Ain Shams Engineering Journal 15 (2024) 102997

Contents lists available at ScienceDirect

Ain Shams Engineering Journal

journal homepage: https://www.sciencedirect.com

Full Length Article

RGN: A Triple Hybrid Algorithm for Multi-level Image Segmentation with Type II Fuzzy Sets

Rohit Salgotra [a,b,c,*], Nitin Mittal [d], Abdulaziz S. Almazyad [e], Ali Wagdy Mohamed [f,*]

[a] Faculty of Physics and Applied Computer Science, AGH University of Kraków, Kraków, Poland
[b] Data Science Institute, University of Technology Sydney, 15 Broadway, Ultimo, NSW 2007, Australia
[c] MEU Research Unit, Middle Eastern University, Amman, Jordan
[d] Skill Faculty of Engineering and Technology, Shri Vishwakarma Skill University, Palwal, Haryana, 121102, India
[e] Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, P.O. Box 51178, Riyadh 11543, Saudi Arabia
[f] Operations Research Department, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza 12613, Egypt

ARTICLE INFO

Keywords:
Multiple algorithms
Image thresholding
Numerical optimization
Parametric adaptation
Naked mole rat algorithm

ABSTRACT

This paper presents a study focused on enhancing the effectiveness of cuckoo search (CS). The goal is to improve its performance in avoiding local optima, improve the exploration and exploit potentially new solutions. To achieve this, we incorporate three additional algorithms – grey wolf optimizer (GWO), red panda optimization (RPO), and naked mole rat algorithm (NMRA) – into the basic CS framework to strengthen its exploration and exploitation capabilities. The resulting hybrid algorithm is named RGN, standing for red panda, grey wolf and naked mole-rat. To make the parameters of the RGN algorithm adaptable, six new mutation operators and inertia weights are added to the proposed RGN algorithm. The proposed algorithm is tested on CEC 2005, CEC 2014, and CEC 2022 benchmark problems to prove its effectiveness. Friedman test and Wilcoxon rank-sum tests, are done to analyse the significance of the proposed RGN algorithm statistically. It has been found that the proposed RGN is significantly better with respect to LSHADE-SPACMA, SaDE, SHADE, CMA-ES, extended GWO, hierarchical learning particle swarm optimization (FHPSO), Kepler optimization algorithm (KOA), improved chef-based optimization algorithm (CBOADP), improved symbiotic herding optimization (IMEHO), blended-biogeography based optimization (B-BBO), and Laplacian BBO (LX-BBO), among others. Application of the proposed algorithm RGN for Multilevel Image Thresholding with Type II Fuzzy Sets, shows that it is better than other algorithms over various performance matrices including mean squared error (MSE), peak signal-to-noise ratio (PSNR), and structural similitude index (SSIM). Experimentally and statistically, it has been proved that the proposed RGN algorithm can be considered as a better alternative for optimization research.

1. Introduction

Optimization algorithms have emerged as indispensable tools across various domains, revolutionizing problem-solving by efficiently discovering optimal or near-optimal solutions in complex and challenging scenarios. From engineering to finance, operations research to machine learning, and engineering design to many others, their application has become ubiquitous, replacing manual trial-and-error approaches and enabling more effective decision-making [1].

The two main categories of meta-heuristic algorithms are evolutionary algorithms (EAs) and swarm intelligent algorithms (SIAs) [2]. EAs maintain a diverse population of solutions, employing genetic operators like mutation, crossover, and selection mechanisms to strike a balance between exploration and exploitation [3]. Some commonly known EAs differential evolution (DE) [4], genetic algorithm (GA) [3], fire hawk algorithm (FHA) [5], and others. On the other hand, SIAs draw inspiration from the collective behaviour and problem-solving abilities of social insect colonies or animal swarms, simulating the interactions of a population of agents to solve complex optimization problems [6]. Some of these algorithms include particle swarm optimization (PSO) [6], grey wolf optimization (GWO) [7,8], naked mole rat algorithm (NMRA) [9], marine predator algorithm (MPA) [10], salp swarm algorithm (SSA) [11,12], red panda optimization (RPO) [13], meerkat optimization algorithm (MOA) [14], flower pollination algorithm (FPA) [15], equilibrium optimizer (EO) [16], and many others.

* Corresponding authors.
E-mail addresses: rohits@agh.edu.pl, r.03dec@gmail.com (R. Salgotra), mittal.nitin84@gmail.com (N. Mittal), mazyad@ksu.edu.sa (A.S. Almazyad), aliwagdy@gmail.com, aliwagdy@staff.cu.edu.eg (A.W. Mohamed).

https://doi.org/10.1016/j.asej.2024.102997
Received 20 October 2023; Received in revised form 4 July 2024; Accepted 30 July 2024
Available online 10 August 2024

Despite their broad applicability and effectiveness, optimization algorithms face several challenges that researchers continually strive to address. One common issue is premature convergence, where algorithms settle on suboptimal solutions instead of reaching the global optimum. This may occur due to getting trapped in local optima or insufficient exploration of the search space. Additionally, as the dimensionality of the problem increases, due to the exponential nature of the search space, thus making it challenging for the optimization algorithm to find optimal solutions within a reasonable time. The computational complexity and evaluation time for numerous potential solutions can become prohibitive in higher-dimensional spaces, limiting the algorithm's effectiveness [2]. Furthermore, constraint handling, parameter tuning, and a lack of domain-specific knowledge can pose challenges, leading to suboptimal solutions. Researchers have been working on algorithmic enhancements, problem-specific adaptations, and parameter tuning to overcome these hurdles and improve the effectiveness of optimization algorithms [15].

In this context, the present work proposes a new multi-algorithm-multi-mutated based evolutionary framework named Red grey naked (RGN) algorithm. This novel approach draws on the best-known settings of three newly developed algorithms: RPO, GWO, and NMRA. Each of these algorithms has demonstrated high efficiency and prowess over various benchmark problems and real-world scenarios. RGN adopts a hybridization strategy, taking inspiration from NMRA's well-defined exploration (worker phase) and exploitation (breeder phase) operations. The exploitation phase in NMRA has already proven highly effective due to the presence of a best-guided solution for finding optimal solutions [2]. In the present work, the exploration operation is enhanced by integrating the exploration tendencies of RPO and GWO. The equations inspired by these algorithms contribute to improved exploration while maintaining exploitative tendencies, striking a better balance between the two aspects.

To further enhance the algorithm's performance, a dynamic adjustment throughout the optimization process is achieved using self-adaptive parameter settings. This dynamic adaptation is facilitated by incorporating six new mutation operators: ($mo's$) namely oscillating (osc) mo, chaotic ($chaos$) mo, simulated annealing (sa) mo, linearly decreasing (lin) mo, logarithmic (log) mo, and exponential decreasing (exp) mo. These algorithms are added by using iterative division. By iterative division, we mean that one algorithm is used for a certain set of iteration. This process is followed to make the algorithm follow a certain pattern based on the type of operation required. That is, if more exploration is required which set of equations must be used and if more exploitation is required, which equation must be used, and what types of equations are better for a balanced operation. More details on which equations have been used, is added in the consecutive sections. Note that prior study has been to identify which equation fares better for exploration or exploitation, respectively.

To assess the proposed RGN algorithm's applicability, extensive testing is performed on CEC benchmark problems namely CEC 2005 [2], CEC 2014 [17], CEC 2022 [18]. These benchmarks problems highly challenging and consists of unimodal (UM) problems, multimodal (MM) problems, fixed dimension (FD) problems, and Hybrid-Composite (HC) problems. These benchmark problems are widely recognized and provide a common ground for evaluating and comparing different algorithms. Apart from these benchmarks, the proposed RGN algorithm is applied for the real-world optimization of Multi-level Image Segmentation [19]. A comprehensive comparison is made with state-of-the-art algorithms, including archive based DE (JADE) [20], laplacian teacher learning algorithm (LX-TLA) [21], artificial rabbit optimization (ARO) [22], improved symbiotic organisms search (ISOS) [23], success history based DE (SHADE) [24], variable neighbourhood BA (VNBA) [25], laplacian teacher learning algorithm (LX-TLA) [21], LSHADE-SPACMA [16], self-adaptive DE (SaDE) [20], multi-hybrid algorithm (MHA) [19], random walk GWO (RW-GWO) [26], light spectrum optimizer (LSO) [27], hierarchical learning particle swarm optimization (FHPSO) [28], Kepler optimization algorithm (KOA) [29], and improved chef-based optimization algorithm (CBOADP) [30], among others.

In this work, the proposed Red Grey Naked mole rat algorithm utilizes Type-II fuzzy sets combined with entropy-based algorithms to perform multilevel thresholding of images. Image thresholding, as a simple and effective approach, finds applications in various areas such as pattern recognition, image processing, and related real-time systems [31]. By accurately separating pixels into different groups, thresholding facilitates tasks like object detection, feature extraction, and image analysis [32]. Bi-level thresholding is a method that involves choosing two thresholds to segment an image, while multilevel thresholding (MLT) partitions an image into multiple objects. MLT is often preferred over bilevel thresholding due to its ability to handle more complex segmentation tasks [33]. Various algorithms have been developed for MLT, with the main focus on methods such as maximum entropy [34], between-class variance [35], Renyi entropy [36], and fuzzy entropy-based approaches [37]. These algorithms have contributed to the advancement of MLT techniques. However, they are associated with drawbacks such as complexity and higher computational time.

To address these challenges, Zhao et al. [38] proposed a technique that utilizes histogram partitions with fuzzy membership values for MLT. Building upon this work, Tao et al. [39] introduced a fuzzy entropy-based system that combines fuzzy partitioning with optimization algorithms. These threshold-based image segmentation methods aim to accurately identify thresholds for effective image segmentation. Type II Fuzzy Sets, a mathematical framework for dealing with uncertainty and imprecision, can be utilized to handle complex dataset. Type II Fuzzy Sets are employed in these methods to incorporate ultra-fuzziness into the image segmentation process [40,41]. This incorporation of ultra-fuzziness using Type II Fuzzy Sets is crucial for obtaining optimal image thresholds in multilevel thresholding operations.

The rest of the paper is organized into six sections. The second section deals with the motivation, modifications and implementation details of the proposed RGN algorithm. The third section showcases the results obtained from testing the RGN algorithm of CEC benchmark problems, demonstrating its competitiveness compared to other algorithms. This section also applies the RGN algorithm to real world optimization of Multi-level Image Segmentation and compares it with various existing algorithms. The fourth section provides a discussion of the research, outlining drawbacks and providing future perspectives for the RGN algorithm. Finally, in the sixth section, future recommendations and important conclusions are drawn.

2. The proposal: hybrid red panda, grey wolf and naked mole-rat (RGN) algorithm

CS algorithm is an efficient algorithm that has shown to produce highly reliable results when compared to basic algorithms such as GWO, DE, PSO, and others. However, when compared to enhanced or hybrid versions of algorithms, CS's reliability decreases. This is primarily due to its limited ability to explore different solutions. A closer examination reveals that the algorithm lacks randomness, which increases the likelihood of getting stuck in local optima and reduces its overall performance [42]. To address this issue, new equations can be introduced for improved global search capabilities. Additionally, the parameters of the basic algorithm have a significant impact on CS's performance. Analysing and adjusting the parameters can make the algorithm more self-sufficient and adaptive. Considering these factors, a new self-adaptive triple hybrid red panda, grey wolf, naked mole rat inspired RGN algorithm is proposed in this study. Major highlights of this work include:

- A new concept of iterative division is added to the basic structure of CS algorithm for proposing the new RGN algorithm.

- For the first iterative half, global search of CS powered by Cauchy mutation is used for enhanced exploration; and NMRA powered local search is used for better exploitation properties.
- The second iterative half uses a combination of GWO powered by CS for exploration; and RPO for exploitation or local search operation.
- We use six mutation operators $mo's$ to enhance the parameters of the proposed RGN algorithm. There are only three parameters, namely λ, pa, and R.
- For λ, we have used simulated annealing sa, exponential decreasing exp and sigmoidal $sig\ mo$; for R we have used chaotic $chaos$, oscillating osc, and logarithmic log; and finally in case of pa we are using exp, osc, and $log\ mo$.

A detailed discussion on the proposed algorithm's methodology and requirement is presented in the next subsections.

2.1. Proposal's requirement

The researchers are motivated by the no free lunch theorem to develop new algorithms tailored to their specific problem requirements [43]. According to this theorem, no single algorithm can be the best fit algorithm for all domain research problems. Therefore, modifications are necessary to enhance the performance of algorithms based on the specific characteristics of each problem. These characteristics include factors such as complexity, dimension size, and whether the problem is constrained or unconstrained. Additionally, challenges arise from the presence of multiple local minimal solutions and high peaks in a problem, which require researchers to develop enhanced versions of algorithms that can provide reliable results. One inherent drawback of the basic CS is its poor exploration capability, as it follows smaller step sizes, causing new solutions to cluster together and hinder the exploration process. Consequently, the algorithm tends to search only in specific regions of the search space, resulting in an inefficient search approach and overall a poor exploration operation.

Given these limitations, it is crucial to formulate new equations for the global and local search phase of the algorithm to prevent it from getting stuck in local minima and to enable better exploration. These additional equations are seamlessly integrated in a manner that preserves the fundamental structure of the basic algorithm. Conversely, the local search phase of the algorithm is dependable and showcases improved exploitation characteristics. This is because the global solution of any problem resides in proximity to the current best solutions, which are continuously updated to converge towards the ultimate solution. This strategy fosters enhanced exploitation by continuously evaluating each new solution against the previous best solution, consequently pinpointing the most optimal solution. To address these problems, a new algorithm RGN is proposed in the next subsection.

2.2. Why and how the modifications are added

In a generalized metaheuristic algorithm, the algorithm follows extensive exploration during the start, followed by exploitation towards the end. The exploration process is meant for improving the global search or exploring the search space in an extensive manner, whereas the exploitation operation is performed for finding solutions in a particular section of the search space. It has been proved that the use of Cauchy based operators help in the enhancing the exploration search [42], and when added to CS, the exploration search improves significantly [44]. Similarly, for exploitation, NMRA is found to have better searching capabilities and can be used for intensive exploitation of the search space [19]. This process is followed for the first half of the iteration, and is meant for more exploration and less exploitation. As the iterative process progresses, the search process should focus more on the exploitation operation and some exploration, so that the algorithm may not fall in some local optima. In order to deal with this problem, CS-GWO based enhancements are added for exploration, whereas for exploitation RPO is used. The CS-GWO based enhancements were introduced in [42], and it was found that it can help in better searching capabilities by providing new solutions in the vicinity of three random solutions. The RPO based modifications are controlled by the parameters of RPO and are in general capable of extensive exploitative search [13]. Overall, these modifications help the algorithm to explore and exploit simultaneously without getting trapped in local optimal solutions.

2.3. Proposed RGN algorithm

In the present work, we are trying to design an algorithm which can be applied to all domains of optimization, with better exploration and exploitation properties, and overall lesser computational burden. We have incorporated GWO, RPO and NMRA into the basic CS algorithm for better performance. Iterative division is followed for enhanced operation and all the parameters are made self adaptive.

2.3.1. Initialization

The algorithm starts by the initialization of solutions between the upper $M_{max,j}$ bounds and the lower bounds $M_{min,j}$, for a D dimensional search space. This is generalized as

$$N_{i,j} = N_{min,j} + r \times (N_{min,j} - N_{max,j}) \tag{1}$$

where $i \in [1, 2, ..., n]$ corresponds to n individuals distributed along the search space, for $r \in [0, 1]$. After initialization, the algorithm follows iterative division for improved performance.

2.3.2. First iterative half

During the first half of iterations, the basic global search of CS is followed and is used in combination with Cauchy distribution for better exploration properties. The general equation is given by

$$x_i^{t+1} = x_i^t + \alpha \otimes L(\beta)(x_{best} - x_i^t) \tag{2}$$

where x_i^{t+1} is the new solution in the $t+1$ iteration, $\otimes$ is entry wise multiplication, x_i^t is the solution of the previous generation, x_{best} is the global best, $\alpha > 0$ is defined by D. The new solution is then calculated using Cauchy distribution. The enhanced equation is thus given as

$$x_i^{t+1} = x_i^t + \alpha \otimes Cauchy(\delta)(n_{best} - x_i^t) \tag{3}$$

The Cauchy randomization is given by

$$f_{Cauchy(0,g)}(\delta) = \frac{1}{\pi} \frac{g}{(g^2 + \delta^2)} \tag{4}$$

The distribution function is,

$$y = \frac{1}{2} + \frac{1}{\pi} arctan(\frac{\delta}{g}) \tag{5}$$

$$\delta = tan(\pi(y - \frac{1}{2})) \tag{6}$$

Thus a Cauchy distributed random number is generated between $[0, 1]$.

The local search during this phase is governed by the NMRA algorithm [9]. We use the breeder phase of NMRA for better exploitation, but to avoid over exploitation, the breeding parameter λ is made self adaptive. More details on how self adaptivity is added, is given in consecutive subsections. The general equation for this phase is given as

$$x_i^{t+1} = (1 - \lambda) y_i^t + \lambda(x_{best} - x_i^t) \tag{7}$$

Thus, addition of new equations using Cauchy randomization help in extensive exploration, and equations inspired by NMRA helps in intensive exploitation operation. Overall, a combination of exploration with exploitation makes the first iterative half, highly efficient in operation.

2.3.3. Second iterative half

For this set of iterations, a more rigorous approach needs to be established. We require an improved operation with better exploitation

tendencies and lower exploration properties. Thus, during the global search phase for the second half of iterations, we use a combination of equations inspired by the GWO algorithm [7]. The generalized equations are given by

$$x_1 = x_i - M_1(N_1.x_{new} - x_i^t); \quad x_2 = x_i - M_2(N_2.x_{new} - x_i^t);$$
$$x_3 = x_i - M_3(N_3.x_{new} - x_i^t) \tag{8}$$
$$x_{new}^{t+1} = \frac{x_1 + x_2 + x_3}{3} \tag{9}$$

Here x_{new} is the new solution and M_1, M_2, M_3 and N_1, N_2, N_3 are derived from M and N respectively, given as

$$M = 2m.r_1 - m; \quad N = 2.r_2 \tag{10}$$

where m decreases linearly from 2 to 0 iteratively, r_1 & $r_2 \in [0, 1]$. This search equation helps in intensive exploration and contributes to better search process.

The local search operation during this phase is controlled by using RPO [13], and is meant for least exploration and higher exploitation. The general equation during this phase is governed by two random variables R and I. For a better operation, $I = 1$, and R is self-adaptive in nature. The mathematical equation for this formulation is

$$x_i^{t+1} = x_i^t + (R \times x_p^t - I \times x_q^t) \tag{11}$$

This phase helps in providing better exploration and exploitation operation. And is the core of the proposed RGN algorithm. Apart from these phases, selection operation is performed for selecting the best solution in the current iteration.

2.3.4. Selection phase

In the last stage, the process involves choosing the most suitable individuals from both the current and previous best individuals that were generated in each iteration. If the fitness of the new solution, denoted as $f(x_{new})$, is superior to the previously known fitness $f(x_i^t)$, it replaces the previous solution. However, if the fitness of the previous solution, denoted as x_i, is better, it is kept. The selection operation is thus given by

$$x_{new}^{t+1} = \begin{cases} x_{new} & if \quad f(x_{new}) < f(x_i^t) \\ x_i^t & otherwise \end{cases} \tag{12}$$

2.3.5. Parametric adaptations

The selection and configuration of parameters can greatly influence the performance and efficacy of an optimization algorithm. Choosing suitable parameter values typically requires experimentation, domain-specific expertise, and iterative adjustments to achieve optimal outcomes for a specific optimization problem. In RGN, there are three parameters namely λ, pa, and R, and adaptation in each of these is required to make the algorithm self-adaptive. We use a combination of three $mo's$ for each parameter to see which mo fits the best for that particular parameter. The details of each of the $mo's$ used is given as

sa mo: was proposed by [45], to regulate the speed of an algorithm, and was employed to address the block fitting problem in urban planning. The equation of $sa\ mo$ is given by

$$sa = \zeta_{min} + (\zeta_{max} - \zeta_{min}) \times p^{(k-1)} \tag{13}$$

where ζ_{min}, ζ_{max} and $k \in [0, 1]$ and value of $p = 0.95$.

chaos mo: This mo assists in identifying solutions that are significantly different from the nearby optimal solutions, thereby enhancing the exploration process for an algorithm [46]. Additionally, this mo improves the speed at which convergence is achieved, and is expressed mathematically as follows:

$$K = 4 \times k \times (1 - k) \tag{14}$$

$$chaos = (\zeta_{max} - \zeta_{min}) \times \frac{t_{max} - t}{t_{max}} + \zeta_{min} \times K \tag{15}$$

for t iteration, $k \in [0, 1]$, $\zeta_{max} = 0.9$, $\zeta_{min} = 0.5$, t_{max} is the maximum iterations.

exp mo: A decreasing exponential function, referred to as $exp\ mo$ in the study [47], has been developed to facilitate a gradual shift from exploration to exploitation. This particular function is advantageous because it allows for divergence during exploration and convergence during exploitation. As a result, it has been observed to be effective for a wide range of optimization problems in continuous domains. The general equation for this function is as follows:

$$exp = \beta_{min} + (\zeta_{max} - \zeta_{min}) exp\left[-\frac{t}{(\frac{t_{max}}{10})}\right] \tag{16}$$

where ζ_{min} and $\zeta_{max} \in [0, 1]$.

osc mo: By employing $osc\ mo$, the algorithm achieves a harmonious combination of global exploration and local exploitation. This approach enables the particles to possess increased velocity and exploration capacity during the initial stages of searching, when it is crucial to explore the entire search space. As the search progresses, the mo is gradually decreased to prioritize exploitation and refinement within the promising regions of the search space. The mathematical equation representing this scenario is as follows:

$$\zeta = \frac{\zeta_{min} + \zeta_{max}}{2} + \frac{\zeta_{max} - \zeta_{min}}{2} cos(\frac{2\pi\, t}{T}) \tag{17}$$
$$T = \frac{2 \times t_{max}}{3 + 2k} \tag{18}$$

where, $\zeta_{max} = 0.9$, $\zeta_{min} = 0.3$ and $k \in [0, 1]$.

log mo: aids the algorithm in reducing parameter values, leading to enhanced performance over subsequent iterations. This facilitates more effective exploration during the initial phase and exploitation towards the later stages [48]. This parameter, plays a crucial role in balancing the algorithm and is expressed mathematically as follows:

$$\zeta = \zeta_{max} + (\zeta_{min} - \zeta_{max}) \times \log_{10}(k + \frac{10t}{t_{max}}) \tag{19}$$

where, $\zeta_{max} = 0.9, \zeta_{min} = 0.25$, $k = 1$.

sig mo: A common strategic approach is $sig\ mo$ function, which does not randomize the process in every iteration [47]. The initial value of mo is set to a high value to facilitate global search, but it progressively decreases throughout the iterations to enhance exploitation. By employing $sig\ mo$, minimal changes in weights are introduced, and a well-balanced operation can be achieved. The mathematical equation representing this approach is as follows:

$$\zeta = \frac{\zeta_{min} - \zeta_{max}}{1 + e^{-u \times (iter - h \times gen)}} + \zeta_{max} \tag{20}$$
$$u = 10^{\log(gen) - 2} \tag{21}$$

where, $\zeta_{max} = 0.9, \zeta_{min} = 0.5$, $gen = 51$, and $h, k \in [0, 1]$.

Thus, we are using six $mo's$ to optimize three parameters. We are using these $mo's$ based on the requirement of the operator in the algorithm and their significant influence. For λ, we have used sa, exp and $sig\ mo$; for R we have used $chaos$, osc, and log; and finally in case of pa we are using exp, osc, and $log\ mo$. Five test functions from CEC 2005 are used to see which mo is the best for each of the parameters involved. Thus, aiming to achieve a self-adaptive RGN algorithm. The details on the use of benchmarks is presented in the sensitivity analysis subsection. The pseudocode of the proposed RGN algorithm is given in Algorithm 1.

3. Results on benchmark datasets and image segmentation problem

In this section, details on the performance of the proposed algorithm in comparison to the numerical benchmarks is presented. For

Algorithm 1 Pseudocode of RGN algorithm.

```
Begin
Define: size of population (N); switching probability (pa);
        stopping criteria; dimension (Dim)
if i = 1: t_max/2 then
    global search using Eqn. (3)
    local search using Eqn. (7)
    evaluate fitness using Eqn. (12)
    update pa using exp, osc, and log mo
    update R using chaos, osc, and log mo
    update λ using sa, exp and sig mo
else
    global search using Eqn. (8), (9), and (10)
    local search using Eqn. (11)
    evaluate fitness using Eqn. (12)
    update pa using exp, osc, and log mo
    update R using chaos, osc, and log mo
    update λ using sa, exp and sig mo
    close;
update final best
End
```

evaluation, we use three benchmark datasets namely CEC 2005 [15], CEC 2014 [17], and CEC 2022 [18]. The definitions of these problems can be had from their respective papers, and not many details regarding the same have been added. Throughout this section, the results are presented as mean and standard deviation values of 51 runs, $D = 30$, population size of 50 and maximum iterations of 500 for CEC2005, and $10^4 \times D$ for CEC 2014 and CEC 2022 benchmark problems. The simulations are performed using MacBook Pro M2 processor, with MATLAB R2022a.

For comparison on CEC 2005, the algorithms used are a success history based DE (SHADE) [49], sine cosine crow search algorithm (SCCSA) [50], JADE [20], fractional-order calculus-based FPA (FA-FPO) [51], SaDE [52], FROBL-GJO [53], Evolution strategy based on covariance adaptation (CMA-ES) [16], extended GWO (GWO-E) [8], and LSHADE-SPACMA [16]. On the other hand for CEC 2014 benchmark problems, laplacian BBO (LX-BBO) [26], variable neighbourhood BA (VNBA) [25], population-based incremental learning (PBIL) [25], blended biogeography-based optimization (B-BBO) [26], random walk GWO (RW-GWO) [26], improved symbiotic organisms search (ISOS) [23], chaotic cuckoo search (CCS) [25], and improved elephant herding optimization (IMEHO) [25] are used. For CEC 2022, the algorithms used are DE [54], non-linear population size reduction SHADE with linear bias Change (NL-SHADE-LBC) [55], PSO [54], successful LSHADE with dynamic perturbation (S-LSHADE-DP) [56], EAs with eigen crossover (EA4eig) [57] and non-linear population size reduction SHADE with rank-based selective pressure midpoint of the population (NL-SHADE-RSP-MID) [58]. The parameter settings for the algorithms under comparison are given in Table 1.

In order to test the performance of RGN statistically, we have used wilcoxon signed-rank test (p-test) and Freidman rank test (f-test) [59]. The p-test, and f-test, are non-parametric statistical tests used to compare two related or paired samples. For present analysis, we perform these tests at 5% level of significance. The p-test is given as $win(w)/loss(l)/tie(t)$, where $win(w)$: " + " means proposed algorithm is better than the test; $loss(l)$: " − " means test is better than the proposed; and $tie(t)$: " = " means t, that is either both the algorithms are statistically similar or have no relevance between them. The f-rank is calculated by using the methods discussed in [59], and we have followed a ranking pattern to depict the results. Here, the best algorithm scores the first rank and the worst algorithm is given the last rank. One thing to keep in mind is that, if two algorithms have similar performance, both of them gets the first rank, and we have given the subsequent rank to the algorithms according to its numerical sequence.

Table 1
Parametric details of algorithms.

Algorithm	Parameters
CCS [25]	$(p_a) = 0.25$
JADE [20]	$F = 0.5$; $CR = 0.9$; $1/c = [5, 20]$; $p = [0.05, 0.20]$
SaDE [20]	F, CR = self adaptive
FA-FPO [51]	$\alpha = [0.1, 1]$, $S = adaptive$
ISOS [23]	$q \in [1, 100]\%$, $r \in [0, 1]$
FO-FPA [51]	$\alpha = [0.1, 1]$; $S = adaptive$; $r = 2$ or 4 or 8
SCCSA [50]	$e_1, e_2, e_3 = [0, 1]$
VNBA [25]	$(A) = 0.5$; $(r) = 0.5$
GWO-E [8]	$\vec{\alpha}$ = Linearly decreasing from 2 to 0
PBIL [25]	$(LR) = 0.1$; $(p_m) = 0.02$
B-BBO [26]	$H = 1$; $I = 1$
LX-BBO [26]	$H = 1$; $I = 1$
IMEHO [25]	(w) = Linearly decreasing [0.9,0.2]; $\alpha \in [0,1]$; $(p_c) = 0.05$
CMA-ES [16]	$n = \mu = 10$; $\lambda = 40$
SHADE [16]	$P_{best} = 0.1$, $ARC\ rate = 2$
RW-GWO [26]	$\vec{\alpha}$ = Linearly decreased from 2 to 0
LSHADE-SPACMA [16]	$c = 0.8$, $P_{best} = 0.11$, $ARC\ rate = 1.4$, $FCP = 0.5$
Proposed RGN	all parameters (pa, R, and λ) are adaptive

3.1. Sensitivity analysis

Three $mo's$ for each of the four parameter (I, R and λ) are added in the proposed RGN algorithm to make it self-adaptive in nature. Here, parametric evaluation based on five benchmark problems from the CEC 2005 is done. For parameter λ, we have used sa, exp and $sig\ mo$; for R we have used $chaos$, osc, and log; and finally in case of pa we are using exp, osc, and $log\ mo$. We have used two UMs (H_1, H_2), and three MMs (H_6, H_7 & H_{11}) for parametric analysis and the best among these $mo's$ is reported.

An analysis on five benchmark problems shows that; for λ, the exp mo gives the best results; for R, $chaos$ is the best; and for pa, $log\ mo$ is the overall best. Thus for the rest of the article we are using these values for the three parameters of the algorithm (Table 3).

3.2. Component wise analysis

In this section, a component wise analysis of the proposed algorithm is performed. This analysis is meant for dividing the algorithm in different set of components based on the number of algorithms used. As we have used GWO, RPO, CS and NMRA for formulating the proposed algorithm, a combination of either one or at least two algorithms is used for the comparison. So, a comparison is performed with respect to GWO, CS, NMRA, CS-GWO, RPO-CS, NMRA-CS, and finally RGN is performed to prove the significance of the proposed work. We are using five benchmarks for the classical CEC 2005 benchmark problems, and the analysis of results is presented in Table 4.

From the results in the Table 4, we can clearly see that for some of the functions, the CS-GWO, NMRA-CS and RGN are giving highly competitive results, and among the basic CS, GWO, and NMRA the results, GWO gives efficient performance. For function H_2 and H_7, all the hybrid variants have similar performance. For other problems, RGN algorithm gives much better results. Competitiveness among results can also be seen for function H_6 and $H - 10$. Thus, overall we can say that the proposed RGN is a highly competitive algorithm and is superior with respect to its individual components.

3.3. CEC 2005 benchmarks

In order to perform a comparison of the proposed RGN algorithm for CEC 2005 benchmark functions, we have used GWO-E [8], SaDE [20], CMA-ES [16], SHADE [16], EO [16], and LSHADE-SPACMA [16]. We are using the basic benchmarks from [2], and they consist of five UMs (H_1 to H_5), four MMs (H_6 to H_9) and two fixed dimensional functions (H_{10} and H_{11}), for performance evaluation as shown in Table 2. The dimension size used for UMs and MMs is 30, a total of 51 runs, population

Table 2
Benchmark functions (CEC 2005).

Function	Shift position	Range	Dim	f_{min}
Unimodal				
$H_1(c) = \sum_{i=1}^{n} c_i^2$	$[-30, -30, .., -30]$	$[-100, 100]$	30	0
$H_2(c) = \sum_{i=1}^{n} \lvert c_i \rvert + \Pi_{i=1}^{n} \lvert c_i \rvert$	$[-3, -3, .., -3]$	$[-10, 10]$	30	0
$H_3(c) = max_i\{\lvert c_i \rvert, 1 \le i \le n\}$	$[-3, -3, .., -3]$	$[-100, 100]$	30	0
$H_4(c) = \sum_{i=1}^{n} ([c_i + 0.5])^2$	$[-3, -3, .., -3]$	$[-10, 10]$	30	0
$H_5(c) = \sum_{i=1}^{n} ic_i^4 = rand[0, 1]$	$[-3, -3, .., -3]$	$[-1.28, 1.28]$	30	0
Multimodal				
$H_6(c) = -20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} c_i^2}) - exp(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi c_i)) + 20 + e$	$[-30, -30, .., -30]$	$[-100, 100]$	30	0
$H_7(c) = \frac{1}{4000}\sum_{i=1}^{N} c_i^2 - \Pi_{i=1}^{N} cos(\frac{c_i}{\sqrt{i}}) + 1$	$[-30, -30, .., -30]$	$[-600, 600]$	30	0
$H_8(c) = \frac{\pi}{n}10sin(\pi y_1) + \sum_{i=1}^{n} -1(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})]$ $(y_n - 1)^2 + \sum_{i=1}^{u}(c_i, 10, 100, 4) y_i = 1 = \frac{c_i+1}{4}$	$[-30, -30, .., -30]$	$[-50, 50]$	30	0
$H_9(c) = 0.1(\sin^2(3\pi c_1) + \sum_{i=1}^{n}(c_i - 1)^2(1 + \sin^2(3\pi c_i + 1)))$ $+0.1((c_n - 1)^2[1 + \sin^2(2\pi c_n]) + \sum_{i=1}^{n} u(c_1, 5, 100, 4)$	$[-30, -30, .., -30]$	$[-50, 50]$	30	0
Fixed dimension				
$H_{10}(c) = [1 + (c_1 + c_2 + 1)^2(19 - 14c_1 + 3c_1^2 - 14c_2 + 6c_1k_2 + 3c_2^2)] *$ $[30 + (2c_1 - 3c_2)^2 * (18 - 32c_1 + 12c_1^2 + 48c_2 - 36c_1c_2 + 27c_2^2)]$		$[-2, 2]$	2	3
$H_{11}(c) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(c_j - p_{ij})^2)$		$[1, 3]$	3	−3.86

Table 3
Parameter analysis of RGN.

Parameters	Variable		Functions					Statistical analysis		Best Strategy
			H_3	H_5	H_8	H_9	H_{10}	Average f-rank value	Overall f-rank	
λ	λ_{sa}	mean	**5.3547E-42**	**1.034E-03**	2.3532E-02	6.1342E-04	**3.0000E+00**			λ_{exp}
		std	**3.2389E-28**	**8.1346E-04**	4.5624E-03	1.4903E-03	**1.7423E-15**			
		f-rank	1	1	3	3	1	1.8	2	
	λ_{exp}	mean	5.8743E-38	1.4378E-03	**1.1187E-02**	**1.7247E-04**	3.0000E+00			
		std	2.6233E-37	1.5435E-03	**4.8345E-03**	**1.4433E-04**	2.1324E-15			
		f-rank	2	2	1	1	2	1.6	1	
	λ_{sig}	mean	1.5432E-38	1.5233E-03	1.4536E-02	2.2343E-04	2.9988E+00			
		std	1.5435E-37	**1.8454E-03**	5.7864E-03	1.4346E-04	6.3643E-15			
		f-rank	3	3	2	2	3	2.6	3	
R	F_{chaos}	mean	**6.2345E-39**	**1.4633E-03**	2.4623E-02	**1.4353E-04**	2.9998E+00			F_{chaos}
		std	**2.4352E-38**	**2.3452E-03**	1.3542E-01	**9.4553E-05**	2.3523E-15			
		f-rank	1	1	2	1	2	1.4	1	
	F_{osc}	mean	4.3253E-38	1.4352E-03	**1.3252E-02**	1.2534E-03	**3.0000E+00**			
		std	2.3453E-37	1.5534E-03	**1.4523E-02**	2.3252E-03	**1.4534E-15**			
		f-rank	2	3	1	3	1	2	2	
	F_{log}	mean	1.3345E-37	1.37341E-03	2.9385E-02	2.4534E-04	2.9999E+00			
		std	6.4532E-37	2.1564E-03	2.3451E-02	2.1451E-04	7.1345E-15			
		f-rank	3	2	3	2	3	2.6	3	
pa	pa_{exp}	mean	4.8001E-38	1.5123E-03	**9.3532E-03**	1.2535E-04	2.9999E+00			pa_{log}
		std	3.2351E-37	1.3512E-03	**4.1243E-03**	1.5122E-04	3.2341E-15			
		f-rank	2	2	1	2	2	1.8	2	
	pa_{osc}	mean	4.3245E-36	**1.2352E-03**	1.5234E-02	**1.2544E-04**	2.9999E+00			
		std	2.3452E-35	1.2523E-03	2.3358E-02	1.9823E-04	8.2817E-15			
		f-rank	3	1	2	3	3	2.4	3	
	pa_{log}	mean	**5.4523E-39**	1.4535E-03	1.4532E-02	**1.2353E-04**	**2.9999E+00**			
		std	**3.34234E-42**	1.3432E-03	2.4523E-02	**1.5432E-04**	**2.2143E-15**			
		f-rank	1	3	3	1	1	1.8	1	

size 50 and maximum number of iterations as 500, are taken for finding the best solutions. We are using mean and standard deviation values for analysing the results.

From the results in Table 5, it can be seen that for problems H_1, H_2, H_3, H_6, H_{10}, and $H_[11]$; RGN performs the best whereas for problems H_4, and H_8, CMA-ES is the best; for H_5, GWO-E is the best; for H_7, EO, GWO-E and RGN have similar performance; and for H_9, SaDE is the best. Thus overall in terms of $w/l/t$, SaDE gives 2/9/0 meaning SaDE is better than RGN for two problems, whereas RGN is better than SaDE for nine functions, and zero functions are statistically similar/or have no statistical relevance. For CMA-ES, we have 3/8/0; for LSHADE-SPACMA, we get 2/9/0; GWO-E is 2/8/1; SHADE is 3/8/2 and finally for EO we have 4/6/1. Apart from the ranksum tests, f-rank test predicts rank 1 for RGN. Overall, provimg the significance of the proposed RGN algorithm for this benchmark data set.

Table 4
Component wise comparison of the proposed algorithm with other algorithms.

Function		CS [60]	GWO [8]	NMRA [9]	CS-GWO	NMRA-CS	RGN
H_1	mean	5.79E+00	2.85E-35	1.25E-06	1.04E-67	3.48E-47	**2.64E-95**
	std	2.25E+00	4.21E-35	5.00E-06	1.04E-66	3.28E-46	**1.52E-94**
H_2	mean	1.69E-05	7.27E-21	1.02E+01	8.54E-57	3.43E-23	**2.70E-52**
	std	2.34E-04	7.55E-21	9.46E+00	2.13E-35	9.32E-20	**1.36E-51**
H_3	mean	4.14E-11	4.13E-09	2.10E-15	1.49E-18	3.78E-30	**4.98E-38**
	std	1.63E-10	9.99E-09	1.57E-13	1.12E-17	1.71E-27	**2.66E-37**
H_4	mean	1.69E+01	4.93E-09	9.8E-01	0.00E+00	1.00E-02	4.44E-44
	std	5.26E+00	4.55E-09	5.16E-05	0.00E+00	1.00E-02	2.76E-43
H_5	mean	3.21E-02	2.80E+01	1.04E-09	4.09E-03	1.99E-01	2.80E+01
	std	1.20E-02	7.474E-01	1.48E-10	4.09E-02	1.34E-01	4.01E-01
H_6	mean	7.22E+00	9.74E-04	4.50E-10	2.02E-15	2.14E-15	**1.00E-15**
	std	2.04E+00	4.13E-04	2.22E-09	1.66E-15	1.76E-15	**1.30E-15**
H_7	mean	1.59E-09	2.48E+00	6.06E-15	**0.00E+00**	**0.00E+00**	**0.00E+00**
	std	1.05E-08	4.04E+00	3.49E-14	0.00E+00	0.00E+00	0.00E+00
H_8	mean	5.60E+00	8.88E-02	1.03E-01	2.25E+00	2.92E-01	1.48E-02
	std	3.53E+00	3.06E-02	2.21E-01	3.28E+00	6.96E-01	5.80E-03
H_9	mean	6.84E+00	3.08E-01	4.00E-03	3.01E-01	1.28E-01	5.50E-03
	std	3.46E+00	1.51E-01	1.04E-02	1.19E-01	5.18E-02	481E-03
H_{10}	mean	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	**3.00E+00**
	std	2.92E-06	3.04E-11	7.74E-01	7.24E-12	4.85E-13	**1.64E-15**

3.4. CEC 2014 benchmarks

For comparison on CEC 2014 benchmark problems, the test problems are taken from [17], and the algorithms used for comparison are B-BBO [26], LX-BBO [26], RW-GWO [26], PBIL [25], LX-TLA [21], ISOS [23], VNBA [25], CCS [25], hybrid self-adaptive cuckoo search (HSA-CS) [62], and IMEHO [25]. The parameters of these algorithms are taken from Table 6. The results for this benchmark dataset are given in terms of mean error and standard deviation values. The mean error in the current scenario is calculated by defining the difference between the global minimum and the measured value [17]. If the error is less than 10^{-8}, the mean is considered as zero. We are taking a population size of 50, with a total number of function evaluations equation to $10000 \times D$, where D is the dimension size and is equal to 30 for the current scenario. The maximum number of runs is 30. The algorithms under comparison use a similar/same set of initial conditions, so that diversity in solutions can be maintained.

From the results in Table 6, we find that for function G_1, G_3, G_4, G_9, G_{12}, G_{17}, G_{20} to G_{25}, and G_{28} to G_{30}, RGN is the best; whereas G_2, G_6, G_{10}, G_{18}, and G_{19}, HSA-CS is found to provide the best results; for G_5, G_{26} and G_{27}, LX-BBO is the best algorithm; for G_{11}, ISOS gives the best results; for G_7, G_{14} and G_{15}, IMEHO provides the best set of results; for G_8, B-BBO is the best; and for G_9, and G_{16}, RW-GWO is the best found algorithm. Overall, we can say that our proposed algorithm is far better when compared to other algorithms in literature. This can be further proved from the non-parametric statistical rank-sum and f-rank test, which show that for most of the cases, RGN provides the best results. Also, f-rank shows that the proposed algorithm achieves the best rank and is therefore considered the best for the current case.

3.5. CEC 2022 benchmarks

The CEC 2022 benchmark problems are taken from [18]. We are using the mean and standard deviation values of 30 runs. A population size of 50, $D = 10$ and total number of function evaluations equal to $10,000 \times D$, is taken for performance evaluation. The algorithms used for comparison include PSO [54], YDSE [54], EDO [66], LSO [27], FHPSO [28], KOA [29], and CBOADP [30]. The parameters for these algorithms can be seen from their respective papers. In order to evaluate the performance of these algorithms, we are using mean and standard deviation values of 30 runs, with a population size of 100, and maximum number of function evaluations as suggested in [18].

From the results in Table 7, in function nt_1, we get same mean value for all the algorithms but YDSE, LSO, EDO, and FHPSO gives the best std values and score the first rank. For functions nt_2, nt_4, nt_7, nt_8, nt_9, nt_{10}, and nt_{12} RGN provides the best results. For function nt_3, LSO, YDSE, and FHPSO are the best performers; for nt_5 EDO and YDSE gives the most optimal solutions; for functions nt_6 and nt_{11}, EDO is the best. From the $w/l/t$ p-rank comparison, we can see that statistically, LSO is better than RGN for 3 functions; KOA and FHPSO for 2 functions respectively; EDO for 4 functions; CBOADP for 3 functions; and PSO for 1 benchmark problem. The results of f-rank further verifies that RGN algorithm is highly competitive with respect to other algorithms under comparison and scores the first rank. Overall, these results show that RGN algorithm is a significant addition in the field of optimization and in the next section we are performing multilevel image thresholding using RGN algorithm.

3.6. Computational complexity

The computational complexity of the proposed algorithm with respect to the basic algorithms used for the proposal of the RGN algorithm is presented in this section. We are using a generalized equation presented in [17] for complexity analysis as given by

$$\begin{aligned} &for \;\; i = 1 \;\; : \;\; k_t; \\ &xy_i = 0.55; \;\; xy_i = xy_i + xy_i; \;\; xy_i = xy_i/2; \;\; xy_i = xy_i \times xy_i; \\ &xy_i = sqrt(xy_i); \\ &xy_i = log(xy_i); \;\; xy_i = exp(xy_i); \;\; xy_i = xy_i/(xy_i + 2); \\ &end \end{aligned} \tag{22}$$

This equation is used as a basis for calculating the general time taken for $k_t = 1000000$; and is represented as T_0. For analysis, we are using the first function from CEC 2005 benchmarks. Here we have two parameters T_1 and T_2, and T_1 is the time taken to evaluate the algorithm for a single run, whereas T_2 is the average time taken to evaluate the algorithm for five runs. The complexity of given in Table 8, and it shows the competitive performance of the proposed RGN algorithm with respect to other algorithms under consideration.

Table 5
Simulation outcomes of RGN algorithm in contrast to the existing MH algorithms for benchmark (CEC 2005) functions.

Function		SaDE [52]	CMA-ES [61]	LSHADE-SPACMA [24]	GWO-E [8]	SHADE [49]	EO [16]	RGN
H_1	mean	4.50E-20	1.42E-18	2.23E-01	3.92E-67	1.42E-09	3.32E-40	**2.64E-95**
	std	6.90E-20	3.13E-18	1.48E-01	1.11E-66	3.09E-09	6.78E-40	**1.52E-94**
	p-rank	–	–	–	–	–	–	
	f-rank	4	5	7	2	6	3	1
H_2	mean	1.90E-14	2.98E-07	2.11E+01	4.31E-36	8.70E-03	7.12E-23	**2.70E-52**
	std	1.05E-14	1.78E+00	9.57E+00	6.57E-36	2.13E-02	6.36E-23	**1.36E-51**
	p-rank	–	–	–	–	–	–	
	f-rank	4	5	7	2	6	3	1
H_3	mean	7.40E-11	2.39E-25	2.15E-17	2.01E-06	9.79E-01	5.39E-10	**4.98E-38**
	std	1.82E-10	1.25E-06	4.92E-01	6.80E-25	7.99E-01	1.38E-09	**2.66E-37**
	p-rank	–	–	–	–	–	–	
	f-rank	4	2	3	6	7	5	1
H_4	mean	9.30E+02	6.83E-19	2.48E-01	2.65E+01	5.31E-10	8.29E-06	4.44E-44
	std	1.80E+02	6.71E-19	1.13E-01	5.19E-01	6.35E-10	5.02E-06	2.76E-43
	p-rank	–	+	–	–	+	+	
	f-rank	7	1	5	6	2	3	4
H_5	mean	4.80E-03	2.75E-02	4.70E-03	**9.90E-04**	2.35E-02	1.17E-03	2.80E+01
	std	1.20E-03	7.90E-03	1.90E-03	**8.37E-04**	8.80E-03	6.54E-04	4.01E-01
	p-rank	–	–	–	+	–	+	
	f-rank	5	7	4	1	6	2	3
H_6	mean	2.70E-03	1.55E+01	3.93E-02	5.58E-15	3.95E-01	8.34E-14	**1.00E-15**
	std	5.10E-04	7.92E+00	1.51E-02	1.67E-15	5.86E-01	6.90E-10	**1.30E-15**
	p-rank	–	–	–	–	–	–	
	f-rank	6	7	5	2	6	3	1
H_7	mean	7.80E-04	5.76E-15	8.94E-01	**0.00E+00**	4.80E-03	**0.00E+00**	0.00E+00
	std	1.20E-03	6.18E-15	1.07E-01	**0.00E+00**	7.70E-03	**0.00E+00**	0.00E+00
	p-rank	–	–	–	=	–	=	
	f-rank	5	4	7	1	6	1	1
H_8	mean	1.90E-05	2.87E-16	8.18E-04	1.98E-02	3.46E-02	7.97E-07	1.48E-02
	std	9.20E-06	5.64E-16	1.00E-03	1.01E-02	8.75E-02	7.69E-07	5.80E-03
	p-rank	+	+	+	+	+	+	
	f-rank	3	1	4	5	6	2	7
H_9	mean	6.10E-05	3.66E-04	1.02E-02	2.50E-01	7.32E-04 =	2.92E-02	5.50E-03E
	std	2.00E-05	2.00E-03	1.03E-02	1.63E-01	2.80E-03 =	3.52E-02	481E-03
	p-rank	+	+	+	–	+	+	
	f-rank	1	2	4	7	3	5	6
H_{10}	mean	3.00E+00	8.40E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	**3.00E+00**
	std	3.00E-15	2.05E+01	1.25E-15	1.25E-05	1.87E-15	1.56E-15	**1.64E-15**
	p-rank	–	–	–	–	–	–	
	f-rank	5	7	2	6	4	3	1
H_{11}	mean	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00
	std	3.10E-15	2.70E-15	2.70E-15	4.16E-06	2.69E-15	2.59E-15	**8.28E-16**
	p-rank		–	–	–	–	–	
	f-rank	5	7	2	6	4	3	1
w/l/t		2/9/0	3/8/0	2/9/0	2/8/1	3/8/2	4/6/1	NA
Overall f-rank value		49	48	45	38	56	33	27
Overall f-rank		6	5	4	3	7	2	1

3.7. Real world optimization: multilevel thresholding (MLT) using RGN with type II Fuzzy Sets

In this subsection, a novel approach for MLT of images is introduced, combining Type II Fuzzy sets with the RGN algorithm to determine the threshold values. Entropy-based algorithms are commonly employed to separate the image histogram. Initially, Shannon's entropy from information theory was applied to the thresholding problem [34]. Tao et al., to enhance the performance of MLT, proposed a fuzzy entropy-based technique [39]. This approach utilizes histogram partitions with predefined fuzzy membership values for MLT, enabling the extraction of objects present in the image.

Fuzzy Type II set

In Type-I fuzzy sets, a set A is expressed as $Y = \{y_1, y_2, y_3 ..., y_n\}$ in a finite set. It can be represented as:

$$A = \{y,\ \mu_A(y) | y \epsilon Y, 0 \le \mu_A(y) \le 1\} \tag{23}$$

Here, μ_A is set A's membership function. On the other hand, Type-II fuzzy sets use multiple membership values to handle increased uncertainty, expressed as:

$$A = \{y, \mu_A^{high}(y), \mu_A^{low}(y) | y \epsilon Y, 0 \le \mu_A^{high}(y), \mu_A^{low}(y) \le 1\} \tag{24}$$

In this case, μ_A^{low} and μ_A^{high} represent the lower and upper membership function respectively.

MLT using Type II fuzzy sets

Image thresholding is a commonly used method for segmenting an image. It involves selecting threshold values and applying them to the histogram of the image until a specific criterion is met.

Let's consider a grayscale image I with dimensions $U \times V$, where each pixel (u, v) has a gray value represented by $f(u, v)$ with $u \epsilon \{0, 1, 2, \ldots U\}$ and $v \epsilon \{0, 1, 2, \ldots V\}$. The set of all gray levels $\{0, 1, 2, \ldots, L-1\}$ is denoted as GL, where L is the number of gray levels in the image.

Table 6
Statistical results for CEC 2014 benchmark functions.

Functions	Error	LX-BBO [63]	B-BBO [63]	RW-GWO [26]	ISOS [23]	PBIL [25]	VNBA [64]	IMEHO [25]	CCS [65]	LX-TLA [21]	HSA-CS [62]	RGN
G_1	mean	1.011E+07	6.503E+06	8.023E+06	9.823E+05	3.423E+08	2.432E+08	2.374E+06	1.462E+08	8.273E+08	2.091E+04	**4.22E+02**
	std	1.012E+07	1.302E+06	3.311E+06	7.052E+05	1.094E+08	5.933E+07	4.323E+06	3.273E+07	1.412E+08	1.751E+04	**1.981E+02**
	f-rank	7	5	6	3	10	9	4	8	11	2	1
G_2	mean	5.343E+04	2.355E+04	2.235E+05	**5.273E+00**	4.083E+10	1.925E+10	5.494E+03	2.604E+09	5.481E+10	0.000E+00	1.000E+10
	std	2.144E+04	9.994E+03	5.514E+05	**1.722E+01**	3.394E+09	4.234E+09	4.875E+03	5.223E+08	9.172E+09	0.000E+00	0.000E+00
	f-rank	5	4	6	2	10	9	3	7	11	1	8
G_3	mean	1.632E+04	6.034E+03	3.163E+02	4.794E+02	9.193E+04	2.933E+04	1.415E+02	2.703E+03	9.201E+04	2.281E-03	**0.000E+00**
	std	1.703E+04	3.154E+03	4.344E+02	6.245E+02	1.754E+04	1.394E+04	1.585E+02	7.744E+04	2.631E+04	2.601E-03	**1.065E-11**
	f-rank	8	7	4	5	10	9	3	6	11	2	1
G_4	mean	9.995E+01	1.023E+02	3.415E+01	5.985E+01	3.435E+03	1.603E+03	1.245E+02	3.223E+02	5.98E+03	1.340E+00	**2.073E-01**
	std	2.846E+01	3.134E+01	1.804E+01	3.574E+01	7.566E+02	3.634E+02	4.774E+01	4.092E+01	8.64E+03	8.790E+00	8.054E+01
	f-rank	5	6	3	4	10	9	7	8	11	2	1
G_5	mean	**3.065E+00**	3.745E+00	2.056E+01	2.033E+01	2.104E+01	2.104E+01	2.104E+01	2.103E+01	1.251E+01	2.000E+01	2.070E+01
	std	7.864E-01	4.915E-01	7.465E-02	6.674E-02	5.565E-02	5.433E-02	5.995E-02	8.812E-02	8.521E-01	2.951E-03	6.592E-02
	f-rank	1	2	7	5	11	10	9	8	4	3	6
G_6	mean	1.703E+01	1.994E+01	9.844E+00	1.054E+01	3.802E+01	3.303E+01	1.203E+01	2.503E+01	3.991E+01	2.761E+00	2.124E+01
	std	3.122E+00	2.705E+00	3.493E+00	2.393E+00	1.163E+00	2.582E+00	2.724E+00	2.000E+00	4.752E+02	1.551E+00	2.220E+00
	f-rank	6	7	3	4	10	9	5	8	11	1	2
G_7	mean	1.755E-01	7.813E-02	2.533E-01	1.563E-02	3.402E+02	1.112E+02	**0.000E+00**	2.302E+01	4.752E+02	5.801E-04	0.000E+00
	std	8.564E-02	4.442E-02	1.434E-01	1.834E-02	2.743E+01	1.813E+01	1.194E-01	3.523E+00	5.741E+01	2.371E-03	1.945E-10
	f-rank	6	5	7	4	10	9	1	8	11	3	2
G_8	mean	5.533E+01	**4.713E-01**	4.383E+01	1.473E+01	3.000E+02	1.743E+02	3.303E+01	2.902E+02	3.591E+02	2.201E+00	2.879E+01
	std	3.782E+02	6.794E-01	8.482E+00	3.343E+00	1.034E+01	1.612E+01	9.194E+00	2.234E+01	3.881E+02	1.742E+00	4.575E+00
	f-rank	7	1	6	4	10	8	5	9	11	2	3
G_9	mean	7.666E+01	9.111E+01	6.334E+01	2.562E+02	3.705E+02	2.503E+02	3.204E+01	2.903E+02	3.971E+02	2.521E+01	1.571E+01
	std	1.612E+01	1.543E+01	1.303E+01	1.344E+01	1.694E+01	2.032E+01	1.154E+01	2.384E+01	2.000E+01	5.152E+00	2.276E+01
	f-rank	5	6	3	8	10	7	4	9	11	2	1
G_{10}	mean	1.252E+04	6.684E+03	9.611E+02	1.783E+03	6.262E+03	3.501E+03	2.262E+03	8.553E+03	7.231E+03	1.271E+01	7.540E+02
	std	1.163E+02	4.582E+02	2.722E+02	4.094E+01	3.052E+02	3.473E+02	5.721E+02	4.914E+02	3.241E+02	1.761E+01	1.545E+02
	f-rank	10	7	2	3	6	5	4	9	8	1	2
G_{11}	mean	1.234E+04	6.713E+03	2.683E+03	**1.483E+03**	7.103E+03	6.803E+03	2.863E+03	8.834E+03	8.471E+03	1.472E+03	2.973E+03
	std	3.412E+02	5.174E+02	3.682E+02	4.544E+02	2.973E+02	3.792E+02	5.382E+02	5.502E+02	2.191E+02	2.771E+02	2.252E+02
	f-rank	11	6	4	1	8	7	5	10	9	2	3
G_{12}	mean	1.111E-02	1.111E-02	5.454E-01	3.554E-01	1.000E+01	1.000E+01	1.002E+01	1.000E+01	1.111E-02	1.700E-01	**6.023E-03**
	std	1.753E-18	1.754E-18	1.665E-01	5.733E-02	3.383E-01	3.512E-01	5.263E-01	1.092E+00	1.83E-18	5.291E-02	1.084E-01
	f-rank	2	2	4	3	8	7	9	6	2	5	1
G_{13}	mean	6.553E-01	6.782E-01	2.804E-01	3.774E-01	0.000E+00	0.000E+00	0.000E+00	**0.000E+00**	5.661E+00	2.331E-01	3.021E-02
	std	1.564E-01	7.983E-02	6.305E-02	7.102E-02	2.562E-01	3.643E-01	6.252E-02	1.762E-01	3.271E-01	3.321E-02	3.851E-02
	f-rank	9	10	7	8	2	3	4	1	11	6	5
G_{14}	mean	6.202E-01	3.932E-01	4.234E-01	2.712E-01	1.000E+02	6.000E+01	**0.000E+00**	1.000E+01	1.401E+02	2.541E-01	2.005E-02
	std	2.963E-01	1.551E-01	2.153E-01	5.121E-02	1.163E+01	1.223E+01	9.852E-02	1.887E+00	1.651E+01	7.261E-02	2.571E-02
	f-rank	7	5	6	4	10	9	1	8	11	3	2
G_{15}	mean	1.554E+01	1.884E+01	8.814E+00	1.061E+01	6.844E+05	2.394E+03	**0.000E+00**	8.000E+01	1.93E+05	9.281E+00	7.754E+00
	std	5.493E+00	5.645E+00	1.512E+00	3.713E+00	2.855E+05	1.223E+03	1.353E+00	3.053E+01	4.102E+05	4.831E-01	1.408E+00
	f-rank	6	7	4	5	10	9	1	8	11	3	2
w/l/t		**1/14/0**	**0/15/0**	**1/14/0**	**2/13/0**	**0/15/0**	**0/15/0**	**2/13/0**	**1/14/0**	**0/15/0**	**8/9/0**	
f-rank		95	80	69	64	135	119	65	105	145	38	40

$$f(u,v)\ \epsilon\ GL\ for\ all\ (u,v)\ \epsilon\ I$$
$$P_k = \{(u,v) : f(u,v) = a,\ where\ a\ \epsilon\ 0,1,2,\ldots L-1\} \tag{25}$$

The pixels can be grouped based on their gray values, and we define P_a as the set of pixels with gray value a, where a ranges from 0 to $L-1$.

To measure the uniformity of the histogram, we define $S = \{s_1, s_2, s_3, \ldots, s_{L-1}\}$ as the uniform histogram of image I, where each $s_a = n_a/(U \times V)$ value represents the proportion of pixels with gray value a.

$$s_a = \frac{(L-1)_a}{(L-1)}, \sum_{a=1}^{L-1} s_a = 1 \tag{26}$$

To incorporate the concept of fuzzy sets, we introduce the measure of ultra-fuzziness. It quantifies the ambiguity in defining membership values. A value of 0 indicates unambiguous membership values, while a value of 1 indicates a range of membership values. The ultra-fuzziness for the k^{th} level is calculated using

$$O_k = \sum_{a=0}^{L-1} \left(s_a * \left(\mu_k^{high}(a) - \mu_k^{low}(a) \right) \right) \tag{27}$$

where μ_k^{high} and μ_k^{low} are the upper and lower membership functions, respectively, and $k\ \epsilon\ \{1,2,3,\ldots,PR+1\}$, PR is the segmentation level count.

The trapezoidal fuzzy membership function μ_k is defined in Equation (28), with parameters b_k and d_k determining the range of gray values for each level.

Table 6 (*continued*)

Functions	Error	LX-BBO [63]	B-BBO [63]	RW-GWO [26]	ISOS [23]	PBIL [25]	VNBA [64]	IMEHO [25]	CCS [65]	LX-TLA [21]	HSA-CS [62]	RGN
G_{16}	mean	1.082E+01	1.064E+01	1.033E+01	9.214E+01	2.000E+01	2.000E+01	2.000E+01	2.000E+01	1.321E+01	9.280E+00	1.201E+01
	std	5.843E-01	6.252E-01	6.111E-01	7.313E-01	2.123E-01	3.666E-01	7.643E-01	1.754E-01	2.57E-01	4.831E-01	3.261E-01
	f-rank	3	2	1	11	8	9	10	7	6	4	5
G_{17}	mean	1.493E+06	1.274E+06	5.714E+05	1.754E+05	9.743E+06	2.534E+06	7.694E+04	1.152E+07	1.481E+05	1.651E+03	9.991E+02
	std	9.342E+05	5.463E+05	4.103E+05	1.643E+05	2.792E+06	3.343E+06	8.383E+04	4.594E+06	4.681E+04	4.111E+02	2.123E+02
	f-rank	8	7	6	5	10	9	3	11	4	2	1
G_{18}	mean	2.893E+03	8.224E+02	6.524E+03	3.892E+03	6.164E+08	1.665E+08	3.303E+03	1.103E+08	4.511E+02	1.281E+02	3.851E+02
	std	4.272E+03	1.003E+03	4.633E+02	5.153E+03	1.683E+08	1.034E+08	3.522E+03	4.664E+07	3.3381E+02	4.291E+01	1.120E+01
	f-rank	5	4	8	7	11	10	6	9	3	1	2
G_{19}	mean	5.193E+03	7.814E+03	1.144E+01	7.792E+01	1.903E+02	1.204E+02	1.054E+01	4.000E+01	1.401E+02	5.811E+00	**8.012E+00**
	std	5.672E+03	4.673E+03	2.033E+00	1.784E+00	3.422E+01	3.823E+01	1.742E+00	5.931E+00	9.941E+01	1.511E+00	6.546E-01
	f-rank	10	11	4	6	8	7	3	5	9	1	2
G_{20}	mean	2.614E+04	1.624E+04	6.273E+02	4.983E+03	3.594E+04	1.694E+04	2.102E+02	1.033E+06	1.851E+04	4.351E+01	**3.041E+01**
	std	1.575E+04	4.112E+03	1.124E+03	3.405E+03	1.773E+04	6.573E+03	8.174E+01	9.052E+05	7.231E+03	3.581E+01	6.9988E+00
	f-rank	9	6	4	5	10	7	3	11	8	2	1
G_{21}	mean	1.113E+06	1.22E+06	2.582E+05	8.904E+04	2.523E+06	2.305E+06	2.724E+04	5.665E+06	3.333E+04	6.491E+02	**5.351E+02**
	std	7.954E+05	7.963E+05	1.763E+05	1.074E+05	1.192E+06	1.354E+06	1.834E+04	2.734E+06	1.481E+03	2.371E+02	1.694E+02
	f-rank	7	8	6	5	10	9	2	11	4	3	1
G_{22}	mean	1.885E+03	1.684E+02	2.085E+02	2.753E+02	1.023E+03	8.403E+02	2.104E+02	1.344E+03	6.162E+02	**1.331E+02**	1.775E+02
	std	2.034E+02	2.473E+02	2.084E+02	1.452E+02	1.884E+02	1.282E+02	1.015E+02	1.885E+02	3.821E+02	8.552E+01	8.310E+01
	f-rank	11	3	4	6	9	8	5	10	7	1	2
G_{23}	mean	4.112E+02	3.434E+02	3.152E+02	3.154E+02	6.000E+02	3.904E+02	3.204E+02	3.504E+02	3.451E+02	3.155E+02	**2.000E+02**
	std	6.433E+01	2.843E+01	2.773E-01	1.603E+01	6.703E+01	2.475E+01	4.783E-01	7.662E+00	2.101E+01	0.000E+00	5.742E+01
	f-rank	10	6	2	3	11	9	5	8	7	4	1
G_{24}	mean	1.473E+04	3.413E+04	2.000E+02	2.000E+02	4.000E+02	2.302E+02	2.400E+02	2.204E+02	6.341E+03	2.251E+02	**2.000E+02**
	std	8.374E+03	2.352E+04	3.043E-03	1.504E-03	1.423E+01	2.533E+01	6.463E+00	2.513E+00	4.301E+03	1.681E+00	1.476E-07
	f-rank	10	11	2	3	7	5	6	4	9	8	1
G_{25}	mean	5.292E+02	6.534E+02	2.044E+02	2.000E+02	2.403E+02	2.103E+02	2.103E+02	2.203E+02	4.651E+02	2.004E+02	**2.000E+02**
	std	4.363E+01	6.011E+01	1.183E+00	8.073E-01	6.001E+00	1.074E+01	2.082E+00	4.443E+00	2.621E+01	1.040E+00	1.337E+00
	f-rank	10	11	4	2	8	6	5	7	9	3	1
G_{26}	mean	**2.122E+00**	3.645E+01	1.000E+02	1.000E+02	1.000E+02	1.000E+02	1.000E+02	1.000E+02	5.561E+00	1.000E+02	1.000E+02
	std	3.464E+00	5.622E+01	7.364E-02	9.553E-02	2.095E-01	4.303E-01	6.000E-02	2.062E-01	8.291E+00	4.590E-02	4.484E-02
	f-rank	1	3	9	8	6	7	4	5	2	11	10
G_{27}	mean	**1.953E+02**	3.042E+02	4.093E+02	5.433E+02	1.082E+03	1.294E+03	5.802E+02	5.303E+02	2.441E+02	3.812E+02	3.093E+02
	std	1.044E+02	1.603E+02	6.091E+00	1.364E+02	2.801E+02	3.303E+01	1.413E+02	7.922E+01	9.331E+01	4.152E+01	3.658E+01
	f-rank	1	4	6	8	9	10	11	7	2	5	3
G_{28}	mean	1.942E+03	2.124E+03	4.344E+02	9.682E+02	1.392E+03	1.683E+03	9.703E+02	1.443E+03	1.462E+03	8.271E+02	7.460E+02
	std	1.043E+02	4.445E+02	8.453E+00	4.123E+01	1.331E+02	2.334E+02	2.442E+02	9.464E+02	3.691E+02		3.074E+02
	f-rank	10	11	3	4	6	9	5	7	8	2	1
G_{29}	mean	1.983E+07	3.093E+07	2.141E+02	5.704E+05	5.702E+06	7.473E+06	1.212E+03	1.202E+06	1.761E+07	9.141E+02	7.183E+02
	std	3.952E+06	6.912E+06	2.373E+00	2.143E+06	3.332E+06	1.202E+06	2.163E+02	7.033E+05	4.081E+06	1.531E+02	1.248E+01
	f-rank	10	11	1	5	8	7	4	6	9	3	2
G_{30}	mean	6.952E+06	1.384E+07	6.694E+02	2.382E+05	1.494E+05	1.892E+05	4.084E+03	7.673E+04	2.331E+06	1.951E+03	1.260E+03
	std	1.033E+07	1.083E+07	2.143E+02	1.103E+03	5.553E+04	1.033E+05	1.443E+03	3.374E+04	1.361E+06	9.951E+02	1.380E+02
	f-rank	10	11	3	8	6	7	4	5	9	2	1
w/l/t		**2/13/0**	**0/15/0**	**1/14/0**	**0/15/0**	**0/15/0**	**0/15/0**	**0/15/0**	**0/15/0**	**0/15/0**	**04/11/0**	
Overall w/l/t		**3/27/0**	**1/29/0**	**2/28/0**	**2/28/0**	**0/30/0**	**0/30/0**	**3/27/0**	**1/29/0**	**0/30/0**	**12/17/0**	
f-rank		115	105	63	86	127	120	76	113	98	52	34
Overall f-rank		210	185	159	150	262	239	141	218	243	90	74
Average f-rank		7.0	6.11	5.3	5.00	8.73	7.1	4.7	7.26	8.1	3.0	2.46
f-rank		7	6	4	5	11	10	3	8	9	2	1

$$\mu_k(a)=\begin{cases}0, & a\le b_{k-1}\\ \frac{a-b_{k-1}}{d_{k-1}-b_{k-1}}, & b_{k-1}<a\le d_{k-1}\\ 1, & d_{k-1}<a\le b_k\\ \frac{a-d_k}{b_k-d_k}, & b_k<a\le d_k\\ 0, & a>d_k\end{cases} \tag{28}$$

Note that $b_0=d_k=0$ and $b_{PR+1}=d_{PR+1}=L-1$. For the kth segmentation level, the fuzzy type-II entropy, is computed using

$$H_k=-\sum_{a=0}^{L-1}\left(\frac{\left((s_a*\left(\mu_k^{high}(a)-\mu_k^{low}(a)\right)\right)}{O_k}\right) *ln\left(\frac{\left(s_a*\left(\mu_k^{high}(a)-\mu_k^{low}(a)\right)\right)}{O_k}\right) \tag{29}$$

Table 7
Statistical results for CEC 2022 numerical test problems.

Problem		RGN	PSO [54]	YDSE [54]	LSO [27]	EDO [66]	FHPSO [28]	KOA [29]	CBOADP [30]
nt_1	Mean	**3.000E+02**	**3.000E+02**	**3.000E+02**	**3.000E+02**	**3.000E+02**	**3.000E+02**	**3.00E+02**	**3.000E+02**
	Std	4.224E-05	2.585E-16	**0.000E+00**	**0.000E+00**	0.000E+00	2.1414E-05	1.06E-14	3.42E-01
	p-rank		+	+	+	+	+	+	−
nt_2	Mean	**4.001E+02**	4.235E+02	4.002E+02	4.009E+02	4.022E+02	4.0265E+02	4.02E+02	4.00E+02
	Std	4.2289E-01	3.033E+01	3.0452E-01	2.13E+00	2.010E+00	3.5548E+00	3.39E+00	9.31E-04
	p-rank		−	−	−	−	−	−	−
nt_3	Mean	**6.000E+02**	6.003E+02	**6.000E+02**	**6.000E+02**	**6.000E+02**	6.009E+02	**6.00E+02**	**6.000E+02**
	Std	4.2387E-03	6.140E-01	7.4947E-03	**0.000E+00**	4.47E-01	1.0405E+00	6.33E-14	9.25E-06
	p-rank		−	+	+	−	+	+	+
nt_4	Mean	**8.004E+02**	8.114E+02	8.0404E+02	8.107E+02	8.065E+02	8.104E+02	8.044E+02	8.08E+02
	Std	**1.126E+00**	5.123E+00	2.956E+00	2.96E+00	3.30E+00	3.7841E+00	1.62E+00	1.71E+00
	p-rank		−	−	−	−	−	−	−
nt_5	Mean	**9.000E+02**	9.001E+02	**9.0000E+02**	**9.000E+02**	**9.000E+02**	**9.002E+02**	**9.000E+02**	**9.08E+02**
	Std	5.439E-05	1.181E-01	**0.000E+00**	8.29E-02	**0.000E+00**	3.8395E-02	2.00E-02	7.77E+00
	p-rank		−	+	−	+	−	−	−
nt_6	Mean	1.723E+03	5.308E+03	**1.800E+03**	**1.800E+03**	**1.800E+03**	1.8619E+03	1.80E+03	2.07E+03
	Std	7.135E+01	2.651E+03	1.1919E-01	3.400E-01	**1.130E-01**	4.0632E+01	1.80E+03	1.41E+02
	p-rank		−	+	+	+	−	−	
nt_7	Mean	**2.000E+03**	2.013E+03	2.0058E+03	**2.000E+03**	2.007E+03	2.0246E+03	**2.000E+03**	**2.000E+03**
	Std	**4.836E-03**	9.796E+00	5.7989E+00	5.400E-01	6.79E+00	**8.2778E+00**	1.100E-01	1.41E+02
	p-rank		−	−	−	−	−	−	−
nt_8	Mean	**2.200E+03**	2.227E+03	2.203E+03	2.201E+03	2.206E+03	2.2283E+03	2.22E+03	2.201E+03
	Std	**6.753E-02**	3.930E+01	3.7917E+00	3.83E+00	3.84E+00	2.0753E+00	6.20E+00	8.27E+00
	p-rank		−	+	−	−	−	−	−
nt_9	Mean	**2.523E+03**	2.534E+03	2.5250E+03	2.529E+03	2.529E+03	2.5294E+03	2.529E+03	2.53E+03
	Std	**3.746E+01**	1.202E+01	2.3604E+01	5.680E-14	0.00E+00	5.7454E-01	0.00E+00	9.58E-02
	p-rank		−	−	−	−	−	−	−
nt_{10}	Mean	**2.490E+03**	2.572E+03	2.4976E+03	2.500E+03	2.500E+03	2.5472E+03	2.500E+03	2.50E+03
	Std	**3.162E+01**	9.683E+01	1.4886E+01	6.000E-01	5.280E-02	5.4624E+01	4.000E-01	1.27E-01
	p-rank		−	−	−	−	−	−	−
nt_{11}	Mean	**2.600E+03**	2.763E+03	**2.6000E+03**	2.825E+03	**2.600E+03**	2.6395E+03	2.770E+03	**2.600E+03**
	Std	6.642E-03	1.178E+02	1.1942E-13	1.291E+02	**0.000E+00**	1.2783E+02	1.512E+02	2.07E-05
	p-rank		−	−	−	+	−	−	+
nt_{12}	Mean	**2.857E+03**	2.8589E+03	2.871E+03	2.862E+03	2.860E+03	2.8687E+03	2.864E+03	2.87E+03
	Std	**7.322E-04**	5.9374E-01	1.292E+01	1.72E+00	2.090E+00	3.3069E+00	1.16E+00	1.13E+00
	p-rank		−	−	−	−	−	−	−
w/l/t		NA	3/9/0	3/9/0	1/11/0	4/8/0	2/10/0	2/10/0	3/9/0
Average f-rank		2.5833	7.166	4.083	3.916	3.583	4.583	4.333	4.833
Overall f-rank		1	8	4	3	2	6	5	7

Table 8
Time complexity analysis.

	Function	Dimension	Algorithm	T_0	T_2	T_1	$(T_2 - T_1)/T_0$
CEC 2019	1	9	RGN	0.009406	0.397424	0.359343	5.36343
			GWO	0.009406	0.15125	0.126076	2.676377
			KOA	0.009406	0.308005	0.236723	7.57838
			INFO [67]	0.009406	0.360435	0.311656	5.185988
			NMRA [9]	0.009406	0.114592	0.068748	4.873953

where H_k represents the entropy for that level. The total entropy, H, is the sum of entropies for all levels, as given by

$$H(b_1, d_1, \ldots, b_n, d_n) = \sum_{k=1}^{PR+1} H_k \tag{30}$$

To maximize the fuzzy parameters, we aim to maximize the total entropy. The optimal fuzzy parameter set $(b_1, d_1, \ldots, b_n, d_n)^*$ maximizes H. Finally, the threshold value for the nth level is calculated using

$$Th_n = 0.5 * (b_n + d_n) \tag{31}$$

where Th_n is the threshold value and n ranges from 1 to PR, representing the threshold numbers for multilevel segmentation. This approach utilizes Type-II fuzzy sets combined with entropy-based algorithms to perform multilevel thresholding of images. The resulting threshold values can be used for accurate image segmentation.

Results and discussion

The MLT approach using RGN is evaluated for image segmentation. The performance of the proposed approach is checked with various algorithms such as PSO, DE, PFA, HPFPPA-D [1] and MHA. To assess its effectiveness, the experiments are conducted using ten benchmark image sets that exhibit different histogram distributions and complexities [68]. The evaluation is performed based on 30 independent runs to account for stochastic variations. The dimension size is defined as twice the total number of thresholds in the multilevel thresholding process.

Table 9
The optimum threshold values obtained using various algorithms for different threshold levels.

Im	PR	RGN	MHA	HPFPPA-D	PSO	PFA	DE
41004	3	35 139 232	26 99 228	33 66 137	34 144 235	35 104 193	35 100 185
	5	37 95 163 209 232	53 84 134 166 247	34 100 135 174 233	36 78 114 181 235	35 100 136 174 232	39 78 115 158 209
	7	39 77 115 152 182 211 233	18 62 98 124 157 202 237	41 84 115 146 178 210 233	20 46 71 91 135 178 217	43 78 112 143 176 211 234	42 86 101 133 155 187 234
176035	3	44 114 197	35 117 195	47 113 198	74 138 199	49 116 199	57 106 181
	5	47 93 140 187 221	36 73 114 142 211	51 93 139 184 220	36 73 136 187 222	54 96 141 187 221	51 93 137 185 222
	7	44 88 115 141 169 196 225	20 51 81 117 183 206 215	49 90 117 145 170 195 224	40 78 114 136 171 207 231	43 80 105 132 163 192 224	47 79 106 133 157 184 219
225017	3	64 137 201	14 116 236	68 134 195	68 136 196	70 138 197	67 138 199
	5	23 45 96 146 201	25 73 154 201 229	21 76 128 162 216	25 75 108 142 213	21 76 137 166 211	27 78 131 169 215
	7	22 44 78 111 145 183 221	18 37 90 115 127 184 218	22 44 78 112 147 182 219	20 84 129 146 165 192 233	21 43 81 118 153 190 223	21 41 79 115 139 163 204
241004	3	83 164 217	34 100 142	84 164 217	84 161 214	84 164 217	85 165 216
	5	35 101 136 159 217	20 81 135 167 225	47 102 158 183 218	43 82 111 155 216	50 105 162 195 225	44 100 144 166 218
	7	34 72 102 128 160 196 226	8 26 42 79 132 173 196	44 68 98 127 156 186 219	46 75 91 113 157 197 227	43 97 143 170 199 216 232	50 80 109 138 163 189 220
385028	3	66 131 193	99 116 218	76 132 194	77 157 217	74 128 192	76 133 194
	5	49 98 137 175 215	39 95 137 169 219	56 93 133 173 214	56 119 165 198 231	54 90 119 156 210	59 98 136 175 215
	7	44 87 113 139 166 193 224	61 84 117 135 162 202 226	54 88 114 139 167 196 226	42 66 99 139 172 203 231	53 86 112 137 166 193 225	53 86 112 133 160 188 222
388016	3	53 97 176	92 149 206	52 97 175	51 128 205	46 121 204	51 95 170
	5	51 94 140 186 221	64 114 155 204 246	49 91 133 176 214	41 97 120 159 217	49 91 138 182 217	50 90 135 182 215
	7	48 88 118 147 172 196 226	20 47 99 126 146 200 217	47 87 116 147 173 196 224	48 95 117 139 168 190 221	47 88 113 142 170 193 224	43 81 110 141 170 193 222
2092	3	42 97 181	35 169 204	41 97 175	36 96 180	41 97 175	40 98 174
	5	29 58 92 126 191	43 66 122 212 248	31 61 94 127 181	30 62 92 119 181	32 63 92 121 174	30 62 96 127 175
	7	30 60 94 127 156 185 220	34 70 79 110 159 191 227	24 47 65 88 112 132 185	36 71 105 144 172 194 217	25 52 67 85 112 137 189	21 46 72 96 115 136 174
14037	3	59 106 181	97 172 215	58 105 180	57 145 220	96 183 219	57 102 178
	5	59 107 148 188 222	19 52 128 181 213	60 109 148 187 221	56 105 156 202 229	56 104 148 189 223	55 99 144 188 222
	7	44 77 104 130 161 192 224	30 127 141 152 181 198 223	43 75 103 131 161 190 223	52 98 133 165 187 208 232	38 69 99 126 161 192 224	43 70 103 135 165 192 224
55067	3	40 65 140	46 183 199	40 79 148	40 79 149	40 79 147	40 79 147
	5	39 64 99 134 171	30 63 83 118 164	38 63 99 135 170	39 84 119 159 203	37 63 100 137 172	38 62 98 135 173
	7	39 63 81 110 137 165 200	5 33 74 125 169 187 237	40 63 82 112 139 167 200	36 55 77 113 161 191 206	40 61 81 112 139 170 205	37 59 82 107 146 180 200
169012	3	82 143 199	126 177 210	78 139 197	65 130 197	80 142 199	80 139 197
	5	55 100 141 181 218	15 19 59 80 143	55 100 137 180 220	51 87 122 165 216	52 98 136 177 218	50 103 143 187 222
	7	41 78 111 139 169 198 227	6 23 57 74 119 174 219	40 74 105 137 165 197 229	43 90 114 128 162 190 223	41 74 100 132 168 197 227	40 74 103 129 160 195 226

Table 10
Mean of the objective function values obtained using various algorithms.

Im	PR	RGN	MHA	HPFPPA-D	PSO	PFA	DE
41004	3	17.3866	17.3865	17.3863	17.2156	17.2782	17.2791
	5	24.2118	24.2118	24.2115	23.9600	24.1390	24.0352
	7	31.1428	31.1428	31.1426	29.4424	30.9325	29.9948
176035	3	17.7603	17.7602	17.7601	17.5834	17.7495	17.7489
	5	25.2172	25.2171	25.2168	24.4512	25.1378	25.1447
	7	31.3428	31.3428	31.3426	30.5019	31.1211	30.8947
225017	3	18.0953	18.0952	18.0951	18.0792	18.0926	18.0747
	5	24.8891	24.8891	24.8890	24.5326	24.8027	24.6752
	7	31.7469	31.7469	31.7468	30.5922	31.5669	31.1628
241004	3	17.8439	17.8438	17.8439	17.3242	17.8300	17.8167
	5	24.3559	24.3558	24.3560	23.1237	24.1762	24.0404
	7	30.5444	30.5441	30.5444	29.6791	29.5852	30.0270
385028	3	18.4241	18.4241	18.4240	18.1789	18.4032	18.4061
	5	25.0346	25.0347	25.0345	24.6457	24.9557	24.9560
	7	31.3952	31.3952	31.3952	30.5094	31.1828	30.9945
388016	3	17.7145	17.7145	17.7143	17.3642	17.5825	17.6860
	5	25.0006	25.0008	25.0004	23.5646	24.9431	24.8289
	7	31.1422	31.1422	31.1422	30.5295	30.8938	30.5589
2092	3	17.2422	17.2423	17.2419	17.0647	17.2309	17.2207
	5	24.0935	24.0937	24.0933	23.8854	23.9760	23.8218
	7	29.5949	29.5949	29.5849	28.2241	29.3500	29.1886
14037	3	18.0705	18.0705	18.0702	17.6458	17.8294	18.0524
	5	25.4904	25.4904	25.4897	24.9791	25.3580	25.2849
	7	31.5336	31.5337	31.5335	30.9586	31.2953	31.1121
55067	3	16.8167	16.8168	16.8165	16.6824	16.7545	16.7693
	5	23.0715	23.0714	23.0715	22.1780	22.9317	22.8726
	7	28.3676	28.3676	28.3676	27.6598	28.1647	28.0640
169012	3	18.3465	18.3466	18.3464	18.2438	18.3321	18.3382
	5	25.1464	25.1464	25.1463	24.9975	25.1118	25.1026
	7	31.4109	31.4109	31.4109	31.0277	31.2431	31.2794

To evaluate the RGN-based segmentation approach using TII-FE for thresholding, three criteria are employed to assess the quality of the segmentation outcomes. These criteria are peak signal-to-noise ratio (PSNR), structural similitude index (SSIM), and mean squared error (MSE). PSNR measures the similarity between the segmented image and the original image by considering the mean squared error (MSE) of each pixel [69]. Higher PSNR values indicate a better resemblance between the segmented and original images. SSIM is used to compare the structural characteristics of the segmented image, where a higher SSIM value signifies better preservation of the original image's structures.

Table 9 showcases the best thresholds generated by RGN for different thresholds on the benchmark images. The best results obtained by other comparative algorithms (PSO, DE, PFA, HPFPPA-D and MHA) are also included for comparison purposes in Table 9. The type II fuzzy entropy values obtained are given in Table 10, the proposed RGN attain better solutions with higher fitness values outperforms competitive algorithms.

The graphical representation of the RGN-based MLT results is shown in Fig. 1, which includes the segmented images, histograms, and threshold locations. As the threshold count increases, the output shows enhancement in the resulting images. Additionally, the convergence curves for the benchmark image set are depicted in Fig. 1, indicating that the RGN approach achieves faster convergence in most cases.

Table 11 presents the quality metric values that confirms the segmented images attained using RGN and TII-FE outperform other approaches. The RGN approach demonstrates superior performance in terms of MSE, PSNR, and SSIM metrics, indicating that the threshold images produced by this method have less noise and effectively retain the structures reflecting the objects in the image. Overall, the results and analysis confirm the effectiveness of the RGN-based segmentation approach using TII-FE in generating high-quality segmented images compared to other related methods.

4. Discussion

This section provides a detailed summary of the presented results, with drawbacks and future insightful implications. The details are presented as

4.1. Summary of results

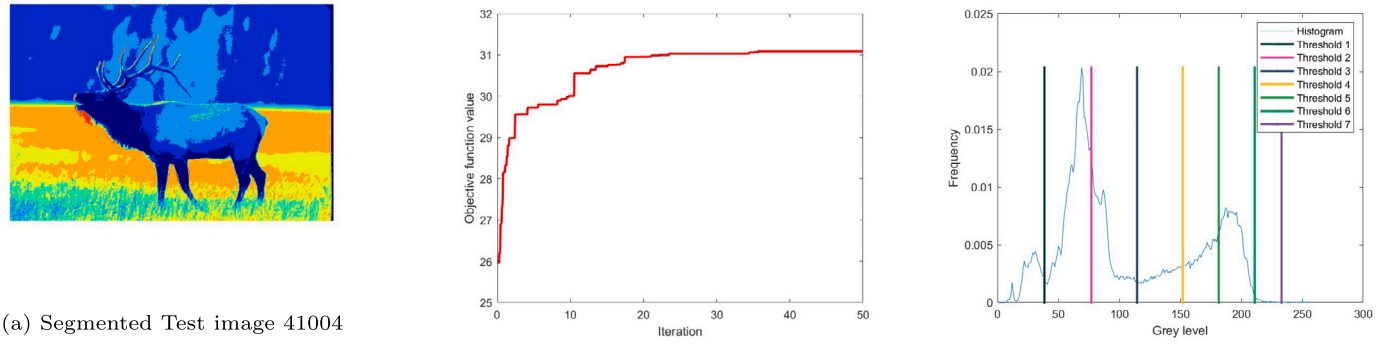
- A new algorithm namely RGN, based on the added properties of GWO-CS, RPO and NMRA is proposed. The added properties are mainly to enhance the exploration and exploitation properties of both the algorithms.
- The algorithms parameters are adapted by using six $mo's$ namely $sig\ mo$, $sa\ mo$, $log\ mo$, $osc\ mo$, $chaos\ mo$ and $exp\ mo$. And sensitivity analysis shows that for the parameters: λ, $exp\ mo$ provides the best results; for R, $chaos\ mo$ is the best; and for $log\ mo$ provides the best set of results. Thus making the algorithm self-adaptive and self-reliant in application.
- The proposed algorithm is tested on CEC 2005, CEC 2014 and CEC 2022 benchmark problems for performance evaluation, and it has been found that the RGN algorithm performs better with respect to SaDE, SHADE, LSHADE-SPACMA, GWO-E, CMA-ES, LX-BBO, LX-TLA, CCS, IMEHO, PBIL, VNBA, RW-GWO, ISOS, NL-SHADE-LBC, S-LSHADE-DP, EA4eig, NL-SHADE-RSP-MID, KOA among others.
- The statistical ranksum and p-rank tests further prove the significance of the proposed algorithm, and it has been found that RGN scores the first rank among all other algorithms under comparison.
- For analysing the performance of the proposed RGN on real-world problems, we utilized multi-level image thresholding with type II Fuzzy sets. The results are presented in terms of MSE, PSNR, and SSIM, and we find that RGN is better with respect to MHA, HPFPPA-D, PSO, PFA and DE. Note that MHA, and HPFPPA-D are two recently introduced hybrid algorithms and a comparison with these prove the significance of the proposed algorithm for real-world optimization problems.

4.2. Drawbacks

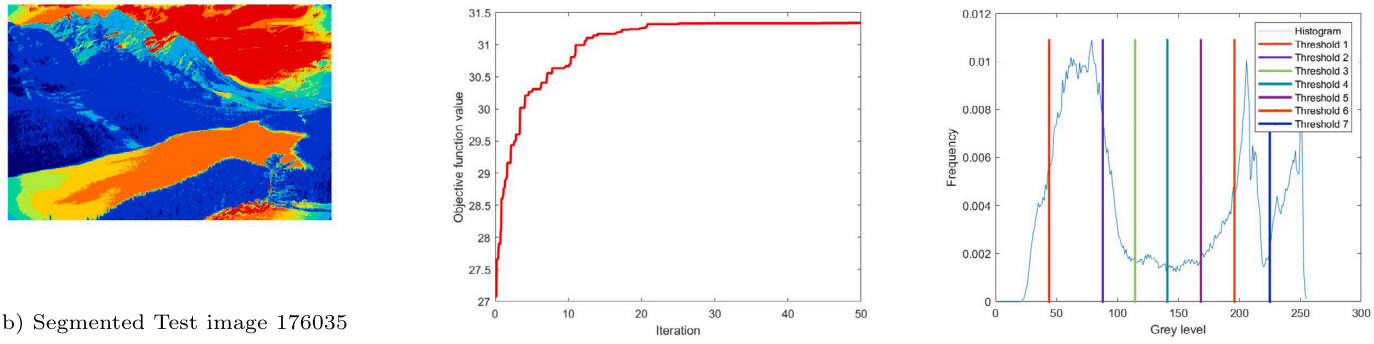
- While the proposed RGN algorithm has demonstrated highly reliable results, there is room for improvement in achieving a better balance between exploration and exploitation. The algorithm may currently favour exploitation through the worker phase, potentially limiting its ability to explore new regions of the search space effectively. Enhancements to the breeder phase and the addition of more sophisticated exploitation operations could address this issue and further improve the algorithm's performance.
- Although the RGN algorithm has shown improvements in its stagnation phase, determining the appropriate conditions for triggering this phase remains critical. A more in-depth analysis is needed to better understand the activation of the stagnation phase. The algorithm should be carefully designed to avoid getting stuck in this phase, as it may limit its ability to explore and exploit the search space effectively, potentially leading to a lack of internal capabilities in the basic structure.
- RGN currently lacks a proper balanced exploration and exploitation phase, as evidenced by suboptimal convergence patterns. More extensive research and development are required to incorporate a well-balanced approach. Rigorous studies should be conducted to improve the algorithm's exploration capabilities, which would likely lead to more effective and efficient convergence patterns. Addressing this aspect would be essential to enhance the overall performance and effectiveness of the RGN algorithm.

4.3. Insightful implications

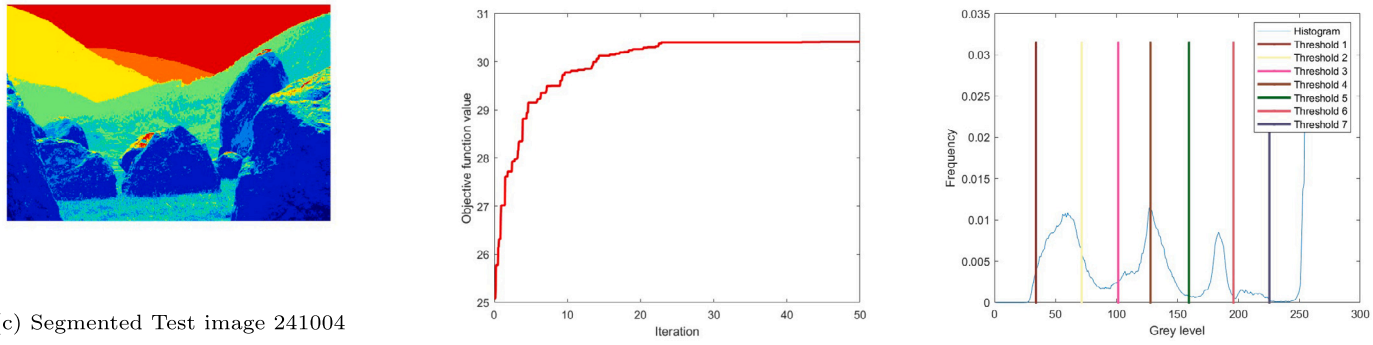
- Despite the mentioned disadvantages, the RGN algorithm has demonstrated its potential to serve as a foundational algorithm for



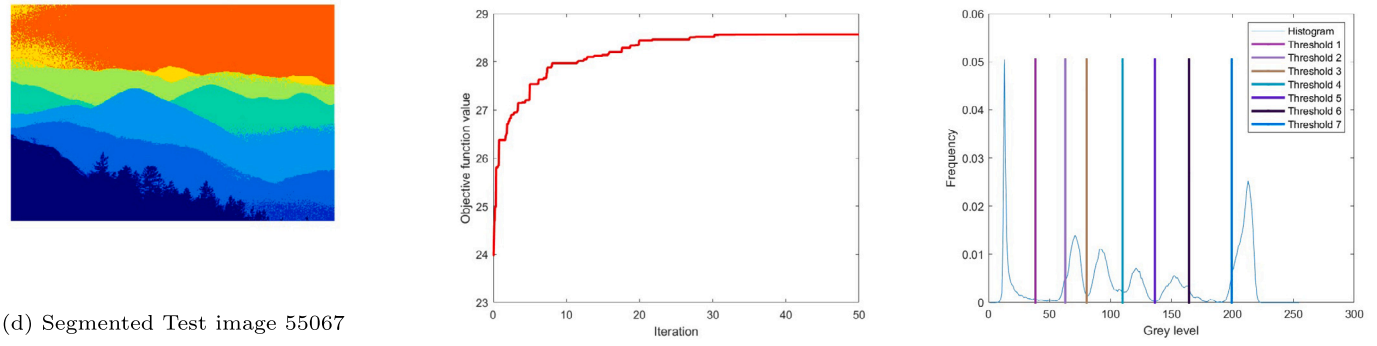
(a) Segmented Test image 41004



(b) Segmented Test image 176035

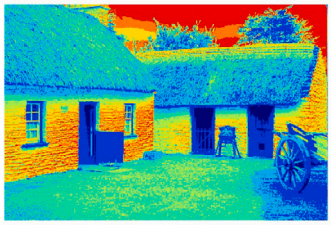


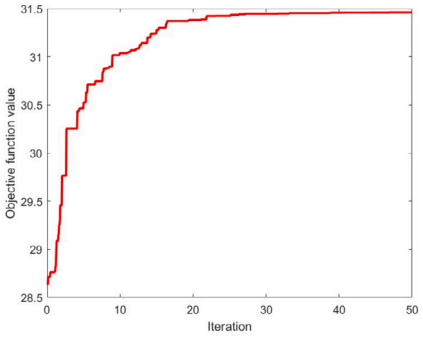
(c) Segmented Test image 241004

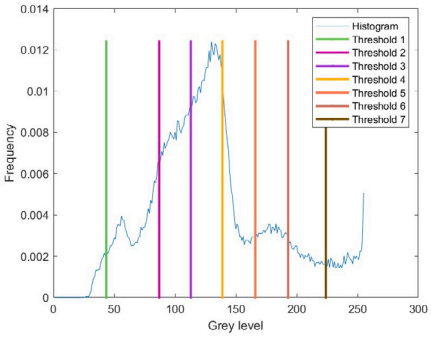


(d) Segmented Test image 55067

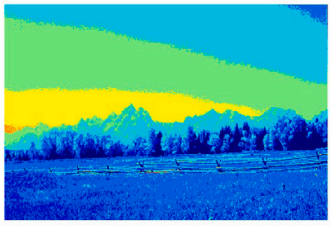
Fig. 1. Test segmented images with corresponding convergence graph and histogram obtained using RGN for $PR = 7$ levels.

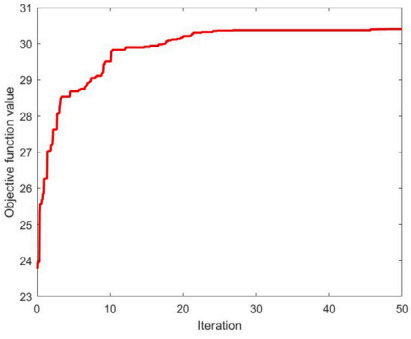


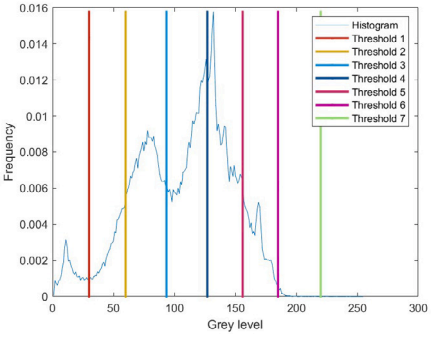




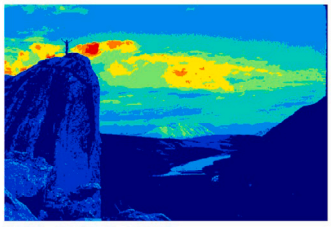
(e) Segmented Test image 385028

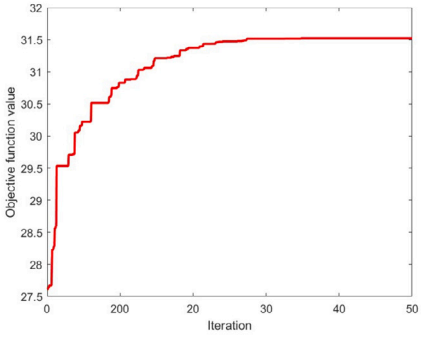


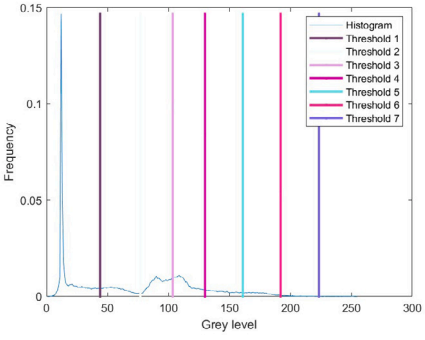




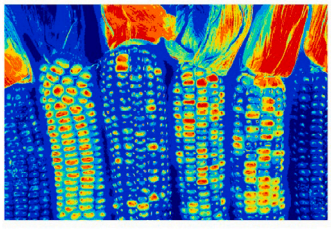
(f) Segmented Test image 2092

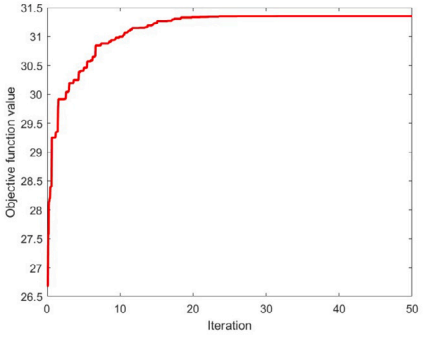


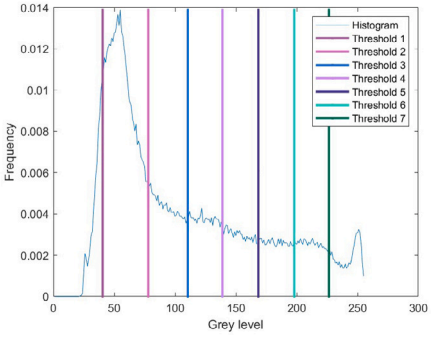




(g) Segmented Test image 14037

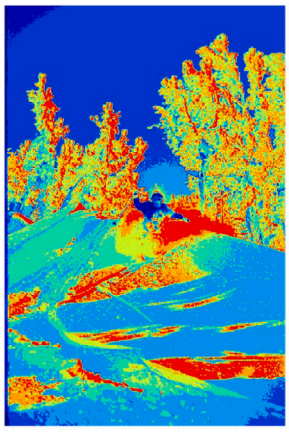


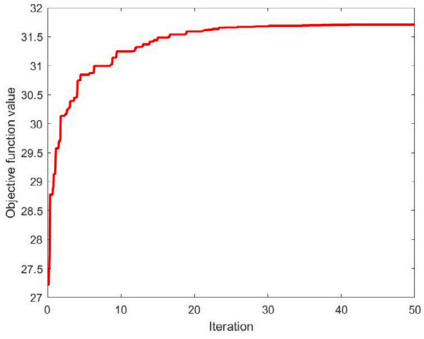


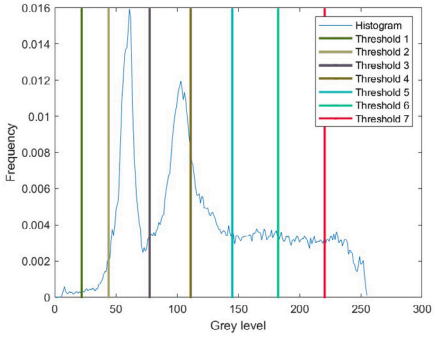


(h) Segmented Test image 169012

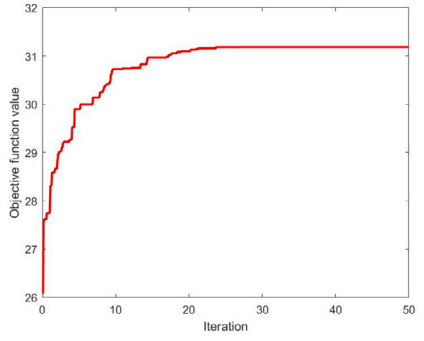
Fig. 1. (*continued*)

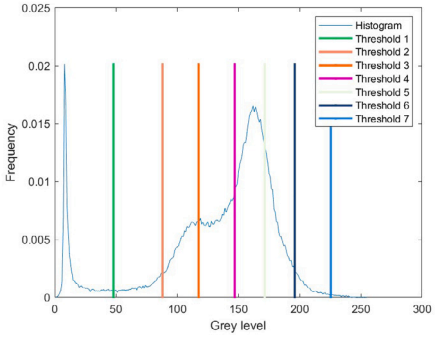






(i) Segmented Test image 225017





(j) Segmented Test image 388016

Fig. 1. (*continued*)

various domain research problems. Its simple structure, ease of implementation, and self-adaptive nature make it a suitable choice for different optimization tasks.

- To make the algorithm more versatile and effective for real-world optimization problems, it is essential to achieve a proper balance between exploration and exploitation. Incorporating a well-balanced approach can enhance its performance and applicability across different domains.
- Moreover, expanding the algorithm to support binary and multi-objective optimization can open up new avenues of research. By applying it to linear antenna arrays, electroencephalogram, and other domains, the algorithm's potential can be further explored.
- To ensure its reliability and efficiency, deeper analyses of stability, convergence properties, and empirical studies are essential. These investigations can provide insights into how well the algorithm performs under different scenarios and help refine its behaviour.
- Additionally, theoretical studies can be conducted to gain a better understanding of the underlying mechanisms and principles of the proposed RGN algorithm. This theoretical foundation can offer valuable insights into its working and potential improvements.
- Overall, with further research and enhancements, the RGN algorithm has the potential to become a valuable tool in various domains, providing efficient and reliable solutions to complex optimization problems

5. Conclusion

This paper presents a hybrid RGH algorithm based on the added properties of RPO, GWO and NMRA. The proposed algorithm uses a combination of equations, formulated using iterative division and adaptive $mo's$ for performance enhancement. We are using a combination of six $mo's$ and sensitivity analysis shows that $exp\ mo$, $chaos\ mo$, and $log\ mo$ is the best combination for parameter adjustments. An analysis of the proposed algorithm with respect to SaDE, CMA-ES, LSHADE-SPACMA, GWO-E, SHADE, and EO for CEC 2005 shows that the proposed RGN scores the first rank. For CEC 2014, comparison with B-BBO, LX-BBO, LX-TLA, RW-GWO, PBIL, ISOS, VNBA, CCS, and IMEHO, further proves the significance of the proposed algorithm for this benchmark dataset also. Again for CEC 2022, NL-SHADE-LBC, NL-SHADE-RSP-MID, S-LSHADE-DP, EA4eig, KOA, DE and PSO, shows that RGN is superior for this dataset also. Note that performance evaluation is done using wilcoxon's ranksum test and Freidmann test, and it proves the significance of the proposed algorithm statistically. Further, this paper introduces a novel MLT method based on type II fuzzy entropy

Table 11
The comparison of quality matrices obtained by various algorithms.

Im	PR	Matric	RGN	MHA	HPFPPA-D	PSO	PFA	DE
41004	3	MSE	3.52E+02	3.54E+02	3.56E+02	4.06E+02	3.83E+02	3.63E+02
		PSNR	2.28E+01	2.28E+01	2.26E+01	2.20E+01	2.23E+01	2.25E+01
		SSIM	8.76E-01	8.75E-01	8.76E-01	8.70E-01	8.73E-01	8.74E-01
	5	MSE	1.30E+02	1.36E+02	1.38E+02	1.28E+02	1.74E+02	1.30E+02
		PSNR	2.71E+02	2.71E+02	2.70E+01	2.57E+01	2.70E+01	2.48E+01
		SSIM	9.22E-01	9.21E-01	9.21E-01	9.07E-01	9.12E-01	8.84E-01
	7	MSE	8.62E+01	8.69E+01	8.71E+01	1.02E+02	1.05E+02	8.19E+01
		PSNR	2.91E+01	2.94E+01	2.87E+01	2.79E+01	2.86E+01	2.80E+01
		SSIM	9.23E-01	9.23E-01	9.22E-01	9.15E-01	9.22E-01	9.22E-01
176035	3	MSE	3.08E+02	3.06E+02	3.09E+02	3.64E+02	3.68E+02	3.94E+02
		PSNR	2.38E+01	2.39E+01	2.35E+01	2.32E+01	2.26E+01	2.25E+01
		SSIM	8.51E-01	8.49E-01	8.53E-01	8.45E-01	8.34E-01	8.29E-01
	5	MSE	1.28E+02	1.28E+02	1.30E+02	1.69E+02	1.32E+02	1.30E+02
		PSNR	2.74E+01	2.73E+01	2.70E+01	2.59E+01	2.69E+01	2.70E+01
		SSIM	8.73E-01	8.71E-01	8.71E-01	8.57E-01	8.70E-01	8.71E-01
	7	MSE	8.02E+01	8.02E+01	8.04E+01	8.25E+01	1.00E+02	8.16E+01
		PSNR	2.90E+01	2.91E+01	2.91E+01	2.81E+01	2.90E+01	2.90E+01
		SSIM	8.96E-01	8.98E-01	8.98E-01	8.84E-01	8.95E-01	8.94E-01
225017	3	MSE	3.30E+02	3.29E+02	3.31E+02	3.19E+02	3.27E+02	3.41E+02
		PSNR	2.31E+01	2.31E+01	2.31E+01	2.30E+01	2.28E+01	2.27E+01
		SSIM	8.47E-01	8.45E-01	8.48E-01	8.45E-01	8.39E-01	8.36E-01
	5	MSE	1.39E+02	1.41E+02	1.43E+02	1.59E+02	1.71E+02	1.49E+02
		PSNR	2.68E+01	2.68E+01	2.66E+01	2.58E+01	2.64E+01	2.61E+01
		SSIM	9.02E-01	9.01E-01	9.01E-01	8.91E-01	9.00E-01	8.97E-01
	7	MSE	8.86E+01	8.86E+01	8.89E+01	9.53E+01	1.09E+02	8.94E+01
		PSNR	2.86E+01	2.86E+01	2.86E+01	2.78E+01	2.86E+01	2.83E+01
		SSIM	9.24E-01	9.23E-01	9.22E-01	9.16E-01	9.22E-01	9.21E-01
241004	3	MSE	2.13E+02	2.13E+02	2.15E+02	2.18E+02	2.18E+02	2.19E+02
		PSNR	2.49E+01	2.49E+02	2.47E+01	2.48E+01	2.48E+01	2.48E+01
		SSIM	8.97E-01	8.97E-01	8.97E-01	8.97E-01	8.96E-01	8.96E-01
	5	MSE	1.37E+02	1.38E+02	1.40E+02	1.78E+02	1.88E+02	1.95E+02
		PSNR	2.64E+01	2.62E+01	2.54E+01	2.67E+01	2.57E+01	2.56E+01
		SSIM	8.90E-01	8.91E-01	8.91E-01	8.71E-01	8.87E-01	8.83E-01
	7	MSE	7.78E+01	7.76E+01	7.79E+01	1.01E+02	8.68E+01	9.55E+01
		PSNR	2.98E+01	2.98E+01	2.92E+01	2.83E+01	2.81E+01	2.87E+01
		SSIM	9.30E-01	9.29E-01	9.30E-01	9.20E-01	9.15E-01	9.21E-01
385028	3	MSE	3.18E+02	3.18E+02	3.19E+02	3.29E+02	3.28E+02	3.32E+02
		PSNR	2.43E+01	2.44E+01	2.40E+01	2.31E+01	2.30E+01	2.31E+01
		SSIM	7.85E-01	7.84E-01	7.86E-01	7.82E-01	7.67E-01	7.69E-01
	5	MSE	1.26E+02	1.24E+02	1.26E+02	1.76E+02	1.41E+02	1.38E+02
		PSNR	2.69E+01	2.71E+01	2.71E+01	2.57E+01	2.66E+01	2.67E+01
		SSIM	8.67E-01	8.69E-01	8.68E-01	8.44E-01	8.46E-01	8.48E-01
	7	MSE	7.14E+01	7.13E+01	7.15E+01	7.27E+01	7.57E+01	7.25E+01
		PSNR	2.98E+01	3.00E+01	2.96E+01	2.93E+01	2.95E+01	2.95E+01
		SSIM	9.04E-01	9.05E-01	9.07E-01	9.02E-01	9.06E-01	9.03E-01

(TII-FE) and RGN. The proposed approach is evaluated on benchmark images and compared to other algorithms in terms of accuracy, convergence, and the quality of segmented images. Quality metrics such as MSE, PSNR, and SSIM are used for evaluation. The results show that the RGN-based thresholding approach using TII-FE is effective in image thresholding tasks, outperforming other methods.

For future, the proposed algorithm can be used for solving various other domain research problems including medical imaging, wireless sensor networks, antenna array designs, parametric estimation of fuel cells, photovoltaic systems and others. The algorithm can be enhanced further by analysing which parameter is best for exploration, and which one provides more reliable results for exploitation. Another important aspect to deal with is the parameter dependence, where the current algorithm parameters are self-adaptive, so we don't have to define these parameters over subsequent generations. Prospective researchers can define new $mo's$ for their application and can see how the algorithm behaves. Overall, RGN is found to be highly reliable and can be considered as a potential candidate for optimization research.

CRediT authorship contribution statement

Rohit Salgotra: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Nitin Mittal:** Writing – review & editing, Validation, Investigation, Formal analysis, Data curation. **Abdulaziz S. Almazyad:** Validation, Supervision, Resources, Project administration, Investigation, Funding acquisition. **Ali Wagdy Mohamed:** Supervision, Software, Resources, Funding acquisition.

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.

Table 11 (*continued*)

Im	PR	Matric	RGN	MHA	HPFPPA-D	PSO	PFA	DE
388016	3	MSE	3.50E+02	3.50E+01	3.54E+02	5.32E+02	3.76E+02	5.49E+02
		PSNR	2.24E+01	2.21E+01	2.09E+01	2.26E+01	2.24E+01	2.26E+01
		SSIM	7.49E-01	7.51E-01	7.44E-01	7.39E-01	7.33E-01	7.38E-01
	5	MSE	1.41E+02	1.41E+01	1.46E+02	1.51E+02	1.53E+02	2.13E+02
		PSNR	2.68E+01	2.68E+01	2.66E+01	2.65E+01	2.64E+01	2.63E+01
		SSIM	8.29E-01	8.29E-01	8.29E-01	8.25E-01	8.21E-01	8.20E-01
	7	MSE	8.01E+01	8.02E+02	8.04E+01	8.25E+01	8.35E+01	8.19E+01
		PSNR	2.91E+01	2.92E+01	2.91E+01	2.89E+01	2.90E+01	2.90E+01
		SSIM	8.78E-01	8.79E-01	8.82E-01	8.72E-01	8.78E-01	8.77E-01
2092	3	MSE	3.00E+02	3.01E+02	3.06E+02	3.10E+02	3.15E+02	3.48E+02
		PSNR	2.32E+01	2.34E+01	2.31E+01	2.33E+01	2.32E+01	2.32E+01
		SSIM	8.83E-01	8.83E-01	8.82E-01	8.82E-01	8.82E-01	8.82E-01
	5	MSE	1.27E+02	1.25E+02	1.32E+02	1.51E+02	1.48E+02	1.55E+02
		PSNR	2.71E+01	2.71E+01	2.69E+01	2.64E+01	2.62E+01	2.63E+01
		SSIM	9.22E-01	9.22E-01	9.22E-01	9.22E-01	9.17E-01	9.20E-01
	7	MSE	6.63E+01	6.58E+01	7.00E+01	8.19E+01	8.37E+01	7.03E+01
		PSNR	2.97E+01	2.96E+01	2.97E+01	2.89E+01	2.97E+01	2.90E+01
		SSIM	9.51E-01	9.49E-01	9.52E-01	9.28E-01	9.51E-01	9.46E-01
14037	3	MSE	3.11E+02	3.10E+01	3.13E+02	3.58E+02	3.51E+02	9.28E+02
		PSNR	2.37E+01	2.38E+01	2.36E+01	2.32E+01	2.27E+01	2.27E+01
		SSIM	8.11E-01	8.09E-01	8.11E-01	8.03E-01	7.92E-01	7.99E-01
	5	MSE	2.11E+02	2.10E+01	2.12E+02	2.23E+02	2.32E+02	2.20E+02
		PSNR	2.53E+01	2.56E+01	2.49E+01	2.45E+01	2.47E+01	2.47E+01
		SSIM	8.37E-01	8.36E-01	8.38E-01	8.09E-01	8.21E-01	8.13E-01
	7	MSE	7.28E+01	7.26E+01	7.31E+01	8.78E+01	9.31E+01	8.33E+01
		PSNR	2.95E+01	2.94E+01	2.95E+01	2.84E+01	2.89E+01	2.87E+01
		SSIM	8.98E-01	8.94E-01	8.97E-01	8.75E-01	8.86E-01	8.77E-01
55067	3	MSE	3.77E+02	3.75E+01	3.76E+02	3.87E+02	3.84E+02	3.84E+02
		PSNR	2.26E+01	2.26E+01	2.24E+01	2.23E+01	2.23E+01	2.23E+01
		SSIM	9.46E-01	9.47E-01	9.44E-01	9.41E-01	9.42E-01	9.44E-01
	5	MSE	8.81E+01	8.83E+01	8.88E+01	1.10E+02	1.10E+02	1.08E+02
		PSNR	2.87E+01	2.88E+01	2.86E+01	2.77E+01	2.78E+01	2.77E+01
		SSIM	9.62E-01	9.59E-01	9.63E-01	9.52E-01	9.60E-01	9.60E-01
	7	MSE	4.23E+01	4.24E+02	4.26E+01	4.66E+01	5.30E+01	4.53E+01
		PSNR	3.21E+01	3.20E+01	3.18E+01	3.09E+01	3.16E+01	3.14E+01
		SSIM	9.67E-01	9.67E-01	9.67E-01	9.59E-01	9.61E-01	9.60E-01
169012	3	MSE	2.70E+02	2.67E+02	2.72E+02	3.01E+02	2.79E+02	2.75E+02
		PSNR	2.42E+01	2.44E+01	2.38E+01	2.33E+01	2.37E+01	2.37E+01
		SSIM	8.06E-01	8.05E-01	8.05E-01	7.83E-01	8.03E-01	8.05E-01
	5	MSE	1.45E+02	1.46E+02	1.47E+02	1.65E+02	1.65E+02	1.53E+02
		PSNR	2.64E+01	2.63E+01	2.64E+01	2.60E+01	2.63E+01	2.60E+01
		SSIM	8.51E-01	8.51E-01	8.47E-01	8.41E-01	8.43E-01	8.41E-01
	7	MSE	8.42E+01	8.42E+01	8.45E+01	1.02E+02	8.58E+01	8.57E+01
		PSNR	2.89E+01	2.88E+01	2.89E+01	2.80E+01	2.88E+01	2.88E+01
		SSIM	8.93E-01	8.92E-01	8.91E-01	8.81E-01	8.90E-01	8.91E-01

Funding Agency

The research is funded by Researchers Supporting Program at King Saud University, (RSPD2024R809).

Acknowledgement

The authors present their appreciation to King Saud University for funding the publication of this research through Researchers Supporting Program (RSPD2024R809), King Saud University, Riyadh, Saudi Arabia.

References

[1] Abd Elaziz M, Sarkar U, Nag S, Hinojosa S, Oliva D. Improving image thresholding by the type ii fuzzy entropy and a hybrid optimization algorithm. Soft Comput 2020:1–21.

[2] Salgotra R, Singh G, Kaur S, Singh U. Two new single/multi-objective multi-strategy algorithms for the parametric estimation of dual band-notched ultra wideband antennas. Knowl-Based Syst 2024;286:111412.

[3] Holland JH. Genetic algorithms. Sci Am 1992;267(1):66–73.

[4] Storn R, Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 1997;11(4):341–59.

[5] Azizi M, Talatahari S, Gandomi AH. Fire hawk optimizer: a novel metaheuristic algorithm. Artif Intell Rev 2023;56(1):287–363.

[6] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks, vol. 4. IEEE; 1995. p. 1942–8.

[7] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Adv Eng Softw 2014;69:46–61.

[8] Salgotra R, Singh U, Sharma S. On the improvement in grey wolf optimization. Neural Comput Appl 2019:1–40.

[9] Salgotra R, Singh U. The naked mole-rat algorithm. Neural Comput Appl 2019;31(12):8837–57.

[10] Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH. Marine predators algorithm: a nature-inspired metaheuristic. Expert Syst Appl 2020:113377.

[11] Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM. Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 2017;114:163–91.

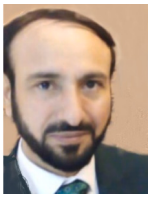
[12] Salgotra R, Singh U, Singh S, Singh G, Mittal N. Self-adaptive salp swarm algorithm for engineering optimization problems. Appl Math Model 2021;89:188–207.

[13] Givi H, Dehghani M, Hubálovskỳ Š. Red panda optimization algorithm: an effective bio-inspired metaheuristic algorithm for solving engineering optimization problems. IEEE Access 2023;11:57203–27.

[14] Xian S, Feng X. Meerkat optimization algorithm: a new meta-heuristic optimization algorithm for solving constrained engineering problems. Expert Syst Appl 2023:120482.
[15] Salgotra R, Singh U. Application of mutation operators to flower pollination algorithm. Expert Syst Appl 2017;79:112–29.
[16] Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S. Equilibrium optimizer: a novel optimization algorithm. Knowl-Based Syst 2020;191:105190.
[17] Liang J, Qu B, Suganthan P. Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report. Singapore: Nanyang Technological University; 2013. p. 635.
[18] Kumar A, Price KV, Mohamed AW, Hadi AA, Suganthan P. Problem definitions and evaluation criteria for the cec 2022 special session and competition on single objective real-parameter numerical optimization. Technical Report. Singapore: Nanyang Technological University; 2022.
[19] Salgotra R, Singh U, Singh S, Mittal N. A hybridized multi-algorithm strategy for engineering optimization problems. Knowl-Based Syst 2021;217:106790.
[20] Zhang J, Sanderson AC. Jade: adaptive differential evolution with optional external archive. IEEE Trans Evol Comput 2009;13(5):945–58.
[21] Garg V, Deep K, Bansal S. Improved teaching learning algorithm with Laplacian operator for solving nonlinear engineering optimization problems. Eng Appl Artif Intell 2023;124:106549.
[22] Wang L, Cao Q, Zhang Z, Mirjalili S, Zhao W. Artificial rabbits optimization: a new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. Eng Appl Artif Intell 2022;114:105082.
[23] Tejani GG, Savsani VJ, Patel VK, Mirjalili S. Truss optimization with natural frequency bounds using improved symbiotic organisms search. Knowl-Based Syst 2018;143:162–78.
[24] Mohamed AW, Hadi AA, Fattouh AM, Jambi KM. Lshade with semi-parameter adaptation hybrid with cma-es for solving cec 2017 benchmark problems. In: 2017 IEEE congress on evolutionary computation (CEC). IEEE; 2017. p. 145–52.
[25] Li W, Wang G-G, Alavi AH. Learning-based elephant herding optimization algorithm for solving numerical optimization problems. Knowl-Based Syst 2020:105675.
[26] Gupta S, Deep K. A novel random walk grey wolf optimizer. Swarm Evol Comput 2019;44:101–12.
[27] Abdel-Basset M, Mohamed R, Sallam KM, Chakrabortty RK. Light spectrum optimizer: a novel physics-inspired metaheuristic optimization algorithm. Mathematics 2022;10(19):3466.
[28] Wang Y, Wang Z, Wang G-G. Hierarchical learning particle swarm optimization using fuzzy logic. Expert Syst Appl 2023:120759.
[29] Abdel-Basset M, Mohamed R, Azeem SAA, Jameel M, Abouhawwash M. Kepler optimization algorithm: a new metaheuristic algorithm inspired by Kepler's laws of planetary motion. Knowl-Based Syst 2023;268:110454.
[30] Onay FK. A novel improved chef-based optimization algorithm with Gaussian random walk-based diffusion process for global optimization and engineering problems. Math Comput Simul 2023;212:195–223.
[31] Aja-Fernández S, Curiale AH, Vegas-Sánchez-Ferrero G. A local fuzzy thresholding methodology for multiregion image segmentation. Knowl-Based Syst 2015;83:1–12.
[32] Masood S, Sharif M, Masood A, Yasmin M, Raza M. A survey on medical image segmentation. Curr Med Imag 2015;11(1):3–14.
[33] Oliva D, Cuevas E, Pajares G, Zaldivar D, Osuna V. A multilevel thresholding algorithm using electromagnetism optimization. Neurocomputing 2014;139:357–81.
[34] Benzid R, Arar D, Bentoumi M. A fast technique for gray level image thresholding and quantization based on the entropy maximization. In: 2008 5th international multi-conference on systems, signals and devices. IEEE; 2008. p. 1–4.
[35] Otsu N. A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 1979;9(1):62–6.
[36] Liu SL, Kong LZ, Wang JG. Segmentation approach based on fuzzy Renyi entropy. In: 2010 Chinese conference on pattern recognition (CCPR). IEEE; 2010. p. 1–4.
[37] Tian W, Geng Y, Liu J, Ai L. Maximum fuzzy entropy and immune clone selection algorithm for image segmentation. In: 2009 Asia-Pacific conference on information processing, vol. 1. IEEE; 2009. p. 38–41.
[38] Zhao M, Fu AM, Yan H. A technique of three-level thresholding based on probability partition and fuzzy 3-partition. IEEE Trans Fuzzy Syst 2001;9(3):469–79.
[39] Tao W-B, Tian J-W, Liu J. Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm. Pattern Recognit Lett 2003;24(16):3069–78.
[40] Tizhoosh HR. Image thresholding using type ii fuzzy sets. Pattern Recognit 2005;38(12):2363–72.
[41] Wu J, Pian Z, Guo L, Wang K, Gao L. Medical image thresholding algorithm based on fuzzy sets theory. In: 2007 2nd IEEE conference on industrial electronics and applications. IEEE; 2007. p. 919–24.
[42] Salgotra R, Singh U, Saha S, Gandomi AH. Self adaptive cuckoo search: analysis and experimentation. Swarm Evol Comput 2020:100751.
[43] Wolpert DH, Macready WG, et al. No free lunch theorems for optimization. IEEE Trans Evol Comput 1997;1(1):67–82.
[44] Salgotra R, Singh U, Saha S. On some improved versions of whale optimization algorithm. Arab J Sci Eng 2019;44(11):9653–91.
[45] Al-Hassan W, Fayek M, Shaheen S. Psosa: an optimized particle swarm technique for solving the urban planning problem. In: 2006 international conference on computer engineering and systems. IEEE; 2006. p. 401–5.
[46] Wang G-G, Deb S, Gandomi AH, Zhang Z, Alavi AH. Chaotic cuckoo search. Soft Comput 2016;20(9):3349–62.
[47] Chen G, Huang X, Jia J, Min Z. Natural exponential inertia weight strategy in particle swarm optimization. In: 2006 6th world congress on intelligent control and automation, vol. 1. IEEE; 2006. p. 3672–5.
[48] Gao H, Zhang Y, Liang S, Li D. A new chaotic algorithm for image encryption. Chaos Solitons Fractals 2006;29(2):393–9.
[49] Tanabe R, Fukunaga A. Success-history based parameter adaptation for differential evolution. In: 2013 IEEE congress on evolutionary computation. IEEE; 2013. p. 71–8.
[50] Khalilpourazari S, Pasandideh SHR. Sine–cosine crow search algorithm: theory and applications. Neural Comput Appl 2019:1–18.
[51] Yousri D, Abd Elaziz M, Mirjalili S. Fractional-order calculus-based flower pollination algorithm with local search for global optimization and image segmentation. Knowl-Based Syst 2020:105889.
[52] Brest J, Zumer V, Maucec MS. Self-adaptive differential evolution algorithm in constrained real-parameter optimization. In: 2006 IEEE international conference on evolutionary computation. IEEE; 2006. p. 215–22.
[53] Mohapatra S, Mohapatra P. Fast random opposition-based learning golden jackal optimization algorithm. Knowl-Based Syst 2023:110679.
[54] Abdel-Basset M, El-Shahat D, Jameel M, Abouhawwash M. Young's double-slit experiment optimizer: a novel metaheuristic optimization algorithm for global and constraint optimization problems. Comput Methods Appl Mech Eng 2023;403:115652.
[55] Stanovov V, Akhmedova S, Semenkin E. Nl-shade-lbc algorithm with linear parameter adaptation bias change for cec 2022 numerical optimization. In: 2022 IEEE congress on evolutionary computation (CEC). IEEE; 2022. p. 01–8.
[56] Van Cuong L, Bao NN, Phuong NK, Binh HTT. Dynamic perturbation for population diversity management in differential evolution. In: Proceedings of the genetic and evolutionary computation conference companion; 2022. p. 391–4.
[57] Bujok P, Kolenovsky P. Eigen crossover in cooperative model of evolutionary algorithms applied to cec 2022 single objective numerical optimisation. In: 2022 IEEE congress on evolutionary computation (CEC). IEEE; 2022. p. 1–8.
[58] Biedrzycki R, Arabas J, Warchulski E. A version of nl-shade-rsp algorithm with midpoint for cec 2022 single objective bound constrained problems. In: 2022 IEEE congress on evolutionary computation (CEC). IEEE; 2022. p. 1–8.
[59] Derrac J, García S, Molina D, Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 2011;1(1):3–18.
[60] Salgotra R, Singh U, Saha S. New cuckoo search algorithms with enhanced exploration and exploitation properties. Expert Syst Appl 2018;95:384–420.
[61] Hansen N, Müller SD, Koumoutsakos P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). Evol Comput 2003;11(1):1–18.
[62] Mlakar U, Fister Jr I, Fister I. Hybrid self-adaptive cuckoo search for global optimization. Swarm Evol Comput 2016;29:47–72.
[63] Garg V, Deep K. Performance of Laplacian biogeography-based optimization algorithm on cec 2014 continuous optimization benchmarks and camera calibration problem. Swarm Evol Comput 2016;27:132–44.
[64] Wang G-G, Lu M, Zhao X-J. An improved bat algorithm with variable neighborhood search for global optimization. In: 2016 IEEE congress on evolutionary computation (CEC). IEEE; 2016. p. 1773–8.
[65] Wang G-G, Guo L, Gandomi AH, Hao G-S, Wang H. Chaotic krill herd algorithm. Inf Sci 2014;274:17–34.
[66] Abdel-Basset M, El-Shahat D, Jameel M, Abouhawwash M. Exponential distribution optimizer (edo): a novel math-inspired algorithm for global optimization and engineering problems. Artif Intell Rev 2023:1–72.
[67] Ahmadianfar I, Heidari AA, Noshadian S, Chen H, Gandomi AH. Info: an efficient optimization algorithm based on weighted mean of vectors. Expert Syst Appl 2022;195:116516.
[68] Martin D, Fowlkes C, Tal D, Malik J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings eighth IEEE international conference on computer vision. ICCV 2001, vol. 2. IEEE; 2001. p. 416–23.
[69] Horng M-H, Liou R-J. Multilevel minimum cross entropy threshold selection based on the firefly algorithm. Expert Syst Appl 2011;38(12):14805–11.

Rohit Salgotra is currently associated with the Faculty of Physics and Applied Computer Science, AGH University of Kraków, Poland. He is also working as a Visiting Fellow at the Data Science Institute, University of Technology Sydney, Australia. He has published more 50 high-impact SCI journals and has reviewed articles for various journals of Elsevier, Springer and IEEE. He has received various travel grants for international travel including IEEE Computational Intelligence Society travel grant for CEC 2018 and CEC 2019 respectively. His research interests include evolutionary computing, optimization and antenna array design applications. His name is included in the list of Indian Researchers in Stanford University's Top 2% Most Influential Scientists List for the Year 2020 and 2021.

Nitin Mittal is currently working as Skill Assistant Professor in Skill Faculty of Engineering and Technology, Shri Vishwakarma Skill University, Haryana, India. He received his B.Tech and M.Tech degree in Electronics and Communication Engineering from Kurukshetra University, Kurukshetra, India in 2006 and 2009 respectively. He has completed his Ph.D. in ECE from Chandigarh University, Mohali, India in 2017. His name is included in the list of Indian Researchers in Stanford University's Top 2% Most Influential Scientists List for the Year 2020, 2021 and 2022. His research interests include Wireless Sensor Networks, Image Segmentation, and Soft Computing.



Abdulaziz S. Almazyad received a Ph.D. degree in computer engineering from Syracuse University, Syracuse, NY, USA. He is a Professor at the College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. His research interests include the Internet of Things, cloud computing, artificial intelligence, mobile and wireless networks, and information security.

Ali Wagdy Mohamed received his B.Sc., M.Sc. and PhD. degrees from Cairo University, in 2000, 2004 and 2010, respectively. Ali Wagdy is Professor and chair, Operations Research department, Faculty of Graduate Studies for Statistical Research), Cairo University, Egypt. He was an Associate Professor of Statistics at Wireless Intelligent Networks Center (WINC), Faculty of Engineering and Applied Sciences, Nile University (2019-2021). Currently, I am a Professor with Mathematics and Actuarial Science Department, School of Sciences and Engineering, The American University in Cairo, Cairo, Egypt. Recently, he has appointed as a member of Education and Scientific Research Policy Council, Academy of Scientific Research (2021-2024). Recently, he has recognized among the top 2% scientists according to Stanford University report 2020, 2021 and 2022, Respectively. He serves as reviewer of more than 100 international accredited top-tier journals and has been awarded the Publons Peer Review Awards 2018, for placing in the top 1% of reviewers worldwide in assorted field. He is an associate editor with Swarm and Evolutionary Computation Journal, Elsevier. He is the Chair of the Egyptian Chapter of AFROS (African Federation of Operations Research Societies). He is editor in more than 10 journals of information sciences, applied mathematics, Engineering, system science and Operations Research. He has presented and participated in more than 10 international conferences. He participated as a member of the reviewer committee for 35 different conferences sponsored by Springer and IEEE. He published more than 140 papers in reputed and high impact journals.