The Journal of Supercomputing (2024) 80:10746–10795
https://doi.org/10.1007/s11227-023-05790-3

Prism refraction search: a novel physics-based metaheuristic algorithm

Rohit Kundu[1] · Soumitri Chattopadhyay[2] · Sayan Nag[3] · Mario A. Navarro[4] · Diego Oliva[4]

Accepted: 3 November 2023 / Published online: 4 January 2024

Abstract

Single-solution-based optimization algorithms are computationally cheap yet powerful methods that can be used on various optimization tasks at minimal processing expenses. However, there is a considerable shortage of research in this domain, resulting in only a handful of proposed algorithms over the last four decades. This study proposes the Prism Refraction Search (PRS), a novel, simple yet efficient, single-solution-based metaheuristic algorithm for single-objective real-parameter optimization. PRS is a physics-inspired algorithm modeled on a well-known optimization paradigm in ray optics arising from the refraction of light through a triangular prism. The key novelty lies in its scientifically sound background that is supported by the well-established laws of physical optics. The proposed algorithm is evaluated on several numerical objectives, including 23 classical benchmark functions, the CEC-2017 test suite, and five standard real-world engineering design problems. Further, the results are analyzed using standard statistical tests to prove their significance. Extensive experiments and comparisons with state-of-the-art metaheuristic algorithms in the literature justify the robustness and competitive performance of the PRS algorithm as a lightweight and efficient optimization strategy.

Keywords Prism refraction · Metaheuristic · Physics-inspired · Single solution · Single-objective optimization · Continuous search space

1 Introduction

Optimization [42] is one of the most common and fundamental tasks associated with technology in our daily lives. It is used, for example, in GPS systems [55], shipping companies [33] that deliver packages to our homes, finance [23] and airline reservation systems [42], among others. A deeper dive into the mathematical backbone of these systems would reveal an objective function, ranging from simple

Extended author information available on the last page of the article numerical functions to overly complex nonlinear challenging problems, whose optimality is to be searched for. As such, traditional gradient-based optimization methods require derivation of the search space and suffer from the risk of getting trapped in local optima, making them highly inefficient [68, 71]. This has been the primary motivation for researchers to introduce metaheuristic algorithms—those derived from various optimization paradigms occurring in nature [31, 53] and physical systems [62, 80]. These algorithms have proven reliable and effective in finding optimal solutions to challenging real-world problems. Apart from the classical numerical optimization, metaheuristics have proven their mettle on a wide range of optimization tasks, such as feature selection [8, 21, 47], image contrast enhancement [7, 32], image compression [74], traveling salesman problem [52], dynamic vehicle routing [94], multi-level thresholding [1, 15, 77, 103], wireless network routing [10], neural architecture search [9, 63], influence maximization [20], data clustering [54, 66], task scheduling [86] and solving the class-imbalance problem [44, 84], to name a few.

Several metaheuristic optimizers have been proposed in the past, taking inspiration from the evolutionary paradigm [25, 53], wildlife [39, 71, 99], swarm systems [31, 56] and physics-based processes [58, 60, 80], and a method related to light refraction called Light Spectrum Optimizer (LSO) [3]. These methods are population-based, where the constituent agents are candidate solutions that interact among themselves via a given strategy for convergence. The strength of these approaches primarily depends upon balancing between the abilities to (1) explore the search space to avoid local optima and (2) exploit a local region for faster convergence. In contrast, single-solution-based metaheuristics [11, 62] iteratively refine a single solution in the search space and thus mainly exhibit exploitation ability. Further, such methods' weak (or no) exploration ability puts them at risk of getting trapped in local optima, yielding poorer results. As such, single-solution-based approaches have mostly been used as local search strategies and embedded with population-based algorithms [8, 21] to improve the exploitative searchability of the latter. Hence, the challenge remains to develop single-solution-based algorithms that can perform as well as their population-based counterpart, thus being used as a standalone optimizer.

However, the development of metaheuristics in such large numbers each year has led to researchers pondering upon the need for a new algorithm to be proposed at all. A rebuttal to the same is provided by the "No Free Lunch" (NFL) Theorem [97], which states that *no single best optimization algorithm* can provide the global optimum for *all* problems, i.e., some algorithms are suited better to some specific types of problems. This has been the primary motivation for researchers to continue exploring natural or physical phenomena for developing new optimization strategies that might better suit particular kinds of problems. In this paper, we propose the Prism Refraction Search (PRS), a novel single-solution-based algorithm inspired by the optimality paradigm associated with the refraction of light rays by a prism. Specifically, we frame our algorithm based on the *dependency of the angle of deviation* (δ) *on the angle of incidence* (i) *for a light ray after passing through a prism*, which is found to reach a global minimum iteratively. The core idea is to model the measure of deviation as the objective function to be optimized and simulate the experiments mathematically so that the process iteratively arrives at the optimal value. The details of the physical setting along with the proposed algorithm are described in Sect. 3. The proposed optimizer has also been applied to a wide range of constrained engineering optimization problems and compared to several popular metaheuristics [61, 70, 101], where PRS has not only yielded very competitive results but also outperformed the latter on numerous occasions.

1.1 Motivation and contributions

The success of metaheuristics has had significant influence in solving new optimization problems, which demand faster, more robust, or unique algorithms that can be combined with other metaheuristics, either for hybridization or for hyperheuristic approaches to treat this class of algorithms as simpler heuristics controlled by a high-level strategy. Currently, in the state-of-the-art, there is a considerable shortage of research and design of single-solution algorithms, which ultimately could be focused on local search methods and which are sorely lacking in more general approaches due to the inability to establish a balance between exploration and exploitation in their operators, for this reason, the use of these methods to focus on local search can improve the performance of existing algorithms. The proposed method is motivated by the optimization paradigm supported by scientifically possible observations. Although we use elements and concepts of the metaheuristics reported in the state of the art, we try to implement as faithfully as possible to the physical phenomenon to emulate it [92]. Besides having advantages such as being computationally inexpensive and relatively simple to implement. On the other hand, the NFL theorem [97] logically demonstrates that there cannot be a universal algorithm capable of solving all optimization problems, which has been an encouraging factor when proposing a new optimization algorithm among the existing ones.

The main contributions of this paper may be summarized as follows:

1. A new physics-inspired single-solution-based metaheuristic algorithm called Prism Refraction Search (PRS), a method based on the behavior of light rays refracting through a prism, has been proposed. The discussion highlights the optimization paradigm associated with the relevant optical phenomena.
2. A comprehensive survey of physics-based optimization algorithms, which have often been found to be inadequate and deficient in the literature, is given.
3. The proposed optimization algorithm shows a considerable advantage over its competitors in terms of execution time, and this directly influences applications that depend on the speed with which algorithms solve an optimization problem.
4. The proposed optimization algorithm has been extensively tested on 23 classical benchmark functions [102] and the CEC-2017 test suite [14], as well as applied to real-world engineering design problems.
5. Upon comparison in terms of numerical evaluation and nonparametric statistical analysis and in terms of run time, it is found that PRS produces highly comparable

and often better results than those obtained by popular metaheuristic algorithms in the literature.

The rest of this paper is organized as follows: Section 2 provides a comprehensive literature review of metaheuristic algorithms, highlighting various physics-inspired algorithms and single-solution-based methods; Sect. 3 provides an in-depth description of the proposed PRS, including a brief outline on refraction through a prism and the optimization paradigm associated with it, mathematical modeling of PRS, and analysis of its computational complexity; Sect. 4 provides a comparative study of the results obtained by PRS on an extensive range of numerical optimization problems against other popular metaheuristics; and finally, Sect. 5 concludes the present study as well as discusses future directions and extensions.

2 Related work

2.1 Survey on metaheuristic algorithms

Nature-inspired metaheuristics [30, 91] have been a vast field of active research for over three decades and, hence, require no cherry-coated introduction. Metaheuristics can be classified into two categories: (1) single-solution-based methods [11, 29, 62], which start with a single solution in the search space and are iteratively refined to get an improved solution following a trajectory, and (2) population-based methods [53, 61, 71], which deal with a set of solutions that not only iteratively refine themselves but also interact with each other for improvement. As our study aims to propose a single-solution-based metaheuristic, a review of such approaches is provided in Sect. 2.3.

Population-based search algorithms can be further categorized based on the source of inspiration as follows:

- *Evolutionary* algorithms [75], which are inspired by Darwin's theory of evolution, such as the Differential Evolution [25] of the year 1997 and Genetic Algorithm (GA) [53] published in 1992;
- *Swarm-based* approaches [91], which take inspiration from intelligent swarm systems, such as Particle Swarm Optimization (PSO) [61] of year 1995, Ant Colony Optimization (ACO) [31] of year 2005, Artificial Bee Colony (ABC) algorithm [56] published in 2009 that emulates the search for food sources in a honey bee colony, and the Gray Wolf Optimizer (GWO) [71] of year 2014, an algorithm of recent publication in the year 2023, is the Coati Optimization Algorithm [27] in which its fundamental inspirations include the attack and hunting of coatis and the escape behavior of coatis when faced with predators, so it can be classified as swarm-based, with the recent end of the Coronavirus pandemic in the year 2022, algorithms such as the Anti-Coronavirus Optimization Algorithm (ACVO) emerge, which presents a new swarm intelligence strategy in which each agent is a person trying to stay healthy and stop the spread of COVID-19 by observing containment protocols;

- *Nature-inspired* algorithms [90], which model intrinsic laws of nature as well as natural survival strategies of organisms, such as the Water Cycle Algorithm (WCA) [38] published in 2012, Flow Direction Algorithm (FDA) [57] published in 2021, Cuckoo Search (CS) algorithm [99] proposed in 2009, Bat Algorithm (BA) [100] of year 2012, Moth-Flame Optimization (MFO) [68] published in 2015, Marine Predators Algorithm (MPA) [39] and Flower Pollination Algorithm (FPA) [101] published in 2020 and 2014, respectively, a new algorithm published in the year 2022 called Seasons Optimization (SO) algorithm [35], is inspired by the growth cycle of trees in the different seasons of the year. It is a population-based algorithm that works with initial solutions called forest, another algorithm named the Hazelnut Tree Search (HTS) [34] is proposed in 2022 and simulates the search process to find the best hazelnut tree in a forest, this algorithm is composed of three main actuators: growth, fruit dispersion and root propagation, which in different phases guide the heuristic to optimize different problems;
- *Human-related* algorithms [76] that follow common human behavior patterns, such as the Volleyball Premier League (VPL) algorithm [73] and the Teaching-Learning-Based Optimizer (TLBO) [79] published in 2018 and 2011, respectively, or Election algorithm (EA) [37] published in the year 2015, an optimization and search technique, inspired by the presidential election, another algorithm published in 2022 [36] is inspired by human based on traders' behavior and stock price changes in the stock market is the Stock exchange trading, another interesting algorithm is published in 2022 called HFA (Human Felicity Algorithm) [93], which is inspired by the efforts of human society to find happiness, its methodology divides the population of solutions into three different categories: elites, disciples and ordinary people, which allows it to diversify its solutions;
- *Physics-inspired* algorithms [22, 81] of the years 2016 and 2022, respectively, which mimic different phenomena related to various domains of physical sciences, such as Gravitational Search Algorithm (GSA) proposed in 2009 [80], Henry Gas Solubility Optimization (HGSO) of the year 2019 [50] and Multi-Verse Optimizer (MVO) [70] published in 2016, among others.
- *Chemistry-based* takes advantage of the intricate laws governing chemical interactions. Among these algorithms are the innovative Gases Brownian Motion Optimization (GBMO) of the year 2013 [2] and the Artificial Chemical Reaction Optimization Algorithm (ACROA) published in 2011 [13], along with other ingenious creations published in 2017 [87]. These algorithms have carved a niche in the diverse repertoire of computational problem-solving techniques.
- *Mathematical-based* occupy a central place. Metaheuristic algorithms of this class are elaborated by emulating the elegant patterns in mathematical functions. Examples are the charming Golden Sine Algorithm (GSA) [89] proposed in 2017, the robust Base Optimization Algorithm (BOA) published in 2012 [82], and the harmonious Sine-Cosine Algorithm (SCA) [69] published in 2016. These algorithms, inspired by the harmony of numbers, represent a mathematical fusion and computational innovation.
- *Others* Metaheuristic algorithms are born from pure creativity and ingenuity, untouched by the influence of natural phenomena. Here reside some like the Adaptive Large Neighborhood Search Technique (ALNS) [65] and the pio-

neering Large Neighborhood Search (LNS) [78] published in 2022 and 2019, respectively. For instance, the large neighborhood search technique operates on the principles of refinement, employing intricate destroy and repair operators to enhance initial solutions and departing from conventional inspiration sources.

The proposed algorithm in this research is *physics-inspired*, a more detailed survey of which has been provided in the following section.

2.2 Physics-inspired metaheuristics

In addition to natural phenomena, researchers have been inspired by various nonlinear physics processes with an optimization paradigm to model as search strategies and used for engineering optimization tasks. Such simulations have resulted in the development of highly effective optimization algorithms with promising exploration and exploitation capabilities, often outperforming popular nature-inspired metaheuristics on standard engineering problems. The first physics-inspired metaheuristic, Simulated Annealing (SA) [62], was proposed by Kirkpatrick et al. in 1983, which mimicked the annealing process of obtaining optimal configurations of materials by controlled cooling of the same after getting heated sufficiently. Birbil et al. in the year 2003 [18] proposed a metaheuristic algorithm derived from the movement of static charges in a bounded region based on Coulomb's law [48] of electrically charged particles. This law has also inspired a recently proposed metaheuristic artificial electric field algorithm [98] published in 2019. Formato in the year 2008 et al. [41] proposed Central Force Optimization (CFO), the first metaheuristic based on gravitational kinematics that simulated the dynamics of a system of masses. GSA was presented by Rashedi et al. [80] in 2009, which was similar to CFO except that it had a stochastic behavior, unlike the deterministic latter. In a unique work, Geem et al. proposed Harmony Search (HS) in the year 2001 [43] that leveraged principles of acoustics to model the formation of musical harmonies.

Although fewer compared to nature-inspired metaheuristics, the past decade has seen several optimization algorithms being proposed by modeling various physics processes across a plethora of domains, ranging from classical mechanics [26, 28, 80], and fluid dynamics [51, 88] to thermodynamics [50, 59, 72] and cosmology [70, 106]. Moein et al. [72] in th year 2014 proposed an algorithm modeled on the kinetic theory of gases [67]. Mirjalili et al. took inspiration from the multi-verse theory of cosmology and proposed the MVO algorithm [70] published in 2016. Dehghani et al. [26] presented in the year 2017 a spring search algorithm inspired by Hooke's law of spring forces. The authors of [59] proposed in the year 2017 a novel algorithm based on Newton's law of cooling. Hosein et al. proposed the electromagnetic field optimization algorithm [5] published in 2016, which was modeled on the behavior of electromagnets having different polarities in charged fields. In the year 2019, the HGSO [50] proposed by Hashim et al. was based on Henry's law of solubility of gases in a liquid at a given temperature, and Tahani et al. [88] proposed a novel metaheuristic based on fluid mechanics, specifically exploring viscous fluids' laminar or turbulent flow. In contrast, the authors of [51] proposed in 2021 simulated the equilibrium of bodies of varying densities when immersed in a liquid based on Archimedes' principle. Other approaches include a metaheuristic inspired by atmospheric motion and wind directions [16] published in 2013, States of Matter Optimizer (SMO) [24] proposed in 2014 that mimics the extent of motion among molecules in solids, liquids, and gases; Atom Search Optimization (ASO) [105] of the year 2019 that simulates atomic motion through interaction forces, an optimization algorithm based on nuclear fission and fusion reactions [95] published in 2019. In the year 2020, an algorithm modeling the motion of planets in the solar system [106] and a classical mechanics-inspired algorithm based on a system of colliding bodies following the conservation of linear momentum [28]. For a more detailed review of physics-inspired metaheuristics, readers are referred to the survey papers by Biswas et al. [19] and Salcedo et al. [81].

The domain of optics, however, has been significantly less explored in the relevant context. The first optics-based optimization algorithm was proposed by Shen et al. [85] in the year 2009, which aimed at using Fermat's principle to trace the trajectory of light rays while traveling in an uneven transparent medium by incorporating both reflection and refraction. However, its effectiveness was not fully established due to insufficient testing. Kaveh et al. , proposed Ray Optimization [60] published in 2014, which was modeled on ray tracing [45] using Snell's law of refraction when light traveled from a rarer medium to a denser medium. In contrast, in the year 2015, the authors of [58] proposed Optics-Inspired Optimization (OIO) where the surface of the objective function was treated as a spherical mirror, and accordingly, the positions of the light points were updated upon reflection at the surface. Thus, it can be observed that there is a research gap in venturing into different optical phenomena for mathematical modeling and adoption to numerical optimization. To this end, we bring a simple but effective, physical optics-inspired metaheuristic algorithm that mimics the behavior of light rays upon refraction through a prism.

2.3 Single-solution-based methods

Single-solution-based approaches are commonly known as "trajectory methods." They start from an initial position and iteratively move away from it toward the desired minimum, tracing out a search space trajectory. SA [62] published in 1983 was the first proposed single-solution metaheuristic, inspired by the annealing process in material sciences. Other popular single-solution-based methods include the Greedy Randomized Adaptive Search Procedure (GRASP) [40] published in 1995, Variable Neighborhood Search (VNS) [49] published in 1999, and vortex search algorithm [29] of the year 2015. Local search techniques also fall in this category of metaheuristics, such as Tabu Search (TS) [46] of the year 1998, Iterated Local Search (ILS) [64] published in 2003, Hill Climbing (HC) [83] proposed in 2006 and its variants, namely β-Hill Climbing (βHC) [11] and Adaptive β-Hill Climbing (AβHC) [12] published in 2017 and 2019, respectively. Using the above-mentioned works, we propose prism refraction search (PRS), a novel single-solution-based metaheuristic algorithm.

3 Proposed method

3.1 Preliminaries

In this section, we summarize the pre-requisite knowledge of optics that will help to grasp the intuition behind the proposed Prism Refraction Search (PRS) algorithm.

3.1.1 Snell's laws of refraction

Light travels at different speeds in different media. Also, light tends to follow the fastest path to reach one point from the other. Suppose a light ray needs to reach point 'Z' from a point 'X' as shown in Fig. 1. The "shortest path" is a straight line from X to Z, but that is not the "fastest" path because of the different velocities in the two media. The fastest path is X-Y-Z (which can be proved mathematically using calculus). Thus, refraction of light refers to the change in the direction of an obliquely incident light at the interface of two media when the ray travels from one medium to another. Willebrørd Snell experimentally determined the following two laws of refraction:

1. The incident ray (XY), refracted ray (YZ), and the normal drawn at the interface of two media (AB) all lie on the same plane.
2. The ratio of the sines of the angles of incidence (i) and the angle of refraction (r) is constant and is referred to as the "refractive index" of the given pair of media.

Thus, according to Snell's laws of refraction and from Fig. 1, we obtain Eq. 1. μ_1 and μ_2 are called the absolute refractive indices of media 1 and 2, respectively,

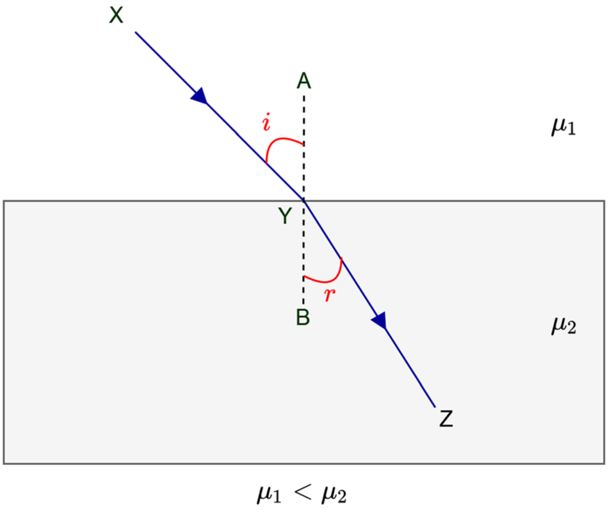


Fig. 1 Illustration of the refraction phenomenon of a monochromatic light ray from a rarer to a denser medium

constituting the media's innate properties. The refractive index may also be given by the ratio of the speed of light in a vacuum and the speed of light in the medium. Refractive indices for a medium are constant since the velocity of light through that medium is constant.

$$\frac{\sin i}{\sin r} = \frac{\mu_2}{\mu_1} = \mu_{21} \tag{1}$$

In the example shown in Fig. 1, $\mu_1 < \mu_2$ that is, light is shown to travel from a "rarer" (lower absolute refractive index) to a "denser" (higher absolute refractive index) medium. Thus, following Eq. 1, light bends *toward* the normal.

3.1.2 Refraction through a prism

Figure 2 illustrates the behavior of light when it passes through a triangular prism AXY. The incident ray MN falls onto the surface AX. It refracts at the interface (assumption: the refractive index of the prism is greater than that of the surrounding, i.e., $\mu_1 < \mu_2$) to follow NO, which refracts again at surface AY to follow path OP. The angle of incidence is i, and the angle of refraction is r_1 at surface AX. At surface AY, the angle of incidence is r_2, and the angle of emergence is e.

Now, consider quadrilateral ANCO in Fig. 2. $\angle ANC$ and $\angle AOC$ both are 90° since NC and OC are normals drawn to the surfaces AX and AY, respectively, of the prism. As the sum of interior angles of a quadrilateral is 360°, it follows from the discussion that $\angle A + \angle NCO = 180°$. Again, from triangle NCO, it follows that $r_1 + r_2 + \angle NCO = 180°$. Therefore, comparing these two expressions, we obtain Eq. 2.

$$r_1 + r_2 = \angle A \tag{2}$$

The total deviation (δ) is the sum of the deviations at the two surfaces AX and AY as shown by Eq. 3.

$$\begin{aligned} \delta =& (i - r_1) + (e - r_2) \\ \implies \delta =& i + e - \angle A \end{aligned} \tag{3}$$

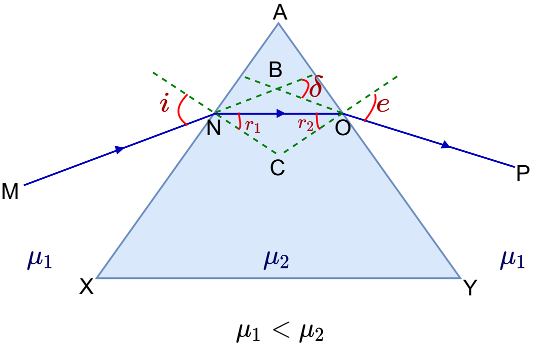


Fig. 2 Illustration of the refraction of a monochromatic light ray through a prism

Let us consider ray MNO. For this refraction, it follows that $\mu_1 \sin i = \mu_2 \sin r_1$. Similarly, for ray NOP, $\mu_2 \sin r_2 = \mu_1 \sin e$. Now, to express angle i in terms of e, μ (where $\mu = \mu_{21} = \frac{\mu_2}{\mu_1}$ and angle A, we follow the steps in Eq. 4.

$$\begin{aligned} \sin i =& \mu \sin r_1 \\ =& \mu \sin(A - r_2) \\ =& \mu\big(\sin A \cos r_2 - \cos A \sin r_2\big) \\ =& \mu\left(\sin A\sqrt{1 - \frac{\sin^2 e}{\mu^2}} - \cos A \frac{\sin e}{\mu}\right) \\ =& \sin A\sqrt{\mu^2 - \sin^2 e} - \cos A \sin e \\ i =& \arcsin\left(\sin A\sqrt{\mu^2 - \sin^2 e} - \cos A \sin e\right) \end{aligned} \tag{4}$$

Equations 3 and 4 will be used to formulate the PRS algorithm later. Figure 3 shows the variation of the deviation angle (δ) concerning the angle of incidence to the prism (i). We see that *minima in the deviation angle occurs when the angle of incidence become equal to the angle of emergence*. Thus, the phenomenon of refraction through a prism can be used as a credible inspiration for formulating an optimization algorithm.

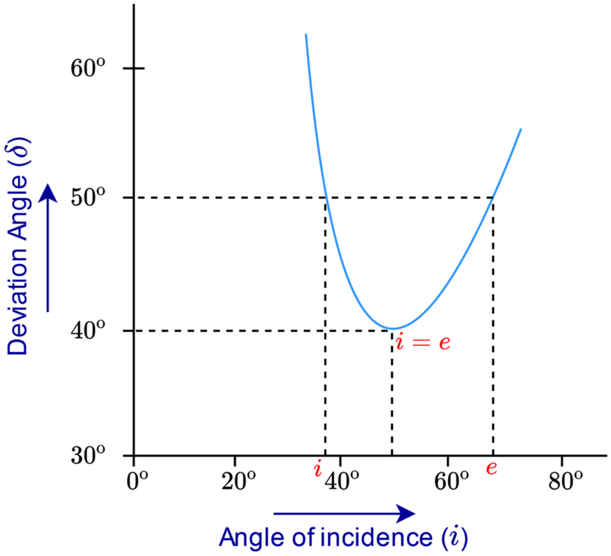


Fig. 3 Plot of the variation of the angle of deviation with respect to the angle of incidence in a prism. i = angle of incidence, e = angle of emergence

3.1.3 Refractive index at minimum deviation

The condition for minimum deviation is that the angle of incidence (i) and the angle of emergence (e) become equal, as can be seen from Fig. 3. Let this minimum deviation be denoted by δ_m. Now, since $i = e$, it implies that $r_1 = r_2 = r$ (say) in Fig. 2. Thus, Eq. 2 becomes $2r = A$ or $r = \frac{A}{2}$.

Similarly, Eq. 3 can be rewritten as Eq. 5 for the minimum deviation case, where A represents the angle of the prism.

$$\begin{aligned} \delta_m =& 2i - A \\ \implies i =& \frac{A + \delta_m}{2} \end{aligned} \tag{5}$$

Thus, the refractive index of the prism at minimum deviation (μ_m) is given by Eq. 6.

$$\mu_m = \frac{\sin i}{\sin r} = \frac{\sin\left(\frac{A+\delta_m}{2}\right)}{\sin\left(\frac{A}{2}\right)} \tag{6}$$

3.1.4 General assumptions

In a real-world scenario, multiple optical phenomena may be associated with refraction, such as a small amount of diffraction [48] due to dust particles in the air or a tiny amount of reflection of the light beam at the edges of the prism. Nevertheless, we have assumed these factors to be absent since our focus solely lies on the optimization paradigm constructed by prism refraction. Further, to keep our algorithm simple and avoid cumbersome geometrical calculations, we assume that an isosceles prism has been used throughout. The prism angle (A) is at the vertex opposite the pair of equal angles.

3.2 Proposed PRS

To show the general characteristics of the PRS proposal, a pseudocode was created that reflects the algorithm's operation.

In this section, we describe the main steps of PRS and correlate them with the intuitions formed in the previous sections. As discussed earlier, the backbone of the proposed algorithm is the variation of the angle of deviation (δ) with the angle of incidence (i) (as shown in Fig. 3), and the paradigm is set on minimizing the deviation value. Hence, δ is modeled as the objective function value, while i forms the solution vector. Mathematically, we can write:

$$\begin{aligned} i_t =& [i_t^1, i_t^2, i_t^3, ..., i_t^N] \\ \delta_t =& \mathcal{F}(i_t) \end{aligned} \tag{7}$$

where $\mathcal{F}(\cdot)$ denotes the objective function to be optimized.

Algorithm 1 General structure of the PRS

```
Initialize population $i_0$ denoted by the solution as incident angle, subject to the bounds
defined by prism angle $A_0$, see Eq.8
for $iter = 1 : MaxIters$ do
    for $i = 1 : OneSolution$ do
        for $j = 1 : Dimensions$ do
            Get fitness calculating the angle of incidence $\delta_t$ using Eq.7
            Obtain BestScore
            if($\delta_t$<BestScore)
            BestScore = fitness
            end if
        end for
    end for
    Calculate the refractive index $\mu_m$ using the Eq.10
    for $i = 1 : OneSolution$ do
        for $j = 1 : Dimensions$ do
            Update emergent angle $E_t$ using Eq.9
            Defined a random number $r1$ into $[-1, 1]$
            Update incident angle $i_{t+1}$ by Eq.11
        end for
    end for
    Update prism angle $A_{t+1}$ by Eq.12
    Update the best solution and position
end for
```

3.2.1 Initial solution

The fundamental quantities required for PRS are the angle of incidence (i) and prism angle (A). We model i as the solution vector of dimension N, which will be updated iteratively during the algorithm's execution. Since the vector represents the angle made by a ray of light with the normal at the point of incidence (refer to Fig. 2 for a better understanding), we initialize it with values between 0^o and 90^o, shifted to the interval bounded by the lower and upper bounds of the objective function to be optimized. The solution vector and the angle of the prism are initialized using Eq. 8.

$$\begin{aligned} i_0 =& LB + (UB - LB) \times \mathcal{U}[0, 90] \\ A_0 =& \max_{1 \le j \le N}(LB) + (\min_{1 \le j \le N}(UB) - \max_{1 \le j \le N}(LB)) \times \mathcal{U}[15, 90] \end{aligned} \tag{8}$$

Here, LB and UB, respectively, denote the vectors containing the lower and upper bounds for the j^{th} variable of the solution vector, while $U(0, 90)$ returns a real number uniformly distributed between 0 and 90 (angle value in degrees). It may be noted that for A, we start with $A_0^j > 15^o$, since beyond 15^o the prism becomes so thin that deviation is almost negligible [48], which, if done at the inception, would lead to premature convergence and limit the performance of the algorithm.

3.2.2 Update solution

This is the core of any optimization algorithm, where the solution obtained at a previous step is updated to a new state expected to be more "fit" (i.e., better) than its former, to gradually converge to a minimum (for a minimization problem).

For a given solution state i_t, the deviation δ can be obtained by evaluating the objective function (shown in Eq. 7). Following Eq. 3, we can obtain the expression for angles of emergence E_t for the current state t as:

$$E_t = \delta_t - i_t + A_t \tag{9}$$

Following the discussion in Sect. 3.1.3, we obtain the expression to calculate the refractive index of the prism at time step t, which is given by:

$$\mu_t = \frac{\sin(\frac{A_t+\delta_t}{2})}{\sin(\frac{A_t}{2})} \tag{10}$$

Modelling upon the derivation provided in Eq. 4 and using the values obtained from Eqs. 8, 9 and 10, we update the solution vector using the following equations:

$$i_{t+1} = \sin^{-1}\left(-\sin E_t \cos A_t + r \times \sin A_t \times \sqrt{\mu_t^2 - \sin^2 E_t}\right) \tag{11}$$

Here, r is a random number $\in [-1, 1]$ used to increase diversity in the solution update step.

Finally, we build on the intuition that thin prisms cause minimal deviation [48] and incorporate a nonlinear decay of the angle of prism A over iterations, given by Eq. 12. This helps in faster convergence since the prism becomes very thin toward the fag end of iterations and drives the algorithm to the optimal value.

$$A_{t+1} = A_t \times \exp\left(\frac{-\alpha \times t}{MaxIter}\right) \tag{12}$$

Here, α is the decay factor whose value is set based on the maximum number of iterations taken. Experimentally, we have set $\alpha = 0.09$. The updated vectors i_{t+1} and A_{t+1} are to be used as the starting state for the next iteration until the maximum number of iterations has been reached. When $t = MaxIter$, the algorithm stops, and the solution vector at that time step, i_t, is returned as the final solution. A flowchart schematically explaining the workflow of PRS is depicted in Fig. 4.

3.3 Computational complexity

Assuming all trigonometric operations occur in constant time, the asymptotic time complexity of PRS is given by $O(N \times MaxIter)$, where N is the dimension of the solution vector (i.e., incident angle) and $MaxIter$ is the maximum number of iterations the algorithm is run. Further, since the algorithm consumes N-dimensional vectors to store E, A, and μ, the space complexity for PRS is $O(N)$.

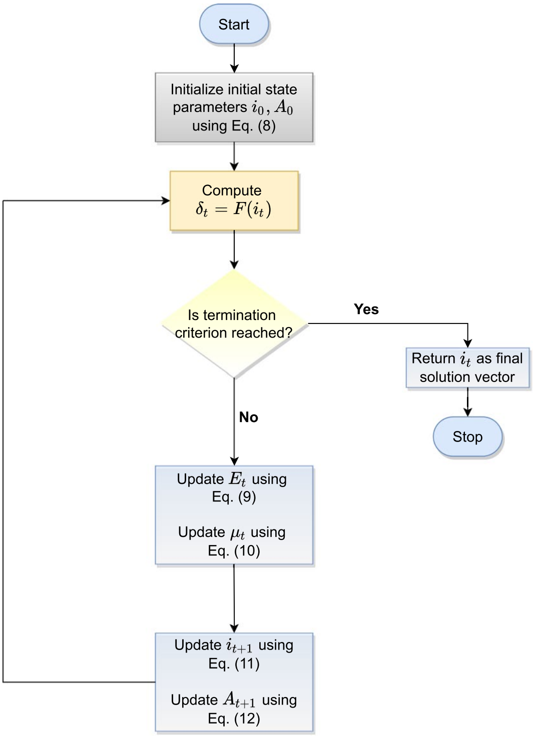

Start
Initialize initial state parameters i_0, A_0 using Eq. (8)
Compute $\delta_t = F(i_t)$
Is termination criterion reached?
Yes
Return i_t as final solution vector
Stop
No
Update E_t using Eq. (9)
Update μ_t using Eq. (10)
Update i_{t+1} using Eq. (11)
Update A_{t+1} using Eq. (12)

Fig. 4 Schematic flowchart depicting the algorithmic steps of the proposed PRS

4 Results and discussion

In this section, we report and analyze the results obtained upon evaluation of the proposed PRS algorithm on a wide range of standard numerical functions that have been specifically proposed [14, 102] to estimate the robustness of an optimization algorithm quantitatively. Further, we compare the results obtained by the PRS algorithm with other state-of-the-art optimization algorithms and perform statistical analysis on the comparative study to demonstrate the competitiveness and reliability of the proposed metaheuristic algorithm. The design of the experiments was conceived in the Matlab 2022B test environment, with a computer equipped with an Intel(R) Core(TM) i7–10,700 series processor, 16 GB of RAM, and GPU (Radeon RX560 Series).

4.1 Evaluation on classical benchmark functions

First, we evaluate our proposed algorithm on a suite of classical benchmark test functions [102] of varying complexities and modalities. The functions used, along with a comprehensive evaluation of the PRS algorithm on them, followed by a comparative analysis against other state-of-the-art metaheuristics, constitute the crux of this section.

4.1.1 Description of benchmark functions

The proposed study uses 23 standard benchmark functions stated in [102] to robustly evaluate the optimization performance of PRS on various types of real-valued objectives. The functions can be broadly classified into three categories:

1. Variable dimensional unimodal functions (F1–F7)
2. Variable dimensional multimodal functions (F8–F13)
3. Fixed dimensional multimodal functions (F14–F23)

The unimodal functions are simple algebraic objectives with a single minimum and, therefore, help estimate the exploitation ability of a given algorithm. On the other hand, the multimodal functions have several local minima and, therefore, are better estimators of an algorithm's overall optimization performance. Following this categorization, the functions are defined in Tables 1, 2 and 3, respectively.

4.1.2 Scalability test for variable dimension functions

For the variable dimensional functions (F1–F13), we test for the dimensional scalability of PRS to investigate its performance across a range of dimensions. We select the dimension range as [5, 10, 30, 50, 100]. We evaluate the PRS algorithm on the functions for 30 independent runs and report the average and Standard Deviation (SD) values of fitness obtained across the runs for each dimension. The maximum number of iterations for PRS is kept experimentally fixed as 5000, and it is worth mentioning that in the case of the tests with the CEC2017 functions, we used function accesses as a stopping criterion to give variety to the experiments performed and to observe the performance in both cases. The overall results are tabulated in Table 4.

Table 1 Variable dimensional unimodal test functions (F1–F7) used to evaluate the proposed PRS algorithm

Function	Definition	Dimensions	Range	F_{min}
F1	$F_1(x) = \sum_{i=1}^{n} x_i^2$	5, 10, 30, 50, 100	$[-100, 100]$	0
F2	$F_2(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	5, 10, 30, 50, 100	$[-10, 10]$	0
F3	$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j\right)^2$	5, 10, 30, 50, 100	$[-100, 100]$	0
F4	$F_4(x) = \max_i \left\{\lvert x_i\rvert, 1 \leq i \leq n\right\}$	5, 10, 30, 50, 100	$[-100, 100]$	0
F5	$F_5(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2\right]$	5, 10, 30, 50, 100	$[-30, 30]$	0
F6	$F_6(x) = \sum_{i=1}^{n} \left(\left[x_i + 0.5\right]\right)^2$	5, 10, 30, 50, 100	$[-100, 100]$	0
F7	$F_7(x) = \sum_{i=1}^{n} ix_i^4 + \text{random}[0, 1)$	5, 10, 30, 50, 100	$[-128, 128]$	0

Table 2 Variable dimensional multimodal test functions (F8–F13) used to evaluate the proposed PRS algorithm

Function	Definition	Dimensions	Range	F_{min}
F8	$F_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	5, 10, 30, 50, 100	$[-500, 500]$	$-418.9829 \times n$
F9	$F_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right]$	5, 10, 30, 50, 100	$[-5.12, 5.12]$	0
F10	$F_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} cos\left(2\pi x_i\right)\right) + 20 + e$	5, 10, 30, 50, 100	$[-32, 32]$	0
F11	$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	5, 10, 30, 50, 100	$[-600, 600]$	0
F12	$F_{12}(x) = \frac{\pi}{n}\left\{10\sin\left(\pi y_1\right) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	5, 10, 30, 50, 100	$[-50, 50]$	0
	$y_i = 1 + \frac{x_i+1}{4}u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 - a & < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
F13	$F_{13}(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}\left(x_i - 1\right)^2\left[1 + \sin^2(3\pi x_i + 1)\right]\right.$	5, 10, 30, 50, 100	$[-50, 50]$	0
	$\left.+(x_n - 1)^2\left[1 + \sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$			

Table 3 Fixed dimensional multimodal test functions (F14–F23) used to evaluate the proposed PRS algorithm

Function	Definition	Dimensions	Range	F_{min}
F14	$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6}\right)^{-1}$	2	$[-65, 65]$	1
F15	$F_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1\left(b_i^2+b_i x_2\right)}{b_i^2+b_i x_3+x_4}\right]^2$	4	$[-5, 5]$	1
F16	$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	−1.0316
F17	$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	$[-5, 5]$	0.398
F18	$F_{18}(x) = \left[1 + \left(x_1 + x_2 + 1\right)^2\left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right]$	2	$[-2, 2]$	3
	$\times\left[30 + \left(2x_1 - 3x_2\right)^2\times\left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]$			
F19	$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	3	$[1, 3]$	−3.86
F20	$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	6	$[0, 1]$	−3.32
F21	$F_{21}(x) = -\sum_{i=1}^{5}\left[(X - a_i)\left(X - a_i\right)^T + c_i\right]^{-1}$	4	$[0, 10]$	−10.1532
F22	$F_{22}(x) = -\sum_{i=1}^{7}\left[(X - a_i)\left(X - a_i\right)^T + c_i\right]^{-1}$	4	$[0, 10]$	−10.4028
F23	$F_{23}(x) = -\sum_{i=1}^{10}\left[(X - a_i)\left(X - a_i\right)^T + c_i\right]^{-1}$	4	$[0, 10]$	−10.5363

It is intuitive enough that any optimization algorithm [53, 61, 70] would perform superior in a low-dimension setup than a higher one, and our proposed PRS is no exception to the general trend. It can be observed that when *dim* = 5, PRS achieves the best fitness values more often than at higher-dimensional values. Furthermore, for *dim* = 100, the PRS algorithm shows a relatively poorer performance regarding fitness scores. However, it is to be noted that the performance of PRS is majorly stable when the dimension value gradually increases from *dim* = 5 to *dim* = 10, after which the performance drops for further higher dimensions. This implies that PRS shows a considerable degree of invariance to small changes in dimension, which adds to its reliability across various optimization tasks.

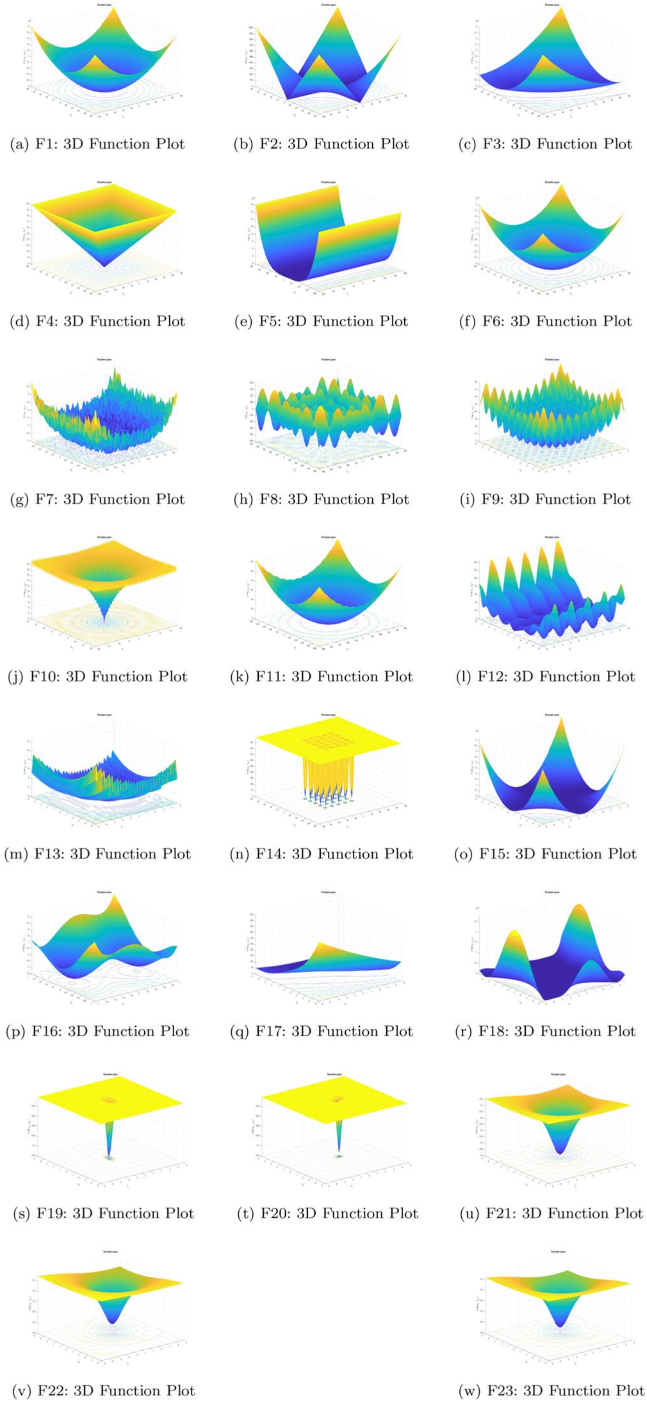
An optimal choice of dimension can be reached at *dim* = 30, where PRS has shown remarkable stability in fitness scores obtained, almost performing at par with the scenario where *dim* = 5 was taken. We set this as the dimension for variable dimension test functions for a fair comparison with other algorithms.

Figure 5 shows the 3D representation of the classified optimization functions in Tables 1, 2, and 3 used in the test to test population-based algorithms against the proposed PRS.

Table 4 Dimensional scalability analysis of PRS on the classical benchmark functions with variable dimensions. The function definitions are provided in Table 1 and Table 2

Function	*dim* = 5		*dim* = 10		*dim* = 30		*dim* = 50		*dim* = 100	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
F1	**3.37E-03**	1.51E-04	7.88E-03	5.94E-06	1.32E-01	9.58E-07	8.49E-01	9.96E-03	1.36E+00	7.44E-05
F2	3.68E+01	2.95E+00	3.37E+02	2.14E+00	**6.27E+00**	1.15E-02	4.54E+02	1.14E+01	5.49E+07	2.45E+02
F3	**3.81E-02**	7.94E-03	2.21E+00	1.15E+00	8.50E+00	5.12E-02	1.48E+01	9.80E-01	3.41E+01	1.20E+01
F4	**1.43E-01**	5.70E-03	2.66E-01	8.90E-03	2.81E-01	3.44E-08	1.57E-01	1.47E-05	2.10E-01	1.14E-03
F5	6.96E+04	2.50E+02	2.76E+07	1.25E+03	**2.56E+01**	5.67E+00	1.19E+04	2.23E+02	3.52E+08	2.74E+04
F6	**4.10E-03**	1.15E-05	4.22E-02	1.25E-05	2.14E-01	5.64E-05	1.09E+00	1.00E-01	5.20E-02	1.40E-04
F7	**1.09E+00**	1.00E+00	9.02E+00	1.00E-01	4.28E+02	1.22E-01	7.41E+03	3.65E+02	1.14E+03	7.89E+02
F8	–2.82E+02	1.94E-02	–6.36E+02	1.11E-04	–1.91E+03	5.58E-05	**–6.36E+03**	5.45E+00	–3.18E+03	5.89E-01
F9	**6.60E+01**	2.25E+01	1.15E+02	2.51E+00	4.27E+02	1.60E+00	1.84E+03	2.80E+01	7.59E+02	1.50E+01
F10	1.91E+01	1.00E+00	1.99E+01	1.02E+00	**1.39E-04**	1.01E-04	1.98E+01	1.45E-01	1.99E+01	6.25E+00
F11	1.04E+00	1.20E-01	3.29E+00	4.26E-03	**1.47E-03**	1.20E-04	2.67E-03	4.45E-05	3.40E-02	4.70E-04
F12	**9.36E-16**	0.00E+00	2.00E-09	1.56E-10	8.25E+01	2.76E+01	7.14E-02	1.12E-06	1.63E+04	5.26E+02
F13	**7.86E-07**	4.26E-10	7.21E+03	1.15E+01	2.37E-03	7.99E-04	2.38E+00	1.15E+00	2.31E+05	6.51E+02

The bold values corresponds to the best value obtainedThe bold values corresponds to the best value obtained



(a) F1: 3D Function Plot

(b) F2: 3D Function Plot

(c) F3: 3D Function Plot

(d) F4: 3D Function Plot

(e) F5: 3D Function Plot

(f) F6: 3D Function Plot

(g) F7: 3D Function Plot

(h) F8: 3D Function Plot

(i) F9: 3D Function Plot

(j) F10: 3D Function Plot

(k) F11: 3D Function Plot

(l) F12: 3D Function Plot

(m) F13: 3D Function Plot

(n) F14: 3D Function Plot

(o) F15: 3D Function Plot

(p) F16: 3D Function Plot

(q) F17: 3D Function Plot

(r) F18: 3D Function Plot

(s) F19: 3D Function Plot

(t) F20: 3D Function Plot

(u) F21: 3D Function Plot

(v) F22: 3D Function Plot

(w) F23: 3D Function Plot

Fig. 5 3D visualization of search spaces of the optimization functions **a**-**r** (F1–F18) defined in Tables 1,2, 3, **s**-*w* (F19–F23) defined in Table 3

4.1.3 Comparison with single-solution-based metaheuristic algorithms

Since the proposed PRS is a single-solution-based algorithm, we must compare its performance on the classical benchmark functions with popular single-solution-based approaches in the literature. Specifically, PRS has been compared with VNS [49], TS [46], SA [62], Hill Climbing [83] and its variants, namely βHC [11] and AβHC [12]. The parameter values of these optimization algorithms, considered in the present work, are provided in Table 5.

For a fair comparison, each algorithm is run for maximum iterations of 5000, and the mean and SD values over 30 independent runs on every function are reported. Following the preceding discussion, the dimension for variable dimensional functions has been set to 30. The results of this comparative study are shown in Table 6, while the results from ranking-based Friedman's test obtained by each algorithm from the comparison are put in Table 7.

The results show the competitive performance of PRS against other single-solution-based approaches. For unimodal functions (F1–F7), the performance of PRS is somewhat weaker compared to TS and SA, achieving the best result only for F5. For high-dimensional multimodal functions (F8–F13), the PRS ranks first for two functions (F11 and F13) while achieving results close to the best-performing algorithms for F8 and F10. It can be seen from Table 6 that the PRS, AβHC, and βHC algorithms rank first on this category of functions, each achieving the best values non-concurrently on two occasions. However, the PRS algorithm shows commendable performance for fixed (i.e., lower) dimensional multimodal functions, achieving the best mean fitness on five occasions. At the same time, AβHC and βHC jointly rank second, achieving the best results concurrently on four functions. Thus, the results imply that the proposed PRS algorithm has the potential to compete with state-of-the-art single-solution-based methods and, therefore, can be used as a reliable optimization algorithm.

Table 5 Parameter settings for the comparative single-solution-based algorithms

Algorithm	Parameter(s)	Value(s)
VNS [49]	Perturbation factor (k)	$k = 5$
TS [46]	Tabu list size (TN)	$TN = N/2$
SA [62]	Initial temperature (T_o) Temperature decay rate (α)	$T_o = 100$ $\alpha = 0.85$
AβHC [12]	Exploitation factor (K) Minimum value of β rate (β_{min}) Maximum value of β rate (β_{max})	$K = 30$ $\beta_{min} = 0.1$ $\beta_{max} = 1$
βHC [11]	Exploitation operator (η) Exploration operator (β)	$\eta = 0.5$ $\beta = 0.05$
HC [83]	NA	NA

Table 6 Comparison between PRS and other single-solution-based methods in the literature on classical benchmark functions

Function	PRS		VNS		TS		SA		AβHC		βHC		HC	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
F1	1.32E-01	9.58E-07	7.33E+01	1.29E+02	**1.78E-10**	8.11E-11	8.06E-06	2.23E-06	7.01E-07	2.38E-07	1.02E-02	1.06E-02	3.26E-09	1.35E-09
F2	6.27E+00	1.15E-02	**4.75E–06**	1.02E-06	4.66E-05	7.64E-06	5.75E-03	1.22E-03	3.06E-03	5.47E-04	5.62E-05	1.16E-05	1.85E-04	3.67E-05
F3	8.50E+00	5.12E-02	4.57E+01	9.86E+01	**2.05E-04**	5.74E-05	2.22E-03	1.67E-03	1.54E+03	4.81E+02	4.94E+02	1.71E+02	4.64E-03	1.84E-03
F4	2.81E-01	3.44E-08	6.75E+01	4.42E+00	2.67E+01	7.13+00	**1.69E-03**	1.31E-03	2.45E+00	4.54E-01	1.13E+00	1.84E-01	1.84E-03	4.63E+01
F5	**2.56E+01**	5.67E+00	4.10E+01	4.88E+01	3.73E+01	3.92E+01	3.20E+01	4.02E+01	5.25E+01	4.20E+01	8.87E+01	3.48E+01	4.31E+01	6.00E+01
F6	2.14E-01	5.64E-05	5.21E+01	1.07E+02	**1.67E–10**	8.33E-11	8.15E-06	1.75E-06	7.20E-07	2.91E-07	1.36E-02	2.37E-02	3.06E-09	1.31E-09
F7	4.28E+02	1.22E-01	4.73E-03	1.77E-03	**2.53E–04**	9.11E-05	1.71E-02	6.38E-03	1.29E-02	5.12E-03	1.73E-02	5.29E-03	9.85E-04	3.56E-04
F8	–1.91E+03	5.58E-05	–7.52E+03	9.22E+02	–7.70E+03	9.30E+02	–7.83E+03	7.98E+02	**–1.26E+04**	5.37E-08	–1.26E+04	9.17E-02	–7.54E+03	9.20E+02
F9	4.27E+02	1.60E+00	2.61E+02	3.78E+01	2.63E+02	4.78E+01	2.63E+02	3.68E+01	2.65E-04	1.71E-04	**3.63E–08**	1.66E-08	2.61E+02	5.04E+01
F10	1.39E-04	1.01E-04	1.95E+01	1.36E-01	1.94E+01	1.61E-01	1.95E+01	1.14E-01	9.10E-04	2.67E-04	**2.20E–05**	1.50E-05	1.95E+01	1.92E-01
F11	**1.47E–03**	1.20E-04	3.86E+02	1.00E+02	2.47E+01	1.23E+01	4.02E+01	1.71E+01	7.74E-02	7.49E-02	1.40E-01	4.68E-02	3.75E+01	1.42E+01
F12	8.25E+01	2.76E+01	2.52E+01	1.39E+01	2.18E+01	1.79E+01	1.76E+01	1.71E+01	**1.14E–07**	2.13E-07	4.37E-05	1.68E-04	1.95E+01	1.75E+01
F13	**2.37E–03**	7.99E-04	9.58E+01	2.71E+01	8.79E+01	2.76E+01	6.10E+01	3.92E+01	3.70E-03	5.24E-03	2.58E-03	4.59E-03	9.45E+01	2.59E+01
F14	**1.01E+00**	9.85E-08	1.28E+01	8.29E+00	1.27E+01	7.77E+00	1.61E+01	7.31E+00	1.98E+00	2.72E-16	1.98E+00	1.98E-16	1.39E+01	7.00E+00
F15	**2.90E-04**	1.20E-06	3.71E-02	4.70E-02	4.24E-02	5.37E-02	3.60E-03	8.01E-03	1.08E-03	5.39E-04	5.21E-04	3.78E-04	2.45E-02	4.12E-02
F16	–1.37E-01	1.57E-03	–5.96E-01	4.14E-01	–4.14E-01	7.93E-01	–8.96E-01	3.09E-01	**–1.03E+00**	3.76E-14	**–1.03E+00**	1.24E-15	5.80E-02	1.09E+00
F17	2.47E+00	3.46E-03	**3.98E-01**	1.24E-14	**3.98E-01**	7.60E-13	**3.98E-01**	9.43E-07	**3.98E-01**	6.01E-14	**3.98E-01**	8.96E-16	**3.98E-01**	2.38E-11
F18	3.49E+02	1.88E+01	2.42E+02	3.67E+02	1.07E+02	2.50E+02	1.30E+02	2.84E+02	**3.00E+00**	5.90E-12	**3.00E+00**	7.63E-14	1.88E+02	3.33E+02
F19	–3.31E+00	5.41E-01	–2.61E+00	1.28E+00	–2.94E+00	1.23E+00	–3.10E+00	1.37E+00	**–3.86E+00**	4.88E-12	**–3.86E+00**	3.58E-14	–2.61E+00	1.28E+00
F20	-2.36E+00	4.23E–03	-3.30E+00	4.84E-02	–3.29E+00	5.54E-02	**–1.21E+00**	1.61E+00	–3.28E+00	5.83E-02	–3.30E+00	4.51E-02	–3.28E+00	5.83E-02
F21	**–9.95E+00**	1.00E+00	–5.90E+00	3.60E+00	–7.24E+00	3.65E+00	–6.56E+00	3.52E+00	–6.14E+00	3.44E+00	–5.65E+00	3.37E+00	–5.41E+00	3.49E+00
F22	**–9.94E+00**	0.00E+00	–5.02E+00	3.13E+00	–5.27E+00	3.47E+00	–4.81E+00	2.98E+00	–6.27E+00	3.26E+00	–5.21E+00	3.28E+00	–4.75E+00	2.97E+00
F23	**–9.97E+00**	0.00E+00	–4.08E+00	2.70E+00	–3.79E+00	2.17E+00	–4.80E+00	3.32E+00	–6.60E+00	3.32E+00	–5.20E+00	3.39E+00	–4.92E+00	3.30E+00

The bold values corresponds to the best value obtained

Table 7 Results obtained on ranking-based Friedman's statistical test upon comparison with each single-solution-based algorithm on classical benchmark functions. The ranks are based on the results shown in Table 6

Function	PRS	VNS	TS	SA	AβHC	βHC	HC
F1	6	7	1	4	3	5	2
F2	7	1	2	6	5	3	4
F3	4	5	1	2	7	6	3
F4	3	7	6	1	5	4	2
F5	1	4	3	2	6	7	5
F6	6	7	1	4	3	5	2
F7	7	3	1	5	4	6	2
F8	7	6	4	3	1	1	5
F9	7	3	5	5	2	1	3
F10	2	5	4	5	3	1	5
F11	1	7	4	6	2	3	5
F12	7	6	5	3	1	2	4
F13	1	7	5	4	3	2	6
F14	1	5	4	7	2	2	6
F15	1	6	7	4	3	2	5
F16	6	4	5	3	1	1	7
F17	7	1	1	1	1	1	1
F18	7	6	3	4	1	1	5
F19	3	6	5	4	1	1	6
F20	6	1	3	7	4	1	4
F21	1	5	2	3	4	6	7
F22	1	5	3	6	2	4	7
F23	1	6	7	5	2	3	4
Average rank	4.04	4.91	3.57	4.09	2.87	2.96	4.35
Final rank	**4**	**7**	**3**	**5**	**1**	**2**	**6**

The bold values corresponds to the best value obtained

4.1.4 Execution time comparison for single-solution-based metaheuristic algorithms running on benchmark functions

Radial charts, also known as spider charts or spiderweb charts, provide a unique and effective visual representation for displaying multiple variables or attributes of a dataset. In these charts, each axis represents a specific dimension of the dataset, and the lines connecting these axes form a structure resembling a spiderweb. Each data point is represented as a line extending from the center of the chart to the corresponding point on the associated axis. In this case, the charts shown in Figs. 6 and 7 were used to represent the classification of execution times for the algorithms under investigation in each experiment. Similarly, Table 8 shows the execution times and an average ranking, as well as the final ranking for the tests of algorithms based on a single solution in the benchmark functions tests. Although it is evident that PRS does not have the first position, it is observed that PRS has the second-best execution time on the table, which gives it an advantage by converging much faster than the rest of the competing algorithms. Something we can comment on is that PRS was

Table 8 Runtime comparison for single-solution-based metaheuristic algorithms

Function		PRS	VNS	TS	SA	AβHC	βHC	HC
F1	$\bar{x}(time)$	1.21	1.40	2.04	1.25	1.48	1.13	1.54
F2	$\bar{x}(time)$	1.25	1.44	2.06	1.28	1.54	1.15	1.58
F3	$\bar{x}(time)$	1.20	1.46	2.09	1.31	1.56	1.18	1.61
F4	$\bar{x}(time)$	1.25	1.50	2.21	1.34	1.58	1.20	1.62
F5	$\bar{x}(time)$	1.34	1.57	2.22	1.39	1.64	1.30	1.83
F6	$\bar{x}(time)$	1.20	1.30	2.03	1.22	1.43	1.15	1.58
F7	$\bar{x}(time)$	0.88	1.03	1.72	0.99	1.07	0.84	1.24
F8	$\bar{x}(time)$	0.96	1.10	1.74	0.95	1.15	0.86	1.27
F9	$\bar{x}(time)$	1.13	1.23	1.70	0.59	0.97	0.67	1.17
F10	$\bar{x}(time)$	0.38	0.87	1.28	0.43	0.78	0.58	0.80
F11	$\bar{x}(time)$	0.43	0.57	1.36	0.47	0.60	0.42	0.82
F12	$\bar{x}(time)$	1.10	1.15	2.12	1.16	1.36	1.02	1.57
F13	$\bar{x}(time)$	1.66	1.64	2.35	1.58	1.73	1.47	1.86
F14	$\bar{x}(time)$	0.59	0.72	1.42	0.62	0.76	0.53	0.94
F15	$\bar{x}(time)$	0.71	0.95	1.65	0.81	0.92	0.68	1.11
F16	$\bar{x}(time)$	0.77	0.93	1.63	0.83	1.06	0.74	1.21
F17	$\bar{x}(time)$	0.77	0.94	1.59	0.82	0.99	0.77	1.15
F18	$\bar{x}(time)$	0.88	1.07	1.76	0.92	1.14	0.86	1.26
F19	$\bar{x}(time)$	1.02	1.16	1.94	1.05	1.22	0.96	1.46
F20	$\bar{x}(time)$	0.96	1.14	1.93	1.07	1.29	0.99	1.41
F21	$\bar{x}(time)$	0.97	1.15	1.81	1.01	1.25	0.98	1.34
F22	$\bar{x}(time)$	1.26	1.46	2.13	1.35	1.55	1.24	1.65
F23	$\bar{x}(time)$	1.42	1.58	2.24	1.52	1.66	1.33	1.74
	$\bar{x}(Rank)$	1.01	1.19	1.87	1.04	1.25	0.96	1.38
	Total rank	2	4	7	3	5	1	6

conceived as a local search algorithm; due to the nature of its source and the fact that it is a single solution, it can be affected by very complex or high-dimensionality functions being possible to get stuck in local solutions, which can be evidenced in Fig. 6 with functions 9 and 13 where it has a hard time converging to better solutions which is reflected in time.

4.1.5 Comparison with popular population-based metaheuristic algorithms

To further demonstrate the competitiveness of PRS as an optimization algorithm, we also compare its performance on classical test functions against several state-of-the-art population-based metaheuristics, namely MFO [68], GA [53], BA [100], PSO [61] and GSA [80]. The parameter settings taken for each algorithm are provided in Table 9.

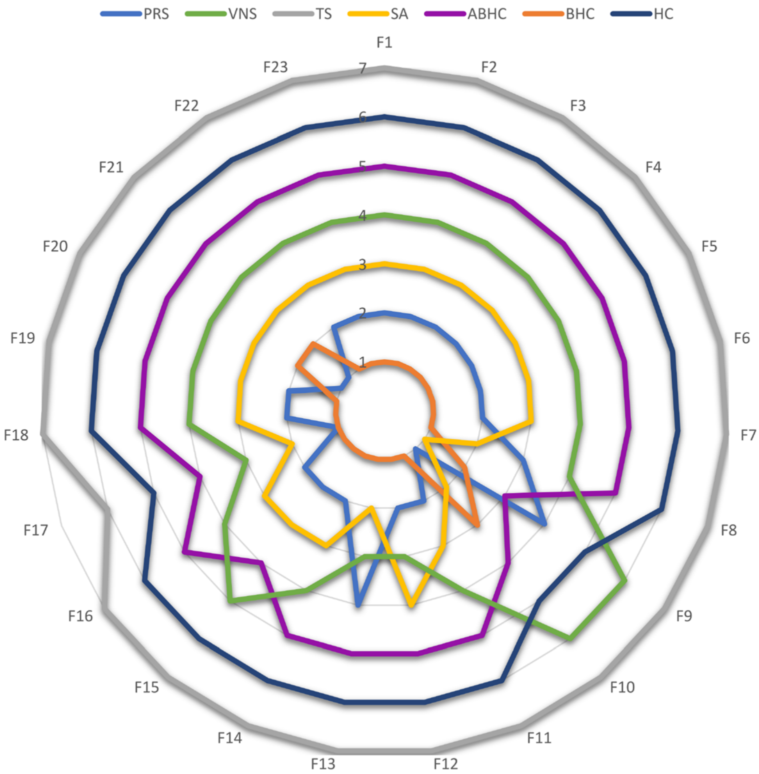


Fig. 6 Radial graph showing the ranking of lowest running times on the benchmark functions with single-solution algorithms

For a fair comparison, the population size of each algorithm has been set to 20. Each algorithm is run for maximum iterations of 5000, and the mean and SD values over 30 independent runs on every function are reported. The results of this comparative analysis are tabulated in Table 10, while the results from ranking-based Friedman's test obtained by each algorithm from the comparison are put in Table 11.

The results demonstrate the highly competitive performance of PRS when pitted against several population-based optimization algorithms, achieving the best results on 13 out of 23 test functions upon comparison. Among unimodal functions (F1–F7), the PRS algorithm performs best on six occasions, depicting its superior exploitation ability to other optimizers. The PRS yields the best results on five functions among multimodal functions with variable dimensions (F8–F13). In contrast, for fixed dimensional multimodal functions (F14–F23), the MFO performs best on five occasions compared to four by PRS. However, the results obtained by the latter are very much comparable. In a nutshell, our PRS algorithm performs auspicious, often outperforming popular metaheuristics, thereby justifying its reliability as a robust optimizer. Convergence plots are added in Fig. 8 to show the iterative process

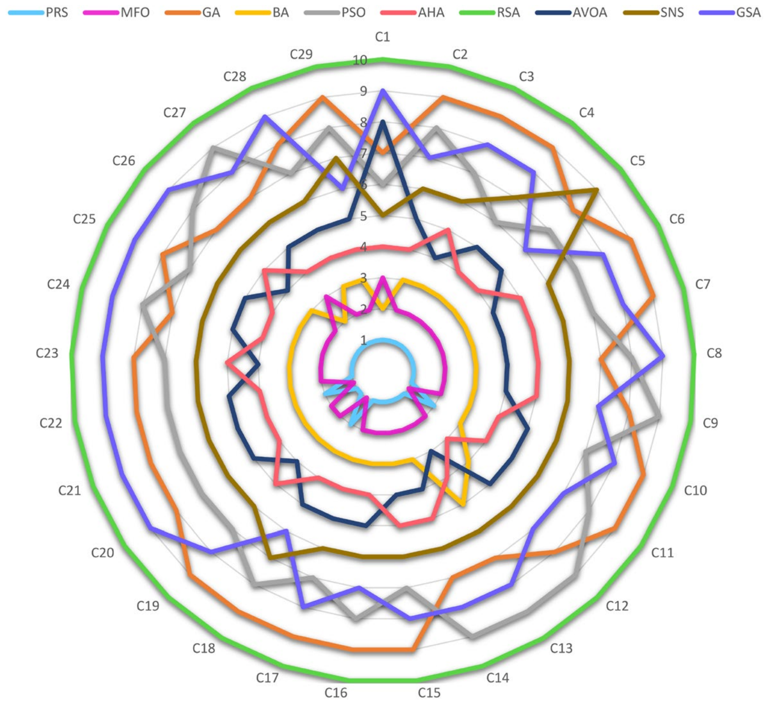


Fig. 7 Radial graph showing the ranking of lowest running times on the CEC-2017 test suit

performance of all tested algorithms on the 23 test functions in Tables (1,2, 3), and a total of 500 iterations was set as the stopping criterion for plotting.

4.2 Performance evaluation on the CEC-2017 test suite with popular and recent state-of-the-art algorithms

We also evaluate the proposed PRS algorithm on the CEC-2017 test suite designed by Awad et al. [14] consisting of real-valued continuous optimization problems. The test suite comprises 29 objective functions across the following categories:

1. Unimodal (C1 and C2) comprising shifted and rotated Bent Cigar and Zakharov functions, respectively;
2. Multimodal (C3 – C9), respectively, denoting the shifted and rotated Rosenbrock's, Rastringen's, Expanded Scaffer's, Lunacck Bi-Rastringen's, Non-continuous Rastringen's, Levy's and Schwefel's functions;
3. Hybrid functions (C10 – C19); and
4. Composition functions (C20 – C29)

Table 9 Parameter settings for the comparative population-based metaheuristic algorithms

Algorithm	Parameter(s)	Value(s)
GA [53]	Crossover probability Mutation probability	0.9 0.01
PSO [61]	Inertia weight (I) Cognitive constant (C_1) Social constant (C_2)	I lies in [1 0] $C_1 = 1$ $C_2 = 1$
BA [100]	Initial loudness (A) Pulse rate (r) Frequency bounds (Q_{min}, Q_{max})	$A = 0.5$ $r = 0.5$ $Q_{min} = 0, Q_{max} = 2$
GSA [80]	Gravitational constant (G_o) Decay factor (α)	$G_o = 100$ $\alpha = 20$
MFO [68]	Convergence operator (a) Spiral shape operator (b)	a lies in $[-2\ -1]$ $b = 1$
AHA [104]	Migration coefficient ($2n$) Where (n) is the number of hummingbirds	$n = 50$
RSA [6]	Alpha constant (α) Beta constant (β)	$\alpha = 0.1$ $\beta = 0.005$
AVOA [4]	Probability constant($p1$) Probability constant($p2$) Probability constant($p3$) Alpha constant (α) Beta constant (β) Gamma constant (γ)	$p1 = 0.6$ $p2 = 0.4$ $p3 = 0.6$ $\alpha = 0.8$ $\beta = 0.2$ $\gamma = 2.5$
SNS [17]	Random mood	$Mood = randi(4)$

We have followed the specifications stated in [14] and have set the search space range as $[-100, 100]$, while the variable dimension has been experimentally fixed to 10.

The PRS algorithm has been evaluated on the CEC'17 test suite and compared with popular metaheuristics found in the literature, such as MFO [68], GA [53], BA [100], PSO [61] and GSA [80]; however, we wanted to contrast with recent algorithms so we added four algorithms from recent years to test their performance compared to our PRS proposal. The algorithms chosen for their good performance reported in the literature are the following: AHA [104], AVOA [4], RSA [6], and SNS [17]. The parameter settings for each comparative algorithm are already listed in Table 9. We report the mean fitness and SD values across 30 independent runs of the algorithm for each function for the experimentation. The stopping criterion used on this test was 100,000 function evaluations, as specified in the technical report of CEC17. Tests will be set with 10,000 function evaluations multiplied by the number of dimensions of the problem. In this case, the number of dimensions for the problems was 10. The results are tabulated in Table 12, while the results from ranking-based Friedman's test obtained by each algorithm from the comparison are put in Table 13. The results show the competitive performance of PRS against state-of-the-art algorithms, outperforming the latter on several occasions. Overall, PRS ranks

Table 10 Comparison between PRS and popular population-based metaheuristics in the literature on classical benchmark functions

Function	PRS		MFO		GA		BA		PSO		GSA	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
F1	**1.32E-01**	9.58E-07	1.01E+03	3.05E+03	1.03E+03	5.79E+02	6.59E+04	7.51E+03	1.83E+04	3.01E+03	6.08E+02	4.64E+02
F2	**6.27E+00**	1.15E-02	3.19E+01	2.06E+01	2.47E+01	5.68E+00	2.71E+04	1.30E+09	3.58E+02	1.35E+03	2.27E+01	3.36E+00
F3	**8.50E+00**	5.12E-02	2.34E+04	1.41E+04	2.65E+04	3.44E+03	1.38E+05	4.72E+04	4.05E+04	8.21E+03	1.35E+05	4.86E+04
F4	**2.81E-01**	3.44E-08	7.00E+01	7.06E+00	5.17E+01	1.05E+01	8.51E+01	2.95E+00	4.39E+01	3.64E+00	7.87E+01	2.81E+00
F5	**2.56E+01**	5.67E+00	7.35E+03	2.26E+04	1.95E+04	1.31E-04	2.10E+08	4.17E+07	1.96E+07	6.25E+06	7.41E+02	7.81E+02
F6	**2.14E-01**	5.64E-05	2.68E+03	5.84E+03	9.01E+02	2.84E+02	6.69E+04	6.69E+04	1.87E+04	2.92E+03	3.08E+03	8.98E+02
F7	4.28E+02	1.22E-01	4.50E+00	9.21E+00	1.91E-01	1.50E-01	4.57E+01	7.82E+00	1.07E+01	3.05E+00	**1.12E-01**	3.76E-02
F8	–1.91E+03	5.58E-05	–8.48E+03	7.98E+02	**–1.26E+04**	4.51E+00	–2.33E+03	2.96E+02	–3.86E+03	2.49E+02	–2.35E+03	3.82E+02
F9	4.27E+02	1.60E+00	1.59E+02	3.21E+01	**9.04E+00**	4.58E+00	1.92E+02	3.56E+01	2.87E+02	1.95E+01	3.10E+01	1.36E+01
F10	**1.39E-04**	1.01E-04	1.74E+01	4.95E+00	1.36E+01	1.51E+00	1.92E+01	2.43E-01	1.75E+01	3.67E-01	3.74E+00	1.71E-01
F11	**1.47E-03**	1.20E-04	3.10E+01	5.94E+01	1.01E+01	2.43E+00	6.01E+02	5.50E+01	1.70E+02	3.17E+01	4.86E-01	4.97E-02
F12	8.25E+01	2.76E+01	2.46E+02	1.21E+03	4.77E+00	1.56E+00	4.71E+08	1.54E+08	1.51E+07	9.88E+06	**4.63E-01**	1.37E-01
F13	**2.37E-03**	7.99E-04	2.73E+07	1.04E+08	1.52E+01	4.52E+00	9.40E+08	1.67E+08	5.73E+07	2.68E+07	7.61E+00	1.22E+00
F14	**1.01E+00**	9.85E-08	2.74E+00	1.82E+00	9.98E−01	4.52E−16	1.27E+01	6.96E+00	1.39E+00	4.60E−01	3.80E+00	2.60E+00
F15	**2.90E-04**	1.20E-06	2.35E-03	4.92E-03	3.33E−02	2.70E−02	3.00E-02	3.33E-02	1.61E−03	4.60E−04	4.10E-03	3.20E-03
F16	−1.37E-01	1.57E-03	**–1.03E+00**	6.78E-16	1.54E+02	4.26E+01	3.48E+02	1.12E+02	1.73E+02	7.56E+01	−1.00E+00	4.00E-16
F17	2.47E+00	3.46E-03	**3.98E-01**	1.69E-16	3.05E+02	3.60E+01	4.53E+02	1.14E+02	3.79E+02	1.28E+02	4.00E-01	3.40E-16
F18	3.49E+02	1.88E+01	**3.00E+00**	0.00E+00	5.31E+01	7.90E+01	8.52E+01	1.51E+02	9.24E+01	1.37E+02	**3.00E+00**	2.20E-15
F19	–3.31E+00	5.41E-01	**–3.86E+00**	1.44E-03	−3.42E+00	3.03E−01	−3.84E+00	1.41E−01	**–3.86E+00**	1.24E−03	–3.60E+00	3.00E-01
F20	–2.36E+00	4.23E-03	**–3.24E+00**	6.42E-02	−1.61E+00	4.60E−01	–3.25E+00	5.89E-02	–3.11E+00	2.91E−02	–1.90E+00	5.40E-01
F21	**–9.95E+00**	1.00E+00	–6.89E+00	3.18E+00	−6.66E+00	3.73E+00	–4.27E+00	2.55E+00	–4.15E+00	9.20E-01	–5.10E+00	7.40E-03
F22	**–9.94E+00**	0.00E+00	–8.26E+00	3.08E+00	−5.58E+00	2.61E+00	–5.61E+00	3.02E+00	–6.01E+00	1.96E+00	–7.50E+00	2.70E+00
F23	–9.97E+00	0.00E+00	–7.66E+00	3.58E+00	−4.70E+00	3.26E+00	–3.97E+00	3.01E+00	–4.72E+00	1.74E+00	**–1.00E+01**	7.80E-01

The bold values corresponds to the best value obtained

Table 11 Results obtained on ranking-based Friedman's statistical test upon comparison with each population-based algorithm on classical benchmark functions. The ranks are based on the results shown in Table 10

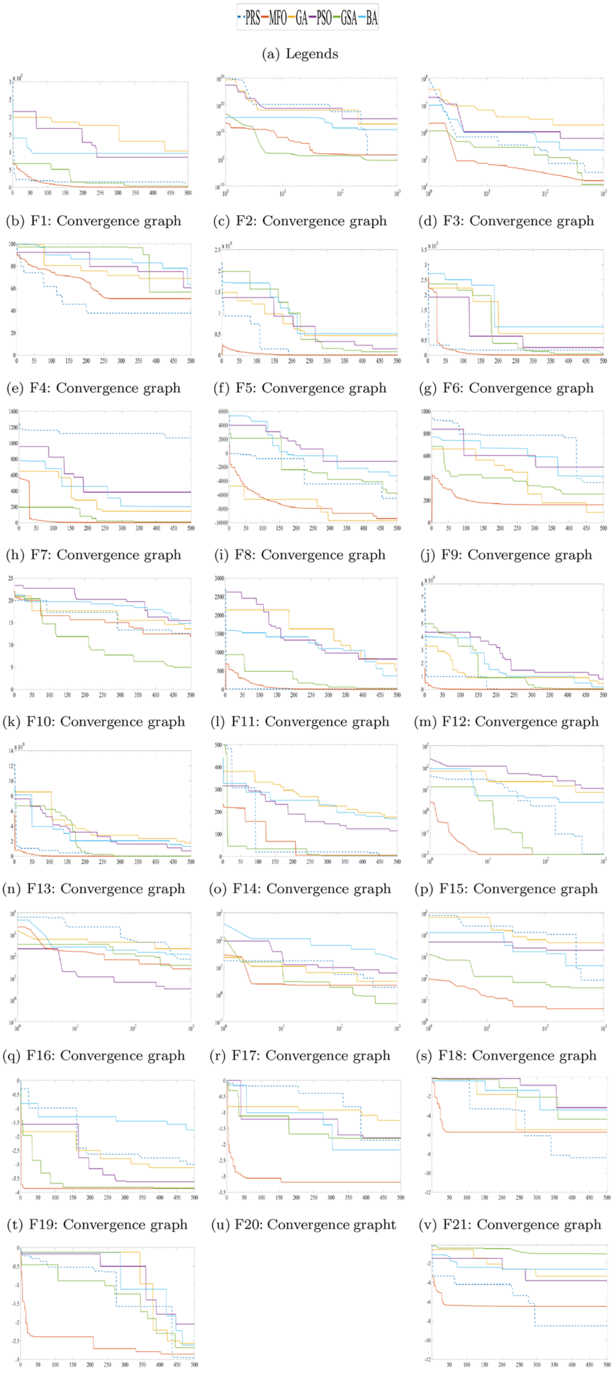
Function	PRS	MFO	GA	BA	PSO	GSA
F1	1	3	4	6	5	2
F2	1	4	3	6	5	2
F3	1	2	3	6	4	5
F4	1	4	3	6	2	5
F5	1	3	4	6	5	2
F6	1	3	2	6	5	4
F7	6	3	2	5	4	1
F8	6	2	1	5	3	4
F9	6	3	1	4	5	2
F10	1	4	3	6	5	2
F11	1	4	3	6	5	2
F12	3	4	2	6	5	1
F13	1	4	3	6	5	2
F14	2	4	1	6	3	5
F15	1	3	6	5	2	4
F16	3	1	4	6	5	2
F17	3	1	4	6	5	2
F18	6	1	3	4	5	1
F19	6	1	5	3	1	4
F20	4	2	6	1	3	5
F21	1	2	3	5	6	4
F22	1	2	6	5	4	3
F23	2	3	5	6	4	1
Average rank	2.57	2.74	3.35	5.26	4.17	2.83
Final rank	1	2	4	6	5	3

The bold values corresponds to the best value obtained

first, achieving the best fitness on 8 out of 29 functions, followed by GA, which achieves the best results on 6 functions, and we can see that in third place is the AHA algorithm and tied in fourth place are the MFO, AVOA and SNS algorithms tied with 3 best values. Further, that PRS can achieve superior results on complex composition functions (C22,C23,C27,C28) justifies its proposal as a robust optimization algorithm.

4.2.1 Execution time comparison for popular and recent population-based metaheuristic algorithms on CEC-2017 test suite

The analysis of execution times presented in Fig. 7 and Table 14 shows the performance ranking of recent and popular metaheuristic algorithms. The PRS consistently outperforms the others, showcasing its efficiency in solving complex optimization problems. Algorithm MFO, although competitive, exhibits slightly longer execution times, suggesting room for improvement. Algorithms AVOA and AHA, on the other



(a) Legends

(b) F1: Convergence graph

(c) F2: Convergence graph

(d) F3: Convergence graph

(e) F4: Convergence graph

(f) F5: Convergence graph

(g) F6: Convergence graph

(h) F7: Convergence graph

(i) F8: Convergence graph

(j) F9: Convergence graph

(k) F10: Convergence graph

(l) F11: Convergence graph

(m) F12: Convergence graph

(n) F13: Convergence graph

(o) F14: Convergence graph

(p) F15: Convergence graph

(q) F16: Convergence graph

(r) F17: Convergence graph

(s) F18: Convergence graph

(t) F19: Convergence graph

(u) F20: Convergence grapht

(v) F21: Convergence graph

(w) F22: Convergence graph

(x) F23: Convergence graph

Fig. 8 Convergence curves of optimization algorithms for functions **a**-**s** F1–F18, **t**-**x** F19–F23

Table 12 Comparison of PRS with popular metaheuristic algorithms on the CEC-2017 test suite [14]. The parameters of the comparative algorithms are tabulated in Table 9

Function	PRS		GA		PSO		BA		GSA	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
C1	6.86E+09	1.58E+09	1.32E+10	2.30E+09	3.06E+11	2.66E+10	1.15E+10	6.36E+09	6.77E+10	1.04E+11
C2	1.32E+04	4.26E+03	6.00E+03	**1.27E+02**	**1.73E+03**	1.34E+02	3.56E+04	2.96E+04	8.68E+04	6.77E+04
C3	8.64E+02	2.63E+02	**4.62E+02**	6.87E+01	1.33E+03	2.08E+02	6.19E+03	5.41E+03	2.22E+03	2.60E+02
C4	5.95E+02	1.20E+01	5.30E+02	1.46E+01	6.70E+02	1.60E+01	5.94E+02	2.65E+01	7.16E+02	2.49E+01
C5	6.53E+02	8.74E+00	6.16E+02	2.24E+01	2.25E+03	4.97E+02	6.23E+03	5.74E+01	6.06E+03	3.10E+03
C6	8.64E+02	3.61E+01	7.46E+02	7.16E+02	1.88E+03	1.99E+02	7.49E+02	2.24E+01	2.66E+03	3.54E+02
C7	8.78E+02	9.40E+00	**8.20E+02**	8.48E+01	4.55E+04	1.88E+04	8.48E+02	1.64E+01	1.23E+04	4.81E+04
C8	**1.98E+03**	3.98E+02	2.12E+04	1.12E+04	1.80E+04	4.11E+03	4.74E+03	1.75E+02	3.23E+04	8.90E+02
C9	**2.74E+03**	**1.39E+02**	2.81E+06	3.52E+06	6.25E+05	3.09E+07	3.66E+03	3.70E+02	2.50E+05	1.88E+05
C10	2.00E+03	3.82E+02	**1.34E+03**	1.12E+02	1.30E+06	1.10E+07	3.24E+04	4.27E+03	2.54E+07	4.70E+07
C11	1.82E+08	8.09E+07	7.55E+06	**5.37E+04**	1.11E+05	3.84E+06	7.77E+07	6.27E+07	3.32E+05	8.89E+06
C12	1.84E+06	1.65E+06	8.68E+05	**1.88E+04**	4.30E+06	6.03E+07	4.39E+06	6.56E+05	8.21E+07	5.46E+07
C13	3.83E+03	1.93E+03	**2.41E+03**	**7.16E+02**	6.39E+06	1.91E+08	2.21E+05	9.45E+03	4.97E+07	2.15E+08
C14	1.16E+04	6.99E+03	1.66E+04	1.49E+04	7.14E+03	2.22E+03	4.25E+04	1.46E+04	1.52E+04	2.80E+03
C15	**2.04E+03**	**1.05E+02**	2.91E+05	3.13E+05	2.00E+05	5.01E+06	6.36E+03	1.59E+02	7.34E+05	4.36E+06
C16	1.90E+03	**3.46E+01**	3.95E+07	6.52E+07	6.82E+06	7.87E+07	4.88E+03	5.63E+01	1.55E+08	1.52E+08
C17	6.04E+06	1.79E+07	2.08E+09	3.63E+09	1.30E+08	2.08E+09	6.38E+07	**1.36E+04**	8.52E+09	6.47E+09
C18	1.96E+05	4.02E+05	2.97E+05	6.50E+05	5.27E+03	**4.49E+02**	1.46E+06	1.87E+04	8.14E+03	5.04E+02
C19	**2.26E+03**	**5.33E+01**	1.09E+04	1.60E+04	3.60E+03	2.48E+02	3.96E+04	4.36E+02	4.25E+03	3.33E+02
C20	2.29E+03	2.65E+01	**2.22E+03**	6.75E+02	2.14E+04	4.11E+03	2.74E+04	5.78E+02	3.56E+04	6.91E+02
C21	2.74E+03	1.83E+02	4.32E+03	1.42E+02	5.32E+03	4.29E+02	2.14E+04	1.12E+02	4.95E+03	2.12E+02
C22	**2.71E+03**	1.49E+01	6.64E+03	5.49E+02	5.05E+03	6.22E+02	5.63E+04	1.46E+01	6.76E+03	4.49E+02
C23	**2.78E+03**	5.87E+01	5.59E+03	6.53E+01	3.68E+03	4.20E+03	2.97E+05	6.87E+01	4.97E+04	3.98E+04

Table 12 (continued)

Function	PRS		GA		PSO		BA		GSA	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
C24	3.29E+03	1.25E+02	4.08E+04	3.05E+04	2.21E+04	4.70E+03	9.36E+04	6.59E+01	4.08E+04	5.41E+03
C25	3.71E+03	**1.92E+02**	1.84E+04	2.98E+04	3.44E+03	1.08E+03	7.17E+04	4.39E+02	8.12E+03	8.62E+02
C26	3.19E+03	1.33E+02	2.26E+04	1.50E+04	3.93E+04	3.72E+03	4.20E+03	7.70E+01	3.57E+04	1.29E+04
C27	**3.53E+03**	1.31E+02	1.03E+05	1.26E+05	1.67E+04	1.65E+04	5.27E+04	4.52E+02	3.18E+05	5.90E+05
C28	**3.40E+03**	6.83E+01	6.58E+05	5.31E+05	4.22E+05	7.79E+04	4.63E+04	7.97E+01	1.37E+06	1.49E+06
C29	6.14E+06	5.71E+06	**9.20E+05**	**4.64E+02**	1.39E+06	2.75E+04	7.42E+06	2.89E+06	2.14E+06	2.01E+06
Function	MFO		AHA		RSA		AVOA		SNS	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
C1	1.66E+11	5.49E+10	4.48E+09	2.07E+08	1.10E+10	3.77E+09	3.40E+09	2.54E+09	**3.66E+08**	**3.95E+03**
C2	3.47E+04	2.55E+04	6.09E+04	7.66E+04	8.99E+03	2.46E+03	3.04E+04	3.05E+04	3.00E+04	5.44E+02
C3	1.76E+03	1.91E+02	8.18E+02	2.73E+02	1.28E+03	4.82E+02	6.96E+02	2.41E+02	9.07E+02	**1.69E+01**
C4	6.72E+02	1.20E+01	**5.30E+02**	**1.11E+01**	5.76E+02	1.60E+01	5.61E+02	2.42E+01	5.31E+02	1.33E+01
C5	5.94E+03	1.19E+03	**6.01E+02**	**1.56E+00**	6.45E+02	8.43E+00	6.46E+02	1.24E+01	6.07E+02	6.95E+00
C6	2.12E+03	1.59E+02	**7.37E+02**	1.46E+01	8.02E+02	**9.59E+00**	8.43E+02	5.42E+01	7.53E+02	1.28E+01
C7	4.89E+04	2.24E+04	8.99E+02	1.24E+01	8.51E+02	**6.75E+00**	8.48E+02	1.69E+01	8.25E+02	7.57E+00
C8	2.01E+04	2.28E+03	9.72E+03	1.99E+02	2.44E+03	1.54E+02	2.97E+03	6.60E+02	1.02E+04	**1.35E+02**
C9	7.55E+06	3.50E+08	2.93E+03	3.22E+02	3.51E+03	2.36E+02	2.81E+03	3.88E+02	1.71E+04	2.72E+02
C10	4.04E+06	7.13E+07	2.16E+03	**9.90E+01**	3.92E+03	1.98E+03	5.25E+03	7.31E+03	1.13E+04	1.22E+02
C11	**9.64E+04**	3.39E+06	1.51E+08	2.88E+07	3.41E+08	2.37E+08	1.50E+07	3.27E+07	1.66E+05	2.34E+05
C12	6.53E+06	6.37E+07	1.33E+06	1.26E+05	3.96E+07	3.87E+07	**6.08E+04**	1.47E+05	1.07E+05	8.56E+04
C13	1.59E+07	2.32E+08	3.98E+04	5.08E+03	5.92E+03	6.72E+03	1.30E+04	2.17E+04	1.54E+04	1.20E+03
C14	**6.92E+03**	2.27E+03	9.17E+04	8.34E+03	1.09E+04	5.68E+03	4.07E+04	5.85E+04	2.62E+04	**1.13E+03**

Table 12 (continued)

Function	MFO		AHA		RSA		AVOA		SNS	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
C15	4.85E+05	1.18E+07	2.77E+03	1.30E+02	2.08E+03	1.49E+02	2.11E+03	1.94E+02	2.77E+03	1.25E+02
C16	1.03E+07	4.35E+07	1.96E+03	3.36E+02	**1.84E+03**	6.23E+01	2.04E+03	1.49E+02	1.92E+03	4.08E+02
C17	4.36E+08	3.42E+09	2.27E+06	1.30E+05	7.59E+07	1.48E+08	**3.17E+04**	3.54E+04	1.25E+05	9.46E+04
C18	**4.66E+03**	5.03E+02	1.21E+04	1.13E+04	4.86E+06	1.66E+07	1.41E+05	6.02E+05	6.71E+03	5.51E+03
C19	3.78E+03	1.74E+02	2.38E+03	6.76E+01	2.29E+03	6.45E+01	2.31E+03	1.01E+02	2.38E+03	4.96E+02
C20	2.29E+04	2.53E+03	2.31E+03	4.40E+01	2.32E+03	6.29E+01	2.36E+03	**2.49E+01**	2.23E+03	5.84E+01
C21	4.15E+03	1.95E+02	**2.30E+03**	**1.38E+01**	3.09E+03	3.02E+02	2.80E+03	2.36E+02	2.31E+03	5.53E+01
C22	4.31E+03	4.64E+02	2.83E+03	1.23E+01	3.70E+03	**1.22E+01**	2.72E+03	4.92E+01	2.74E+03	1.46E+01
C23	1.42E+04	1.30E+04	2.86E+03	1.14E+02	2.86E+03	**4.17E+01**	2.86E+03	5.17E+01	2.88E+03	1.04E+02
C24	2.48E+04	3.51E+03	3.94E+03	2.72E+02	3.37E+03	1.50E+02	3.14E+03	1.93E+02	**2.91E+03**	**6.46E+01**
C25	4.07E+03	6.56E+02	3.99E+03	4.63E+02	4.09E+03	3.07E+02	3.92E+03	4.44E+02	**3.05E+03**	2.77E+02
C26	2.21E+04	3.77E+03	3.39E+03	2.89E+02	3.19E+03	8.96E+01	**3.18E+03**	**2.68E+01**	3.80E+03	1.25E+02
C27	1.10E+05	1.87E+06	3.70E+03	8.72E+02	3.77E+03	1.32E+02	3.72E+03	**1.08E+02**	3.62E+03	1.24E+02
C28	1.04E+06	2.63E+06	3.54E+03	**6.58E+01**	4.36E+03	8.59E+01	3.61E+03	2.02E+02	3.53E+03	6.41E+02
C29	1.47E+06	1.10E+05	6.19E+06	3.59E+05	3.11E+06	3.83E+06	2.58E+06	5.03E+06	1.21E+06	2.99E+05

The bold values corresponds to the best value obtained

Table 13 Results obtained on ranking-based Friedman's statistical test upon comparison with popular metaheuristic algorithms on the CEC-2017 test suite [14]. The ranks are based on the results shown in Table 12

Function	PRS	GA	PSO	BA	GSA	MFO	AHA	RSA	AVOA	SNS
C1	1	2	5	3	9	8	6	10	7	4
C2	5	6	3	9	7	10	8	1	2	4
C3	5	2	6	3	8	9	10	7	1	4
C4	3	4	5	1.5	9	10	6	8	1.5	7
C5	2	5	4	1	8	9	10	7	3	6
C6	4	6	5	1	9	10	3	8	2	7
C7	2	3.5	5	7	10	8	3.5	9	1	6
C8	6	3	2	5	8	10	4	7	9	1
C9	6	2	4	3	10	7	5	8	9	1
C10	6	5	4	3	9	10	7	8	1	2
C11	3	6	10	8	1	4	7	2	5	9
C12	2	1	9	4	8	10	7	6	3	5
C13	5	4	3	6	9	10	7	8	1	2
C14	7	8	3	10	1	5	9	2	6	4
C15	4.5	3	2	4.5	9	10	6	7	8	1
C16	3	5	1	4	8	10	6	7	9	2
C17	2	1	6	3	8	10	5	7	9	4
C18	3	6	10	5	1	4	9	2	8	7
C19	4.5	3	2	4.5	7	8	10	6	9	1
C20	2	6	5	4	8	10	9	7	1	3
C21	2	4	5	1	6	8	10	9	7	3
C22	3	2	5	4	6	9	10	7	8	1
C23	5	3	3	3	8	9	10	6	7	1
C24	1	2	4	5	7	8.5	10	6	8.5	3
C25	1	4	7	5	6	8	10	2	9	3
C26	5	1	2.5	4	7	9	6	10	8	2.5
C27	2	4	5	3	9	10	7	6	8	1
C28	2	4	5	3	9	10	6	7	8	1
C29	2	6	7	9	4	5	10	3	1	8
Average rank	3.41	3.84	4.74	4.36	7.21	8.57	7.47	6.31	5.52	3.57
Final rank	**1**	**3**	**5**	**4**	**8**	**10**	**9**	**7**	**6**	**2**

The bold values corresponds to the best value obtained

hand, demonstrate similar execution speeds, indicating comparable performance levels. These results shed light on the computational efficiency of each algorithm, guiding researchers and practitioners in selecting the most suitable approach for solving real-world problems. Moreover, the variations in execution times emphasize the need for further optimization and fine-tuning, ensuring these algorithms can be applied effectively across diverse domains.

4.3 Statistical analysis

The Wilcoxon rank-sum statistical test [96] is performed to investigate the robustness of the proposed PRS algorithm. Wilcoxon rank-sum test is a non-parametric statistical test that determines whether two models are statistically similar. A null hypothesis is assumed, which says that two models are statistically similar, and to reject this null hypothesis, the probability (p-value) needs to be lower than 5% or 0.05 (also known as the significance level). The p-values obtained by this statistical test on comparing the proposed algorithms with (a) single-solution-based algorithms on the classical benchmark functions are shown in Table 15, (b) population-based metaheuristics on the classical benchmark functions are shown in Table 16 and (c) popular metaheuristics on the CEC test suite is shown in Table 17. As can be seen from the tables, the p-values are lower than 0.05 for the majority of the cases. Thus, we can conclude that the proposed model is statistically dissimilar to the models proposed in the literature. Further, the competitiveness of the proposed PRS algorithm can be justified by the results obtained by using the ranking-based Friedman test, the results of which are highlighted in Tables 7, 11, 13 and 20.

4.4 Additional test: application of PRS in engineering design problems

Further, as additional tests to validate the multi-faceted applicability of PRS, we apply our proposed optimizer to five classical engineering design problems, namely (1) pressure vessel design, (2) tension/compression spring design, (3) welded beam design, (4) gear train design, and (5) closed coil helical spring design. These are real-world problems and are challenging due to the presence of a large number of independent attributes and the presence of multiple local optima. Thus, evaluating such problems tests an optimization algorithm's robustness and suitability. When working with parameter optimization in engineering problems, it is necessary to use a method to handle the constraints; in this study, we handle the constraints through penalization; in this case, the method modifies the fitness function to penalize the solutions that violate the constraints and a high value is assigned to the invalid solutions to disapprove their selection and restrict that numerical range.

4.4.1 Pressure vessel design (E1)

The main objective of this problem is to minimize the overall expense of designing a pressure vessel, which comprises manufacturing, welding, and material costs. Four design parameters are used for formulating the design objective: the thickness of head (T_h), the inner radius of head (R), the thickness of cylinder (T_c), and the length of the cylindrical section excluding the head (L). A solution to this problem may be modeled as $x = (x_1, x_2, x_3, x_4) = (T_c, T_h, R, L)$. A schematic diagram describing the system is illustrated in Fig. 9, while Eq. 13 shows the full objective of the function along with the associated inequality constraints.

Table 14 Comparison of execution times in population-based algorithms on CEC2017 test suit

Function		PRS	GA	PSO	BA	GSA	MFO	AHA	RSA	AVOA	SNS
C1	$\bar{x}(time)$	6.05	9.09	8.93	6.28	10.03	6.51	9.98	13.85	7.36	8.48
C2	$\bar{x}(time)$	5.77	9.37	8.93	6.41	8.87	6.17	7.34	13.42	6.99	8.83
C3	$\bar{x}(time)$	5.91	9.50	8.82	6.58	9.34	5.96	7.24	13.68	7.36	8.68
C4	$\bar{x}(time)$	6.11	9.27	8.97	6.49	9.05	6.17	7.19	13.64	7.18	9.00
C5	$\bar{x}(time)$	6.39	9.39	9.30	6.89	9.27	6.48	7.94	14.07	7.60	9.60
C6	$\bar{x}(time)$	6.39	9.34	9.10	6.80	9.19	6.63	7.43	13.89	7.90	8.69
C7	$\bar{x}(time)$	6.33	9.47	9.22	6.91	9.36	6.51	7.44	14.26	7.61	8.78
C8	$\bar{x}(time)$	6.53	9.46	9.71	7.04	9.77	6.58	7.54	14.18	7.56	9.14
C9	$\bar{x}(time)$	6.58	9.57	9.72	7.17	9.50	6.63	7.50	14.30	7.66	9.09
C10	$\bar{x}(time)$	6.53	9.62	9.39	7.33	9.45	6.70	8.26	13.99	7.73	9.11
C11	$\bar{x}(time)$	6.83	10.18	9.76	7.61	9.55	6.76	8.08	14.27	7.96	9.05
C12	$\bar{x}(time)$	6.85	9.92	10.31	8.03	9.83	7.03	8.05	14.60	7.95	9.17
C13	$\bar{x}(time)$	6.90	10.17	10.22	8.42	10.18	7.39	8.20	14.62	8.34	9.40
C14	$\bar{x}(time)$	6.95	10.01	10.27	7.80	10.09	7.55	8.03	14.88	8.33	9.30
C15	$\bar{x}(time)$	6.97	10.30	10.01	7.88	10.19	7.47	8.58	14.67	8.97	9.39
C16	$\bar{x}(time)$	7.33	10.32	10.26	7.81	10.02	7.58	9.13	15.06	8.71	9.72
C17	$\bar{x}(time)$	7.25	10.64	9.98	7.63	10.08	7.34	9.44	14.88	8.82	9.67
C18	$\bar{x}(time)$	8.48	11.68	11.27	8.72	10.89	8.46	10.40	16.02	10.36	10.98
C19	$\bar{x}(time)$	7.57	11.01	10.58	8.11	10.82	7.97	8.79	15.48	9.30	10.20
C20	$\bar{x}(time)$	7.73	10.77	10.59	8.16	11.46	8.02	9.40	15.97	9.24	10.14
C21	$\bar{x}(time)$	7.89	11.04	10.73	8.44	11.58	7.80	9.88	16.36	9.27	10.56
C22	$\bar{x}(time)$	7.88	11.39	10.93	8.53	11.64	8.01	9.75	16.49	9.65	10.41
C23	$\bar{x}(time)$	8.03	11.59	11.14	8.63	11.60	8.06	9.42	16.41	10.09	10.46
C24	$\bar{x}(time)$	8.03	11.08	11.08	8.83	12.34	8.12	10.37	16.07	9.19	10.44
C25	$\bar{x}(time)$	8.41	11.42	11.37	8.94	12.04	8.89	9.43	16.02	9.37	10.87
C26	$\bar{x}(time)$	9.00	11.57	11.75	9.12	11.76	9.05	9.47	16.45	9.58	10.96
C27	$\bar{x}(time)$	8.72	11.41	11.70	8.95	11.57	9.09	9.60	16.10	9.43	11.05
C28	$\bar{x}(time)$	8.61	11.95	11.56	9.12	12.18	8.99	9.77	16.69	9.71	11.04
C29	$\bar{x}(time)$	9.24	12.76	12.44	10.21	12.23	9.56	10.84	17.35	10.80	12.34
	Avg Rank	1.10	8.21	7.55	3.05	7.97	1.97	4.69	10.00	4.31	6.17
	Total Rank	**1**	**9**	**7**	**3**	**8**	**2**	**5**	**10**	**4**	**6**

The bold values corresponds to the best value obtained

Table 15 Results (i.e. p–values) achieved by using Wilcoxon rank-sum statistical test of PRS versus other single-solution-based methods on classical benchmark functions

Function	PRS vs. VNS	PRS vs. TS	PRS vs. SA	PRS vs. AβHC	PRS vs. βHC	PRS vs. HC
F1	0.0492	0.0104	0.012	0.0348	0.0173	0.0163
F2	0.0354	0.0315	0.0033	0.00112	0.0126	0.0308
F3	0.0105	0.35	0.0153	0.0234	0.0314	0.026
F4	0.0394	0.0322	0.00809	0.0427	0.0404	0.0352
F5	0.0127	0.00654	0.0238	0.0424	0.0102	0.0373
F6	0.0347	0.0407	0.00451	0.0444	0.0215	0.0231
F7	0.0103	0.00166	0.0374	0.000504	0.0127	0.0481
F8	0.0181	0.0361	0.0478	0.104	0.0434	0.0483
F9	0.0154	0.00723	0.0351	0.0071	0.0184	0.0211
F10	0.0152	0.0296	0.00849	0.02	0.0439	0.0469
F11	0.0000328	0.0354	0.0276	0.0477	0.014	0.0432
F12	0.0295	0.035	0.037	0.0321	0.0305	0.00996
F13	0.016	0.023	0.0151	0.0226	0.0337	0.00913
F14	0.00365	0.0489	0.018	0.035	0.0417	0.0323
F15	0.00737	0.0425	0.05	0.0476	0.00584	0.0322
F16	0.0142	0.00888	0.0215	0.00614	0.0264	0.0255
F17	0.0222	0.00489	0.0473	0.031	0.00462	0.00251
F18	0.00758	0.205	0.401	0.0451	0.0108	0.042
F19	0.0202	0.0349	0.0121	0.00725	0.0397	0.0219
F20	0.0405	0.0112	0.035	0.0484	0.00832	0.00546
F21	0.0203	0.024	0.0287	0.0324	0.331	0.00578
F22	0.0138	0.0053	0.0339	0.0264	0.0146	0.0235
F23	0.0137	0.0404	0.0252	0.008	0.0373	0.0218

$$
\begin{aligned}
&\text{Minimize:}\\
&E_1(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 19.84x_1^2x_3 + 3.1661x_1^2x_4\\
&\\
&\text{Subject to}\\
&0.0193x_3 - x_1 \le 0\\
&0.00954x_3 - x_2 \le 0\\
&1296000 - \frac{4}{3}\pi x_3^3 - \pi x_3^2 x_4 \le 0\\
&x_4 - 240 \le 0\\
&\\
&\text{such that}\\
&x_1, x_2 \in [0.0625, 6.1875]\\
&x_3, x_4 \in [10, 200]
\end{aligned}
\tag{13}
$$

Table 16 Results (i.e. p–values) obtained by using Wilcoxon rank-sum statistical test of PRS versus other population solution-based methods on classical benchmark functions

Function	PRS vs. MFO	PRS vs. GA	PRS vs. BA	PRS vs. PSO	PRS vs. GSA
F1	0.0428	0.00872	0.0195	0.0301	0.0033
F2	0.0266	0.0132	0.0132	0.0437	0.000597
F3	0.0479	0.0431	0.015	0.0371	0.000029
F4	0.0259	0.0112	0.0181	0.0275	0.0477
F5	0.027	0.0376	0.034	0.0393	0.00696
F6	0.0221	0.00698	0.0194	0.000173	0.0165
F7	0.00402	0.00728	0.0372	0.0371	0.0373
F8	0.00666	0.0499	0.0259	0.00655	0.00643
F9	0.0183	0.0318	0.0231	0.00394	0.0191
F10	0.0447	0.023	0.0175	0.0143	0.0293
F11	0.0427	0.0457	0.0437	0.0379	0.0228
F12	0.045	0.0448	0.0252	0.00901	0.00683
F13	0.0171	0.0398	0.00155	0.0298	0.0166
F14	0.00594	0.0391	0.0444	0.00881	0.0231
F15	0.015	0.0286	0.000754	0.00536	0.0375
F16	0.000544	0.0358	0.0414	0.0151	0.0338
F17	0.0311	0.0314	0.012	0.0286	0.0361
F18	0.03	0.0339	0.0204	0.0459	0.0397
F19	0.00278	0.0169	0.00678	0.0319	0.0495
F20	0.0302	0.0196	0.0294	0.0244	0.0428
F21	0.0486	0.0473	0.0332	0.0172	0.041
F22	0.0325	0.0062	0.0429	0.0267	0.00897
F23	0.0254	0.0254	0.0471	0.0029	0.0426

4.4.2 Tension/compression spring design (E2)

This problem aims at finding the minimum weight of the tension/compression spring, given its three design parameters: wire diameter (d), mean coil diameter (D), and the number of active coils (N). The schematic diagram representing this engineering design problem is shown in Fig. 10. Mathematically, a solution to this problem can be represented as $x = (x_1, x_2, x_3) = (d, D, N)$. The objective expression and the inequality constraints imposed on the parametersare provided in Eq. 14.

Table 17 Results (i.e. $p-$values) attained by using Wilcoxon rank-sum statistical test of PRS versus other popular metaheuristics on CEC-2017 test suite

Function	PRS vs. GA	PRS vs. PSO	PRS vs. BA	PRS vs. GSA	PRS vs. MFO	PRS vs. AHA	PRS vs. RSA	PRS vs. AVOA	PRS vs. SNS
C1	0.0297	0.00792	0.00947	0.0497	0.0287	0.0224	0.00938	0.00913	0.0281
C2	0.0131	0.00535	0.0174	0.0403	0.0253	0.0489	0.0176	0.0187	0.0139
C3	0.0488	0.0406	0.0466	0.00103	0.02	0.0156	0.0401	0.0412	0.0473
C4	0.0282	0.02	0.00219	0.0342	0.0225	0.0419	0.00201	0.00257	0.0288
C5	0.0206	0.0281	0.00628	0.041	0.0213	0.0032	0.00643	0.00592	0.0217
C6	0.0322	0.0172	0.0405	0.0488	0.0259	0.0375	0.0407	0.0419	0.0315
C7	0.043	0.0174	0.0397	0.0101	0.0115	0.0447	0.0402	0.0401	0.0432
C8	0.00259	0.0159	0.0000838	0.00154	0.0088	0.0243	0.000092	0.00012	0.00245
C9	0.000499	0.022	0.00316	0.0265	0.0454	0.0415	0.00314	0.00307	0.000552
C10	0.0482	0.0172	0.0278	0.0226	0.0235	0.0372	0.0276	0.0272	0.0483
C11	0.00693	0.00201	0.0224	0.0268	0.0453	0.0192	0.0222	0.0235	0.00691
C12	0.0175	0.0447	0.0216	0.00962	0.0219	0.0421	0.0217	0.0219	0.0322
C13	0.0163	0.0297	0.0396	0.000532	0.00165	0.0398	0.0395	0.0396	0.0167
C14	0.00994	0.03	0.0208	0.0236	0.041	0.0278	0.0209	0.0201	0.00959
C15	0.0000827	0.0449	0.00827	0.0476	0.00973	0.0289	0.00835	0.00878	0.0232
C16	0.00712	0.0117	0.0226	0.0161	0.0161	0.0336	0.0225	0.0221	0.0471
C17	0.0291	0.0466	0.00747	0.0414	0.00434	0.00486	0.00754	0.00732	0.0167
C18	0.0494	0.0395	0.0132	0.0017	0.0367	0.0361	0.0133	0.0139	0.0421
C19	0.0372	0.0318	0.00101	0.0361	0.0132	0.0137	0.00104	0.00123	0.0023
C20	0.0132	0.033	0.0405	0.0438	0.017	0.0175	0.0404	0.0409	0.0369
C21	0.00521	0.0412	0.0206	0.025	0.0235	0.0232	0.0203	0.0215	0.0442
C22	0.0205	0.0457	0.0459	0.0402	0.044	0.0435	0.0458	0.0465	0.0255
C23	0.0392	0.00555	0.00193	0.0355	0.0239	0.0244	0.00198	0.00203	0.0408
C24	0.0123	0.00689	0.0248	0.0199	0.0169	0.0174	0.0249	0.0242	0.0363
C25	0.0402	0.000249	0.0155	0.0404	0.0393	0.0387	0.0158	0.0151	0.0203

Table 17 (continued)

Function	PRS vs. GA	PRS vs. PSO	PRS vs. BA	PRS vs. GSA	PRS vs. MFO	PRS vs. AHA	PRS vs. RSA	PRS vs. AVOA	PRS vs. SNS
C26	0.0339	0.00615	0.00802	0.0382	0.00666	0.00612	0.008	0.00843	0.0408
C27	0.00698	0.0341	0.00866	0.0262	0.0341	0.0335	0.0087	0.00877	0.0387
C28	0.0251	0.0443	0.021	0.0275	0.0202	0.0196	0.0211	0.0207	0.0259
C29	0.000568	0.00673	0.0494	0.0318	0.0246	0.024	0.0496	0.0501	0.0271

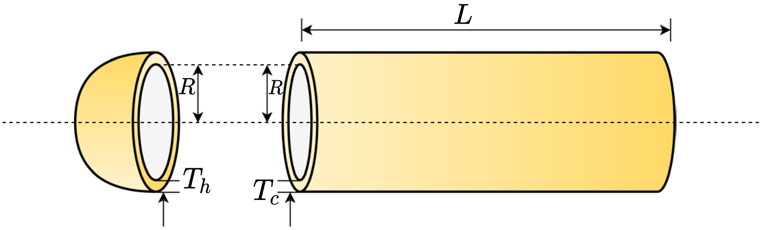


Fig. 9 Schematic diagram representing the pressure vessel design problem

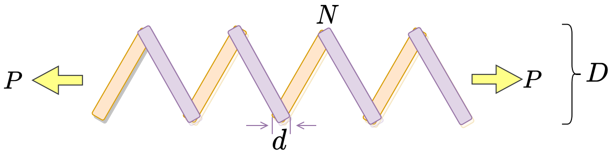


Fig. 10 Schematic diagram representing the tension/compression spring design problem

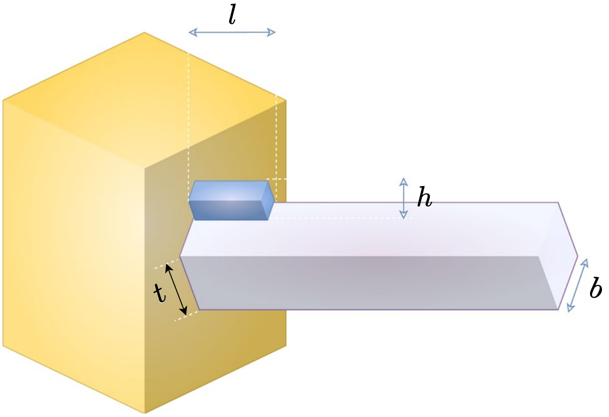


Fig. 11 Schematic diagram representing the welded beam design problem

$$
\begin{aligned}
&\text{Minimize:}\\
&E_2(x) = (x_3 + 2)x_2x_1^2\\
\\
&\text{Subject to:}\\
&1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0\\
&\frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0\\
&1 - \frac{140.45x_1}{x_2^2x_3} \leq 0\\
&\frac{x_1 + x_2}{1.5} - 1 \leq 0\\
\\
&\text{such that}\\
&x_1 \in [0.05, 2]\\
&x_2 \in [0.25, 1.30]\\
&x_3 \in [2, 15]
\end{aligned}
\tag{14}
$$

4.4.3 Welded beam design (E3)

The objective of the welded beam design problem is to find the minimum fabrication cost, given four independent variables: the width of the weld (h), length of the attached

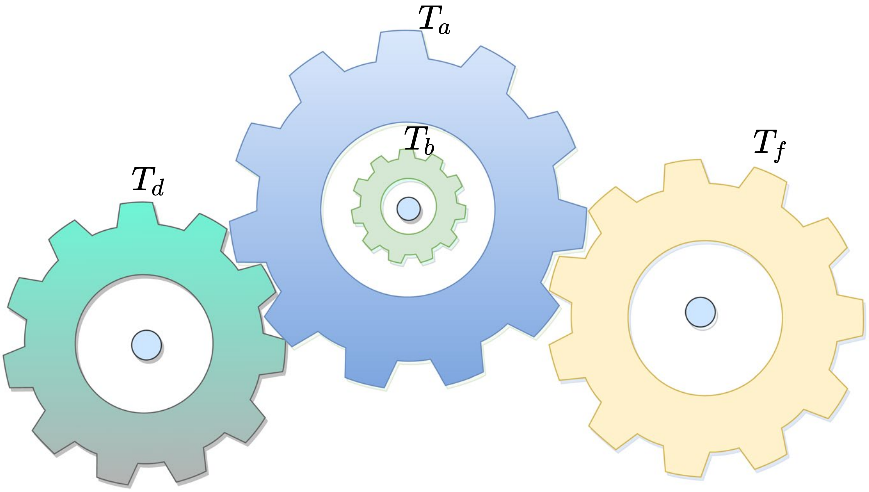


Fig. 12 Schematic diagram representing the gear train design problem

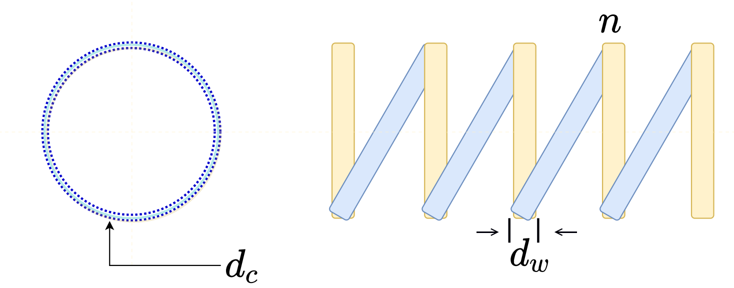


Fig. 13 Schematic diagram representing the closed coil helical spring design problem

portion of bar (l), the height (t) and thickness of the bar (b). Thus, a solution to this problem can be mathematically denoted by $x = (x_1, x_2, x_3, x_4) = (h, l, t, b)$. The objective and the numerical constraints imposed on this problem are described in Eq. 15. For a lucid explanation, a graphical illustration of the said problem is provided in Fig. 11.

$$\begin{aligned}
&\text{Minimize:}\\
&E_3(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)\\
\\
&\text{Subject to}\\
&\tau(x) - \tau_{max} \le 0\\
&\sigma(x) - \sigma_{max} \le 0\\
&\delta(x) - \delta_{max} \le 0\\
&x_1 - x_4 \le 0\\
&P - P_c(z) \le 0\\
&0.125 - x_1 \le 0\\
&1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) - 5 \le 0\\
\\
&\text{such that}\\
&x_1 \in [0.05, 2.00]\\
&x_2 \in [0.25, 1.30]\\
&x_3 \in [2.00, 15.00]\\
&x_4 \in [0.10, 2.00
\end{aligned} \tag{15}$$

4.4.4 Gear train design (E4)

In this problem, the objective is to minimize the cost of gear ratio of a gear train, the decision parameters being teeth of each gear wheel T_a, T_b, T_d and T_f. Mathematically its solution is of the form $x = (x_!, x_2, x_3, x_4) = (T_a, T_b, T_d, T_f)$. The objective of the problem is given in Eq. 16. Note that in this problem, no inequality constraint

Table 18 Comparison between PRS and other popular metaheuristics on standard engineering design problems

Problem		PRS	GWO	GA	PSO	AβHCSCA
E1	Avg	**8.81E+03**	**8.81E+03**	8.90E+03	4.12E+04	**8.81E+03**
	Std	0.00E+00	0.00E+00	6.35E+01	1.98E+04	0.00E+00
E2	Avg	7.27E-03	**7.27E-03**	1.16E-02	1.00E+300	**7.27E-03**
	Std	1.94E-06	0.00E+00	2.35E-03	0.00E+00	0.00E+00
E3	Avg	3.33E+00	**3.33E+00**	3.47E+00	6.00E+299	3.33E+00
	Std	1.08E-03	1.20E-08	2.66E-02	6.55E+04	8.83E-05
E4	Avg	1.46E-10	**1.49E-11**	6.81E-11	1.18E-09	2.97E-10
	Std	2.87E-10	1.12E-11	6.30E-11	1.14E-09	5.43E-10
E5	Avg	4.65E+01	**4.58E+01**	4.67E+01	4.59E+01	4.59E+01
	Std	1.39E+01	0.00E+00	1.38E+01	1.18E-01	0.00E+00

The problems are described in Sect. 4.4

The bold values corresponds to the best value obtained

has been imposed. For ease of understanding, a visual representation of the problem is provided in Fig. 12.

$$\begin{aligned}&\text{Minimize:}\\&E_4(x) = \left(\frac{1}{6.931} - \frac{x_2x_3}{x_1x_4}\right)^2\\&\text{such that}\\&x_1, x_2, x_3, x_4 \in [12, 60]\end{aligned} \tag{16}$$

4.4.5 Closed coil helical spring design (E5)

This engineering design problem aims at minimizing the volume of a closed coil helical spring. The objective, defined in Eq. 17, is modeled using three independent attributes: diameter of the wire used (d_w), coil diameter (d_c) and the number of active coils (n). Thus, its solution can be modeled mathematically as $x = (x_1, x_2, x_3) = (d_w, d_c, n)$. The problem is visually depicted in Fig. 13.

$$\begin{aligned}&\text{Minimize:}\\&E_5(x) = \frac{\pi^2}{4}(x_3 + 2)x_2x_1^2\\&\text{such that}\\&x_1 \in [0.508, 1.016]\\&x_2 \in [1.270, 7.620]\\&x_3 \in [15.00, 25.00]\end{aligned} \tag{17}$$

Table 19 Parameters obtained by the algorithm in engineering problems

E1. pressurevessel design	PRS	GWO	GA	PSO	AβHCSCA
T_s	1.495	0.908	1.212	3.165	0.983
T_h	1.310	1.047	1.276	0.833	0.832
R	50.612	50.733	44.052	76.234	50.852
L	90.926	90.636	155.271	48.062	91.184
Runtime	5.870	5.976	6.518	6.792	6.554
E2. tension/compression spring design					
d	0.050	0.050	0.051	1.215	0.050
D	0.480	0.480	0.471	0.813	0.480
N	4.061	4.056	6.471	2.979	4.056
Runtime	5.130	5.268	5.635	6.025	5.703
E3. welded beam design					
h	0.620	0.613	0.709	0.313	0.615
l	1.300	1.300	1.293	0.283	1.300
t	6.084	6.177	5.158	13.705	6.152
b	0.620	0.613	0.726	0.913	0.615
Runtime	5.308	5.454	7.197	6.318	5.892
E4. gear train design					
T_a	19.785	19.494	16.051	12.000	15.775
T_b	12.848	16.175	19.276	18.445	18.924
T_d	34.399	43.165	49.207	33.629	48.901
T_f	52.549	49.338	43.134	43.660	42.788
Runtime	5.017	5.218	7.451	6.093	5.632
E5. closed coil helical spring design					
d_w	0.668	0.674	0.670	0.674	0.673
d_c	2.311	2.405	2.340	2.403	2.392
n	16.286	15.004	16.002	15.052	15.192
Runtime	5.812	4.996	5.184	6.405	5.987

Table 20 Results obtained on ranking-based Friedman's statistical test compared with popular metaheuristics on the standard engineering design problems used in this study

Problem	PRS	GWO	GA	PSO	AβHCSCA
E1	2	2	4	5	2
E2	2	2	4	5	2
E3	2	2	4	5	2
E4	3	1	2	5	4
E5	4	1	5	2.5	2.5
Average rank	2.6	1.6	3.8	4.5	2.5
Final rank	3	1	4	5	2

The ranks are based on the results shown in Table 18

The bold values corresponds to the best value obtained

To visualize the solution vectors in the design of engineering problems, Table 19 is created, which contains the value of the parameters obtained in each of the algorithms tested in this section.

The proposed PRS algorithm is evaluated on the aforementioned problems and compared with other state-of-the-art metaheuristic algorithms such as GWO [71], GA [53], PSO [61] and AβHCSCA [21]. The number of iterations for each algorithm is set to 1000, while the mean and SD fitness values obtained across 30 independent runs have been reported. Table 18 tabulates the comparative results on each engineering design problem. At the same time, the average ranks obtained by each algorithm from the comparison are put in Table 20. It can be observed that PRS achieves the best results on exactly three out of five functions, while for the remaining two functions, it yields values very close to the global optimum. Thus, this comparative study indicates the robustness of PRS in effectively optimizing challenging real-world engineering problems.

5 Conclusion

In this research, we have proposed a novel physics-inspired single-solution-based metaheuristic, called PRS algorithm, modeling on an optimization paradigm obtained from the refraction of light at various angles of incidence by a prism. The novelty, as well as the rigorous scientific backbone of our approach, has been adequately elucidated, something found wanting in several proposed algorithms as criticized in [92]. Further, we have performed extensive experiments to evaluate our algorithm on a plethora of numerical optimization functions, analyzed the convergence behaviors, and compared the results to several single-solution and population-based metaheuristics. The results obtained by PRS are highly competitive, often outperforming popular optimization algorithms. We also applied the proposed algorithm to some standard engineering design problems. The results are promising compared to other state-of-the-art algorithms on the same task.

Since metaheuristics can be considered black-box optimizers, the proposed algorithm may be extended to various domains of engineering optimization and combinatorial challenging problems, such as feature selection, traveling salesman problem, multi-level thresholding, and class imbalance, to name a few. Further, as PRS is a single-solution-based approach, it can be fused with a population-based algorithm to enhance the exploitation capability of the latter for various optimization tasks. We also intend to explore further extensions of the proposed algorithm based on the physical phenomena it is modeled upon, e.g. incorporating quantum optics or taking into account other optical phenomena that may occur relevant to the context. We may also try applying the proposed algorithm to other domains such as image thresholding and data mining.

Author contributions RH, SC and MAN performed the experiments. SN and DO Wrote the manuscript. All authors conceptualized the proposal and reviewed the manuscript. All authors contributed equally to the study conception and design.

Funding The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data availability The data used to support the finding are cited within the article. Also, the datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest All the authors declare that there is no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

1. Abd Elaziz M, Sarkar U, Nag S, Hinojosa S, Oliva D (2020) Improving image thresholding by the type ii fuzzy entropy and a hybrid optimization algorithm. Soft Comput 24(19):14885–14905
2. Abdechiri M, Meybodi MR, Bahrami H (2013) Gases brownian motion optimization: an algorithm for optimization (gbmo). Appl Soft Comput 13(5):2932–2946
3. Abdel-Basset M, Mohamed R, Sallam KM, Chakrabortty RK (2022) Light spectrum optimizer: a novel physics-inspired metaheuristic optimization algorithm. Mathematics 10(19):3466
4. Abdollahzadeh B, Gharehchopogh FS, Mirjalili S (2021) African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems. Comput Ind Eng 158:107408
5. Abedinpourshotorban H, Shamsuddin SM, Beheshti Z, Jawawi DN (2016) Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm. Swarm Evol Comput 26:8–22
6. Abualigah L, Abd Elaziz M, Sumari P, Geem ZW, Gandomi AH (2022) Reptile search algorithm (rsa): a nature-inspired meta-heuristic optimizer. Expert Syst Appl 191:116158
7. Ahmed S, Ghosh KK, Bera SK, Schwenker F, Sarkar R (2020) Gray level image contrast enhancement using barnacles mating optimizer. IEEE Access 8:169196–169214
8. Ahmed S, Ghosh KK, Garcia-Hernandez L, Abraham A, Sarkar R (2021) Improved coral reefs optimization with adaptive β-hill climbing for feature selection. Neural Comput Appl 33(12):6467–6486
9. Akay B, Karaboga D, Akay R (2021) A comprehensive survey on optimizing deep learning models by metaheuristics. Artif Intell Rev. https://doi.org/10.1007/s10462-021-09992-0
10. Al-Aboody N, Al-Raweshidy H (2016) Grey wolf optimization-based energy-efficient routing protocol for heterogeneous wireless sensor networks. In: 2016 4th International Symposium on Computational and Business Intelligence (ISCBI), IEEE, pp 101–107
11. Al-Betar MA (2017) β-hill climbing: an exploratory local search. Neural Comput Appl 28(1):153–168
12. Al-Betar MA, Aljarah I, Awadallah MA, Faris H, Mirjalili S (2019) Adaptive β-hill climbing for optimization. Soft Comput 23(24):13489–13512
13. Alatas B (2011) Acroa: artificial chemical reaction optimization algorithm for global optimization. Expert Syst Appl 38(10):13170–13180
14. Awad N, Ali M, Liang J, Qu B, Suganthan P (2017) Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization. Computational intelligence laboratory. Zhengzhou University, China and Nanyang Technological University, Singapore
15. Bandyopadhyay R, Kundu R, Oliva D, Sarkar R (2021) Segmentation of brain MRI using an altruistic Harris hawks' optimization algorithm. Knowl Based Syst 232:107468

16. Bayraktar Z, Komurcu M, Bossard JA, Werner DH (2013) The wind driven optimization technique and its application in electromagnetics. Trans Antennas Propag 61(5):2745–2757
17. Bayzidi H, Talatahari S, Saraee M, Lamarche CP (2021) Social network search for solving engineering optimization problems. Comput Intell Neurosci 2021:1–32
18. Birbil Şİ, Fang SC (2003) An electromagnetism-like mechanism for global optimization. J Glob Optim 25(3):263–282
19. Biswas A, Mishra K, Tiwari S, Misra A (2013) Physics-inspired optimization algorithms: a survey. J Optim 2013:438152. https://doi.org/10.1155/2013/438152
20. Chatterjee B, Bhattacharyya T, Ghosh KK, Chatterjee A, Sarkar R (2021) A novel meta-heuristic approach for influence maximization in social networks. Expert Syst 40(4):e12676
21. Chattopadhyay S, Kundu R, Singh PK, Mirjalili S, Sarkar R (2021) Pneumonia detection from lung x-ray images using local search aided sine cosine algorithm based deep feature selection method. Int J Intel Syst 37(7):1–38
22. Chattopadhyay S, Marik A, Pramanik R (2022) A brief overview of physics-inspired metaheuristic optimization techniques. arXiv preprint arXiv: Arxiv-2201.12810
23. Consigli G (2019) Optimization methods in finance. Taylor & Francis, Oxford
24. Cuevas E, Echavarría A, Ramírez-Ortegón MA (2014) An optimization algorithm inspired by the states of matter that improves the balance between exploration and exploitation. Appl Intell 40(2):256–272
25. Das S, Suganthan PN (2010) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evol Comput 15(1):4–31
26. Dehghani M, Montazeri Z, Dehghani A, Seifi A (2017) Spring search algorithm: a new meta-heuristic optimization algorithm inspired by Hooke’s law. In: 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), IEEE, pp 0210–0214
27. Dehghani M, Montazeri Z, Trojovská E, Trojovskỳ P (2023) Coati optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems. Knowl Based Syst 259:110011
28. Dehghani M, Samet H (2020) Momentum search algorithm: a new meta-heuristic optimization algorithm inspired by momentum conservation law. SN Appl Sci 2(10):1–15
29. Doğan B, Ölmez T (2015) A new metaheuristic for numerical function optimization: vortex search algorithm. Inf Sci 293:125–145
30. Dokeroglu T, Sevinc E, Kucukyilmaz T, Cosar A (2019) A survey on new generation metaheuristic algorithms. Comput Ind Eng 137:106040
31. Dorigo M, Blum C (2005) Ant colony optimization theory: a survey. Theor Comput Sci 344(2–3):243–278
32. Draa A, Bouaziz A (2014) An artificial bee colony algorithm for image contrast enhancement. Swarm Evol Comput 16:69–84
33. Dulebenets MA (2018) A comprehensive multi-objective optimization model for the vessel scheduling problem in liner shipping. Int J Prod Econ 196:293–318
34. Emami H (2022) Hazelnut tree search algorithm: a nature-inspired method for solving numerical and engineering problems. Eng Comput 38(Suppl 4):3191–3215
35. Emami H (2022) Seasons optimization algorithm. Eng Comput 38(2):1845–1865
36. Emami H (2022) Stock exchange trading optimization algorithm: a human-inspired method for global optimization. J Supercomput 78(2):2125–2174
37. Emami H, Derakhshan F (2015) Election algorithm: a new socio-politically inspired strategy. AI Commun 28(3):591–603
38. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm-a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 110:151–166
39. Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH (2020) Marine predators algorithm: a nature-inspired metaheuristic. Expert Syst Appl 152:113377
40. Feo TA, Resende MG (1995) Greedy randomized adaptive search procedures. J Glob Optim 6(2):109–133
41. Formato RA (2008) Central force optimization: a new nature inspired computational framework for multidimensional search and optimization. Nature inspired cooperative strategies for optimization (NICSO 2007). Springer, Cham, pp 221–238
42. Fujisawa K, Shinano Y, Waki H (2016) Optimization in the real world. Springer, Cham

43. Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68
44. Gillala R, Vuyyuru KR, Jatoth C, Fiore U (2021) An efficient chaotic salp swarm optimization approach based on ensemble algorithm for class imbalance problems. Soft Comput 25(23):1–11
45. Glassner AS (1989) Introduction to ray tracing. Morgan Kaufmann, Burlington
46. Glover F, Laguna M (1998) Tabu search. Handbook of combinatorial optimization. Springer, Cham, pp 2093–2229
47. Guha R, Khan AH, Singh PK, Sarkar R, Bhattacharjee D (2021) CGA: a new feature selection model for visual human action recognition. Neural Comput Appl 33(10):5267–5286
48. Halliday D, Resnick R, Walker J (2013) Fundamentals of physics. Wiley, New York
49. Hansen P, Mladenović N (1999) An introduction to variable neighborhood search. Meta-heuristics. Springer, Cham, pp 433–458
50. Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S (2019) Henry gas solubility optimization: a novel physics-based algorithm. Future Gener Comput Syst 101:646–667
51. Hashim FA, Hussain K, Houssein EH, Mabrouk MS, Al-Atabany W (2021) Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Appl Intell 51(3):1531–1551
52. He F (2012) Swarm intelligence for traveling salesman problems. In: Proceedings of the 2012 International Conference on Electronics, Communications and Control, pp 641–644
53. Holland JH (1992) Genetic algorithms. Sci Am 267(1):66–73
54. José-García A, Gómez-Flores W (2016) Automatic clustering using nature-inspired metaheuristics: a survey. Appl Soft Comput 41:192–213
55. Jwo DJ, Chang SC (2009) Particle swarm optimization for GPS navigation Kalman filter adaptation. Aircr Eng Aerosp Technol 81(4):343–352
56. Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. Appl Math Comput 214(1):108–132
57. Karami H, Anaraki MV, Farzin S, Mirjalili S (2021) Flow direction algorithm (FDA): a novel optimization approach for solving optimization problems. Comput Ind Eng 156:107224
58. Kashan AH (2015) A new metaheuristic for optimization: optics inspired optimization (OIO). Comput Oper Res 55:99–125
59. Kaveh A, Dadras A (2017) A novel meta-heuristic optimization algorithm: thermal exchange optimization. Adv Eng Softw 110:69–84
60. Kaveh A, Khayatazad M (2012) A new meta-heuristic method: ray optimization. Comput Struct 112:283–294
61. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of International Conference on Neural Networks, vol 4, pp 1942–1948
62. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
63. Liu Y, Sun Y, Xue B, Zhang M, Yen GG, Tan KC (2021) A survey on evolutionary neural architecture search. IEEE Trans Neural Netw Learn Syst 34:1–21. https://doi.org/10.1109/TNNLS.2021.3100554
64. Lourenço HR, Martin OC, Stützle T (2003) Iterated local search. Handbook of metaheuristics. Springer, Cham, pp 320–353
65. Mara STW, Norcahyo R, Jodiawan P, Lusiantoro L, Rifai AP (2022) A survey of adaptive large neighborhood search algorithms and applications. Comput Oper Res 146:105903
66. Maulik U, Bandyopadhyay S (2000) Genetic algorithm-based clustering technique. Pattern Recognit 33(9):1455–1465
67. Maxwell JC (1873) Molecules. Nature 8:437–441. https://doi.org/10.1038/008437a0
68. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl Based Syst 89:228–249
69. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl Based Syst 96:120–133
70. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27(2):495–513
71. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
72. Moein S, Logeswaran R (2014) Kgmo: a swarm optimization algorithm based on the kinetic energy of gas molecules. Inf Sci 275:127–144

73. Moghdani R, Salimifard K (2018) Volleyball premier league algorithm. Appl Soft Comput 64:161–185
74. Nag S (2019) Vector quantization using the improved differential evolution algorithm for image compression. Genet Program Evol Mach 20(2):187–212
75. Nakane T, Bold N, Sun H, Lu X, Akashi T, Zhang C (2020) Application of evolutionary and swarm optimization in computer vision: a literature survey. IPSJ Trans Comput Vis Appl 12(1):1–34
76. Nedjah N, Mourelle LDM, Morais RG (2020) Inspiration-wise swarm intelligence meta-heuristics for continuous optimisation: a survey-part i. Int J Bio Inspir Comput 15(4):207–223
77. Oliva D, Nag S, Abd Elaziz M, Sarkar U, Hinojosa S (2019) Multilevel thresholding by fuzzy type ii sets using evolutionary algorithms. Swarm Evol Comput 51:100591
78. Pisinger D, Ropke S (2019) Large neighborhood search. Handbook of metaheuristics. Springer, Cham, pp 99–127
79. Rao RV, Savsani VJ, Vakharia D (2011) Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43(3):303–315
80. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248
81. Salcedo-Sanz S (2016) Modern meta-heuristics based on nonlinear physics processes: a review of models and design procedures. Phys Rep 655:1–70
82. Salem SA (2012) Boa: a novel optimization algorithm. In: 2012 International Conference on Engineering and Technology (ICET), IEEE, pp 1–5
83. Selman B, Gomes CP (2006) Hill-climbing search. Encycl Cogn Sci 81:82
84. Shaw SS, Ahmed S, Malakar S, Garcia-Hernandez L, Abraham A, Sarkar R (2021) Hybridization of ring theory-based evolutionary algorithm and particle swarm optimization to solve class imbalance problem. Complex Intell Syst 7(4):1–23
85. Shen J, Li Y (2009) Light ray optimization and its parameter analysis. In: 2009 International Joint Conference on Computational Sciences and Optimization, vol 2. IEEE, pp 918–922
86. Shukri SE, Al-Sayyed R, Hudaib A, Mirjalili S (2021) Enhanced multi-verse optimizer for task scheduling in cloud computing environments. Expert Syst Appl 168:114230
87. Siddique NH, Adeli H (2017) Nature-inspired computing: physics and chemistry-based algorithms. CRC Press, Boca Raton
88. Tahani M, Babayan N (2019) Flow regime algorithm (FRA): a physics-based meta-heuristics algorithm. Knowl Inf Syst 60(2):1001–1038
89. Tanyildizi E, Demir G (2017) Golden sine algorithm: a novel math-inspired algorithm. Adv Electr Comput Eng 17(2):71–78
90. Torres-Treviño L (2021) A 2020 taxonomy of algorithms inspired on living beings behavior. arXiv preprint arXiv:2106.04775
91. Tzanetos A, Dounias G (2020) A comprehensive survey on the applications of swarm intelligence and bio-inspired evolutionary strategies. Mach Learn Paradig 2020:337–378. https://doi.org/10.1007/978-3-030-49724-8_15
92. Tzanetos A, Dounias G (2021) Nature inspired optimization algorithms or simply variations of metaheuristics? Artif Intell Rev 54(3):1841–1862
93. Veysari EF et al (2022) A new optimization algorithm inspired by the quest for the evolution of human society: human felicity algorithm. Expert Syst Appl 193:116468
94. Vidal T, Crainic TG, Gendreau M, Lahrichi N, Rei W (2012) A hybrid genetic algorithm for multi-depot and periodic vehicle routing problems. Oper Res 60(3):611–624
95. Wei Z, Huang C, Wang X, Han T, Li Y (2019) Nuclear reaction optimization: a novel and powerful physics-based algorithm for global optimization. IEEE Access 7:66084–66109
96. Wilcoxon F (1992) Individual comparisons by ranking methods. Breakthroughs in statistics. Springer, Cham, pp 196–202
97. Wolpert DH, Macready WG et al (1995) No free lunch theorems for search. Santa Fe Institute, Santa Fe
98. Yadav A et al (2019) AEFA: artificial electric field algorithm for global optimization. Swarm Evol Comput 48:93–108
99. Yang XS, Deb S (2009) Cuckoo search via lévy flights. In: 2009 World Congress on Nature and Biologically Inspired Computing (NaBIC), IEEE, pp 210–214
100. Yang XS, Gandomi AH (2012) Bat algorithm: a novel approach for global engineering optimization. Eng Comput 29(5):464–483

101. Yang XS, Karamanoglu M, He X (2014) Flower pollination algorithm: a novel approach for multi-objective optimization. Eng Optim 46(9):1222–1237
102. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3(2):82–102
103. Yousri D, Abd Elaziz M, Mirjalili S (2020) Fractional-order calculus-based flower pollination algorithm with local search for global optimization and image segmentation. Knowl Based Syst 197:105889
104. Zhao W, Wang L, Mirjalili S (2022) Artificial hummingbird algorithm: a new bio-inspired optimizer with its engineering applications. Comput Methods Appl Mech Eng 388:114194
105. Zhao W, Wang L, Zhang Z (2019) Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. Knowl Based Syst 163:283–304
106. Zitouni F, Harous S, Maamri R (2020) The solar system algorithm: a novel metaheuristic method for global optimization. IEEE Access 9:4542–4565

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Rohit Kundu[1] · Soumitri Chattopadhyay[2] · Sayan Nag[3] · Mario A. Navarro[4] · Diego Oliva[4]

✉ Diego Oliva
diego.oliva@cucei.udg.mx

Rohit Kundu
rohitkunduju@gmail.com

Soumitri Chattopadhyay
soumitri.chattopadhyay@gmail.com

Sayan Nag
nagsayan112358@gmail.com

Mario A. Navarro
mario.navarro@academicos.udg.mx

[1] Department of Electrical Engineering, Jadavpur University, Kolkata, India

[2] Department of Information Technology, Jadavpur University, Kolkata, India

[3] Department of Medical Biophysics, University of Toronto, Toronto, Canada

[4] Departamento de Ingeniería Electro-Fotónica, Universidad de Guadalajara, CUCEI, Guadalajara, Mexico