Information Sciences 734 (2026) 123019

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

An area-based computational algorithm for robust extrema detection in noisy environments

Madjid Tavana [a,b,*], Hosein Arman [c], Andreas Dellnitz [d]

[a] Business Systems and Analytics Department, Distinguished Chair of Business Analytics, La Salle University, Philadelphia, USA
[b] Business Information Systems Department, Faculty of Business Administration and Economics, University of Paderborn, Paderborn, Germany
[c] Department of Management, Mobarakeh Branch, Islamic Azad University, Mobarakeh, Isfahan, Iran
[d] Leibniz-FH University of Applied Sciences, Hannover, Germany

ARTICLE INFO

Keywords:
Area-based method
Function optimization
Extrema detection
Univariate analysis
Point-based heuristics
Search algorithms

ABSTRACT

A variety of point-based heuristics and metaheuristics have been developed to approximate the global optimum of univariate functions. However, these Point-Based Search (PBS) algorithms often converge to local optima and fail to detect all extrema due to limited domain exploration. This study proposes an Area-Based Search (ABS) algorithm that systematically partitions the domain into uniformly spaced subintervals and evaluates the area under the curve in each segment. Subintervals with significantly larger or smaller areas than their neighbors are likely to contain local maxima or minima, respectively. We validate this idea on multimodal test functions using a Monte Carlo simulation framework with 1,000 trials. Across all noise levels in a standard benchmark function, ABS consistently detects all 16 local and global extrema. Intuitively, coverage measures the fraction of true extrema that an algorithm successfully recovers within a prescribed positional tolerance. Compared to Genetic Algorithms (GA), ABS achieves up to 37% higher detection accuracy under noise, with an average coverage improvement of 4.75% across all test cases. Additionally, ABS exhibited a 30.41% lower position error and a 36.89% lower value error than GA. The deterministic nature of ABS, with only one tunable resolution parameter, supports its use in noisy environments requiring full-spectrum extrema detection.

1. Introduction

A univariate function $f(x)$ maps each input x to a unique output y. When $f(x)$ takes a value that is larger (or smaller) than those of its immediate neighbors, the point is a local maximum (or minimum). The highest (or lowest) value over the entire domain is called the global maximum (or minimum). In calculus, exact local extrema correspond to points where the derivative $f'(x)$ is zero or undefined, and a sign change in $f'(x)$ typically indicates the presence of an extremum.

However, analytically finding the roots of $f'(x)$ can be complex and computationally demanding, especially for nonlinear or non-differentiable functions. Many studies employ heuristic or approximate algorithms to detect optima and overcome this. A classic example is binary search, which iteratively evaluates the midpoint of a given interval to determine whether it is a local extremum. If not, the method uses comparisons with neighboring points to guide the search direction based on whether the values increase or

* Corresponding author at: Business Systems and Analytics Department, Distinguished Chair of Business Analytics, La Salle University, Philadelphia, United States.

E-mail addresses: tavana@lasalle.edu (M. Tavana), hosein.arman@iau.ac.ir (H. Arman), andreas.dellnitz@leibniz-fh.de (A. Dellnitz).

https://doi.org/10.1016/j.ins.2025.123019
Received 14 October 2025; Received in revised form 16 December 2025; Accepted 19 December 2025
Available online 22 December 2025

decrease. While effective for simple or unimodal functions, such approaches struggle with more complex functions and wide intervals. Consequently, various metaheuristic algorithms have been proposed to enhance search performance, particularly in cases involving multiple extrema. Some studies have specifically addressed multivariate optimization. For instance, Peccini et al. [1] proposed enumeration procedures for the global design of distillation columns, while Moriwaki et al. [2] investigated hybrid membrane distillation structures. Zhou et al. [3] developed a multi-objective differential evolution algorithm using neighborhood-based strategies for Pareto optimization. Oh and Lee [4] introduced a prediction–correction mechanism for the global optimization of multi-objective functions. Liu et al. [5] proposed a Chebyshev metamodel based on global optimality conditions for point set registration.

1.1. Metaheuristics and the challenge of extrema detection

Metaheuristics are widely used to solve complex optimization problems and are commonly grouped into three broad families: evolutionary, physics-based, and swarm-intelligence algorithms. Evolutionary algorithms are inspired by biological evolution, with the genetic algorithm [6]being one of the most well-known. Other algorithms in this class include differential evolution [7], evolutionary programming [8], evolution strategies [9], and the biogeography-based optimizer [10]. Physics-based metaheuristics draw inspiration from physical laws, such as gravitational force [11,12], or astronomical events like the Big Bang and Big Crunch [13], galaxy-based search [14], and black hole optimization [15]. Additional examples include small-world optimization [16], central force optimization [17], charged system search [18], artificial chemical reactions [19], and ray-based methods [20]. Swarm intelligence algorithms, the third major category, simulate collective behaviors observed in animals and insects. Some mimic the behavior of insects such as honey bees [21], ants [22], wasps [23], fireflies [24], and fruit flies [25]. Others emulate animals, such as monkeys [26], cuckoos [27], dolphins [28], birds [29], krill [30], gray wolves [31], and whales [32]. Nadimi-Shahraki et al. [33], for example, reviewed various applications of the Grey Wolf Optimizer in the context of the Internet of Things. Given the diverse strengths and limitations of these techniques, various hybridization strategies have been developed [34,35].

Despite the diversity mentioned above, most metaheuristics share a key limitation: they are Point-Based Search (PBS) methods that iteratively update candidate solutions from a set of discrete initial points. While effective in locating global optima, these approaches often miss local extrema, limiting their usefulness in tasks requiring full-spectrum detection, such as robustness analysis and decision modeling. To address this, we propose a novel Area-Based Search (ABS) algorithm. Unlike PBS methods, ABS partitions the input domain into equal subintervals and calculates the area under the curve for each. Subintervals with larger or smaller areas than their neighbors are flagged as candidates for maxima or minima. This integral-based approach reduces the influence of noise and captures structure missed by point evaluations. Notably, while area calculation techniques have already proven effective in multi-criteria decision-making [36], their potential for identifying local and global extrema remains largely unexplored. ABS bridges this gap by leveraging integral-based evaluations to provide a smoother signal that naturally filters out noise, enabling robust detection of both local and global extrema. The ABS method is benchmarked against classical PBS and Genetic Algorithms (GA) using extensive numerical simulations across multiple test functions, noise levels, and resolution settings. Performance metrics include detection coverage, position accuracy, and value error.

Most existing extrema-detection methods—including optimization-based strategies and common derivative- or filtering-based pipelines—operate on discrete samples and typically emphasize a single global optimum or a small subset of dominant peaks. In contrast, ABS evaluates integral information over subintervals, providing a complementary framework that is explicitly designed to detect all local and global extrema in a structured and noise-robust manner.

1.2. Research objectives

The overarching objective of this research is to overcome the limitations inherent in conventional point-based and metaheuristic optimization techniques, particularly their inability to detect all local and global extrema of univariate functions reliably. Toward this goal, the study aims to:

- Formulate and implement a novel deterministic algorithm—ABS—designed to systematically identify all extrema within a predefined univariate domain through integral-based analysis.
- Conduct a comprehensive performance evaluation of ABS compared to classical Point-Based Search (PBS) techniques and Genetic Algorithms (GA) under a range of experimental conditions, including varying Gaussian noise and resolution levels.
- Assess the proposed method in terms of detection accuracy, noise resilience, computational efficiency, and reproducibility, thereby establishing ABS as a practical and robust alternative for full-spectrum extrema detection in noisy environments.

Overall, ABS differs from prior point-based and metaheuristic approaches by relying on subinterval integrals rather than discrete samples and by explicitly targeting the complete set of local and global extrema, not solely the global optimum.

The remainder of this paper is structured as follows. Section 2 elaborates on the conceptual foundation and methodology underlying the ABS algorithm. Section 3 presents the detailed ABS algorithms for detecting local and global optima and enhances these algorithms by incorporating point-based refinement methods, resulting in hybrid ABS-PBS algorithms. It also illustrates the practical implementation of these algorithms with worked examples. Section 4 provides a comprehensive MATLAB-based numerical analysis, benchmarking ABS against a simple PBS and Genetic Algorithms (GA) using Monte Carlo simulations across noise levels and resolution settings. Section 5 discusses the theoretical and practical implications of the findings. Section 6 explores potential directions for future research, and Section 7 concludes the paper, summarizing the key insights and limitations of the current work.

2. The ABS approach for finding the global extrema

This section introduces the ABS approach for finding the global maximum (or minimum) of an integrable univariate function $f(x)$ on an interval $I = [a_0, a_n]$. We first split the interval into n equal-width subintervals $[a_{k-1}, a_k]$, that is, $I_1 = [a_0, a_1], I_2 = [a_1, a_2], \cdots, I_n = [a_{n-1}, a_n]$, so that $I = I_1 \cup I_2 \cup \cdots \cup I_n$. For each subinterval, we compute the area under $f(x)$. The subinterval with the largest (or smallest) area is the most likely to contain the global maximum (or minimum).

The underlying intuition is simple. As we move toward a maximum, $f(x)$ typically increases. It then decreases as we move away. Hence, the sum of function values around the global maximum tends to be larger than the corresponding sums in other regions. In a continuous setting, this sum is represented by the area under $f(x)$. Therefore, the subinterval containing the global maximum should have a larger area than any other subinterval.

Unlike many existing approaches, ABS does not choose a single representative point in each subinterval and evaluate $f(x)$ only there. Instead, it considers all points in each subinterval and aggregates their function values. In the continuous case, this aggregation corresponds to the area under $f(x)$, which we compute by a definite integral. We calculate this area separately for every subinterval. The largest area indicates the subinterval that is most likely to contain the global maximum. The smallest area indicates the subinterval that is most likely to contain the global minimum. Fig. 1 shows the ABS approach for searching for global extrema.

In Fig. 1, the function $y = f(x)$ is evaluated over the interval $[a_0, a_5]$, which is partitioned into five equal-width subintervals $[a_{k-1}, a_k]$. For each subinterval, S_k denotes the area under $f(x)$. In panel (a), the subinterval with the largest area S_3 is likely to contain the global maximum $y_{\max}$; in panel (b), the subinterval with the smallest area S_3 is likely to contain the global minimum $y_{\min}$.

From a mathematical point of view, the link between area and extrema can be explained as follows. Assume that f is continuous on $[a_0, a_n]$ and has a global maximum at x^*. In a small neighbourhood around x^*, the values of $f(x)$ are larger than at points farther away. If we partition $[a_0, a_n]$ into equal subintervals of width d and choose d sufficiently small, then the subinterval that contains x^* will collect higher function values and therefore have a larger integral than any subinterval that lies entirely in regions where f is lower. A subinterval with a locally maximal area S_k is thus a natural candidate for containing the global maximum. The same reasoning applies to minima by considering $-f(x)$ instead of $f(x)$.

Proposition 1. (*(Local area dominance near a maximum).*) *Let f be continuous on $[a_0, a_n]$ and consider three consecutive subintervals $I_{k-1} = [a_{k-2}, a_{k-1}], I_k = [a_{k-1}, a_k]$, and $I_{k+1} = [a_k, a_{k+1}]$ of equal width d. Suppose that there exists a point $x^* \in I_k$ such that*

- f is nondecreasing on $[a_{k-1}, x^*]$ and nonincreasing on $[x^*, a_k]$, and
- $f(x) \le f(x^*)$ for all $x \in I_{k-1} \cup I_k \cup I_{k+1}$.

Then x^* is a local maximum of f in $I_{k-1} \cup I_k \cup I_{k+1}$, and the corresponding subinterval I_k satisfies

$$S_k \ge S_{k-1}, S_k \ge S_{k+1},$$

where $S_l = \int_{I_l} f(x)dx$ is the area under f over the subinterval I_l.

Proof (sketch).

By the second assumption, $f(x^*)$ is the largest value of f on $I_{k-1} \cup I_k \cup I_{k+1}$; hence x^* is a local maximum on this union. The first assumption implies that the values of f in I_k are at least as large, in aggregate, as the values in the neighbouring subintervals. Because the subintervals have the same width d, integrating over each of them preserves this dominance: the integral over I_k, S_k, cannot be smaller than the integral over I_{k-1} or over I_{k+1}. Thus $S_k \ge S_{k-1}$ and $S_k \ge S_{k+1}$. The same reasoning applies to local minima by considering $-f$ instead of f. □

Example 1. *This example illustrates the ABS approach. Consider two functions, $f_1(x) = 5x - x^2$ and $f_2(x) = x^2 - 5x + 7$. On the interval $I =$*

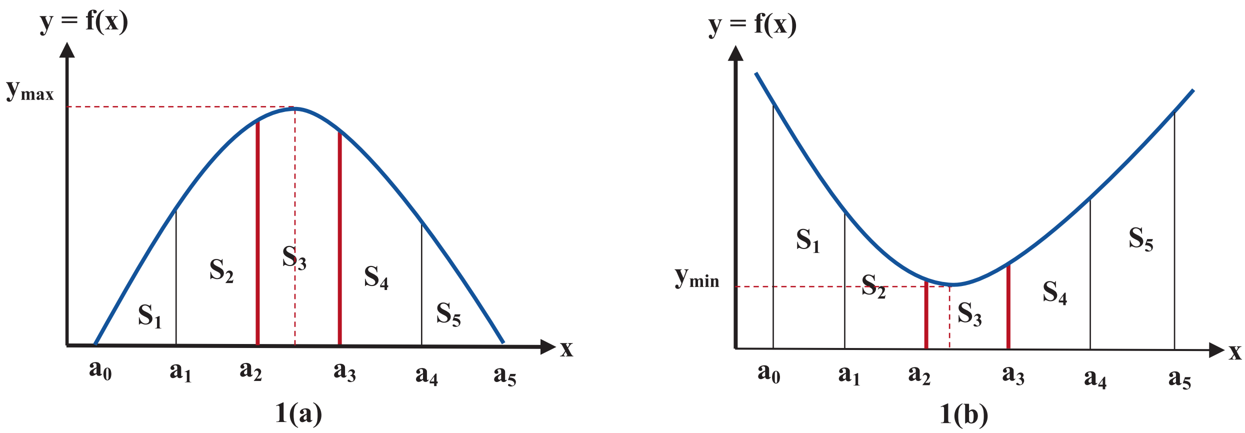


Fig. 1. Illustration of the ABS approach for finding global extrema.

$[0,5]$, *the point* $x = 2.5$ *is the global maximum of* $f_1(x)$ *and the global minimum of* $f_2(x)$. Fig. 2(a) and (b) *show their behavior. When x increases from 0 to 2.5, the value of* $f_1(x)$ *increases and the value of* $f_2(x)$ *decreases. When x increases from 2.5 to 5,* $f_1(x)$ *decreases and* $f_2(x)$ *increases. Therefore, the sum of* $f_1(x)$ *values near* $x = 2.5$ *should be larger than the sums near any other point. The sum of* $f_2(x)$ *values near* $x = 2.5$ *should be smaller than the sums near any other point.*

We now split the interval $I = [0,5]$ into five equal-width subintervals: $I_1 = [0,1]$, $I_2 = [1,2]$, $I_3 = [2,3]$, $I_4 = [3,4]$, and $I_5 = [4,5]$. The sum of $f_1(x)$ (or $f_2(x)$) values over a continuous subinterval equals the area under $f_1(x)$ (or $f_2(x)$) on that subinterval. We expect the area under $f_1(x)$ (or $f_2(x)$) for $I_3 = [2,3]$ to be larger (or smaller) than the areas for the other subintervals because I_3 contains the global maximum (or minimum) at $x = 2.5$. This is indeed the case. The areas under $f_1(x)$ over I_1, I_2, I_3, I_4, and I_5 are 2.17, 5.17, 6.17, 5.17, and 2.17, respectively. The areas under $f_2(x)$ over the same subintervals are 4.83, 1.83, 0.83, 1.83, and 4.83, respectively.

Fig. 3(a) and (b) highlight the largest and smallest areas for I_3, which contains the global maximum of $f_1(x)$ and the global minimum of $f_2(x)$.

Note 1. The area under a univariate function $f(x)$ over an interval $[a,b]$ is

$$S = \int_a^b f(x)dx.$$

This integral represents the total area under the curve $\boldsymbol{f}(\boldsymbol{x})$ on $[\boldsymbol{a},\boldsymbol{b}]$ and can be viewed as the sum of $\boldsymbol{f}(\boldsymbol{x})$ over all points in the continuous domain.

Note 2. The ABS approach identifies subintervals that are most likely to contain a local maximum or minimum by comparing their areas with those of neighboring subintervals. Once such a promising subinterval is found, a point-based search (PBS) method, such as a refined binary search, can be applied within that interval to locate the local extremum more precisely.

Before going into details of our new approach, we summarize the notation in Table 1.

3. Methodology: The ABS algorithms for finding the local and global extrema

3.1. The ABS algorithm for searching local maxima

Building on the idea in Section 2, Section 3.1 describes ABS procedures for detecting local maxima and minima. Section 3.2 combines ABS with a PBS-based binary search. In Section 3.1, we develop an ABS algorithm to search for local maxima of a univariate function $f(x)$. For each subinterval $I_k = [a_{k-1}, a_k]$, let S_k denote again the area under $f(x)$ on I_k, and let x_k^{max} be the candidate maximum in this subinterval. Consider three consecutive subintervals I_{k-1}, I_k, and I_{k+1} with areas S_{k-1}, S_k, and S_{k+1}, respectively. If $S_k \geq S_{k-1}$ and $S_k \geq S_{k+1}$, then I_k is likely to contain a local maximum. The following algorithm formalizes this idea.

Algorithm 1. An ABS algorithm to search for local maxima.

BEGIN

Step 1. Consider $y = f(x)$ a univariate function, and $I = [a_0, a_n]$, a given interval. Split this interval into n equal-width subintervals as $I_1 = [a_0, a_1]$, $I_2 = [a_1, a_2]$, …, and $I_n = [a_{n-1}, a_n]$, where $a_k = a_{k-1} + d$ and $d = \frac{a_n - a_0}{n}$.

Step 2. Define two virtual subintervals as I_0 and I_{n+1}, where $S_0 = S_{n+1} = -\infty$.

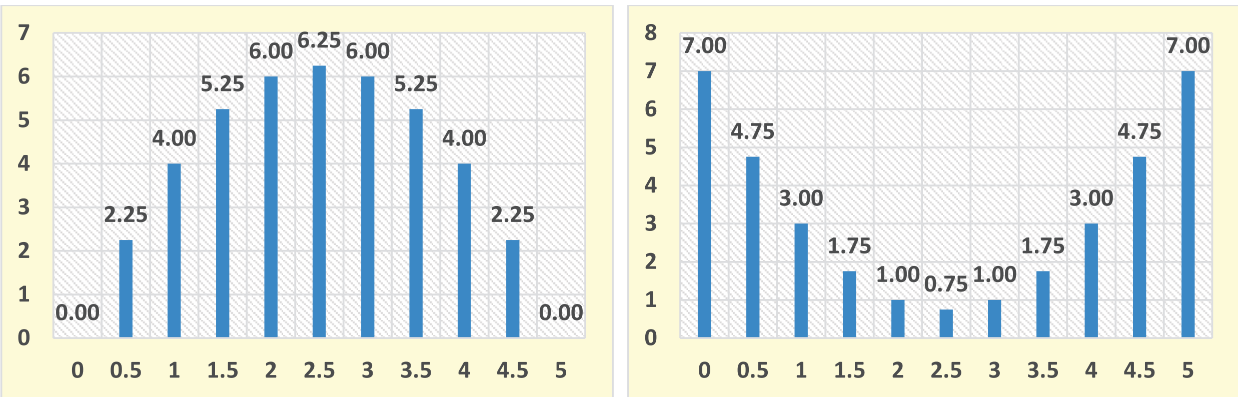
Step 3. Calculate the following integrals:

$S_k = \int_{a_{k-1}}^{a_k} f(x)dx \qquad k = 1, 2, \cdots, n$

where, S_k is the area under $f(x)$ for $I_k = [a_{k-1}, a_k]$. This step calculates the area under $f(x)$ for each subinterval separately.

Step 4. Let $k = 1$.

(*continued on next page*)



2(a). Maximum value **2(b). Minimum value**

Fig. 2. Behavior of function values $f(x)$ near a maximum and a minimum.

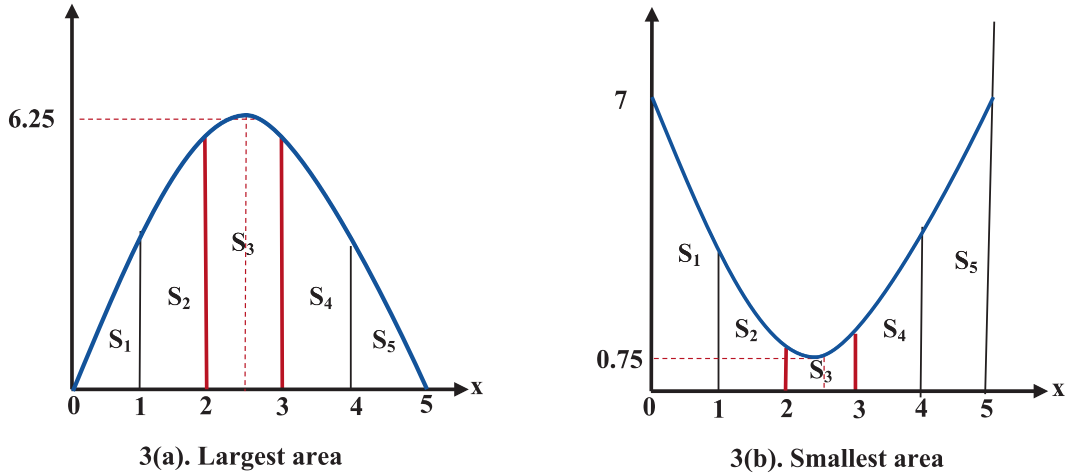


Fig. 3. Subinterval with the largest and smallest area containing the global extrema.

Table 1
Summary of notation.

Symbol	Description
$f(x)$	Univariate function under analysis
$[a_0, a_n]$	Search interval for extrema
n	Number of subintervals (ABS resolution)
$[a_{k-1}, a_k]$	The k-th subinterval of $[a_0, a_n]$
d	Width of each subinterval,$d = (a_n - a_0)/n$
S_k	Area under $f(x)$ over $[a_{k-1}, a_k]$
x_j^{max}	Location of the j-th local maximum
x_j^{min}	Location of the j-th local minimum

(*continued*)

Step 5. Consider the subinterval $I_k = [a_{k-1}, a_k]$.
  If $S_k \geq S_{k-1}$ and $S_k \geq S_{k+1}$, **then** calculate the value of x_k^{max} approximately as $x_k^{max} \approx \dfrac{a_{k-1} + a_k}{2}$
  Else, go to Step 6.
  End If
Step 6. Let $k = k + 1$
If $k \leq n$, **then** go to Step 5
Else, print all maximum points found by the algorithm, denoted by x_k^{max}. The largest value for $f(x_k^{max})$, $k = 1, \cdots, n$, is the leading candidate for the global maximum, while others are the local maxima.
End If
END

Algorithm 1 splits the interval into n subintervals $I_1, I_2, \cdots, I_n$ and compares the area under $f(x)$ in each subinterval with the areas in its immediate neighbours to search for local maxima. However, the first subinterval I_1 has no predecessor, and the last subinterval I_n has no successor, so a direct comparison is not defined for them. To address this, we introduce two virtual subintervals I_0 (before I_1) and I_{n+1} (after I_n) and assign their areas as $S_0 = S_{n+1} = -\infty$.

Since the detection of local minima is fully analogous to the detection of local maxima, we do not present a separate algorithm for minima. In practice, local minima can be found by applying the same procedure to $-f(x)$ (or, equivalently, by reversing the inequality conditions in Algorithm 1). To avoid redundancy, we therefore describe only the algorithm for local maxima; the corresponding minima-detection variant follows directly.

3.2. A combination of ABS and PBS algorithms

In Algorithm 1, we assume that the area under $f(x)$ for $I_k = [a_{k-1}, a_k]$ is larger than for its neighbouring subintervals. In this case, I_k is likely to contain a local maximum. In Step 5, the approximate location of this point is taken as $s = (a_{k-1} + a_k)/2$. If the subinterval width d is small, this midpoint is close to the true optimal value; otherwise, the difference can be significant. To obtain a more accurate estimate of x_k^{max} (and analogously x_k^{min} for minima), Step 5 can be replaced by a point-based search within I_k. In Algorithm 2, we modify Algorithm 1 accordingly and use a binary search within the selected subinterval to refine the estimate of x_k^{max}, as illustrated in Fig. 4.

Fig. 4(a) and (b) show a hybrid algorithm that integrates binary search as a PBS component into the ABS framework. The procedure is identical to Algorithm 1 (and its minima-based analogue), except that it uses binary search within each selected subinterval to locate the local maximum (or minimum) more precisely. Other PBS methods could be used in place of binary search. Because this algorithm searches for all local maxima (or minima), the largest value of $f(x)$ among the detected maxima (or the smallest among the minima) is the leading candidate for the global maximum (or minimum).

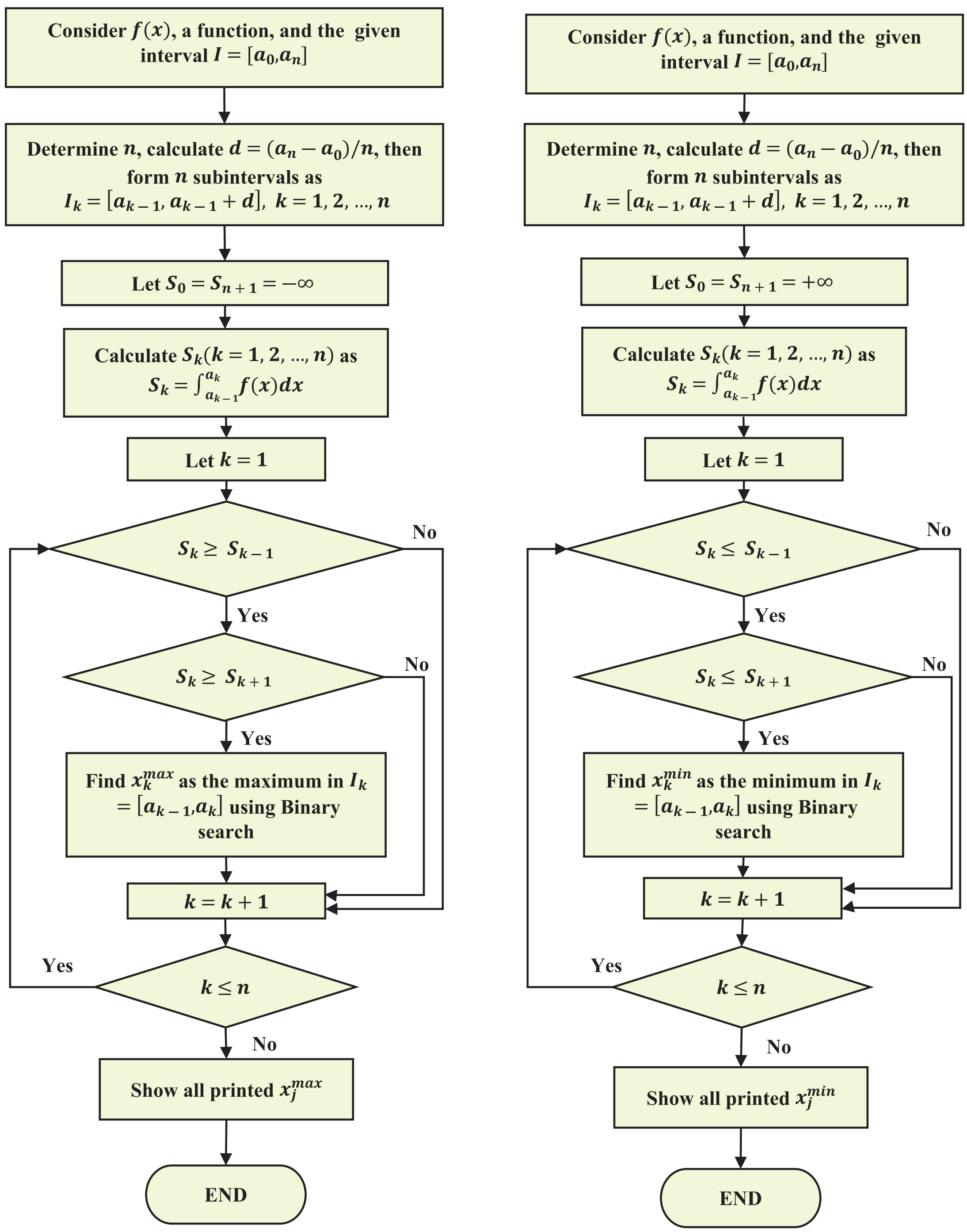


Fig. 4(a). Algorithm 2 **Fig. 4(b). Algorithm 3**

Fig. 4. Finding the local extrema by combining the ABS and PBS algorithms.

3.3. Preliminary numerical evaluation

This study utilizes synthetic benchmark functions commonly used in optimization literature. The main test function includes a multimodal function defined on the interval [-500, 500], rewritten to highlight 16 local and global extrema. The function is discretized into subintervals of widths 10 and 100, resulting in 100 and 10 subintervals, respectively. Example 2 represents one of the univariate functions Mirjalili et al. [31] used to find its global minimum in a given interval. This example illustrates how our approach finds all sixteen local and global extrema in that interval.

Example 2. *Consider the univariate function* $f(x) = -x \cdot \sin\sqrt{|x|}$ *and the given interval* $[-500, 500]$. *In contrast to existing algorithms that search for only the global optimum, the ABS algorithms (Algorithms 1–4) search for all local and global extrema. To search for all extrema of* $f(x)$*, we first rewrite this function as:*

$$f(x) = \begin{cases} -x \cdot \sin\sqrt{-x} & x \in [-500, 0] \\ -x \cdot \sin\sqrt{x} & x \in [0, 500] \end{cases}$$

Then, we consider the width of subintervals equal to $d = 10$. Therefore, the number of subintervals equals $n = 100$, shown as $I_k = [a_{k-1}, a_k] = [a_{k-1}, a_{k-1} + d], k = 1, \cdots, 100$. The area under $f(x)$ for subinterval k, denoted by S_k, is calculated as:

$$S_k = \begin{cases} \int_{a_{k-1}}^{a_k} -x \cdot \sin\sqrt{-x}\, dx & x \in [-500, 0] \\ \int_{a_{k-1}}^{a_k} -x \cdot \sin\sqrt{x}\, dx & x \in [0, 500] \end{cases}$$

Many websites, such as https://www.symbolab.com/solver, calculate definite integrals easily. Table 2 shows one hundred subintervals and their corresponding areas under $f(x)$.

Table 2 shows that the areas under $f(x)$ for subintervals 8, 30, 44, 50, 53, 63, 81, and 100 are larger than their neighbors; each probably contains a local maximum. This table also shows that the areas under $f(x)$ for subintervals 1, 20, 38, 48, 51, 57, 71, and 93 are smaller than their neighbors; each probably contains a local minimum. Note that we defined two virtual subintervals $k = 0$ and $k = 101$ whose characteristics for Algorithm 2 are $S_0 = S_{101} = f(x_0^{max}) = f(x_{101}^{max}) = -\infty$, and their characteristics for Algorithm 3 are $S_0 = S_{101} = f(x_0^{min}) = f(x_{101}^{min}) = +\infty$. The local maxima (minima) are obtained more exactly by searching these subintervals using PBS algorithms. Table 3 gives sixteen local extrema of $f(x)$ extracted from Algorithms 2 and 3.

Table 3 shows that the maximum (minimum) value, among all local maxima (minima), is 418.98 (−418.98) obtained for $x = -420.97$ ($x = 420.97$); therefore, this point is the only candidate for the global maximum (minimum).

Example 3. *This example is the same as Example 2, i.e., the problem is to find all local maxima (minima) for* $f(x) = -x \cdot \sin\sqrt{|x|}$ *in the interval* $[-500, 500]$ *using the ABS algorithms. The only difference is that this example considers the width of subintervals equal to* $d = 100$ *instead of 10. Therefore, the number of subintervals equals* $n = 10$*, shown as* $I_k = [a_{k-1}, a_k] = [a_{k-1}, a_{k-1} + 100], k = 1, \cdots, 10$. *Table 3 shows*

Table 2
Subintervals of width 10 units and their corresponding areas.

k	Subinterval	S_k	k	Subinterval	S_k	k	Subinterval	S_k	k	Subinterval	S_k
1	[-500, −490]	−1259	26	[-250, −240]	133	51	[0, 10]	−24	76	[250, 260]	657
2	[-490, −480]	−154	27	[-240, −230]	862	52	[10, 20]	99	77	[260, 270]	1428
3	[-480, −470]	925	28	[-230, −220]	1455	53	[20, 30]	228	78	[270, 280]	2104
4	[-470, −460]	1921	29	[-220, −210]	1850	54	[30, 40]	117	79	[280, 290]	2620
5	[-460, −450]	2784	30	[-210, −200]	2006	55	[40, 50]	−186	80	[290, 300]	2924
6	[-450, −440]	3466	31	[-200, −190]	1911	56	[50, 60]	−491	81	[300, 310]	2987
7	[-440, −430]	3932	32	[-190, −180]	1583	57	[60, 70]	−625	82	[310, 320]	2799
8	[-430, −420]	4159	33	[-180, −170]	1072	58	[70, 80]	−509	83	[320, 330]	2371
9	[-420, −410]	4135	34	[-170, −160]	454	59	[80, 90]	−168	84	[330, 340]	1734
10	[-410, −400]	3865	35	[-160, −150]	−176	60	[90, 100]	300	85	[340, 350]	934
11	[-400, −390]	3368	36	[-150, −140]	−719	61	[100, 110]	764	86	[350, 360]	27
12	[-390, −380]	2679	37	[-140, −130]	−1086	62	[110, 120]	1099	87	[360, 370]	−921
13	[-380, −370]	1844	38	[-130, −120]	−1218	63	[120, 130]	1218	88	[370, 380]	−1844
14	[-370, −360]	921	39	[-120, −110]	−1099	64	[130, 140]	1086	89	[380, 390]	−2679
15	[-360, −350]	−27	40	[-110, −100]	−764	65	[140, 150]	719	90	[390, 400]	−3368
16	[-350, −340]	−934	41	[-100, −90]	−300	66	[150, 160]	176	91	[400, 410]	−3865
17	[-340, −330]	−1734	42	[-90, −80]	168	67	[160, 170]	−454	92	[410, 420]	−4135
18	[-330, −320]	−2371	43	[-80, −70]	509	68	[170, 180]	−1072	93	[420, 430]	−4159
19	[-320, −310]	−2799	44	[-70, −60]	625	69	[180, 190]	−1583	94	[430, 440]	−3932
20	[-310, −300]	−2987	45	[-60, −50]	491	70	[190, 200]	−1911	95	[440, 450]	−3466
21	[-300, −290]	−2924	46	[-50, −40]	186	71	[200, 210]	−2006	96	[450, 460]	−2784
22	[-290, −280]	−2620	47	[-40, −30]	−117	72	[210, 220]	−1850	97	[460, 470]	−1921
23	[-280, −270]	−2104	48	[-30, −20]	−228	73	[220, 230]	−1455	98	[470, 480]	−925
24	[-270, −260]	−1428	49	[-20, −10]	−99	74	[230, 240]	−862	99	[480, 490]	154
25	[-260, −250]	−657	50	[-10, 0]	24	75	[240, 250]	−133	100	[490, 500]	1259

Table 3
Local extrema of the multimodal function $f(x)$ obtained by Algorithms 2 and 3.

The local maxima					The local minima				
k	Subinterval	S_k	x_k^{max}	$f(x_k^{max})$	k	Subinterval	S_k	x_k^{min}	$f(x_k^{min})$
8	[-430, −420]	4159	−420.97	418.98	1	[-500, −490]	−1259	−500	−180.59
30	[-210, −200]	2006	−203.81	201.84	20	[-310, −300]	−2987	−302.52	−300.54
44	[-70, −60]	625	−65.55	63.64	38	[-130, −120]	−1218	−124.83	−122.88
50	[-10, 0]	24	−5.24	3.94	48	[-30, −20]	−228	−25.88	−24.08
53	[20, 30]	228	25.87	24.08	51	[0, 10]	−24	5.24	−3.95
63	[120, 130]	1218	124.83	122.88	57	[60, 70]	−625	65.55	−63.64
81	[300, 310]	2987	302.52	300.54	71	[200, 210]	−2006	203.81	−201.84
100	[490, 500]	1259	500	180.59	93	[420, 430]	−4159	420.97	−418.98

ten subintervals and their corresponding areas under $f(x)$.

Table 4 shows that the areas under $f(x)$ for subintervals 1, 5, and 8 are larger than their neighbors, and the areas under $f(x)$ for subintervals 3, 6, and 10 are smaller than their neighbors. Therefore, the ABS algorithms can find only six local extrema. Example 2 showed that there are sixteen extrema for $f(x)$ in the interval $[-500, 500]$, as shown in Fig. 5.

Because the width of subintervals equals 100, which is very wide for this example, each subinterval may contain more than one maximum (minimum). See Fig. 6.

Fig. 6 reveals that the function $f(x)$ exhibits two local minima and one local maximum within the subinterval $[0, 100]$. In contrast, Table 4 accounts for only a single local minimum in this region. This discrepancy highlights a key limitation of the ABS algorithm: when subintervals are too wide, the method may fail to capture all local extrema. This observation underscores the importance of resolution in extrema detection—a factor that is systematically explored in the following MATLAB-based analysis.

4. Matlab-based numerical analysis

4.1. Methodology and parameter configurations

We developed a comprehensive benchmarking framework in MATLAB to conduct a rigorous numerical comparison between the proposed ABS method and two established approaches: uniform grid sampling as a simple Point-Based Search (PBS) and Genetic Algorithms (GA). The evaluation focuses on the robustness of each method across varying noise levels and resolution settings. In all experiments, we concentrate on identifying local and global maxima, as outlined in Algorithm 1. While many tasks today are addressed using large language models (LLMs), these models were not considered as baselines here, as univariate extrema detection falls outside their intended domain. LLMs are not designed for deterministic numerical optimization or the analysis of continuous function structures. A Monte Carlo simulation framework was employed with 1,000 repetitions per configuration to ensure statistical robustness.

We select three smooth, multimodal test functions, each defined on the interval [−5,5]. These functions represent diverse challenges in frequency and peak structure:

- $f_1(x) = \sin(5x) \cdot e^{-0.1x^2}$
- $f_2(x) = \cos(3x) \cdot e^{-0.2x^2}$
- $f_3(x) = \sin(2x) + 0.5 \cdot \cos(4x)$These cases provide a representative range of structural complexity for evaluating the optimization behavior of ABS, PBS, and GA. To simulate real-world measurement imperfections, we add Gaussian white noise to the function values. The noisy function is computed as:

$$f_i^{noisy}(x) = f_i(x) + \sigma \cdot \varepsilon, \varepsilon \sim \mathcal{N}(0, 1)$$

where $\mathcal{N}(0, 1)$ denotes the standard normal distribution, and σ controls the noise level. Increasing σ leads to greater distortion in the function's peaks and valleys, mimicking uncertainty in practical scenarios. We test across five noise levels: $\sigma \in \{0, 0.01, 0.05, 0.1, 0.2\}$, ranging from noise-free to highly perturbed settings. This aspect is critical for assessing the robustness of all three methods under uncertainty.

Table 4
Subintervals of width 100 units and their corresponding areas

k	Subinterval	S_k	k	Subinterval	S_k
1	[-500, −400]	23,773	6	[0, 100]	−1258
2	[-400, −300]	−2040	7	[100, 200]	42
3	[-300, −200]	−3427	8	[200, 300]	3427
4	[-200, −100]	−42	9	[300, 400]	2040
5	[-100, 0]	1258	10	[400, 500]	−23773

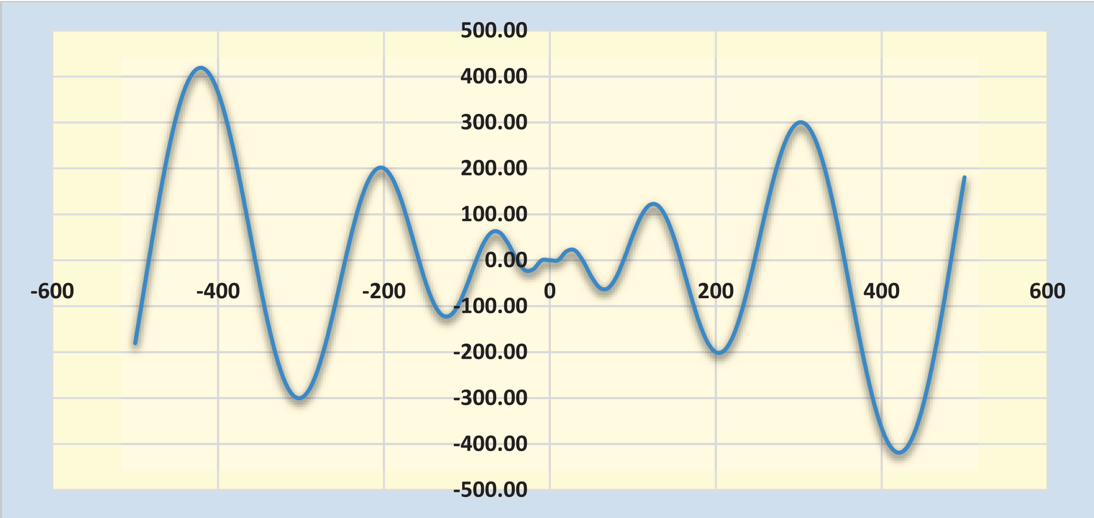


Fig. 5. The sixteen local extrema of the multimodal function $f(x)$ over the interval [-500, 500].

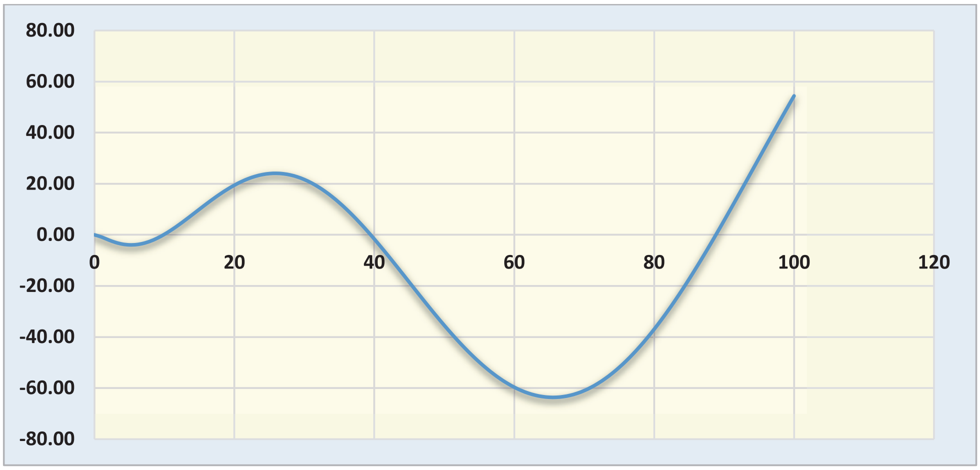


Fig. 6. Zoomed view of the local extrema of $f(x)$ on the subinterval [0, 100].

ABS and PBS share the goal of detecting local maxima, but they differ in how they explore the domain. In ABS, the input domain is divided into $n \in \{50, 100, 200, 500, 1000\}$ subintervals. Within each interval, we compute the area under the curve using the trapezoidal rule. A second-order analysis of the resulting area sequence is used to detect local maxima. In PBS, the function is sampled at $m \in \{50, 100, 200, 500, 1000\}$ evenly spaced points, and local maxima are identified by sign changes in the discrete derivative. In both methods, resolution reflects the granularity with which the domain $[-5,5]$ is scanned:

- For PBS, it corresponds to the number of sampling points m. Higher m improve sensitivity to closely spaced or narrow maxima.
- For ABS, resolution equals the number of intervals n. Finer segmentation enhances sensitivity to local extrema and offers robustness by integrating noisy data within segments.

Higher resolution improves fidelity at the cost of increased computational complexity, while lower resolution may lead to missed or imprecise peak detection. To quantify local maxima detection performance, we employ a coverage metric defined as:

$$coverage = \frac{number\ of\ detected\ maxima\ within\ tolerance}{number\ of\ true\ maxima}$$

Detection is considered successful if the absolute difference between the detected and true positions satisfies:

$$|x_{detected} - x_{true}| < 0.1.$$

This tolerance ensures that minor numerical deviations do not penalize otherwise accurate detections. We implement a simple Genetic Algorithm (GA) to extend the analysis to global optimization. A set of core hyperparameters defines the GA test scenarios:

- Population sizes: {10, 30, 50}
- Generations: {30, 50}
- Crossover rates: {0.3, 0.7}
- Mutation rates: {0.05, 0.1}
- Mutation strengths: {0.05, 0.1}

We conduct a comprehensive parameter sweep and perform extensive Monte Carlo simulations for all three approaches: ABS, PBS, and GA. Each configuration is repeated independently over a large number of trials (default: 1,000). This ensures analytical robustness and captures performance variability due to noise. It also accounts for algorithmic stochasticity, which is particularly relevant for the GA. For each configuration (function, noise level, and resolution), we performed 1,000 Monte Carlo trials. In each trial, an independent realization of the additive Gaussian noise was generated and added to the function values. For GA, the initial population was also reinitialized in every trial. Thus, all reported statistics are based on independent noise realizations and independent runs of the stochastic algorithm. To provide a consistent baseline for comparing ABS and GA, the ABS resolution is fixed at 300 subintervals, representing a balanced mid-range granularity. For each evaluated setting, we compute the average coverage values (for ABS and PBS), as well as the mean position errors (in terms of the x-coordinates) and value errors (in terms of the corresponding $f(x)$-values) for ABS and GA. This repeated and controlled evaluation facilitates a rigorous and fair comparison of the accuracy and resilience of each method across different levels of noise and uncertainty.

4.2. Results and discussion

The MATLAB experiments provide both visual and numerical insights into the comparative performance of the evaluated methods under varying levels of noise and resolution, as detailed in the preceding section. For each of the three benchmark functions, 3D surface plots illustrate the coverage behavior of ABS and PBS across a grid of resolution and noise combinations. These plots report the mean coverage ratios computed over 1,000 Monte Carlo repetitions, offering a statistically robust depiction of performance. Figs. 7–9 summarize these results.

ABS consistently outperforms PBS for two of the three benchmark functions (Figs. 7 and 8), achieving higher coverage across all resolution and noise combinations. This indicates greater robustness in detecting local optima, especially under noise. For the third function, $f_3(x) = \sin(2x) + 0.5 \cdot \cos(4x)$, the picture is more nuanced: at low noise and moderate to coarse resolutions, PBS performs slightly better, but as noise increases, ABS regains the advantage and yields higher coverage at low to mid resolutions. At a resolution of

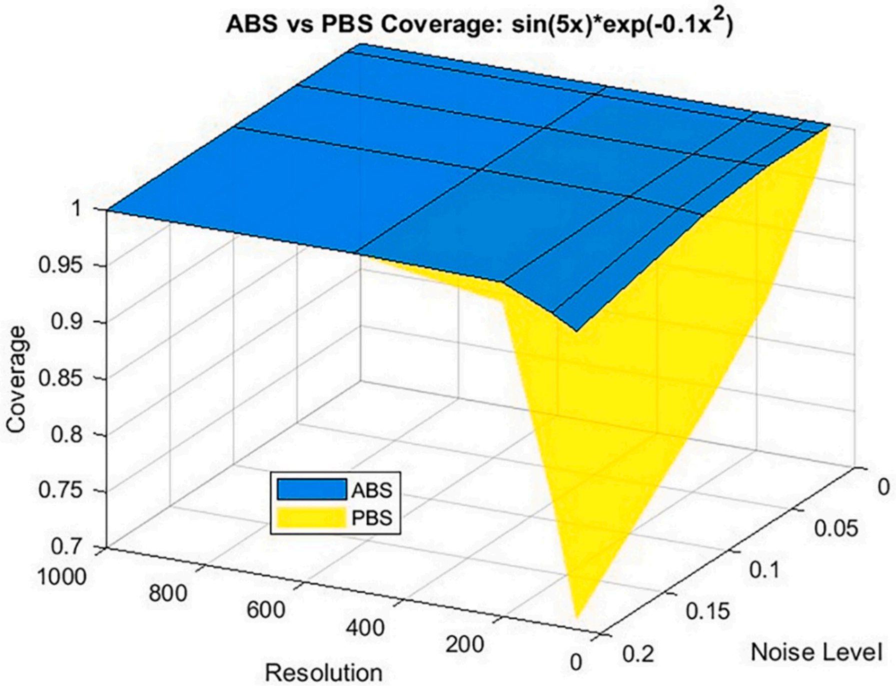


Fig. 7. Average Coverage of ABS vs. PBS for $f_1(x)$.

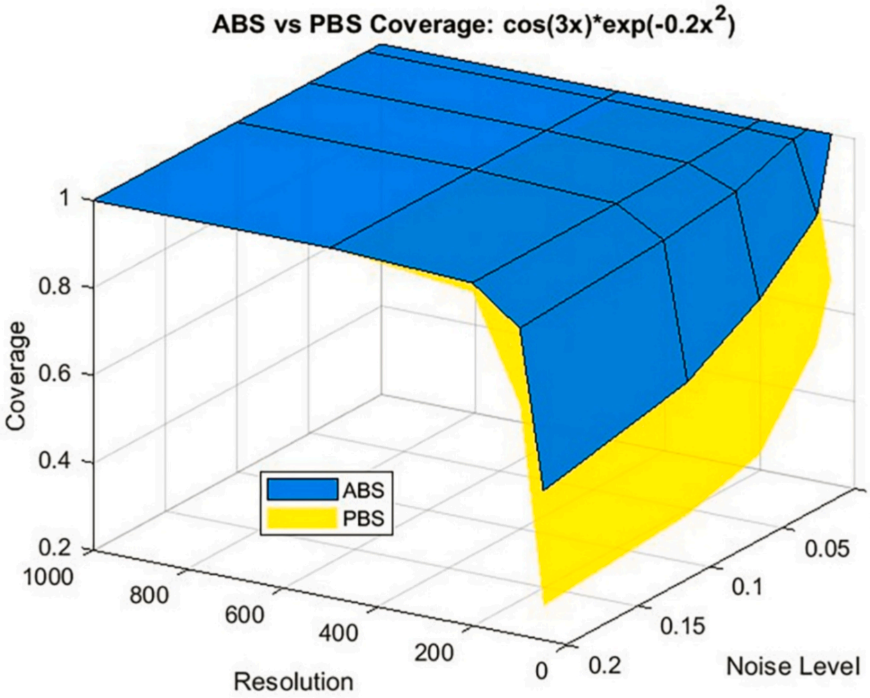


Fig. 8. Average Coverage of ABS vs. PBS for $f_2(x)$.

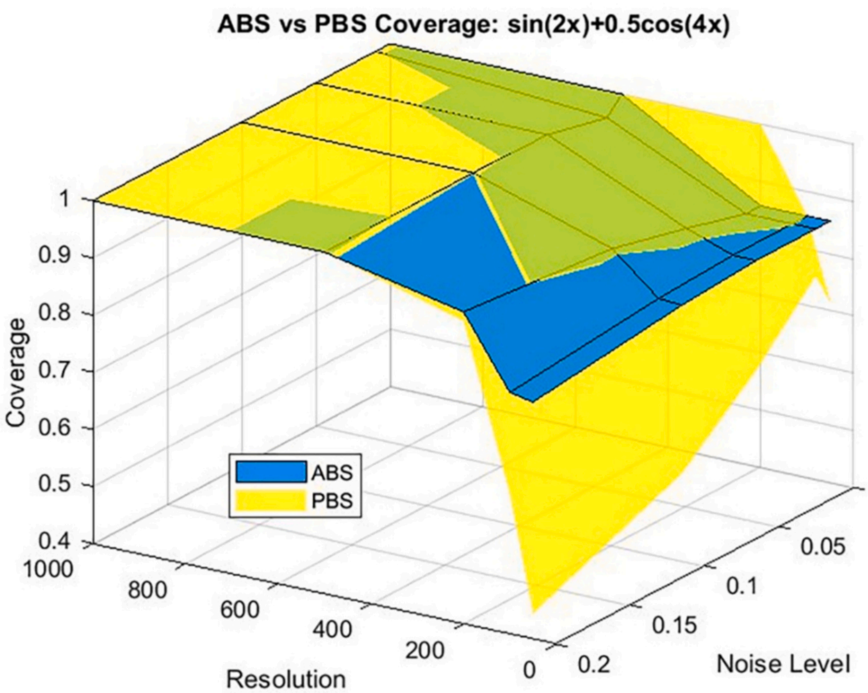


Fig. 9. Average Coverage of ABS vs. PBS for $f_3(x)$.

500, both methods perform almost identically across all noise levels, with mean coverage approaching 1.

In addition to the mean coverage surfaces in Figs. 7–9, we computed the standard deviation of coverage across Monte Carlo trials for each resolution–noise pair. For ABS, the standard deviations range from 0 (noise-free) to approximately 0.075; for PBS, they range from 0 to approximately 0.18. Thus, PBS exhibits higher variability than ABS in several settings, while ABS remains relatively stable across independent noise realizations. These values indicate that the differences observed in the mean coverage plots are not artifacts of a few unstable runs. Overall, ABS maintains high coverage even under significant noise, whereas PBS behaves more erratically, especially at low resolutions and higher noise levels, reflecting its sensitivity to noisy pointwise samples.

To assess the accuracy of ABS and GA in detecting the global maximum of each function, we recorded the mean position errors (based on the x-coordinates) and value errors (based on the corresponding $f(x)$-values) for both methods. For GA, these metrics were tracked across all 48 tested parameter constellations as the noise level increased. A numerical summary of the comparison between ABS and GA is given in Table 5. For each test function and noise level, the table reports the number of GA parameter configurations that achieve a lower mean position error than ABS ("better than ABS") and the number that perform worse ("worse than ABS"). These counts were obtained from a post-analysis of the simulation data: for each of the 48 GA configurations, the position error was averaged over 1,000 Monte Carlo trials per noise level and test function. ABS outperformed most GA configurations at every noise level, often all of them, as reflected in the "worse than ABS" column. Only at the highest noise level ($\sigma = 0.2$) did some GA configurations show improved performance, with up to 12 configurations outperforming ABS for one function. For ABS, the average position and value errors remained consistently low across all noise levels, reflecting its stability and deterministic nature. GA only slightly outperformed ABS in a few specific parameter constellations under high-noise conditions and exhibited significantly greater variability, particularly as the noise level increased.

Detailed line plots of the position and value errors for all GA configurations and ABS are provided in Appendix A (Figs. A1–A3) for completeness.

Key numerical performance metrics:

- ABS achieved 100 % detection coverage across 1,000 trials in the noise-free setting. The first two benchmark functions exceeded 80 % coverage at resolutions of 100 or higher, while the third function achieved comparable performance even at a resolution of 50. At resolutions above 500, ABS consistently reached near-perfect coverage across all noise levels.
- On average, ABS outperformed PBS with a 4.75 % improvement in detection coverage, reaching a maximum gain of 37.56 % in the most challenging test case.
- Across all noise levels and test functions, ABS reduced position error by 30.41 % and value error by 36.89 % compared to GA, using a resolution of 300.
- ABS demonstrated greater consistency across trials, exhibiting lower or comparable variance than GA and significantly lower variability than PBS.

These results highlight ABS's ability to maintain high accuracy and stability across noisy and noiseless scenarios without requiring hyperparameter tuning, in contrast to GA's performance sensitivity and PBS's variability.

5. Discussion and implications

This study introduced a novel approach for identifying all local and global extrema of univariate functions and developed the ABS algorithms based on this concept. The results carry both theoretical and practical implications.

5.1. Theoretical implications

Traditional heuristics, such as binary search, are commonly used to approximate the extrema of univariate functions. However, these methods may fail when faced with complex functions or wide domains. Metaheuristics—categorized by Mirjalili et al. [31] into evolutionary, physics-based, and swarm intelligence algorithms—often offer better estimates due to their flexibility and adaptability.

Table 5
GA Configuration Comparison per Function

test function	noise σ	better than ABS	worse than ABS
$f_1(x)$	0.00	0	48
	0.01	0	48
	0.05	0	48
	0.10	0	48
	0.20	12	36
$f_2(x)$	0.00	5	43
	0.01	0	48
	0.05	0	48
	0.10	3	45
	0.20	10	38
$f_3(x)$	0.00	0	48
	0.01	0	48
	0.05	0	48
	0.10	0	48
	0.20	2	46

Yet, all these methods, whether heuristic or metaheuristic, share a critical limitation: they are point-based search (PBS) algorithms that examine a limited number of points in each iteration. As a result, they may become trapped in local optima and fail to locate the global extremum.

The ABS approach proposed in this study is fundamentally different. Instead of iterating through discrete points, ABS evaluates entire subintervals using definite integration, estimating the sum of function values across each segment. This area-based perspective increases the likelihood of detecting all local extrema and selecting the most promising one as the global optimum. Unlike PBS algorithms, which only seek the global optimum and may overlook local features, ABS attempts to capture the complete extremum structure of a function systematically. Therefore, ABS represents a distinct and complementary methodology that is not directly comparable to existing PBS heuristics and metaheuristics.

From a computational standpoint, the ABS method requires the evaluation of subinterval integrals over n subintervals and a linear scan of the resulting area sequence $\{S_k\}_{k=1}^{n}$. If numerical integration on each subinterval is performed with a fixed accuracy, the overall cost of ABS grows linearly with the number of subintervals, that is, $O(n)$, with a constant factor that depends on the chosen quadrature scheme. A simple point-based search (PBS) with m grid points also scales linearly, $O(m)$, but relies on single-point evaluations instead of subinterval integrals. In contrast, a Genetic Algorithm (GA) evaluates a population of p candidate solutions over g generations, leading to $O(pg)$ function evaluations per run. In our experiments (Section 4.2), using $n = 300$ subintervals for ABS provided a favorable trade-off between accuracy and runtime. For the GA, comparable performance required substantially larger numbers of function evaluations and remained more sensitive to the choice of algorithmic parameters.

5.2. Practical implications

From a practical standpoint, ABS offers several advantages over conventional PBS algorithms:

- ABS is designed to detect all local extrema, not just global ones. This capability significantly reduces the risk of misidentifying a local extremum as the global optimum.
- In real-world decision-making, local optima may be more practical or feasible than the global solution. ABS enables decision-makers to evaluate all available extrema and select the one that best fits their constraints (e.g., resource limitations).
- ABS provides both the locations (x-values) and the corresponding function values (y-values) of all local extrema, facilitating informed trade-offs among multiple solutions.
- When applied to time-based functions, ABS can track changes in extrema over time. For instance, it can reveal how peak performance or losses evolve, which is useful in operational or behavioral contexts.

The proposed ABS method offers several advantages in applied optimization settings. Unlike point-based heuristics that often converge prematurely to local optima, ABS enables comprehensive domain exploration. It is particularly well-suited for problems requiring awareness of all local and global extrema. This is critical in applications such as control systems, engineering design, and decision modeling, where understanding the full spectrum of optima informs robustness and trade-off analysis. ABS demonstrated greater resilience in environments subject to noise, such as sensor signal processing or real-time decision systems, as its area-based integration naturally dampens noise effects without requiring complex filtering. Its deterministic structure and single-parameter control (resolution) make it more interpretable and reproducible than GA, which involves multiple stochastic parameters. Furthermore, ABS can be seamlessly integrated into hybrid pipelines: for example, it may first identify candidate regions for maxima, which can then be locally refined using conventional optimizers.

By enhancing detection coverage, reducing positional and value errors, and offering greater stability under noise, ABS is a reliable tool in academic research and real-world applications that require full-spectrum extremum analysis.

These advantages suggest broad applicability. For example, recent studies using metaheuristics for tasks such as feature selection or optimization in behavioral modeling (e.g., [37,38,39]could benefit from ABS as a more robust alternative. In general, by revealing all local extrema, ABS provides decision-makers with a richer analytical foundation than methods focused solely on global optima.

6. Challenges and future works

This study introduced the ABS method as a means of detecting all local and global extrema of univariate functions. Its reliance on subinterval-based integration offers a different framework from traditional point-based search (PBS) algorithms. At the same time, this paradigm raises challenges in terms of resolution choice, computational cost, and scalability.

A first challenge is selecting the number of subintervals. ABS divides the search interval into equal-width subintervals to approximate the cumulative function's behavior. If the subintervals are too wide, the algorithm may miss narrow extrema. If they are too narrow, the computational burden increases significantly. This trade-off between accuracy and efficiency suggests the need for adaptive mechanisms that adjust the resolution based on the function's behaviour. Future work could, for example, use indicators of smoothness or curvature to refine intervals only where the function changes rapidly, while keeping coarser intervals elsewhere. Wide search intervals further exacerbate the resolution problem. Uniformly narrow subintervals across a large domain are computationally expensive. Several strategies could address this. One option is to run the ABS algorithm multiple times with different subinterval widths to capture both coarse- and fine-scale extrema. Another is to shift the starting point of the segmentation, applying the same resolution with different offsets to improve coverage. A particularly promising approach is to use ABS in a hybrid fashion: ABS first identifies subintervals with high potential—where area comparisons suggest extrema—and then a local point-based method (e.g., binary search) or a metaheuristic refines the search within those regions. This idea is already partially implemented in Algorithms 3 and 4 and can be extended further. A fully iterative ABS variant is also conceivable, where the algorithm recursively splits the most promising subintervals and refines the search space until a stopping criterion is met. Such a design would resemble deterministic zoom-in or tree-based optimization procedures.

Extending ABS to multivariate functions is conceptually straightforward but computationally challenging. In higher dimensions, the concept of “area” generalizes to “volume” or “hypervolume,” and the domain must be partitioned into numerous hyperrectangular subregions. For each region, the multivariate function must be integrated and compared with its neighbouring regions. This leads to rapid growth in the number of regions as the dimension increases (the curse of dimensionality) and to substantially higher integration costs. As a result, a naive uniform partition is only feasible in low dimensions or at coarse resolutions. Practical multidimensional ABS variants would require adaptive partitioning, sparse or low-rank integration schemes, or hybrid methods that use ABS to identify promising regions and then apply local optimizers within them. Investigating such adaptive and hybrid extensions is a crucial direction for future work, particularly in constrained settings where integration must be restricted to feasible regions.

The present experiments focus on additive Gaussian noise. An important extension is to evaluate ABS under non-Gaussian disturbances, such as impulsive or heavy-tailed noise and correlated noise processes. These cases are common in engineering and signal-processing applications, providing a more stringent test of robustness. Additionally, the current comparison utilizes PBS and GA as representative point-based and metaheuristic baselines. Future studies could extend this benchmark to include more recent metaheuristics, such as the Grey Wolf Optimizer and Differential Evolution, as well as related swarm or evolutionary algorithms, to situate ABS within the broader optimization landscape further.

Beyond direct extensions of the ABS algorithm, it may be useful to integrate ABS principles into metaheuristic frameworks. For example, evolutionary or swarm intelligence methods could evolve populations of subintervals instead of individual points and evaluate them via integrated function values. This would form a class of area-based metaheuristics and could improve robustness under noise and uncertainty. Such ideas are related to applications in information fusion and decision-making, where integrated behaviour over regions can be more informative than isolated samples.

Finally, the core principle of ABS—evaluating and comparing cumulative function values across segments—can be applied to other problem classes. Examples include root-finding in univariate functions, discrete optimization (via aggregated scores over subsets), and financial or operational modelling where local and global extrema of performance profiles are of interest. Overall, ABS reduces the reliance on isolated sampling and leverages continuous, integrated information. Building on this foundation through adaptive strategies, hybrid frameworks, multidimensional extensions, and domain-specific applications offers several promising avenues for future research.

7. Conclusion

This study developed ABS algorithms to identify all local and global extrema of univariate functions $f(x)$ using a straightforward approach. The method divides a given interval into several equal-width subintervals and computes their respective areas under the function using definite integration. A subinterval whose area is larger (or smaller) than its neighboring subintervals is likely to contain at least one local maximum (or minimum). While this assumption may not always hold, particularly when subintervals are too wide, the approach systematically identifies potential extrema. In contrast to Point-Based Search (PBS) algorithms, which evaluate the function at selected points and search iteratively, the ABS method aggregates information across continuous ranges, reducing the risk of missing optima. ABS algorithms offer several advantages. Unlike PBS methods, which focus solely on global optima, ABS seeks all extrema, providing decision-makers with a broader set of alternatives. This is especially useful in real-world scenarios where the global optimum may not be feasible or desirable. However, the approach also has limitations. It is currently applicable only to integrable univariate functions, and its performance is highly sensitive to the number of subintervals used.

To complement the conceptual development, we conducted a thorough MATLAB-based analysis that compared the performance of ABS with both PBS and Genetic Algorithms (GA) across different levels of noise and resolution. The numerical experiments provided clear visual and statistical evidence that ABS consistently achieves higher coverage and lower errors, particularly under uncertain or noisy conditions. These findings reinforce the practical strength of ABS, demonstrating its robustness and reliability.

That said, choosing too few subintervals may result in missed extrema due to insufficient resolution, while choosing too many subintervals can make the computation unnecessarily complex. Finding the optimal balance between accuracy and computational effort remains a key direction for future research.

In summary, ABS offers a novel and practical alternative to classical search methods, particularly in noisy or uncertain environments. Its ability to consider the full domain and detect multiple extrema makes it a promising tool in theoretical and applied optimization tasks.

CRediT authorship contribution statement

Madjid Tavana: Writing – review & editing, Writing – original draft, Visualization, Methodology, Formal analysis, Conceptualization. **Hosein Arman:** Writing – review & editing, Writing – original draft, Validation, Methodology, Formal analysis. **Andreas Dellnitz:** Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation.

Ethics approval and consent to participate

Not Applicable.

Consent for publication

All authors have agreed to this submission

Funding

Not Applicable.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Not Applicable

Appendix A

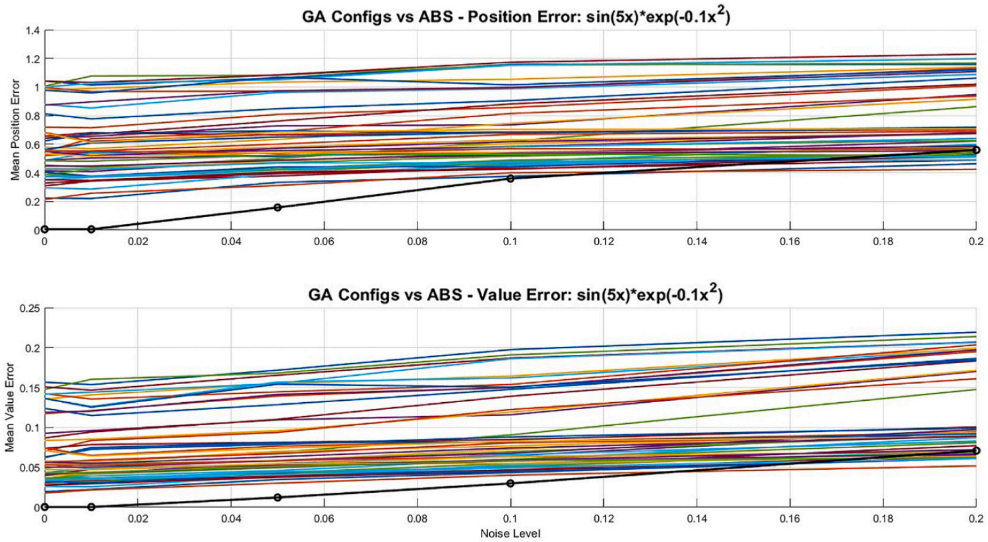


Fig. A1. Average Position and Value Errors of ABS vs. GA for $f_1(x)$.

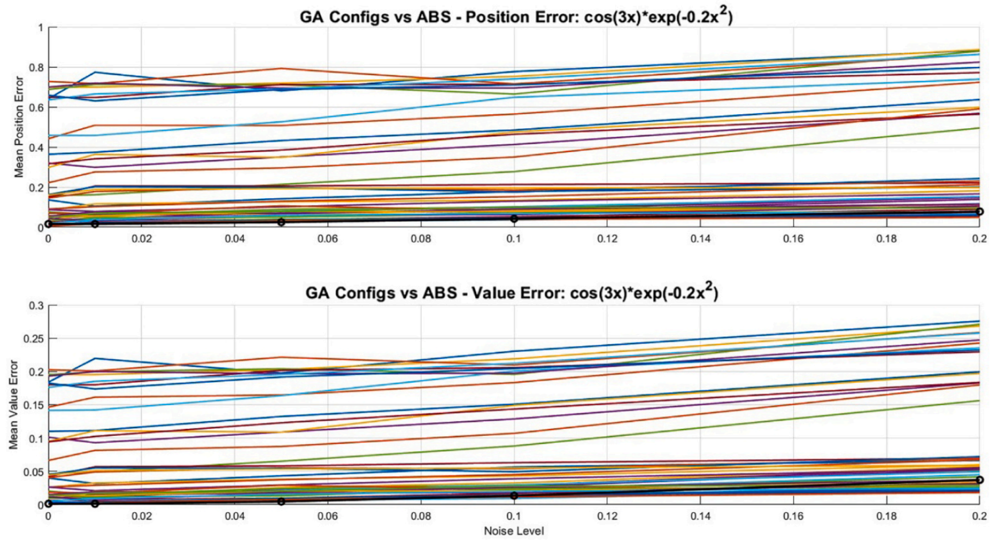


Fig. A2. Average Position and Value Errors of ABS vs. GA for $\boldsymbol{f_2(x)}$.

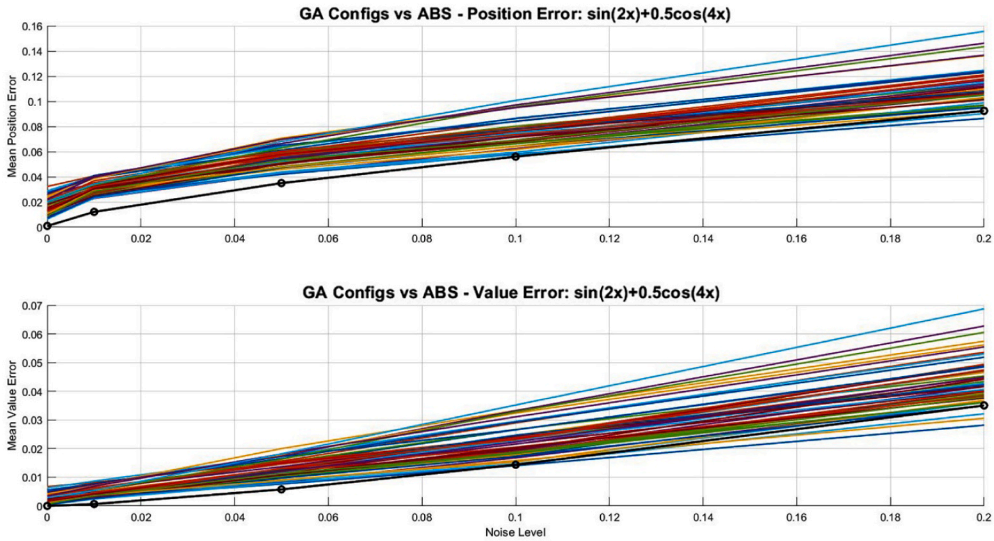


Fig. A3. Average Position and Value Errors of ABS vs. GA for $\boldsymbol{f_3(x)}$.

Data availability

Data will be made available on request.

References

[1] A. Peccini, L.F.S. Jesus, A.R. Secchi, M.J. Bagajewicz, A.L.H. Costa, Globally optimal distillation column design using set trimming and enumeration techniques, Comput. Chem. Eng. 174 (2023) 108254, https://doi.org/10.1016/j.compchemeng.2023.108254.

[2] M. Moriwaki, J.J. Herrera Velázquez, J. Cabrera Ruiz, K. Matsuda, J.R. Alcántara-Avila, Synthesis of hybrid membrane distillation processes with optimal structures for ethanol dehydration, Comput. Chem. Eng. 178 (2023) 108385, https://doi.org/10.1016/j.compchemeng.2023.108385.

[3] T. Zhou, X. Han, L. Wang, W. Gan, Y. Chu, M. Gao, A multi-objective differential evolution algorithm with subpopulation region solution selection for global and local Pareto optimal sets, Swarm Evol. Comput. 83 (2023) 101423, https://doi.org/10.1016/j.swevo.2023.101423.

[4] K.S. Oh, J.W. Lee, Auxiliary algorithm to approach a near-global optimum of a multi-objective function in acoustical topology optimization, Eng. Appl. Artif. Intel. 117 (Part A) (2023) 105488, https://doi.org/10.1016/j.engappai.2022.105488.

[5] Y. Liu, X. Chen, K. Wang, S. Diao, Y. Huang, H. Li, L. Wu, A Chebyshev metamodel based BnB approach to efficiently search global optimum for 3D ICP point set registration, Comput. Aided Geom. Des. 101 (2023) 102178, https://doi.org/10.1016/j.cagd.2023.102178.

[6] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1) (1992) 66–73, https://doi.org/10.1038/scientificamerican0792-66.

[7] R. Storn, K. Price, Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (1997) 341–359, https://doi.org/10.1023/A:1008202821328.
[8] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (2) (1999) 82–102, https://doi.org/10.1109/4235.771163.
[9] N. Hansen, S.D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), Evol. Comput. 11 (1) (2003) 1–18, https://doi.org/10.1162/106365603321828970.
[10] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (6) (2008) 702–713, https://doi.org/10.1109/TEVC.2008.919004.
[11] B. Webster, P.J. Bernhard, A local search optimization algorithm based on natural principles of gravitation, in: Proceedings of the International Conference on Information and Knowledge Engineering (IKE'03), CSREA Press, 2003, pp. 255–261.
[12] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248, https://doi.org/10.1016/j.ins.2009.03.004.
[13] O.K. Erol, I. Eksin, A new optimization method: big bang–big crunch, Adv. Eng. Softw. 37 (2) (2006) 106–111, https://doi.org/10.1016/j.advengsoft.2005.04.005.
[14] H. Shah-Hosseini, Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation, Int. J. Comput. Sci. Eng. 6 (2011) 132–140, https://doi.org/10.1504/IJCSE.2011.041221.
[15] A. Hatamlou, Black hole: a new heuristic optimization approach for data clustering, Inf. Sci. 222 (2013) 175–184, https://doi.org/10.1016/j.ins.2012.08.023.
[16] Du, H., Wu, X., & Zhuang, J. (2006). Small-world optimization algorithm for function optimization. In Jiao, L., Wang, L., Gao, X., Liu, J., & Wu, F. (Eds.), *Advances in Natural Computation. ICNC 2006. Lecture Notes in Computer Science* (Vol. 4222, pp. 264–273). Springer. doi:10.1007/11881223_33.
[17] R.A. Formato, Central force optimization: a new metaheuristic with applications in applied electromagnetics, Prog. Electromagn. Res. 77 (2007) 425–491, https://doi.org/10.2528/PIER07082403.
[18] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, Acta Mech. 213 (2010) 267–289, https://doi.org/10.1007/s00707-009-0270-4.
[19] B. Alatas, ACROA: Artificial chemical reaction optimization algorithm for global optimization, Expert Syst. Appl. 38 (10) (2011) 13170–13180, https://doi.org/10.1016/j.eswa.2011.04.126.
[20] A. Kaveh, M. Khayatazad, A new meta-heuristic method: Ray optimization, Comput. Struct. 111–112 (2012) 283–294, https://doi.org/10.1016/j.compstruc.2012.09.003.
[21] H.A. Abbass, MBO: Marriage in honey bees optimization—A haplometrosis polygynous swarming approach, in: In *Proceedings of the Congress on Evolutionary Computation*, IEEE, 2001, pp. 207–214, 10.1109/CEC.2001.934391.
[22] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE Comput. Intell. Mag. 1 (4) (2007) 28–39, https://doi.org/10.1109/MCI.2006.329691.
[23] Pinto, P. C., Runkler, T. A., & Sousa, J. M. C. (2007). Wasp swarm algorithm for dynamic MAX-SAT problems. In Beliczynski, B., Dzielinski, A., Iwanowski, M., & Ribeiro, B. (Eds.), *Adaptive and Natural Computing Algorithms. ICANNGA 2007. Lecture Notes in Computer Science* (Vol. 4431, pp. 350–357). Springer. doi: 10.1007/978-3-540-71618-1_39.
[24] X.-S. Yang, Firefly algorithm, stochastic test functions and design optimisation, International Journal of Bio-Inspired Computation 2 (2) (2010) 78–84, https://doi.org/10.1504/IJBIC.2010.032124.
[25] W.-T. Pan, A new fruit fly optimization algorithm: taking the financial distress model as an example, Knowl.-Based Syst. 26 (2012) 69–74, https://doi.org/10.1016/j.knosys.2011.07.001.
[26] A. Mucherino, O. Seref, Monkey search: a novel metaheuristic search for global optimization, AIP Conf. Proc. 953 (1) (2007) 162–173, https://doi.org/10.1063/1.2817338.
[27] X.-S. Yang, S. Deb, Cuckoo search via Lévy flights, in: *World Congress on Nature & Biologically Inspired Computing (NaBIC)*, IEEE, 2009, pp. 210–214, 10.1109/NABIC.2009.5393690.
[28] Y. Shiqin, J. Jianjun, Y. Guangxing, A dolphin partner optimization, in: *WRI Global Congress on Intelligent Systems*, IEEE, 2009, pp. 124–128, 10.1109/GCIS.2009.464.
[29] A. Askarzadeh, A. Rezazadeh, A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: Bird mating optimizer, Int. J. Energy Res. 37 (10) (2012) 1196–1204, https://doi.org/10.1002/er.2915.
[30] A.H. Gandomi, A.H. Alavi, Krill herd: a new bio-inspired optimization algorithm, Commun. Nonlinear Sci. Numer. Simul. 17 (12) (2012) 4831–4845, https://doi.org/10.1016/j.cnsns.2012.05.010.
[31] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61, https://doi.org/10.1016/j.advengsoft.2013.12.007.
[32] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67, https://doi.org/10.1016/j.advengsoft.2016.01.008.
[33] M.H. Nadimi-Shahraki, H. Zamani, Z. Asghari Varzaneh, A.S. Sadiq, S. Mirjalili, A systematic review of applying grey wolf optimizer, its variants, and its developments in different internet of things applications, Internet Things 26 (2024) 101135, https://doi.org/10.1016/j.iot.2024.101135.
[34] L. Wu, Z. Wang, Z. Liao, D. Xiao, P. Han, W. Li, Q. Chen, Multi-day tourism recommendations for urban tourists considering hotel selection: a heuristic optimization approach, Omega 126 (2024) 103048, https://doi.org/10.1016/j.omega.2024.103048.
[35] L. Gan, Q. Xiong, X. Chen, Z. Lin, W. Jiang, Optimal dispatch schedule for the coordinated hydro-wind-photovoltaic system with non-priority output utilizing combined meta-heuristic, Omega 131 (2025) 103198, https://doi.org/10.1016/j.omega.2024.103198.
[36] M. Tavana, M. Soltanifar, A. Dellnitz, A novel linear-integral TOPSIS approach to Mars mission simulator planning at NASA, Omega 137 (2025) 103350, https://doi.org/10.1016/j.omega.2025.103350.
[37] B. Liang, S. Han, W. Li, G. Huang, R. He, Spatial-temporal alignment of time series with different sampling rates based on cellular multi-objective whale optimization, Inf. Process. Manag. 60 (1) (2023) 103123, https://doi.org/10.1016/j.ipm.2022.103123.
[38] H. Lin, C. Wang, Q. Hao, A novel personality detection method based on high-dimensional psycholinguistic features and improved distributed Gray Wolf Optimizer for feature selection, Inf. Process. Manag. 60 (2) (2023) 103217, https://doi.org/10.1016/j.ipm.2022.103217.
[39] P. Rasappan, M. Premkumar, G. Sinha, K. Chandrasekaran, Transforming sentiment analysis for e-commerce product reviews: Hybrid deep learning model with an innovative term weighting and feature selection, Inf. Process. Manag. 61 (3) (2024) 103654, https://doi.org/10.1016/j.ipm.2024.103654.