Artificial Intelligence Review (2024) 57:24
https://doi.org/10.1007/s10462-023-10683-1

Portia spider algorithm: an evolutionary computation approach for engineering application

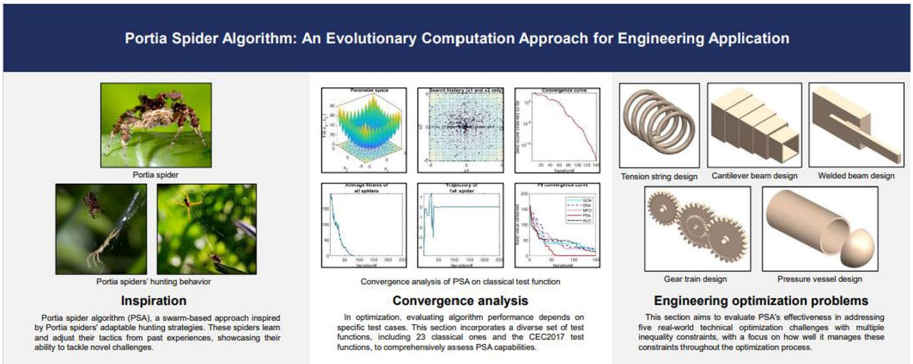
Vu Hong Son Pham[1] · Nghiep Trinh Nguyen Dang[1]

Accepted: 20 December 2023 / Published online: 31 January 2024

Abstract

The Portia spider, a notable member of the jumping spider family (Salticidae), is widely recognized for its intricate hunting strategies and remarkable problem-solving prowess. Several species fall under the "Portia" genus, with habitats spanning regions in Africa, Asia, and Australia. Demonstrating the ability to tackle new challenges, these spiders can learn and adapt their strategies based on prior experiences. This study introduces the Portia Spider Algorithm (PSA), a swarm-based technique inspired by the unique predatory strategies of the Portia spider. We conducted rigorous assessments of PSA performance against 23 classical test functions, 29 CEC2017 test cases, and 5 engineering optimization tasks. To demonstrate the effectiveness of the PSA, outcomes were juxtaposed with those of renowned algorithms. This paper explores the mechanics, advantages, and potential applications of PSA within the vast domain of computational optimization.

Graphical Abstract



Keywords Swarm-based algorithm · Engineering optimization · Hunting behavior · Evolutionary algorithm · Benchmark · Particle swarm optimization

Extended author information available on the last page of the article

1 Introduction

Meta-heuristic algorithms, when compared to traditional optimization methods, distinguish themselves by their simplicity in terms of both comprehension and implementation. Traditional optimization techniques, such as integer programming (Williams 2009), linear programming (Dantzig 2002), mixed programming (Hooker and Osorio 1999), and various constrained optimization methods (Gautier and Granot 1994), are typically designed for well-structured problems. Such problems possess clear structural information, established parameters, and a uniquely identifiable global optimum. These traditional methods are amenable to rigorous evaluation through computational complexity and convergence theories. They are particularly adept at addressing problems characterized by a single extremum. However, their performance can be less than satisfactory in scenarios dominated by multi-extremum challenges.

Conversely, meta-heuristic algorithms, through their astute design, manage to strike a nuanced balance between evading local optima and ensuring convergence to a viable solution. This design feature significantly augments the likelihood of pinpointing the global optimum in an array of optimization contexts. One of the salient characteristics of metaheuristics is their relative indifference to initial conditions, which reinforces the consistency of optimization outcomes. Their inherent robustness, coupled with domain independence, renders them indispensable in a myriad of practical applications.

Given the pivotal role of optimization across diverse fields, there has been an upswing in the emphasis on metaheuristic techniques (Spall 2005). Various domains have leveraged these methods (Boussaïd et al. 2013; Parejo et al. 2012; Zhou et al. 2011) that operate by generating and refining random solutions iteratively [9]. In stark contrast to traditional methods that necessitate well-defined mathematical models, metaheuristics optimize by varying inputs and scrutinizing the corresponding outputs to maximize or minimize objective functions. This attribute ensures they are less susceptible to ensnarement in local optima compared to their traditional counterparts. Another distinguishing feature of metaheuristics is their adaptability. In a broader context, metaheuristics are categorized according to the extent of random choices they engender during each optimization iteration. Additionally, they can be classed by their foundational inspiration, which could be grounded in swarm intelligence, evolutionary principles, physics, mathematics, or human concepts (Fig. 1).

Evolutionary algorithms, a subset of meta-heuristic techniques, emulate the evolutionary mechanisms observed in nature. Among these, the genetic algorithm (GA) stands out as one of the most renowned. Introduced by Holland (1992), the GA has since garnered extensive attention across various domains, serving as a go-to method for optimization and search problems. Similarly inspired by natural processes, Simon (2008) unveiled the biogeography-based optimization (BBO). This approach taps into the mathematical principles underlying the habitat distribution of life forms. Simon drew parallels between BBO and the evolution of GA and artificial neural networks, particularly emphasizing its prowess in addressing optimization challenges. Storn and Price (1997) contributed to the field in the form of an innovative approach - differential evolution (DE) that promises efficient minimization of complex, potentially nonlinear, and non-differentiable continuous space functions. Their method stands out not only for its rapid convergence and superior certainty over other prevalent global optimization techniques but also for its inherent simplicity and adaptability to parallel computing. Adding to the tapestry of evolutionary algorithms, Jaderyan and Khotanlou (2016) introduced the virulence optimization algorithm

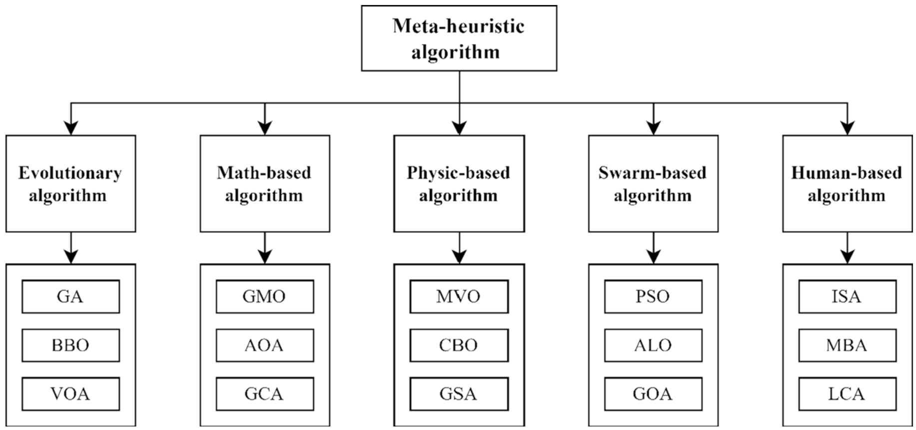


Fig. 1 Categorization of meta-heuristic methods

(VOA). Inspired by the intricate infection mechanisms employed by viruses, this method involves identifying the fittest viruses, cloning them to amplify infection rates, and judiciously evading already infected zones. When tested against 11 benchmark functions, the VOA showcased its potential as a formidable resource for handling complex optimization conundrums.

Numerous techniques have drawn inspiration from mathematical concepts and physical phenomena, as evident in recent advancements in optimization methods. Rezaei et al. (2023) presented the geometric mean optimizer (GMO), a parameter-free meta-heuristic method grounded in the geometric mean operator. Their research indicated that the GMO excels at tackling a diverse array of optimization challenges, surpassing other contemporary meta-heuristic algorithms in multiple benchmark tests. Similarly, Abualigah et al. (2021) introduced the arithmetic optimization algorithm (AOA). This innovative meta-heuristic method uses mathematical arithmetic operators to optimize across varying search spaces. Through comprehensive evaluation on test functions and applied engineering design tasks, AOA demonstrated superior performance, convergence behavior, and computational complexity compared to eleven other renowned optimization algorithms. Chickermane and Gea (1996) proposed the generalized convex approximation (GCA), a cutting-edge method for structural optimization. By leveraging design sensitivity data, they constructed a series of convex subproblems, which in turn showcased enhanced convergence rates when assessed against standard test problems.

Shifting focus to algorithms inspired by physical phenomena, Mirjalili et al. (2016) introduced the multi-verse optimizer (MVO). This nature-inspired algorithm derives its concepts from cosmological phenomena such as white holes, black holes, and wormholes. Through mathematical modeling of these cosmic events, MVO effectively employs exploration, exploitation, and local search techniques. Kaveh and Mahdavi (2014) have put forward the colliding bodies optimization (CBO) meta-heuristic algorithm, a method underpinned by the principles of one-dimensional collisions. When applied to truss designs with discrete dimension variables, this approach – devoid of the need for parameter tuning and offering a lucid formulation—demonstrated its efficacy in structural optimization. Another intriguing method is the gravitational search algorithm (GSA) introduced by Rashedi et al. (2009). This strategy, inspired by gravity and mass interactions, adopts masses as searcher agents. The research underscores its formidable performance in solving diverse nonlinear functions, especially when pitted against well-established heuristic search methods.

A Kaveh and Siamak Talatahari (2010) proposed the charged system search (CSS) algorithm, which employs principles from electrostatics and Newtonian mechanics. This algorithm demonstrates superior performance in optimization tasks amidst discontinuous or non-convex territories, circumventing the need for gradient information or the smoothness of the search area. Abedinpourshotorban et al. (2016) presented the electromagnetic field optimization (EFO). This physics-inspired optimization technique integrates the golden ratio to dictate attraction-repulsion forces among electromagnetic particles within a tri-field structure. Remarkably, its superiority concerning precision and speed of convergence was evident on the 30 high-dimensional CEC 2014 benchmark problems, outperforming other leading optimization algorithms. Kashan (2015) introduced optics-inspired optimization (OIO) which models numerical optimization processes on the reflective behaviors of concave and convex mirrors, yielding a competitive and parameter-efficient method for solving complex optimization problems.

Human concepts have increasingly become a source of inspiration for metaheuristic techniques, leading to the development of novel optimization methods that mimic various human activities and thought processes. Reynolds (1994) presented a computational model for cultural evolution, which employs dual inheritance within a population of individuals characterized by specific behavioral traits and a collective “belief space”. This model is accelerated by GA and version spaces, which serve to enhance learning rates within the structure of the cultural algorithm. Rao et al. (2011) introduced the teaching–learning-based optimization (TLBO). This method, which is inspired by the pedagogical effect of a mentor on learners, proves effective for mechanical design problems. In a series of test cases and real-world applications, TLBO showcased performance superiority over other population-based optimization algorithms. Gandomi (2014) unveiled the interior search algorithm (ISA), a unique optimization technique that takes inspiration from the art and craft of interior design. Noteworthy is its simplicity, requiring tuning of just one parameter. The ISA stands out for its capability to solve optimization challenges, surpassing established algorithms in benchmark tests.

In a blend of music, Lee and Geem (2005) presented the harmony search (HS) method. Drawing inspiration from the musical process of seeking harmony, this method sets itself apart by not relying on gradient information. The research evidences the robustness and effectiveness of HS in addressing diverse engineering optimization problems, often delivering solutions that may surpass those derived from traditional algorithms. Sadollah et al. (2013) have proposed the mine blast algorithm (MBA), an innovative population-based optimization method influenced by the explosive dynamics of mine blasts. The results obtained from their research underscore the algorithm’s efficiency and its exceptional performance in addressing both constrained optimization and engineering design tasks, especially when juxtaposed against well-established optimization methods. Kashan (2014) introduced the league championship algorithm (LCA). This method simulates a sporting championship environment, where artificial teams vie for dominance in a league. Teams adapt their strategies or solutions based on game results, while also modeling player transfers at season’s end. Empirical analysis conducted on a variety of benchmark functions showcases its promising performance, hinting at its potential suitability for practical applications in the future.

In recent years, metaheuristic techniques that draw inspiration from animal behavior have garnered significant attention from researchers, as highlighted in Table 1. Kennedy and Eberhart (1995) presented the concept of optimizing nonlinear functions through the

Table 1 A summary of swarm-based algorithms

Author(s)	Method	Inspiration	Year
Gandomi et al. (2013)	Cuckoo search algorithm	Cuckoo	2013
Cuevas et al. (2013)	Social-spider optimization	Social spider	2013
Cheng and Prayogo (2014)	Symbiotic organisms search	Symbiotic	2014
Bansal et al. (2014)	Spider monkey optimization algorithm	Spider monkey	2014
Mirjalili (2015a)	Ant lion optimizer	Ant lion	2015
Mirjalili (2015b)	Moth-flame optimization algorithm	Moth-flame	2015
James and Li (2015)	Social spider algorithm	Social spider	2015
Mirjalili and Lewis (2016)	The whale optimization algorithm	Whale	2016
Saremi et al. (2017)	Grasshopper optimisation algorithm	Grasshopper	2017
Mirjalili et al. (2017)	Salp Swarm Algorithm	Salp Swarm	2017
Mirjalili et al. (2018)	Grasshopper optimisation algorithm	Grasshopper	2018
Alsattar et al. (2020)	bald eagle search optimisation algorithm	bald eagle	2020
Xue and Shen (2020)	Sparrow search algorithm	Sparrow	2020
Zhao et al. (2020)	Manta ray foraging optimization	Manta	2020
Abualigah et al. (2021)	Aquila optimizer	Aquila	2021
Połap and Woźniak (2021)	Red fox optimization algorithm	Red fox	2021
Xie et al. (2021)	Tuna swarm optimization	Tuna swarm	2021
Abdollahzadeh et al. (2022)	Mountain gazelle optimizer	Mountain gazelle	2022
Chen et al. (2022)	Egret swarm optimization algorithm	Egret swarm	2022
Chopra and Ansari (2022)	Golden jackal optimization	Golden jackal	2022
Hashim et al. (2022)	Honey Badger Algorithm	Honey Badger	2022
Sadeeq and Abdulazeez (2022)	Giant trevally optimizer	Giant trevally	2022
Wang et al. (2022)	Artificial rabbits optimization	Rabbit	2022
Seyyedabbasi and Kiani (2023)	Sand cat swarm optimization	Sand cat	2023
Zhao et al. (2023)	Sea-horse optimizer	Seahorse	2023

particle swarm optimization (PSO) technique. Their research delved into the evolution of different paradigms within this approach, executed one such paradigm, and subjected it to benchmark tests. They further identified potential applications, notably in nonlinear function optimization and neural network training. Gandomi et al. (2013) presented the cuckoo search (CS) metaheuristic optimization algorithm, which is enriched by Lévy flights. Through comprehensive testing on benchmark nonlinear constrained optimization tasks and structural engineering design challenges, CS demonstrated a distinct edge over other cutting-edge techniques. The authors also emphasized its unique search features, positioning it as a valuable subject for future research endeavors. Seyyedabbasi and Kiani (2023) introduced the sand cat swarm optimization (SCSO) algorithm. Drawing inspiration from the survival strategies of sand cats, SCSO proves adept at solving optimization challenges by skillfully balancing exploration and exploitation phases. The algorithm's exceptional performance, evident in benchmark tests and intricate engineering design problems, places it a notch above other metaheuristic competitors.

Chen et al. (2022) unveiled the egret swarm optimization algorithm (ESOA), an inventive meta-heuristic technique influenced by the hunting behaviors of two egret species. Across an array of benchmark functions and engineering challenges, ESOA showcased its marked effectiveness and robustness, often overshadowing other optimization algorithms.

Abualigah et al. (2021) brought to light the Aquila optimizer (AO), a metaheuristic method that mirrors the predatory strategies of Aquila birds. Their work delineates four distinct methods that reflect the bird's hunting strategies. The AO algorithm underwent rigorous validation, with tests on renowned functions, the intricate CEC2017 and CEC2019 functions, and real-world engineering scenarios. Mirjalili et al. (2017) introduced the salp swarm algorithm (SSA) and its multi-objective counterpart (MSSA), inspired by salps' swarming patterns, and validated their effectiveness on mathematical and applied engineering design tasks, showcasing their ability to efficiently converge to optimal solutions and approximate Pareto fronts. Bansal et al. (2014) introduced the spider monkey optimization (SMO) method, an innovative approach to numerical optimization inspired by the adaptable hunting behaviors of spider monkeys, animals known for their fission-fusion social dynamics.

The behavior of spiders in addressing optimization challenges has also inspired the development of metaheuristics and is applied in various fields. Cuevas et al. (2013) introduced the social spider optimization (SSO) technique, inspired by the cooperative interactions of social spiders, featuring distinct evolutionary operators for male and female agents, and exhibits high proficiency in locating global optima on various benchmark functions. Ouadfel and Taleb-Ahmed (2016) explored the effectiveness of the flower pollination (FP) and SSO method in image segmentation through multilevel thresholding, demonstrating that while both outperform the PSO and BAT algorithms, SSO maintains superior stability and efficiency across various threshold numbers, making it a compelling choice for complex image thresholding tasks. Ewees et al. (2017) presented an enhanced adaptive neuro-fuzzy inference system (ANFIS) optimized using the SSO algorithm for forecasting biochar yield from manure pyrolysis, showcasing superior performance over classical ANFIS and other methodologies such as artificial bee colony, PSO, and least square-support vector machine. Nguyen and Vo (2020) introduced an improved SSO technique that enhances the solution generation process for optimal reactive power dispatch problems, demonstrating reduced computational steps and parameters, faster simulation times, and consistently superior solution quality when benchmarked against standard SSO and other leading methods.

In an independent study, James and Li (2015) unveiled a novel social spider algorithm (SOSA) for global optimization. This algorithm uniquely harnesses web vibrations, which social spiders use to locate prey, deviating from conventional swarm intelligence strategies and showing superior results in benchmark assessments. SOSA has been applied in a variety of fields. For instance, Elsayed et al. (2016) presented a modified version of SOSA to address the non-convex economic load dispatch task, which includes real-world restrictions like valve point impacts and prohibited operating areas. The enhanced algorithm demonstrates improved performance across four benchmark systems, surpassing the original SOSA and displaying a competitive advantage in the existing literature. El-Bages and Elsayed (2017) presented an SOSA model for static transmission expansion planning that provides cost-effective solutions, validated on benchmark systems with varying degrees of complexity. Baş and Ülker (2020) introduced a binary version of SOSA, augmented with similarity measures and logic gates, and their results indicate superior performance in solving unimodal, multimodal, and uncapacitated facility location problems when compared with traditional algorithms.

The evolution and enhancement of existing algorithms have become a focal point for researchers in recent years. The thrust to augment existing algorithms for emerging challenges has driven significant academic interest. Kaveh and Talatahari (2010a) brought forth an advanced version of the ant colony optimization (ACO), termed the improved ACO.

Crafted for constrained engineering design challenges, this approach deftly manages both continuous and discrete problems. Central to its design is the sub-optimization mechanism (SOM), inspired by finite element principles, which efficiently trims down pheromone matrices, decision vectors, function evaluations, and optimization durations without jeopardizing the likelihood of pinpointing optimal solutions. Chakraborty et al. (2009) embarked on refining the harmony search (HS) algorithm, which finds its muse in musical improvisation. Their method hybridizes HS with the differential evolution (DE) algorithm, aiming to mitigate slow and premature convergence challenges on intricate fitness terrains. Performance metrics of this hybrid were juxtaposed against classical HS, global best HS, and a renowned DE variant. Assessments were made across diverse benchmark functions and practical optimization challenges, based on criteria like precision, computational pace, and consistency in reaching optima.

Son and Nguyen Dang (2023b) showcased a composite model named the hybrid multi-verse optimizer (hDMVO), which integrates MVO and the sine cosine algorithm (SCA). This model is adept at managing discrete time-cost trade-off (TCTO) quandaries in construction project orchestration. Its prowess shines through benchmark evaluations and its knack for devising superior solutions in large-scale TCTO scenarios for intricate projects. Abd Elaziz et al. (2017) unveiled an augmented SCA, enriched with opposition-based learning (OBL). This enhancement broadens the exploration horizon of the search domain, leading to heightened precision and overall performance. Its efficacy is evident from benchmark evaluations and complex engineering tasks, underscoring the value of this synergistic approach. Pham, Trang, et al. (2023) articulated a proficient scheduling optimization technique for ready-mix concrete (RMC) truck dispatches. At its core is a novel hybrid swarm intelligence algorithm fusing the grey wolf optimizer (GWO) with the dragonfly algorithm (DA). The resultant algorithm excels in performance compared to its standalone counterparts and heralds a leap in multi-independent batch plant cooperation for refined RMC deliveries in construction sectors.

In recent years, optimization techniques have found widespread applications across diverse fields. Pham, Nguyen Dang, et al. (2023) introduced an enhanced SCA that integrates roulette wheel selection with OBL, demonstrating superior performance over traditional optimization algorithms in various engineering optimization contexts. Son and Nguyen Dang (2023a) presented the MVO model as an effective tool for addressing time–cost optimization issues in construction project management, surpassing other techniques in small-scale applications. Kumar et al. (2023) unveiled the multi-objective MVO, a dual-archive metaheuristic tailored for complex structural optimization, and illustrated its superior performance in real-world applications over leading algorithms such as MOEA/D and NSGA-II through comprehensive evaluations employing established performance metrics. Aye et al. (2023) introduced an innovative surrogate-aided optimization approach for enhancing airfoil shapes, incorporating both CFD and XFoil simulations. This method demonstrated superior performance in comparison to conventional surrogate-assisted techniques.

Nonut et al. (2022) advocated for the application of metaheuristics in the system identification of fixed-wing UAVs by optimizing aerodynamic and stability derivatives to minimize R-squared errors, with the L-SHADE algorithm proving to be the most effective method through extensive statistical analysis. Singh et al. (2022) unveiled an improved follow-the-leader (iFTL) technique, inspired by the foraging behaviors of sheep. Comprehensive testing within the Comparing Continuous Optimisers framework, alongside a suite of benchmark functions and truss design problems, revealed that iFTL outperforms fourteen well-established optimization algorithms in terms of performance and stability. In another study, Kumar et al. (2022) enhanced the efficacy of discrete meta-heuristics for truss design by incorporating both a random mutation search phase and selection based on simulated annealing into five pre-existing algorithms. The resultant modifications led to improved search diversification and intensification, yielding superior optimization results in complex structural engineering challenges. Kumar, Tejani, Pholdee, Bureerat, et al. (2022) introduced a groundbreaking multi-objective TLBO method. This novel approach incorporates non-dominated sorting and an external archive, adeptly mitigating early convergence and preventing local optima entrapment in multi-objective contexts.

Recent trends in algorithmic research spotlight both the inception of new algorithms and the refinement of pre-existing ones. The burgeoning interest in this area owes much to the No Free Lunch (NFL) theorem (Wolpert and Macready 1997), which posits that no single optimization technique can universally address all optimization challenges.

In light of the NFL theorem, this study introduces the Portia spider algorithm (PSA), inspired by the predatory strategies of the Portia spider. The PSA's efficacy was rigorously assessed using a suite of benchmarks: 23 classical test cases, 29 CEC2017 benchmark test functions, and five practical engineering optimization tasks. The algorithm's performance was then benchmarked against a collection of established meta-heuristic algorithms.

The primary attributes and contributions of the PSA include:

- Absence of any tunable parameters.
- An agile adjustment of solution positions, grounded in the spider's stalking and striking hunting behaviors, fostering expansive exploration and ensuring robust population diversity.
- The updating of solution positions, blending the stalking and striking tactics (exploration phase) with the invading and imitating tactics (exploitation phase), which harmonizes the exploration and exploitation phases.

The subsequent sections of this study are structured as follows: Sect. 2 introduces a comprehensive framework for the PSA. In Sects. 3, 23 classical and 29 CEC2017 test functions were employed to probe the convergence properties of the PSA. Section 4 evaluates the real-world applicability of the PSA through five engineering optimization scenarios. Finally, Sect. 5 summarizes the findings, spotlighting the study's novel contributions, and signposting avenues for prospective research in this domain.

2 Portia spider algorithm

2.1 Inspiration

Over 48,000 unique spider species coexist on our planet. Much like humans who have evolved to inhabit diverse environments over millennia, spiders have undergone adaptations spanning millions of years. These adaptations have produced species ranging from fishers to desert inhabitants, high-altitude enthusiasts, and astute strategists like the Portia spider. The Portia spider primarily resides in tropical forests, typified by their humid and warm climate. Intriguingly, this spider preys on various other spiders, some of which possess the ability to turn the tables on the predator-prey relationship (Robson 2020).

On an average hunting day, a Portia spider might discreetly stalk another jumping spider or cautiously venture into the web of an orb-weaving spider. Both pursuits come with risks,

Fig. 2 Portia spider

a. Portia spiders using the vibration of an ensnared insect

b. Portia spider positioning a silk dragline directly above the web

Fig. 3 Portia spiders' hunting behavior when invading the webs of other spiders

as the spiders Portia targets can invert the prey-predator dynamic, posing a threat to Portia. The necessity to confront such hazardous scenarios for sustenance has likely sculpted Portia into a master strategist.

One of the most notable strategies employed by the Portia spider (Fig. 2) becomes apparent when it detects chemical signals from a fellow jumping spider. Even devoid of direct visual cues, the Portia ascertains the presence of its quarry through these chemical indicators. In response to this chemical intelligence, the Portia adjusts its customary movement, initiating an abrupt leap to momentarily divert the attention of the observant jumping spider. This fleeting action momentarily breaks the latter's camouflage, betraying its location. With calculated patience, the Portia waits for its target to divert its attention before drawing nearer, primed for its climactic strike (Jackson and Wilcox 1998).

Additionally, Portia spiders employ another strategy: invading the webs of other spiders. To lure their prey, they imitate the vibrations of an ensnared insect by disturbing the web (Cross and Jackson 2005) (Fig. 3a). Yet, many web-spinning spiders seem to have decoded this tactic, making discreet stalking frequently a more successful approach. Observations have revealed that Portia spiders occasionally capitalize on natural web vibrations caused by ambient breezes. If circumstances necessitate, they can even produce these vibrations independently. Researchers documented higher prey capture success rates for Portias in environments where webs were intermittently perturbed, compared to scenarios with static webs. At times, when the direct vibratory mimicry falls short in attracting prey, the Portia spider opts for an alternate approach, positioning a silk dragline directly above the web, and then pouncing once in close proximity to its target (Fig. 3b).

Distinctly, while many predators employ natural elements for camouflage or utilize environmental facets for stealth, Portia spiders display a novel behavior, serving as the first documented instance where an organism effectively creates its own diversion. The ingenuity of Portia spiders extends to adaptive learning during hunts. They engage in intricate signaling on their prey's web, dynamically adjusting their methods based on observed responses. This indicates a trial-and-error approach in their hunting style. Occasionally, upon detecting prey from afar, Portia spiders adopt an indirect route, often embarking on journeys lasting up to two hours, showcasing their aptitude for strategic anticipation.

This particular species appears to determine its approach or web signal for a specific target spider through a sophisticated process of trial and error, enhanced by cues from the prey or its web. The signaling tactics of Portia continuously evolve until the targeted prey inadvertently responds, often sealing its fate. Such behaviors exemplify a problem-solving acumen seldom seen among spiders. It is this unparalleled hunting aptitude that inspired the development of the Portia spider algorithm (PSA).

2.2 Mathematical model

In the PSA framework, the behaviours and strategies of the Portia spider are abstracted and transformed into a computational model. Here, each individual Portia spider is treated as a unique solution to a given problem. The specific location or position of each spider (or solution) is defined by a set of variables, potentially representing different parameters or attributes of the solution.

To visualize and manage these solutions, they are organized within a matrix. In this matrix, every row could represent an individual Portia spider (or a solution) while every column could correspond to a specific variable or parameter of the solution:

$$\boldsymbol{S} = \begin{bmatrix} s_1^1 & s_1^2 & \cdots & s_1^d \\ s_2^1 & s_2^2 & \cdots & s_2^d \\ \cdots & \cdots & \cdots & \cdots \\ s_N^1 & s_N^2 & \cdots & s_N^d \end{bmatrix} \tag{1}$$

In Eq. (1), the variable N stands for the number of Portia spiders, equating to the population size, while d denotes the parameters associated with the optimization issue. This matrix provides a holistic representation of all possible solutions along with their specific attributes. Over the course of the algorithm, the positions of the Portia spiders (which represent solutions) may alter, indicating the optimization or evolution of the solution toward the problem's objective. The fitness scores of these spiders are then ranked and compiled into a vector according to the following expression:

$$\boldsymbol{S} = \begin{bmatrix} F(S_1) \\ F(S_2) \\ \cdots \\ F(S_N) \end{bmatrix} \tag{2}$$

Besides the Portia spiders, the prey is pivotal to the PSA framework. As outlined in Eq. (2), the prey vector represents the optimal solution identified up to that point. All Portia spiders orient their movement relative to this prey. The position of the prey is detailed in Eq. (3).

$$\boldsymbol{T} = \left[t_1 t_2 \dots t_d\right] \tag{3}$$

The movement of the Portia spiders, oriented relative to their prey, is crucial for obtaining optimal results. By emulating the hunting strategies of the Portia spider, the PSA can adaptively explore the solution space. This approach enables the algorithm to use trial-and-error and adjust strategies based on feedback, mirroring how Portia spiders modify their hunting techniques in response to their prey's behaviors. In subsequent sections, the mechanism for updating positions in the PSA is detailed, encompassing both the stalking and striking actions (during the exploration phase) as well as the invading and imitating actions (during the exploitation phase).

2.2.1 Stalking and striking (exploration phase)

This stage is inspired by the hunting strategies of the Portia spider, especially its reaction to chemical signals from another jumping spider (Fig. 2). When these signals are detected, the Portia spider momentarily deviates from its typical waddling gait, executing what can be described as a spontaneous jump. This sudden move induces the prey to break its camouflage, inadvertently revealing its position to the Portia spider. This observed behavior is then translated into the following mathematical formulation:

$$s_{ij}^{t+1} = \begin{cases} t_j if \alpha_1 < SF\left(S_i\right) \\ s_{ij}^t if \alpha_1 \geq SF\left(S_i\right) \end{cases} \tag{4}$$

subject to:

$$SF\left(S_i\right) = \frac{F\left(S_i\right)}{\sqrt[2]{\sum_1^N F\left(S_i\right)^2}} \tag{5}$$

where t_j symbolizes the j^{th} parameter of the prey (which is the best solution identified up to this point); s_{ij} denotes the j^{th} parameter of the i^{th} solution; α_1 is a random value that falls between 0 and 1; $NF(S_i)$ and $F(S_i)$ correspond to the standardized fitness score and the actual fitness score of solution S_i, respectively.

This mechanism triggers sudden variations, promoting a more extensive search or exploration and ensuring population diversity. In optimization techniques, striking a balance between exploration and exploitation is crucial. While exploitation focuses on refining solutions within a localized region, exploration delves into uncharted areas of the solution landscape in search of superior alternatives. This concept parallels the crossover technique in GA, which emphasizes thorough traversal of the search domains.

2.2.2 Invading and imitating (exploitation phase)

This phase is inspired by a unique hunting strategy displayed by the Portia spider: the act of encroaching upon the webs of other spiders. In this behavior, the Portia spider sneaks into the webs of other spiders and lures its prey by producing web vibrations. These vibrations mimic the movements of a trapped insect (Fig. 3a). While the Portia spider sometimes takes advantage of naturally occurring vibrations in the web, like those caused by a gentle breeze, it is also skilled at producing these vibrations on its own when necessary. This intriguing natural behavior is translated into the subsequent mathematical model, which draws inspiration from the vibration function that incorporates damping:

$$s_{ij}^{t+1} = t_j^t + \left|\alpha_2 t_j^t - s_{ij}^t\right| \times e^{\alpha_3} \times \cos\left(2\pi\alpha_4\right) \tag{6}$$

However, many web-weaving spiders seem to have caught on to these tricks. When simple vibrational mimicry fails to attract the prey, the Portia spider employs an alternative strategy. It attaches a silk dragline just above the web and descends cautiously, stopping only when it's within striking distance of its intended prey (Fig. 3b). With this hunting method, Portia spiders engage in direct pursuit of their prey, foregoing the mimicry of movements characteristic of trapped insects. This observed behavior is captured in Eq. (7), wherein the current solution position is adjusted directly based on the optimal solution identified up to that point, a strategy that differs from the solution position modification presented in Eq. (6).

$$s_{ij}^{t+1} = t_j^t + \alpha_2 \times \left(t_j^t - s_{ij}^t\right) \tag{7}$$

These two equations Eq. (6) and Eq. (7) are combined to be used as follows:

$$s_{ij}^{t+1} = \begin{cases} t_j^t + \left|\alpha_2 t_j^t - s_{ij}^t\right| \times e^{\alpha_3} \times \cos\left(2\pi\alpha_4\right) \alpha_5 < 0.5 \\ t_j^t + \alpha_2 \times \left(t_j^t - s_{ij}^t\right) \alpha_5 \geq 0.5 \end{cases} \tag{8}$$

subject to:

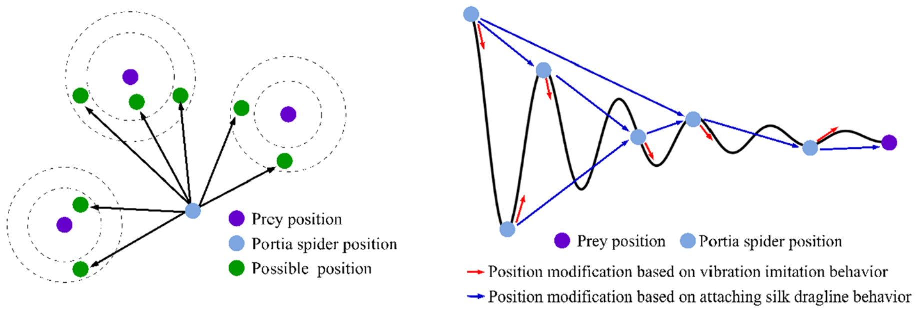
$$\alpha_3 = 1 - \left(\frac{I_{cur}}{I_{max}}\right)^2 \tag{9}$$

where t_j represents the j^{th} parameter of the prey (the best solution identified thus far); s_{ij} denotes the j^{th} parameter of the i^{th} solution; α_2, α_4 and α_5 are random values within the range [0,1]; α_3 is determined by Eq. (9) ensuring a harmony between the exploration and exploitation phases. I_{cur} and I_{max} denote the current and the maximum iteration number, respectively.

Specifically, Table 2 presents the PSA algorithm in the form of pseudocode. This structured breakdown elucidates the algorithm's step-by-step progression, allowing readers to intuitively understand its logic and sequence of operations. Furthermore, Fig. 4 depicts the exploration and exploitation mechanisms of the PSA. A mathematical model inspired by the stalking and striking behavior of the Portia spider, as expressed in Eq. (5), induces abrupt changes in the solution position to enhance exploration (see Fig. 4a). In contrast, a model derived from the invading and imitating behavior typical of the Portia spider, detailed in Eq. (8), refines the solution position to foster exploitation (refer to Fig. 4b). Complementing these models, Fig. 5 presents a flowchart that succinctly outlines the PSA framework.

2.3 Differences between PSA and other spider-inspired algorithms

In recent years, the field has seen the emergence of numerous swarm intelligence algorithms, inspired by the collective behavior observed in nature. Prominent among these are social spider optimization (SSO) (Cuevas et al. 2013) and social spider algorithm (SOSA)



a. Exploration phase b. Exploitation phase

Fig. 4 PSA exploration and exploitation concept

Table 2 Pseudocode of the PSA

Input: Population size (N); number of iteration (I_{max})

Begin
Generate random Portia spiders;
while (I_cur < I_max) **do**
Calculate and sort the fitness value;
Determine prey position;
Stalking and striking (exploration phase)
Update value of alpha_1;
Calculate standardized fitness score;
Update Portia spider position using Eq. (4);
Invading and imitating (exploitation phase)
Update value of alpha_2, alpha_3, alpha_4, and alpha_5;
Update Portia spider using position Eq. (8);
end
Update Portia spider set;
Determin the best solution;
I_cur = I_cur + 1;
end

Output: Optimal solution and its fitness score.

(James and Li 2015), which are metaheuristic algorithms that mimic spider behavior. While PSA also draws inspiration from spider behavior, it diverges significantly from SSO and SOSA in various aspects of its operational methodology and design principles.

The divergence between PSA and SSO originates from their respective sources of inspiration and intricacies of design. SSO takes cues from the sophisticated mating behavior of social spiders, incorporating a system of gender distinction into its search strategy. This model allows male and female agents within the algorithm to undertake unique roles and engage in distinct search behaviors. In contrast, PSA takes a cue from the isolated foraging patterns of the Portia spider. By discarding gender distinctions, all agents in PSA adhere to a uniform search approach, which simplifies the algorithm's application.

The strategies to ensure diversity in solutions also show significant differences. SSO employs a mating algorithm, simulating mate selection to generate a range of solutions akin to the genetic diversity found in spider populations. Conversely, PSA uses mathematical models to abstract the predatory behavior of Portia spiders, such as stalking and

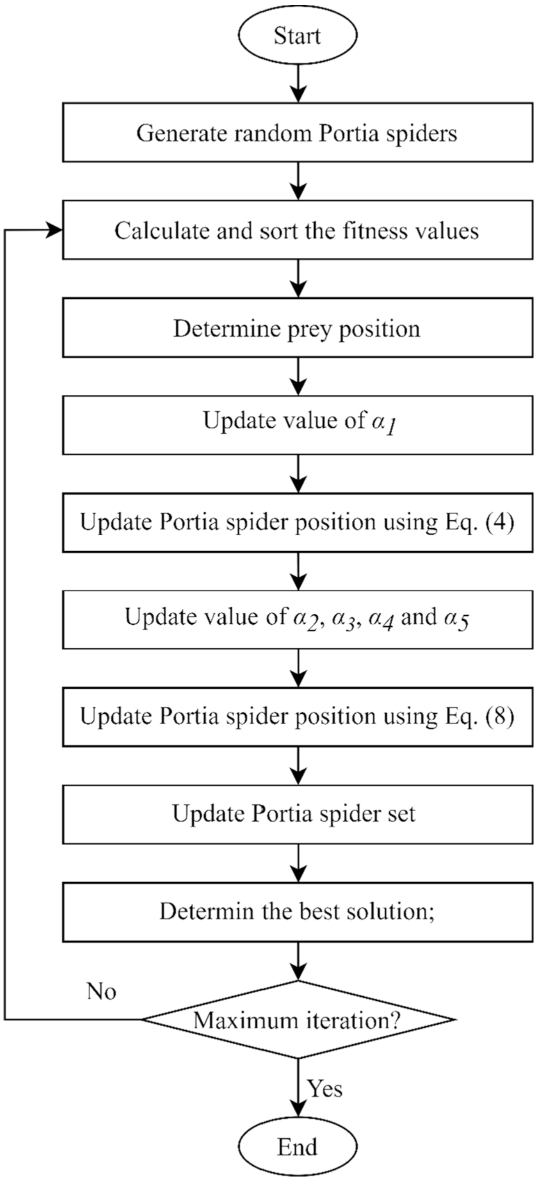


Fig. 5 Flowchart of the PSA

striking, to introduce significant and strategic variations in the search patterns of the algorithm, as detailed in Eq. (5). This approach bolsters the exploratory capabilities of the algorithm, allowing for more effective navigation through complex search spaces.

While PSA and SOSA both emulate spider predatory behavior, they diverge in operational mechanisms. SOSA is based on the social behavior of spiders, using web vibrations to pinpoint the location of prey. In contrast, PSA takes cues from the Portia spider hunting strategy, which involves simulating the vibrations of trapped insects to attract other spiders, with the mathematical model for this behavior expressed in Eq. (6). Additionally, the solution position adjustment mechanism in PSA is more varied. When vibrational mimicry fails in attracting prey, the Portia spider employs an alternative strategy: it attaches a silk dragline above the web and descends cautiously, stopping when it is within striking distance of its prey, as modeled in Eq. (7).

Furthermore, PSA also introduces abrupt changes in the solution position to enhance exploration, like the crossover technique in GA which greatly expands exploration of the search space. This exploratory mechanism, inspired by the Portia spider's stalking and striking behavior, is expressed mathematically in Eq. (5). In terms of algorithm parameters, PSA operates without the need for tunable parameters, which simplifies its implementation. On the other hand, SOSA uses three user-controlled parameters, named r_a, p_c, and p_m, to steer the search process.

3 Convergence analysis

In the context of optimization, especially when considering evolutionary algorithms and metaheuristics, demonstrating algorithmic efficiency relies on the application of predefined test cases. Given the intrinsically stochastic attributes of these methodologies, the attainment of optimal outcomes mandates the deployment of a meticulously curated test function suite. It remains quintessential to ascertain that any discerned enhancements are not merely manifestations of stochastic variance. Presently, the field is devoid of a universally endorsed benchmarking paradigm, thus driving scholars to delve into an extensive assortment of test instances. Considering this, the present investigation encompasses an eclectic compilation of test functions, integrating 23 classical test cases and the CEC2017 test functions, with diverse traits, to facilitate an exhaustive appraisal of algorithmic prowess.

3.1 Classical test functions

The efficacy of PSA was thoroughly evaluated using a diverse set of twenty-three test functions (Yao et al. 1999), segmented into three unique categories: unimodal, multimodal, and composite functions, as detailed in Table 3. The unimodal functions, which are defined by a singular global optimum devoid of local optima, serve to gauge the algorithm's proficiency in rapid convergence and effective exploitation. In contrast, the multimodal functions encompass several local optima alongside a global optimum. They present an opportunity to assess the algorithm's capability to navigate beyond local optima and explore the search terrain comprehensively. Lastly, the fixed functions are derived by strategically modifying certain unimodal and multimodal functions through methods such as rotation, shifting, and the introduction of bias. These functions play a crucial role in dissecting PSA's performance under intricate optimization circumstances.

To evaluate the optimization capability of PSA, a cohort of 25 search agents was deployed across 500 iterations, with the aim of identifying the optimal solution that PSA could determine from the group of 23 test functions. The efficacy of PSA was contrasted with several other prominent population-based optimization algorithms, namely SSA, GOA, MFO, and ALO. Given the stochastic nature of these methodologies, each technique was executed 30 times to affirm the reliability and consistency of outcomes. Central statistical metrics, particularly mean values *(avg)* and standard deviations *(std)* were calculated and presented in Table 4. This table offers a comprehensive analysis of the performance metrics for each algorithm.

Table 3 23 classical benchmark test function data

Type	Function	Range	Dim	fmin
Uni-modal	$f1(x) = \sum_{i=1}^{n} x_i^2$	[− 100,100]	10	0
Uni-modal	$f2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[− 10,10]	10	0
Uni-modal	$f3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$	[− 100,100]	10	0
Uni-modal	$f4(x) = max\{\lvert x_i \rvert, 1 \le i \le n\}$	[− 100,100]	10	0
Uni-modal	$f5(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2 \right]$	[− 30,30]	10	0
Uni-modal	$f6(x) = \sum_{i=1}^{n} (\lvert x_i + 0.5 \rvert)^2$	[− 100,100]	10	0
Uni-modal	$f7(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	[− 1.28,1.28]	10	0
Multi-modal	$f8(x) = \sum_{i=1}^{n} - x_i \sin\left(\sqrt{\lvert x_i \rvert} \right)$	[− 500,500]	10	
Multi-modal	$f9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	[− 5.12,5.12]	10	0
Multi-modal	$f10(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} \right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e$	[− 32,32]	10	0
Multi-modal	$f11(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	[− 600,600]	10	0
Multi-modal	$f12(x) = \frac{\pi}{n}\left\{ 10\sin^2(\pi y_1) + \sum_{i=1}^{n} (y_i - 1)^2 \left[1 + 10\sin^2(\pi y_{i+1}) \right] + (y_n - 1)^2 + \sum_{i=1}^{n} u(x_i, 10,100,4) \right\}$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m x_i > a \\ 0 - a < x_i < a \\ k(-x_i - a)^m x_i < -a \end{cases}$	[− 50,50]	10	0
Multi-modal	$f13(x) = 0.1\left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2 \left[1 + \sin^2(3\pi x_i + 1) \right] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} + \sum_{i=1}^{n} u(x_i, 5,100,4)$	[− 50,50]	10	0
Fixed	$f14(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6} \right)^{-1}$	[− 65, 65]	2	1

Table 3 (continued)

Type	Function	Range	Dim	fmin
Fixed	$f15(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$	[− 5,5]	4	0.0003
Fixed	$f16(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	[− 5,5]	2	− 1.0316
Fixed	$f17(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_1 + 10$	[− 5,5]	2	0.398
Fixed	$f18(x) = \left[1 + \left(x_1 + x_2 + 1\right)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right]$ $\times\left[30 + (2x_1 - 3x_2)^2 \times \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]$	[− 2,2]	2	3
Fixed	$f19(x) = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	[0,1]	3	− 3.86
Fixed	$f20(x) = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	[0,1]	6	− 3.32
Fixed	$f21(x) = -\sum_{i=1}^{5}\left[\left(X - a_i\right)\left(X - a_i\right)^T + c_i\right]^{-1}$	[0,10]	4	− 10.1532
Fixed	$f22(x) = -\sum_{i=1}^{7}\left[\left(X - a_i\right)\left(X - a_i\right)^T + c_i\right]^{-1}$	[0,10]	4	− 10.4028
Fixed	$f23(x) = -\sum_{i=1}^{10}\left[\left(X - a_i\right)\left(X - a_i\right)^T + c_i\right]^{-1}$	[0,10]	4	− 10.5363

From the data presented in Table 4, PSA surpasses other metaheuristic techniques when evaluated on unimodal test functions. Specifically, PSA demonstrates a remarkable aptitude for exploitation, consistently surpassing benchmark techniques such as SSA, GOA, MFO, and ALO. This underscores the prowess of PSA in tackling unimodal optimization tasks. Further, within the realm of multimodal functions, the performance of PSA is exemplary, often besting other competing techniques. This is indicative of its superior ability to comprehensively explore the search terrain, thereby avoiding premature convergence to local optima. Upon delving into the fixed function tests, the performance of PSA aligns closely with the average achievements of other algorithms. It is noteworthy that, in the context of functions *f14* to *f21*, PSA remains competitive with SSA, GOA, MFO, and ALO. However, its superiority becomes more pronounced for functions *f22*, and *f23*, where PSA emerges as the leader. Collectively, these results underscore the commendable expertise of PSA when juxtaposed against other contemporary optimization methods.

To assess the statistical significance of performance disparities between the PSA and other well-established algorithms on classical test functions, a t-test was conducted with a 95% confidence level. The outcomes of this t-test are detailed in Table 5. In this table, a "+" symbol indicates that the PSA holds a statistically significant advantage over the comparison method at the 95% confidence level, a "-" symbol shows that the PSA performs significantly worse, and "≈" denotes no significant difference in performance between the techniques. It is revealed by the data in Table 5 that superior performance is exhibited by the PSA across most of the classical test cases when compared to other techniques such as GOA, SSA, MFO, and ALO.

Additional evaluation measures, such as the convergence curve, mean fitness across all solutions, trajectory of the first solution, and search history, were employed to assess the effectiveness of PSA. For this analysis, 250 iterations with 20 search agents were used across three categories of test functions: unimodal (*f1*, *f4*, *f7*), multimodal (*f9*, *f12*), and specific functions (*f15*, *f17*, *f23*), as presented in Fig. 6. Upon examining the convergence graph and average fitness value, a discernible decline in the quality of search agents emerges as the iterations advance. This trend highlights the capacity of PSA to improve the quality of initially randomized solutions tailored to specific optimization challenges. The trajectory of the first solution underscores the adeptness of PSA in ensuring convergence during a localized search. This proficiency is further illuminated by marked variations in mean fitness value during the discovery phase, followed by minor adjustments in the refinement phase (Van den Bergh and Engelbrecht 2006). Furthermore, the search history associated with these functions underscores the robust efficiency of PSA in navigating the exploration domain and pinpointing areas of promise. Figure 7 subsequently delves into the convergence patterns observed in the 23 classical test functions, analysed over a span of 150 iterations with the assistance of 20 search agents. From this data, the superiority of PSA becomes clear, especially when benchmarked against other techniques such as SSA, GOA, MFO, and ALO. PSA consistently demonstrates superior convergence performance primarily in the considered test functions.

3.2 CEC2017 benchmark test functions

The CEC2017 test functions comprise a set of benchmark functions introduced during the 2017 IEEE Congress on Evolutionary Computation (CEC) competition, with a primary focus on real-parameter optimization. Held in high regard within the evolutionary computation community and associated domains, these benchmarks serve as instrumental tools

Table 4 Results of different algorithms on classical benchmark test functions

Alg./Func.	PSA		GOA		SSA		MFO		ALO	
	avg	*std*	*avg*	*std*	*avg*	*std*	*avg*	*std*	*avg*	*std*
f_1	2.8188E-57	4.360E-57	1.744E-07	1.280E-07	1.617E-09	7.445E-10	1.002E-04	2.334E-04	1.504E-06	1.583E-06
f_2	2.5402E-37	2.295E-37	2.310E+00	2.455E+00	8.223E-02	2.620E-01	3.041E+00	4.646E+00	4.502E+00	9.160E+00
f_3	8.2036E-31	1.072E-30	1.972E-03	8.447E-03	9.279E-01	1.945E+00	1.103E+03	2.346E+03	4.979E+01	6.376E+01
f_4	6.6946E-22	7.551E-22	1.503E-03	4.178E-03	5.856E-02	2.626E-01	2.289E+01	1.106E+01	1.314E+00	2.124E+00
f_5	6.8011E+00	2.267E-01	1.638E+03	2.226E+03	2.063E+02	3.637E+02	3.480E+03	1.609E+04	1.535E+02	3.230E+02
f_6	2.1508E-01	9.315E-02	1.596E-07	2.910E-07	1.542E-09	5.850E-10	1.228E-04	2.070E-04	2.367E-06	2.692E-06
f_7	3.6030E-04	1.472E-04	1.213E-01	2.402E-01	2.706E-02	1.597E-02	1.276E-01	4.792E-01	6.527E-02	3.856E-02
f_8	− 2.8765E+03	1.708E+02	− 1.435E+03	2.626E+02	− 2.736E+03	3.370E+02	− 2.813E+03	2.337E+02	-2.267E+03	5.230E+02
f_9	0.0000E+00	0.000E+00	1.343E+01	8.028E+00	1.592E+01	6.773E+00	3.041E+01	1.276E+01	2.547E+01	1.107E+01
f_{10}	1.4803E-15	1.324E-15	9.947E-01	9.483E-01	1.028E+00	9.968E-01	1.353E+00	3.362E+00	9.338E-01	1.055E+00
f_{11}	0.0000E+00	0.000E+00	2.027E-01	9.295E-02	1.991E-01	1.814E-01	2.034E-01	1.587E-01	1.679E-01	9.656E-02
f_{12}	5.2926E-02	9.201E-03	8.770E-02	2.259E-01	1.242E+00	1.210E+00	8.802E-01	1.357E+00	6.194E+00	6.371E+00
f_{13}	9.8420E-02	3.757E-02	6.343E-03	1.845E-02	9.142E-03	1.792E-02	2.194E+01	9.485E+01	6.182E-03	7.175E-03
f_{14}	9.9800E-01	3.190E-10	2.116E+00	1.985E+00	1.923E+00	1.112E+00	4.714E+00	4.446E+00	4.072E+00	3.692E+00
f_{15}	6.7415E-04	1.394E-04	1.628E-02	2.532E-02	3.025E-03	5.900E-03	2.655E-03	5.209E-03	2.863E-03	4.982E-03
f_{16}	− 1.0316E+00	2.218E-08	− 1.032E+00	1.716E-12	− 1.032E+00	5.466E-14	− 1.032E+00	0.000E+00	− 1.032E+00	2.506E-13
f_{17}	3.9789E-01	6.336E-07	3.979E-01	1.228E-11	4.029E-01	5.365E-03	3.979E-01	1.110E-16	3.979E-01	2.119E-13
f_{18}	3.0000E+00	1.442E-08	5.700E+00	1.454E+01	3.000E+00	6.429E-13	3.000E+00	5.408E-15	3.000E+00	1.909E-12
f_{19}	− 3.8628E+00	1.502E-09	− 3.605E+00	3.303E-01	− 3.863E+00	1.214E-05	− 3.862E+00	2.364E-03	− 3.863E+00	1.819E-10
f_{20}	− 3.3220E+00	1.332E-15	− 3.291E+00	5.676E-02	− 3.168E+00	2.026E-02	− 3.278E+00	5.729E-02	− 3.218E+00	4.720E-02
f_{21}	− 9.6683E+00	1.110E+00	− 6.223E+00	3.337E+00	− 6.143E+00	3.392E+00	− 6.308E+00	3.274E+00	− 5.300E+00	3.093E+00
f_{22}	− 9.7109E+00	1.600E+00	− 6.343E+00	3.829E+00	− 6.324E+00	3.615E+00	− 6.167E+00	3.518E+00	− 5.766E+00	3.162E+00
f_{23}	− 9.7145E+00	2.108E+00	− 4.345E+00	3.151E+00	− 6.777E+00	3.814E+00	− 6.974E+00	3.616E+00	− 6.060E+00	3.272E+00

Table 5 T-test results on 23 classical test functions

Alg./ Func.	PSA/GOA	PSA/SSA	PSA/MFO	PSA/ALO
f_1	+	+	+	+
f_2	+	≈	+	+
f_3	≈	+	+	+
f_4	+	≈	+	+
f_5	+	+	≈	+
f_6	+	−	+	+
f_7	+	+	≈	+
f_8	+	≈	+	+
f_9	+	+	+	+
f_{10}	+	+	+	+
f_{11}	+	+	+	+
f_{12}	≈	+	+	+
f_{13}	+	+	≈	−
f_{14}	+	+	+	+
f_{15}	+	+	≈	+
f_{16}	+	+	−	+
f_{17}	+	+	−	+
f_{18}	≈	+	−	+
f_{19}	+	+	≈	−
f_{20}	+	+	+	+
f_{21}	+	+	+	+
f_{22}	+	+	+	+
f_{23}	+	+	+	+

for gauging and contrasting the effectiveness of optimization techniques. Building upon the benchmark suites from preceding years, the CEC2017 collection has been meticulously tailored to offer a diverse range of challenges to optimization techniques. Notably, these functions are perceived as more realistic compared to the 23 traditional benchmark functions. Their comprehensive scope includes both unimodal and multi-modal domains, embracing both separable and non-separable functions. Furthermore, they incorporate shifted and rotated variations, providing a thorough testbed for algorithms. Such a diverse set of test cases allows researchers to discern the merits and demerits of various optimization techniques across different settings.

In line with this discussion, the effectiveness of PSA is validated using the IEEE CEC2017 (Wu et al. 2017) test suites. These suites are primarily classified into four different classifications: unimodal, multimodal, hybrid, and composition. Table 6 delineates the detailed definitions of the CEC2017 benchmark problems. To rigorously evaluate the capability of the PSA in handling complex optimization problems, all functions within the CEC2017 suite were configured to have 30 dimensions. This increase in dimensionality presents a more challenging scenario, aimed at thoroughly testing the algorithm's proficiency in navigating and optimizing within a high-dimensional search space.

Table 7 presents comprehensive statistical results comparing the PSA with other swarm-based methods, including GOA, SSA, MFO, and ALO. To guarantee a thorough and unbiased assessment, each technique was subjected to 30 runs for every benchmark

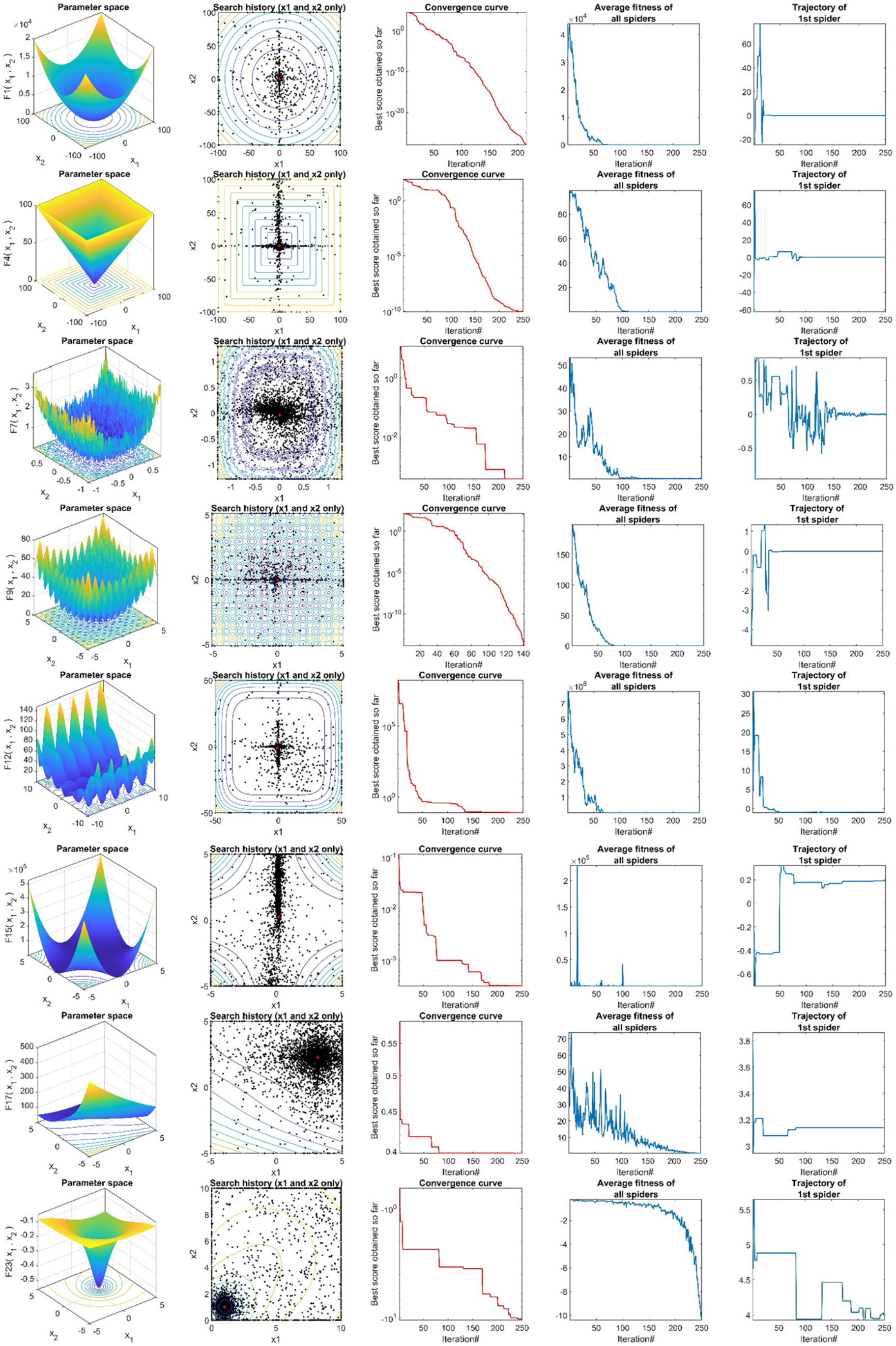


Fig. 6 Search history, convergence curve, average fitness of all solutions and trajectory of the first solution

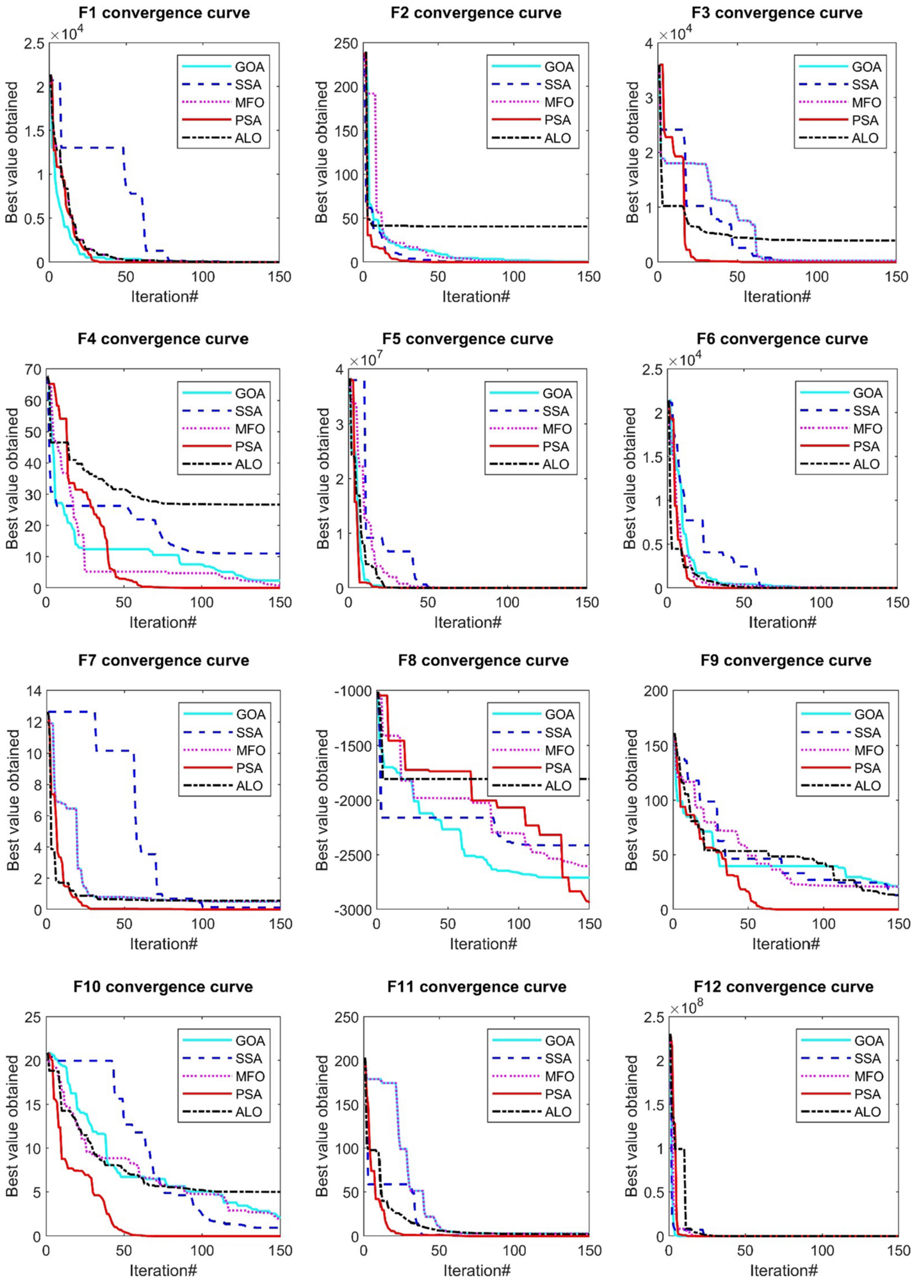


Fig. 7 Convergence behavior of PSA, GOA, SSA, MFO, and ALO for classical test functions

function. Following these simulations, statistical evaluations were conducted to ascertain both the mean values (*avg*) and the standard deviations (*std*) from these trials. For this study's framework, 50 search agents were employed, with each confined to a limit of 400 iterations. An examination of the data in Table 7 clearly demonstrates the outstanding

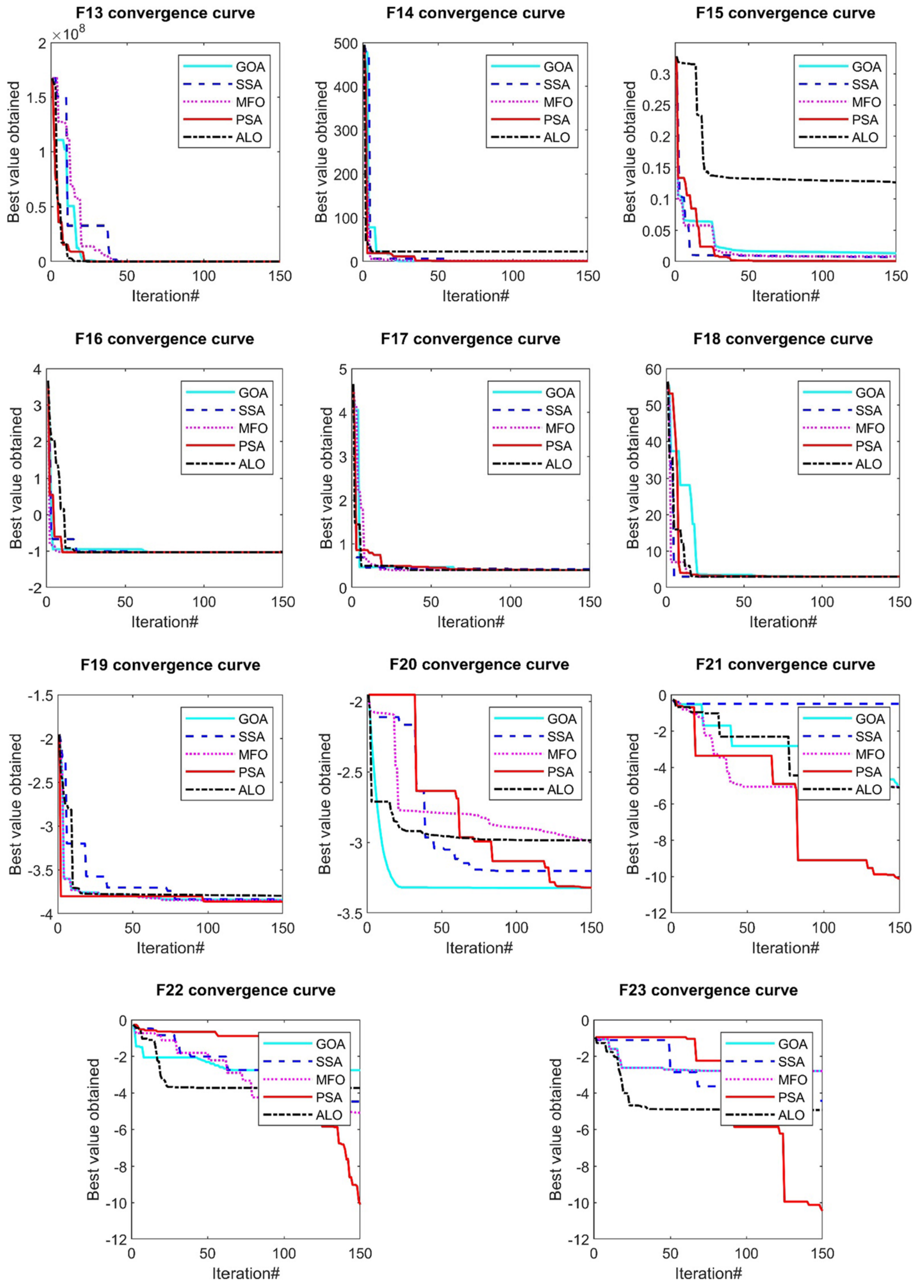


Fig. 7 (continued)

effectiveness of PSA over other techniques like GOA, SSA, MFO, and ALO, particularly in the multimodal, hybrid, and composition benchmark domains.

A t-test, conducted with a 95% confidence level, was employed to evaluate the statistical significance of performance disparities between the PSA and other established algorithms

Table 6 CEC2017 benchmark function data

Function	Type	Name	Range	n	Fmin
*f*1	Unimodal	Shifted and Rotated Bent Cigar Function	[– 100, 100]	30	100
*f*2	Unimodal	Shifted and Rotated Zakharov Function	[– 100, 100]	30	200
*f*3	Multimodal	Shifted and Rotated Rosenbrock's Function	[– 100, 100]	30	300
*f*4	Multimodal	Shifted and Rotated Rastrigin's Function	[– 100, 100]	30	400
*f*5	Multimodal	Shifted and Rotated Expanded Scaffer's *F7* Function	[– 100, 100]	30	500
*f*6	Multimodal	Shifted and Rotated Lunacek Bi_Rastrigin Function	[– 100, 100]	30	600
*f*7	Multimodal	Shifted and Rotated Non-Continuous Rastrigin's Function	[– 100, 100]	30	700
*f*8	Multimodal	Shifted and Rotated Levy Function	[– 100, 100]	30	800
*f*9	Multimodal	Shifted and Rotated Schwefel's Function	[– 100, 100]	30	900
*f*10	Hybrid	Hybrid Function 1 ($N=3$)	[– 100, 100]	30	1000
*f*11	Hybrid	Hybrid Function 2 ($N=3$)	[-100, 100]	30	1100
*f*12	Hybrid	Hybrid Function 3 ($N=3$)	[– 100, 100]	30	1200
*f*13	Hybrid	Hybrid Function 4 ($N=4$)	[– 100, 100]	30	1300
*f*14	Hybrid	Hybrid Function 5 ($N=4$)	[– 100, 100]	30	1400
*f*15	Hybrid	Hybrid Function 6 ($N=4$)	[– 100, 100]	30	1500
*f*16	Hybrid	Hybrid Function 7 ($N=5$)	[– 100, 100]	30	1600
*f*17	Hybrid	Hybrid Function 8 ($N=5$)	[– 100, 100]	30	1700
*f*18	Hybrid	Hybrid Function 9 ($N=5$)	[– 100, 100]	30	1800
*f*19	Hybrid	Hybrid Function 10 ($N=6$)	[– 100, 100]	30	1900
*f*20	Composition	Composition Function 1 ($N=3$)	[– 100, 100]	30	2000
*f*21	Composition	Composition Function 2 ($N=3$)	[– 100, 100]	30	2100
*f*22	Composition	Composition Function 3 ($N=4$)	[– 100, 100]	30	2200
*f*23	Composition	Composition Function 4 ($N=4$)	[– 100, 100]	30	2300
*f*24	Composition	Composition Function 5 ($N=5$)	[– 100, 100]	30	2400
*f*25	Composition	Composition Function 6 ($N=5$)	[– 100, 100]	30	2500
*f*26	Composition	Composition Function 7 ($N=6$)	[– 100, 100]	30	2600
*f*27	Composition	Composition Function 8 ($N=6$)	[– 100, 100]	30	2700
*f*28	Composition	Composition Function 9 ($N=3$)	[– 100, 100]	30	2800
*f*29	Composition	Composition Function 10 ($N=3$)	[– 100, 100]	30	2900

on the CEC2017 test functions, similar to the methodology applied to classical test cases. The results of this t-test are detailed in Table 8. The data in Table 8 reveal that the PSA exhibits superior performance across most of the CEC2017 test cases when compared to other techniques such as GOA, SSA, and ALO. Notably, the PSA outperformed the MFO in all CEC2017 test cases.

4 Engineering optimization problems

The objective of this section is to thoroughly evaluate the effectiveness of the PSA by applying it to five practical engineering optimization challenges, each distinguished by multiple inequality constraints. The overarching goal is to ascertain the algorithm's adeptness in handling these constraints throughout the entire optimization process.

Table 7 Results of different algorithms on classical benchmark test functions

Alg./Func.	PSA		GOA		SSA		MFO		ALO	
	avg	*std*	*avg*	*std*	*avg*	*std*	*avg*	*std*	*avg*	*std*
f_1	2.887E+09	8.253E+08	2.935E+10	9.392E+09	7.120E+08	2.030E+09	8.394E+10	8.394E+10	2.050E+04	3.141E+04
f_2	1.825E+04	7.103E+03	6.252E+04	1.612E+04	1.682E+04	1.825E+04	1.657E+05	1.657E+05	1.031E+04	3.621E+03
f_3	5.974E+02	6.516E+01	4.953E+03	2.100E+03	1.883E+03	5.974E+02	2.549E+04	2.549E+04	5.036E+02	3.867E+01
f_4	4.025E+03	1.447E+03	3.843E+04	1.224E+04	1.397E+04	4.025E+03	8.807E+04	8.807E+04	9.193E+02	1.563E+02
f_5	5.000E+02	2.034E-04	5.000E+02	7.122E-03	5.000E+02	5.000E+02	5.001E+02	5.001E+02	5.000E+02	1.699E-03
f_6	1.359E+04	2.361E+03	1.619E+04	1.072E+04	5.054E+03	1.359E+04	6.753E+04	6.753E+04	8.602E+03	3.776E+03
f_7	7.001E+02	2.576E-02	7.010E+02	7.088E-01	7.003E+02	7.001E+02	7.053E+02	7.053E+02	7.003E+02	1.596E-01
f_8	8.064E+02	1.448E+00	8.274E+02	7.464E+00	8.136E+02	8.064E+02	8.773E+02	8.773E+02	8.137E+02	3.192E+00
f_9	4.219E+03	2.692E+02	6.558E+03	5.390E+02	6.282E+03	4.219E+03	7.962E+03	7.962E+03	5.969E+03	5.300E+02
f_{10}	9.410E+04	1.254E+04	1.691E+05	6.910E+04	1.021E+06	9.410E+04	4.552E+07	4.552E+07	1.702E+05	3.922E+04
f_{11}	2.747E+07	1.213E+07	1.023E+09	8.102E+08	1.349E+09	2.747E+07	8.470E+09	8.470E+09	4.699E+07	1.684E+07
f_{12}	7.956E+06	5.527E+06	9.781E+08	1.008E+09	5.227E+08	7.956E+06	1.139E+10	1.139E+10	3.030E+06	2.522E+06
f_{13}	2.607E+05	5.950E+04	2.614E+05	1.148E+05	6.015E+06	2.607E+05	1.056E+07	1.056E+07	3.220E+06	1.514E+06
f_{14}	5.346E+05	2.342E+05	1.016E+08	1.018E+08	5.858E+08	5.346E+05	5.576E+09	5.576E+09	5.717E+05	1.590E+05
f_{15}	5.504E+04	2.115E+04	6.096E+05	1.055E+06	7.595E+06	5.504E+04	1.862E+09	1.862E+09	3.693E+05	4.069E+05
f_{16}	3.559E+04	8.119E+03	8.066E+04	4.402E+04	4.840E+12	3.559E+04	1.008E+12	1.008E+12	6.552E+04	1.971E+04
f_{17}	8.815E+04	1.531E+04	6.665E+04	2.107E+04	1.744E+06	8.815E+04	6.324E+06	6.324E+06	1.680E+05	1.811E+04
f_{18}	2.734E+06	4.700E+06	5.818E+09	7.851E+09	1.578E+08	2.734E+06	6.497E+13	6.497E+13	1.076E+05	1.821E+04
f_{19}	2.564E+03	1.339E+02	6.644E+03	1.782E+03	1.204E+04	2.564E+03	1.860E+04	1.860E+04	7.037E+03	9.825E+02
f_{20}	3.496E+03	4.091E+02	2.754E+04	9.721E+03	1.278E+04	3.496E+03	7.178E+04	7.178E+04	2.800E+03	1.850E+02
f_{21}	2.345E+03	1.350E+01	3.575E+03	9.433E+02	3.419E+03	2.345E+03	3.593E+03	3.593E+03	2.499E+03	4.231E+01
f_{22}	8.992E+03	2.316E+03	3.650E+04	9.185E+03	1.572E+04	8.992E+03	4.797E+04	4.797E+04	3.616E+03	2.217E+03
f_{23}	7.078E+03	1.147E+03	2.187E+04	6.374E+03	1.116E+04	7.078E+03	2.909E+04	2.909E+04	2.678E+03	3.462E+02
f_{24}	3.013E+03	2.996E+01	4.386E+03	6.120E+02	3.723E+03	3.013E+03	1.273E+04	1.273E+04	3.000E+03	3.095E+01
f_{25}	3.471E+03	3.749E+01	4.051E+03	1.043E+03	9.565E+03	3.471E+03	3.666E+03	3.666E+03	3.521E+03	4.293E+01

Table 7 (continued)

Alg./Func.	PSA		GOA		SSA		MFO		ALO	
	avg	*std*	*avg*	*std*	*avg*	*std*	*avg*	*std*	*avg*	*std*
f_{26}	3.173E+03	1.382E+01	3.551E+03	2.895E+02	4.004E+03	3.173E+03	3.354E+03	3.354E+03	3.427E+03	7.539E+01
f_{27}	3.216E+03	2.866E+01	4.267E+03	6.600E+02	5.180E+03	3.216E+03	4.561E+03	4.561E+03	3.208E+03	1.195E+01
f_{28}	3.638E+07	2.114E+07	5.219E+08	6.265E+08	9.082E+10	3.638E+07	7.150E+11	7.150E+11	3.681E+07	1.220E+08
f_{29}	6.882E+07	5.754E+07	2.872E+09	4.054E+09	2.990E+11	6.882E+07	1.551E+11	1.551E+11	8.511E+07	9.007E+07

Table 8 T-test results on CEC2017 test functions

Alg./Func.	PSA/GOA	PSA/SSA	PSA/MFO	PSA/ALO
f_1	+	+	+	−
f_2	+	≈	+	−
f_3	+	+	+	−
f_4	+	+	+	−
f_5	+	+	+	+
f_6	≈	-	+	+
f_7	+	+	+	+
f_8	+	+	+	+
f_9	+	+	+	+
f_{10}	+	≈	+	+
f_{11}	+	+	+	+
f_{12}	+	≈	+	−
f_{13}	≈	+	+	+
f_{14}	+	+	+	≈
f_{15}	+	≈	+	+
f_{16}	+	≈	+	+
f_{17}	-	+	+	+
f_{18}	+	≈	+	−
f_{19}	+	+	+	+
f_{20}	+	+	+	−
f_{21}	+	+	+	+
f_{22}	+	+	+	−
f_{23}	+	+	+	−
f_{24}	+	+	+	−
f_{25}	+	+	+	+
f_{26}	+	+	+	+
f_{27}	+	+	+	−
f_{28}	+	≈	+	≈
f_{29}	+	≈	+	≈

4.1 Tension string design task

The primary objective of this task is to minimize the weight of the tension string. Key variables in this pursuit include the wire diameter (d), the mean coil diameter (D), and the number of active coils (n), each clearly illustrated in Fig. 8. A detailed mathematical representation of this challenge is provided as:

Consider:

$$\vec{x} = \left[x_1 x_2 x_3\right] = [dDN]$$

Minimize:

$$f(\vec{x}) = \left(x_3 + 2\right) x_2 x_1^2$$

Subject to:

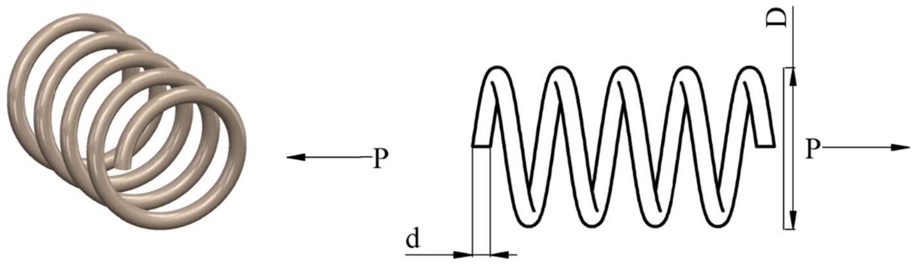


Fig. 8 Tension string design task

$$g_1(\vec{x}) = 1 - \frac{x_2^3 x_3}{71,785 x_1^4} \le 0$$

$$g_2(\vec{x}) = \frac{4x_2^2 - x_1 x_2}{12,566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} \le 0$$

$$g_3(\vec{x}) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \le 0$$

$$g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \le 0$$

Variable range:

$$0.05 \le x_1 \le 2.00$$

$$0.25 \le x_2 \le 1.30$$

$$2.00 \le x_3 \le 15.0$$

Results from the comparative analysis of PSA with established algorithms, in the context of the best outcome, are presented in Table 9. Examination of this data suggests that PSA exhibits enhanced performance relative to various methods, including ES (Mezura-Montes and Coello 2008), HS (Chakraborty et al. 2009), RO (Kaveh and Khayatazad 2012), WOA (Mirjalili and Lewis 2016), RFO (Połap and Woźniak 2021), SCSO (Seyyedabbasi and Kiani 2023), SSA (Mirjalili et al. 2017), and MVO (Mirjalili et al. 2016). This finding reinforces the notion that the newly introduced PSA, in conjunction with OBSCA (Abd Elaziz et al. 2017), and GMO (Rezaei et al. 2023), holds considerable practical significance in engineering applications. Figure 9 illustrates the best solutions attainable by different algorithms. Based on this figure, it is evident that PSA demonstrates notable efficacy, positioning it competitively when compared to other cutting-edge algorithms.

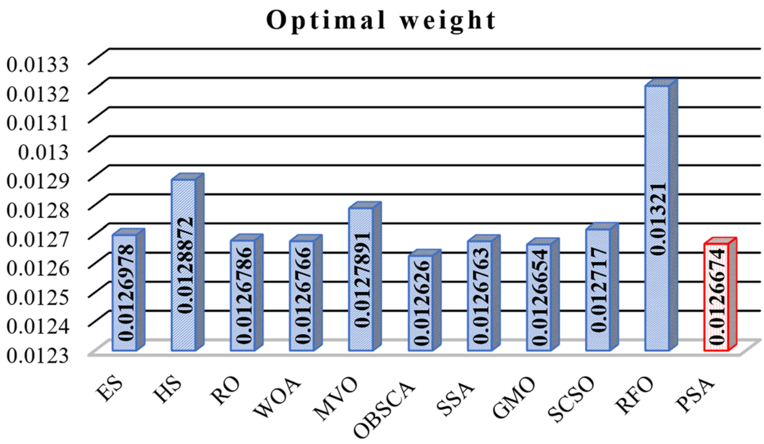


Fig. 9 Comparison of optimal outcomes in tension string design task

Table 9 The best design of tension string design task obtained by different algorithms

Optimization technique	Optimal parameters			Optimal weight
	h	*l*	*t*	
ES (Mezura-Montes and Coello 2008)	0.051643	0.35536	11.39793	0.0126978
HS (Chakraborty et al. 2009)	0.051154	0.349871	12.07643	0.0128872
RO (Kaveh and Khayatazad 2012)	0.05137	0.349096	11.76279	0.0126786
WOA (Mirjalili and Lewis 2016)	0.051207	0.345215	12.00403	0.0126766
MVO (Mirjalili et al. 2016)	0.05251	0.37602	10.33513	0.0127891
OBSCA (Abd Elaziz et al. 2017)	0.0523	0.31728	12.54854	0.012626
SSA (Mirjalili et al. 2017)	0.051207	0.345215	12.004032	0.0126763
GMO (Rezaei et al. 2023)	0.051792	0.359198	11.145041	0.0126654
SCSO (Seyyedabbasi and Kiani 2023)	0.0500	0.3175	14.0200	0.0127170
RFO (Połap and Woźniak 2021)	0.05189	0.36142	11.58436	0.01321
PSA (This study)	0.205341	3.261517	9.034808	0.0126674

4.2 Cantilever beam design task

This task examines a cantilever beam composed of five hollow square blocks. The first block is firmly anchored, while the fifth block is subjected to a vertical load. The primary goal is to minimize the beam's weight. Figure 10 depicts the five parameters that define the cross-sectional shape of the blocks. Detailed formulations for this problem are provided as:

Consider:

$$\vec{x} = [x_1 x_2 x_3 x_4] = [T_s T_h R L]$$

Minimize:

$$f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

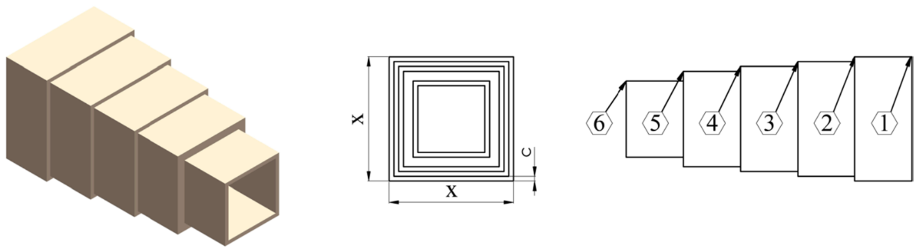


Fig. 10 Cantilever beam design task

Subject to:

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \le 0$$

$$g_2(\vec{x}) = -x_2 + 0.00954x_3 \le 0$$

$$g_3(\vec{x}) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \le 0$$

$$g_4(\vec{x}) = x_4 - 240 \le 0$$

Variable range:

$$0 \le x_1, x_2 \le 99$$

$$10 \le x_3, x_4 \le 200$$

The results from this study have been rigorously analyzed and are presented in Table 10. Upon detailed examination of the data, it is evident that PSA consistently delivers results commensurate with, if not superior to, cutting-edge optimization methods such as COA (Jia et al. 2023), RFO (Połap and Woźniak 2021), GOA (Saremi et al. 2017), MVO (Mirjalili et al. 2016), ALO (Mirjalili 2015a), MMA (Chickermane and Gea 1996), GCA_I (Chickermane and Gea 1996), GCA_II (Chickermane and Gea 1996), CS (Gandomi et al. 2013), and SOS (Cheng and Prayogo 2014). Figure 11 offers a comprehensive visualization of the optimal solutions derived from various algorithms. The data underscores PSA's distinct edge over its contemporaries. These findings underscore the unparalleled efficacy of PSA in addressing and refining complex, constraint-laden challenges. Furthermore, this analysis accentuates the relevance of PSA in engineering and allied fields, demonstrating its adeptness at tackling sophisticated problem structures.

4.3 Gear train design task

The objective of the gear train design task, as depicted in Fig. 12, is to secure the minimal gear ratio by optimizing four discrete parameters: the total of teeth on each gear, denoted as n_A, n_B, n_C, and n_D. The gear ratio establishes the correlation between the angular velocities of the output and input shafts. Each parameter, being intrinsically discrete, increments by one unit. Central to the problem formulation is the task of constraining the feasible

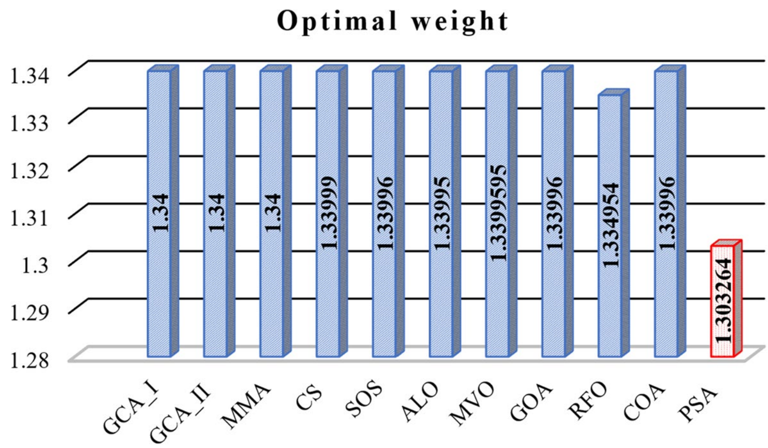


Fig. 11 Comparison of optimal outcomes in cantilever beam design task

domains of these variables. A detailed mathematical representation of this task is provided as follows:

Consider:

$$\vec{x} = [x_1 x_2 x_3 x_4] = [n_A n_B n_C n_D]$$

Minimize:

$$f(\vec{x}) = \left(\frac{1}{6.931} - \frac{x_2 x_3}{x_1 x_4}\right)^2$$

Variable range:

$$12 \leq x_1, x_2, x_3, x_4 \leq 60$$

Table 11 offers a detailed comparison of PSA with several esteemed optimization techniques. A thorough analysis of the data underscores the congruence between solutions obtained through PSA and those from leading optimization methodologies, including MVO (Mirjalili et al. 2016), ABC (Sadollah et al. 2013), MBA (Sadollah et al. 2013), CS (Gandomi et al. 2013), ISA (Gandomi 2014), as well as frameworks proposed by Kannan and Kramer (1994), as well as Deb and Goyal (1996). These results compellingly validate the proposition that PSA can not only match but potentially surpass contemporary optimization strategies, particularly in scenarios rife with discrete parameters. PSA's adeptness in handling discrete parameters accentuates its versatility, further cementing its position as an essential tool for handling diverse optimization tasks across various domains.

4.4 Welded beam design task

In the design task presented in Fig. 13, the system is defined by several crucial structural parameters. The design specifically pivots around four primary variables: the length of the attached bar (l), weld thickness (h), bar thickness (b), and bar height (t). The predominant objective of this engineering pursuit is to minimize the fabrication costs inherent to the design

Table 10 The best design of cantilever beam design task obtained by different algorithms

Optimization technique	Optimal parameters					Optimal weight
	x_1	x_2	x_3	x_4	x_5	
GCA_I (Chickermane and Gea 1996)	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA_II (Chickermane and Gea 1996)	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
MMA (Chickermane and Gea 1996)	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
CS (Gandomi et al. 2013)	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
SOS (Cheng and Prayogo 2014)	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996
ALO (Mirjalili 2015a)	6.01812	5.31142	4.48836	3.49751	2.158329	1.33995
MVO (Mirjalili et al. 2016)	6.023940221548	5.30601123355	4.4950113234	3.4960223242	2.15272617	1.3399595
GOA (Saremi et al. 2017)	6.011674	5.31297	4.48307	3.50279	2.16333	1.33996
RFO (Połap and Woźniak 2021)	6.00845	5.30485	4.49215	3.4984	2.14463	1.334954
COA (Jia et al. 2023)	6.017257314	5.307150983	4.491255551	3.508156789	2.149913022	1.33996
PSA (This study)	5.998361	4.871582	4.461329	3.471494	2.136561	1.303264

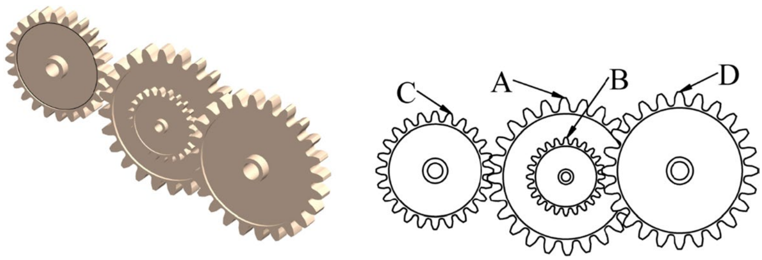


Fig. 12 Gear train design task

Table 11 The best design of gear train design task obtained by different algorithms

Optimization technique	Optimal parameters				Optimal gear ratio
	n_A	n_B	n_C	n_D	
Kannan and Kramer (1994)	33	15	13	41	2.1469E-08
Deb and Goyal (1996)	49	16	19	43	2.7019E-12
MBA (Sadollah et al. 2013)	43	16	19	49	2.7009E-12
ABC (Sadollah et al. 2013)	49	16	19	43	2.7009E-12
CS (Gandomi et al. 2013)	43	16	19	49	2.7009E-12
ISA (Gandomi 2014)	N/A	N/A	N/A	N/A	2.7009E-12
MVO (Mirjalili et al. 2016)	43	16	19	49	2.7009E-12
PSA (This study)	43	16	19	49	2.7009E-12

of this apparatus. For the design to maintain both its practicality and structural integrity, adherence to seven essential constraints is vital, particularly when a load is applied to the bar. These constraints encompass a variety of considerations, including lateral constraints, end deflection of the beam (δ), shear stress (τ), bending stress exhibited in the beam (θ), and the buckling load sustained by the bar (P_c). A comprehensive mathematical exposition of this complex challenge is provided as:

Consider:

$$\vec{x} = \left[x_1 x_2 x_3 x_4\right] = [hltb]$$

Minimize:

$$f\left(\vec{x}\right) = 1.10471x_1^2 x_2 + 0.04811x_3 x_4 (14.0 + x_2)$$

Subject to:

$$g_1\left(\vec{x}\right) = \tau\left(\vec{x}\right) - \tau_{max} \leq 0$$

$$g_2\left(\vec{x}\right) = \sigma\left(\vec{x}\right) - \sigma_{max} \leq 0$$

$$g_3\left(\vec{x}\right) = \delta\left(\vec{x}\right) - \delta_{max} \leq 0$$

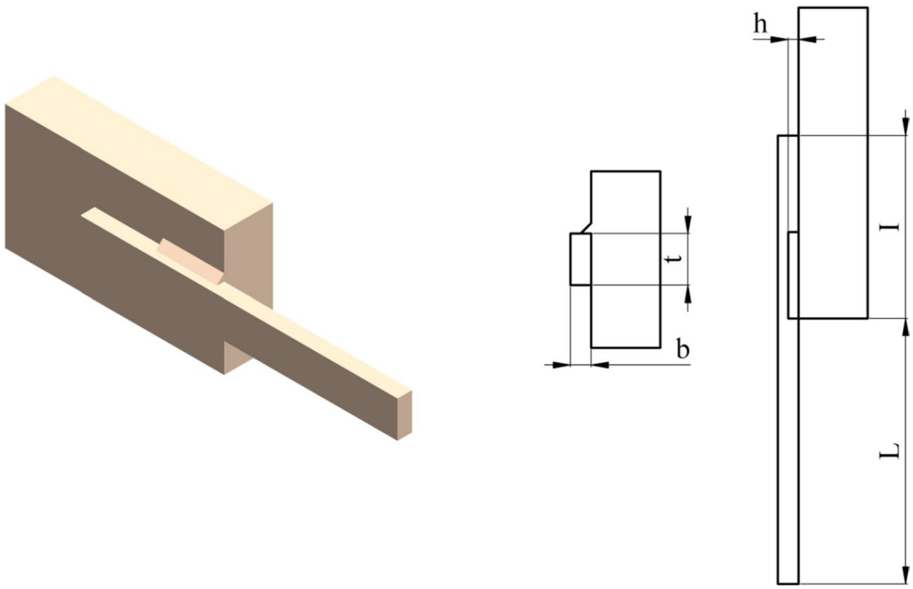


Fig. 13 Welded beam design task

$$g_4(\vec{x}) = x_1 - x_4 \leq 0$$

$$g_5(\vec{x}) = P - P_c(\vec{x}) \leq 0$$

$$g_6(\vec{x}) = 0.125 - x_1 \leq 0$$

$$g_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5.0 \leq 0$$

Variable range:

$$0.1 \leq x_1, x_4 \leq 2$$

$$0.1 \leq x_2, x_3 \leq 10$$

where:

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + \frac{2\tau'\tau''x_2}{2R} + (\tau'')^2}, \tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}$$

$$M = P\left(L + \frac{x_2}{2}\right), R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

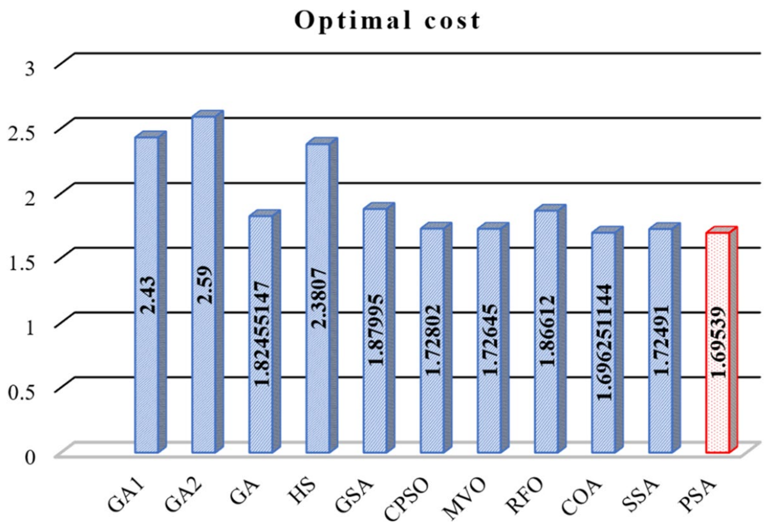


Fig. 14 Comparison of optimal outcomes in welded beam design task

$$\sigma(\vec{x}) = \frac{6PL}{x_4 x_3^2}, \delta(\vec{x}) = \frac{6PL^3}{Ex_3^2 x_4}, P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P = 6000lb, L = 14in., \delta_{max} = 0.25in., E = 30 \times 10^6 psi, G = 12 \times 10^6 psi,$$

$$\tau_{max} = 13,600psi, \sigma_{max} = 30,000psi$$

Table 12 conducts a detailed comparative analysis of PSA against a range of contemporary optimization methods. The data within this table unequivocally showcases PSA's prowess in consistently outperforming results derived from renowned algorithms such as RFO (Połap and Woźniak 2021), COA (Jia et al. 2023), SO (Hashim and Hussien 2022), MVO (Mirjalili et al. 2016), GSA (Mirjalili et al. 2016), CPSO (Mirjalili et al. 2016), GA (Coello Coello 2000), GA1 (Deb 1991), GA2 (Deb 1991) and HS (Lee and Geem 2005). Figure 37 visually delineates the optimal solutions attained by each algorithm. From this illustration, it's clear that PSA not only aligns with but often surpasses the performance of other cutting-edge methods. Furthermore, the insights gleaned from Table 12; Fig. 14 unequivocally indicate that PSA demonstrates exceptional capability in pinpointing optimal solutions, even within the intricate matrix of multi-dimensional constraints.

4.5 Pressure vessel design task

The primary objective of this task is to minimize the fabrication costs related to the design of a pressure vessel. As depicted in Fig. 15, the vessel embodies a distinctive geometric configuration, juxtaposing a flat section with a hemispherical profile. Central to the optimization

Table 12 The best design of welded beam design task obtained by different algorithms

Optimization technique	Optimal parameters				Optimal cost
	h	l	t	b	
GA1 (Deb 1991)	0.2489	6.173	8.1789	0.2533	2.43
GA2 (Deb 1991)	0.2918	5.2141	7.8446	0.2918	2.59
GA (Coello Coello 2000)	0.1828	4.0483	9.3666	0.2059	1.82455147
HS (Lee and Geem 2005)	0.2442	6.2231	8.2915	0.2443	2.3807
GSA (Mirjalili et al. 2016)	0.182129	3.856979	10	0.202376	1.87995
CPSO (Mirjalili et al. 2016)	0.202369	3.544214	9.04821	0.205723	1.72802
MVO (Mirjalili et al. 2016)	0.205463	3.473193	9.044502	0.205695	1.72645
RFO (Połap and Woźniak 2021)	0.21846	3.51024	8.87254	0.22491	1.86612
COA (Jia et al. 2023)	0.205557662	3.25636618	9.04034118	0.20575381	1.696251144
SO (Hashim and Hussien 2022)	0.2057	3.4714	9.0366	0.2057	1.72491
PSA (This study)	0.205745	3.252936	9.036665	0.205748	1.695390

process are parameters such as the inner radius (R), shell thickness (T_s), the length of the cylindrical segment excluding the head (L), and the head thickness (T_h). These parameters, representing pivotal structural aspects, significantly influence the vessel's optimal design. To comprehensively formulate the problem, precise mathematical delineations and constraints have been devised to both encapsulate the aim of cost reduction and adhere to the stipulated design benchmarks. The ensuing equations provide a rigorous mathematical exposition of this sophisticated endeavor.

Consider:

$$\vec{x} = [x_1 x_2 x_3 x_4] = [T_s T_h R L]$$

Minimize:

$$f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Subject to:

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0$$

$$g_2(\vec{x}) = -x_2 + 0.00954x_3 \leq 0$$

$$g_3(\vec{x}) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \leq 0$$

$$g_4(\vec{x}) = x_4 - 240 \leq 0$$

Variable range:

$$0 \leq x_1, x_2 \leq 99$$

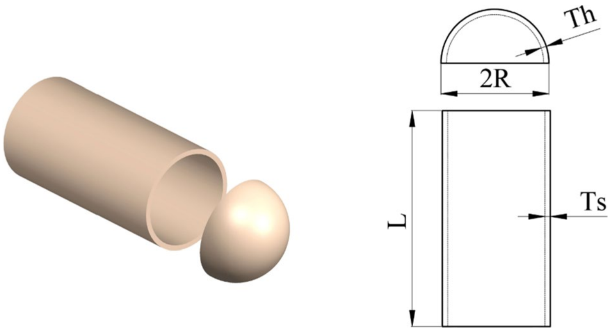


Fig. 15 Pressure vessel design task

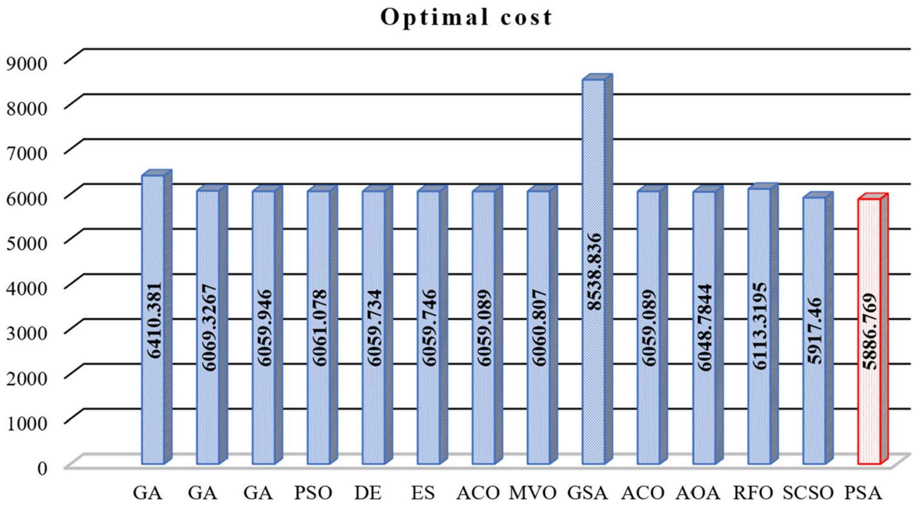


Fig. 16 Comparison of optimal outcomes in pressure vessel design task

$$10 \leq x_3, x_4 \leq 200$$

Table 13 presents a comprehensive analytical review of the challenge, meticulously evaluating the existing data. The furnished evidence robustly attests to the enduring efficacy of the proposed PSA. Figure 16 elucidates the elevated performance benchmarks achieved by distinct algorithms. This depiction underscores PSA's capability to yield outcomes surpassing other avant-garde techniques. Impressively, it provides solutions aligning with, or in some cases, surpassing the benchmarks established by recognized optimization strategies such as ACO (Kaveh and Talatahari 2010a), RFO (Połap and Woźniak 2021), SCSO (Seyyedabbasi and Kiani 2023), MVO (Mirjalili et al. 2016), GSA (Mirjalili et al. 2016), PSO (Lee and Geem 2005), multiple variants of GA (Coello Coello 2000), (Coello and Montes 2002), (Deb 1997), ES (Mezura-Montes and Coello 2008), DE (Li et al. 2007), ACO (Kaveh and Talatahari 2010a) and AOA (Abualigah et al. 2021). Beyond the primary insights, this scrutiny emphasizes the relevance of PSA in practical engineering design tasks, especially in scenarios where the characteristics of the investigatory domain are nebulous or indeterminate.

Table 13 The best design of pressure vessel design task obtained by different algorithms

Optimization technique	Optimal parameters				Optimal cost
	T_s	T_h	R	L	
GA (Deb 1997)	0.9375	0.5	48.329	112.679	6410.381
GA (Coello Coello 2000)	0.8750	0.5000	42.0939	177.0850	6069.3267
GA (Coello and Montes 2002)	0.8125	0.4375	42.0974	176.6541	6059.946
PSO (Lee and Geem 2005)	0.8125	0.4375	42.09127	176.7465	6061.078
DE (Li et al. 2007)	0.8125	0.4375	42.09841	176.6377	6059.734
ES (Mezura-Montes and Coello 2008)	0.8125	0.4375	42.09809	176.6405	6059.746
ACO (Kaveh and Talatahari 2010a)	0.8125	0.4375	42.10362	176.5727	6059.089
MVO (Mirjalili et al. 2016)	0.8125	0.4375	42.09074	176.7387	6060.807
GSA (Mirjalili et al. 2016)	1.125	0.625	55.98866	84.4542	8538.836
ACO (Kaveh and Talatahari 2010a)	0.8125	0.4375	42.10362	176.5727	6059.089
AOA (Abualigah et al. 2021)	0.830374	0.416206	42.751270	169.345400	6048.784400
RFO (Połap and Woźniak 2021)	0.81425	0.44521	42.20231	176.62145	6113.3195
SCSO (Seyyedabbasi and Kiani 2023)	0.7798	0.9390	40.3864	199.2918	5917.46
PSA (This study)	0.77844	0.38477	40.33164	199.86613	5886.769

5 Conclusion

Portia spiders are known for their ability to solve novel problems. For instance, when faced with an obstacle that hinders a direct path to their prey, these spiders can choose a detour, even if it leads to a temporary loss of sight of the target. Furthermore, they demonstrate a capacity for learning from past experiences and adjusting tactics accordingly.

This article introduces a nature-inspired model based on the hunting behaviours of the Portia spider, termed the Portia spider algorithm (PSA). Within this framework, the stalking and striking behaviours of the Portia spider inform the exploration phase, while the tactics of invading and imitating guide the exploitation phase.

Our extensive evaluations, which range from 23 classical test functions to 29 CEC2017 benchmarks and five engineering optimization problems, highlight the robustness and adaptability of PSA. Significantly, when compared to other meta-heuristic algorithms, PSA exhibits potential advantages, underscoring its potential in computational optimization. PSA is well-equipped to adapt and traverse complex optimization landscapes, marking its distinctive position in the field of evolutionary algorithms. Subsequent studies could explore further refinements to PSA, leveraging its inherent strengths and customizing it to address specific domain challenges.

Although PSA offers innovative approaches to optimization, it may encounter limitations such as slow convergence rates, particularly when applied to high-dimensional or intricate optimization challenges. To enhance its performance, integrating techniques like Lévy flights, mutation, and additional evolutionary operators could be beneficial. These methods can introduce new diversity into the population of solutions and allow the algorithm to escape local optima more effectively.

Moreover, hybridizing the PSA with other stochastic optimization algorithms could further improve its efficacy. By combining the strengths of PSA with other proven strategies, it is possible to create a more robust algorithm that leverages the advantages of each component method. This could lead to improved convergence speeds, better solution quality, and increased reliability across a wider range of optimization problems.

Such enhancements would be particularly useful in complex optimization landscapes, where the balance between exploration (diversification) and exploitation (intensification) is crucial for finding global optima. The adaptive capabilities of hybrid algorithms can be particularly adept at navigating these challenges, offering a powerful tool for complex optimization tasks.

Acknowledgements We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

Author contributions Both authors, including V.H.S.P. and N.T.N.D., jointly contributed to the writing of the main manuscript, the preparation of all figures and tables, and reviewed and approved the final version prior to submission.

Funding This research did not receive dedicated funding from public, commercial, or non-profit grant agencies.

Data availability The corresponding author is available to provide the data, model, or source codes of the proposed PSA underlying the findings of this study upon request, in accordance with reasonable conditions.

Declarations

Conflict of interest There is no conflict of interest.

References

Abd Elaziz M, Oliva D, Xiong S (2017) An improved opposition-based sine cosine algorithm for global optimization. Expert Syst Appl 90:484–500

Abdollahzadeh B, Gharehchopogh FS, Khodadadi N, Mirjalili S (2022) Mountain Gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Adv Eng Softw 174:103282

Abedinpourshotorban H, Shamsuddin SM, Beheshti Z, Jawawi DN (2016) Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm. Swarm Evol Comput 26:8–22

Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021a) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609

Abualigah L, Yousri D, Abd Elaziz M, Ewees AA, Al-Qaness MA, Gandomi AH (2021b) Aquila optimizer: a novel meta-heuristic optimization algorithm. Comput Ind Eng 157:107250

Alsattar HA, Zaidan A, Zaidan B (2020) Novel meta-heuristic bald eagle search optimisation algorithm. Artif Intell Rev 53:2237–2264

Aye CM, Wansaseub K, Kumar S, Tejani GG, Bureerat S, Yildiz AR, Pholdee N (2023) Airfoil shape optimisation using a multi-fidelity surrogate-assisted metaheuristic with a new multi-objective infill sampling technique. CMES-Computer Model Eng Sci 137(3):21111

Bansal JC, Sharma H, Jadon SS, Clerc M (2014) Spider monkey optimization algorithm for numerical optimization. Memetic Comput 6:31–47

Baş E, Ülker E (2020) A binary social spider algorithm for uncapacitated facility location problem. Expert Syst Appl 161:113618

Boussaïd I, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. Inf Sci 237:82–117
Chakraborty P, Roy GG, Das S, Jain D, Abraham A (2009) An improved harmony search algorithm with differential mutation operator. Fundamenta Informaticae 95(4):401–426
Chen Z, Francis A, Li S, Liao B, Xiao D, Ha TT, Li J, Ding L, Cao X (2022) Egret swarm optimization algorithm: an evolutionary computation approach for model free optimization. Biomimetics 7(4):144
Cheng M-Y, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112
Chickermane H, Gea HC (1996) Structural optimization using a new local approximation method. Int J Numer Methods Eng 39(5):829–846
Chopra N, Ansari MM (2022) Golden Jackal optimization: a novel nature-inspired optimizer for engineering applications. Expert Syst Appl 198:116924
Coello CAC, Montes EM (2002) Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inform 16(3):193–203
Coello Coello CA (2000) Constraint-handling using an evolutionary multiobjective optimization technique. Civil Eng Syst 17(4):319–346
Cross FR, Jackson RR (2005) Spider heuristics. Behav Process 69(2):125–127
Cuevas E, Cienfuegos M, Zaldívar D, Pérez-Cisneros M (2013) A swarm optimization algorithm inspired in the behavior of the social-spider. Expert Syst Appl 40(16):6374–6384
Dantzig GB (2002) Linear programming. Oper Res 50(1):42–47
Deb K (1991) Optimal design of a welded beam via genetic algorithms. AIAA J 29(11):2013–2015
Deb K (1997) GeneAS: A robust optimal design technique for mechanical component design. Evolut Algorithm Eng Appl 1997:497–514
Deb K, Goyal M (1996) A combined genetic adaptive search (GeneAS) for engineering design. Comput Sci Inf 26:30–45
El-Bages M, Elsayed W (2017) Social spider algorithm for solving the transmission expansion planning problem. Electr Power Syst Res 143:235–243
Elsayed W, Hegazy Y, Bendary F, El-Bages M (2016) Modified social spider algorithm for solving the economic dispatch problem. Eng Sci Technol Int J 19(4):1672–1681
Ewees AA, Abd El Aziz M, Elhoseny M (2017) Social-spider optimization algorithm for improving ANFIS to predict biochar yield. 2017 8th international conference on computing, communication and networking technologies (ICCCNT)
Gandomi AH (2014) Interior search algorithm (ISA): a novel approach for global optimization. ISA Trans 53(4):1168–1183
Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Engineering with Computers 29:17–35
Gautier A, Granot F (1994) On the equivalence of constrained and unconstrained flows. Discrete Appl Math 55(2):113–132
Hashim FA, Hussien AG (2022) Snake Optimizer: a novel meta-heuristic optimization algorithm. Knowl Based Syst 242:108320
Hashim FA, Houssein EH, Hussain K, Mabrouk MS, Al-Atabany W (2022) Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems. Math Comput Simul 192:84–110
Holland JH (1992) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press, Cambridge
Hooker JN, Osorio MA (1999) Mixed logical-linear programming. Discrete Appl Math 96:395–442
Jackson RR, Wilcox RS (1998) Spider-eating spiders: despite the small size of their brain, jumping spiders in the genus Portia outwit other spiders with hunting techniques that include trial and error. Am Sci 86(4):350–357
Jaderyan M, Khotanlou H (2016) Virulence optimization algorithm. Appl Soft Comput 43:596–618
James J, Li VO (2015) A social spider algorithm for global optimization. Appl Soft Comput 30:614–627
Jia H, Rao H, Wen C, Mirjalili S (2023) Crayfish optimization algorithm. Artif Intell Rev 56:1–61
Kannan B, Kramer SN (1994) An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. J Mech Des 116(2):405–411
Kashan AH (2014) League Championship Algorithm (LCA): an algorithm for global optimization inspired by sport championships. Appl Soft Comput 16:171–200
Kashan AH (2015) A new metaheuristic for optimization: optics inspired optimization (OIO). Comput Oper Res 55:99–125
Kaveh A, Khayatazad M (2012) A new meta-heuristic method: ray optimization. Comput Struct 112:283–294
Kaveh A, Mahdavi V (2014) Colliding bodies optimization method for optimum discrete design of truss structures. Comput Struct 139:43–53

Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. Acta Mech 213(3–4):267–289
Kaveh A, Talatahari S (2010a) An improved ant colony optimization for constrained engineering design problems. Eng Comput 27(1):155–182
Kennedy J, Eberhart R (1995) Particle swarm optimization. Proceedings of ICNN'95-international conference on neural networks
Kumar S, Tejani GG, Pholdee N, Bureerat S, Jangir P (2022) Multi-objective teaching-learning-based optimization for structure optimization. Smart Sci 10(1):56–67
Kumar S, Tejani GG, Pholdee N, Bureerat S (2022a) Performance enhancement of meta-heuristics through random mutation and simulated annealing-based selection for concurrent topology and sizing optimization of truss structures. Soft Comput 26(12):5661–5683
Kumar S, Panagant N, Tejani GG, Pholdee N, Bureerat S, Mashru N, Patel P (2023) A two-archive multi-objective multi-verse optimizer for truss design. Knowl Based Syst 270:110529
Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 194(36–38):3902–3933
Li L-J, Huang Z, Liu F, Wu Q (2007) A heuristic particle swarm optimizer for optimization of pin connected structures. Comput Struct 85(7–8):340–349
Mezura-Montes E, Coello CAC (2008) An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. Int J Gen Syst 37(4):443–473
Mirjalili S (2015a) The ant lion optimizer. Adv Eng Softw 83:80–98
Mirjalili S (2015b) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl Based Syst 89:228–249
Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27(2):495–513
Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191
Mirjalili SZ, Mirjalili S, Saremi S, Faris H, Aljarah I (2018) Grasshopper optimization algorithm for multi-objective optimization problems. Appl Intell 48:805–820
Nguyen TT, Vo DN (2020) Improved social spider optimization algorithm for optimal reactive power dispatch problem with different objectives. Neural Comput Appl 32(10):5919–5950
Nonut A, Kanokmedhakul Y, Bureerat S, Kumar S, Tejani GG, Artrit P, Yıldız AR, Pholdee N (2022) A small fixed-wing UAV system identification using metaheuristics. Cogent Eng 9(1):2114196
Ouadfel S, Taleb-Ahmed A (2016) Social spiders optimization and flower pollination algorithm for multilevel image thresholding: a performance study. Expert Syst Appl 55:566–584
Parejo JA, Ruiz-Cortés A, Lozano S, Fernandez P (2012) Metaheuristic optimization frameworks: a survey and benchmarking. Soft Comput 16(3):527–561
Pham VHS, Nguyen Dang NT, Nguyen VN (2023) Hybrid sine Cosine Algorithm with Integrated Roulette Wheel Selection and Opposition-based learning for Engineering optimization problems. Int J Comput Intell Syst 16(1):171. https://doi.org/10.1007/s44196-023-00350-2
Pham VHS, Trang NTN, Dat CQ (2023b) Optimization of production schedules of multi-plants for dispatching ready-mix concrete trucks by integrating grey wolf optimizer and dragonfly algorithm. *Engineering, Construction and Architectural Management*
Połap D, Woźniak M (2021) Red fox optimization algorithm. Expert Syst Appl 166:114107
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
Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248
Reynolds RG (1994) An introduction to cultural algorithms. Proceedings of the 3rd annual conference on evolutionary programming, World Scientific Publishing, Chennai
Rezaei F, Safavi HR, Abd Elaziz M, Mirjalili S (2023) GMO: geometric mean optimizer for solving engineering problems. Soft Comput 27(15):10571–10606
Robson D (2020) Spider smarts. New Sci 245(3268):42–45
Sadeeq HT, Abdulazeez AM (2022) Giant trevally optimizer (GTO): a novel metaheuristic algorithm for global optimization and challenging engineering problems. Ieee Access 10:121615–121640
Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 13(5):2592–2612
Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47

Seyyedabbasi A, Kiani F (2023) Sand cat swarm optimization: a nature-inspired algorithm to solve global optimization problems. Engineering with Computers 39(4):2627–2651
Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12(6):702–713
Singh P, Kottath R, Tejani GG (2022) Ameliorated follow the leader: algorithm and application to truss design problem. Structures 42:181–204
Son PVH, Nguyen Dang NT (2023aa) Optimizing time and cost simultaneously in projects with multi-verse optimizer. Asian J Civil Eng 24:1–7
Son PVH, Nguyen Dang NT (2023bb) Solving large-scale discrete time–cost trade-off problem using hybrid multi-verse optimizer model. Sci Rep 13(1):1987. https://doi.org/10.1038/s41598-023-29050-9
Spall JC (2005) Introduction to stochastic search and optimization: estimation, simulation, and control. Wiley, Hoboken
Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341
Van den Bergh F, Engelbrecht AP (2006) A study of particle swarm optimization particle trajectories. Inf Sci 176(8):937–971
Wang L, Cao Q, Zhang Z, Mirjalili S, Zhao W (2022) Artificial rabbits optimization: a new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. Eng Appl Artif Intell 114:105082
Williams HP (2009) Integer programming. Logic and Integer Programming. Springer, Heidelberg, pp 25–70
Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82
Wu G, Mallipeddi R, Suganthan PN (2017) Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization. *National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report*
Xie L, Han T, Zhou H, Zhang Z-R, Han B, Tang A (2021) Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization. Comput Intell Neurosci 2021:1–22
Xue J, Shen B (2020) A novel swarm intelligence optimization approach: sparrow search algorithm. Syst Sci Control Eng 8(1):22–34
Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3(2):82–102
Zhao W, Zhang Z, Wang L (2020) Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications. Eng Appl Artif Intell 87:103300
Zhao S, Zhang T, Ma S, Wang M (2023) Sea-horse optimizer: a novel nature-inspired meta-heuristic for global optimization problems. Appl Intell 53(10):11833–11860
Zhou A, Qu B-Y, Li H, Zhao S-Z, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: a survey of the state of the art. Swarm Evol Comput 1(1):32–49

Authors and Affiliations

Vu Hong Son Pham[1] · Nghiep Trinh Nguyen Dang[1]

✉ Nghiep Trinh Nguyen Dang
ndntrinh.sdh232@hcmut.edu.vn

Vu Hong Son Pham
pvhson@hcmut.edu.vn

[1] Faculty of Civil Engineering, Ho Chi Minh City University of Technology (HCMUT), Vietnam National University (VNU-HCM), Ho Chi Minh City, Vietnam