Otimização Não-Linear em Engenharia | MEM.2024
(artigo/relatório de projecto)

Resolution of optimization's problems using the Butterfly Optimization Algorithm

Maximizing coverage and reducing communication latency in a satellites system and maximizing profits in online services

Ilaria Gallo · Remo Irtuso

Submitted: data

Summary This paper explores three distinct optimization challenges using the Butterfly Optimization Algorithm. The first optimizes satellite data transmission to reduce latency and increase coverage. The second manages server room requests to maximize profits. The third optimizes a ten-bar truss to minimize mass while maintaining structural integrity. These projects demonstrate the algorithm's effectiveness in diverse engineering applications.

Key-Words Butterfly Optimization Algorithm · Metaheuristics · Satellites system · Online services · Optimization

1 Introduction

In recent years, the importance of data has grown exponentially in every field. However, in addition to collecting and analyzing data, optimizing their transmission and processing is crucial. The two proposed projects focus on these emerging challenges. The first project aims to optimize data transmission through MEO (Medium Earth Orbit) satellites. The goal is to reduce latency in communications between the satellite and ground stations, while also maximizing the coverage area that the satellites' signals reach on Earth.

The second project focuses on efficiently managing user requests to a specific server room. Given the enormous amounts of data to be processed in modern times, it is essential to allocate resources efficiently and trying to handle as much request as possible. The goal is to maximize profits related to user requests, thus trying to reduce losses associated with the inability to handle requests or excessively long waiting times.

The third problem is a Benchmark that focuses on optimizing a ten-bar truss to minimize its mass while maintaining mechanical integrity and structural constraints. The optimization includes two approaches: the optimal sizing of the cross-sectional areas of the bars and the modification of the node positions.

Ilaria Gallo
n.º 122338
E-mail: ilaria.gallo@ua.pt

Remo Irtuso
n.º 122422
E-mail: remo.irtuso@ua.pt

2 Definition of engineering problems

2.1 Maximizing coverage and reducing communication latency in a satellites system

In the context of modern communications, optimizing data transmission via satellites is of fundamental importance, since latency represents a significant obstacle in the transmission of information in real time. Furthermore, it is also necessary to consider the coverage problem, in order to guarantee that the satellite signal covers a specific geographical area. To achieve the goal, it is essential to determine the optimal distance that minimizes data transmission time while simultaneously maximizing area coverage. The fundamental data needed for the analysis includes the data transmission speed in space V_0, the earth's radius R and an accurate measurement of the distance. As regards the latter, the Euclidean distance has been choosen.

This study was created with the aim of carrying out a preliminary analysis, in order to choose the optimal initial configuration for the satellite communication system.

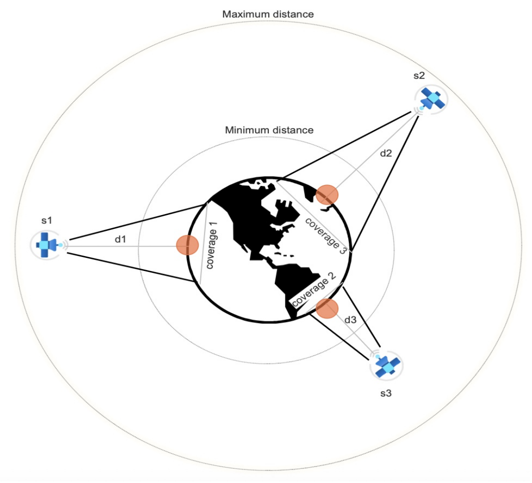

Figure 1. Satellites system composed by 3 satellites and 3 ground stations based on [4].

To describe the data more compactly, two matrices, which represent the coordinates of the satellites and ground stations, have been introduced:

$$\mathbf{Cs} = \begin{bmatrix} xs_1 & ys_1 \\ xs_2 & ys_2 \\ \cdots & \cdots \\ xs_N & ys_N \end{bmatrix} ; \ \mathbf{Ce} = \begin{bmatrix} xe_1 & ye_1 \\ xe_2 & ye_2 \\ \cdots & \cdots \\ xe_M & ye_M \end{bmatrix} .$$

Cs is a $N \times 2$ matrix. Every row of ***Cs*** represent a satellite and, for each one, we have its coordinates. ***Ce*** has the same structure of ***Cs***, but it represents the earth station's coordinates.

A binary variable, a_{ij}, with $i = 1, \cdots, N$ and $j = 1, \cdots, M$, was introduced to indicate the active connections between a satellite and an earth station. If the variable a_{ij} is 1, then there is communication between satellite i and earth station j, 0 otherwise. So the matrix $\mathbf{A}$ is of size $N \times M$. The fixed data are:

- $V_0 = 299792.458$ km/h
- $R = 6.371$ km
- $d_{\text{min}} = 2.000$ km
- $d_{\text{max}} = 36.000$ km
- $-8371+R \leq x_s \leq 8371+R, -8371+R \leq y_s \leq 8371+R$
- $T_{\text{max}} = 15\text{e-}2$ s

2.2 Maximizing online service profits

The main task of servers is to handle numerous requests from clients. However, an excess of requests can pose the risk of augmenting costs for the company. To address the issue, were examined various categories of requests, denoted by N. Each request presents distinctive specifications, such as required resources, execution time, and especially the gain associated with its completion.

We can conceptualize a server as a binary matrix of dimensions $N \times T$, where T represents the duration of the considered time interval, and where the value 1 indicates that request i was assigned to the server at time t, 0 otherwise. The logic of this process is graphically illustrated in Figure 2. This leads us to obtain a hyper-matrix $\boldsymbol{X}$ of dimensions $N \times T \times S$, where S is the number of servers, which is the decision variable.

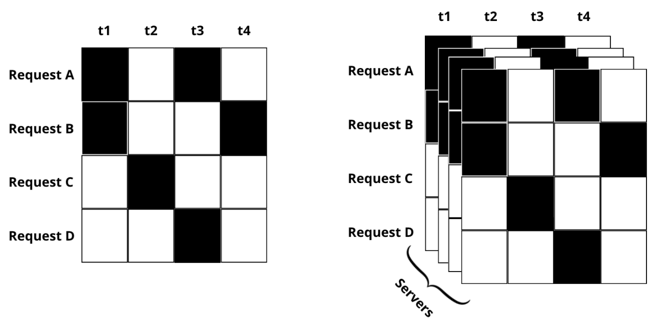

Figure 2. On the left single server representation with 4 different request and 4 different time instants, on the right illustration of the allocation hyper-matrix $\boldsymbol{X}$ with 4 servers.

3 Formulation of the optimization problems

3.1 Maximizing coverage and reducing communication latency in a satellites system

The objective function that describes the problem is composed of two terms, the first related to the calculation of latency, and the second related to the calculation of the satellite coverage. The formulation of the optimization problem is given below:

Find $\mathbf{Cs}$ to:

minimize $f(\mathbf{Cs}) =$

$$\sum_{i=1}^{\mathrm{N}} \sum_{j=1}^{M} w_1 \frac{d(Cs_i,Ce_j)}{V_0 \cdot T_{max}} + w_2 \left(1 - \frac{50\%}{1 + \frac{R}{d(Cs_i,Ce_j)}} \right) \mathrm{a}_{ij}$$

subject to

$$d_{\text{min}} \leq d(Cs_i,Ce_j) \leq d_{\text{max}} \quad \forall i,j$$

$$\sum_{j=1}^{M} \mathrm{a}_{ij} \geq 1, \quad i = 1,...,N$$

$$\sum_{i=1}^{N} \mathrm{a}_{ij} \geq 1, \quad j = 1,...,M$$

$$\text{coordinate}_{s,\text{min}} \leq Cs_{ij} \leq \text{coordinate}_{s,\text{max}} \text{ , with } i = 1, \cdots, N \quad j = 1, \cdots, M$$

with:

$$d(Cs_i,Ce_j) = \sqrt{(Cs_{i1} - Ce_{j1})^2 + (Cs_{i2} - Ce_{j2})^2}$$

which is the euclidean distance between the satellite i and earth station j. In the formulation, it's noteworthy that $T_{\max}$ represents the maximum latency of MEO satellites [3] and serves a normalization function, ensuring that both components of the objective function are dimensionless.

This optimization problem is a non-linear configuration problem due to the non-linearity of the objective function terms, particularly involving square root and division operations. The constraints, in addition to being non-linear, include inequality constraints. The variables of the problem, that are the coordinates of the satellites, are continuous.

3.2 Maximizing online service profits

The objective function of the optimization problem, along with its constraints, is defined as follows:

Find $\mathbf{X}$ to:

maximize $f(\mathbf{X}) =$

$$\sum_{i=0}^{N}\sum_{t=0}^{T}\sum_{j=0}^{\mathrm{S}} price_i x_{itj}$$

$$-\sum_{i=0}^{N}\sum_{t=0}^{T} Q_{\text{cost},i} \cdot q_{it}(X) - \sum_{i=0}^{\mathrm{N}}\sum_{t=0}^{\mathrm{T}} R_{\text{cost},i} \cdot \mathrm{r}_{it}(X)$$

subject to

$$\sum_{i=0}^{N} cores_i \cdot x_{i,t,j} \leq cores_{server},\ \forall j \in S,\ \forall t \in T.$$

$$0 \leq \mathrm{q}_t(X) \leq capacity_{queue},\ \forall t \in T.$$

$$\mathrm{q}_{i,t}(X) = \mathrm{q}_{i,t-1}(X) + \mathrm{R}_{it} - \sum_j x_{itj},\ \forall i \in \mathrm{N},\ \forall j \in \mathrm{S},\ \forall t \in \mathrm{T}$$

$$\mathrm{r}_{it} \geq 0\ i = 0,\cdots,\mathrm{N}\ \ t = 0,\cdots,\mathrm{T}.$$

$$x_{itj} \in \{0,1\}\ \forall i \in \mathrm{N},\ \forall j \in \mathrm{S},\ \forall t \in \mathrm{T}$$

with:

$$q_{it}(X) = \sum_{i=1}^{\mathrm{N}} \max\left(0,\mathrm{Req}_{it} - \sum_j x_{itj}\right)$$

and:

$$r_{it}(X) = \sum_{i=1}^{\mathrm{N}} \max\left(0,\mathrm{Req}_{it} - \left(\sum_j x_{itj} + q_{it}(X)\right)\right)$$

The function $\mathrm{q}_{it}(X)$ calculates the number of type i requests that are in the queue. Instead, the function $\mathrm{r}_{it}(X)$ determines the number of requests that must be rejected. A cost is associated with both functions, which will affects the final profit. The term Req_{it} represents the number of type i requests arriving at time instant t. The constraints state that the utilized resources, as well as the queued requests, cannot exceed a certain limit. Additionally, the queue must be consistent across consecutive time instants.

Finally, it can be said that this problem intrinsically falls within the field of resource allocation optimization problems.

4 Sensitivity Analysis

4.1 Maximizing coverage and reducing communication latency in a satellites system

In this study, a sensitivity analysis was conducted to investigate how variations in the parameters, the number of satellites N, the number of ground stations M, and the weights *w1* and *w2*, impact the objective function of the satellite network. Several scenarios were generated with varying combinations of these parameters.

The sensitivity analysis revealed that increasing the number of satellites or the number of ground stations generally leads to higher values and greater variability in the objective function, indicating increased network complexity. This suggests that larger configurations require careful parameter balancing to manage complexity effectively.

Furthermore, the weights assigned significantly affect the objective function. The parameters that most influence the objective function are therefore the weights, whose modification causes a drastic change in the shape of the function. In particular, if one of the weights is at its maximum value and the other is zero, the function exhibits a linear trend. Conversely, if the weights are balanced, the trend becomes quadratic.

Additionally, an analysis was performed to examine how the objective function varies with respect to the coordinates of the satellites, keeping the weights *w1* and *w2* equal to 0.5. It was observed that both the x and y coordinates have a similar impact on the objective function. The function achieves the best values when the coordinates reach an intermediate position between the minimum and maximum distance achievable by the satellite, as expected.

In summary, this sensitivity analysis highlights the necessity of precise parameter adjustment and network configuration to optimize the performance of a complex satellite system. Balancing the number of satellites, ground stations, the weighting and the satellites positions is crucial for effective network management.

4.2 Maximizing online service profits

For the second problem, the analyzed parameters include queue capacity, the number of servers, and time duration. In order to do this, different hyper-matrix $\boldsymbol{X}$ configurations were generated to better understand how various assignments impact the system.

The manipulation of these input parameters has led to both increases and decreases in profit across the analyzed scenarios. An increase in queue capacity showed consistent profit improvements across all scenarios by reducing the number of rejected requests and associated costs. This highlights the significant impact of queue capacity on system performance. However, it also reveals that merely increasing capacity without strategic resource allocation does not fully optimize performance. Similar conclusions can be drawn when analyzing the number of servers and the considered time interval.

Overall, the analysis highlights that the effective configuration of the binary matrix $\boldsymbol{X}$ plays a significant role in determining the system's performance under different scenarios. This emphasizes the need for strategic resource allocation in online service optimization.

5 Evaluation

To program the two problems Python was used. In each case, the methodology involved assigning values to the decision variables, followed by modeling the problem, which included evaluating the objective function and constraints. To handle the constraints, external penalty functions were used, which penalized the objective function when the constraints were not satisfied. After calculating the final value of the objective function, an automatic generation of graphical results was implemented, allowing the visualization of both the obtained solution and the path taken to reach it.

6 Optimization Algorithm

To optimize the modeled problems, was chosen the "Butterfly Optimization Algorithm" or BOA. This approach relies on the use of fragrance to guide butterfly movement, inspired by their natural behavior in searching for food and mates. Through this metaheuristic, the algorithm extensively explores the solution space, allowing for rapid convergence towards both local and global optimal solutions [1].

The Python implementation of the butterfly optimization algorithm began with the random generation of a butterfly population within the decision variable limits, each with a random initial position and an initial fitness set to infinity, or minus infinity in the case of a maximization problem. The fragrance of each butterfly was then updated based on its fitness, guiding butterfly movement during iterations. The algorithm terminates if there are no fitness improvements for a certain number of consecutive iterations. The flowchart of the algorithm is depicted in Figure 3.

The initial parameters are: the power exponent which represent the absorption of the fragrance, a; the probability of moving towards the best butterfly, p; the number of individuals; the number of iterations and, finally, c will determine the speed convergence of the algorithm. Convergence, in this context, refers to the speed at which the algorithm finds an optimal or sub-optimal solution. In practice, the parameter c influences how quickly the butterflies move through the solution space towards better positions or explore new areas.

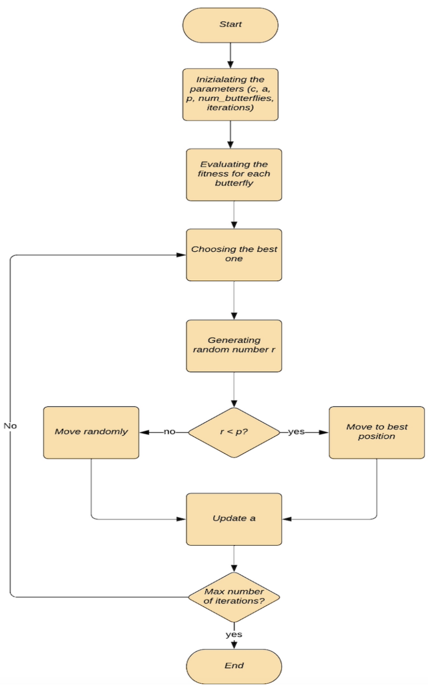

Figure 3. Flow chart diagram Butterfly Optimization Algorithm based on [5].

Since the first problem is a minimization problem and the second is a maximization problem, a parameter was introduced to specify the type of optimization to perform. Additionally, the second problem requires the optimization of binary variables. To handle this aspect, an additional parameter was added to specify whether the problem is binary. Since the initialization of butterfly positions produces continuous values, a method was implemented to convert these real numbers into binary values. Considering the rapid convergence of the binary, the "teleport" method was introduced. This method, similar to mutation in classical genetic algorithms, selects one or more coordinates of the butterflies, whose values are either 0 or 1, and flips their values with a certain probability. This results in a change in the overall position of the butterfly, hence the name "teleport." Changing the bits allows a deeper exploration of the search space.

7 Results

7.1 Maximizing coverage and reducing communication latency in a satellites system

The problem of optimizing satellite communications was tackled using the Butterfly Optimization Algorithm. In this study 5 satellites were considered. Three runs with different butterfly configurations (50, 100, 150) were conducted to evaluate the effectiveness of the algorithm in terms of fitness.

Table 1. Best solution of the satellite problem

Obj. fun.	Solution	Latency	Coverage
56.56	[13325.26, -17441.21 -17441.21, 9375.39 -15745.54, 9439.77 -16929.93, -11121.61 -9838.56, 9375.39]	0.2782	65.76%
56.24	[-11474.49, -13119.74 -11474.49, 11474.49 -13119.74, -13119.74 11474.49, 11474.49 -13119.74, 13119.74]	0.2747	63.46%
55.99	[11090.18, -9258.62 -11090.18, -11090.18 9244.42, 11090.18 9244.42, -9244.42 -11090.18, 11090.18]	0.2046	69%

The performance analysis of the algorithm showed in Figure 4 underline an improvement in fitness through the evaluations and iterations. The comparison between the three runs highlighted that increasing the number of butterflies led to faster convergence and an improvement in the optimal solution, the results are shown in Table 1. Specifically, the third run with 150 butterflies achieved the best fitness of 55.99, indicating that a larger number of butterflies can improve the quality of the solution. We can state that the third solution is the better one also checking the latency value and the coverage percentage obtained using the positions of the satellites.

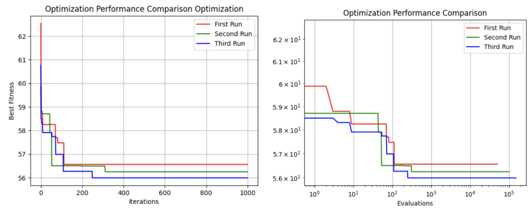

Figure 4. Evolution of the Objective Function for the Satellite Problem: Iteration on the left, Evaluation on the right.

The BOA algorithm has proven to be a powerful tool for optimizing satellite communications, achieving good solutions and significantly improving system coverage and latency. Its ability to extensively explore the solution space and rapidly converge towards optimal solutions makes it suitable for tackling complex optimization challenges in the context of satellite communication.

7.2 Maximizing online service profits

The second problem addressed is the optimization of server resource allocation to maximize profits from online services. Also in this case, the BOA algorithm was used by executing three runs of the code and the results are in Table 2. To simulate the arrival of requests, the Poisson distribution was used. This distribution is particularly suitable because it effectively models the occurrence of random events (requests) over a fixed interval of time. The Poisson distribution is defined by the average rate (λ) at which events occur and is characterized by the property that events happen independently of each other [2].

In this case, a time interval of 30 minutes was considered for the simulation. This approach is advantageous because it realistically captures the variability and unpredictability of request arrivals, which are common characteristics in real-world online services. By using this distribution, we can generate a more accurate and reliable simulation of server load, helping to optimize resource allocation and better manage the penalties for queued and rejected requests.

The second run, with 100 butterflies, achieved the highest fitness of 5020.800, indicating an optimal allocation of server resources that maximized profits while effectively managing penalties for queued and rejected requests. The comparison of optimization performance

Table 2. Best solution of the server allocation problem

Run	Objective function
1	4305
2	5020.8
3	4902.8

across the three runs, as depicted in the provided graphs in Figure 5, demonstrates a clear trend of fitness improvement with an increase in the number of evaluations and also iterations.

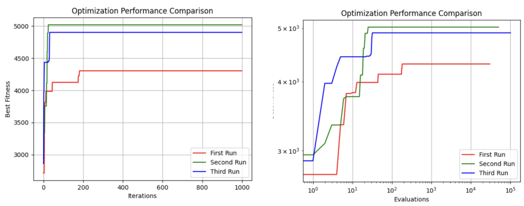

Figure 5. Evolution of the Objective Function for the Server Problem: Iteration on the left, Evaluation on the right.

The performance graphs show the convergence behavior of the algorithm, with the second run reaching the highest fitness level more rapidly compared to the other runs. This suggests that a moderate number of butterflies may provide an optimal solution with a better exploration of the search space.

8 Conclusions

The performance of the Butterfly Optimization Algorithm (BOA) in the two addressed problems clearly demonstrates its effectiveness in nonlinear optimization. Furthermore, the significant difference between the two proposed problems necessitated modifications to the algorithm, which increased the awareness of the limitations and advantages of the basic version of the BOA. However, as a metaheuristic, BOA does not guarantee finding the global optimum. The performance of the algorithm can also be highly dependent on the choice of parameters, such as the number of butterflies and the exploration-exploitation balance.

In conclusion, BOA is a powerful tool for tackling complex optimization problems. Future work could explore further refinements of the algorithm, such as adaptive parameter tuning and hybridization with other optimization techniques, and its application to an even wider range of optimization challenges.

References

1. Arora, Sankalap and Singh, Satvir, *Butterfly optimization algorithm: a novel approach for global optimization*, Soft Computing, 2019.
2. Turney, S. *Poisson Distributions | Definition, Formula & Examples*, https://www.scribbr.com/statistics/poisson-distribution/, 2022.
3. SES, (2024, March 11), *SES Lightboard Series: o3b mPOWER: The Optimal Orbit*, YouTube, https://www.youtube.com/watch?v=J9FOEQTffYw.
4. *Communication Technology | ShareTechnote. (n.d.).* https://www.sharetechnote.com/html/Communication_Satellite.htmlgoogle_vignette
5. Odam, Doaa & Doma, Mohamed & Fawzy, Hossam & Sedeek, Ahmed & Farahan, Magda. (2021). *Design of global positioning system (GPS) networks using different artificial intelligence techniques.* Journal of Engineering Research.

A Benchmark 2024

A.1 Introduction to the benchmark

In the Otimização Não-linear em Engenharia course, the 2024 benchmark focuses on the structural optimization of a ten-bar truss. The primary goal of this benchmark is to minimize the mass of the structure while maintaining mechanical integrity and adhering to structural balance and material elasticity constraints.

The problem is divided into two main optimization approaches, the first one is a sizing optimization approach focuses on identifying the optimal cross-sectional area for each of the ten bars that make up the truss; the second one is a shape optimization which involves modifying the positions of the structure's nodes [1]. The algorithm used to solve the problem is the BOA or Butterfly Optimization Algorithm.

A.2 Implementation

The programming language used for implementing the problem and the BOA algorithm is Python, chosen for its simplicity yet efficiency.

It should be noted that, for solving this particular type of problem, in addition to the basic algorithm parameters, an additional parameter indicating the presence of fixed nodes has been added. This parameter helps the algorithm recognize that the problem is one of shape optimization. After adjusting the positions of the butterflies, the positions of the fixed nodes are restored to their original ones.

A.3 Results

After the optimization process, that was executed three times, the results shown in table 3 were obtained (only the best solution out of the three is reported). To obtain this results the following parameter were used: $a = 0.0001$ with a maximum value of 0.9, 5000 iterations, 150 butterflies, $p = 0.6$ and $c = 0.4$.

Table 3. Best solution of the optimization process

Variables	Solution	Objective function
Areas	[4.65193832e-04	
	2.54698596e-04	
	6.04240066e-04	
	2.60649917e-04	
	9.54080766e-05	
	4.42777051e-05	
	5.19130196e-04	
	1.90940829e-04	
	1.38711543e-04	
	4.84162856e-041]	9.901
Coordinates	[0.49416138	
	0.17619362	
	0.57891094	
	0.06441345	
	0.36896025	
	0.18463933]	20.711

In Figure 6 and Figure 7, the results for the two formulations of the optimization problem are shown. More precisely in Figure 6.a the graph shows how the objective function varies with an increasing number of evaluations. A typical trend is the decreasing value of the objective function, indicating that the algorithm is finding lighter and more optimal structural configurations as it progresses through the iterations; in Figure 6.b are shown the optimized areas of the truss bars at the end of the optimization process. The specific values of the areas, as reported in the results, demonstrate a variation in the sizing of the bars to achieve an optimal structural configuration that minimizes mass while maintaining desired performance.

Similar to Figure 6.a, the Figure 7.a, shows a decreasing objective function, but specifically for the optimization of the truss shape. The Figure 7.b represents the final configuration of the truss, highlighting how the optimization has altered the position of the nodes to achieve an optimal shape. The coordinates mentioned in the results of the document reflect these changes and their impact on the objective function.

However, it is important to note that as a metaheuristic, BOA does not guarantee the finding of the global optimum. The solutions provided by the algorithm are sub-optimal but are usually sufficiently close to the best possible solution to be of practical use.

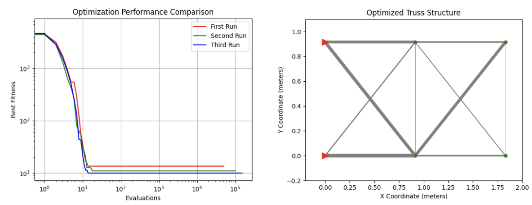

Figure 6. (a) Objective Function with Respect to the number of evaluation (b) Truss Bars Area

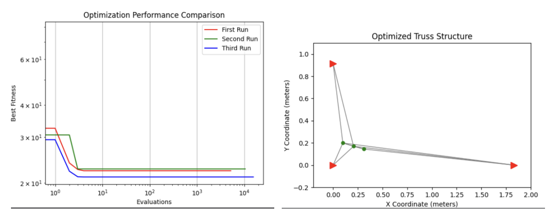

Figure 7. (a) Objective Function with Respect to the number of evaluation (b) Truss Shape

A.4 Optimization with 36 bars

In the 2024 benchmark, an additional challenge was introduced to test the robustness of the algorithm. Therefore, a truss structure comprising 15 nodes and 36 bars was analyzed. The best results for optimizing the area and shape of this structure are respectively are shown in Table 4.

In this context as well, the first two diagrams in Figure 8 depict the optimization of the structure in terms of the

Table 4. Best solution of the optimization process 36 Bars Truss

Variables	Solution	Objective function
Areas	[7.97e-05, 2.93e-04, 3.41e-04 2.78e-05, 7.69e-05, 2.85e-04 1.60e-04, 2.72e-04, 3.39e-04, 1.68e-04, 6.18e-05, 2.34e-04 8.65e-05, 1.51e-04, 6.12e-05 2.49e-04, 4.36e-04, 4.50e-04 1.96e-05, 7.24e-05, 1.96e-05 2.81e-04, 4.55e-04, 9.88e-05 3.05e-04, 1.008e-04, 2.77e-04 1.96e-05, 2.64e-04, 1.13e-04 2.12e-04, 2.53e-04, 2.64e-04 1.32e-04, 3.69e-05, 1.04e-04]	32.006
Coordinates	[0.35357601, 0.51648572 0.42439275, 0.9182457 0.32346149, 0.98957396 0.19321955,b0.91292161 0.19390933, 0.35052978 0.54386957, 0.45909583 0.65842671, 0.88671185 0.9527046, 0.52065415 0.49692594, 0.27137797 0.75528832, 0.72571851 0.64677297, 0.03203518]	72.263

area of the bars, highlighting how the distribution of the area evolved to achieve the optimal result. On the other hand, Figure 9 illustrates the optimization concerning the shape of the truss, showing the modifications in the geometric configuration that led to the most efficient possible shape. These results reflect the effectiveness of the algorithm in significantly improving structural performance in terms of both area and shape, while also demonstrating its ability to tackle the additional complexities introduced in this year's benchmark.

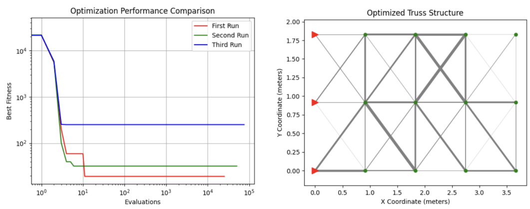

Figure 8. (a) Objective Function with Respect to the number of evaluation (b) Truss Bars Area

In conclusion, while the BOA has proven to be an efficient and effective tool for structural optimization in engineering, it is critical for users to understand its limitations as a metaheuristic. Future work could focus on combining BOA with other optimization techniques, adjusting its parameters for better convergence, or applying it to other complex engineering problems to further explore its capabilities and limitations.

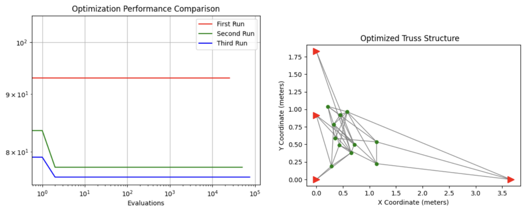

Figure 9. (a) Objective Function with Respect to the number of evaluation (b) Truss Shape

References

1. A. Andrade-Campos and J. Dias-de-Oliveira, *Benchmark 2024: Structural optimisation of a ten-bar- truss*, 2024.
2. Arora, Sankalap and Singh, Satvir, *Butterfly optimization algorithm: a novel approach for global optimization*, Soft Computing, 2019.