration, the less desired solutions are removed from the solution search space and randomly generated small changes are incorporated to widen the solution search space [19]. The major algorithms lying under the umbrella of evolutionary computation are GA, GP, evolutionary programming, gene expression programming, evolutionary strategy, differential evolution, neuroevolution and learning classifiers system.

A widely applicable, famous and recently emerged subfield of AI known as swarm intelligence (SI) has drawn the attention of researchers over the last few decades targeting problems of optimization nature. SI gets inspiration from the collective behavior of social swarms of bees, ants, worms, termites, schools of fish and flock of birds in their goals achieving activities. A collective behavior is shown by the swarm of individuals for foraging, reproduction, living and division of important tasks among the available individuals. In fact, a decentralized manner by individuals based on local information collected from the environment is used for decision making [20], [21]. Examples of swarm intelligence methods used for solving optimization problems include PSO (Particle swarm optimization) [22], ant colony optimization (ACO), artificial bee colony optimization (ABC), cuckoo search optimization algorithm [23], bat algorithm [24]; krill herd bio-inspired optimization algorithm, clustering algorithms and firefly algorithm (FA).

In finding solution of optimization problems, the domain of optimization function is called solution search space. In other words, the collection of all considerable or feasible solutions is called solution search space of the optimization problem [25]. Every solution of the problem in the solution search space is called candidate solution. Like other characteristics of optimization problems, the solution search space is dependent on various parameters of the problem and the algorithm being used for solving the problem. Different solutions of an optimization problem present in the solution search space are associated with the values of independent variables and their corresponding values of dependent variables [26].

When searching the solution search space for finding the optimal solution, two phenomena namely local search and global search are used for covering the all possible solutions of the problem. The local search is performed by introducing local or small changes to the current solution to check the local solutions of the problem whereas for performing global search, the searching mechanism is widened to cover more diverse solutions [27]. Different optimization algorithms use different mechanism for performing local and global search process. For example, selection, mutation and crossover operators of GA are used for controlling the local search and global search of GA [28]. In ABC, different mechanisms of employed bees, onlooker bees and scout bees are used for performing local and global search [29]. The local and global search of optimization algorithm can be defined by its exploration and exploitation capability.

All SI approaches are inspired from collective coordinated behavior of various types of swarms e.g. ants, termites, bees, flock of birds and schools of fish. The major role in establishing this strong coordinated and well organized system is played by the individuals of the swarm but for achieving different types of complex and sophisticated goals, these individuals provide their contribution to the decentralized system. In order to accomplish complex tasks, a strong intercommunication system is required among all the participating individuals. For example, making sophisticated nets by worms and termites and search of food by ants and bees in a systematic approach is the result of their well-managed coordinated and collective behavior. All these swarms utilize two phenomena namely exploration and exploitation to make their coordinated system more powerful in achieving their desired goals. Exploration means collecting new information whereas the exploitation means using the existing information for communication among different individuals of the swarms to better manage their coordination. During solving the optimization problems, the exploration is the process of increasing the solution search space to bring variations in the values of optimization function (collecting new information) [30] whereas the exploitation means focusing on the so far found solutions to check all the nearby solutions to enable the search space to not skip the most optimal solution present in the local solution search space (using the existing information) [31].

A trade-off is required between the exploration and exploitation during finding solution of the optimization problems [32]. In swarm intelligence, the swarm of individuals shows a collective behavior based on a self-organized and decentralized coordination system for achieving different goals like reproduction, foraging, food search and other day to day activities. In this decentralized approach, the local information is collected by the individuals from the environment and this locally collected information contributes towards the overall decision making process of the whole swarm. The trade-off between exploration and exploitation is maintained by all optimization algorithms for finding the most optimal solution of the problems using different mechanisms which are explained in their corresponding sections.

The rest of the paper is organized as follows: Section II shows related work; Section III presents the proposed solution; Section IV shows experimental results; Section V presents results discussion while Section VI presents the conclusion and future work.

#### **II. RELATED WORK**

In order to solve optimization problems, different types of algorithms, methods, techniques and procedures have been proposed with all the approaches having their own advantages and disadvantages. Most of the optimization algorithms developed come under the umbrella of subfields of AI known as evolutionary intelligence (EI) and swarm intelligence (SI) [33], [34]. The foremost and most widely applied algorithm of EI approaches is GA which is mainly used for solving optimization problems. The major steps of GA for solving optimization problems include initial random solution generation, selection process, mutation operation and crossover operation. All these operations mimic the biological evolutionary stages of living organisms. The main components of GA are genes and chromosomes. Genes represent the decision variable whereas the chromosomes represent different solutions of the optimization problem. GA is an iterative procedure and in each iteration the value of optimization function is evaluated according to some pre-defined evaluation parameter [35]. GP is considered to a developed form of standard GA but in GP, the solutions of optimization problem are represented by computer programs. The prominent characteristic of GP that makes it discriminant from GA is its representation format. In GA, the solutions are represented linearly whereas in GP, the solutions are given a hierarchical representation. A tree is developed representing different solutions of the problem. The mathematical symbols (e.g.  $+, -, /, \times$ ) represent the internal nodes of the tree whereas the variables of the function represent the external nodes of the tree [36].

Evolutionary programming is another important evolutionary algorithm paradigm with similarity to GP in representation and working procedure. In evolutionary programming, the values of numerical parameters can change whereas the program representing the optimization function remains fixed [37]. The main operator that introduces variations in the solution search space is mutation in evolutionary programming. Every parent in evolutionary programming generates its own offspring because every member of the population is considered as a part of some species instead of member of the same species [38]. Similarly, gene expression programming is an evolutionary algorithm borrowing some characteristics and operators from both GA and GP when solving optimization problems. When manipulating the chromosomes, the gene expression programing is more similar to GA than GP. Like GP, the representation is tree like whereas performing manipulations; crossover operator is taken from the GA [39]. The primary base of the differential evolution is vector differences and hence this technique is majorly more suitable for solving optimization problems of numerical representation. Similar to GA and evolutionary strategies approaches, differential evolution also possesses iterative nature. The differential evolution comes under the category of metaheuristic algorithms which use a larger solution search pace of candidate solutions [40]. The main drawback associated with differential evolution is that it does not guarantee the most optimal solution of the problem being targeted. In differential evolution, a population made up of candidate solutions is maintained and new candidate solutions are created by considering the current solution search space and the best solution in each iteration is maintained [41].

Likewise, a similar technique to genetic programming known as neuroevolution combines some characteristics from GA and ANN. The structure description and weights connections in ANN are represented by genomes which are GA concepts. The major phenomenon of neuroevolution is that the neural network structure, rules, parameters are generated and controlled by an evolutionary algorithm [42]. The next evolutionary intelligence technique is learning classifiers systems (LCS) which is AI approach combining the discovery and learning components [43]. For making various types of predictions, LCS seeks procedures for identification of context dependent rules that use piecewise manner for collecting, storing and applying knowledge. The piecewise manner means breaking down the whole solution search space into simpler smaller parts [44].

In [14], the authors have presented Henry Gas Solubility Optimization (HGSO), a novel physics-based algorithm that mimics the behavior of Henry's law is proposed and termed HGSO. This work is being inspired by the Henry's law of solubility which proves that, "on a constant temperature, the specific amount of a gas which dissolvable in a specific volume and type of liquid is directly proportional to the partial pressure of that gas in equilibrium with that liquid". The experimental results conducted from 47 optimization problems, CEC'17 test suite with three engineering design problems revealed that HGSO targets a balance between exploration and exploitation abilities of the search space which avoid the problem of local optima.

Shimin Li et al [15] proposed a new stochastic optimizer known as slime mould algorithm (SMA) based on the oscillation mode of slime mould in nature. In their research work, they have presented few novel features using a unique mathematical model based on adaptive weights in order to simulate the process of producing positive and negative feedback of slime mould waves during propagation. The stimulation has been controlled with the help of biooscillator for generating the optimal path for connecting food with excellent exploratory ability and exploitation propensity. They conducted an extensive comparative experiments of proposed SMA with several metaheuristics approaches using a set of benchmarks functions to verify its efficiency. Additionally, they used four classical engineering problems in order to validate the efficacy of the algorithm in strict optimizing problems. The final results presented in different table and figures proved that the proposed SMA is a competitive optimization technique on different search landscapes.

In [16], the authors have presents a new optimization algorithm named Equilibrium Optimizer (EO). EO is inspired by the control volume mass balance models, which were used to estimate both dynamic and equilibrium states in physics. In their proposed EO approach, each particle is considered as single solution and the concentration of that particle refers to its position and act as a searching agent. The position are randomly updated with respect to current-best solutions by the search agents and named as equilibrium candidates. When the final best particle from equilibrium candidates reach to the equilibrium state, it is then considered as optimal result. A distinct "generation rate" term is proved to boost the exploration and exploitation ability of EO, for avoidance of local minima. The EO algorithm is tested based on 58 unimodal, multimodal, and composition benchmark functions with three engineering application problems. The compare the results of EO with many widely used optimization approaches including Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), Gravitational Search Algorithm (GSA), Salp Swarm Algorithm (SSA), CMA-ES, SHADE, and LSHADE-SPACMA. They considered average rank of Friedman test, for all 58 mathematical functions which proved the efficiency and significance of EO.

As previously discussed, few important SI algorithms for solving optimization problems include PSO, ACO, ABC, bat algorithm, cuckoo search algorithm and FA. PSO is one of the most common computational methods applied for solving optimization problems. In its basic operations, the candidate solutions are used to create the initial population. In PSO, the candidate solutions are dubbed particles that move around the solution search space with specific velocity using some mathematical operations. The local best position held by particle influences the movement of each particle and all the particles move in their coordinated system to move towards global best solution [22].

An important and commonly used SI approach is ACO in which computational problems are solved by using probabilistic approach. The major constituent of ACO is artificial ant which is a type of multi-agent taking inspiration from the behavior of natural ants [45]. The predominant paradigm used in ACO is the strong coordination and communication among real ants based on the quantity of pheromone. In order to assist each other in exploring the surrounding environment for searching their food, the natural ants direct and guide each other by laying down a material called pheromone. For solving optimization problems, the optimal solutions are located by artificial ants that move around a solution search space that represents the possible solutions of the problem being targeted [46].

In a similar way, ABC is a type of SI algorithms inspired from natural behavior of honey bees for the searching their food source. In ABC, the initial population is generated representing the food sources. There are three types of bees namely employed bees, onlooker bees and scout bees [47]. The assumption is that one employed bee represents one food source which means the total number of food sources is equal to the number of employed bees in the colony. The major responsibility of employed bees is to go to source food and collect nectar. The onlooker bees check the amount of nectar brought by employed bees. The scout bees search for new food sources. For solving optimization problems, the nectar amount is associated with the fitness value of optimization function [48].

Another newly developed SI algorithm named grey wolf optimization (GWO) was introduced by Mirjalili in 2014 [49]. The major source of inspiration leading to the development of this optimization technique was the hunting mechanism of grey wolves. Similar to many other optimization algorithms, GWO works in three stages namely solution search space initialization, updating the solution search space and the termination of the algorithm. After the initialization of the solution search space, the algorithm runs in two stages during the total iterations. The first half of the iterations is used for exploration of the solution search space where as the second half of the iterations is used for exploitation of the solution search space. During exploration, the prey is searched where as during the exploitation stage, the prey is attacked which is correlate with the successful termination of the algorithm by finding the optimal solution of the problem considered [50].

Similarly, whale optimization algorithm (WOA) is another newly proposed optimization model developed by Mirjalili in 2016 by taking inspiration from hunting behavior of whales. The algorithm consists of three stages namely searching the prey, encircling the prey and attacking the prey. For solving optimization problem, the search process is associated with the solution search space of the problem. The solution search space is updated which is associated with the exploration of the solution search space which is represented by the searching phase of the algorithm. Similarly the attacking phenomenon of WOA technique is associated with the exploitation capability of the solution search space [51].

Another SI technique developed by Mirjalili in 2015 was named ant lion optimizer (ALO). This model is inspired from hunting process of antlions in nature. The description of hunting mechanism of ant lions in nature has been outlined in five stages namely random walk of the ants, trap building, entrapping of the ants in the traps, attacking the prey and rebuilding of the traps. When implemented as a technique for solving optimization problems, the whole process is divided into three main stages namely initialization of the solution search space, the updating the solution search space and the termination stage. The newly developed algorithm was tested by using different benchmark optimization functions and some real world problems and was found efficient in solving problems of the optimization [52].

In 2019, another optimization algorithm named harris hawk optimization (HHO) was developed by Mirjalili. The mathematical model developed was inspired from the cooperative and chasing behavior of the harris hawks in nature. This mechanism is called surprise pounce in which many harris hawks pounce a prey from different directions to surprise the prey. The harris hawks have many patterns which are based on changing scenarios and patterns of the prey. The mathematical model developed was tested and validated on various types of benchmark optimization functions and some engineering problems (Heidari *et al.*, 2019).

In a similar attempt, another SI mathematical model was developed in 2019 by Wang and his colleagues. The newly introduced SI technique is inspired from the migration behavior of monarch butterflies population. The migration is usually form America and Canada towards Mexico and some other destinations covering miles of distance. During their migration, there is strong communication and coordination system among all the butterflies that leads to the development of new nature inspired algorithm called monarch butterfly optimization (MBO). In MBO, the residing areas for the butterflies are divided into two zones. One zone consists of North America and Canada (Land 1) and Mexico (Land 2). In order to solve the optimization problems, the positions of the butterflies and their offspring are updated which are associated with the updating the solution search space. The newly developed technique has been tested on different types of benchmark optimization functions and compared with many standards other optimization algorithms [53].

Similarly, an important optimization algorithm of SI area was developed by Wang in 2018 by taking inspiration from reproduction process of earthworms in nature. The algorithm was named earthworm optimization algorithm (EOA). There are two kinds of reproductions of earthworms in nature called reproduction 1 and reproduction 2. The offspring of earthworms are generated by reproduction 1 and reproduction 2 independent of each other. After their independent reproduction, the weighted sums of all the offspring are used from which the next generations are created. Reproduction 1 can generate only one offspring at a time where as reproduction 2 can generate many offspring at a time. This second kind of offspring use crossover operator of standard GA and DE. For avoidance of trapping of solution in local optima, the EOA uses a new type of mutation operator called Cauchy mutation (CM). The newly developed model was tested on several types of benchmark optimization functions [54].

Another swarm based optimization algorithm called elephant herding optimization algorithm was developed by Wang in 2015. The newly developed optimization model was named elephant herding optimization (EHO) algorithm after its inspiration from the herding mechanism of groups of elephants. Naturally, elephants that belong to different clans live together in a group led by matriarch. In their living mechanism, the male elephant will generally leave the group when he is sufficiently grown and can survive independently. These two properties of their livings can be mathematically modeled using two operators called updating and separating operator. In EHO, the elephants can be updated by updating operator followed by separating operator. The developed technique was tested by using some benchmark optimization functions and the results have been compared with few standard well known optimization algorithms [55].

A similar bio inspired algorithm was developed by Wang in 2018 called moth search (MS) algorithm inspired from some properties of moths. A dominant feature of moths is their movement towards or away from source of light. Another characteristic of moth is that they follow levy flights. In MS, the light source is considered as the best moth. Other moths that are close to the best fly around their positions in the form of levy flights. On the other hand, the moths that are far from the best will try to fly toward the best using a big step. These two features of the moth correspond to the exploration and exploitation capabilities of the algorithm when it is applied for solving optimization problems. The performance and efficiency of the newly developed technique have been tested on few standard benchmark functions as well as few real world applications [56].

Cuckoo search is another kind of swarm intelligence technique inspired from the natural behavior of some cuckoos that lay their eggs in the nests of birds of some other species. This behavior of laying eggs in the nests of other birds makes the host birds resist them in few ways. For instance, if an intruder egg is discovered by the host bird in its nest, it is either thrown by the host bird or the host bird will leave the nest and make a new nest. For solving optimization problem, the solutions are represented by eggs in the nest [57].

Similarly, bat algorithm is also a type of SI algorithm mainly used for global optimization. This technique takes inspiration from echolocation behavior of bats. When catching a prey, the bat uses three parameters namely its velocity, its position and loudness. These three properties change continuously when the bats see their prey. The values associated with all these parameters used to calculate the fitness values of optimization function when bat algorithm is used for solving optimization problems [58].

FA is a newly introduced stochastic, nature-inspired meta-heuristic algorithm that has attracted the attention of researchers since its inception for solving various types of optimization problems. FA is inspired from the natural phenomenon of light emission by fireflies for their strong intercoordination to achieve different types of goals. The major source of attraction among different fireflies is the light emitted by fireflies. Higher the light emitted by a firefly, higher will be its force of attraction for other fireflies and vice versa. The light emitted by the fireflies is associated with the objective function to be optimized when solving problems of optimization nature. In order to solve optimization problems, Xin-She Yang associated the intensity of the light emitted by fireflies with the objective function to be optimized to be optimized [59]. In order to have an in depth analysis and understanding of FA, its modified and hybrid versions, few review articles have been written. For example, [60] presented a comprehensive research article consisting of modifications and hybridizations of FA with other optimization algorithms. In the same year, Yang *et al.* published another review article highlighting the recent work done related to FA till that date. In 2015, Fister again published a review article on FA highlighting the challenges and recent advances of FA [61].

Tilahun and Ngnotchouye [62] reviewed the FA and presented a comprehensive study on the development and modifications introduced in the FA. In a similar fashion, [63] reviewed the dynamic parameters adaptation methods of FA. In [64], Tilahun and Ngnotchouye again presented a comprehensive review article highlighting the continuous versions of FA. Similarly, [65] reviewed the applications of FA and its different variants applied in image processing.

Palit et al. [66], proposed a binary firefly algorithm for cryptanalysis of the cipher text from plain text. In their research work, enhancements were made in almost all components of firefly algorithm due to the new representation of the fireflies. The results of the proposed approach were compared with genetic algorithm, which proved the efficiency of the proposed algorithm. [67] improved FA based on Gaussian distribution for moving the fireflies towards the gbest in each iteration to improve the convergence speed. In the conducted research work, the step size or the randomization parameter was set fixed. The proposed algorithm was evaluated on five standard optimization functions. A new firefly algorithm was developed by Yang in [59] based on the Levy flights movement strategy. In order to improve the convergence rate of the conventional firefly algorithm, Dos et al. [68] combined the firefly with the chaotic maps. They used chaotic sequence to avoid the FA to be trapped in local optima. These chaotic maps were used by the proposed algorithm by tuning the light absorption coefficient and the randomization parameter present in the traditional firefly algorithm that are used for position changing by the fireflies. For solving unconstrained optimization problems, in [69] Subutic et al., proposed a parallelized FA. The authors tested the efficiency of the proposed approach with few standard benchmark functions. The experimental results proved the significance of parallelized approach in term of execution time for the parallelized approach which was much less as compared to the standard firefly algorithm.

In conclusion, FA is one of the most recently and well known SI technique applied for solving optimization problems and different types of modifications and hybridization of FA with other algorithms have been introduced which were highlighted in this work.

#### A. PROBLEM CONTEXTUALIZATION

The operational steps of all the iterative optimization algorithms consist of three major stages namely the algorithm initialization, updating the solutions and termination of the algorithm [70]. During the initialization stage, the initial



FIGURE 1. Problem context.

solution search space is defined by randomly generating the solution search space by keeping in consideration the dependent and independent variables of the optimization problem and the dependent values of optimization function. During each iteration of the algorithm, the solution search space generated in the initialization stage is updated to consider more values worth contribution towards finding the most optimal solution [71]. In the termination stage, the algorithm is terminated in either of two conditions. Firstly, the required optimal value of the optimization function is obtained. Secondly, the maximum number of iteration is reached. If the most optimal value of the optimization function is obtained, the approach is said to be successful and no further processing is required. But, if the maximum no of iteration is reached and the algorithm terminates without getting the most optimal value, then the algorithm is suffered from poor exploration [72] or poor exploitation [73] of the solution search space during the initialization or solution search space updating stage. The complete description of this context is shown in Figure 1. The exploitation and exploration of the solution search space can be improved at that different stages but that is a complicated procedure as the internal working mechanism of iterative algorithms is complex process. In the proposed approach, the issue has been resolved by simply embedding the PS at the termination stage of standard FA which takes the solutions from the standard algorithm and improves the exploration and exploitation of the solution search space that further optimizes the values obtained.

#### **B. PROBLEM FORMULATION**

FA is one of the recently developed SI algorithm used for solving various types of optimization problems. Like other optimization problems, the FA consists of three major stages

when applied for solving problems of optimization nature; the initialization stage, the firefly position changing stage and the termination stage. In the initialization stage, the solution search space is randomly generated by keeping in consideration the variables used in the optimization function and the associated value of the optimization function. In the firefly position changing stage, the positions of fireflies are updated by using a factor known as randomization factor for discovering more solutions of the problem been targeted. In the termination stage, the algorithm terminates. In the initialization stage, the randomness in the initial solution space may create imbalanced relationship between exploration and exploitation of the solution search space leading to slower local and global convergence rates of the algorithm resulting in degraded solution quality. The randomization factor in firefly position changing stage decides the movements of the fireflies. If this value is not handled very carefully, it may lead to degraded solution quality of the problem. During the termination stage, the optimized values obtained depend upon the relationship between exploration and exploitation of the solution search space established in the initialization and firefly position changing stage. If the algorithm terminates and the values obtained are the most optimal values, then there is no need of further processing. But, if the algorithm terminates and the values obtained are not the most optimal values, then further processing is required to obtain the most optimal value. In the proposed model, this has been achieved by incorporating PS at the termination stage of standard FA. The problem formulation is shown in Figure 2.

In Figure 2, Maxitr shows the maximum number of iterations for which the algorithm is run. The red color in figure shows the point where the problem arises and the solution is provided at this point.

#### C. LIMITATIONS OF FA

As mentioned in previous sections, the FA is one of the most recently introduced swarm intelligence techniques applied for solving optimization problems. When it is applied for solving optimization problems, it work in three major stages namely the initialization stage, the firefly position changing stage and the termination stage. There are some major drawbacks associated with each of these stages. The first limitation of standard FA is the initial random solution generation at the initialization stage. This randomness in the initialization stage leads to imbalanced relationship between the exploration and exploitation of the solution search space that ultimately degrades the solution quality. Secondly, in each iteration the solution search space is updated using a factor known as the randomization factor. If the value of this randomization factor is taken small, the solution search space will be highly exploitative and poor explorative which will lead the solution to be trapped in local optima. Similarly, if this randomization factor is taken large, the solution search space is highly explorative and poor exploitative resulting in skipping the most optimal solution even present in the vicinities of the current solution. A third major drawback with standard FA



FIGURE 2. Problem formulation.

is its termination without being able to get the most optimal solution of the optimization problem.

#### **III. PROPOSED SOLUTION**

In the proposed model, the performance of standard FA has been improved by embedding PS algorithm at the termination stage of FA if the solution obtained by standard FA is not the most optimal solution. The proposed model consists of three main stages. The initial solution is randomly generated as in the standard FA. The randomization factor in the firefly position changing is used for updating the solution search space and the PS is used at the termination stage that improves the solution quality. The pseudo code of the proposed model is shown in Algorithm 1.

#### A. CONVENTIONAL FA

Xin-She Yang introduced the FA for the first time who was inspired from the light emission capability of fireflies which is the primary source of communication among different fireflies to achieve different goals e.g. reproduction and food search [74]. In order to solve optimization problems, Xin-She Yang associated the intensity of the light emitted by fireflies with the objective function to be optimized. According to physics rules of light, the intensity of the light emitted by firefly, I(r), at distance r from the firefly can be observed by Equation 1:

$$I(r) = \frac{l_o}{r^2} \tag{1}$$

#### Algorithm 1 Proposed Model

#### Stage 1

- Initial random population of fireflies is generated using dependent and independent variables
- Evaluation of fitness of all fireflies
- Find the best firefly

#### Stage 2

- The distance of each firefly from the best firefly is calculated using Equation 4
- The positions of all fireflies are updated using Equation 5
- The values of fitness function are evaluated.
- The fireflies are sorted and ranked based on fitness value.

#### Stage 3

- Check Maximum number of iterations.
- IF (Maximum number of iterations not reached)
- Repeat stage 2
- IF (Maximum number of iterations reached)
- IF (Criteria met)
- Exit
- ELSE
- embed pattern search algorithm
- END

where  $I_0$  represents the light intensity generated at the light source. If  $\gamma$  is the absorption coefficient of the medium, then the light intensity I, at distance r is given by the Equation 2:

$$I = I_o exp(-\gamma r^2) \tag{2}$$

where r represents the distance between source of light and the light observation point. This light intensity can be associated with the attractiveness between the fireflies in the FA which is given by Equation 3:

$$B = B_o exp(-\beta rm)(m \ge I)$$
(3)

where  $\beta_o$  represents the attractiveness at distance r = 0. The distance between two fireflies'  $x_i$  and  $y_j$  called as Euclidean distance is given by Equation 4:

$$r_{ij} = |x_{i} - Y_{j}| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - y_{j,k})^{2}}$$
(4)

In each generation, the position of the firefly can be changed according to the following Equation 5:

$$X_i = x_i + B_o exp(-\gamma r_{ij}^2)(x_i - y_j) + \alpha \epsilon$$
(5)

where  $\alpha$  represents the parameter of randomization,  $\epsilon$  represents random number generated in Gaussian distribution. The parameter of randomization is used to control the solution search space. The working mechanism of conventional FA is shown in Algorithm 2.

#### B. COMPUTATIONAL COMPLEXITY OF STANDARD FA

The less computational complexity of almost all heuristic and metaheuristic techniques make them easily implementable. This phenomenon of these algorithms leads to their applicability in very complex and difficult problems solving procedures. Like other heuristic and metaheuristic algorithms, FA is also considered to be one of the simplest and easiest approaches in terms of computational complexity.

In order to calculate the time complexity of the standard FA, we need three parameters namely the population size, the number of iterations and the calculation of objective function. Suppose, the population size is represented by n, the iteration numbers is represented by *i*. The population is represented by a two dimensional array for which we need two loops to operate. In order to perform the iteration process, we need a single loop. So, the worst case computational complexity is  $\mathcal{O}(n^2 i)$ . In most of the cases, the value of n is very small (like 20) as compared to the value of i (say 500 or greater than 1000), the computational cost is very less due to the linear behavior of *i*. The major factor that increases the computational cost is the computation of the objective function. In all metaheuristic algorithms, the major contribution towards the complexity of these techniques is the later one case namely the calculation of the objective function. In case of FA, if *n* is taken as large, there is possibility of using the one inner loop for calculating the light intensity of the fireflies which is associated then with the objective function by applying a sorting algorithm. In this case, the computational complexity of the algorithm will be  $\mathcal{O}(n \log i)$  (Yang and He, 2013).

#### C. PATTERN SEARCH ALGORITHM (PS)

Pattern search is a kind of optimization algorithm having strong capability of solving various kinds of optimization problems where other standard optimization algorithms face failure in finding the most optimal solution [75]. Its easy implementation, conceptual simplicity and computational efficiency make it more applicable than the other optimization techniques. The basic operations of the pattern search

## Algorithm 2 Firefly Algorithm (FA)

Initialization of FA parameters

- Total number of fireflies in the algorithm: No

- Number of generations: Max
- The values of  $\beta_o, \gamma, \alpha, \epsilon$
- Put the generation counter i=0

Generate initial population of fireflies "Pt" using  $x_1, x_2, x_3 \dots, x_n$  of Equation 6,..., Equation 19 representing light intensity values

Evaluate fitness values "FV" of Equation 6,..., Equation 19 using randomly generated values of  $x_1, x_2, x_3, ..., x_n$ Sort the fireflies according to the values of fitness function "FV"

Find the best firefly from the sorted fireflies

Put i = 0

#### While (i<Max)

- {
- STEP1: Calculate the Euclidean distance  $r_{ij}$  of each firefly from all fireflies using Equation 4
- STEP2: Calculate the attractiveness  $\beta$  of each firefly for other fireflies using Equation 3
- STEP3: Update position Xi of each firefly using Equation 5
- STEP4: Evaluate new "FV" values associated with each firefly
- STEP5: Sort and rank the updated population for values of "FV"
- STEP6: i=i+1
- }

Return the best solution

consist of few technical steps. First of all, the mesh size (MS), the expansion factor (EF), the contraction factor (CF) and the maximum number of iterations (Maxitr) is specified. After specifying these parameters, the starting points of PS are set initially or these points are taken form some other algorithms. In our work, these points have been taken form the standard FA. Using these starting points, the mesh points and pattern vectors are created. The mesh points are used to evaluate the values of objective function to be optimized. In order to check the optimized function values, the mesh points values obtained are checked. If the obtained values obtained are better than the previous values, it means the algorithm is going in right direction, the mesh point's values are expanded and the procedure proceeds. If the values obtained are not better than the previous values, it means the mesh size needs to be contracted and new points are set as starting points. This procedure continues until the termination stages reaches or the maximum number of iteration reaches. The pattern search algorithm has been applied for solving various types of optimization problems. The working mechanism of PS is shown in Algorithm 3.

#### D. PATTERN SEARCH BASED FIREFLY MODEL (FA-PS)

In the basic operation of standard FA, the algorithm terminates in two cases; the required optimal value is obtained or the maximum number of iteration reaches. If the algorithm terminates after getting the required optimal value, then the approach is said to be successful in achieving the target goal and there is no further processing required. If the algorithm terminates after the maximum number of iteration reaches and the value obtained is not the most optimal value, then the number of iterations can be increased. Even after increasing the number of iterations, the value obtained is not the most optimal value, and then some other technique can be applied to get the most optimal value or at least the value better than the value obtained by standard FA in its maximum iterations. If the maximum number of generations of firefly algorithm reaches and the solution obtained is not the most optimal solution, then the pattern search algorithm is introduced to further enhance the exploration and exploitation of the solution search space to obtain the most optimal solution or at least the solution better than the so far obtained solution. Pattern search is a kind of optimization algorithm having strong capability of solving various kinds of optimization problems where other standard optimization algorithms face failure in finding the most optimal solution. Its easy implementation, conceptual simplicity and computational efficiency make it more applicable than the other optimization techniques. In the proposed approach, pattern search (PS) has been applied to target this issue and to improve the results obtained by standard FA. The PS takes the value obtained by standard FA as starting point and performs further processing to get the minimum or maximum value of the minimization or maximization functions, respectively. The working mechanism of the PS based FA is shown in Algorithm 4. The detailed data flow diagram of the proposed model is shown in Figure 3.

## E. ADVANTAGES AND LIMITATIONS OF THE PROPOSED MODEL

The newly developed model is a hybrid model of standard FA and PS which is advantageous over the few other standard SI techniques apart from the standard FA. The benefits of the proposed technique are three fold.

**IEEE**Access

#### Algorithm 3 Pattern Search (PS)

Initialization of parameters - MS, EF, CF, Maxitr Set initial mesh points "MP" using  $x_1, x_2, x_3 \dots, x_n$  of Equation 6,..., Equation 19 Evaluate fitness values "FV" of Equation 6,..., Equation 19 using values of  $x_1, x_2, x_3, \ldots, x_n$  from FA Put i = 0While (i<Maxitr) -{  $IF(FV_{i+1} < F_i)$  // for minimization functions Expand "MS" ELSE Contract "MS" Set new "MP" using  $x_1, x_2, x_3, \ldots, x_n$  of Equation 6,..., Equation 19 END IF  $IF(FV_{i+1} > F_i) //$  for maximization functions Expand "MS" ELSE Contract "MS" Set new "MP" using  $x_1, x_2, x_3, \ldots, x_n$  of Equation 6,7, 8 and 19 END IF } Return the result

Firstly, its easy implementation in terms of making a simple hybridization of two conventional models leading to an efficient mathematical model makes it distinct from other sophisticated techniques which involve complicated mathematical operations. The easiness behind the scenario is its straight forward operational steps. The output values obtained from the standard FA are given as inputs to the PS which further performs its processing and tries to optimize these values to improve the power of the standard FA in terms of getting the most optimal value of the optimization problem being kept under consideration. The starting points of the standard PS are the values obtained from the FA which are further processed in order to enhance the strength and power of the optimization model. No complicated operations are involved in this whole procedure.

Secondly, the result oriented technique in terms of getting the most optimal solution or at least better solution than the solution obtained by the standard FA makes the new model an efficient one. The performance of the proposed model is pretty prominent as it has been outlined in the experimental setup section of the document. Since, the PS technique used in the model further performs processing on the values obtained from the standard FA, the improvement in the efficiency of the model is logically, mathematically and experimentally quite eminent. The provides distinction to the proposed model as compared to other modified models or hybrid models which are suffered from ambiguity when physically implemented.

Lastly, almost all optimization models are suffered from the drawback of the failure in getting the most optimal solution in their total number of iterations. The proposed model provides a strong and standard procedure for solving this issue which can be implemented in other optimization models as well by keeping in consideration the operational stages performed in this work.

Although, the proposed model has been quite successful in achieving a better quality solution as compared to the standard FA but there is a serious disadvantage associated with this new model. When the values obtained from maximum number of iterations of standard FA are given as inputs to the PS for further processing, it increases the complexity of the algorithm. So, it will need proper attention in near future to address this issue of the proposed model and propose a valid, easy and acceptable solution.

The major aim of the proposed FA-PS model is to achieve improvement in the optimization capability of conventional FA in obtaining the optimized values but not the complexity of the algorithm. The proposed model has been developed by adding functionalities of PS algorithm to the standard FA. These two optimization algorithms have different types of working mechanism for handling optimization functions. So, embedding the concepts of one algorithm in another algorithm involves few technical procedures. For example, developing hybrid model of FA and PS involves the mapping of fireflies of FA into the PS parameters and converting the PS attributes back to fireflies. During this process in the proposed approach, no operational step has been removed from the conventional FA and PS algorithms, rather new operators have been added. This addition of new operators increases the complexity of the hybrid models. Secondly, the validity of the proposed FA-PS model has been tested on eight minimization functions and six maximization functions. In order to make more generalization of the experimental results, it is necessary to further perform the experimentation on various types

## Algorithm 4 FA-PS

- Initialization of FA parameters
  - Total number of fireflies in the algorithm: No
  - Number of generations: Max
  - The values of  $\beta_o, \gamma, \alpha, \epsilon$
  - Put the generation counter i=0
  - Mesh size: MS
  - Expansion Factor: EF
  - Contraction Factor CF
  - Maximum number of iteration in PS: Maxitr

Generate random initial solutions of fireflies "Pt" using  $x_1, x_2, x_3 \dots, x_n$  of Equation 6,..., Equation 19 representing light intensity values

Evaluate fitness values "FV" of Equation 6,..., Equation 19 using randomly generated values of X1, X2, X3, ..., XnSort the fireflies according to the values of fitness function "FV"

Find the best firefly from the sorted fireflies

Put i = 0

While (i<Max)

- {
- STEP1: Calculate the Euclidean distance r<sub>ij</sub> of each firefly from all fireflies using Equation 4
- STEP2: Calculate the attractiveness  $\beta$  of each firefly for other fireflies using Equation 3
- STEP3: Update position Xi of each firefly using Equation 5
- STEP4: Evaluate new "FV" values associated with each firefly
- STEP5: Sort and rank the updated population for values of "FV"
- STEP6: i=i+1
- }

Assign the parameter of FA to PS

Set initial mesh points "MP" using  $x_1, x_2, x_3 \dots, x_n$  of Equation 6,..., Equation 19

Evaluate fitness values "FV" of Equation 6,..., Equation 19 using values of  $x_1, x_2, x_3 ..., x_n$  from FA Put i = 0

## While (i<Maxitr)

{ \_  $IF(FV_{i+1} < F_i) //$  for minimization functions Expand "MS" ELSE Contract "MS" Set new "MP" using  $x_1, x_2, x_3, \ldots, x_n$  of Equation 6,..., Equation 19 END IF  $IF(FV_{i+1} > F_i) //$  for maximization functions Expand "MS" ELSE Contract "MS" Set new "MP" using  $x_1, x_2, x_3, \ldots, x_n$  of Equation 6,7, 8 and 19 END IF } Return the result

of other uni-modal functions and multi-modal functions and also other multi-objective real world optimization problems with various types of affecting parameters.

Further, as explained in previous sections, two other drawbacks apart from the one targeted in this work are also associated with the standard FA. In order to have complete and concrete representations of the FA working mechanism and its proper applications for solving various types of complicated optimization problems, these issues must be resolved. The proposed model is one problem oriented approach focusing on only one major problem of the FA which is the major drawback of the proposed model.

#### **IV. EXPERIMENTAL RESULTS**

#### A. EXPERIMENTAL SETUP

This section describes the hardware and software resources used in our research activity. The experiments in for this paper were conducted on Intel(R) core(TM) i5 CPU with 2.7 GHz





FIGURE 3. The proposed FA-PS model.

processor and 8GB of random access memory. For quick implementation, the proposed model and other comparative approaches were coded using MATLAB R2017a.

#### **B. PARAMETERS TUNING**

The parameters of all the considered optimization algorithms were tuned with different values for checking the output results related with different values. The technique adopted for tuning the parameters is trial and error mechanism as followed by other researchers working in this area because there are no fixed hard and fast rules for tuning the parameters of the optimization technique. The values and ranges for tuning parameters of all the optimization algorithms presented in this work have been taken from the standard models specified by their authors. The maximum iterations of all the algorithms namely FA, GA, ABC, ACO and the proposed were varied from 100 to 400 with an increment of 20. The final number

value all the algorithms gave excellent results. The initial population size for FA and GA was in the range of 40 to 100 with final population size considered as 80. The colony size of ABC was taken as in the range of 100 to 400 with increment of 20 whereas the number of ants in ACO was varied from 50 to 150 with increment of 10. The gamma, alpha and beta values for FA was varied in the range of 0.5 to 2 with increment of 0.5, 1 to 4 with increment of 0.5 and 0.1 to 0.5, respectively. In case of GA, one point crossover with crossover probability of 0.5 to 0.9 and mutation rate of 0.1 to 0.3 were considered. For ABC, the employed bees and onlooker bees were used in the range of 50 to 200, respectively. In case of ACO, the randomness factor, evaporation rate, ppheromone control parameter and heuristic information parameter were taken in the range of 0.6 to 0.9, 0.1 to 0.5, 1 to 5 and 1 to 5.

of iterations for all the algorithms was set to 300 as for this

#### C. OPTIMIZATION FUNCTIONS

The performance of the proposed approach has been tested on total of fourteen functions including eight minimization functions and six maximization functions outlined in Equation 6 to Equation 19.

$$F1 = e^{-(x_1-4)^2 - (x_2-4)^2} + e^{-(x_1-4)^2 - (x_2-4)^2} + 2e^{(x_1)^2 - (x_2+4)^2} + 2e^{(x_1)^2 - (x_2+4)^2}$$
(6)

where  $x_i \in \{-5, 5\}$ 

$$F2 = (|x_1| - 5) + (|x_2| + 5)^2$$
(7)

where  $x_i \in \{-10, 10\}$ 

$$F3 = 100 * (x_2 - x_1)^2 + (6.4 * (x_2 - 0.5)^2 - x_1 - 0.6)^2$$
(8)

where  $x_i \in \{-5, 5\}$ 

$$F4 = 100 * (x_2 - x_1)^2 + (x_1 - 1)^2$$
(9)

where  $x_i \in \{-2.048, 2.048\}$ 

$$F5 = (x_2 - \frac{5 \cdot 1}{4\pi^2} (x_1)^2 + \frac{5}{\pi} (x_1) - 6)^2 + 10(1 - \frac{1}{8\pi} \cos_{x_1} + 10) \quad (10)$$

where  $x_i \in \{-5, 5\}$ 

$$F6 = (x_1)^2 + (x_2)^2 - \cos(18(x_2)) \tag{11}$$

where  $x_i \in \{-1, 1\}$ 

$$F7 = (x_1 + 2x_2 - 7)^2 + 2((x_1)(x_2) - 5)^2$$
(12)

where  $x_i \in \{-10, 10\}$ 

$$F8 = 4(x_1)^2 + 2.1(x_1)^4 + \frac{1}{3}(x_1)^6 + (x_1)(x_2) - 4(x_1)^2 + 4(x_1)^4$$
(13)

where  $x_i \in \{-5, 5\}$ 

$$F9 = -(20 + (x_1^2 - 10 * \cos(2\pi x_1))) + x_2^2 - 10 * \cos(2\pi x_2))$$
(14)

where  $x_i \in \{-2.048, 2.048\}$ 

$$F10 = \cos x_1 * \cos x_2 e^{-(x_1 - \pi)^2 - (x_2 - \pi)^2}$$
(15)

where  $x_i \in \{-20, 20\}$ 

$$F11 = 280 - \frac{(x_1^4 - 16x_1^2 - 15x_1)}{2} - \frac{(x_2^4 - 16x_2^2 - 15x_2)}{2}$$
(16)

where  $x_i \in \{-5, 5\}$ 

$$F12 = ln\{[(sin(cosx_1 + cosx_2)^2)^2 - (cos(sinx_1 + sinx_2)^2)^2 + x_1]^2\} - 0.1((x_1 - 1)^2 + (x_2 - 1)^2)$$
(17)









where  $x_i \in \{-10, 10\}$ 

$$F13 = \left[\sum_{i=1}^{5} i\cos(i+1)x_1 + i\right] \left[\sum_{i=1}^{5} i\cos(i+1)x_{2i}\right]$$
(18)

where  $x_i \in \{-10, 10\}$ 

$$F14 = 660 - ((x_1)^2 + x_2 - 11)^2 - (x_1 + (x_2)^2 - 7)^2 \quad (19)$$

where  $x_i \in \{-6, 6\}$ 

### D. CONVERGENCE RATE

The convergence rates of fourteen optimization functions are shown graphically in this section. Figure 4 to Figure 11 show the convergence of the proposed model in comparison with the ABC, ACO, GA and FA in minimizing the values of the functions. The convergence of six maximization functions by the proposed model and the other optimization standard approaches is shown in Figure 12 to Figure 17.

Figure 4 to Figure 11 shows the convergence rates of all the considered minimization functions for FA, GA, ABC, ACO and the proposed FA-PS model. All the algorithms were run for 300 iterations. The figures clearly show that the proposed FA-PS go beyond the optimization algorithms namely FA, GA, ABC and ACO in terms of getting the minimum value for all the minimization functions given in the figure. In case of



FIGURE 6. F3 convergence.



FIGURE 7. F4 convergence.



FIGURE 8. F5 convergence.









FIGURE 11. F8 convergence.

the proposed FA-PS model, the FA was run for 200 iterations and then the PS was embedded which was run for 100 iterations. For different minimization functions, the convergence of the all the considered algorithms are different and no improvement can be observed after specific iteration which leads to the degraded solution quality resulting in failure the algorithms to get the most optimal solution (minimum value in case of minimization functions). The solution quality of the standard FA is worse than different algorithms in different minimization functions. After the introduction of PS at the

termination stage, the solution quality gets better and the optimal values obtained by the developed FA-PS then outperform all the other considered algorithms. Figure 12 to Figure 17 show the convergence rate of six maximization functions for obtaining the maximum values for the optimization functions.

Figure 12 to Figure 17 shows the comparison of the developed FA-PS model with other optimization algorithms for maximization functions in terms of convergence rate in getting the maximum value. The figure clearly shows that the



FIGURE 12. F9 convergence.



FIGURE 13. F10 convergence.



FIGURE 14. F11 convergence.

proposed FA-PS is more proficient than all the other considered optimization techniques in terms of getting the maximum value for all the maximization functions presented in the figure. In FA-PS model, the FA was run for 200 iterations and then the PS was introduced which was run for 100 iterations. Since there is no improvement in the solution quality of the standard FA, PS was embedded to further improve the exploration and exploitation of the solution search space the enhances the solution quality resulting in better convergence



FIGURE 15. F12 convergence.





FIGURE 17. F14 convergence.

of the optimization algorithm towards getting the maximum value. The introduction of PS after a fixed number of iteration improves the convergence rate of the FA that results in better solution quality as compared to other standard algorithms.

#### E. COMPARATIVE ANALYSIS

In this section, the performance of the proposed model has been compared with other standard optimization algorithms including GA, FA, ABC and ACO. The performance

#### TABLE 1. Comparison of FA-PS model with other techniques for minimization functions.

Function	Technique	Worst Case Solution	Average Case Solution	Best Case Solution	Standard Deviation
	FA	6.67E-10	7.55E-13	6.00E-16	5.43E-09
	GA	8.65E-10	6.35E-12	3.54E-16	8.10E-08
F1	ABC	5.78E-08	4.66E-10	3.65E-14	3.66E-08
	ACO	8.90E-11	5.77E-14	7.55E-16	6.79E-09
	DE	7.53E-10	6.77E-12	9.34E-15	3.22E-07
	BA	4.34E-09	7.23E-13	2.36E-16	5.21E-08
	GWO	6.34E-07	6.45E-09	4.76E-13	4.33E-06
	SASFA	1.04E-12	1.02E-12	1.017E-12	5.72E-15
	FA-CROSS	8.56E-12	4.78E-13	6.45E-18	1.91E-12
	FA-PS	3.77E-12	6.57E-15	9.55E-18	6.90E-10
	FA	1.57E+01	1.46E+01	1.20E+01	3.17E-01
	GA	1.39E+01	1.25E+01	1.16E+01	2.55E-01
	ABC	1.57E+01	1.39E+01	1.18E+01	4.66E-01
	ACO	1.47E+01	1.27E+01	1.16E+01	5.77E-01
<b>F</b> 2	DE	1.73E+01	1.54E+01	1.24E+01	6.32E-01
F2	BA	1.29E+01	1.21E+01	1.07E+01	4.32E-01
	GWO	1.54E+01	1.43E+01	1.23E+01	5.34E-01
	SASFA	2.50E+01	2.50E+01	2.50E+01	3.35E-03
	FA-CROSS	1.17E+01	1.02E+01	9.02E+00	6.60E-01
	FA-PS	1.28E+01	1.06E+01	8.11E+00	3.66E-01
	FA	8.44E-02	6.45E-02	3.67E-02	2.79E-02
	GA	7.34E-01	4.34E-01	1.45E-02	6.34E-02
	ABC	9.43E-03	7.45E-03	6.65E-03	4.45E-02
	ACO	8.33E-02	6.89E-02	3.25E-02	5.45E-03
	DE	6.45E-03	4.56E-03	2.12E-03	6.54E-02
F3	BA	9.45E-02	7.34E-02	3.26E-02	5.23E-03
	GWO	6.34E-03	4.21E-03	1.65E-03	2.54E-03
	SASFA	5.47E-03	1.07E-03	1.64E-14	1.15E-03
	FA-CROSS	6.90E-03	5.91E-04	6.57E-05	1.52E-03
	FA-PS	9.34E-04	7.55E-04	3.56E-04	7.45E-03
	FA	9.34E-03	6.97E-03	2.54E-03	3.67E-02
	GA	8.55E-03	6.45E-03	1.34E-03	4.67E-02
	ABC	7.25E-03	8.55E-04	6.78E-04	5.33E-03
	ACO	8.55E-02	7.55E-03	6.54E-03	3.55E-02
-	DE	6.87E-02	3.23E-02	8.45E-03	6.45E-03
F4	BA	9.34E-03	5.76E-03	1.34E-03	8.67E-02
	GWO	5.45E-03	1.43E-03	7.65E-04	4.65E-03
	SASFA	4.74E-04	2.18E-05	4.99E-08	5.39E-05
	FA-CROSS	8.56E-02	6.31E-03	6.56E-05	1.95E-02
	FA-PS	2.67E-04	5.66E-05	1.38E-06	4.44E-05
	FA	7.44E-02	4.67E-02	1.87E-02	7.23E-02
	GA	9.15E-03	5.33E-03	1.75E-03	4.88E-02
	ABC	9.85E-03	7.45E-03	5.45E-03	3.85E-02
	ACO	8.45E-01	6.55E-01	1.77E-01	7.45E-02
<b>F</b> <sup>2</sup>	DE	7.34E-02	3.56E-02	1.52E-02	6.34E-02
F5	BA	5.32E-03	2.43E-03	8.34E-04	4.76E-02
	GWO	7.34E-02	4.65E-02	1.87E-02	7.76E-02
	<u>.</u>	1	1	·	Continued on next page

Continuation of Table 1								
Function	Technique	Worst Case Solution	Average Case Solution	Best Case Solution	Standard Deviation			
	SASFA	3.98E-01	3.97E-01	3.978E-01	3.73E-05			
	FA-CROSS	3.24E+00	8.79E-01	2.68E-04	1.07E+00			
	FA-PS	9.34E-03	5.55E-03	1.98E-03	2.45E-02			
	FA	-1.49E+00	-1.62E+00	-1.80E+00	4.93E-01			
E(	GA	-1.52E+00	-1.70E+00	-1.88E+00	6.33E-01			
F6	ABC	-1.53E+00	-1.69E+00	-1.83E+00	4.34E-01			
	ACO	-1.49E+00	-1.54E+00	-1.73E+00	6.45E-01			
	DE	-1.45E+00	-1.49E+00	-1.53E+00	6.43E-01			
	BA	-1.54E+00	-1.59E+00	-1.62E+00	5.34E-01			
	GWO	-1.47E+00	-1.52E+00	-1.59E+00	4.98E-01			
	SASFA	-1.85E+00	-1.99E+00	-2.00E+00	1.83E-02			
	FA-CROSS	2.29E+00	-2.38E+00	-2.88E+00	1.10E+00			
	FA-PS	-1.60E+00	-1.87E+00	-1.99E+00	4.23E-01			
	FA	9.79E+00	7.44E+00	6.90E+00	6.45E-01			
	GA	7.34E+00	5.45E+00	3.67E+00	4.34E-01			
	ABC	7.45E+01	3.79E+01	1.06E+01	5.55E-01			
	ACO	8.34E+00	6.46E+00	4.68E+00	7.55E-01			
E7	DE	9.43E+00	7.45E+00	4.87E+00	5.65E-01			
1.1	BA	7.56E+00	6.45E+00	5.45E+00	6.45E-01			
	GWO	8.34E+00	7.34E+00	5.76E+00	4.65E-01			
	SASFA	8.96E-04	9.25E-05	5.54E-14	1.22E-04			
	FA-CROSS	7.57E+00	4.91E+00	1.66E+00	1.55E+00			
	FA-PS	9.34E-02	7.34E-02	5.75E-02	5.34E-02			
	FA	-1.17E+00	-1.22E+00	-1.37E+00	4.33E-02			
	GA	-1.16E+00	-1.25E+00	-1.43E+00	5.67E-02			
	ABC	-7.87E-01	-9.67E-01	-1.29E+00	3.66E-01			
	ACO	-9.55E-01	-1.16E+00	-1.40E+00	6.66E-01			
E0	DE	-1.13E+00	-1.19E+00	-1.25E+00	5.45E-01			
1.0	BA	-1.16E+00	-1.19E+00	-1.21E+00	4.35E-01			
	GWO	-1.21E+00	-1.24E+00	-1.26E+00	3.56E-01			
	SASFA	-1.031E+00	-1.03E+00	-1.031E+00	1.26E-05			
	FA-CROSS	-1.44E+00	-1.57E+00	-1.62E+00	4.52E-02			
	FA-PS	-1.33E+00	-1.49E+00	-1.53E+00	4.67E-02			

#### TABLE 1. (Continued.) Comparison of FA-PS model with other techniques for minimization functions.

evaluation parameters include best case solution, average case solution, worst case solution and standard deviation of solutions for twenty runs of all the considered algorithms. The comparison has been carried out for both the minimization functions and maximization functions considered for evaluation. As stated earlier, total of eight minimization functions and six maximization functions have been used in this work for comparison.

#### 1) COMPARISON FOR MINIMIZATION FUNCTIONS

Table 1 shows the comparison of FA-PS model with standard FA, GA, ABC and ACO in terms of the performance evaluation parameters considered in the work. A Total of eight functions have been considered for all the algorithms. In the conduction experiments for the first three minimization functions, all the comparative models are smoothly converge until half iterations. The fast convergence of the proposed FA-PS can be observed after 200 iterations. In the few last complex minimization functions, The proposed model has fast convergence from 100 to straight 300 iterations. The Table 1 shows that the proposed FA-PS model outperforms all the other approaches showing the efficiency of the proposed technique.

#### 2) SIGNIFICANCE OF RESULTS FOR MINIMIZATION FUNCTIONS

In order to show that the results obtained for the proposed model, t-test is conducted to reveal that there is significant

Model	Function	FA			GA			ABC		
		t	df	Sig.	t	df	Sig.	t	df	Sig.
	F9	14.3	41.4	0.03	12.3	37.3	0.03	16.1	51.2	0.03
	F10	12.3	27.4	0.03	14.5	35.7	0.03	16.5	46.3	0.04
	F11	11.3	32.6	0.03	13.6	36.5	0.04	12.6	37.2	0.03
	F12	13.5	36.6	0.04	12.7	34.6	0.03	12.7	36.9	0.04
	F13	12.7	39.7	0.03	11.6	36.6	0.03	13.6	40.6	0.03
	F14	11.7	32.2	0.03	12.3	31.4	0.04	12.2	37.5	0.03
		ACO			DE			BA		
		t	df	Sig.	t	df	Sig.	t	df	Sig.
	F9	12.4	45.4	0.03	12.3	31.4	0.04	13.6	36.5	0.04
	F10	15.2	36.3	0.04	12.3	37.3	0.03	12.7	34.6	0.03
FA-PS	F11	10.5	31.5	0.04	13.6	40.6	0.03	12.3	27.4	0.03
14-15	F12	11.5	35.7	0.03	12.7	34.6	0.04	14.3	41.4	0.03
	F13	12.4	38.7	0.03	12.3	37.3	0.03	12.3	27.4	0.03
	F14	12.4	34.5	0.04	13.6	40.6	0.03	10.5	31.5	0.04
		GWO			SASFA			FA-CROSS		
		t	df	Sig.	t	df	Sig.	t	df	Sig.
	F9	11.5	35.7	0.03	12.3	37.3	0.03	14.3	41.4	0.03
	F10	13.6	40.6	0.03	14.5	35.7	0.03	12.3	27.4	0.03
	F11	10.5	31.5	0.04	12.3	37.3	0.03	14.3	41.4	0.03
	F12	14.5	33.7	0.04	13.5	36.6	0.04	12.7	34.6	0.04
	F13	15.5	37.7	0.04	16.5	46.3	0.04	11.3	32.6	0.03
	F14	14.3	41.4	0.03	12.7	36.9	0.04	13.5	36.6	0.04

TABLE 2. T-test values of maximization function for the proposed model against standard techniques.

contrast between the values of FA-PS and the standard optimization algorithms. The null hypothesis,  $H_o$  in this case is "there is no significant difference between the values of proposed model and the other optimization techniques with which the developed model has been compared". The t-test results are shown in Table 4. This table shows that if p <= 0.05 (95% confidence level), then the proposed model perform better than the other considered approaches. In the case of all the functions, FA-PS have better results; therefor the null hypothesis is rejected.

#### 3) COMPARISON FOR MAXIMIZATION FUNCTIONS

Table 3 shows the comparison of the proposed FA-PS model with other well-known optimization algorithms namely standard FA, GA, ABC and ACO. The parameters for the comparison include worst case solution, best case solution, average case solution and standard deviation. Similarly like minimization functions, the proposed shows better convergence for maximization as well. The reason of smooth convergence of the proposed FA-PS model is the properties of both FA and PS together in one model. The Table 3 reveals that the FA-PS model outperforms other standard optimization techniques. As shown in Table 3, a total of six maximization functions have been considered for the evaluation of the proposed model.

#### 4) SIGNIFICANCE OF RESULTS FOR MAXIMIZATION FUNCTIONS

This section shows the significance of the results obtained by conducting t-test for maximization functions. The test has been conducting for showing that a significant contrast exists between the results of standard optimization techniques and the proposed model. The null hypothesis,  $H_o$  in this case is "there is no significant difference between the values of developed FA-PS and the other optimization techniques with which the proposed model has been compared". The t-test results for maximization functions, shown in Table 2 reveals that if p <= 0.05 (95% confidence level), then the FA-PS

#### TABLE 3. Comparison of FA-PS model with other techniques for maximization functions.

Function	Technique	Worst Case Solution	Average Case Solution	Best Case Solution	Standard Deviation
	FA	-9.56E-03	-7.44E-03	-4.34E-03	6.33E-03
	GA	-3.67E-03	-8.67E-04	-4.35E-04	4.78E-03
	ABC	-8.45E-03	-7.44E-03	-8.77E-04	3.67E-03
F9	ACO	-8.66E+00	-8.67E-01	-2.27E-01	3.55E-01
	DE	-4.65E-03	-2.34E-03	-7.45E+00	2.67E-02
F9	BA	-7.45E-02	-5.34E-02	-1.43E-02	4.54E-02
	GWO	-4.34E-01	-1.34E-01	-7.56E+00	4.34E-02
	SASFA	3.01E+00	3.01E+00	3.00E+00	2.76E-03
	FA-CROSS	-3.57E-08	-1.42E-07	-6.57E-07	1.81E-07
	FA-PS	-7.55E-03	-3.66E-04	-6.69E-05	4.55E-04
	FA	9.45E-12	3.34E-12	8.66E-13	3.55E-06
	GA	4.45E-11	8.45E-12	3.38E-12	6.44E-05
	ABC	3.55E-11	7.45E-12	4.34E-12	4.65E-05
	ACO	9.45E-15	3.46E-15	9.45E-14	3.55E-05
510	DE	4.34E-14	9.54E-14	4.34E-13	6.21E-06
FIO	BA	3.34E-15	8.58E-14	1.22E-13	7.23E-05
	GWO	4.78E-16	9.11E-16	5.34E-15	4.99E-05
	SASFA	9.71E-03	7.43E-03	9.98E-08	4.036E-03
	FA-CROSS	9.57E-06	3.73E-06	3.57E-07	3.66E-06
	FA-PS	1.55E-12	5.55E-12	7.66E-11	5.44E-07
	FA	5.80E+02	5.86E+02	5.97E+02	4.67E+01
	GA	5.82E+02	5.96E+02	6.09E+02	3.66E+01
	ABC	5.62E+02	5.73E+02	5.87E+02	2.55E+01
	ACO	5.57E+02	5.65E+02	5.79E+02	8.55E+00
	DE	4.23E+02	4.29E+02	5.34E+02	7.34E+00
F11	BA	5.34E+02	5.46E+02	5.51E+02	6.34E+00
	GWO	4.87E+02	5.13E+02	5.27E+02	6.99E+00
	SASFA	-2.84E-04	-2.84E-04	-2.84E-04	2.17E-19
	FA-CROSS	7.65E+02	7.77E+02	7.87E+02	5.87E+00
	FA-PS	5.90E+02	6.15E+02	6.24E+02	3.55E+00
	FA	1.26E+01	1.55E+01	1.84E+01	1.34E+00
	GA	1.66E+01	1.86E+01	2.27E+01	1.65E+00
	ABC	1.36E+01	1.65E+01	1.98E+01	1.25E+00
	ACO	9.55E+00	1.26E+01	1.68E+01	1.44E+00
<b>F10</b>	DE	6.97E+00	7.83E+00	9.87E+00	1.64E+00
F12	BA	4.65E+00	5.34E+00	8.41E+00	1.53E+00
	GWO	7.43E+00	8.12E+00	9.78E+00	1.79E+00
	SASFA	7.68E-05	7.68E-01	1.00E+00	4.10E-01
	FA-CROSS	1.84E+01	2.00E+01	2.27E+01	1.33E+00
	FA-PS	1.95E+01	2.16E+01	2.57E+01	1.22E+00
	FA	8.85E+01	9.65E+01	104.8798	7.85E+00
	GA	9.37E+01	1.02E+02	110.6556	5.96E+00
	ABC	9.55E+01	1.03E+02	107.767	7.55E+00
	ACO	8.47E+01	9.16E+01	98.665	6.55E+00
E12	DE	7.54E+01	8.54E+01	9.56E+01	4.45E+00
F15	BA	6.43E+01	7.34E+01	9.34E+01	5.43E+00
	GWO	7.23E+01	8.11E+01	9.89E+01	6.34E+00
	SASFA	3.58E+02	3.58E+02	3.58E+02	4.19E-04
	FA-CROSS	1.24E+02	1.28E+02	1.31E+02	2.19E+00
	FA-PS	1.10E+02	1.17E+02	121.887	5.55E+00
	FA	8.25E+02	8.28E+02	830.945	3.27E+00
	GA	8.24E+02	8.26E+02	828.5645	5.55E+00
				(	Continued on next page

Continuation of Table 3									
Function	Technique	Worst Case Solution	Average Case Solution	<b>Best Case Solution</b>	Standard Deviation				
	ABC	8.26E+02	8.29E+02	832.76	4.23E+00				
	ACO	8.27E+02	8.30E+02	834.3432	3.56E+00				
	DE	8.21E+02	8.26E+02	8.31E+02	4.33E+00				
	BA	8.24E+02	8.27E+02	8.29E+02	5.11E+00				
	GWO	8.16E+02	8.21E+02	8.27E+02	6.12E+00				
	SASFA	1.77E+00	2.19E+00	2.20E+00	6.15E-02				
	FA-CROSS	8.27E+02	8.30E+02	8.33E+02	1.90E+00				
	FA-PS	8.31E+02	8.34E+02	837.2243	2.17E+00				

TABLE 3.	(Continued.)	Comparison o	f FA-PS model	with other	techniques	for maximization	functions.
----------	--------------	--------------	---------------	------------	------------	------------------	------------

perform better than the other considered approaches. In the case of all the functions, the proposed model have better results; therefor the null hypothesis is rejected.

#### **V. RESULTS DISCUSSION**

The results obtained from the extensive experimentation are analyzed based on the convergence rates and the numerical comparison outlined in previous sections. Total of 14 functions have been considered out of which 8 are minimization functions and 6 are standard maximization functions. For all minimization and maximization functions, there are noticeable fluctuations in the convergence rates in the initial iterations of all the algorithms. These fluctuations can be seen for initial 150 iterations in case of minimization functions F1, F2 and F3 whereas these are visible in initial 100 iterations in case of F5 and F8, these changes can be observed after nearly 140<sup>th</sup> iterations.

As it can be observed in all the minimization functions, there is no observable change in the convergence rate of these functions after that specified iterations for all the optimization algorithms considered in our experimentation except the standard FA in which there are still some fluctuations after these iterations but these fluctuations also take a smooth convergence rate after few iterations. If the convergence rates of the techniques are keenly observed, it is evident that the values obtained by standard FA are adequately worse than few other techniques for different minimization functions. The major reason behind this low performance of all the standard algorithms is their poor relationship between the exploration and exploitation capabilities of the solution search space. Similar to other standard optimization techniques, FA is also suffered from imbalanced relationship between the exploration and exploitation capability of the solution search space that leads to degraded solution quality ultimately resulting in not getting the most optimal solution. In order to overcome this drawback, the PS is introduced at the position when there is no further improvement in the convergence rate of the standard FA.

After integrating the PS at the termination of the FA, the results of the convergence are sufficiently improved. This improvement is the result of enhancing the strength of the FA

in getting the most optimal solution or atleast better solution than the solution obtained by standard FA or the other wellknown optimization algorithms e.g. GA, ABC, ACO, bat algorithm, DE, GWO, SASFA and FA-CROSS techniques. As revealed by the convergence rates of the minimization functions, the proposed FA-PS model is adequately better than the all standard optimization algorithms considered in the experimentation due to the introduction of PS after the execution of standard FA when there is no improvement observed. The integration of PS at the termination stage introduces improvement in the exploration and exploitation capabilities of the solution search space resulting in better quality solution.

In case of all the maximization functions, there is no improvement in the convergence rates of all the functions for all the considered optimization models nearly after 150th iteration. The convergence of different techniques is different at various levels as shown in the figures. In case of some functions, the FA behaves better than the other algorithms whereas in case of few functions, the other algorithms are better than the FA and other techniques. The major phenomenon behind these differences is the difference in the nature of different optimization techniques to build a relationship between the exploration and exploitation of the solution search space to update their solutions to adapt to the situation.

As shown in the figures, there is no improvement in the convergence rates of all the techniques after a fixed number of iterations. Similar to the minimization functions, the convergence rates of the standard FA to get the most optimal value (in case of maximization functions, the largest value obtained) or at least better value than the value obtained by the standard FA and other conventional optimization algorithms have different capabilities in getting the quality solution.

The differences in the solution qualities of standard FA, GA, ABC, ACO, bat algorithm, DE, GWO, SASFA, FA-CROSS and the proposed FA-PS model are also quite evident in the numerical representations of the solutions provided by these techniques. The performance evaluation parameters of the algorithms are the best case solution, average case solution, best case solution and the standard deviation of the solutions as shown in Table 1 and Table 2. There are sufficient variations in the solutions provided by different standard

Model	Function	FA			GA			ABC		
		t	df	Sig.	t	df	Sig.	t	df	Sig.
	F1	11.2	32.1	0.04	16.3	51.5	0.04	14.1	46.3	0.04
	F2	13.1	51.32	0.04	11.4	35.2	0.03	15.5	44.4	0.03
	F3	11.4	34.3	0.03	11.5	34.3	0.04	12.7	33.7	0.04
	F4	13.6	34.8	0.04	13.6	38.8	0.03	11.6	27.7	0.04
	F5	14.3	43.7	0.04	15.7	46.7	0.04	13.4	37.7	0.04
	F6	9.5	30.4	0.04	12.3	34.6	0.03	12.4	39.7	0.03
	F7	11.5	33.6	0.04	12.6	34.7	0.03	14.6	45.7	0.03
	F8	11.8	34.4	0.03	12.6	34.7	0.03	11.5	34.7	0.04
			ACO			DE		BA		
		t	df	Sig.	t	df	Sig.	t	df	Sig.
FA-DS	F1	9.3	33.1	0.04	11.4	35.2	0.03	11.5	34.3	0.04
FA-1 5	F2	13.4	34.8	0.04	11.2	32.1	0.04	13.6	38.8	0.03
	F3	9.6	25.7	0.04	12.6	34.7	0.03	15.7	46.7	0.04
	F4	12.6	32.6	0.04	16.3	51.5	0.04	14.1	46.3	0.04
	F5	12.6	36.8	0.04	11.5	34.3	0.04	12.6	32.6	0.04
	F6	15.4	44.7	0.03	13.4	37.7	0.04	13.4	34.8	0.04
	F7	14.4	43.7	0.03	9.5	30.4	0.04	13.4	37.7	0.04
	F8	12.5	33.5	0.04	13.4	37.7	0.04	9.5	30.4	0.04
			GWO			SASFA		FA-CROSS		
		t	df	Sig.	t	df	Sig.	t	df	Sig.
	F1	11.6	27.7	0.04	12.6	32.6	0.04	11.5	34.3	0.04
	F2	12.3	34.6	0.03	9.5	30.4	0.04	12.6	36.8	0.04
	F3	12.4	39.7	0.03	13.1	51.32	0.04	14.6	45.7	0.03
	F4	11.5	34.3	0.04	14.4	43.7	0.03	12.4	39.7	0.03
	F5	11.4	35.2	0.03	15.7	46.7	0.04	12.6	32.6	0.04
	F6	12.2	34.1	0.05	12.6	36.8	0.04	13.6	34.8	0.04
	F7	14.3	43.7	0.04	9.5	30.4	0.04	13.1	51.32	0.04
	F8	11.4	35.7	0.03	13.6	34.8	0.04	9.5	30.4	0.04

#### TABLE 4. T-test values of minimization function for proposed model against standard techniques.

optimization algorithms for different minimization and maximization functions. For minimization functions F1 and F2, the standard FA outperforms the standard ABC, GA and its solution quality is a little bit degraded than ACO where as in case of F2, the standard FA is better than ABC and ACO whereas worse than the standard GA. Similarly, for function F3, the standard FA outperforms all the other standard optimization algorithms. If keenly observed all the minimization and maximization functions, similar differences can be found in all the cases for all the considered algorithms. The reason behind these differences has already been outlined in the last few paragraphs. The worst case solution, average case solution, best case solution and the standard deviations of the solutions for FA-PS show that this model outperforms all the other standard optimization algorithms. Similar observations of results can be found in comparison with BA, DE and standard GWO algorithm.

If critically analyzed, all the standard techniques of optimization follow almost the same flow when handling maximization and minimization functions. Resultantly, it is evident that if the optimization algorithms are used in their standard format for solving specific problems, they all have similar strength and pattern for resolving issues. The only distinction comes into existence if there is some modification in the standard algorithm or if it is used in hybridization with other algorithms. The introduction of modification or hybridization gives further strength to the algorithm making it more efficient. Eventually, the performance of the standard FA has been adequately improved when PS is introduced at the termination stage of the FA. The introduction of the PS at the termination stage leads to further processing of the FA which results in better quality solution making the model quite powerful in terms of convergence to minimum or maximum value as requirement of the optimization functions considered. The fairness, concreteness, efficiency and accuracy of the results is quite logical, mathematical and understandable by keeping in considerations the overall analysis and clarifications presented in this regards.

The contributions of this research are very significant and related to swarm intelligence specifically to optimization algorithms. Further, this research activity has a prominent role towards improving the performance of optimization algorithm which paves the way in its application for resolving real world optimization problems by keeping in consideration the operations, model, mathematical working mechanism and logical execution of the developed technique. The contributions of this work can be viewed in many dimensions. The literature has been reviewed and the whole working mechanism of many optimization approaches have been explored in special reference with the standard FA. The problems associated with FA have been identified and explored for clarification of the technique and proper in depth understanding of the drawbacks associated with this newly introduced optimization algorithm. Exploring the standard FA for the identification of unexplored problems and providing an acceptable solution is the major contribution of the work carried out in this activity. In this research total of three major issues related to optimization algorithm were highlighted. Two out of these three problems have not been previously identified explicitly in the literature.

- Firstly, initial solution search space in almost optimization algorithms is generated randomly. This randomly generated initial solution degrades the quality of solution to a considerable amount. The standard FA is also suffered from this limitation.
- Secondly, a major drawback associated with standard FA is its failure in getting the most optimal value due to the fact that after a fixed number of iterations, no significant improvement can be observed in the solution quality.
- 3) In this work, the validity of second problem has been proved and solved by outperforming experimental results targeting the problem.
- 4) Introducing a new hybrid model of FA in combination with PS is the major work performed in this activity.

In this work, an easily implementable model has been developed by making a simple hybridization of two conventional models leading to the development of an efficient mathematical model that gives a distinction to this model from other complicated approaches which involve complicated mathematical operations. The easiness behind the scenario is its straight forward operational steps which have been adequately explained in previous sections. The output values obtained from the standard FA are given as inputs to the PS which further performs its processing and tries to optimize these values to improve the power of the standard FA in terms of getting the most optimal value of the optimization problem being kept under consideration. The starting points of the standard PS are the values obtained from the FA which are further processed in order to enhance the strength and power of the optimization model. No complicated operations are involved in this whole procedure.

Further, huge experimentation has been performed which leads to the development of a result oriented technique which is quite successful in getting the most optimal solution or at least better solution than the solution obtained by the standard FA makes the new model an efficient one. The performance of the developed technique is sufficiently better than other standard models as it has been outlined in the experimental setup section of this work. Since, the PS technique used in the model further performs processing on the values obtained from the standard FA; the improvement in the efficiency of the model is logically, mathematically and experimentally quite eminent. This gives a distinct role to the proposed model as compared to other modified models or hybrid models which are suffered from ambiguity when practically implemented.

Additionally, the comparative analysis of the proposed model with other standard optimization techniques further strengthens the proof that the developed model is better than many conventional optimization algorithms. Lastly, the identification of the problem targeted reveals that almost all optimization models are suffered from the drawback of the failure in getting the most optimal solution in their total number of iterations. The proposed model provides a strong and standard procedure for solving this issue which can be implemented in other optimization models as well by keeping in consideration the operational stages performed in this work.

#### A. LIMITATIONS

Firstly, the major aim of the proposed FA-PS model is to achieve improvement in the optimization capability of conventional FA in obtaining the optimized values but not the complexity of the algorithm. The proposed model has been developed by adding functionalities of PS algorithm to the standard FA. These two optimization algorithms have different types of working mechanism for handling optimization functions. So, embedding the concepts of one algorithm in another algorithm involves few technical procedures. For example, developing hybrid model of FA and PS involves the mapping of fireflies of FA into the PS parameters and converting the PS attributes back to fireflies. During this process in the proposed approach, no operational step has been removed from the conventional FA and PS algorithms, rather new operators have been added. This addition of new operators increases the complexity of the hybrid models. Secondly, the validity of the proposed FA-PS model has been

tested on eight minimization functions and six maximization functions. In order to make more generalization of the experimental results, it is necessary to further perform the experimentation on various types of other uni-modal functions and multi-modal functions and also other multi-objective real world optimization problems with various types of affecting parameters.

#### **VI. CONCLUSION AND FUTURE WORK**

In this work, a serious drawback associated with optimization algorithms has been targeted in special reference to firefly algorithm. Almost all evolutionary intelligence and swarm intelligence algorithms terminate after reaching the maximum iteration number without obtaining the most optimal value in many cases. This problem is the result of imbalanced relationship between the exploration and exploitation capability of the solution search space. This issue of imbalanced relationship between the two properties can be resolved by introducing some modifications in different operational stages of the optimization algorithms. This solution does not give guarantee of the improving the solution quality of the problem to the maximum instant in addition to its failure in identification of the problem location. This research work is focused on resolving the problem at the termination stage of the optimization technique. Firefly algorithm has been considered as the test case for checking the performance of the proposed model for solving optimization problems. The model has been tested on eight minimization functions and six maximization functions. All the results reveals that the proposed optimization solution has fast convergence to the most optimal solution before reaching to the maximum iterations. A t-test has been conducting on both maximization and minimization functions in order to validate the significance of the proposed FA-PS output in a comparison with state-of-art optimization algorithms. As in this research, we had improved the convergence of standard FA for optimization problems, in near future, the solution proposed in this research will be extended to other optimization methods with keeping in considerations the results analyzed in this work.

#### REFERENCES

- L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for largescale machine learning," *SIAM Rev.*, vol. 60, no. 2, pp. 223–311, Jan. 2018.
- [2] F. Wahid, L. H. Ismail, R. Ghazali, and M. Aamir, "An efficient artificial intelligence hybrid approach for energy management in intelligent buildings," *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 12, pp. 5904–5927, 2019.
- [3] G. Villarrubia, J. F. De Paz, P. Chamoso, and F. D. la Prieta, "Artificial neural networks used in optimization problems," *Neurocomputing*, vol. 272, pp. 10–16, Jan. 2018.
- [4] G. M. Khan, Evolution of Artificial Neural Development (Studies in Computational Intelligence). Springer, 2018.
- [5] M. Nouiri, A. Bekrar, A. Jemai, S. Niar, and A. C. Ammari, "An effective and distributed particle swarm optimization algorithm for flexible jobshop scheduling problem," *J. Intell. Manuf.*, vol. 29, no. 3, pp. 603–615, Mar. 2018.
- [6] J. Liebowitz, *The Handbook of Applied Expert Systems*. Boca Raton, FL, USA: CRC Press, 2019.
- [7] S. Shanmuganathan, "Artificial neural network modelling: An introduction," in Artificial Neural Network Modelling. Springer, 2016, pp. 1–14.

- [8] M. Aamir, N. M. Nawi, H. B. Mahdin, R. Naseem, and M. Zulqarnain, "Auto-encoder variants for solving handwritten digits classification problem," *Int. J. FUZZY Log. Intell. Syst.*, vol. 20, no. 1, pp. 8–16, Mar. 2020.
- [9] M. Aamir, N. Mohd Nawi, F. Wahid, and H. Mahdin, "A deep contractive autoencoder for solving multiclass classification problems," *Evol. Intell.*, pp. 1–15, Jun. 2020, doi: 10.1007/s12065-020-00424-6.
- [10] H. T. Nguyen, C. L. Walker, and E. A. Walker, A First Course in Fuzzy Logic. Boca Raton, FL, USA: CRC Press, 2018.
- [11] G. Dudek, "Artificial immune system with local feature selection for short-term load forecasting," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 116–130, Feb. 2017.
- [12] H. K. Cigizoglu and M. Alp, "Generalized regression neural network in modelling river sediment yield," *Adv. Eng. Softw.*, vol. 37, no. 2, pp. 63–68, Feb. 2006.
- [13] R. Pal and D. Roy, "Enhancing saliency of an object using genetic algorithm," in *Proc. 14th Conf. Comput. Robot Vis. (CRV)*, May 2017, pp. 337–344.
- [14] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, and S. Mirjalili, "Henry gas solubility optimization: A novel physicsbased algorithm," *Future Gener. Comput. Syst.*, vol. 101, pp. 646–667, Dec. 2019.
- [15] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.*, vol. 111, pp. 300–323, Oct. 2020.
- [16] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowl.-Based Syst.*, vol. 191, Mar. 2020, Art. no. 105190.
- [17] R. Nayak and R. Padhye, "Artificial intelligence and its application in the apparel industry," in *Automation in Garment Manufacturing*. Amsterdam, The Netherlands: Elsevier, 2018, pp. 109–138.
- [18] G. Wu, R. Mallipeddi, and P. N. Suganthan, "Ensemble strategies for population-based optimization algorithms—A survey," *Swarm Evol. Comput.*, vol. 44, pp. 695–711, Feb. 2019.
- [19] C. Pizzuti, "Evolutionary computation for community detection in networks: A review," *IEEE Trans. Evol. Comput.*, vol. 22, no. 3, pp. 464–483, Jun. 2018.
- [20] A. Slowik and H. Kwasnicka, "Nature inspired methods and their industry applications—Swarm intelligence algorithms," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 1004–1015, Mar. 2018.
- [21] M. Aamir, N. M. Nawi, A. Shahzad, H. Mahdin, and M. Z. Rehman, "A new argumentative based reasoning framework with rough set for decision making," in *Proc. 6th ICT Int. Student Project Conf. (ICT-ISPC)*, May 2017, pp. 1–4.
- [22] G.-C. Chen and J.-S. Yu, "Particle swarm optimization algorithm," *Inf. Control*, vol. 34, no. 3, p. 318, 2005.
- [23] A. Sahu and S. K. Hota, "Performance comparison of 2-DOF PID controller based on moth-flame optimization technique for load frequency control of diverse energy source interconnected power system," in *Proc. Technol. Smart-City Energy Secur. Power (ICSESP)*, Mar. 2018, pp. 1–6.
- [24] T. Jayabarathi, T. Raghunathan, and A. Gandomi, "The bat algorithm, variants and some practical engineering applications: A review," in *Nature-Inspired Algorithms and Applied Optimization*. Springer, 2018, pp. 313–330.
- [25] D. V. Medhane and A. K. Sangaiah, "Search space-based multi-objective optimization evolutionary algorithm," *Comput. Electr. Eng.*, vol. 58, pp. 126–143, Feb. 2017.
- [26] M. Abdullahi, M. A. Ngadi, and S. M. Abdulhamid, "Symbiotic organism search optimization based task scheduling in cloud computing environment," *Future Gener. Comput. Syst.*, vol. 56, pp. 640–650, Mar. 2016.
- [27] S. Bhojanapalli, B. Neyshabur, and N. Srebro, "Global optimality of local search for low rank matrix recovery," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3873–3881.
- [28] S. Varun Kumar and R. Panneerselvam, "A study of crossover operators for genetic algorithms to solve VRP and its variants and new sinusoidal motion crossover operator," *Int. J. Comput. Intell. Res*, vol. 13, no. 7, pp. 1717–1733, 2017.
- [29] M. K. Hon, M. S. Mohamad, A. H. M. Salleh, Y. W. Choon, M. A. Remli, M. A. Ismail, S. Omatu, and J. M. Corchado, "A hybrid of simple constrained artificial bee colony algorithm and flux balance analysis for enhancing lactate and succinate in *Escherichia coli*," in *Proc. Int. Conf. Practical Appl. Comput. Biol. Bioinf.* Springer, 2018, pp. 1–8.

- [30] S. Das, J. R. Doppa, P. P. Pande, and K. Chakrabarty, "Design-space exploration and optimization of an energy-efficient and reliable 3-D smallworld network-on-chip," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 5, pp. 719–732, May 2017.
- [31] I. Aljarah, H. Faris, and S. Mirjalili, "Optimizing connection weights in neural networks using the whale optimization algorithm," *Soft Comput.*, vol. 22, no. 1, pp. 1–15, Jan. 2018.
- [32] B. Yang, X. Zhang, T. Yu, H. Shu, and Z. Fang, "Grouped grey wolf optimizer for maximum power point tracking of doubly-fed induction generator based wind turbine," *Energy Convers. Manage.*, vol. 133, pp. 427–443, Feb. 2017.
- [33] S. Sumathi, T. Hamsapriya, and P. Surekha, *Evolutionary Intelligence:* An Introduction to Theory and Applications With MATLAB. Springer, 2008.
- [34] C. Blum and D. Merkle, "Swarm intelligence," in *Swarm Intelligence in Optimization*, C. Blum and D. Merkle, Eds. Springer, 2008, pp. 43–85.
- [35] A. Pregelj, M. Begovic, and A. Rohatgi, "Recloser allocation for improved reliability of DG-enhanced distribution networks," *IEEE Trans. Power Syst.*, vol. 21, no. 3, pp. 1442–1449, Aug. 2006.
- [36] D. J. Munk, G. A. Vio, and G. P. Steven, "Topology and shape optimization methods using evolutionary algorithms: A review," *Struct. Multidisciplinary Optim.*, vol. 52, no. 3, pp. 613–631, Sep. 2015.
- [37] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 124–141, Jul. 1999.
- [38] L. Hong, J. H. Drake, J. R. Woodward, and E. Özcan, "A hyper-heuristic approach to automated generation of mutation operators for evolutionary programming," *Appl. Soft Comput.*, vol. 62, pp. 162–175, Jan. 2018.
- [39] K. Li, Y. Chen, W. Li, J. He, and Y. Xue, "Improved gene expression programming to solve the inverse problem for ordinary differential equations," *Swarm Evol. Comput.*, vol. 38, pp. 231–239, Feb. 2018.
- [40] M. P. Saka and E. Dogan, "Recent developments in metaheuristic algorithms: A review," *Comput. Technol. Rev.*, vol. 5, no. 4, pp. 31–78, Sep. 2012.
- [41] M. Geitle and R. Olsson, "Using automatic programming to design improved variants of differential evolution," in *Proc. 21st Asia Pacific Symp. Intell. Evol. Syst. (IES)*, Nov. 2017, pp. 13–18.
- [42] B. D. Bryant and R. Miikkulainen, "A neuroevolutionary approach to adaptive multi-agent teams," in *Foundations of Trusted Autonomy*. Cham, Switzerland: Springer, 2018, pp. 87–115.
- [43] M. Aamir, N. Mohd, F. Wahid, and H. Mahdin, "An efficient normalized restricted Boltzmann machine for solving multiclass classification problems," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 8, pp. 416–426, 2019, doi: 10.14569/IJACSA.2019.0100856.
- [44] R. J. Urbanowicz and W. N. Browne, Introduction to Learning Classifier Systems. Springer, 2017.
- [45] O. Deepa and A. Senthilkumar, "Swarm intelligence from natural to artificial systems: Ant colony optimization," in *Proc. Netw. (GRAPH-HOC)*, 2016, vol. 8, no. 1, pp. 9–17.
- [46] S. Mirjalili, "Ant colony optimisation," in Evolutionary Algorithms and Neural Networks. Springer, 2019, pp. 33–42.
- [47] P.-W. Tsai, J.-S. Pan, B.-Y. Liao, and S.-C. Chu, "Enhanced artificial bee colony optimization," *Int. J. Innov. Comput., Inf. Control*, vol. 5, no. 12, pp. 5081–5092, 2009.
- [48] J. C. Bansal, A. Gopal, and A. K. Nagar, "Stability analysis of artificial bee colony optimization algorithm," *Swarm Evol. Comput.*, vol. 41, pp. 9–19, Aug. 2018.
- [49] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," Adv. Eng. Softw., vol. 69, pp. 46–61, Mar. 2014.
- [50] M. Kohli and S. Arora, "Chaotic grey wolf optimization algorithm for constrained optimization problems," *J. Comput. Des. Eng.*, vol. 5, no. 4, pp. 458–472, Oct. 2018.
- [51] S. Mirjalili and A. Lewis, "The whale optimization algorithm," Adv. Eng. Softw., vol. 95, pp. 51–67, May 2016.
- [52] S. Mirjalili, "The ant lion optimizer," Adv. Eng. Softw., vol. 83, pp. 80–98, May 2015.
- [53] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.
- [54] G.-G. Wang, S. Deb, and L. dos Santos Coelho, "Earthworm optimisation algorithm: A bio-inspired metaheuristic algorithm for global optimisation problems," *Int. J. Bio-Inspired Comput.*, vol. 12, no. 1, pp. 1–22, Jan. 2018.
- [55] G.-G. Wang, S. Deb, and L. D. S. Coelho, "Elephant herding optimization," in *Proc. 3rd Int. Symp. Comput. Bus. Intell. (ISCBI)*, Dec. 2015, pp. 1–5.

- [56] G.-G. Wang, "Moth search algorithm: A bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Comput.*, vol. 10, no. 2, pp. 151–164, Jun. 2018.
- [57] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems," *Eng. Comput.*, vol. 29, no. 1, pp. 17–35, Jan. 2013.
- [58] X. Yang and A. H. Gandomi, "Bat algorithm: A novel approach for global engineering optimization," *Eng. Comput.*, vol. 29, no. 5, pp. 464–483, Jul. 2012.
- [59] X.-S. Yang, "Firefly algorithm, Levy flights and global optimization," in *Research and Development in Intelligent Systems XXVI*. Springer, 2010, pp. 209–218.
- [60] I. Fister, X.-S. Yang, J. Brest, and I. Fister, Jr., "Modified firefly algorithm using quaternion representation," *Expert Syst. Appl.*, vol. 40, no. 18, pp. 7220–7230, Dec. 2013.
- [61] I. Fister, Jr., M. Perc, S. M. Kamal, and I. Fister, "A review of chaos-based firefly algorithms: Perspectives and research challenges," *Appl. Math. Comput.*, vol. 252, pp. 155–165, Feb. 2015.
- [62] S. L. Tilahun and J. M. T. Ngnotchouye, "Firefly algorithm for discrete optimization problems: A survey," *KSCE J. Civil Eng.*, vol. 21, no. 2, pp. 535–545, Feb. 2017.
- [63] C. Soto, F. Valdez, and O. Castillo, "A review of dynamic parameter adaptation methods for the firefly algorithm," in *Nature-Inspired Design* of Hybrid Intelligent Systems. Springer, 2017, pp. 285–295.
- [64] S. L. Tilahun, J. M. T. Ngnotchouye, and N. N. Hamadneh, "Continuous versions of firefly algorithm: A review," *Artif. Intell. Rev.*, vol. 51, no. 3, pp. 445–492, Mar. 2019.
- [65] N. Dey, J. Chaki, L. Moraru, S. Fong, and X.-S. Yang, "Firefly algorithm and its variants in digital image processing: A comprehensive review," in *Applications of Firefly Algorithm and Its Variants*. Springer, 2020, pp. 1–28.
- [66] S. Palit, S. N. Sinha, M. A. Molla, A. Khanra, and M. Kule, "A cryptanalytic attack on the knapsack cryptosystem using binary firefly algorithm," in *Proc. 2nd Int. Conf. Comput. Commun. Technol. (ICCCT)*, Sep. 2011, pp. 428–432.
- [67] S. M. Farahani, A. Abshouri, B. Nasiri, and M. Meybodi, "A Gaussian firefly algorithm," *Int. J. Mach. Learn. Comput.*, vol. 1, no. 5, p. 448, 2011.
- [68] L. dos Santos Coelho, D. L. de Andrade Bernert, and V. C. Mariani, "A chaotic firefly algorithm applied to reliability-redundancy optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2011, pp. 517–521.
- [69] M. Subotic, M. Tuba, and N. Stanarevic, "Parallelization of the firefly algorithm for unconstrained optimization problems," *Latest Adv. Inf. Sci. Appl.*, vol. 22, no. 3, pp. 264–269, 2012.
- [70] S. K. Agarwal and S. Yadav, "A comprehensive survey on artificial bee colony algorithm as a frontier in swarm intelligence," in *Ambient Commu*nications and Computer Systems. Springer, 2019, pp. 125–134.
- [71] S. Duman, "Symbiotic organisms search algorithm for optimal power flow problem based on valve-point effect and prohibited zones," *Neural Comput. Appl.*, vol. 28, no. 11, pp. 3571–3585, Nov. 2017.
- [72] U. Singh and R. Salgotra, "Synthesis of linear antenna arrays using enhanced firefly algorithm," *Arabian J. Sci. Eng.*, vol. 44, no. 3, pp. 1961–1976, Mar. 2019.
- [73] A. Khelifi, B. Bentouati, and S. Chettih, "Optimal power flow problem solution based on hybrid firefly krill herd method," *Int. J. Eng. Res. Afr.*, vol. 44, pp. 213–228, Aug. 2019.
- [74] X.-S. Yang, "Firefly algorithm," in *Nature-Inspired Metaheuristic Algorithms*, vol. 20. Luniver Press, 2008, pp. 79–90.
- [75] E. Cardelli, A. Faba, M. Pompei, and S. Quondam Antonio, "Pattern search approach to ferromagnetic material modelling," *Int. J. Numer. Model.*, *Electron. Netw., Devices Fields*, vol. 32, no. 4, p. e2271, Jul. 2019.



**FAZLI WAHID** received the B.S. degree in computer science from the University of Malakand, Pakistan, in 2006, the M.S. degree in computer science from SZABIST, Islamabad, Pakistan, in 2015, and the Ph.D. degree in information technology from the University Tun Hussein Onn Malaysia. His areas of interest are energy consumption prediction, optimization, and management using multilayer perceptron, artificial bee colony, ant colony, swarm intelligence, and other machine learning techniques.



**M. SULTAN ZIA** received the M.S.-C.S. and Ph.D. degrees in computer science from the FAST-National University of Computer and Emerging Sciences, Islamabad, in 2008 and 2016 respectively, and the M.C.S. degree from International Islamic University, Islamabad, in 2005. He has 15 years of total teaching/research experience at university level and 4.5 years of industrial (software development) experience. He is currently an Associate Professor and the Head of the Depart-

ment of Computer Science and IT, The University of Lahore, Chenab Campus. His research interest includes the IoT, machine learning, and computer vision.



**RAO NAVEED BIN RAIS** received the M.S. and Ph.D. degrees in computer engineering with specialization in networks and distributed systems from the University of Nice, Sophia Antipolis, France, in 2007 and 2011, respectively. He is currently working as an Associate Professor with the Department of Electrical and Computer Engineering, College of Engineering and Information Technology, Ajman University, UAE. He has experience of more than 15 years in teaching,

research, and industrial development. His research interests include network protocols and architectures, information-centric and software-defined networks, network virtualization, machine learning, and Internet naming and addressing issues.



**MUHAMMAD AAMIR** received the master's degree in computer science from the City University of Science and Information Technology, Pakistan, and the Ph.D. degree in information technology from University Tunn Hussien Onn Malaysia. He had worked for two years in Xululabs LLC as a Data Scientist. He is currently working on research related to big data processing and data analysis. His fields of interest are data science, deep learning, and computer programming.



**UMAIR MUNEER BUTT** received the B.S. (CS) degree from GIFT University, Pakistan, in 2012, and the M.S. (CS) degree from the National University of Sciences and Technology (NUST), Pakistan, in 2016. He is currently pursuing the Ph.D. degree with the School of Computer Science, Universiti Sains Malaysia. He has more than three years of teaching and research experience in machine learning, data science, and image processing. He has served as a Research Associate

for three years and worked on different real-world applications. His current research interests are data science, data mining, and machine learning.



**MUBASHIR ALI** received the B.S. degree in computer science from Allama Iqbal Open University Islamabad, Pakistan, in 2011, and the M.S. degree in software engineering from Bahria University Islamabad, Pakistan, in 2014. He is currently pursuing the Ph.D. degree with the School of Engineering and Applied Sciences, The University of Bergamo, Italy. He has also served as a Software Engineer for more than five years in research and development-based public sector organizations in

Pakistan. His research interests include NLP, machine learning, data science, social media analysis, and software repository mining.



**ADEEL AHMED** (Graduate Student Member, IEEE) received the M.Phil. degree in computer science from Quaid-i-Azam University, Islamabad, Pakistan, in 2011, where he is currently pursuing the Ph.D. degree with the Department of Computer Science. He has software industry experience for few years and his research interests include social network analysis, recommender systems, machine learning, and swarm intelligence.



**IMRAN ALI KHAN** received the master's degree from Gomal University, D. I. Khan, Pakistan, and the Ph.D. degree from the Graduate University Chinese Academy of Sciences, China. He is currently working as an Associate Professor with the Department of Computer Science, COM-SATS University Islamabad, Abbottabad Campus, Pakistan. He has produced over 50 publications in journal of international repute and presented papers in international conferences. His research

areas include wired and wireless networks, and distributed systems.



**OSMAN KHALID** received the master's degree from the Center for Advanced Studies in Engineering and the Ph.D. degree from North Dakota State University, USA. He is currently an Assistant Professor with COMSATS University Islamabad, Abbottabad Campus. His research areas include: recommender systems, network routing protocols, the Internet of Things, and fog computing.

...

148288