Perfectionism Search Algorithm (PSA): An Efficient Meta-Heuristic Optimization Approach

Amin Ghodousian[*1], Mahdi Mollakazemiha[†2] and Noushin Karimian[‡3]

[1]School of Engineering Science, College of Engineering, University of Tehran, Tehran, Iran.
[2]Department of Mathematical Sciences, Sharif University of Technology, Tehran, Iran.
[3]Department of Electrical and Electronic Engineering, The University of Manchester, Manchester, UK

ABSTRACT

This paper proposes a novel population-based meta-heuristic optimization algorithm, called Perfectionism Search Algorithm (PSA), which is based on the psychological aspects of perfectionism. The PSA algorithm takes inspiration from one of the most popular model of perfectionism, which was proposed by Hewitt and Flett. During each iteration of the PSA algorithm, new solutions are generated by mimicking different types and aspects of perfectionistic behavior. In order to have a complete perspective on the performance of PSA, the proposed algorithm is tested with various nonlinear optimization problems, through selection of 35 benchmark

*Corresponding author: A. Ghodousian. Email: a.ghodousian@ut.ac.ir
†mollakazemiha.mahdi@gmail.com
‡noushin.karimian@manchester.ac.uk

1   Abstract continued

functions from the literature. The generated solutions for these problems, were also compared with 11 well-known meta-heuristics which had been applied to many complex and practical engineering optimization problems. The obtained results confirm the high performance of the proposed algorithm in comparison to the other well-known algorithms.

2   Introduction

Over the past decades, development and application of optimization models have attracted growing attention amongst mathematicians and engineers. In recent years, , they have received enormous attention, due to the rapid progress of computer technology, including development and availability of different types of user-friendly software, high-speed and parallel processors, and artificial neural networks. Even though most practical optimization problems have some restrictions that must be satisfied, study of available techniques for unconstrained optimization problems is important for several reasons. For example, many algorithms solve a constrained problem by converting it into a sequence of unconstrained problems via either Lagrangian multipliers or penalty and barrier functions. Moreover, most methods proceed by finding an improving direction and then function minimization takes place along this direction. This line search is equivalent to minimizing a function of one variable without constraints or with simple constraints, such as lower and upper bounds of the variables [2]. It is worth stating that, improving directions are the backbone of many well-known optimization algorithms such as the Cyclic Coordinate method, the Hooke and Jeeves, the Rosenbrock, the Steepest Descent method, the Newton's method and its modifications, and Quasi-Newton and Conjugate Gradient methods.

Although, the above-mentioned mathematical based methods are fast, quick in finding a local minimum, and can guarantee global optima in both simple and ideal models, but they suffer from limitations, as the optimality conditions and convergence of these procedures all rely on some specific assumptions. For instance, some types of convexity, the first (or higher) order differentiability of an objective function; continuity, linearity or the order of nonlinearity of constraints; the linear independence or orthogonality of directions along which an objective function is optimized and the closeness of some mappings or regions in problems. Furthermore, to guarantee convergence to the global optimum, some algorithms require careful attention in selecting the initial values. However, these assumptions do not hold true for many problems, and in any case, they cannot be easily verified.

On the other hand, over the past decades, there has been a growing interest in the development of meta-heuristic algorithms. These algorithms are inspired by the behaviors of natural phenomena and do not require any consideration of previous prerequisites [31]. Thus, meta-heuristics can allow for near-optimum solutions to be found within a reasonable computation-time, with efficient use of memory, without any loss of subtle nonlinear characteristics of the model and any need for complex derivatives or gradient information,

and require careful selection of initial values. Metaheuristic optimization algorithms rely on rather simple concepts and are easy to implement. Moreover, these algorithms can bypass local optima and applied to a wide range of problems covering different disciplines. Additionally, it has been experimentally shown that these algorithms which are able to perform well, can provide suitable solutions for complex engineering optimization problems. These algorithms are in fact, good alternative approach to traditional methods, especially for solving combinatorial problems, NP-hard problems and problems in which the search space grows exponentially with the size of the problem [18]. For all these reasons, many meta-heuristics were designed to solve various types of applied problems such as Scheduling problems [56], pattern recognition [15], data clustering [37], tuning of neural networks [33], data mining [64], engineering [5,20] and optimization [4,22,36,54,55]. Also, in recent years, there has been a growing attempt in developing algorithms inspired by nature, providing some proofs of convergence, solving different classes of optimization problems and investigating some other theoretical analyses [11,18].

Nature-inspired metaheuristic algorithms can be roughly classified into four categories according to their inspirations: evolution-based, physics-based, swarm-based and human-based methods. Evolution-based methods are inspired by the laws of natural evolution. The most popular evolution-inspired technique is Genetic Algorithm [27]. Other popular algorithms are Biogeography Based Optimization (BBO) [44], Evolution Strategy (ES) [42] and Probability Based Incremental Learning (PBIL) [9]. Physics-based methods imitate the physical rules in the universe. Examples of these algorithms include Gravitational Search Algorithm (GSA) [41], Black Hole (BH) algorithm [22], Simulated Annealing [32], Central Force Optimization (CFO) [14] and Water wave optimization (WWO) [65]. The third group of nature-inspired methods includes swarm-based techniques that imitate the social behavior of groups of animals. The most popular algorithms are Particle Swarm Optimization (PSO) [29] and Ant Colony Optimization (ACO) [10,48]. Other examples of swarm-based techniques are Bees Algorithm (BA) [38], Firefly Algorithm (FA) [61], Whale Optimization Algorithm (WOA) [35], Forest Optimization Algorithm (FOA) [17], Social Spider Optimization (SSO) [8], Cuckoo Optimization Algorithm (COA) [39], Grey Wolf Optimizer (GWO) [34], Lion Pride Optimizer (LPO) [58] and Bat-inspired Algorithm (BA) [60]. The final category consists of methods which are inspired by human behaviors. Some of the most popular algorithms are Teaching Learning Based Optimization (TLBO) [40], Harmony Search (HS) [16], Cultural Algorithm (CA) [43], Group Search Optimizer (GSO) [23], Exchange Market Algorithm (EMA) [19] and Group Counseling Optimization (GCO) algorithm [12].

In addition to the fundamental differences between these algorithms in terms of their inspirations, they also differ in many other ways such as: types of selection strategies, the generation of new solutions and the definition of exploitation and exploration operators. For instance, CFO, Stochastic Algorithm (SA) and (1+1)- Evolution Strategy (ES) [30] are the three approaches that are shown to be very different amongst the heuristic algorithms. CFO is a deterministic heuristic search algorithm which works based on the metaphor of gravitational kinematics and does not use any random parameter in its formulation; SA is a stochastic algorithm in which the search starts from a single point

and continues in a sequential manner; and (1+1) - ES is a simple evolutionary algorithm with one parent generating one offspring per iteration. However, most of the heuristic algorithms have a stochastic behavior, where they do search in a parallel manner with multiple initial points and generate several offsprings from several parents. In spite of the previously mentioned differences among the heuristic algorithms, population-based meta-heuristic optimization algorithms share a common feature regardless of their nature. The search process is divided into two phases: (i) exploration - the ability to expand the search space, and (ii) exploitation - the ability to find the optima around a good solution [1]. Finding a right balance between exploration and exploitation is the most challenging task in the development of any metaheuristic algorithm. Moreover, the individuals within population-based algorithms usually pass three steps to complete the two phases of exploration and exploitation: (i) self-adaptation (each individual improves its performance), (ii) cooperation (individuals cooperate with each other through transfer of information), and (iii) competition (individuals compete to survive).

It is important to mention that there is no single algorithm to achieve the most appropriate solutions for all classes of problems. Some algorithms, in comparison to others, provide better solutions for some particular problems. Generally, the effectiveness of different optimization algorithms vary based on their generality, reliability, precision, sensitivity to parameters and data, computational effort and convergence. Therefore, pursuing for a new optimization technique based on its similarities with either natural or artificial phenomena is an open problem [59]. This study describes a novel human-based meta-heuristic optimization algorithm (namely, Perfectionism Search Algorithm - PSA). The source of inspiration for PSA is the behaviors of extreme perfectionists from a psychological perspective. Perfectionism is a multi-dimensional personality trait characterized by a person's continuous striving for flawlessness and setting exceedingly high standards of performance, accompanied by tendencies for overly critical evaluations of one's own behavior [52]. Perfectionism is best conceptualized as a multi-dimensional characteristic [51], since psychologists agree that such type of characteristic consists of many positive and negative aspects. The maladaptive form of perfectionism, drives people to try and achieve an unattainable ideal, while the adaptive form of perfectionism can motivate them to reach their goals. When perfectionists do not reach their goals, they often fall into depression [62]. Recognizing that perfectionism has personal and social dimensions, Hewitt and Flett [24] proposed a model with an ability to distinguish between the three forms of perfectionism: Self-Oriented perfectionism (SOP), Other-Oriented perfectionism (OOP), and Socially Prescribed Perfectionism (SPP), where the three forms comprise of different attitudes, motivations, and behaviors.

The model proposed by Hewitt and Flett which explains the concept of perfectionism has been the main source of inspiration for the development of PSA algorithm. In the proposed algorithm, the above-mentioned types of perfectionism are mathematically modeled. Then, PSA assigns a probability to each type, which are dynamically changed by each round of iterations. During each iteration, PSA attempts to generate perfect solutions; i.e. solutions with higher objective function values. In order to generate a new solution, the algorithm initially selects one type of perfectionism based on its associated

probability, and then generates a solution according to the behaviors of perfectionists belonging to that particular group. In each case, a comparison is made to decide if the new generated solution is a good solution or not (compared with the worst solution in the population). If the generated solution is not a good one, the probability of the selected type of perfectionism is decreased for the next round of generation. This case is correlating to the depression case when perfectionists do not reach their goals. The efficiency of the PSA algorithm proposed and developed in this research is evaluated by solving 35 mathematical optimization problems. Optimization results from these problems demonstrate that PSA is an efficient algorithm for complex optimization problems compared to the 11 well-known metaheuristic algorithms.

The remainder of this paper is organized as follows. In section 2, some characteristics of SOP, OOP and SPP perfectionists are briefly discussed. This section provides some background information which are the psychological sources of inspiration for the proposed algorithm. In Section 3, the proposed algorithm (PSA) is outlined, and its characteristics and implementation steps are explained in details. Section 4 provides a discussion of the proposed approach which presents the most considered theoretical aspects and applied suggestions concerning PSA. Comparative study, test problems and experimental results are presented and discussed in Section 5. In this section, the performance of the presented algorithm is illustrated with several benchmark datasets and compared with ACO, PSO, TLBO, BBO, Differential Evolution (DE) [53], HS, FA, WOA, BA, GSA and BH. Finally, Section 6 summarizes the main findings of this study and suggests directions for future research.

3   Basic Rules for PSA – A Brief Conceptual Discussion on Perfectionism

As mentioned in the previous section, Hewitt and Flett's tripartite perfectionism model divides perfectionism into three forms of SOP, OOP and SPP. In the following, a brief overview of important characteristics for these three forms will be provided. This discussion provides a complete background into creating an outline of PSA, which is done by deriving the basic aspects from the perfectionistic behaviors. In the next section, these basic rules are mathematically modeled in order to perform the optimization.

3.1   Self-Oriented Perfectionism (SOP)

SOP reflects beliefs that are striving for perfection and generally, being perfect is important. Self-oriented perfectionists have exceedingly high personal standards from themselves, continuous expectation of being perfect, and are very self-critical if they fail to meet the expected criteria and demands [24]. In addition to the above, SOP can be considered to be double-edged, in a way that can also be associated to some positive characteristics such as conscientiousness, nurturing, intimacy, social development, and altruism [49,51]. On the other hand, these characteristics often cause self-esteem deficits

and self-evaluation and make individuals more prone to depression [46].

3.2 Other-Oriented Perfectionism (OOP)

OOP reflects beliefs, where it is important for others to strive for perfection and be perfect [24,26]. Other-oriented perfectionists have exceedingly high standards from others, expect others to be perfect, and are also highly critical of others who fail to meet the expectations [24,26]. So, individuals with tendencies towards OOP have behaviors that are similar to self-oriented perfectionists, with a difference of behavior directed outward towards others rather than requiring the self to be perfect. Such characteristics result in lack of trust towards others [24]. OOP is similarly double-edged and displays positive associations with grandiose narcissism, Machiavellianism, psychopathy, and aggressive humor [45, 49,50]. Nonetheless, OOP is also tied to lower burnout [7], superior problem solving [13], and positive self-regard [49].

3.3 Socially Prescribed Perfectionism (SPP)

Socially prescribed perfectionists believe that exceedingly high standards are being imposed on them. They believe others expect them to be perfect, and think that others will be highly critical of them if they fail to meet their expectations [24,26]. SPP involves a tendency towards having a fear of negative social evaluation and desire for approval from those around them. When the perceived standards are not met, individuals blame themselves and lower their self-worth, consequently resulting in depressive symptoms [24]. Socially prescribed perfectionists think and behave in ways that generate stress, which in turn increases the risk for depression disorder [25].

For ease of future reference and before going into in-depth description of the PSA algorithm, the basic rules of PSA algorithm, which have been derived from the above statements, will be summarized below. The generated solution from PSA algorithm applied to an optimization problem, is based on the following two main rules:

Rule1 (Striving step) – Perfectionists strive for flawlessness, set high standards of performance and have tendencies for overly critical evaluations of one's behavior [52].

Rule2 (Depression step) – When perfectionists do not reach their expectation, they often fall into depression mood [62].

In PSA, implementation of the striving step is conducted by modeling some basic characteristics of the above three types of perfectionism. These characteristics can be expressed as follows:

CHARACTERISTICS OF SELF-ORIENTED PERFECTIONISTS IN PSA

SOP1	They expect to be perfect [24].
SOP2	They have exceedingly high personal standards [24].

CHARACTERISTICS OF OTHER-ORIENTED PERFECTIONISTS IN PSA

OOP1	They expect others to be perfect [24,26].
OOP2	They have high standards from others [24,26].
OOP3	They have lack of trust towards others [24,26].
OOP4	They display a positive correlation with narcissism [45,49,50].

CHARACTERISTICS OF SOCIALLY PRESCRIBED PERFECTIONISTS IN PSA

SPP1	They believe others expect them to be perfect [24].
SPP2	They desire approval from those around them [24].

Also, the depression step is performed by modeling the rate of depression based on the following observations:
DEPRESSION EFFECTS

Dep1	When perceived standards are not met, the characteristics of SOP and SPP can make individuals more prone to depression [25,46].
Dep2	OOP is tied to lower burnout [7] and positive self-regard [49].

4   Mathematical Model and Optimization Algorithm

PSA is initialized with random solution vectors (individuals or persons). For generating new solution, the algorithm iterates over two main steps called Striving and Depression. Striving step is done in two phases. In phase one, the PSA selects one of the three types of perfectionism SOP, OOP and SPP according to their associated probabilities $P_{SOP}, P_{OOP}, P_{SPP}$, respectively. In phase two, a novel solution is produced with regards to the mathematical model of the selected type of perfectionism. In order to achieve this, SOP, OOP and SPP are individually mathematically modeled as three operators, based on their characteristics.

After generating a new solution, the depression step is executed. If the new solution has a desirable performance (i.e., it is better than the previously obtained worst solution in the population), algorithm continues to progress by randomly selecting a type of perfectionism and generating another solution. Otherwise, the probability of the selected type is updated. Fig. 1 shows the flow chart for a typical iteration performed by PSA. Moreover, in order to further improve the performance of the algorithm (in terms of complexity), PSA has been made to control the number of function evaluations by generating as many new solutions as the population size.

Figure 1: Outline of PSA

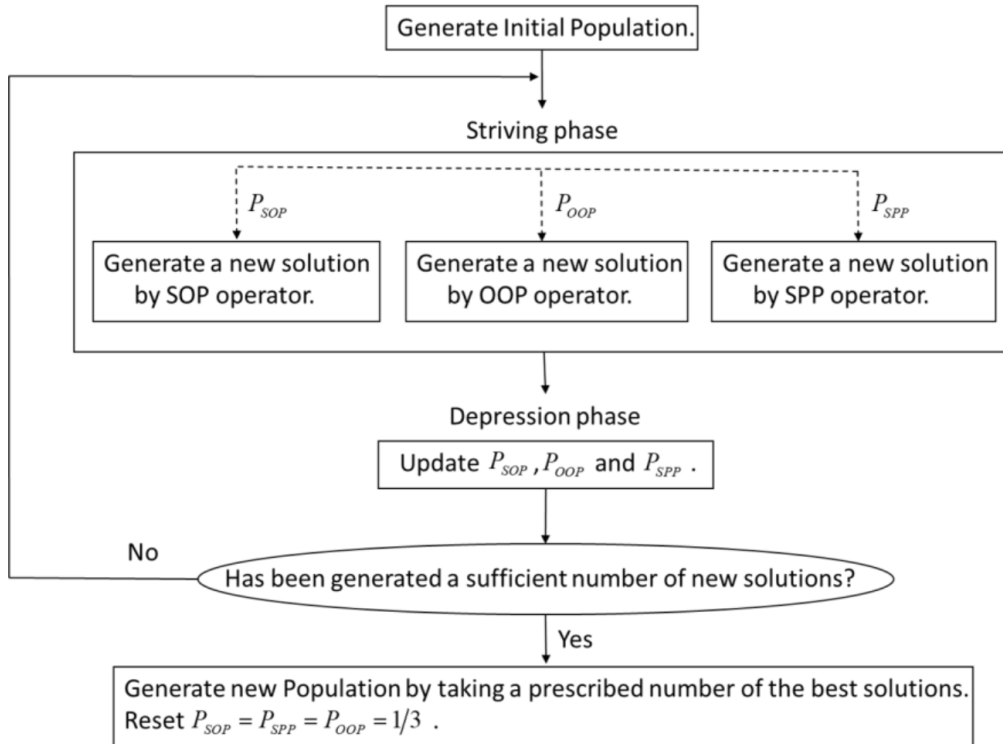In the following subsections, the details of the different steps, consisting of generating the initial individuals and performing the striving and depression steps, will be presented. Then, the mathematical models of SOP, OOP and SPP are provided. The detailed description of how the depression step is created will also be presented later, followed by the complete description of the PSA algorithm.

4.1   Initialization

As mentioned earlier, the first step is to randomly generate the population over the solution space. The objective function value for each solution is treated as the performance of that solution, and PSA focuses on achieving solutions with high levels of performance. In an n-dimensional optimization problem, a solution (individual or person) is represented as $x = (x_1, x_2, ..., x_n)$ where n is the number of the decision variables (such as activities, traits or behavioral factors of that person). The performance of each solution $x = (x_1, x_2, ..., x_{n)}$ is computed by evaluating its objective function value, $f(x) = f(x_1, x_2, ..., x_n)$. Also, the number of solution vectors in the population is denoted by N which is kept constant during the optimization process. During the searching process, the population update is accomplished by adding a set of newly generated solutions to the population and then removing the same number of worst solutions from the population; this is done in order to maintain the same number for the total population size (N). This process ensures that only the best solutions are kept within the population, therefore adhering to our main inspiration, i.e., perfectionism - as the name suggests. At the end of each iteration, solutions $X_1, X_2, ..., X_N$ are put in order (sorted) from the best to worst according to their objective values (their performance values), i.e., for a minimization problem: $f(X_1) \leq ... \leq f(X_N)$. So, X_1 is the best solution (the solution with the smallest objective value), X_2 is the solution with the second smallest value and so forth, up to X N that is the solution with the largest objective value (i.e. the worst solution).

As mentioned before, for each new solution, the generation process is carried out over two phases. Phase one consists of choosing one of the three types of perfectionism. Phase two consists of using the characteristics from the chosen type to generate a new solution. In the following section, the mathematical formulations for SOP, OOP and SPP as optimization operators are explained in more details.

4.2   Self-Oriented Perfectionism (SOP)

SOP operator generates a new solution $X' = \{x'_1, x'_2, ..., x'_n\}$ by modeling the characteristics SOP1 and SOP2, which were defined in section 2. In this case, new solution X' is considered as a self-oriented perfectionist who expects to be perfect (SOP1) and has exceedingly high personal standards (SOP2). For this purpose, 10% of the best solutions (i.e., 10% of the first solutions in the sorted population) are considered as a set of patterns for X' The X' perfectionist, in the first instance, randomly selects a solution (person) $X^* = (x_1^*, x_2^*, ..., x_n^*)$ as a finite sequences $\{(j, x_j^*)\}_{j=1}^n$ including n terms in the xy-plane. So, finite sequence $\{(j, x_j^*)\}_{j=1}^n$ can be shown as number of points in the plane where the horizontal axis (x) is the index number of the term j , and the vertical axis (y) is its value, x_j^* . Since this finite sequence is a function whose domain is set $1, 2, ..., n$, its graph (denoted by G^*) consists of some isolated points with coordinates $(1, x_1^*), (2, x_2^*), ..., (n, x_n^*)$. In PSA algorithm, Graph G^* is interpreted as the behavior of X^* . An example of how such a (behavioral) graph may look like is depicted in Fig. 2.
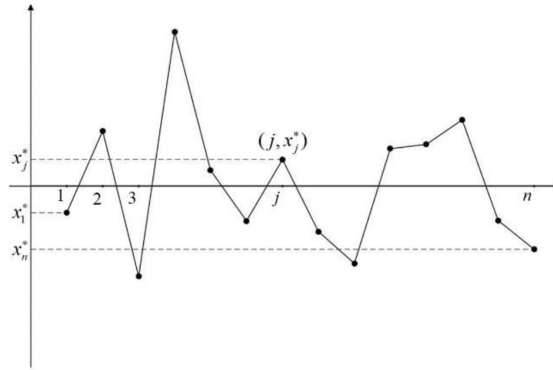
Figure 2: Graph G^* : the interpretation of the behavior of point X^* .

A new solution X' is generated by imitating the behavior of X^* . In doing so, a random pattern X * is first selected from the set of patterns. Then, SOP operator chooses a random component $j_0 \in \{1, 2, ..., n\}$, and assigns the value of the j'th component from the selected solution (x^*), to the corresponding component of the new solution (x'_{j_0}). Thus, we will have $(j_0, x_{j_0}^*) = (j_0, x'_{j_0})$. Point $(j_0, x_{j_0}^*)$ (or equivalently, (j_0, x'_{j_0}) is labelled by "*" in Fig. 3. Finally, other components $x'_k (k \neq j_0)$ are then generated by the following equation:

$$x'_k = x'_{j_0} + r_k(x_k^* - x_{j_0}^*), \quad k \in \{1, 2, ..., n\} - \{j_0\} \tag{1}$$

Where r_k is a uniform random number between -2 and 2 . Algorithm 1, as shown below presents the generation of a new solution $X' = \{x'_1, x'_2, ..., x'_n\}$ using SOP operator.

Algorithm 1 (SOP operator)

Randomly select solution (pattern) x^* from $\{X_1, X_2, ..., X_{\frac{N}{10}}\}$

Randomly choose a component $x^*_{j_0}$ ($j_0 \in \{1, 2, ..., n\}$) from the selected solution $X^* = (x^*_1, x^*_2, ..., x^*_n)$.

Set $x'_{j_0} = x^*_{j_0}$

for each $k \in \{1, 2, ..., n\} - \{j_0\}$

    Choose a uniform random number; $r_k \in U(-2, 2)$

    Generate x'_k byEquation(1).

end for

A clear visualization of how a new position is generated using SOP operator is presented in Fig. 3, where in graph G^*, the behavior of X^* is shown by solid line and in graph G', the behavior of X' is shown by dotted line, which are all sawtooth-shaped. Graph G^* includes points (k, x^*_k) – shown by black circles – and G' consists of points k (k, x'_k) – shown by white circles.

Figure 3: Graph G' : the imitation of the behavior of X^* by new solution X' .

To illustrate the rationale behind equation (1), graphs G^* and G' need to be considered, and since $x'_{j_0} = x^*_{j_0}$, two points $(j_0, x^*_{j_0}) \in G^*$ and $(j_0, x'_{j_0}) \in G'$ coincide and are equal to each other. Then, G' is obtained from G^* by replacing every point of $(k, x^*_k) \in G^*$ by point $(k, x^*_{j_0} + r_k(x^*_k 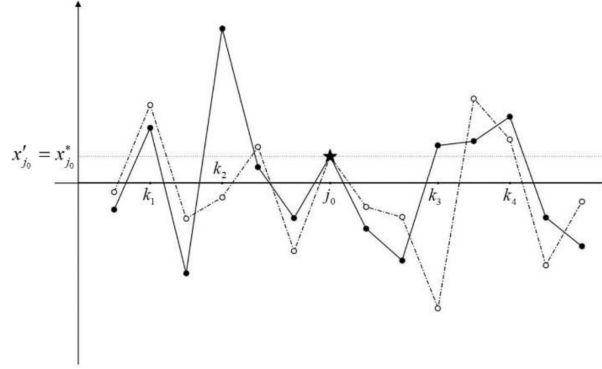- x^*_{j_0}))$ whose vertical difference from the point $(j_0, x^*_{j_0})$ has been enlarged if $r_k > 1$ and reduced if $r_k < 1$. This construction can be called r_k-fold vertical expansion of the graph G^*, where the term "expansion" allows for enlargement ($r_k > 1$, e.g., point $(k_1, x^*_{k_1})$ as shown in Fig.3). no change at all ($r_k = 1$, e.g., point $(j_0, x^*_{j_0})$) and reduction ($0 < r_k < 1$, e.g., point $(k_4, x^*_{k_4})$). Moreover, it also allows for enlargement or reduction coupled with reflection ($r_k < 0$) in the x-axis, that is, $-2 < r_k < -1$ for reflected enlargement (e.g., point $(k_3, x^*_{k_3})$) and $-1 < r_k < 0$ for reflected reduction (e.g., point $(k_2, x^*_{k_2})$).

Consequently, in order to mathematically model the behavior of self-oriented perfectionists, condition SOP1 is achieved by generating a new solution according to one of the best solutions, which is selected from the top 10 percent of solutions. Also, Equation (1) satisfies condition SOP2 in the sense that it can be considered as a high personal standard performed to each component of the new solution.

4.3   Other-oriented perfectionism (OOP)

OOP operator generates new solutions based on the best solution obtained so far (the most perfect person) within the population. Similar to previously, the population is sorted at the end of each iteration. So, the existing best solution is the first solution (i.e., $X_1 = \{x_{1,1}, x_{1,2}, ..., x_{1,n}\}$) in the sorted population. By this operator, X_1 is considered as the other-oriented perfectionist. For generating a new solution, OOP operator focuses only on X_1 (condition OOP4, i.e., narcissism) and does not pay attention to other solutions of the population. Also, it does not allow other solutions $(X_2, X_3, ..., X_n)$ to participate directly in the production of new solutions (condition OOP3 , i.e., a lack of trust). The fundamental idea underlying OOP operator is the generation of a new solution around $X_1 = \{x_{1,1}, x_{1,2}, ..., x_{1,n}\}$ (condition OOP1, where X_1 expects X' (new solutions) to be perfect). To model the only remaining condition (OOP2 , i.e., X_1 has high standards for X'), some restrictions are created on the position of X' , and as such the j'th component of a new solution $X' = \{x'_1, x'_2, ..., x'_n\}$ is calculated as follows:

$$x'_j = x_{1.j} + n(0, 2\sigma_j), \quad j = 1, 2, ..., n \tag{2}$$

Where $n(0, 2\sigma_j)$ is a randomly generated number with a normal distribution, mean value of 0 and standard deviation of $2\sigma_j$. In order to establish the value of standard deviations, the average distance from the j'th component of best solution to that of other solutions $X_1 = \{x_{1,1}, x_{1,2}, ..., x_{1,n}\}$, $i = 1, 2, ..., K_{best}$ are calculated, where $K_{best} \in \{2, 3, ..., N\}$. So, using approach, the distance between $x_{1,j}$ and all $x_{i,j}$'s, $i = 1, 2, ..., K_{best}$ are initially calculated, and then the value of parameter σ_j^2 is created as follows:

$$\frac{\sum_{i=2}^{K_{best}} |x_{1,j} - x_{i,j}|}{N - 1} \tag{3}$$

Where $|x_{1,j} - x_{i,j}|$ indicates the distance from j'th component of X_1 (the best solution obtained so far) to that of X_i. The whole process is repeated for each dimension $j = 1, 2, ..., n$ and each time the average distance is calculated using only the single dimension j. An example of how this approach can be used to generate the j'th component (x_j') from the new solution $X' = \{x_1', x_2', ..., x_n'\}$, is presented in Fig. 4. This figure illustrates equation (2), where $P(x_j')$ is the probability that value x_j' will be selected as the j'th component of X'.
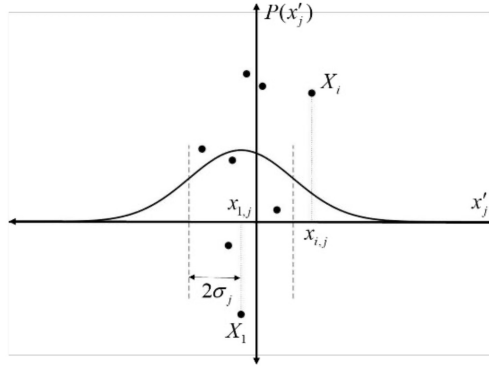
Figure 4: Probability density function $P(x_j')$ for generating j'th component of X'.

The position of X_1 and n–dimensional cube generated by σ_j's $(j = 1, 2, ..., n)$, for $n = 2$ has been shown in Fig. 5. According to equations (2) and (3), the probability that a certain point $X' = \{x_1', x_2', ..., x_n'\}$ is generated as a new solution is equal to:

$$\Pi_{j=1}^n \left\{ \frac{1}{2\sigma_j \sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x_j' - x_{1,j}}{2\sigma_j})^2} \right\} \tag{4}$$

Fig. 5 shows these probabilities for all the points in the plane. The points located in the darker regions are more likely to be selected as the new solutions.

During the entire process of iterations, new solutions which are generated by the OOP operator, explore a good area surrounding the best solution in a random manner. However, using the standard normal distribution presented in equation (2), the search is more biased toward the best solution and less biased towards points which their distances from the best solution are greater than the average distance calculated by equation (3). This guarantees the exploitation step of the algorithm. Summary of how Algorithm 2 (OOP operator) can be expressed is as follows:

Algorithm 2 (OOP operator)
Choose a uniform integer random number $K_{best} \in \{1, 2, ..., N\}$
for $j = 1, 2, ...N$
Calculate σ_j by Equation (3).
Generate a random number x_j' by Equation (2)..
end for

The stochastic character of variable K_{best} can be used as the way to achieve a reasonable compromise between exploration and exploitation steps. In such a way, only the best solutions (K_{best}) are used in equation (3). The lower the value of K_{best}, the higher the convergence speed of the algorithm. In this case, the worse solutions in the population are not considered.

4.4   Socially Prescribed Perfectionism (SPP)

On the contrary to the previous approach, the SPP operator generates a new solution corresponding to the solutions from the population. In this case, a new solution X' is interpreted as a socially prescribed perfectionist who is influenced by all
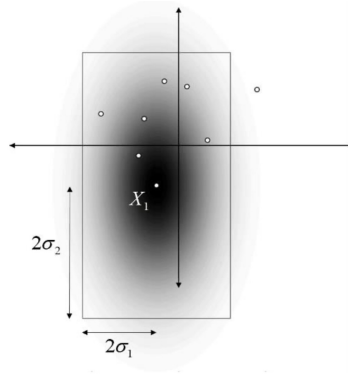
Figure 5: Cube generated by σ_1 and σ_2 ; the probabilities of points which may be generated as a new solution.

the solutions of the population. The main idea behind SPP consists of two steps: (1) all the solutions have the same chance to generate X' (condition SPP1, i.e., X' believes others expect him/her to be perfect); (2) the components of X' are the copies of the corresponding components from the solutions $X_1, X_2, ..., X_N$ (condition SPP2 , i.e., X' desires approval from those around him/her). Therefore, SPP operator is simply stated as follows:

Algorithm 3 (SPP operator)
for $j = 1, 2, ...N$
$\quad$ Randomly select a solution $\bar{X} = \{\bar{x_1}, \bar{x_2}, ..., \bar{x_N}\}$.
$\quad$ Set $x'_j = \bar{x_j}$.
end for

This operator generates the new solutions within an area of minimal cell (n-dimensional cube) including all the previously obtained solutions $(X_1, X_2, ..., X_N)$. So, this operator enhances the diversity of the population and shares information among individuals. As a consequence, it assists the proposed algorithm to search solution space randomly and avoid any trap within the local optima.

4.5   Depression

The fundamental idea underlying the depression step is the formulation of depression and its effects on the self- oriented and socially prescribed perfectionists (see condition Dep1). Depression can be mathematically modelled based on the biased probabilistic choice of SOP and SPP operators. Furthermore, due to the condition Dep2 , no negative effects have been considered for OOP operator in this step. These probabilities are initialized with the initial values of $P_{SOP} = P_{OOP} = P_{SPP} = \frac{1}{3}$. During an arbitrary iteration, if a new solution (X') which is generated by SOP operator fails to meet the set expectation (i.e., X' is worse than the worst solution within the current population), then probability (P_{SOP}) is decreased by a certain value of $\Delta = \frac{1}{3N}$. To make sure that the equation $P_{SOP} + P_{OOP} + P_{SPP} = 1$ remains true for the next generation, the probabilities of P_{SPP} and P_{OOP} are increased to $P_{SPP} + \frac{\Delta}{2}$ and $P_{OOP} + \frac{\Delta}{2}$, respectively. So, for generating new solution, SOP , OOP and SPP may be selected with probabilities of $P_{SOP} - \Delta$, $P_{OOP} + \frac{\Delta}{2}$, and $P_{SPP} + \frac{\Delta}{2}$, respectively. Similarly, for each of the failed new solution (X') generated by SPP operator, P_{SPP} is decreased to $P_{SPP} - \Delta$; P_{SOP} and P_{OOP} are increased by $\frac{\Delta}{2}$. This process will continue until the end of the iteration.

As mentioned earlier, N number of new solutions are produced during each iteration and, therefore, each of the SOP, SPP and OOP operators can be selected at most N times during each iteration. Hence, in an arbitrary iteration, if SOP (or SPP) operator is selected N times with N number of failed new solutions, then the associated probability will be reduced from the initial value $\frac{1}{3}$ to $\frac{1}{3} - N\Delta = 0$ at the end of the iteration. However, at the beginning of the next iteration, we set $P_{SOP} + P_{OOP} + P_{SPP} = \frac{1}{3}$. Using this strategy, the algorithm will have the same opportunity (in each iteration) to search the space using the three operators. Hence, Depression step makes the algorithm more likely to choose the most successful types of operators.

The probability update, however, is optional in the sense that P_{SOP}, P_{OOP}, and P_{SPP} may be considered as the same constant values over the course of iterations. In this case, any of the perfectionism type has the same chance of being selected, i.e., each of the types (SOP, OOP, or SPP) has a probability of 33.3% being selected. On the one hand, this strategy can enable the algorithm to quickly compute approximate solutions, but on the other hand, it causes the algorithm to achieve less accurate solutions. In the second way, the probabilities can be updated during an iteration. Although this method

allows carefully considered selections of SOP, OOP and SPP operators (and hence increases the), but it also decreases the speed of the algorithm. In this paper, we have employed a method to compromise between the accuracy of the algorithm and the speed at which the best solutions can be achieved. To this end, we set $P_{OOP} = \frac{1}{2}$ and $P_{SOP} = P_{SPP} = \frac{1}{4}$ as the fixed probabilities for all iterations.

Finally, Algorithm 4 shows the PSA algorithm for a minimization problem, where $OP \in \{SOP, OOP, SPP\}$ is a variable indicating which operator has been selected for generating a new solution.

Algorithm 4 (Perfectionism Search Algorithm)

Generate the initial population.
Evaluate each solution and sort the solutions in an ascending order of their objective values.
Calculate the best solution X^* by setting $X^* = X_1$.
While ($t \leq$ maximum number of iterations)
    for $i = 1, 2, ...N$
        Select operator $OP \in \{SOP, OOP, SPP\}$ according to probabilities
        $P_{OOP} = \frac{1}{2}$ and $P_{SOP} = P_{SPP} = \frac{1}{4}$.
        if OP = SOP,
            Generate new solution X' by Algorithms 1 and evaluate X'.
        elseif OP = OOP,
            Generate new solution X' by Algorithms 2 and evaluate X'.
        else
            Generate new solution X' by Algorithms 3 and evaluate X'.
        end if
    end for
Generate the new population by keeping
the N best solutions and removing the N worst ones.
Sort the solutions of the resulted population.
Update the best solution X^* by setting $X^* = X_1$, if $f(x_1) \leq f(x^*)$.
$t = t + 1$.
end while

---

5 Theoretical Aspects, Applied Suggestions and Brief Comparative Study

In order to see how PSA solves optimization problems, it's important to note some points:

(A) In PSA, new solutions are generated using the three types of optimization operators. It assists the algorithm to look for an optimal point from different strategies.

(B) In PSA, each search strategy focuses on a specific region; SOP considers neighborhoods around 10% of the best solutions, OOP pays attention to the neighborhood of the best-so-far solution and SPP includes the region between the entire solutions. As such, it creates a balance between exploration and exploitation, and increases the capability of the algorithm to be applied to the optimization of multi-modal problems.

(C) By considering $P_{OOP} = \frac{1}{2}$ and $P_{SOP} = P_{SPP} = \frac{1}{4}$, an appropriate balance is made between the accuracy and speed of the algorithm.

(D) In SOP operator, if $|r|$ is close to 2 ($|r| \approx 2$), the exploration power is increased around the current best so-far solution, which allows for more diversification of the search (higher robustness). Also, if $|r|$ is close to 0 ($|r| \approx 0$), the intensification capability is increased, that leads to a faster convergence (higher convergence speed and higher efficiency).

(E) In OOP, since new solutions are generated from the neighborhoods of 10% of the best solutions, more promising regions are investigated.

(F) SOP and OOP assist PSA to exploit information from the neighborhood of the best solutions that hold valuable knowledge.

(G) In SOP (when $|r| < 1$) and OOP (by lapse of iterations), the space is searched more locally. So, it can be considered as an adaptive learning rate.

(H) The minimal covering cell used by SPP operator assists the algorithm to increase the exploration power. So, the search mechanism in SPP has two advantageous; escaping from local optima and avoiding rapid convergence of the algorithm.

(I) Since in SPP, all the previously obtained solutions contribute towards generation of new solutions, this operation can be considered as an information – transferring tool.

The following points highlight some of the main characteristics of PSA that differentiates it from the other algorithms.

(i) Unlike many of the previously introduced heuristic algorithms (such as ACO, PSO, CA, HS, etc), PSA is a memory-less algorithm that works efficiently and has a good convergence rate.

(ii) One major difference between PSA and most of the other heuristic algorithms is that many heuristic algorithms use some improving directions (e.g., PSO, BBO, GSA, FA, BA, TLBO, BH, ...), while PSA does not apply any particular direction to improve the solutions.

(iii) SOP operator, unlike many optimization operators, does not consider the distance between solutions; however, it works only by changing the differences between the components of the best solution.

6   Experimental Results

In order to evaluate the performance of our algorithm in comparison to other known methods, the typical benchmark test functions have been employed by selecting a comprehensive set of 35 benchmark functions that are typically used in the literature for presenting the performance of different methods and algorithms. List of the test functions, number of dimensions used, initialization interval, optimal solutions and the optimal values are presented in Appendix A. A more detailed description of the test functions may be found in [28,63]. These functions listed in Appendix, Tables A.1-A.3 range from very simple to quite complex, and were selected for their particular characteristics. Functions $f_i(i = 1, 2, ..., 11)$ are unimodal (Table A.1) and $g_i(1 = 1, 2, ..., 13)$ are multimodal (Table A.2). Also, $\widetilde{h_i}(i = 1, 2, ..., 11)$ are the shifted functions which have resulted from some unimodal and multimodal functions h_i so that $\widetilde{h_i}(x) = h_i(x - \mathbf{1})$ where $\mathbf{1} = [1, 1, 1, ..., 1]_{1 \times n}$. The proposed PSA is compared with 11 well-known metaheuristic methods including the most famous algorithms and also some of the recent popular ones, namely, Ant Colony Optimization (ACO) [6,48], Particle Swarm Optimization (PSO) [29], Teaching Learning Based Optimization (TLBO) [40], Biogeography Based Optimization (BBO) [44], Differential Evolution (DE) [53], Harmony Search (HS) [16], Firefly Algorithm (FA) [61], Whale Optimization Algorithm (WOA) [35], Bees Algorithm (BA) [38], Gravitational Search Algorithm (GSA) [41] and Black Hole (BH) algorithm [22]. In order to ensure a fair comparison, we carefully used the parameters suggested by the authors for each algorithm.

In all cases, population size (N) is set to $N = 50$. 30-dimensional versions of the functions ($n = 30$) were used and 30 independent runs were performed in all the experiments. In addition, two types of comparison were made. For the first type, maximum iteration is considered as a stopping condition and is set to 1000 . For each test problem, the following performance indices (averaged over 30 runs) are reported: the best optimal value (Table 1), the average best-so-far solutions (Avg), median of the best solutions (Mdn) and standard deviation (SD) for the last iterations (Tables 2 − 4). For the second type, a comparison was made based on the number of function evaluations needed to reach the following stopping condition:

$$|f_{best} - f^*| < \epsilon f_{best} + \epsilon_2 \tag{5}$$

Where f_{best} is the value of the best solution found by an algorithm, f^* is the known optimal value for a given test problem, and ϵ_1 and ϵ_2 are respectively the relative and absolute errors [48]. For all the test runs, we used $\epsilon_1 = \epsilon_2 = 10^{-4}$. For each test problem, the mean number of function evaluations (NFE) is supplied for all the algorithms. Moreover, numbers in square brackets indicate the percentage of successful runs (PSR), i.e., average number of cases in which the required accuracy was reached (Tables 2 − 4). In all the tables, shaded boxes have been used to show the best results for a given problem.

	WOA	TLBO	BBO	DE	ACO	PSA
f_1	4.7485e-276	3.7512e-246	1.0295e-10	7.9684e-19	6.6077e-72	0
f_2	0.6667	0.9041	0.3490	0.6667	0.6667	2.2288e-19
f_3	3.6732e-266	9.8671e-245	6.1491e-08	6.9791e-10	1.3055e-09	0
f_4	0.5	0.5	0.5	0.5	0.5	0.5
f_5	26.2707	28.7923	5.2982	18.3086	0.71274	0
f_6	34.2804	1.5839e-243	1.5839e02	0.98238	0.0011337	0
f_7	5.7295e-149	6.2017e-121	2.1409e-05	3.6638e-08	4.8992e-41	3.2188e-245
f_8	3.402e-152	1.3215e-121	2.1443e-06	7.268e-09	5.6872e-42	5.8602e-247
f_9	1.6524e-273	7.6983e-245	4.3522e-09	8706e-15	6.416e-69	0
f_{10}	0	1	0	0	0	0
f_{11}	2.0678e-275	3.0112e-244	3.8436e-10	1.1107e-16	4.3393e-71	0
g_1	8.8818e-16	4.4409e-15	1.1597e-05	1.2372e-08	4.4409e-15	8.8818e-16
g_2	-0.86175	-0.7625	-1	-0.76236	-0.66986	-1
g_3	-1	-0.93625	-0.47778	-0.81731	-0.78575	-1
g_4	0	0	3.6267e-08	2.6645e-15	0	0
g_5	0.2571	1.3853	5.2297e-11	2.4151e-16	0.0895	1.4998e-32
g_6	0	2.9848e-07	4.0758	0	11.707	0
g_7	0.2409	1.7004	4.3376e-10	1.4941e-15	2.3359e-32	1.3498e-32
g_8	9.146e-275	2.3084e-243	0.1475	7.3163e-09	2.795e-51	0
g_9	12.533	47.1373	3.7756e-05	4.8925e-06	4.4409e-16	0
g_{10}	3.618e-134	0.0999	0.5999	0.1999	0.1999	0
g_{11}	2.0415e-146	8.7349e-243	0.0001	1.5972e-11	4.7422e-50	-1.2302e-320
g_{12}	-3.4551	-3.165	-3.5	-1.4239	-0.0657	-3.5
g_{13}	-38.8512	-27.1919	-37.7525	-39.1662	-37.2813	-39.1662
$\widetilde{h}_1$	0.0942	5.9096	4.5125e-07	1.258e-07	3.0705e-16	0
$\widetilde{h}_2$	1.2157e05	1.1110e07	0.0028	1.3957e-09	3.6978e-26	0
$\widetilde{h}_3$	22.2976	50.3096	5.0246e-08	6.8945e-14	0	0
$\widetilde{h}_4$	0.1101	1.7681e02	4.6465e-17	6.5906e-30	5.4694e-63	0
$\widetilde{h}_5$	1.4442e-05	6.9221	2.889e-20	4.3705e-14	2.2613e-88	1.6005e-94
$\widetilde{h}_6$	1.4569e02	3.6972e03	0.0004	1.6213e-13	5.1465e-30	0
$\widetilde{h}_7$	26.0988	29.9126	13.9294	10.6000e01	19.8992	0
$\widetilde{h}_8$	6.9944e-05	5.4866	4.3223e-14	8.8788e-32	2.5627e-144	0
$\widetilde{h}_9$	5.6469	33.3308	2.0375	0.128	0.0023	0
$\widetilde{h}_{10}$	0.3005	0.5197	0.2422	0.4975	0.06738	0
$\widetilde{h}_{11}$	3.7697e02	19.4955	10.8001	0.1280	1.0532e-06	0

Table 1: THE BEST OPTIMAL VALUES OF THE TEST PROBLEMS FOUND BY WOA, TLBO, BBO, DE, ACO AND PSA

	FA	BA	PSO	HS	BH	GSA
f_1	2.3386	2.2448e-23	2.6202e-16	0.0112	1.6336	4.4408e02
f_2	26.7541	0.6667	0.6668	6.4017	0.6667	0.6667
f_3	1.4256e02	0.0012	8.8712e-07	2.1427	6.5624e-05	2.6722e-06
f_4	3.1697	2.353	0.5	0.5000	0.5	0.5
f_5	18.1205e01	19.6798	25.8392	24.7479e01	26.766	25.1698
f_6	5.4192	17.4221	91.5358e02	22.0260e02	86.5808	28.7561
f_7	5.8056	16.0956	1.9953e-07	10.3322	4.8536e-06	1.0754e-08
f_8	5.2023	2.9572	1.0309e-08	0.79269	3.3232e-08	1.0185e-08
f_9	1.5422	1.9182e-20	2.2316e-15	9.0222	5.7282e-09	5.1986e-18
f_{10}	0	25	0	10	0	0
f_{11}	19.8964	1.0639e-08	2.4254e-16	1.1945	1.0761e-09	2.9106e-17
g_1	18.9259	10.5707	0.0045	1.1693	1.5185e-05	1.8428e-09
g_2	-0.7042	-1	-0.97548	-0.98858	-0.99636	-1
g_3	-0.6195	-0.15277	-0.17987	-0.78568	-0.6195	-0.47773
g_4	1.4024e02	2.3315e-15	2.2502e-06	1.0714	1.0145e-06	1.1449
g_5	18.1618	22.9755	6.9424e-12	0.0437	6.1388e-10	2.0405e-18
g_6	10.6185	10.0504	10.3806	5.0166	9.0053	7.7226
g_7	0.2776	0.0141	4.594	0.6096	9.0718e-08	5.8712e-19
g_8	13.6073e02	44.2623e02	51.2685e01	10.2741e01	43.7389e01	20.9931
g_9	36.5972	3.5685	10.0071	10.821	1.1311e-05	1.18e-07
g_{10}	0.20869	0.59987	2.4009	0.96516	0.49987	0.90598
g_{11}	32.4466	24.9864e04	0.0002	36.1400e01	5.4209e-05	9.3929e-17
g_{12}	-1.1778	-3.4994	-0.051724	-0.64447	-3.4765	-3.5
g_{13}	-34.5706	-37.2811	-35.3964	-39.155	-36.7842	-36.3388
$\widetilde{h_1}$	11.8138	2.3454	0.0008	0.1004	0.0003	1.1154e-09
$\widetilde{h_2}$	1.5635e06	3.6084	7.7902e-08	6.9181e06	0.0618	0.3074
$\widetilde{h_3}$	20.4495	11.5618	2.8914e-07	19.2699	5.544e-07	0
$\widetilde{h_4}$	2.738	2.8102e-31	1.11e-34	1.0473e02	8.234e-15	2.4956e-35
$\widetilde{h_5}$	0.0555	4.3294e-42	1.4731e-08	4.2817e-09	0.3111	5.5874e-54
$\widetilde{h_6}$	1.7006e04	2.5809e04	4.6137e-14	3.5254e03	24.696	73.5399
$\widetilde{h_7}$	14.8043e01	10.3606e01	31.8505	10.0787	24.2123	11.9395
$\widetilde{h_8}$	0.0008	7.3501e-08	1.3389e-16	8.7973e-11	0.0490	1.0409e-15
$\widetilde{h_9}$	1.4600e02	0.5661	4.5534	3.1674	3.2997	0.4865
$\widetilde{h_{10}}$	0.5458	0.4065	0.1382	0.0174	0.2365	0.6184
$\widetilde{h_{11}}$	1.7421	2.4295e-09	2.2660e02	37.1223	0.0412	52.5817

Table 2: The best optimal values of the test problems found by FA, BA, PSO, HS, BH and GSA

Unimodal functions. Table 1 include the best values,$f^*(x)$, found by the algorithms over 30 runs. These tables demon-

strate the impressive ability of the proposed algorithm to detect the optimal solutions from the test problems. PSA is a clear winner for $f_1 - f_3$, $f_5 - f_9$ and f_{11}. For f_4 and f_{10} almost all the algorithms found the global optimum.

The good convergence rate of PSA can be observed from Table 2. According to these Tables, PSA produced the best results in terms of Avg, Mdn and SD for all the test problems, with the only exception of f_2 where ACO produced a better standard deviation. For $f_1 - f_3$, $f_5 - f_9$ and f_{11}, PSA is the only algorithm that obtained the best Avg and Mdn. For f_4, almost all the algorithms performed similarly well and for f_{10}, some algorithms attained the best results. Based on these results, PSA tends to find the global optimum much faster in comparison to other algorithms and hence has a higher convergence rate. Furthermore, it can produce high quality solutions with low average error (good accuracy) and low standard deviation (good stability) for all the unimodal functions.

Moreover, using PSA the required accuracy is reached in every run (i.e., $PSR = 100\%$) for each unimodal function; and in all cases it has the minimum number of function evaluations. Also, it is to be noted that for f_2 and f_5, PSA is the only algorithm that reached this accuracy.

		WOA	TLBO	BBO	DE	ACO	PSA
f_1	Avg	1.4377e-257	3.7929e-242	8.0804e-09	5.4333e-17	7.7876e-69	0
	Mdn	1.1616e-266	6.7492e-244	1.4835e-09	2.9971e-17	1.7912e-69	0
	SD	0	0	1.6572e-08	5.5677e-17	1.584e-68	0
	NFE [PSR]	12,312.3 [**100%**]	2,601.3 [**100%**]	22,622.6 [**100%**]	15,740.7 [**100%**]	4,954.9 [**100%**]—	600.6 [**100%**]
f_2	Avg	0.6667	0.9550	1.209	0.7605	0.6667	3.6769e-13
	Mdn	0.6667	0.9605	0.6903	0.6667	0.6667	9.9902e-17
	SD	2.7959e-05	0.0210	0.9522	0.4956	1.4189e-16	1.799e-12
	NFE [PSR]	50,050 [0%]	100,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	11,186.2 [**100%**]
f_3	Avg	6.3149e-27	4.5583e-240	0.0001	9.2893e-07	5.3943e-09	0
	Mdn	6.6672e-188	6.4553e-242	8.74e-05	9.9539e-08	4.8139e-09	0
	SD	3.3934e-26	0 3,551.8 [**100%**]	9.1247e-05	1.8022e-06	3.0165e-09	0
	NFE [PSR]	13,963.9 [**100%**]		49,949.9 [53.3%]	22,022 [**100%**]	6,631.6 [**100%**]	1,051 [**100%**]
f_4	Avg	0.5	0.5	0.5	0.5	0.5	0.5
	Mdn	0.5	0.5	0.5	0.5	0.5	0.5
	SD	3.3307e-17	0	8.2834e-17	2.7481e-10	5.2336e-17	0
	NFE [PSR]	11,186.2 [**100%**]	1,400.7 [**100%**]	5,280.3 [**100%**]	11,436.4 [**100%**]	2,952.9 [**100%**]	325.3 [**100%**]
f_5	Avg	27.3042	28.8927	64.541	37.4842	48.8164	**6.6429e-28**
	Mdn	27.073	28.8999	77.3741	25.8895	22.0362	0
	SD	0.7647	0.0433	33.8184	23.7393	52.6671	**3.1271e-27**
	NFE [PSR]	50,050 [0%]	100,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	**4,254.2 [100%]**
f_6	Avg	5.5902e03	2.7584e-239	7.4737e02	9.8156	0.0186	3.3907e-294
	Mdn	4.6003e03	1.0467e-240	6.4937e02	6.1309	0.0132	1.6304e-321
	SD	4.7965e03	0	3.2609e02	9.9271	0.0143	0
	NFE [PSR]	50,050 [0%]	3,801.9 [**100%**]	50,050 [0%]	50,050 [0%]	50,050 [0%]	1,876.9 [**100%**]
f_7	Avg	3.1658e-142	6.6069e-119	0.0002	1.768e-07	1.4602e-39	8.52e-225
	Mdn	2.0906e-144	1.2272e-119	0.0001	1.4899e-07	6.7007e-40	1.5417e-233
	SD	1.2868e-141	1.1014e-118	0.0002	1.0872e-07	1.6051e-39	0
	NFE [PSR]	14,639.6 [**100%**]	6,003 [**100%**]	47,072 [36.7%]	35,435.4 [**100%**]	8,358.3 [**100%**]	1,351.3 [**100%**]
f_8	Avg	2.1075e-141	1.1094e-119	1.6921e-05	4.5861e-08	2.4516e-40	3.9638e-225
	Mdn	7.5248e-147	1.7977e-120	1.2936e-05	4.2129e-08	1.4192e-40	2.4596e-238
	SD	1.1259e-140	2.34e-119	1.4481e-05	2.4938e-08	2.5119e-40	0
	NFE [PSR]	14,664.6 [**100%**]	5,252.6 [**100%**]	35,160.1 [**100%**]	31,406.4 [**100%**]	7,757.7 [**100%**]	1,226.2 [**100%**]
f_9	Avg	1.012e-255	7.596e-240	2.4358e-07	2.2608e-14	1.398e-66	0
	Mdn	2.4036e-262	1.039e-241	8.8815e-08	1.2951e-14	3.0118e-67	0
	SD	0	0 3,751.9 [**100%**]	3.5563e-07	2.1513e-14	3.0587e-66	0
	NFE [PSR]	14,589.6 [**100%**]		24,699.7 [**100%**]	22,722.7 [**100%**]	6,281.3 [**100%**]	825.8 [**100%**]
f_{10}	Avg	0	4.7667	0	0.1333	1.3667	0
	Mdn	0	5	0	0	0	0
	SD	0	1.3337	0	0.3399	2.7384	0
	NFE [PSR]	10,310.3 [**100%**]	100,050 [0%]	7,432.4 [**100%**]	14,514.5 [86.7%]	26,826.8 [56.7%]	350.3 [**100%**]
f_{11}	Avg	3.4449e-255	4.3177e-240	7.9675e-08	2.2508e-15	3.0985e-67	0
	Mdn	2.0673e-264	3.7965e-242	1.5504e-08	2.4628e-15	1.4344e-67	0
	SD	0	0	1.3562e-07	1.6665e-15	6.0618e-67	0
	NFE [PSR]	13,413.4 [**100%**]	3,451.7 [**100%**]	21,196.2 [**100%**]	21,821.8 [**100%**]	5,580.6 [**100%**]	850.8 [**100%**]

Table 3: Minimization result of benchmark functions from Table 9 & 10 (Unimodal test functions) found by WOA, TLBO, BBO, DE, ACO and PSA, where $n = 30$ and maximum number of iterations are 1000

		FA	BA	PSO	HS	BH	GSA
f_1	Avg	3.1585	31.4751	0.0096	0.0158	2.4951	2.8493e06
	Mdn	3.2604	0.1947	2.3295e-12	0.0158	2.532	1.4669e04
	SD	0.3448	49.1488	0.0519	0.0021 50,050 [0%]	0.3481	83.6615e05
	NFE [PSR]	2,200,356 [0%]	1,275,050 [40%]	24,374.3 [96.7%]		50,050 [0%]	50,050 [0%]
f_2	Avg	45.7822	15.8948e03	29.2921e04	10.0314	0.9917	0.6755
	Mdn	41.6323	9.7789	68.9928e03	9.3557	0.6708	0.6667
	SD	15.1427	38.9877e03	46.7798e04	2.9611	0.6884	0.0300
	NFE [PSR]	2,170,322.5 [0%]	1,275,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]
f_3	Avg	1.8942e02	6.9592e02	60.9412e02	2.7806	0.0054	7.272e-05
	Mdn	1.8805e02	49.8555	19.8971	2.8373	0.0028	6.5751e-05
	SD	21.7892	12.9251e02	13.2010e03	0.2990	0.0073	3.4463e-05
	NFE [PSR]	2,199,033.5 [0%]	1,275,050 [0%]	50,050 [3.3%]	50,050 [0%]	50,050 [3.3%]	37,412.4 [76.7%]
f_4	Avg	4.2587	2.353	27.4722	**0.5000**	0.5	0.5
	Mdn	4.314	0.5	5.336	**0.5000**	0.5	0.5
	SD	0.5768	8.1566	39.4775	5.1406e-06	1.2651e-12	0
	NFE [PSR]	2,291,196.5 [0%]	887,434.8 [73.3%]	50,050 [6.7%]	31,281.2 **[100%]**	6,431.4 **[100%]**	12,612.6 **[100%]**
f_5	Avg	5.2905e02	8.9811e06	82.5440e06	5.6377e02	1.0514e02	28.8558
	Mdn	2.5318e02	35.1230e02	35.7361e06	4.8958e02	78.0888	25.7126
	SD	5.4441e02	20.5514e06	84.4992e06 50,050 [0%]	3.8020e02	1.0116e02	13.6662
	NFE [PSR]	2,008,748 [0%]	1,275,050 [0%]		50,050 [0%]	50,050 [0%]	50,050 [0%]
f_6	Avg	10.3257	94.9561e02	30.1030e03	61.0023e02	1.6389e02	1.0234e02
	Mdn	8.7419	44.6956e02	31.4372e03	57.4462e02	1.6942e02	1.0104e02
	SD	5.0396	10.4469e03	11.8775e03	18.8530e02	45.5732	35.8842
	NFE [PSR]	1,339,726.5 [0%]	1,275,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]
f_7	Avg	9.2836	1.6889e02	0.0403	12.3562	0.0002	0.4660
	Mdn	7.0446	54.4746	0.0002	12.4165	8.9254e-05	1.5737e-08
	SD	4.939	2.1129e02	0.1187	0.9441	0.0004	1.8153
	NFE [PSR]	1,349,762.5 [0%]	1,275,050 [0%]	50,050 [36.7%]	50,050 [0%]	50,050 [53.3%]	31,331.3 [80%]
f_8	Avg	5.9462	18.6205	0.0319	1.188	2.3094e-06	0.0400
	Mdn	5.9285	9.8503	6.4271e-05	1.1897	4.517e-07	1.5185e-08
	SD	0.3582	24.1327	0.1476	0.1239	5.7193e-06	0.1834
	NFE [PSR]	2,128,931.5 [0%]	1,275,050 [0%]	36,211.2 [60%]	50,050 [0%]	33,483.4 **[100%]**	31,506.8 [93.3%]
f_9	Avg	1.9569	43.1083e02	2.3678e02	13.1538	1.3031e-07	1.0826e-17
	Mdn	1.9741	1.0844e-09	0.0145	12.9999	7.4392e-08	9.6156e-18
	SD	0.1910	86.7370e02	7.2513e02	1.8022	1.4585e-07	4.4511e-18
	NFE [PSR]	1,718,742 [0%]	314,937.3 [56.7%]	39,514.5 [40%]	50,050 [0%]	32,157.1 **[100%]**	16,942 **[100%]**
f_{10}	Avg	1.3667	2.2032e03	17.1257e02	14.1667	0.066667	0
	Mdn	1	4.985e02	9	14	0	0
	SD	0.54671	3.7426e03	6.0310e03	1.9336	0.24944	0
	NFE [PSR]	1,731,419.5 [3.3%]	1,275,050 [0%]	50,050 [6.7%]	50,050 [0%]	28,153.1 [93.3%]	6,206.2 **[100%]**
f_{11}	Avg	26.7122	7.7361e02	5.1003e-05	1.777	2.787e-07	9.0687e-17
	Mdn	26.2813	73.2921	1.6173e-08	1.8128	4.8899e-08	8.3128e-17
	SD	3.0663	12.0210e02	0.0002	0.24023	6.6026e-07	3.0573e-17
	NFE [PSR]	2,179,059.5 [0%]	1,033,428 [20%]	41,241.2 [96.7%]	50,050 [0%]	34,334.3 **[100%]**	21,121.1 **[100%]**

Table 4: Minimization result of benchmark functions from Table 9 & 10 (Unimodal test functions) found by FA, BA, PSO, HS, BH and GSA, where $n = 30$ and maximum number of iterations is 1000

Multimodal functions. As Table 1 & 2 illustrate, PSA achieves the best values in all the multimodal test problems. In each case, there are few other algorithms whose results are similar to the results found by PSA. Exceptions are g_5, g_7 , g_8 and g_{11} where the performance of PSA is strictly better than that of all other algorithms. The reported Avg, Mdn and SD results for multimodal functions have been summarized in Table 5 & 6.

		WOA	TLBO	BBO	DE	ACO	PSA
g_1	Avg	3.9672e-15	4.4409e-15	7.5645e-05	3.7728e-08	0.0310	1.125e-15
	Mdn	4.4409e-15	4.4409e-15	6.0913e-05	3.4195e-08	7.9936e-15	8.8818e-16
	SD	2.5509e-15	0	5.5484e-05	1.5908e-08	0.1672	1.2755e-15
	NFE [PSR]	14,990 [**100%**]	4,902.4 [**100%**]	44,769.7 [80%]	31,281.2 [**100%**]	8,358.3 [96.7%]	1,226.2 [**100%**]
g_2	Avg	-0.7622	-0.6860	-1	-0.6289	-0.6068	-1
	Mdn	-0.7766	-0.6881	-1	-0.6105	-0.601	-1
	SD	0.0559	0.0338	4.2973e-10	0.0500	0.0275	0
	NFE [PSR]	50,050 [0%]	100,050 [0%]	9,959.9 [**100%**]	50,050 [0%]	50,050 [0%]	1,126.1 [**100%**]
g_3	Avg	-0.9297	-0.3311	-0.3311	-0.7871	-0.6568	-0.9936
	Mdn	-0.9362	-0.9362	-0.2888	-0.7857	-0.6195	-1
	SD	0.0526	4.9521e-17	0.0838	0.0058	0.1099	0.0191
	NFE [PSR]	50,050 [13.3%]	100,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	1,651.6 [90%]
g_4	Avg	0.0020	0	0.0102	0.0025	0.0067	0
	Mdn	0	0	0.0086	7.7549e-14	0	0
	SD	0.0110	0	0.0100	0.0046	0.0128	0
	NFE [PSR]	13,588.6 [96.7%]	4,152.1 [**100%**]	41,016 [30%]	25,300.3 [76.7%]	6,831.8 [66.7%]	775.8 [**100%**]
g_5	Avg	0.7933	2.091	0.4483	0.1445	1.2708	1.4998e-32
	Mdn	0.7471	2.1168	1.2967e-09	1.5596e-15	0.9534	1.4998e-32
	SD	0.3219	0.3355	0.6559	0.2633	0.9554	1.0948e-47
	NFE [PSR]	50,050 [0%]	100,050 [0%]	32,532.5 [53.3%]	35,660.6 [63.3%]	50,050 [0%]	850.8 [**100%**]
g_6	Avg	2.0075	2.3438	6.3019	4.6805	12.3604	0
	Mdn	0	3.3723	6.4489	4.8145	12.3783	0
	SD	3.9385	2.0551	0.8323	1.2038	0.2587	0
	NFE [PSR]	22,998 [73.3%]	100,050 [16.7%]	50,050 [0%]	50,050 [3.3%]	50,050 [0%]	1,001 [**100%**]
g_7	Avg	1.0839	2.6068	8.7806e-08	0.1208	0.2464	**1.3498e-32**
	Mdn	1.1711	2.6507	6.5568e-09	1.526e-12	2.4591e-31	**1.3498e-32**
	SD	0.5197	0.3785	2.9285e-07	0.6456	0.7697	5.4738e-48
	NFE [PSR]	50,050 [0%]	100,050 [0%]	22,397.4 [**100%**]	26,952 [86.7%]	28,528.5 [63.3%]	650.6 [**100%**]
g_8	Avg	3.612e-258	2.4971e-239	3.9828e02	11.6561	5.1257e02	0
	Mdn	9.5972e-264	1.0067e-240	3.4211e02	2.7311e-07	4.8337e02	0
	SD	0	0	2.9528e02	33.188	4.3578e02	0
	NFE [PSR]	13,363.3 [**100%**]	4,502.2 [**100%**]	50.050 [0%]	37,837.8 [86.7%]	50,050 [16.7%]	950.9 [**100%**]
g_9	Avg	28.3982	66.12	0.0027	0.0002	9.3259e-15	0
	Mdn	27.9624	66.6546	0.0016	3.3341e-05	6.8834e-15	0
	SD	9.0584	9.8457	0.0038	0.0005	7.6111e-15	0
	NFE [PSR]	50,050 [0%]	100,050 [0%]	50,050 [6.7%]	48,898.8 [70%]	10,335.3 [**100%**]	1,301.3 [**100%**]
g_{10}	Avg	0.1199	0.0999	0.8467	0.2247	0.3832	**0.0333**
	Mdn	0.0999	0.0999	0.7999	0.2001	0.2999	**0**
	SD	0.0748	0	0.1686	0.0406	0.2130	0.0649
	NFE [PSR]	50,050 [16.7%]	100,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	**26,051 [73.3%]**
g_{11}	Avg	4.3827e-78	9.5776e-238	0.0010	2.3925e-10	1.7177e-46	**-5.1284e-321**
	Mdn	4.0478e-99	1.2101e-239	0.0008	1.4233e-10	1.4223e-47	**-4.6689e-321**
	SD	2.3575e-77	0	0.0006	3.523e-10	6.4331e-46	**0**
	NFE [PSR]	21,221.2 [**100%**]	4,402.2 [**100%**]	50,050 [0%]	30,305.3 [**100%**]	9,584.6 [**100%**]	1,076.1 [**100%**]
g_{12}	Avg	-0.4961	-2.8297	-2.6667	-0.7986	-0.0087	-3.3336
	Mdn	-0.0189	-2.8175	-3.5	-0.6578	-0.0059	-3.5
	SD	1.0452	0.17595	1.1785	0.3233	0.0111	0.6225
	NFE [PSR]	50,050 [0%]	100,050 [0%]	37,387.3 [66.7%]	50,050 [0%]	50,050 [0%]	1,526.5 [93.3%]
g_{13}	Avg	-35.0577	-25.4244	-35.9776	-38.1295	-35.5849	**-39.1662**
	Mdn	-35.1778	-25.35	-36.1032	-38.2237	-35.632	**-39.1662**
	SD	2.3959	1.1513	1.2025	0.54951	0.9929	**0**
	NFE [PSR]	50,050 [0%]	100,050 [0%]	50,050 [0%]	50,050 [6.7%]	50,050 [0%]	**500.5 [100%]**

Table 5: Minimization result of benchmark functions from Table 10 & 11 (Multimodal test functions) found by WOA, TLBO, BBO, DE, ACO and PSA, where $n = 30$and maximum number of iterations is 1000

		FA	BA	PSO	HS	BH	GSA
g_1	Avg	19.6837	17.8574	12.4815	1.7832	0.01024	2.3768e-09
	Mdn	19.7007	18.5735	13.509	1.8374	0.0018	2.3432e-09
	SD	0.2346	1.9865	5.5947	0.18052	0.0214	3.0133e-10
	NFE [PSR]	2,289,964.5 [0%]	1,275,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [13.3%]	27,552.5 **[100%]**
g_2	Avg	-0.6624	-0.9919	-0.8293	-0.9844	-0.9855	-0.9989
	Mdn	-0.6630	-0.9998	-0.8807	-0.9846	-0.9886	-1
	SD	0.01560	0.0253	0.1126	0.0021	0.0085	0.0060
	NFE [PSR]	2,367,082 [0%]	867,034 [53.3%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	19,744.7 [96.7%]
g_3	Avg	-0.6094	-0.0536	-0.0536	-0.6674	-0.5392	-0.2735
	Mdn	-0.6193 0.0352	-0.0481	-0.0431	-0.6195	-0.4778	-0.2882
	SD		0.0250	0.0321	0.0739	0.0702	0.1233
	NFE [PSR]	2,235,940 [0%]	1,275,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]
g_4	Avg	2.2736e02	30.6158	9.9293	1.1191	0.0180	2.1985
	Mdn	2.3281e02	4.8173	0.2009	1.1217	0.0099	1.9595
	SD	39.963	58.4573	50.3343	0.0178	0.0198	1.0096
	NFE [PSR]	1,277,054 [0%]	1,275,050 [16.7%]	50,050 [3.3%]	50,050 [0%]	50,050 [20%]	50,050 [0%]
g_5	Avg	36.89	34.0034	33.9449	0.0680	0.0060	0.0936
	Mdn	36.946	32.4314	23.3921	0.0696	5.3948e-08	4.7577e-18
	SD	8.8642	10.5226	25.8672	0.0111	0.0223	0.1895
	NFE [PSR]	2,277,246 [0%]	1,275,050 [0%]	50,050 [3.3%]	50,050 [0%]	34,159.1 [90%]	16,416.4 [73.3%]
g_6	Avg	11.615	10.9257	12.1838	6.596	10.551	9.5538
	Mdn	11.6374	10.9854	12.3338	6.6573	10.5379	9.2904
	SD	0.3786	0.3340	0.6422	0.4748	0.5729	1.0491
	NFE [PSR]	2,262,254 [0%]	1,275,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]
g_7	Avg	0.3273	11.2764e06	51.1919e07	0.8414	0.0008	0.0015
	Mdn	0.3317	61.1277	54.2408e07	0.7970	4.097e-06	1.1127e-18
	SD	0.0241	27.5248e06	35.0222e07	0.1511	0.0027	0.0037
	NFE [PSR]	1,891,140 [0%]	1,275,050 [0%]	50,050 [0%]	50,050 [0%]	45,195.15 [83.3%]	32,432.4 [86.7%]
g_8	Avg	2.1681e03	6.4985e03	4.2626e03	1.5742e02	1.8685e03	2.3091e02
	Mdn	20.2030e02	64.0538e02	39.7341e02	1.4514e02	19.3556e02	2.0661e02
	SD	6.2277e02	10.4159e02	17.5046e02	42.646	10.0055e02	1.5733e02
	NFE [PSR]	2,002,219 [0%]	1,275,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]
g_9	Avg	56.3622	29.5766e02	85.7735e03	14.8272	0.0072	4.4172
	Mdn	55.6749	18.3567	68.7687e03	14.6629	0.0005	2.1003
	SD	10.7762	97.9610e02	77.8947e03	1.8852	0.0156	4.9525
	NFE [PSR]	2,179,551 [0%]	1,275,050 [0%]	50,050 [0%]	50,050 [0%]	49,899.8 [16.7%]	50,050 [33.3%]
g_{10}	Avg	0.3873	7.3632	7.33	1.2166	0.6899	1.586
	Mdn	0.2999	6.6499	6.3353	1.2168	0.6999	1.6893
	SD	0.1498	4.7592	2.8086	0.1583	0.1012	0.3420
	NFE [PSR]	1,293,717.5 [0%]	1,275,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]
g_{11}	Avg	41.7632	59.2780e04	18.1836e02	5.5315e02	0.0008	2.4097e-16
	Mdn	41.6016	58.1649e04	2.1673e02	5.2091e02	0.0004	2.4479e-16
	SD	4.5022	11.3170e04	51.0800e02	1.2103e02	0.0009	6.1442e-17
	NFE [PSR]	1,651,058 [0%]	1,275,050 [0%]	50,050 [0%]	50,050 [0%]	48,823.8 [3.3%]	21,221.2 **[100%]**
g_{12}	Avg	-0.9480	-3.2293	-0.0077	-0.4392	-2.0039	-3.0833
	Mdn	-0.9452	-3.2937	-0.0060	-0.4262	-2.4114	-3.5
	SD	0.0835	0.2281	0.0086	0.0974	1.1432	0.9317
	NFE [PSR]	2,277,505.5 [0%]	1,275,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	34,434.4 [83.3%]
g_{13}	Avg	-33.0316	-33.6979	-18.7046	-39.1494	-33.7709	-33.9827
	Mdn	-32.9629	-34.9251	-17.2052	-39.149	-33.9049	-33.9827
	SD	0.8953	3.1559	5.4533	0.0029	1.2115	1.0466
	NFE [PSR]	2,226,590 [0%]	1,275,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]

Table 6: Minimization result of benchmark functions from Table 10 (Multimodal test functions) found by FA, BA, PSO, HS, BH and GSA, where $n = 30$ and maximum number of iterations is 1000

By comparing these obtained results with the best-so-far values of average, median and standard deviation of PSA, we can see that PSA provides the best results except for the standard deviations related to g_1 , g_3 and g_{10} (compared with TLBO) and g_{12} (compared with PSO). For other results, there are few other algorithms (in comparison with the unimodal functions) with the same performance as the PSA. Also, for g_5 , g_7 , g_9 and g_{13} , PSA provides strictly better results than all the algorithms in all the criteria Avg, Mdn and SD.

Table 3 also include the number of function evaluation (NFE) and the percentage of successful runs (PSR) for the algorithms. Based on these tables, PSA has the maximum successful rates and uses the minimum number of evaluation to reach the required accuracy. Therefore, the performance of PSA is shown to be relatively good, as it achieves the best PSR ($=100$) in 10 out of 13 test problems, $PSR = 90\%$ for g_3 , $PSR = 73.3\%$ for g_{10} and $PSR = 93.3\%$ for g_{12}.

These results prove the good convergence rate, accuracy and stability of PSA in handling the multimodal functions.

Shifted functions. As mentioned previously, functions $\widetilde{h}_i$ have resulted from basic test function h_i, listed in table A.5 , A.6, by shifting their global optimums. More precisely, $\widetilde{h}_i(x) = h_i(x - \mathbf{1}), i = 1, 2, ..., 11$, where $\mathbf{1}$ is an n-dimensional vector with each component equal to one. Therefore, the optimal values will remain unchanged. In other words, if x^* denotes the optimum of $\widetilde{h}_i$ then $x^* - \mathbf{1}$ is the optimum of h_1, and $\widetilde{h}_i(x) = h_i(x - \mathbf{1})$. Table 7 and 8 present the basic benchmark function $\widetilde{h}_i, i = 1, 2, ..., 11$. Some of the test functions are unimodal and others are multimodal functions. Comparison of the results of best values found by PSA and those obtained using other algorithms show that PSA is the only best performing algorithm for all the shifted test problems; results of which are presented in Table 1 and 2. In this respect only for $\widetilde{h}_3$, ACO, and GSA perform similarly well.

On the shifted test functions (especially, multimodal ones with a large number of local optima), finding good solutions and escaping from local optima is very hard. However, by comparing the indices Avg, Mdn and SD which are summarized in Table 4, it can be seen that PSA exhibits significant performance, and provides much better results in comparison to other algorithms on these functions. Although, the only exceptions are Avg and SD for $\widetilde{h_1}$ obtained by ACO, Avg and SD for $\widetilde{h_3}$ obtained by BBO, SD for $\widetilde{h_1}$ obtained by TLBO and SD for $\widetilde{h_{10}}$ obtained by FA.

Furthermore, PSA achieved the best PSR (=100) for 8 out of 11 test problems. Also, it attained PSR=96.7% for $\widetilde{h_3}$, PSR=53.3% for $\widetilde{h_1}$ and PSR = 63.3% for $\widetilde{h_{10}}$. Additionally, for $\widetilde{h_1}$, $\widetilde{h_9}$, and $\widetilde{h_{10}}$, PSA is the only algorithm achieving the required accuracy. All other algorithms in this category require many more function evaluations in order to reach the required accuracy. Finally, like the unimodal functions (f_i) and multimodal functions (g_i), PSA used the minimum number of function evaluation for solving shifted functions $\widetilde{h_i}$ in all the test problems. To sum up, PSA is capable of handling these types of problems very effectively.

		WOA	TLBO	BBO	DE	ACO	PSA
$\widetilde{h_1}$	Avg	5.2305	8.8366	0.0014	0.0012	2.3997e-15	5.3308e-15
	Mdn	4.7087	8.4279	0.0002	4.3305e-05	1.4764e-15	1.5613e-16
	SD	2.8235	1.5058	0.0032	0.0027	2.3495e-15	1.4228e-14
	NFE [PSR]	50,050 [0%]	100,050 [0%]	50,050 [40%]	44,869.8 [63.3%]	7,157.1 [**100%**]	925.9 [**100%**]
$\widetilde{h_2}$	Avg	32.9372e05	17.9546e06	0.1605	3.2173e-08	3.858e-25	0
	Mdn	31.5471e05	18.3487e06	0.0625	1.2392e-08	3.0199e-25	0
	SD	20.1215e05	29.8458e05	0.3014	4.4412e-08	3.2807e-25	0
	NFE [PSR]	50,050 [0%]	100,050 [0%]	50,050 [0%]	38,763.7 [**100%**]	10,635.6 [**100%**]	1,051 [**100%**]
$\widetilde{h_3}$	Avg	31.3882	73.6256	2.6121e-06	0.6924	2.2209	1.0092
	Mdn	31.1355	74.6363	6.5957e-07	1.0498	2.0996	0
	SD	6.0552	10.1223	4.3797e-06	0.7700	1.4975	5.4348
	NFE [PSR]	50,050 [0%]	100,050 [0%]	30,205.2 [**100%**]	38,288.2 [46.7%]	28,453.4 [10%]	900.9 [96.7%]
$\widetilde{h_4}$	Avg	25.2863	3.8960e02	1.0074e-12	1.5526e-27	1.3837e-60	0
	Mdn	10.6199	3.7391e02	3.0819e-15	2.092e-28	8.0978e-62	0
	SD	37.5031	1.2054e02	4.4873e-12	3.7287e-27	5.4722e-60	0
	NFE [PSR]	50,050 [0%]	100,050 [0%]	14,764.7 [**100%**]	17,792.8 [**100%**]	5,130.1 [**100%**]	875.9 [**100%**]
$\widetilde{h_5}$	Avg	0.0030	19.0203	1.0772e-17	1.349e-06	6.5441e-08	3.0601e-76
	Mdn	0.0003	19.1642	1.3193e-18	2.2837e-07	3.2606e-09	1.6005e-94
	SD	0.0105	4.8144	2.1481e-17	3.9214e-06	1.3092e-07	1.6479e-75
	NFE [PSR]	50,050 [26.7%]	100,050 [0%]	23,873.8 [**100%**]	22,597.6 [**100%**]	2,602.6 [**100%**]	250.2 [**100%**]
$\widetilde{h_6}$	Avg	1.2360e03	7.7749e03	1.6365	2.887e-12	4.7861e-28	0
	Mdn	7.9364e02	6.9387e03	0.2409	1.9148e-12	2.3802e-28	0
	SD	1.0787e03	3.1597e03	3.5795	3.0327e-12	5.4757e-28	0
	NFE [PSR]	50,050 [0%]	100,050 [0%]	50,050 [0%]	30,530.5 [**100%**]	7,707.7 [**100%**]	1,251.2 [**100%**]
$\widetilde{h_7}$	Avg	28.9636	29.9497	31.3412	16.0096e01	34.3924	19.8992
	Mdn	28.94	29.9506	31.8387	16.4263e01	32.3361	0
	SD	0.7954	0.0205	9.4835	21.8432	10.2233	30.1786
	NFE [PSR]	50,050 [0%]	100,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [53.3%]
$\widetilde{h_8}$	Avg	0.0887	16.4095	5.8166e-09	2.7032e-06	8.2173e-34	4.1087e-34
	Mdn	0.0003	17.0282	3.2843e-10	3.1885e-15	7.5965e-65	0
	SD	0.3732	4.9455	1.1784e-08	1.2511e-05	3.0746e-33	2.2126e-33
	NFE [PSR]	50,050 [13.3%]	100,050 [0%]	19,694.7 [**100%**]	6,031 [**100%**]	2,577.6 [**100%**]	325.3 [**100%**]
$\widetilde{h_9}$	Avg	78.4202	82.3708	13.1452	2.0126	4.0905	**0**
	Mdn	63.2084	71.655	11.8146	1.7747	2.0264	**0**
	SD	56.8427	38.5358	10.3958	2.3634	4.1748	0
	NFE [PSR]	50,050 [0%]	100,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	650.6 [**100%**]
$\widetilde{h_{10}}$	Avg	0.4668	0.5832	0.3545	0.6079	0.4419	**0.0772**
	Mdn	0.4790	0.5808	0.3475	0.6187	0.5494	**0**
	SD	0.0923	0.0361	0.0617	0.0528	0.2511	**0.1198**
	NFE [PSR]	50,050 [0%]	100,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	**25,175.1 [63.3%]**
$\widetilde{h_{11}}$	Avg	4.7887e02	42.2103	31.0847	1.0199	0.0023	**0**
	Mdn	4.5577e02	35.6207	28.4182	0.7077	0.0001	**0**
	SD	81.2657	17.8737	13.8752	0.7690	0.0066	**0**
	NFE [PSR]	50,050 [0%]	100,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [33.3%]	**3,753.7 [100%]**

Table 7: Minimization result of benchmark functions in Table 11 (shifted functions) found by WOA, TLBO, BBO, DE, ACO and PSA, where $n = 30$ and maximum number of iterations is 1000

		FA	BA	PSO	HS	BH	GSA
$\widetilde{h_1}$	Avg	16.5535	7.4923	5.8091	0.0004	0.0626	0.0004
	Mdn	16.7776	7.3576	0.9473	1.3989e-09	0.0269	1.3989e-09
	SD	2.2543	3.8495	14.0008	0.0010	0.1056	0.0010
	NFE [PSR]	2,237,292 [0%]	1,275,050 [0%]	50,050 [0%]	26,301.3 [80%]	50,050 [0%]	26,301.3 [80%]
$\widetilde{h_2}$	Avg	18.8133e05	36.3748e08	13.1364e08	98.8708	4.2282e02	98.8708
	Mdn	19.0087e05	86.7148e02	2.9052	44.1272	1.9881e02	44.1272
	SD	1.5070e05	73.7024e08	66.8277e08	107.579	5.7525e02	1.0758e02
	NFE [PSR]	1,714,698 [0%]	1,275,050 [0%]	50,050 [3.3%]	50,050 [0%]	50,050 [0%]	50,050 [0%]
$\widetilde{h_3}$	Avg	23.0051	29.39376e02	3.7845e02	0.61947	0.3648	0.6195
	Mdn	23.2354	1.0472e02	4.877	0.4414	0.0040	0.4414
	SD	1.3196	62.6571e02	18.2722e02	0.6424	0.4921	0.6424
	NFE [PSR]	1,964,025 [0%]	1,275,050 [0%]	50,050 [3.3%]	35,985.9 [40%]	45,570.5 [30%]	35,985.9 [40%]
$\widetilde{h_4}$	Avg	4.2601	13.5384e06	16.2622e04	6.9258e-35	4.249e-11	6.9258e-35
	Mdn	4.2018	79.5771	1.4027e-07	5.593e-35	8.1923e-13	5.593e-35
	SD	0.7055	49.7340e06	70.6541e04	4.0983e-35	1.922e-10	4.0983e-35
	NFE [PSR]	1,668,421 [0%]	1,275,050 [43.3%]	39,939.9 [60%]	11,486.5 [**100%**]	25,850.8 [**100%**]	11,486.5 [**100%**]
$\widetilde{h_5}$	Avg	0.1107	3.66	1.3524	1.5744e-52	0.3703	1.5744e-52
	Mdn	0.1099	0.6361	0.0442	3.3009e-53	0.3769	3.3009e-53
	SD	0.0273	6.4116	3.1091	4.4847e-52	0.0355	4.4847e-52
	NFE [PSR]	2,266,519 [0%]	1,275,050 [20%]	50,050 [16.7%]	9,959.9 [**100%**]	50,050 [0%]	9,959.9 [**100%**]
$\widetilde{h_6}$	Avg	4.2740e04	4.0173e05	1.1783e-07	9.6724e02	4.6072e02	9.6724e02
	Mdn	3.8753e04	4.5656e04	6.7044e-10	9.0891e02	3.3705e02	9.0891e02
	SD	1.4213e04	1.2319e06	4.6328e-07	5.4377e02	4.1809e02	5.4377e02
	NFE [PSR]	1,288,368 [0%]	1,275,050 [0%]	33,708.7 [**100%**]	50,050 [0%]	50,050 [0%]	50,050 [0%]
$\widetilde{h_7}$	Avg	1.9142e02	1.6481e02	1.6663e02	19.9655	51.9033	19.9655
	Mdn	1.9346e02	1.5773e02	1.4775e02	18.9042	51.8105	18.9042
	SD	19.0159	41.6145	1.0037e02	5.744	11.1337	5.744
	NFE [PSR]	2,281,812.5 [0%]	1,275,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]
$\widetilde{h_8}$	Avg	0.0075	9.1415	0.1180	2.2962e-11	0.0713	2.2962e-11
	Mdn	0.0068	0.0425	1.0283e-06	5.1698e-12	0.0727	5.1698e-12
	SD	0.0035	21.9564	0.5612	6.0402e-11	0.0096	6.0402e-11
	NFE [PSR]	2,221,983.5 [0%]	1,275,050 [36.7%]	43,018 [76.7%]	12,862.8 [**100%**]	50,050 [0%]	12,862.8 [**100%**]
$\widetilde{h_9}$	Avg	2.1263e02	1.0686e02	98.1528	14.1375	11.6503	14.1375
	Mdn	2.1927e02	41.6048	17.3008	13.2904	12.8832	13.2904
	SD	27.6932	1.3891e02	1.9507e02	12.947	7.4511	12.947
	NFE [PSR]	2,293,639.5 [0%]	1,275,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]
$\widetilde{h_{10}}$	Avg	0.6116	0.4698	0.3232	0.7545	0.3128	0.7545
	Mdn	0.6115	0.4690	0.301	0.7691	0.3183	0.7691
	SD	**0.0288**	0.0427	0.1058	0.0541	0.0378	0.0541
	NFE [PSR]	2,335,961.5 [0%]	1,275,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]
$\widetilde{h_{11}}$	Avg	3.0594	12.148	8.15971e02	97.1866	0.2276	97.1866
	Mdn	3.0853	0.0143	7.4653e02	94.3317	0.1820	94.3317
	SD	0.3933	35.8589	3.8495e02	24.1285	0.1451	24.1285
	NFE [PSR]	2,013,768 [0%]	1,275,050 [36.7%]	50,050 [0%]	50,050 [0%]	50,050 [0%]	50,050 [0%]

Table 8: MINIMIZATION RESULT OF BENCHMARK FUNCTIONS IN TABLE 11 (SHIFTED FUNCTIONS) FOUND BY FA, BA, PSO, HS, BH AND GSA, WHERE $n = 30$ AND MAXIMUM NUMBER OF ITERATIONS is 1000

7 Conclusion

In this study, a novel population-based meta-heuristic optimization algorithm called Perfectionism Search Algorithm (PSA), was introduced. PSA was designed based on simulation of the most popular model and psychological aspects of perfectionism, initially proposed by Hewitt and Flett. This research aimed at presenting the idea, theoretical and applied aspects of the algorithm to continuous domains, as well as an implementation that performs well on standard benchmark test problems. In order to evaluate the efficiency of the introduced algorithm, it was applied to a set of various standard benchmark functions by solving 35 mathematical nonlinear optimization problems. These test functions were categorized into three classes of unimodal, multimodal, and shifted functions (which resulted from some unimodal and multimodal basic functions). These benchmark functions were used to make two kinds of comparisons, based on different termination criteria: i) the maximum iteration and ii) the number of function evaluations needed to reach an required accuracy. Also, the performance of PSA was compared against a substantial number of famous algorithms and approaches such as: ACO, PSO, TLBO, BBO, DE, HS, FA, WOA, BA, GSA and BH. The results obtained by PSA provided superior results in global optima achievement in comparison to the other approaches. Also, from the best-so-far results of average, median and standard deviation obtained from the execution of PSA, it was observed that this algorithm has a high stability and capability of finding high quality solutions with fast convergence rate. Moreover, PSA has high percentage of successful runs in satisfying the accuracy requirements and uses the minimum number of function evaluation in all the test problems. The optimization results demonstrated that PSA is an efficient algorithm to deal with complex optimization problems and may, therefore, be considered as a competitive approach when compared to other renowned meta- heuristic algorithms. Finally, further research into the performance of PSA on the constrained optimization problems will be the subject of future publications.

8 Appendix A

Table 9 (unimodal functions), Table 10 (multimodal functions) and Table 11 (shifted functions) where n denotes the dimension of a given function.

Name	Function	Interval
Brown	$f_1(x) = \sum_{i=1}^{n-1}((x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{(x_i^2+1)})$	$[-1,4]^n$
Dixon-Price	$f_2(x) = (x_1 - \mathbf{1})^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2$	$[-10,10]^n$
Powell singular 2	$f_3(x) = \sum_{i=2}^{n-2}(x_{i-1} + 10x_i)^2 + 5(x_{i+1} - x_{i+2})^2 + (x_i - 2x_{i+1})^4 + 10(x_{i-1}x_{i+2})^4$	$[-4,5]^n$
Quartic	$f_4(x) = \sum_{i=1}^{n} ix^4 + Random([0,))$	$[-1.28, 1.28]^n$
Rosenbrock	$f_5(x) = \sum_{i=1}^{n-1}(100(x_{i+1} - x_i^2))^2 + (x_i - 1)^2$	$[-30,30]^n$
Schwefel 1.2	$f_6(x) = \sum_{i=1}^{n}(\sum_{j=1}^{i} x_j)^2$	$[-100,100]^n$
Schwefel 2.20	$f_7(x) = \sum_{i=1}^{n} |x_i|$	$[-100,100]^n$
Schwefel 2.22	$f_8(x) = \sum_{i=1}^{n} |x_i| + \Pi_{i=1}^{n} |x_i|$	$[-10,10]^n$
Sphere	$f_9(x) = \sum_{i=1}^{n} x_i^2$	$[-100,100]^n$
Step 2	$f_{10}(x) = \sum_{i=1}^{n}([x_i + 0.5])^2$	$[-10,10]^n$
Sum squares	$f_{11}(x) = \sum_{i=1}^{n} ix_i^2$	$[-10,10]^n$

Table 9: UNIMODAL TEST FUNCTIONS USED IN THE EXPERIMENTAL STUDY

In table 9, most of the test functions being minimized have a global minimum at $x^* = \mathbf{0}$ (where $\mathbf{0}$ is an n-dimensional zero vector) with optimal value equal zero (i.e., $f_{x^*} = 0$). Exceptions are as follows: Dixon-Price with $f_2(x^*) = 0$ at $[2^{(2^{1-i}-1)}]_{i=1}^n$; Quartic with $f_2(x^*) = Random[0.1]$ (in this paper we set $Random[0.1] = 0.5$) at $x^* = \mathbf{0}$; Rosenbrock with $f_5(x^*) = 0$ at $x^* = [1]^n$; Step 2 with $f_{10}x^{x^*}$ at $x_i^* \in [-0.5, 0.5](i = 1, 2, ..., n)$.

In table 10, forfunctions g_1 , g_4 , g_6 , g_8 , g_{10} and g_{11}, the global minimums are located at $x^* = \mathbf{0}$ with $f(x^*) = 0$. Exceptions are as follows: Deb 1 with $g_2(x^*) = -1$ at $x_i^* = \frac{(2t_i+1)}{2}$ where $t \in (1, 2, ..., n)$; Drop wave with $g_3(x^*) = -1$ at $x^* = [0]^n$; Levy with $g_5(x^*) = 0$ at $x^* = [1]^n$; Penalty 2 with $g_7(x^*) = 0$ at $x^* = [1]^n$; Quintic with $g_9(x^*) = 0$ at $x^* = [-1]^n$ or $x^* = [2]^n$; Sinusoidal with $g_{12}(x^*) = -3.5$ at $x^* = [180]^n$ and and Styblinski–Tang with $g_{13}(x^*) = -39.16599$ at $x^* = [-2.903534]^n$.

In Table 11, all the test functions have a global minimum at $x^* = \mathbf{0}$ with $f(x^*) = 0$.

Name	Function	Initial Interval
Ackley	$g_1(x) = -20\exp(-0.20\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + \epsilon$	$[-32,32]^n$
Deb 1	$g_2(x) = -\frac{1}{n}\sum_{i=1}^{n}\sin^6(5\pi x_i)$	$[-1,1]^n$
Drop wave	$g_3(x) = -\frac{1+\cos(12\sqrt{\sum_{i=1}^{n} x_i^2})}{2+0.5\sum_{i=1}^{n} x_i^2}$	$[-5.12, 5.12]^n$
Griewank	$g_4(x) = 1 + (\frac{1}{4000})\sum_{i=1}^{n} x_i^2 - \Pi_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})$	$[-100,100]^n$
Levy	$g_5(x) = \sin^2(\pi w_1) + \sum_{i=1}^{n}(w_i - 1)^2[1 + 10\sin^2(\pi w_i + 1)] + (w_n - 1)^2$ $w_i = 1 + \frac{x_i-1}{4}$	$[-10,10]^n$
Pathological	$g_6(x) = \sum_{i=1}^{n}(0.5 + \frac{\sin^2(\sqrt{100x_i^2+x_{i+1}^2})-0.5}{1+0.001(x_i-x_{i+1})^4})$	
Penalty2	$g_7(x) = 0.1\{\sin^2(3\pi x_1 + \sum_{i=1}^{n}[(x_i - 1)^2(1 + \sin^2(3\pi x_{i+1}))])\}$ $+(x_n - 1)^2[1 + \sin^2(2\pi x_n)] + \sum_{i=1}^{n} u(x_i, 5, 100.4)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m \end{cases}$	$[-50.50]^n$
Pinter	$g_8(x) = \sum_{i=1}^{n} ix_i^2 + \sum_{i=1}^{n} 20i\sin^2(x_{i-1}\sin(x_i) + \sin(x_{i+1}))$ $+\sum_{i=1}^{n} i\log_{10}(1 + i(x_{i-1}^2 - 2xi + 3x_{i+1} - \cos(x_i) + 1)^2), x_0 = x_n, x_{n+1} = x_i$	$[-10,10]^n$
Quintic	$g_9(x) = \sum_{i=1}^{n} |x_i^5 - 3x_i^4 + 4x_i^3 + 2x_i^2 - 10x_i - 4|$	$[-10,10]^n$
Salomon	$g_{10}(x) = 1 - \cos(2\pi\sqrt{\sum_{i=1}^{n}}) + 0.1\sqrt{\sum_{i=1}^{n}}$	$[-100,100]^n$
sargan	$g_{11}(x) = \sum_{i=1}^{n} n(x_i^2 + 0.4\sum_{i\neq j} x_ix_j)$	$[-100,100]^n$
Sinusoidal	$g_{12}(x) = -[2.5\Pi_{i=1}^{n}\sin(x_i - 30) + \Pi_{i=1}^{n}\sin(5(x_i - 30))]$	$[0, 180]^n$
Styblinski - Tang	$g_{13}(x) = (-\frac{1}{2}n)\sum_{i=1}^{n}(x_i^4 - 16x_i^2 + 5x_i)$	$[-5,5]^n$

Table 10: MULTIMODAL TEST FUNCTIONS USED IN THE EXPERIMENTAL STUDY

Name	Function	Interval
Alpine 1	$h_1(x) = \sum_{i=1}^{n} |x_i \sin(x_i + 0.1 x_i)|$	$[-10, 10]^n$
Bent Cigar	$h_2(x) = x_1^2 + 10^6 \sum_{i=2}^{n} x_i^2$	$[-100, 100]^n$
Bohachevsky	$h_3(x) = \sum_{i=1}^{n-1} [x_i^2 + 2 x_{i+1}^2 - 0.3 \cos(3\pi x_i) - 0.4 \cos(4\pi x_{i+1}) + 0.7]$	$[-50, 50]^n$
Chung Reynold	$g_4(x) = (\sum_{i=1}^{n} x_i^2)^2$	$[-100, 100]^n$
Csendes	$h_5(x) = \sum_{i=1}^{n} x_i^6 (2 + \sin(\frac{1}{x_i}))$	$[-1, 1]^n$
Ellipsoid	$h_6(x) = \sum_{i=1}^{n} (100^{\frac{i-1}{n-1}} x_i)^2$	$[-100, 100]^n$
Rastrigin	$h_7(x) = \sum_{i=1}^{n} [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^n$
Sum of different powers	$h_8(x) = \sum_{i=1}^{n} |x_i|^{i+1}$	$[-1, 1]^n$
Trigonometric 1	$h_9 = \sum_{i=1}^{n} [n - \sum_{j=1}^{n} \cos(x_j) + i(1 - \cos(x_i) - \sin(x_i))]^2$	$[0, \pi]^n$
Wavy	$h_{10}(x) = 1 - \{(\frac{1}{n}) \sum_{i=1}^{n} [\cos(10 x_i) \exp \frac{-x_i^2}{2}]\}$	$[-\pi, \pi]^n$
zakharov	$h_{11}(x) = \sum_{i=1}^{n} x_i^2 + (0.5 \sum_{i=1}^{n} i x_i)^2 + (0.5 \sum_{i=1}^{n} i x_i)^4$	$[-5, 10]^n$

Table 11: BASIC FUNCTIONS WHOSE SHIFTED VERSIONS HAVE BEEN USED IN THE EXPERIMENTAL STUDY

References

[1] Alba, E., Dorronsoro, B., "The exploration/exploitation tradeoff in dynamic cellular genetic algorithms", IEEE Transactions on Evolutionary Computation, vol. 9, pp. 126–142, 2005.

[2] Bazaraa, M. S., Sherali, H. D., Shetty, C. M., "Nonlinear programming: Theory and Algorithms", John Wiley and Sons, New York, NY, 2006.

[3] Bhargava, V., Fateen, S. E. K., "A. Bonilla-Petriciolet, Cuckoo search: a new nature-inspired optimization method for phase equilibrium calculations", Fluid Phase Equilibria, vol. 337, pp. 191-200, 2013.

[4] Cai, Z., Wang, Y., "A multiobjective optimization based evolutionary algorithm for constrained optimization", IEEE Transactions on Evolutionary Computation, vol. 10, pp. 658-675, 2006.

[5] Changdar, C., Mahapatra, G. S., Kumar Pal, R., "An efficient genetic algorithm for multi-objective solid travelling salesman problem under fuzziness", Swarm and Evolutionary Computation, vol. 15, pp. 27-37, 2014.

[6] Chen, Z., Wang, C., "Modelling RFID signal distribution based on neural network combined with continuous ant colony optimization", Neurocomputing, vol. 123, pp. 354-361, 2014.

[7] Childs, J. Stoeber, H., J., "Self-oriented, other-oriented and socially prescribed perfectionism in employees: Relationships with burnout and engagement", Journal of Workplace Behavioral Health, vol. 25, pp. 269-281, 2010.

[8] Cuevas, E., "A swarm optimization algorithm inspired in the behavior of the social-spider", Expert Systems with Applications, vol. 40, no. 16, pp. 6374-6384, 2013.

[9] Dasgupta, D., Zbigniew, M., "Evolutionary algorithms in engineering applications", Springer Science & Business Media, 2013.

[10] DorigoM., Gambardella, L., Middendorf, M., Stutzle, T., "Special section on ant colony optimization", IEEE Transactions on Evolutionary Computation, vol. 6, no. 4, pp. 317-365, 2002.

[11] Du, W., Li, B., "Multi-strategy ensemble particle swarm optimization for dynamic optimization", Information Sciences, vol. 178, pp. 3096-3109, 2008.

[12] Eita, M. A., Fahmy, M. M., "Group counseling optimization", Applied Soft Computing, vol. 22, pp. 585–604, 2014.

[13] Flett, G. L., Hewitt, P. L., Blankstein, K. R., Solnik, M., Van Brunschot, M., "Perfectionism, social problem-solving ability, and psychological distress", Journal of Rational-Emotive and Cognitive-Behavior Therapy, vol. 14, pp. 245-274, 1996.

[14] Formato, R. A., "Central force optimization: a new nature inspired computational framework for multidimensional search and optimization", Studies in Computational Intelligence, vol. 129, pp. 221-238, 2008.

[15] Garai, G., Chaudhurii, B. B., "A novel hybrid genetic algorithm with Tabu search for optimizing multi-dimensional functions and point pattern recognition", Information Sciences, vol. 221, pp. 28-48, 2013.

[16] Geem, Z. W., Kim, J. H., Loganathan, G. V., "A new heuristic optimization algorithm: harmony search", Simulation, vol. 76, no. 2, pp. 60-68, 2001.

[17] Ghaemi, M., Feizi-Derakhshi, M. R., "Forest optimization algorithm", Expert Systems with Applications, vol. 41, no. 15, pp. 6676-6687, 2014.

[18] Ghodousian, A., Raeisian Parvari, M., "A modified PSO algorithm for linear optimization problem subject to the generalized fuzzy relational inequalities with fuzzy constraints (FRI-FC)", Information Sciences, vol. 418-419, pp. 317-345, 2017.

[19] Ghodousian, A., Babalhavaeji, A., "An efficient genetic algorithm for solving nonlinear optimization problems defined with fuzzy relational equations and max-Lukasiewicz composition", Applied Soft Computing, vol. 69, pp. 475–492, 2018.

[20] Ghodousian, A., Naeeimib, M., Babalhavaeji, A., "Nonlinear optimization problem subjected to fuzzy relational equations defined by Dubois-Prade family of t-norms", Computers & Industrial Engineering, vol. 119, pp. 167–180, 2018.

[21] Ghorbani, N., Babaei, E., "Exchange market algorithm", Applied Soft Computing, vol. 19, pp. 177–187, 2014.

[22] Hatamlou, A., "Black hole: A new heuristic optimization approach for data clustering", Information Sciences, vol. 222, pp. 175-184, 2013.

[23] He, S., Wu, Q. H., Saunders, J., "Group search optimizer: an optimization algorithm inspired by animal searching behavior", IEEE Transactions on Evolutionary Computation, vol. 13, pp. 973–990, 2009.

[24] Hewitt, P. L., Flett, G. L., "Perfectionism in the self and social contexts: Conceptualization, assessment, and association with psychopathology", Journal of Personality and Social Psychology, vol. 3, pp. 456-470, 1991.

[25] Hewitt, P. L., Flett, G. L., Perfectionism and stress processes in psychopathology, In Flett, G. L., and Hewitt, P. L., (Eds.), Perfectionism: Theory, research, and treatment (pp. 255-284), Washington, DC: American Psychological Association, 2002.

[26] Hewitt, P. L., Flett, G. L., "Multidimensional Perfectionism Scale (MPS): Technical manual", Toronto: Multi-Health Systems, 2004.

[27] Holland, J. H., "Adaptation in natural and artificial systems: An introductory analysis with applications to biology, Control, and Artificial Intelligence", U Michigan Press, 1975.

[28] Jamil, M., Yang, X. S., "A literature survey of benchmark functions for global optimization problems", International Journal of Mathematical Modelling and Numerical Optimization, vol. 4, no. 2, pp. 150-194, 2013.

[29] Kennedy, J., Eberhart, R.C., "Particle swarm optimization", Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942-1948, 1995.

[30] Kern, S., Müller, S. D., Hansen, N., Büche, D., ocenasek, J., Koumoutsakos, P., "Learning probability distributions in continuous evolutionary algorithms-A comparative review", Natural Computing, vol. 3, no. 1, pp. 77-112, 2004.

[31] Kim, D. H., Abraham, A., Cho, J. H., "A hybrid genetic algorithm and bacteria foraging approach for global optimization", Information Sciences, vol. 177, pp. 3918-3937, 2007.

[32] Kirkpatrick, S., Gelatto, C. D., Vecchi, M. P., "Optimization by simulated annealing", Science, vol. 220, pp. 671-680, 1983.

[33] Malviya, R., Pratihar, D. K., "Tuning of neural networks using particle swarm optimization to model MIG welding process", Swarm and Evolutionary Computation, vol. 1, no. 4, pp. 223-235, 2011.

[34] Mirjalili, S., Mirjalili, S. M., Lewis, A., "Grey wolf optimizer", Advances in Engineering Software, vol. 69, pp. 46-61, 2014.

[35] Mirjalali, S., Lewis, A., "The whale optimization algorithm", Advances in Engineering Software, vol. 95, pp. 51-67, 2016.

[36] Montes, E. M., Coello, C., "A simple multimembered evolution strategy to solve constrained optimization problems", IEEE Transactions on Evolutionary Computation, vol. 9, pp. 1-17, 2005.

[37] Nanda, S. J., Panda, G., "A survey on nature inspired methaheuristic algorithms for partitional clustering", Swarm and Evolutionary Computation, vol. 16, pp.1-18, 2014.

[38] Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M., "The Bees Algorithm-A novel tool for complex optimization problems", Manufacturing Engineering Centre, Cardiff University, Cardiff CF24 3AA, UK, 2005.

[39] Rajabioun, R., "Cuckoo optimization algorithm", Applied Soft Computing, vol. 11, pp. 5508-5518, 2011.

[40] Rao, R. V., Savsani, V. J., Vakharia, D. P., "Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems", Information Sciences, vol. 183, pp. 1–15, 2012.

[41] Rashedi, E., Nezamabadi-pour, H., Saryazdi, S., "GSA: A gravitational search algorithm", Information Sciences, vol. 179, pp. 2232-2248, 2009.

[42] Rechenberg, I., "Evolutionsstrategien", Springer Berlin Heidelberg, 1978 pp. 83–114.

[43] Reynolds, R.G., Peng, B., "Cultural Algorithms: computation modeling of how cultures learn to solve problems: an engineering example", Cybernetics and Systems, vol. 36, pp. 753-771, 2005.

[44] Simon, D., "Biogeography-based optimization", IEEE Transactions on Evolutionary Computations, vol. 12, no. 6, pp. 702-713, 2008.

[45] Smith, M. M., Sherry, S. B., Chen, S., Flett, G. L., Hewitt, P. L., "Perfectionism and narcissism: A meta-analytic review", Journal of Personality Research, vol. 64, pp. 90-101, 2016.

[46] Smith, M. M., Sherry, S. B., Rnic, K., Saklofske, D. H., Enns, M., Gralnick, T., "Are perfectionism dimensions vulnerability factors for depressive symptoms after controlling for neuroticism? A meta-analysis of 10 longitudinal studies", European Journal of Personality, vol. 30, pp. 201-212, 2016.

[47] Smith, M. M., Speth, T. A., Sherry, S. B., Saklofske, D. H., Stewart, S. H., Glowacka, M., "Is socially prescribed perfectionism veridical? A new take on the stressfulness of perfectionism", Personality and Individual Differences, vol. 110, pp. 115-118, 2017.

[48] Socha, K., Dorigo, M., "Ant colony optimization for continuous domain", European Journal of operational Research, vol. 185, pp. 1155-1173, 2008.

[49] Stoeber, J., "How other-oriented perfectionism differs from self-oriented and socially prescribed perfectionism: Further findings", Journal of Psychopathology and Behavioral Assessment, vol. 37, pp. 611-623, 2015.

[50] Stoeber, J., "How other-oriented perfectionism differs from self-oriented and socially prescribed perfectionism", Journal of Psychopathology and Behavioral Assessment, vol. 36, pp. 329-338, 2014.

[51] Stoeber, J., Corr, P. J., Smith, M. M., Saklofske, D. H., Perfectionism and personality. In J. Stoeber (Ed.), The psychology of perfectionism: Theory, research, applications. London: Routledge, 2018, pp. 68-88.

[52] Stoeber, J., Hoyle, A., Last, F., "The consequences of perfectionism scale: Factorial structure and relationships with perfectionism, performance perfectionism, affect, and depressive symptoms", Measurement and Evaluation in Counseling and Development, vol. 46, no. 3, pp. 178-191, 2013.

[53] Storn, R.M., Price, K., "Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization, vol. 11, pp. 341-359, 1997.

[54] Storn, R., "System design by constraint adaptation and differential evolution", IEEE Transactions on Evolutionary Computation, vol. 3, pp. 22-34, 1999.

[55] Stroud, P., "Kalman-extended genetic algorithm for search in nonstationary environments with noisy fitness evaluations", IEEE Transactions on Evolutionary Computation, vol. 5, pp. 66-77, 2001.

[56] Suresh, K., Kumarappan, N., "Hybrid improved binary particle swarm optimization approach for generation maintenance scheduling problem", Swarm and Evolutionary Computation, vol. 9, pp. 69-89, 2013.

[57] Tarasewich, P., McMullen, P. R., "Swarm intelligence: power in numbers", Communication of ACM, vol. 45, pp. 62-67, 2002.

[58] Wang, B., Jin, X., Cheng, B., "Lion pride optimizer: an optimization algorithm inspired by lion pride behavior", Science China Information Sciences, vol. 55, no. 10, pp. 2369-2389, 2012.

[59] Wolpert, D. H., Macready, W. G., "No free lunch theorems for optimization", IEEE transactions on evolutionary computation, vol. 1, no. 1, pp. 67-82, 1997.

[60] Yang, X. S., "A new metaheuristic bat-inspired algorithm", Proceedings of the workshop on nature inspired cooperative strategies for optimization (NICSO 2010), Springer, p. 65-74, 2010.

[61] Yang, X. S., "Firefly algorithm Stochastic Test Functions and Design Optimization", Int. J. Bio-Inspired Computation, vol. 2, no. 2, pp. 78-84, 2010.

[62] Yang, H., Stoeber, J., "The Physical Appearance Perfectionism Scale: Development and Preliminary Validation", Journal of Psychopathology and Behavioral Assessment, vol. 34, no. 1, pp. 69–83, 2012.

[63] Yao, X., Liu, Y., Lin, G., "Evolutionary programming made faster", IEEE Transactions on Evolutionary Computation, vol. 3, pp. 82-102, 1999.

[64] Yeh, W. C., "Novel swarm optimization for mining classification rules on thyroid gland data", Information Sciences, vol. 197, pp. 65-76, 2012.

[65] Zheng, Y. J., "Water wave optimization: a new nature-inspired metaheuristic", Computers and Operations Research, vol. 55, pp. 1-11, 2015.