Science, Technology and Innovation
ISSN 2544-9125 ▪ 2023; 18 (3-4): 1-14
doi: 10.55225/sti.528

STI • 2023; 18 (3-4): 1-14

Application of RRAP reliability optimization as a test of nature-inspired algorithms

Adam Pieprzycki[1] A-G, Bogusław Filipowicz[1] A-G

[1]University of Applied Sciences in Tarnow, Polytechnic Faculty, Department of Computer Science, ul. Mickiewicza 8, 33-100 Tarnów, Poland

Original article

Abstract

This paper presents a discussion on the application of two swarm intelligence algorithms, Cuckoo Search (CS) and Firefly Algorithm (FA), to maximize the reliability of two complex systems with resource constraints, which have been well-known in the literature. The reliability of the systems is also evaluated using several classical methods. The results indicate that although the CS algorithm, which utilizes Lévy flight, is eective, the FA rey algorithm outperformed it in the presented optimization tasks, within the given parameter range. These ndings contribute to the ongoing discussion on using nature-inspired algorithms for solving Reliability Redundancy Allocation Problem (RRAP) problems, and the two test scenarios used in the study can be useful for validating other algorithms in RRAP problems. The paper introduces metrics and methods for analyzing and comparing the performance of algorithms in RRAP optimization, including the comparison of criterion function values and other parameters introduced in the paper. Additionally, the paper discusses statistical analyses of variance (ANOVA) with post-hoc RIR Tuckey tests.

Keywords

- reliability optimization
- RRAP
- Firefly Algorithm (FA)
- Cuckoo Search (CS)
- ANOVA
- Lévy flight

Authors contributions

A – Preparation of the research project
B – Assembly of data for the research undertaken
C – Conducting of statistical analysis
D – Interpretation of results
E – Manuscript preparation
F – Literature review
G – Revising the manuscript

Corresponding author

Adam Pieprzycki
e-mail: a_pieprzycki@atar.edu.pl
Akademia Tarnowska
Wydział Politechniczny
Katedra Informatyki
ul. Mickiewicza 8
33-100 Tarnów, Poland

Article info

Article history

- Received: 2023-08-29
- Accepted: 2023-12-29
- Published: 2023-02-15

Publisher

University of Applied Sciences in Tarnow
ul. Mickiewicza 8, 33-100 Tarnow, Poland

Financing

This research did not received any grants from public, commercial or non-profit organizations.

Conflict of interest

None declared.

1. Introduction

The concept of reliability in non-renewable objects refers to their ability to maintain the required properties for their intended purpose [1]. Optimizing reliability can involve increasing safety or reducing costs by decreasing reliability. To enhance reliability, each component of the system can be improved, or redundancy can be introduced to individual subsystems. Designing an extremely reliable system requires balanced approach that considers price, weight, volume, and lifetime when introducing redundant elements. This type of non-linear design with resource constraints is known as the Reliability Redundancy Allocation Problem (RRAP), which is a type of NP-hard problem. The optimization task involves allocating system components (number of elements, reliability level) to maximize total reliability while satisfying the constraints. Redundancy is desirable and can be protected in the event of a system failure. Critical components can be duplicated, or several components can be used in parallel to avoid system breakdown. For example, in engineering, parallel operation of temperature sensor contacts or redundant devices in the power grid can enhance reliability. In computing, the use of RAID (Redundant Array of Independent Disks) disk arrays can increase system reliability. In this study, we aim to determine the number of redundant components and their reliability values in each subsystem to maximize the system's total eliability while adhering to the system design constraints using the Reliability Redundancy Allocation Problem (RRAP).

To address this problem, we utilized two heuristic algorithms inspired by social behavior in animals, namely the Cuckoo Search (CS) algorithm and the Firefly Algorithm (FA) for non-linear design with resource constraints optimization problems [2]. The effciency of these algorithms was evaluated by applying them to two different scenarios [3–5].

Recent studies have shown that the CS algorithm is as effective as other popular algorithms such as PSO (Particle Swarm Optimization) and genetic algorithms [6]. However, the FA algorithm proved to be more effective for the optimization tasks presented in this study within the considered parameter range.

Swarm optimization algorithms have also been successfully applied to reliability optimization in power grid systems, where a reliability function is represented using various methods such as Binary Decision Diagrams (BDDs), Edge Expansion Diagrams (EEDs), Composition After Expansion (CAE), and fixed-sink algorithms for k-terminal networks [7–9].

Based on the literature compilation [2,5], other algorithms such as Simulated Annealing (SA) [10], Particle Swarm Optimization (PSO) [11], Modied Particle Swarm Optimization (MPSO) [12], Articial Bee Colony (ABC) [13], CS-GA [14], BAT (Bat algorithm) [15], ACO (Ant Colony Optimization) [16], Enhanced Nest Cuckoo Optimization Algorithm [17], Grey Wolf Optimization Algorithm (GWO) [18,19] and the results presented in the works [13,20–28] were not as good as those obtained using the FA algorithm. However, the question remains open as to whether other algorithms could lead to different global maximum solutions for the considered (or other) RRAP criterion functions?

2. Problem denition

The topic being discussed concerns the optimization of a criterion function, represented as:

$$max\,(F_c(\mathbf{r}, \mathbf{n})), \tag{1}$$

within constraints:

$$g_y(\mathbf{r}, \mathbf{n}) \le b_y, \tag{2}$$

$$0 \le r \le 1,\ n \in Z^+,\ y = \{1, \ldots, k\},\ y \in Z^+,\ k \in Z^+, \tag{3}$$

where: $F_c(\mathbf{r}, \mathbf{n})$ is the system reliability function (Equation 1) that measures the overall reliability of the system based on the reliability vector $\mathbf{r}$ and the number of elements in the various subsystems $\mathbf{n}$, $\mathbf{n}$ is a vector representing the number of elements in each subsystem of the system being optimized, $\mathbf{r}$ is a vector representing the eliability of each element (Equation 2) in each subsystem of the system being optimized, g_y is a function that calculates the physical characteristics of an element in a subsystem for a specic constraint number y, b_y is an upper limit for the physical characteristics of an element in a subsystem for a specic constraint number y, and k is the number of constraints that must be satised in the optimization problem (Equation 3).

3. Models of the RRAP system

The criterion functions for the scenarios considered have been described in detail in the literature [2,5,29]. In addition to the cases discussed in the article, other test cases include an overspeed system for a gas turbine (Figure 1) and a more complex 15-unit system reliability problem with various parameter combinations [29].

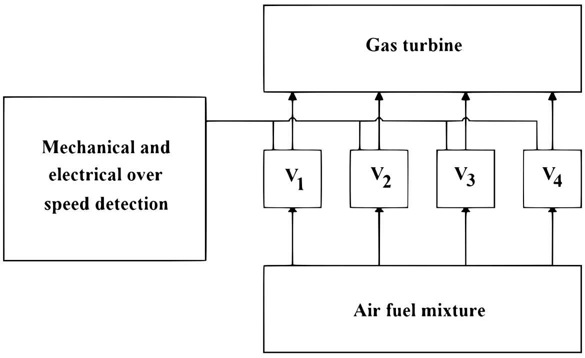

Figure 1. The schematic diagram of an overspeed system for a gas turbine [29,30]

3.1. Scenario 1 — bridge system

In the first scenario, the five-element bridge system (Figure 2) [2,5,29] was considered.

The criterion function for a bridge system has been extensively derived in the literature and can be formulated as follows:

$$F_{c1}(\mathbf{r},\mathbf{n}) = R_1 \cdot R_2 + R_3 \cdot R_4 + R_1 \cdot R_4 \cdot R_5 + R_2 \cdot R_3 \cdot R_5 - R_1 \cdot R_2 \cdot R_3 \cdot R_4 - R_1 \cdot R_2 \cdot R_3 \cdot R_5 + R_1 \cdot R_2 \cdot R_4 \cdot R_5 - R_1 \cdot R_3 \cdot R_4 \cdot R_5 - R_2 \cdot R_3 \cdot R_4 \cdot R_5 + 2 \cdot R_1 \cdot R_2 \cdot R_3 \cdot R_4 \cdot R_5, \quad (4)$$

where the individual reliability values R_i of each subsystem are calculated as:

$$R_i = 1 - (1 - r_i)^{n_i}, \ \forall\, i \in \{1, 2, \ldots, m_1\} \quad (5)$$

where m_1 represents the number of subsystems in the whole system in scenario 1 (Figure 2). In this case, m_1 is equal to 5.

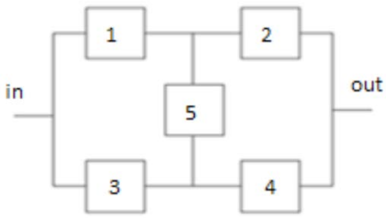

Figure 2. Diagram of the bridge system analyzed in scenario 1

For the bridge system, the optimization involves 10 decision variables, consisting of five variables r_i and five variables n_i with integer values. In scenario 1 (Figure 2), there are five subsystems in the whole system ($m_1 = 5$), and the individual reliability values (Equation 5) R_i of each subsystem are calculated using Equation (4).

Three constraints were introduced with $k_1 = 3$, which include total weight and volume (V), cost (C), and lifetime (T), as well as a constraint on the system weight (W) (Table 1).

$$g_1(\mathbf{r},\mathbf{n}) = \sum_{i=1}^{m_1} w_i \cdot v_i^2 \cdot n_i^2 - V \le 0, \quad (6)$$

$$g_2(\mathbf{r},\mathbf{n}) = \sum_{i=1}^{m_1} \alpha_i \cdot \left(-\frac{T}{ln(r_i)}\right)^{\beta_i} \cdot (n_i + e^{0.25 \cdot n_i}) - C \le 0, \quad (7)$$

$$g_3(\mathbf{r},\mathbf{n}) = \sum_{i=1}^{m_1} w_i \cdot n_i \cdot e^{0.25 \cdot n_i} - W \le 0, \quad (8)$$

$$i = \{1, \ldots, m_1\},\ 0 \le r_i \le 1,\ n_i \in Z^{+} \quad (9)$$

Table 1. Bridge system settings

β_i $i = \{1, \ldots, m_1\}$	V	C	W	T [h]
1.5	110	175	200	1000

In Equation (4), n_i represents the number of elements in the i-th subsystem, r_i represents the reliability of each element in the i-th subsystem, α_i and β_i represent the physical characteristics of the element in the i-th subsystem, and w_i, v_i, and c_i represent the weight, volume, and cost of the element in the i-th subsystem (Equations 6–9).

The parameter settings for the bridge system presented in Table 2 [5] were adopted to ensure the comparability of the results with other solutions reported in the literature. In addition to these parameter settings, other methods for system reliability analysis are also employed, such as path method optimization. The objective function for the path method optimization is given by:

$$F_{c2}(\mathbf{r},\mathbf{n}) = 1 - (1 - R_1 \cdot R_2) \cdot (1 - R_3 \cdot R_4) \cdot (1 - R_1 \cdot R_4 \cdot R_5) \cdot (1 - R_2 \cdot R_3 \cdot R_5), \quad (10)$$

or a cutting plane method, for which the criterion function can be expressed as:

$$F_{c3}(\mathbf{r},\mathbf{n}) = [1 - (1 - R_1) \cdot (1 - R_3)] \cdot [1 - (1 - R_2) \cdot (1 - R_4)] \cdot [1 - (1 - R_2 \cdot (1 - R_3) \cdot (1 - R_5)] \cdot [1 - (1 - R_1) \cdot (1 - R_4) \cdot (1 - R_5)]. \quad (11)$$

The reliability of the system, calculated using the minimum cut method (Equation 11), is always smaller than the value of reliability computed using the minimum path method (Equation 10). This difference between the two methods can be utilized for more precise optimization, is illustrated in Figure 3.

Table 2. Bridge system settings

Subsystem i	$10^5 \cdot \alpha_i$	$w_i \cdot v_i^2$	w_i
1	2.330	1	7
2	1.450	2	8
3	0.541	3	8
4	8.050	4	6
5	1.950	2	9

After analyzing the constraints used in the model, it becomes apparent that the second constraint ($g_2(r, n)$) (7) has a significant impact on the permissible range of component reliability values, as illustrated in Figure 4.

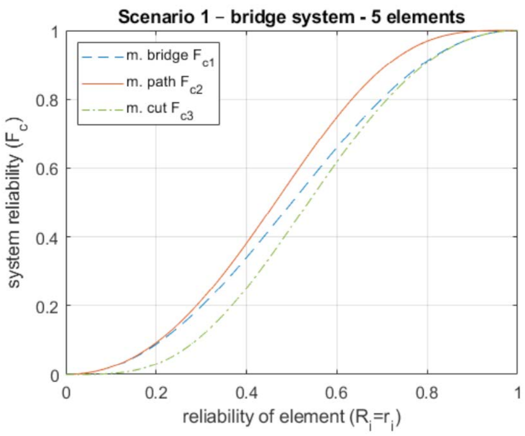

Figure 3. The system's reliability was evaluated using different analysis methods with a xed number of redundant elements ($n_i = 1$ for all i)

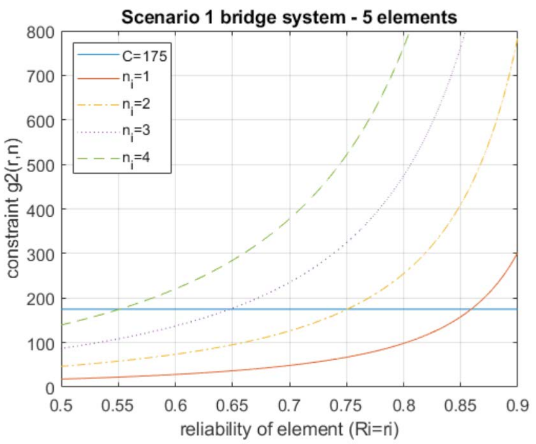

Figure 4. Analysis of the second constraint $g_2(r, n)$

3.2. Scenario 2 — a system consisting of 10 elements

The criterion function for the ten-element system (Figure 5) is more complicated than that for the bridge system [21,36]. It can be formulated as follows:

$$F_{c4}(\mathbf{r}, \mathbf{n}) = R_1 \cdot R_2 \cdot R_3 \cdot R_4 + (R_1 \cdot R_2 \cdot R_6 \cdot R_{10}) \cdot (Q_3 + R_3 \cdot Q_4) + + (R_1 \cdot R_5 \cdot R_9 \cdot R_{10}) \cdot (Q_2 + R_2 \cdot Q_3 \cdot Q6 + R_2 \cdot R_3 \cdot Q_4 \cdot Q_6) + R7 \cdot \cdot R_8 \cdot R_9 \cdot R_{10} \cdot (Q_1 + R_1 \cdot R_2 \cdot Q_3 \cdot Q5 \cdot Q_6 + R_1 \cdot R_3 \cdot Q_4 \cdot Q_5 \cdot Q_6) + + R_2 \cdot R_3 \cdot R_4 \cdot R_5 \cdot R_7 \cdot R_8 \cdot Q_1 \cdot (Q_9 + R_9 \cdot Q_{10}) + Q1 \cdot R_3 \cdot R_4 \cdot R_6 \cdot \cdot R_7 \cdot R8 \cdot Q_{10} \cdot (Q_2 + R_2 \cdot Q_5) + Q_1 \cdot Q2 \cdot R_3 \cdot R_4 \cdot R_6 \cdot R_7 \cdot R_8 \cdot R_9 \cdot \cdot Q_{10} + R_1 \cdot Q2 \cdot R_3 \cdot R_4 \cdot R_5 \cdot R_6 \cdot R_9 \cdot Q_{10} \cdot (Q_7 + R_7 \cdot Q_8) + Q1 \cdot \cdot R_2 \cdot R_5 \cdot R_6 \cdot R_7 \cdot R_8 \cdot Q_9 \cdot R_{10} \cdot (Q_3 + R_3 \cdot Q_4), \quad (12)$$

where R_i is defined as in expression (5), and Q_i is defined as:

$$Q_i = 1 - R_i, \forall i \in \{1, 2, ..., m_2\}. \quad (13)$$

where m_2 is the number of subsystems in the whole system ($m_2 = 10$) for scenario 2 (Figure 5), and the variables n_i, r_i have the same meaning as in scenario 1.

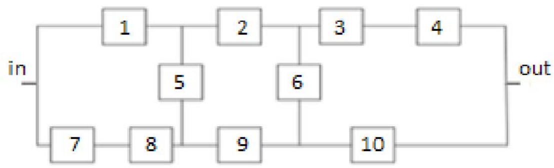

Figure 5. Diagram of the system consisting of 10 elements analyzed in scenario 2

For scenario 2, the constraints are expressed as:

$$g_y(\mathbf{n}) = \sum_{i=1}^{m_2} c_{y_i} \cdot n_i \leq b_y, \quad (14)$$

$$y = 1, 2, ..., k_2; \; n_i \in Z^+ \quad (15)$$

In scenario 2 (Equation 15) k_2 represents the number of constraints, which is assumed to be five.

Corrected sentence: The coffecients c_{yi} are randomly generated numbers from the interval [0, 100].

The coeffcient r is searched within the range $[lb_1, ub_1]$, and the parameter b_y is calculated as follows:

$$b_y = d_{rand} \cdot \sum_{i=1}^{m_2} c_{y_i}, \quad (16)$$

where: d_{rand} is a randomly generated number with a uniform distribution in the range (1.5, 3.5).

The values of these parameters, as used in the works of Yang et al. [6] and Kwiecien et al. [5], are taken from Tables 3 and 4 to compare the obtained solutions.

Table 3. Parameters used in scenario 2

i	c_{1i}	c_{2i}	c_{3i}	c_{4i}	c_{5i}
1	33.2468	35.6054	13.7848	44.1345	10.9891
2	27.5668	44.9520	96.7365	25.9855	68.0713
3	13.3800	28.6889	85.8783	19.2621	1.0164
4	0.4710	0.4922	63.0815	12.1687	29.4809
5	51.2555	39.6833	78.5364	23.9668	59.5441
6	82.9415	59.2294	11.8123	28.9889	46.5904
7	51.8804	78.4996	97.1872	47.8387	49.6226
8	77.9446	86.6633	45.0850	25.0545	59.2594
9	26.8835	7.8195	3.6722	76.9923	87.4070
10	85.8722	27.7460	55.3950	53.3007	55.3175

Table 4. Parameters used in scenario 2

Parameters	d_{rand}
c_{1i}	3.1250
c_{2i}	3.4710
c_{3i}	3.3247
c_{4i}	2.6236
c_{5i}	3.4288

4. Selected optimization algorithms

Currently, there are over 100 optimization algorithms (OPAs) that use different behaviors observed in the natural world of plants and animals [31,32]. The development of optimization methods is necessary because there is a lack of universal and effcient methods for searching for the global extrema of analyzed functions. Therefore, it is desirable to have familiarity with and utilize multiple optimization methods [33].

In this paper, two selected heuristic algorithms [34] were utilized to search for the global extrema of the criteria functions F_{c1} (Equality 4) and F_{c4} (Equality 12). It is worth noting that the Firefly Algorithm (FA) is known for its eciency in optimizing RRAP problems [7].

4.1. Firefly Algorithm FA

The Firefly Algorithm (FA) is an optimization algorithm developed by Xin-She Yang at Cambridge University in 2007 [2]. The concept of the FA is based on the behavior of fireflies towards the light source and their interaction through short, rhythmic bio-luminescence signals. The algorithm mimics the movement of fireflies and uses their natural behavior to nd optimal solutions.

The basic principle of the algorithm is that brighter fireflies attract other individuals, leading to a more effcient search for space exploration. The algorithm utilizes the difference in light intensity, which is proportional to the value of the criterion function F_{c1}, to find the optimal solution. One of the rules used in the Firefly Algorithm is that all fireflies are unisex, and the attractiveness (β) between them depends on the distance and the light absorption coeffcient (γ), which is expressed as:

$$\beta(d) = \beta_0 e^{-\gamma d^2} \tag{17}$$

where: β_0 denotes the attractiveness at distance $d = 0$.

In the solution space examined in this paper, the k-th step during which a rey with index i, located at position x_i, attempts to approach a 'more attractive' individual with index j, located at position x_j, can be expressed by the equation [6]:

$$x_i^k = x_i^{k-1} + \beta_0 \cdot e^{-\gamma \cdot d_{ij}^{k-1}} \cdot (x_j^{k-1} - x_i^{k-1}) + \alpha_{ff} \cdot (x_{rand} - 0.5), \tag{18}$$

where: x_{rand} is a random number with a uniform distribution in the range [0,1], d_{ij}^{k-1} represents the distance between fireflies with index i and j in the previous $(k - 1)$ step, β_0 denotes the attractiveness at distance $d = 0$, γ denotes the light absorption coeffcient, and α_{ff} is a randomization parameter.

The general structure of the Firefly Algorithm is as follows:

1. Initialize parameters: β_0, α_{ff}, stopping criteria – maximum number of iterations N_{iter}^{max} with repeat ing the calculations for each case $N_{repcalc}$ times, randomly generate an initial population of N_{ff} fireflies, dene objective functions $F_c(\mathbf{r}, \mathbf{n})$ with constraints $g(\mathbf{r}, \mathbf{n})$.
2. Compute the light intensity of each individual, value of the objective function $F_c(\mathbf{r}, \mathbf{n})$ expresses the light intensity of i–th firefly I_i – when the constraints are met, checked at the time of each position calculation x_i^k.
3. While the stopping criteria have not been met, is executed:

- compare all pairs of fireflies in terms of light intensity – if $I_j > I_i$ the firefly i moves in the direction of the firefly *j*;
- evaluate new solution **r**, **n**, determine the new value of the objective function $F_c(\mathbf{r}, \mathbf{n})$ with the constraints, update the light intensity.

4. If the stopping criterion (maximum number of iterations N_{iter}^{max})has been met, determine the best solution.

4.2 Cuckoo Search CS

The second algorithm used to search for solutions to the two RRAP problems is the Cuckoo Search (CS) algorithm [6]. This algorithm, proposed in 2009 by Xin-She Yang and Suash Deb [35], mimics the behavior of cuckoo birds at effcient search space exploration of their nests. The process of generating a new solution x_i^k for the *i*-th cuckoo/ nest by randomly selecting it and using Lévy flight can be expressed as follows:

$$x_i^k = x_i^{k-1} + \alpha_{CS} \oplus s, \tag{19}$$

where: $\oplus$ is point-to-point multiplication (entry-wise product of two vectors), k is the step number (i.e., the next iteration), x_i^k is the solution obtained in the k-th step for the i-th cuckoo, α_{CS} is a scale factor whose value depends on the size of the problem; s is the step length determined by a Lévy probability distribution, and $L(\lambda)$ denotes a Lévy flight step.

Here, $\alpha_{CS} > 0$ is the step size scaling factor, which should be related to the scales of the problem of interest. In most cases, we can use α_{CS} = O(L/10) or α_{CS} = O(L/100), where L is a characteristic scale of the problem of interest [36]. The exploitation mechanism of the *CS* algorithm is based on local search, and the exploration mechanism is based on Lévy flights, which are drawn from the Lévy probability distribution [6] and can be expressed by the formula:

$$L(s,\lambda) = \frac{\lambda \cdot \Gamma(1+\lambda) \cdot \sin\left(\frac{\pi \cdot \lambda}{2}\right)}{\pi} \cdot \frac{1}{|s|^{\lambda}}, s >> 0, 1 \le \lambda \le 3. \tag{20}$$

For the stable Lévy distribution with α = 0.5, the probability density function is given by:

$$L(s,\alpha,\delta,\gamma_l) = \sqrt{\frac{\gamma_l}{2\pi}} \frac{1}{(s-\delta)^{1+\alpha}} exp\left(-\frac{\gamma_l}{2(s-\delta)}\right), \alpha = 0.5, \delta < s < \infty, \tag{21}$$

where: α – first shape parameter $0 < \alpha \le 2$, δ – location parameter $-\infty < \delta < \infty$, γl-scale parameter $0 < \gamma l < \infty$, $\Gamma(\cdot)$ – the gamma function:

$$\Gamma(x) = \int_0^{\infty} e^{-t} \cdot t^{x-1} dt. \tag{22}$$

The step length (or size) can be calculated using Mantegna and Stanley's algorithm [37] as:

$$s = \frac{U}{|V|^{\frac{1}{\lambda}}}, \tag{23}$$

where:

$$U = N(0, \sigma_v^2) \cdot \sigma_u,\ V \sim N\,N(0, \sigma_v^2), \tag{24}$$

and:

$$\sigma_v = 1,\ \sigma_u = \left(\frac{\Gamma(1+\lambda)\cdot \sin\left(\frac{\pi\cdot\lambda}{2}\right)}{\Gamma(\frac{1+\lambda}{2})\cdot\lambda\cdot 2^{\frac{\lambda-1}{2}}}\right)^{\frac{1}{\lambda}}, \lambda = 1+\alpha, 0 < \alpha \le 2, \tag{25}$$

where: α – first shape parameter distribution.

Figure 6 shows an example implementation of a Lévy flight (visualization restricted to 3D space), with the parameters listed in Table 5. Upon analyzing the steps generated by the CS algorithm, it can be observed that among a large number of small steps, the algorithm occasionally performs large jumps known as Lévy flights. These jumps are named after the French mathematician Paul Pierre Lévy. A characteristic feature of the Lévy distribution [38] is the long 'tails' that occur for large values, unlike the Gaussian (Normal) distribution (Figure 7) [39]. The trajectory of a Lévy flight has a fractal dimension $d_f = \lambda$ [40–42].

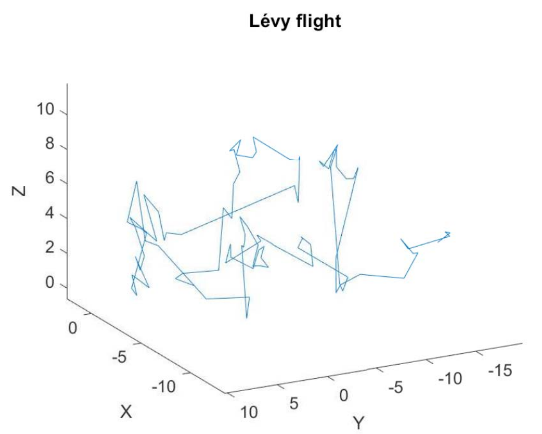

Figure 6. Visualization of an example Lévy flight (λ = 1.5) in Euclidean space R3

A search for the maximum of the criterion functions F_{c1} (4) and F_{c4} (12) was performed using the two selected algorithms. For each combination of the selected control parameters of the algorithm, $N_{repcalc}$ (the number of repetitions), calculations were performed (Table 5). The range of control parameters of the algorithms was chosen arbitrarily, limiting them to the most characteristic cases for the considered algorithm. In addition, the calculation parameters for the CS algorithm (the

Table 5. Parameters of both scenarios

Parameters	Values
Problem dimension	$N_D = m_{sc} \cdot 2$; $sc \in \{1, 2\}$; $m_1 = 5$, $m_2 = 10$
Maximum number of iterations	N_{iter}^{max} = 100 and 1,000 (CS)
Number of repetitions (each case)	$N_{repcalc}$ = 10 and 10,000 (CS)
Constraints values (lower bound, upper bound)	$lb_1 = 0.65$; $ub_1 = 0.85$; $lb_2 = 1$; $ub_2 = 4$
FA Firefly Algorithm	
Number of fireflies	$N_{ff} = 50$
Randomization parameter	$\alpha_{ff} = \{0.1, 0.2, \ldots, 1.0\}$
Reference factor of 'attractiveness'	$\beta_0 = \{0.1, 0.2, \ldots, 1.0\}$
Absorption coecient	$\gamma = \{0.01, 0.10, 1.00\}$
CS Cuckoo Search Algorithm	
Number of nests	$N_{nests} = 50$
Probability of detecting a cuckoo's egg	$p_a = \{0.25, 0.26, \ldots, 0.50\}$
Lévy distribution parameter	$\lambda = \{1.1, 1.2, \ldots, 1.9\}$
Positive step size scaling factor	$\alpha_{CS} = 1/100 \cdot N(0, 1) \cdot (x_i^{k-1} - x_i^{best})$, where x_i^{best} – the best solution

number of iterations N_{iter}^{max} and the number of repetitions $N_{repcalc}$) were chosen so that the calculation times were comparable.

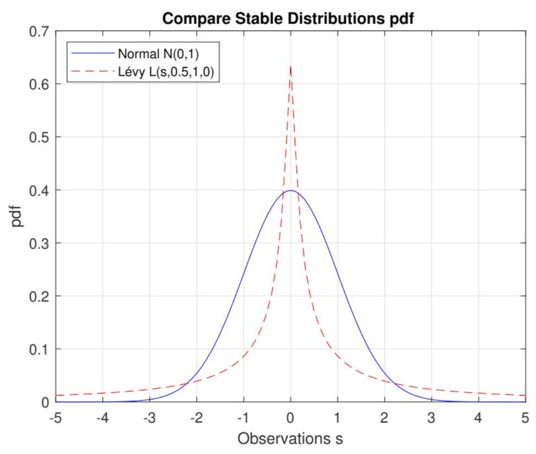

Figure 7. Compare Stable Distributions pdf

5. Calculation results

In Table 5, the control variables of the selected algorithms are presented, and results of calculations and solutions for scenario 1 and scenario 2 obtained with the two analyzed algorithms were obtained. The calculations for the CS algorithm [43] concerned changes in the probability of detecting an egg tossed by the cuckoo, p_a, and Lévy distribution parameter, λ.

For the FA algorithm [44], all combinations of the four parameters listed in Table 5 were considered: randomness, α_{ff}, reference 'attractiveness' factor, β_0, absorption coeffcient, γ, and population size, N.

To measure the improvement of the best solutions found by the FA algorithm in comparison with those given by CS, an improvement index is required. This index [29], which has been called Maximum Possible Improvement (MPI), is defined as follows:

$$MPI(\%) = \frac{F_c^{bestalg} - F_c^{alg}}{1 - F_c^{alg}}, \tag{26}$$

where: $F_c^{bestalg}$ – the best algorithm, in this case $F_c^{bestalg} = F_c^{FA}$; F_c^{alg} – the best system reliability for the objective function F_{c1} (Equation 4) and F_{c4} (Equation 12) obtained by the algorithm $alg \notin FA$, e.g., CS algorithm.

Let F_{c1}^{alg} and F_{c4}^{alg} be the best system reliability values obtained by an algorithm (alg) for criteria F_{c1} and F_{c4}, respectively. Let $\overline{F_{c1}^{alg}}$ and $\overline{F_{c4}^{alg}}$ be their respective means. Then, the correlation coefficient between the results (criterion function values) of the two scenarios obtained using the FA and CS algorithms can be calculated using the formula:

$$r^{alg} = \frac{\sum_{par1,par2,\ldots} \left(F_{c_1}^{alg} - \overline{F_{c_1}^{alg}}\right) \cdot \left(F_{c_4}^{alg} - \overline{F_{c_4}^{alg}}\right)}{\sqrt{\left(\sum_{par1,par2,\ldots} \left(F_{c_1}^{alg} - \overline{F_{c_1}^{alg}}\right)^2\right)} \cdot \sqrt{\left(\sum_{par1,par2,\ldots} \left(F_{c_4}^{alg} - \overline{F_{c_4}^{alg}}\right)^2\right)}}, \tag{27}$$

where N_{rep} is the number of repetitions, *par*1, *par*2, ... are the selected control parameters of the algorithms, r^{alg} is the resulting correlation coeffcient between the two criteria functions F_{c1} and F_{c4} obtained by the algorithm alg, $\overline{F_{c1}^{alg}}$ and $\overline{F_{c4}^{alg}}$ are the mean values of the criteria functions F_{c1} (Equation 4) and F_{c4} (Equation 12) obtained by the algorithm (*alg*), and *par*1, *par*2, ... are the parameters of the algorithms. For example, for the FA algorithm, these parameters could be α_{ff}, β_0, and γ, while for the CS algorithm, they could be p_a and λ.

We used the analysis of variance (ANOVA) test statistics to determine the statistically significant differences in average performance. Post hoc analysis using Tukey's honestly significant difference (HSD) test was conducted to investigate which of the parameters were signicantly different from each other for the tested algorithm.

The criterion and constraint values presented in the tables and gures are unitless.

5.1. Results for FA

For both scenarios, the best results were obtained using the parameter set of the FA algorithm: α_{ff} = 0.5, β_0 = 1.0, γ = 0.01 (Tables 6–9). For the criterion function F_{c1}, the maximum value obtained was 0.99995661 (Table 6). In scenario 2, the best solutions were obtained with F_{c4} = 0.99992902 (Table 8) using the FA algorithm.

Table 6. The best solution for scenario 1 u the FA

Parameters	α_{ff} = 0.5, β_0 = 1.0, γ = 0.01
Solutions F_c^{min}	0.99995653
Solutions F_c^{max}	**0.99995661**
Solutions F_c^{mean}	0.99995657
Solutions σ	2.27604463e–08
Calc. time [s] hline	216.0in

Table 7. The best solution for scenario 1 using the FA α_{ff} = 0.5, β_0 = 1.0, γ = 0.01

Parameters	Values
r_1	0.78571
r_2	0.8500
r_3	0.8500
r_4	0.7520
r_5	0.6601
n_1	4
n_2	4
n_3	3
n_4	2
n_5	3

Table 8. The best solution for scenario 2 was achieved using the FA algorithm

Parameters	α_{ff} = 0.5, β_0 = 1.0, γ = 0.01
Solutions F_c^{min}	0.99992644
Solutions F_c^{max}	**0.99992902**
Solutions F_c^{mean}	0.99992773
Solutions σ	9.85429365e–07
Calc. time [s]	189.3

Table 9. The best solution for scenario 2 was achieved using the FA algorithm with the parameter values α_{ff} = 0.5, β_0 = 1.0, γ = 0.01

Parameters	Values	Parameters	Values
r_1	0.8500	n_1	4
r_2	0.8500	n_2	4
r_3	0.8500	n_3	4
r_4	0.8500	n_4	4
r_5	0.8500	n_5	4
r_6	0.7429	n_6	1
r_7	0.7989	n_7	2
r_8	0.6626	n_8	2
r_9	0.6902	n_9	2
r_{10}	0.8295	n_{10}	2

By analyzing the parameter space using the post-hoc Tukey's honestly significant difference (HSD) test for ANOVA, we observed that the null hypothesis of no significant difference should be rejected for three cases when using the parameter set of α_{ff} = 0.4, β_0 = 0.8, and γ = 0.01 (Table 11).

We conducted a more detailed analysis of the parameter space and found that the most significant values of the criterion function in both scenarios were obtained with α_{ff} = 0.4, β_0 = 0.8, and γ = 0.01, with the exact maximum value of F_{c1} = 0.999956606987731. For the second scenario, the exact maximum value of the criterion function F_{c4} = 0.999929024736168 was reached in 28 cases (white boxes in Figure 9).

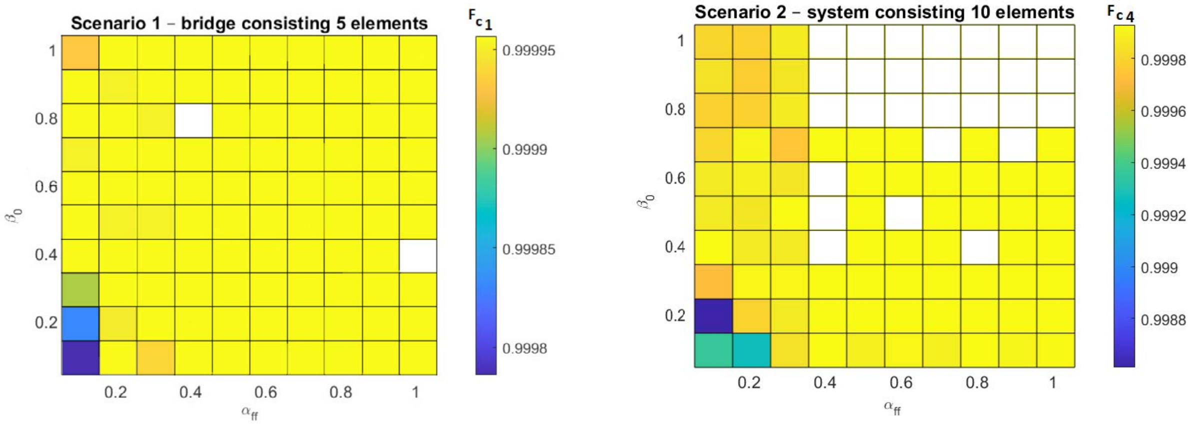

Figure 8. The solutions for both scenarios were obtained using the FA algorithm with the parameter γ set to 0.01

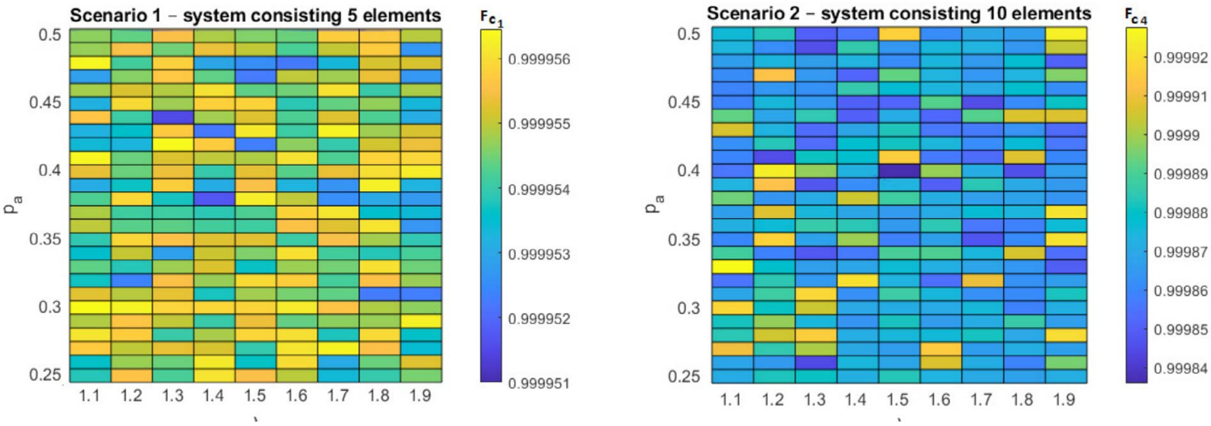

Figure 9. Solutions for scenario 1 and 2 has been achieved using CS algorithm

By reviewing the control parameter space of the FA algorithm (Figure 8), it is possible to observe a high compliance of the range of algorithm parameters leading to worse solutions. The highest values of correlation coeffcient r (27) were achieved for $\alpha_{ff} = 0.5$, $r^{FA} = 0.9912$, $\alpha_{ff} = 0.7$, $r^{FA} = 0.9881$, and $\alpha_{ff} = 0.9$, $r^{FA} = 0.9858$ (Figure 8).

5.2. Results for CS

Due to the variability of solutions obtained using the CS algorithm (as shown in Tables 7 and 13), we experimented with different stopping criteria and numbers of iterations. The values of p_a, λ, N_{iter}^{max}, and $N_{repcalc}$ were adjusted accordingly (see Table 5). In addition, we also tested the performance of the algorithm with larger values of N_{iter}^{max} and N_{repcal}, specically $N_{iter}^{max} = 1{,}000$ and $N_{repcalc} = 10{,}000$ (Table 14).

Due to the variability of solutions obtained using the CS algorithm (as shown in Tables 7 and 13), we experimented with different stopping criteria and numbers of iterations. The values of p_a, λ, N_{iter}^{max}, and $N_{repcalc}$ were adjusted accordingly (see Table 5). In addition, we also tested the performance of the algorithm with larger values of N_{iter}^{max}, and $N_{repcalc}$, specically $N_{iter}^{max} = 1{,}000$ and $N_{repcalc} = 10{,}000$ (Table 14).

Similarly, by analyzing the control parameter space of the CS algorithm (Figure 9), it is possible to obtain similar but lower-quality solutions (Table 14, Table 16). The highest values of correlation coeffcient r^{CS} (27) were achieved for $p_a = 0.39$, $r^{CS} = 0.7855$, $p_a = 0.25$, $r^{CS} = 0.7105$, and $p_a = 0.30$, $r^{CS} = 0.6264$.

For the rst scenario, the FA algorithm resulted in better solutions (Table 6) compared to the CS algorithm. However, the CS cuckoo algorithm was able to find a solution much faster (Table 12). All calculations were performed using the Matlab package R2020b with the Statistics and Machine Learning Toolbox on the Windows 10 Pro operating system and Intel® Core™ i5–7200U CPU 2.50GHz.

The solutions obtained using the CS algorithm diered from those obtained using the FA algorithm.

For instance, the number of elements ($n_3 \div n_5$) (Table 7 vs. Table 13) for scenario 1, the number of elements n_5 (Table 11 vs. Table 17) for scenario 2, and of course, the values of reliability.

In the investigated space of FA control parameters, difficulties were encountered in obtaining a solution that satisfies the assumed constraints in both scenarios for the set of $\alpha_{ff} = 0.1$, $\beta_0 = 1.0$, and $\gamma = 0.1$.

Additionally, for the second scenario, the set of $\alpha_{ff} = 0.1$, $\beta_0 = 0.5$, and $\gamma = 0.1$ was found to be problematic.

Table 10. Post-hoc Tuckey RIR tests for FA scenario 1 and $\alpha_{ff} = 0.4$, $\beta_0 = 0.8$, and $\gamma = 0.01$

Parameters	*p*-values
$\alpha_{ff} = 0.1, \beta_0 = 0.1, \gamma = 0.01$	0.000116
$\alpha_{ff} = 0.1, \beta_0 = 0.2, \gamma = 0.01$	**0.000116**
$\alpha_{ff} = 0.2, \beta_0 = 0.1, \gamma = 0.01$	0.000116
$\alpha_{ff} = 0.1, \beta_0 = 0.1, \gamma = 0.1$	0.000116
$\alpha_{ff} = 0.1, \beta_0 = 0.2, \gamma = 0.1$	0.000116
$\alpha_{ff} = 0.1, \beta_0 = 0.3, \gamma = 0.1$	0.000116
$\alpha_{ff} = 0.2, \beta_0 = 0.1, \gamma = 0.1$	0.000116
$\alpha_{ff} = 0.2, \beta_0 = 0.2, \gamma = 0.1$	0.000116
$\alpha_{ff} = 0.1, \beta_0 = 0.1, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.1, \beta_0 = 0.2, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.1, \beta_0 = 0.3, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.1, \beta_0 = 0.4, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.1, \beta_0 = 0.5, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.1, \beta_0 = 0.6, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.1, \beta_0 = 0.7, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.1, \beta_0 = 0.8, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.1, \beta_0 = 0.9, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.1, \beta_0 = 1.0, \gamma = 1.0$	0.000117
$\alpha_{ff} = 0.2, \beta_0 = 0.1, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.2, \beta_0 = 0.2, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.2, \beta_0 = 0.3, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.2, \beta_0 = 0.4, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.2, \beta_0 = 0.5, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.2, \beta_0 = 0.6, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.3, \beta_0 = 0.1, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.3, \beta_0 = 0.2, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.3, \beta_0 = 0.3, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.3, \beta_0 = 0.4, \gamma = 1.0$	0.014025
$\alpha_{ff} = 0.3, \beta_0 = 0.5, \gamma = 1.0$	0.000116
$\alpha_{ff} = 0.3, \beta_0 = 0.6, \gamma = 1.0$	0.000144
$\alpha_{ff} = 0.3, \beta_0 = 0.7, \gamma = 1.0$	0.000116

Table 11. Post-hoc Tukey RIR tests were performed for the FA algorithm in scenario 1 using the parameter set $\alpha_{ff} = 0.4$, $\beta_0 = 0.8$, and $\gamma = 0.01$

Parameters	*p*-values
$\alpha_{ff} = 0.1, \beta_0 = 0.1$	0.000047
$\alpha_{ff} = 0.2, \beta_0 = 0.1$	0.000047
$\alpha_{ff} = 0.1, \beta_0 = 0.2$	0.000047
$\alpha_{ff} = 0.1, \beta_0 = 0.3$	0.483894
$\alpha_{ff} = 0.3, \beta_0 = 0.7$	0.997808
$\alpha_{ff} = 0.1, \beta_0 = 0.8$	0.857372
$\alpha_{ff} = 0.2, \beta_0 = 0.1$	0.987661
$\alpha_{ff} = 0.3, \beta_0 = 0.1$	0.999464
$\alpha_{ff} = 0.1, \beta_0 = 0.9$	0.991139
$\alpha_{ff} = 0.2, \beta_0 = 0.9$	0.999796
$\alpha_{ff} = 0.3, \beta_0 = 0.9$	0.952106
$\alpha_{ff} = 0.1, \beta_0 = 1.0$	0.278482
$\alpha_{ff} = 0.2, \beta_0 = 1.0$	0.832396
$\alpha_{ff} = 0.3, \beta_0 = 1.0$	0.999918
Other cases	1.000000

Table 12. The best solution for scenario 1 by CS

Parameters	$p_a = 0.27$, $\gamma = 1.5$, $N_{iter}^{max} = 100$, $N_{repcalc} = 10$
Solutions F_c^{min}	0.99972966
Solutions F_c^{max}	**0.99994780**
Solutions F_c^{mean}	0.99984451
Solutions σ	7.40185829e–05
Calc. time [s]	1.7
MPI (%)	0.1688
$\|\|r^{FA} - r^{CS}\|\|_2$	0.1300
$\|\|n^{FA} - n^{CS}\|\|_2$	1.7321

Table 13. The best solution for scenario 1 using CS is $p_a = 0.27$, $\lambda = 1.5$, $N_{iter}^{max} = 100$, and $N_{repcalc} = 10$

Parameters	Values
r_1	0.8298
r_2	0.8380
r_3	0.8500
r_4	0.6500
r_5	0.7265
n_1	4
n_2	4
n_3	2
n_4	3
n_5	2

The study aimed to investigate the effect of changing the lambda parameter (λ) of the Lévy distribution on the obtained solutions with an increased number of iterations. For this purpose, Table 5 was utilized.

In scenario 1, increasing the lambda parameter and the number of iterations (i.e., N_{iter}^{max} and $N_{repcalc}$) resulted in the highest value of the criterion function for the CS algorithm. The optimal parameters were $p_a = 0.27$, $\lambda = 1.5$, $N_{iter}^{max} = 100$, and $N_{repcalc} = 10$ (Table 12), and $p_a = 0.30$, $\lambda = 1.1$ for $N_{iter}^{max} = 1{,}000$ and $N_{repcalc} = 10{,}000$ (Table 15). The increase in the number of iterations not only improved the value of the criterion function but also reduced the Euclidean difference between the obtained solutions of the FA and CS algorithms. In scenario 2, the highest value of the criterion function was obtained for the CS algorithm with $p_a = 0.33$ and $\lambda = 1.1$ (Table 17). Although the difference in MPI was small, the solutions obtained for the sought variables diered (Table 11, Table 17).

Table 14. The best solution for scenario 1 using CS algorithm

Parameters	$p_a = 0.30$, $\gamma = 1.1$, $N_{iter}^{max} = 1{,}000$, $N_{repcalc} = 10{,}000$
Solutions F_c^{min}	0.99921071
Solutions F_c^{max}	**0.99995645**
Solutions F_c^{mean}	0.99984281
Solutions σ	6.41364325e–05
Calc. time [s]	346.4
MPI (%)	0.0037
$\|\|\boldsymbol{r}^{FA} - \boldsymbol{r}^{CS}\|\|_2$	0.0124
$\|\|\boldsymbol{n}^{FA} - \boldsymbol{n}^{CS}\|\|_2$	0

Table 15. The best solution for scenario 1 using CS algorithm was achieved with the parameter settings of $p_a = 0.30$, $\lambda = 1.1$, $N_{iter}^{max} = 1{,}000$, and $N_{repcalc} = 10{,}000$

Parameters	Values
r_1	0.7919
r_2	0.8500
r_3	0.8499
r_4	0.7483
r_5	0.6500
n_1	4
n_2	4
n_3	3
n_4	2
n_5	3

Table 16. The best solution for scenario 2 using CS algorithm

Parameters	$p_a = 0.33$, $\gamma = 1.1$, $N_{iter}^{max} = 1{,}000$, $N_{repcalc} = 10{,}000$
Solutions F_c^{min}	0.99550799
Solutions F_c^{max}	**0.99992758**
Solutions F_c^{mean}	0.99892402
Solutions σ	5.51629947e–04
Calc. time [s]	387.5
MPI (%)	0.0198
$\|\|\boldsymbol{r}^{FA} - \boldsymbol{r}^{CS}\|\|_2$	0.1893
$\|\|\boldsymbol{n}^{FA} - \boldsymbol{n}^{CS}\|\|_2$	1

Table 17. The best solution for scenario 2 using CS algorithm was achieved with the parameter settings of $p_a = 0.33$, $\lambda = 1.1$, $N_{iter}^{max} = 1{,}000$, and $N_{repcalc} = 10{,}000$

Parameters	Values	Parameters	Values
r_1	0.8500	n_1	4
r_2	0.8500	n_2	4
r_3	0.8500	n_3	4
r_4	0.8500	n_4	4
r_5	0.8500	n_5	3
r_6	0.7671	n_6	1
r_7	0.8453	n_7	2
r_8	0.8432	n_8	2
r_9	0.6973	n_9	2
r_{10}	0.8500	n_{10}	2

6. Conclusions

The presented results contribute to the ongoing discussion on using nature-inspired algorithms for solving RRAP problems. Based on the two test scenarios described, including the Firefly Algorithm (FA), these algorithms can be considered appropriate tools for validating other optimization algorithms in RRAP problems.

Although the CS algorithm is known for its eectiveness due to its use of Lévy flight (Figure 6), the FA algorithm was found to be more effective in the considered parameter range. The use of the FA algorithm led to solutions with a higher value of the criterion function (Table 6, Table 8).

It is worth noting that the best solutions using CS were achieved for λ values different from the default value of 1.5 (with $\alpha = 0.5$) for the Lévy stable distribution used.

In addition to comparing the values of the criterion function, the MPI (Eqation 26), and the Euclidean distance dierences of both the reliability ($|\boldsymbol{r}^{FA} - \boldsymbol{r}^{CS}|_2$) and the number of redundant elements ($|\boldsymbol{n}^{FA} - \boldsymbol{n}^{CS}|_2$) obtained by carefully selecting the parameters of the CS algorithm can also be compared.

The comparison of the criterion function values, the linear r-Pearson correlation coeffcient,and the data from the post-hoc RIR Tukey test led to the selection of the same (or similar) control parameters for the analyzed algorithms. Therefore, the presented analysis methods can also be used to compare other optimization algorithms.

Such an approach can extend the application of well-known test function benchmarks for global RRAP optimization. For example, test functions such as Michalewicz's, Rosenbrock's, De Jong's, Schwefel's, Ackley's, Rastring's, Easom's, Griewank's, Shubert's [35,45], Bohachevsky's, Matyas's, Zakharov's, and Goldstein-Price's [46], as well as other functions [6,47] or Tallard's test functions [48], can benefit from the proposed analysis methods.

References

[1] Sobczak W. Podstawy probabilistyczne teorii systemów informacyjnych. Warszawa: Wydawnictwa Naukowo-Techniczne; 1981.

[2] Yang X-S. Multiobjective firefly algorithm for continuous optimization. Engineering with Computers. 2013;29:175–184. https://doi.org/10.1007/s00366-012-0254-1.

[3] Klempka R, Filipowicz B. Comparison of using the genetic algorithm and cuckoo search for multicriteria optimisation with limitation. Turkish Journal of Electrical Engineering and Computer Sciences. 2017;25:1300–1310. https://doi.org/10.3906/elk-1511-252.

[4] Kwiecień J. Algorytmy stadne w rozwiązywaniu wybranych zagadnień optymalizacji dyskretnej i kombinatorycznej. Kraków: Wydawnictwa AGH; 2015.

[5] Kwiecień J, Filipowicz B. Optymalizacja niezawodności złożonych systemów za pomocą algorytmu świetlika. Eksploatacja i Niezawodność. 2017;19(2):296–301. https://doi.org/10.17531/ein.2017.2.18.

[6] Yang X-S. Nature-Inspired Optimization Algorithms. Cham: Elsevier; 2014. https://doi.org/10.1016/C2013-0-01368-0.

[7] Fuksa AK. Zastosowanie sztucznej inteligencji w optymalizacji niezawodnościowej systemów. [doctoral dissertation]. Kraków: Akademia Górniczo-Hutnicza im. H. Kołłątaja; 2017.

[8] Kuo S-Y, Lu S-K, Yeh F-M. Determining terminal-pair reliability based on edge expansion diagrams using OBDD. IEEE Transactions on Reliability. 1999;48(3):234–246. https://doi.org/10.1109/24.799845.

[9] Yeh F-M, Lu S-K, Kuo S-Y. OBDD-based evaluation of k-terminal network reliability. IEEE Transactions on Reliability. 2002;51(4):443–451. https://doi.org/10.1109/TR.2002.804736.

[10] Kim H-G, Bae C-O, Park D-J. Reliability–redundancy optimization using simulated annealing algorithms. Journal of Quality in Maintenance Engineering. 2006;12(4):354–363. https://doi.org/10.1108/13552510610705928.

[11] dos Santos Coelho L. An effcient particle swarm approach for mixed-integer programming in reliability–redundancy optimization applications. Reliability Engineering and System Safety. 2009;94(4):830–837. https://doi.org/10.1016/j.ress.2008.09.001.

[12] Liu Y, Qin G. A modified particle swarm optimization algorithm for reliability redundancy optimization problem. Journal of Computers. 2014;9(9):2124–2131.

[13] Yeh W-C, Hsieh T-J. Solving reliability redundancy allocation problems using an articial bee colony algorithm. Computers and Operations Research. 2011;38(11):1465–1473. https://doi.org/10.1016/j.cor.2010.10.028.

[14] Kanagaraj G, Ponnambalam SG, Jawahar N. A hybrid cuckoo search and genetic algorithm for reliability–redundancy allocation problems. Computers and Industrial Engineering. 2013,66(4):1115–1124. https://doi.org/10.1016/j.cie.2013.08.003.

[15] Liu Y. Improved bat algorithm for reliability–redundancy allocation problems. International Journal of Security and Its Applications. 2016;10(2):1–12. http://dx.doi.org/10.14257/ijsia.2016.10.2.01.

[16] Agarwa M, Sharm VK. Ant colony approach to constrained redundancy optimization in binary systems. Applied Mathematical Modelling. 2010;34(4):992–1003. https://doi.org/10.1016/j.apm.2009.07.016.

[17] Mellal MA, Zio E. System reliability–redundancy optimization with cold-standby strategy by an enhanced

nest cuckoo optimization algorithm. Reliability Engineering and System Safety. 2020;201:106973. https://doi.org/10.1016/j.ress.2020.106973.

[18] Rakhi K, Pahuja GL. Solving reliability redundancy allocation problem using grey wolf optimization algorithm. Journal of Physics: Conference Series. 2020;1706:012155. https://doi.org/10.1088/1742-6596/1706/1/012155.

[19] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Advances in Engineering Software. 2014;69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007.

[20] Ardakan M, Rezvan M. Multi-objective optimization of reliability–redundancy allocation problem with cold-standby strategy using NSGA-II. Reliability Engineering and System Safety. 2018;172:225–238. https://doi.org/10.1016/j.ress.2017.12.019.

[21] Chen T. IAs based approach for reliability redundancy allocation problems. Applied Mathematics and Computation. 2006;182(2):1556–1567. https://doi.org/10.1016/j.amc.2006.05.044.

[22] Harish G. An approach for solving constrained reliability–redundancy allocation problems using cuckoo search algorithm, Beni-Suef University Journal of Basic and Applied Sciences. 2015;4(1):14–25. https://doi.org/10.1016/j.bjbas.2015.02.003.

[23] Hikita M, Nakagawa Y, Nakashima K, Narihisa H. Reliability optimization of system by a surrogate-constraints algorithm. IEEE Transactions on Reliability. 1992;41(3):473–480. https://doi.org/10.1109/24.159825.

[24] Hsieh Y-C, Chen T-C, Bricker DL. Genetic algorithms for reliability design problems. Microelectronics Reliability. 1998;38(10):1599–1605. https://doi.org/10.1016/S0026-2714(98)00028-6.

[25] Liu Z, Chen J-H, Tan S-Y, Yeh W-C. A novel simplied swarm optimization for generalized reliability redundancy allocation problem. arXiv:2110.00133; 2021. https://doi.org/10.48550/arXiv.2110.00133.

[26] Saleem E, Dao T-M, Liu Z. Multiple-objective optimization and design of series-parallel systems using novel hybrid genetic algorithm meta-heuristic approach. World Journal of Engineering and Technology. 2018;6:532–555. https://doi.org/10.4236/wjet.2018.63032.

[27] Wu P, Gao L, Zou D, Li S. An improved particle swarm optimization algorithm for reliability problems. ISA Transactions. 2011;50(1):71–81. https://doi.org/10.1016/j.isatra.2010.08.005.

[28] Marouani H, Al-mutiri O. Optimization of reliability redundancy allocation problems: A review of the evolutionary algorithms. Computers, Materials and Continua. 2022;71(1):537–571. https://doi.org/10.32604/cmc.2022.020098.

[29] Valian E. Solving reliability optimization problems. In: Yang X-S, editor. Cuckoo Search and Firefly Algorithm Theory and Applications. London: Springer; 2014, p. 195–216.

[30] Ghasemi M, Rahimnejad A, Hemmati R, Akbari E, Gadsden SA. Wild Geese Algorithm: A novel algorithm for large scale optimization based on the natural life and death of wild geese. Array. 2021;11:100074. https://doi.org/10.1016/j.array.2021.100074.

[31] Devarapalli R, Kumar V. Power system oscillation damping controller design: A novel approach of integrated HHO-PSO algorithm. Archives of Control Sciences. 2021;31(67):553–591. https://doi.org/10.24425/acs.2021.138692.

[32] Pijarski P. Optymalizacja heurystyczna w ocenie warunków pracy i planowania rozwoju systemu elektroenergetycznego. Lublin: Politechnika Lubelska; 2019.

[33] Kusiak J, Danielewska-Tułecka A, Oprocha P. Optymalizacja: wybrane metody z przykładami zastosowań. Warszawa: Wydawnictwo Naukowe PWN; 2009.

[34] Filipowicz B, Kwiecień J. Algorytmy stadne w problemach optymalizacji. Pomiary Automatyka, Robotyka. 2011;12:152–157.

[35] Yang X-S, Deb S. Cuckoo search via Lévy flight. In: Abraham A, Carvalho A, Herrera F, Pai V, editors. 2009 World Congress on Nature & Biologically Inspired Computing, 9–11 December 2009, Coimbatore, India: Proceedings. IEEE; 2009, p. 210–214. https://doi.org/10.1109/NABIC.2009.5393690.

[36] Vázquez R, Sandoval G, Ambrosio J, B., How to generate the input current for exciting spiking neural model using the cuckoo search algorithm. In: Yang XS, editor. Cuckoo Search and Firefly Algorithm. Studies in Computational Intelligence. Vol. 516. Cham: Springer; 2014, p. 155–178. https://doi.org/10.1007/978-3-319-02141-6_8.

[37] Manteng RN. Fast, accurate algorithm for numerical simulation of Lévy stable stochastic process. Physical Review E. 1994;49:46774683. https://doi.org/10.1103/PhysRevE.49.4677.

[38] Viswanathan GM, Afanasyev V, Buldyrev SV, Havlin S, da Luz MGE, Raposo E, Stanley HE. Lévy flights in random searches. Physica A: Statistical Mechanics and its Applications. 2000;282(1–2):1–12. https://doi.org/10.1016/S0378-4371(00)00071-6.

[39] Nolan J. Stable Distributions: Models for Heavy-Tailed Data. New York: Springer; 2016.

[40] Bovet A. An introduction to non-diusive transport models. ArXiv e-prints. 2015. https://doi.org/10.48550/arXiv.1508.01879.

[41] Chechkin AV, Metzler R, Klafter J, Gonchar VY. Introduction to the theory of Lévy flights. In: Klages R, Radons G, Sokolov IM, editors. Anomalous Transport: Foundations and Applications. Chichester: Wiley; 2008, p. 129–162. https://doi.org/10.1002/9783527622979.ch5.

[42] Hughes BD. Random Walks and Random Environments. Oxford: Clarendon Press; 1995.

[43] Yang X-S. Cuckoo Search (CS) Algorithm Version 1.3.0.0. MathWorks. File Exchange. [Internet] 2022. Available from: https://www.mathworks.com/matlabcentral/fileexchange/29809-cuckoo-search-cs-algorithm.

[44] Yang X-S. Firefly Algorithm Version 1.2.0.0. MathWorks. File Exchange. [Internet] 2021. Available from: https://www.mathworks.com/matlabcentral/fileexchange/29693-firefly-algorithm.

[45] Roy S, Chaudhuri SS. Cuckoo search algorithm using Lévy flight: A review. International Journal of Modern Education and Computer Science. 2013;12:10–15. https://doi.org/10.5815/ijmecs.2013.12.02.

[46] Mareli M, Twala B. An adaptive Cuckoo search algorithm for optimisation. Applied Computing and Informatics. 2018;14(2):107–115. https://doi.org/10.1016/j.aci.2017.09.001.

[47] Surjanovic S, Bingham D. Virtual library of simulation experiments: Test functions and datasets [Internet]. 2023 [cited 2023 Nov 15]. Available from: https://www.sfu.ca/%7Essurjano/index.html.

[48] Taillard E. Benchmarks for basic scheduling problems. European Journal of Operational Research. 1993;64(2):278–285. https://doi.org/10.1016/0377-2217(93)90182-M.