BIOGEOGRAPHY-BASED RELIABILITY OPTIMIZATION

Deepika Garg[1], Sarita Devi[2*], Doorgeshwaree Devi Heeramun[3], Nakul Vashishth[4]

•

[1,2,3,4] GD Goenka University, Gurugram, Haryana

[1]deepikanit1234@gmail.com, [2]sarita.fame@gmail.com
[3]vidishaheeramun@yahoo.com, [4]nakulsir@gmail.com
*Corresponding author

Abstract

Biogeography-based optimization (BBO) is an effective optimization algorithm originally introduced by Dan Simon in 2008. It is inspired by the principles of biogeography. It mimics the process of migration and colonization of different species, where the behavior of species is determined by their habitats and the available resources. However, BBO can suffer from slow convergence and premature convergence for high-dimensional optimization problems. To address these issues, opposition-based learning (OBL) has been integrated with BBO. OBL uses the concept of generating an opposite solution to an existing solution and combining them to form a new search direction. This helps to improve the diversity of the population and accelerate the convergence speed of the algorithm. The combination of BBO and OBL has shown promising results in solving various optimization problems, including engineering design, scheduling, and feature selection. This approach has proved to be efficient and robust, making it an attractive alternative to other optimization algorithms. In this research study, the redundancy Allocation Problem (RAP) is solved with BBO and OBL-BBO. Computational results confirm the robustness and superiority of this new model over previous ones in optimizing the reliability of the system. Besides, the proposed algorithm OBL-BBO outperforms the previous ones BBO in finding good results.

Keywords: Biogeography-based optimization (BBO), opposition-based learning (OBL), opposition-based learning based BBO (OBL-BBO), RAP.

I. Introduction

Reliability optimization aims to boost a system's dependability while cutting costs or improving overall efficiency [1]. In practice, reliability is typically pursued through two main avenues: (i) enhancing the intrinsic reliability of individual components and (ii) introducing redundant units. Although using highly reliable components can raise dependability, performance often plateaus before the target level is reached. Redundancy, meanwhile, can deliver the desired reliability but inevitably adds cost, weight, and volume [2]. The redundancy allocation problem (RAP) was first formulated for systems with multiple states and interacting failures [3]. A subsequent study proposed a heuristic procedure to solve RAP for a manufacturing system with series-connected subsystems, relying on redundancy to improve overall reliability [4]. Later, researchers tackled RAP with a Hybrid Constraint Optimization Genetic Algorithm combined with

Particle Swarm Optimization (H-PSOCOGA) and compared its results with those from the heuristic method and a standalone Constraint-Optimization Genetic Algorithm (COGA) [5].

The K-mixed strategy has been applied to tackle RAP, aiming to elevate system reliability. After using this strategy to build a mathematical model for reliability estimation, its effectiveness was tested on several benchmark and custom problems to gauge power and efficiency [6]. Fault-tree analysis—a graphical, logic-based approach that examines system failure modes—has also been used to address the reliability-optimization task and calculate failure probabilities in manufacturing systems [7]. In another investigation, three approaches— a Heuristic Algorithm (HA), a Constrained-Optimization Genetic Algorithm (COGA), and a Hybrid Genetic Algorithm coupled with Particle Swarm Optimization (HGAPSO)—were implemented for RAP, with a comparative performance assessment provided for all methods [8].

A Hybrid Genetic Simulated Annealing Algorithm (HGSAA) has been proposed to solve the RAP in manufacturing systems. By blending the strengths of genetic algorithms and simulated annealing, HGSAA mitigates the individual weaknesses of each method, and its performance has been benchmarked against a Heuristic Algorithm (HA), a Constrained-Optimization Genetic Algorithm (COGA), and the Hybrid Particle Swarm Optimization–COGA approach (H-PSOCOGA) for the same problem set [9]. In another study, three dedicated heuristics—HASL1, HASL2, and HASL3, each defined by a different selection factor—were created to allocate redundancy within the cost constraints of Yaris Pharmaceuticals' liquid-medicine production line, with post-allocation results compared graphically [10]. Separately, the Jaya algorithm has been employed to estimate the probability $P(X > Y)$ for a two-parameter Weibull distribution [11].

A broad survey spanning 2000 – 2022 catalogued the literature on the redundancy-allocation problem, organizing and analyzing prior studies by system type, configuration, solution method, and constraint set. Drawing on this synthesis, the review pinpointed under-explored areas and proposed avenues for future research where coverage remains thin [12].

Spider Monkey Optimization (SMO)—a comparatively new meta-heuristic previously unused for RAP—has been applied to redundancy allocation in manufacturing systems. Benchmarks against Particle Swarm Optimization (PSO) indicate that SMO yields superior solutions. The same work also analyzed time-dependent availability to forestall total system shutdowns, illustrating how mathematical models and search algorithms can identify reliability-maximizing configurations. This research area overlaps with maintenance planning and resource management and sees applications from engineering to telecommunications [13]. In a separate study, the Jaya algorithm was employed to solve two RAP formulations with nonlinear constraints, aiming to maximize system reliability [14].

The overarching objective across industries is to keep systems running reliably and to minimize downtime. Nature-inspired, or bio-inspired, optimization methods borrow ideas from evolutionary biology—such as natural selection, adaptation, and migration—to tackle complex search and optimization tasks. By mirroring the adaptability of living systems, these algorithms deliver robust and flexible problem-solving tools. A prominent example is Biogeography-Based Optimization (BBO), which models the migration of species among habitats [15]. BBO blends exploration and exploitation via migration operators, offering computational efficiency, versatility, simplicity, and a derivative-free search. These qualities have made it one of the fastest-growing meta-heuristics for difficult optimization problems. Nevertheless, BBO is not without drawbacks: its success depends on careful parameter tuning, its exploratory power can taper off in very high-dimensional spaces, its convergence may lag some alternatives, and performance can fluctuate across problem domains.

Researchers have contrasted Biogeography-Based Optimization (BBO) with the Genetic Algorithm (GA) for designing damping controllers in power-system applications [16]. A separate study blended BBO with the Shuffled Frog-Leaping Algorithm and applied the hybrid to minimum-spanning-tree problems. While limited theoretical grounding and reduced adaptability to dynamic or heavily constrained scenarios have been noted as weaknesses, current work seeks to address these gaps and extend BBO's range of use [17]. Further efforts have merged BBO with additional meta-heuristics; experimental evaluations of these hybrids highlight both their performance improvements and their remaining trade-offs [18].

Reference [19] delivers an in-depth overview of Biogeography-Based Optimization (BBO) and illustrates its value for difficult optimization problems. Opposition-Based Learning (OBL) further enriches the search process by generating each candidate solution alongside its opposite, expanding the initial exploration of the design space. When this strategy is integrated with BBO—yielding the so-called Oppositional BBO—experiments report faster convergence and superior solution quality compared with the conventional BBO algorithm [20]. Subsequent surveys highlight BBO's competitive advantages over other meta-heuristics, its wide-ranging applications, and open avenues for future research [21]. Building on OBL's diversity and BBO's framework, investigators have also combined BBO with Ant Colony Optimization (ACO); this hybrid improves global search capability, helps the solver avoid local optima, and performs particularly well on binary optimization tasks [22].

Building on prior redundancy-allocation research, this investigation addresses a realistic manufacturing system with multiple subsystems that must meet stringent reliability targets under fixed cost, weight, and volume caps. The study applies two meta-heuristic optimizers-standard Biogeography-Based Optimization (BBO) and its Opposition-Based extension (OBL-BBO) to the identical mixed-integer formulation, enabling a direct, fair performance comparison. BBO leverages migration and mutation operators to balance exploration and exploitation, whereas OBL-BBO supplements each candidate solution with its opposite counterpart, enriching population diversity and accelerating global search. Analysis reveals that OBL-BBO reaches the desired reliability within tighter budgets and fewer iterations, although it exhibits slightly greater sensitivity to parameter tuning. The comparative insights gained guide practitioners in selecting the most appropriate algorithm for redundancy-allocation tasks with similar industrial constraints and anticipated future enhancements. The subsequent Section 2 outlines the methodological frameworks for BBO and OBL-BBO, followed by a detailed problem formulation in Section 3. Section 4 presents the experimental results alongside an in-depth discussion, while Section 5 distils the main findings and contributions of the research.

II. Methods

I. Methodology of OBL

Opposition-based learning (OBL) accelerates convergence to optimal solutions by exploring both candidates and their opposites, enhancing efficiency and solution quality. It also helps in escaping local optima, leading to more robust optimization results.

Symbols:

- X: A candidate solution in the search space.
- X': The opposite (complementary) solution of X
- $f(X)$: Objective function value of the solution X
- $f(X')$: Objective function value of the opposite solution X'
- α: a parameter controlling the step size
- R_i: Reliability of the system
- Q_i: Unreliability of the system

The methodology of OBL is explained in following steps:

1) Initialization

Two individuals are created - one represents the initial solution and the other represents its opposite. The opposite solution is created by altering the signs of the decision variables.

Generate an initial population of solutions X
Evaluate the fitness of each solution using the objective function: $f(X)$
$X_i' = -X_i$ for each variable i

2) Fitness Evaluation

The fitness of both solutions is determined using the objective function. The solution with better fitness is selected as the parent for the next generation.

$$X_{next} = argmax\ (f(X), f(X')) \quad (1)$$

3) Search space exploration

To efficiently explore the search space, an opposition-based approach is utilized. This allows the algorithm to break away from local optima by considering solutions that are "opposite" to the current best solution.

$$X_i(t+1) = X_i(t) + \alpha \left(X_i' - X_i(t+1)\right) \quad (2)$$

4) Iterative refinement

The process is then repeated iteratively, with each iteration aiming to refine the solution by implementing the opposition-based strategy. This approach allows for a more human-like thought process, as we also consider opposing ideas in our decision-making. Repeat steps 2-6 until a termination criterion is met (e.g., a maximum number of iterations, satisfactory solution found).

II. Methodology of BBO

Symbols**:**

- X_i : A candidate solution (habitat) in the population.
- $f(X_i)$: Objective function value of the habitat X_i.
- M: Migration rate (probability of migration).
- P_m: Mutation rate (probability of mutation).
- N: total number of habitats in the population.
- X_{new}: a randomly selected habitat from the population.
- $rand()$: a random number between 0 and 1
- X_{rand}: a randomly selected habitat from the population.

The methodology of OBL is explained as:

1. Initialization

   Generate an initial population of habitat X_i.

2. Evaluate fitness.
   Evaluate the fitness of each habitat using objective function: $f(X_i)$

3. Habitat Suitability Index (HSI):
   To determine which habitats are most suitable for each solution, a Habitat Suitability Index (HSI) is assigned based on its fitness. Solutions with higher fitness values are seen as more desirable habitats.

$$HSI_i = \frac{f(X_i)}{\sum_{j=1}^{N} f(X_j)} \tag{3}$$

4. Migration Rate and Mutation Rate:
   The process of migration and mutation is not random, as it is controlled by the migration and mutation rates. The migration rate determines the likelihood of a solution moving to a different habitat.

$$X_i(t+1) = X_i(t) + M \cdot HSI_i\big(X_{new} - X_i(t)\big) \tag{4}$$

   while the mutation rate dictates the chances of introducing changes to a solution.

$$X_i(t+1) = X_i(t+1) + P_m \cdot rand().\big(X_{rand} - X_i(t+1)\big) \tag{5}$$

   Evaluate the fitness of the new habitats.

5. Population Update:
   As the population evolves, solutions are updated through a combination of migration, mutation, and the HSI. The ultimate goal of this algorithm is to continuously improve the overall fitness of the population with each iteration.

$$X_i(t+1) = argmax\ \big(f\big(X_i(t+1)\big), f\big(X_i(t)\big)\big) \tag{6}$$

6. Convergence:
   The algorithm will continue until a convergence criterion is met, such as reaching a maximum number of iterations or finding a satisfactory solution. Just like in nature, it is all about finding the perfect balance and adapting to the changing environment. Repeat steps 2-7 until a termination criterion is met (e.g., a maximum number of iterations, satisfactory solution found).

III. Methodology of Opposition-Based Learning based BBO (OBL-BBO)

1. Initialization:
   Generate an initial population of solutions randomly in the search space.

2. Opposition Generation:
   For each solution X_i generate its opposite solution X_i' by changing the signs of its decision variables.
   $X_i' = -X_i$ for each variable =–for each **variable** i

3. Opposition Fitness Evaluation:
   Evaluate the fitness of each opposite solution using the objective function.

4. Biogeography-Based Optimization:
   Apply BBO steps:
   a. Calculate Habitat Suitability Index (HSI) for each solution based on its fitness.

b. Perform Migration Operation:
   1. Determine which solutions will undergo migration based on the migration rate M.
   2. Update the solutions using migration.

c. Perform Mutation Operation:
   1. Determine which solutions will undergo mutation based on the mutation rate Pm.
   2. Update the solutions using mutation.

5. Evaluate fitness of the new solutions.

6. Update the population based on fitness and HSI.

7. Termination:
   Repeat steps 2-4 until a termination criterion is met (e.g., a maximum number of iterations or a satisfactory solution found).

An example of a real-world application of OBL-BBO is Oppositional biogeography-based optimization (OBBO), which is an improved version of the moth flame optimization (MFO) algorithm that uses OBL-BBO to optimize complex objectives. OBBO has been used to solve 18 benchmark functions and two constrained real-world problems, including global optimization.

III. Problem Definition

The manufacturing system consists of various units such as overhead crane, Blanking machine, Stacker machine, Press machine, Molding and Packing. All these machines work in a series system. In manufacturing system, the problem is to maximize reliability with cost constraint. The given cost limitation is C= 30490000 (Here number of constraints is one). Here Table 1 presents the reliability cost of each subsystem.

The problem is to maximize:

$$R_s(y) = g\,(R_1(y_1), \ldots\ldots\ldots\ldots, R_n(y_n)) = \prod_{j=1}^{7} R_j\left(y_j\right) \tag{7}$$

$$R_j(y_j) = \prod_{j=1}^{7}[1 - [Q_j(y_j)]^{n_j}] \tag{8}$$

$$\sum_{j=1}^{7} h^1(y_j) * n_j \leq 30490000 \tag{9}$$

Table 1: *Reliability cost of each subsystem*

Subsystem	**y_1**	**y_2**	**y_3**	**y_4**	**y_5**	**y_6**	**y_7**
Reliability of subsystem $R_j(y_j)$	0.99	0.9762	0.9188	0.8155	0.8655	0.9287	0.9453
Cost of subsystem $h_1(y_j)$	1240000	860000	2100000	1010000	10200000	840000	200000

I. Mechanism for BBO

The brief overview of the flow chart of Biogeography-Based Optimization is presented in Figure 1. The actual implementation of the algorithm may involve additional steps and parameters that can be customized based on the specific optimization problem.

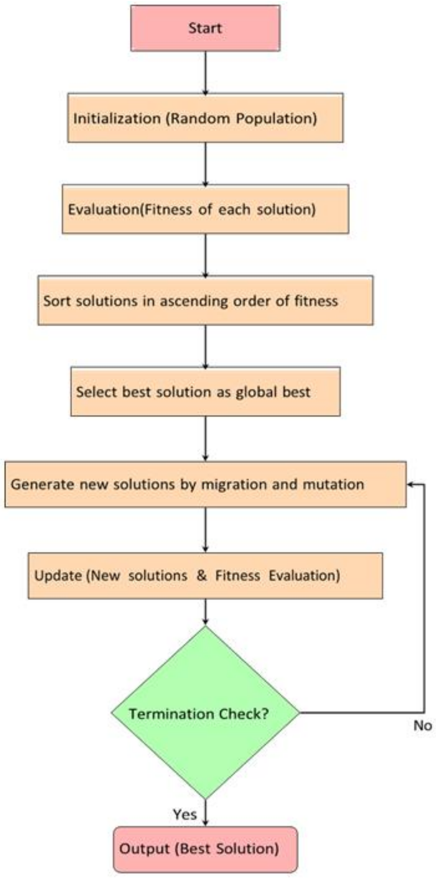

Figure 1: *Mechanism for BBO*

IV. Results and Discussion

The results obtained by the OBL, BBO and OBL-BBO techniques are represented in the tabular form. In Table 2 and Table 3, the subsystems and units of the subsystems are given.

Table 2: *Results of RAP of manufacturing system by BBO*

Subsystems	y_1	y_2	y_3	y_4	y_5	y_6	y_7	Reliability	CPU time
No. of units	3	3	4	3	1	4	4	0.8600	0.001s

The reliability value of the subsystem before applying BBO technique is 0.5502. The redundant units of the subsystem are considered y*= (1 1 1 1 1 1 1). After applying BBO technique, the optimal solution of RAP is y*= (3 3 1 4 1 4 4) and the reliability value R(y*) = 0.8600 as shown in Table 2. The reliability of the system is increased by 56%. The CPU time taken by software is 0.001s by BBO technique. The graphical representation of the result from BBO is demonstrated in Figure 2. The number of optimum redundancy units of the various subsystems are represented in the graph as shown in figure.

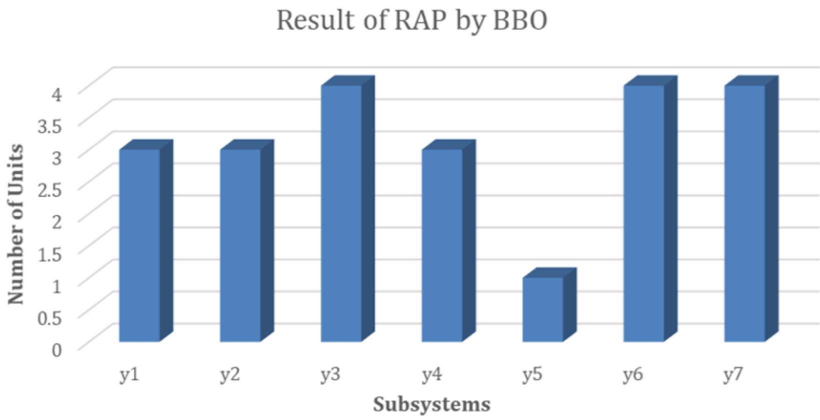

Figure 2: *Results of RAP obtained by BBO*

Similarly, in Table 3, the reliability value of the Subsystem before applying OBL-BBO technique is 0.5502 and the redundant units of the subsystems are considered y*=(1 1 1 1 1 1 1). After applying OBL-BBO technique, the optimal solution of RAP is y*= (3 4 3 4 1 4 4) and the reliability value is R(y*) = 0.8640. The reliability of the system is increased by 57%. The CPU time taken by software is 0.001s by OBL-BBO technique.

Table 3: *Results of RAP of manufacturing system by OBL-BBO*

Subsystems	y_1	y_2	y_3	y_4	y_5	y_6	y_7	Reliability	CPU time
No. of units	3	4	3	4	1	4	4	0.8640	0.001s

Similarly, the graphical representation of the result obtained by OBL-BBO is shown in Figure 3. The results obtained from OBL-BBO are better than results of BBO with respect to the redundancy units of different subsystems and reliability also.

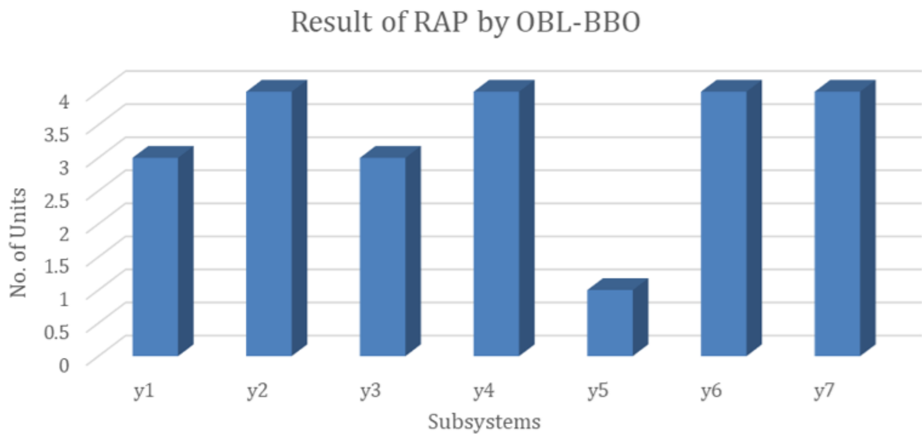

Figure 3: *Results of RAP obtained by OBL-BBO*

The CPU time taken by both the techniques are same, but the result of total reliability of the system and optimum number of redundant units are obtained better from OBL-BBO than BBO.

V. Conclusion

In this paper, RAP of a manufacturing system is investigated first by BBO technique. After that the RAP of the manufacturing system is solved by OBL-BBO. OBL-BBO contributes to the rich tapestry of optimization research by introducing a novel combination of opposition-based learning and biogeography-based optimization. Its implications extend beyond theoretical advancements, offering practical solutions and insights that can shape the future landscape of optimization algorithms. The results obtained from both the techniques are presented in tabular form. The numerical outcomes demonstrate the superiority of this OBL-BBO over BBO in solving RAP of the manufacturing system

References

[1] Simon, D. (2008). Biogeography-Based Optimization. IEEE Transactions on Evolutionary Computation, 12(6):702–713.

[2] Sharifi, M., Taghipour, S. (2022). Redundancy allocation problem of a Multi-State system with Binary-State continuous performance level components. Expert Systems with Applications, 200: 117161.

[3.] Yeh, W. C., Hsieh, T. J. (2011). Solving reliability redundancy allocation problems using an artificial bee colony algorithm. Computers & Operations Research, 38(11):1465-1473.

[4.] Wang, J., Li, M. (2015). Redundancy Allocation Optimization for Multistate Systems with Failure Interactions Using Semi-Markov Process. ASME.-Journal of Mechanical Design, 137(10): 101403.

[5] Devi, S., & Garg, D. (2017). Redundancy-Allocation in Neel Metal Products Limited. Indian Journal of Science and Technology, 10(30).

[6] Devi, S., Sahu, A., Garg, D. (2017). Redundancy optimization problem via comparative analysis of H-PSOCOGA. International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN), Gurgaon, India, 2017, 18-23,

[7] Peiravi, A., Karbasian, M., Ardakan, M.A., Coit, D.W. (2019). Reliability optimization of series-parallel systems with k-mixed redundancy strategy. Reliability Engineering & System Safety, 183: 17–28.

[8] Devi, S., Garg, D. (2019). Reliability analysis of manufacturing plant via fault tree analysis. Journal of Advances and Scholarly Researches in Allied Education, 16(5):256-259.

[9] Devi, S., Garg, D. (2020). Hybrid genetic and particle swarm algorithm: redundancy allocation problem. International Journal of System Assurance Engineering and Management, 11: 313–319.

[10] Garg, D., Devi, S. (2021). RAP via hybrid genetic simulating annealing algorithm. International Journal of System Assurance Engineering and Management, 12: 419–425.

[11] Dahiya, T., Garg, D., Devi, S., Kumar, R. (2021). Reliability optimization using heuristic algorithm in pharmaceutical plant. Reliability Theory & Applications, 16(3): 195-205.

[12] Raikar, S.L., Gaonkar, R.S.P. (2022). Jaya algorithm in estimation of P [X> Y] for two parameter Weibull distribution. AIMS Mathematics, 7(2): 2820-2839.

[13] Devi, S., Garg, H., & Garg, D. (2023). A review of redundancy allocation problem for two decades: bibliometrics and future directions. Artificial Intelligence Review, 56(8): 7457-7548.

[14] Devi, S., Vashishth, N., Garg, D. (2023). RAP and Availability analysis of manufacturing system: SMO and PSO. Reliability: Theory & Applications, 18, 4(76): 680-691.

[15] Aswin, B., Lokhande, T., Gaonkar, R. (2023). Solving Redundancy Allocation Problems using Jaya Algorithm. International Journal of Mathematical, Engineering and Management Sciences, 8:804-816.

[16] Said, A. I., El-Saadany, E. F., and Salama, M. M. A. (2013). Comparison of Biogeography Based Optimization and Genetic Algorithm for Power System Damping-Based Controllers Design. IEEE Electrical Power & Energy Conference, IEEE, 1–5.

[17] Zheng, Y., Lu, X., Zhang, M., Chen, S., (2019). Hybrid Biogeography-Based Optimization Algorithms, Biogeography-Based Optimization: Algorithms and Applications. Springer Singapore, Singapore, 89–115.

[18] Zhang, X., Kang, Q., Wang, X. (2019). Hybrid Biogeography-Based Optimization with Shuffled Frog Leaping Algorithm and Its Application to Minimum Spanning Tree Problems. Swarm Evolutionary Computation, 49: 245–265.

[19] Mirjalili, S. (2019). Biogeography-Based Optimisation. 57–72.

[20] Ergezer, M., Simon, D., and Du, D. (2009). Oppositional Biogeography-Based Optimization. IEEE International Conference on Systems, Man and Cybernetics, IEEE, 1009–1014.

[21] Sharma, S., Chandrasekhar Rao, D. (2022). Biogeography-Based Optimization. 471–479.

[22] Dai, Z., Ma, Q., Zhao, D., Yan, S. (2020). A Novel Hybrid Optimization Algorithm Combined with BBO and ACO. *39th Chinese Control Conference (CCC)*, IEEE, 1581–1586.