

Rainbow Trout Optimization Algorithm

Jincheng Zhang^{*1}[0009-0005-1860-0009]

¹ Faculty of Science and Technology, Rajabhat Maha Sarakham University, Maha Sarakham
44000, Thailand
zjc1639834588@gmail.com

Abstract. To address the problems of traditional swarm intelligence algorithms, such as reliance on gradient or gradient-like information, conservative dynamic structures, and susceptibility to local optima, this paper proposes a rainbow trout optimization algorithm based on non-conservative induced flow field, curvature refraction modulation, topological backflow memory, entropy-driven splitting, and energy conservation violation mechanisms. This algorithm, from the perspective of fluid dynamics and non-equilibrium statistical physics, treats the objective function as the source term of the induced flow field. By introducing an antisymmetric curl coupling matrix, a non-conservative dynamic system is constructed, enabling the search trajectory to maintain a downward trend while possessing flow-around capability. Furthermore, a curvature refraction tensor is constructed to achieve anisotropic step-size modulation, and a continuous backflow kernel replaces the traditional skip-memory strategy. The algorithm as a whole adopts a second-order dynamic structure and combines swarm entropy and system energy evolution laws to achieve adaptive splitting and energy injection. Theoretical analysis shows that this method belongs to a dissipative non-equilibrium system at the dynamic system level, possessing stronger global exploration capabilities and structural representation capabilities. Finally, the algorithm's complexity and stability are analyzed to demonstrate its potential advantages in high-dimensional complex optimization problems.

Keywords: Rainbow trout optimization algorithm; non-conservative flow field; vortex coupling; curvature refraction; topological recirculation.

I. Introduction

In continuous space optimization problems, the objective function is usually defined on the d -th power of a high-dimensional Euclidean space R . Its function shape often exhibits high non-convexity, multiple local extrema, and complex curvature distribution structures. Especially in high-dimensional cases, the isosurface of the function may exhibit strong anisotropy, meaning the function changes drastically in some directions while remaining relatively flat in others. This complex structure leads to many challenges for traditional optimization methods in practical applications [1-47].

Classical gradient-based optimization methods essentially construct a conservative dynamical system. Its basic form is:

$x(t+1)$ equals $x(t)$ minus η multiplied by the gradient $f(x(t))$.

This form can be viewed as a descent motion in the potential field induced by the objective function f . Since the gradient field satisfies zero curl, the flow field is a conservative field. An important property of a conservative field is that its path integral is independent of the path, and the system's trajectory will eventually converge to a stable equilibrium point. For non-convex functions, this equilibrium point may only be a local minimum rather than a global optimum. While the local optima problem can be mitigated to some extent by introducing random perturbations, momentum terms, or adaptive step-size mechanisms, as long as the basic structure remains dominated by the negative gradient, the overall dynamical system retains an approximately conservative structure. In other words, the search trajectory macroscopically still evolves monotonically along the direction of decreasing potential energy, lacking the ability to overcome potential barriers and move around currents.

Swarm intelligence algorithms, starting from group cooperation and randomness, alleviate the problem of single trajectories getting trapped in local optima through the cooperative evolution of multiple individuals in the solution space. However, most swarm intelligence methods still implicitly contain some form of "central attraction structure," such as approaching the global or local optimum. Their individual update formula can typically be expressed as:

$x_i(t+1)$ equals $x_i(t)$ plus attraction term plus random term

The attraction term is often proportional to the difference vector of the current optimal solution; this structure is geometrically similar to the potential field attraction mechanism. Essentially, its macroscopic dynamics can still be approximated as a generalized conservative system or a weakly dissipative system, lacking a clear non-conservative curl structure.

On the other hand, most algorithms employ a first-order update model, where position is directly determined by the current state without explicitly introducing acceleration or inertia terms. This lack of higher-order dynamics limits the system's ability to traverse narrow potential wells or complex canyon structures. In high-curvature regions, first-order systems often exhibit oscillations or excessive decay, impacting search efficiency.

Therefore, reconstructing the optimization process from a dynamical system perspective, imbuing it with the following characteristics, is crucial:

First, constructing a non-conservative flow field to enable the search trajectory to rotate and flow around;

Second, introducing a spatial geometry modulation mechanism to adapt the step size to the local curvature structure;

Third, treating the system as a non-equilibrium dissipative system, maintaining long-term exploration activity through energy mechanisms;

Fourth, constructing a continuous memory flow, rather than a simple discrete jump strategy.

The behavior of rainbow trout in nature provides biological inspiration for these ideas. When rainbow trout swim upstream, they do not simply move in a straight line against the current, but rather utilize vortex structures in complex flow fields to achieve path navigation. Their motion exhibits inertial characteristics, and energy consumption is closely related to flow velocity. After reaching a suitable area, individuals possess the ability to re-enter historically favorable waters. These behaviors correspond to mathematical structures such as non-conservative flow fields, second-order dynamics, energy coupling, and backflow memory.

Based on these observations, this paper proposes a rainbow trout optimization algorithm. By constructing a non-conservative induced flow field and curvature refraction mechanism, a discrete dynamic system with curl, backflow, and energy dissipation characteristics is established, thereby enhancing the global search capability in complex non-convex spaces.

II. Problem Definition

Consider the following continuous optimization problem:

Minimize $f(x)$

Where x belongs to the d -th power of R , and the function f is a real-valued function from the d -th power of R to R .

In practical applications, $f(x)$ may not be analytically differentiable, or gradient computation costs may be too high. Therefore, the algorithm design must allow operation under unknown gradient conditions.

Let the population size be N . The position of the i -th individual at time t is denoted as $x_i(t)$. If a velocity variable is introduced, its velocity is denoted as $v_i(t)$.

The global optimal solution is defined as:

$x_best(t)$ equals the point with the smallest f value among all individual historical positions.

The historical memory set is defined as:

$M(t)$ equals $\{m_1, m_2, \dots, m_K\}$

where each m_k is a historically optimal solution.

To analyze the population distribution structure, the individual position distribution probability can be defined. The space is divided into several regions, and the proportion of individuals in the k -th region is denoted as $p_k(t)$. The following condition must be met:

The sum of $p_k(t)$ for k equals 1.

The above definition lays the foundation for subsequent entropy and splitting mechanisms.

III. Construction of Non-Conservative Induced Flow Field

3.1 Gradient Approximation Term

To avoid dependence on analytical gradients, the central difference method is used to construct a gradient approximation. Let δ be a sufficiently small positive number, and let e_j be the j -th dimension unit vector. Then, we define:

$G_j(x,t)$ equals

$(f(x + \delta e_j) - f(x - \delta e_j))$

divided by $(2 \text{ times } \delta)$.

The gradient approximation vector $G(x,t)$ is defined as:

$G(x,t)$ equals $(G_1(x,t), G_2(x,t), \dots, G_d(x,t))$.

This term approximates the true gradient when δ is sufficiently small. Geometrically, it constructs a linear approximate plane near the current point, providing a local steepest descent direction.

However, relying solely on this term will cause the system to become an approximately conservative field. To overcome this limitation, a curl structure needs to be introduced.

3.2 Curl Generation Term

In vector field theory, if the curl of a vector field is not zero, it is a non-conservative field. To construct the curl term within the discrete framework, an antisymmetric matrix $S(t)$ is introduced.

Matrix $S(t)$ satisfies:

The transpose of $S(t)$ equals the negative $S(t)$.

That is, for any i and j ,

$S_{ij}(t)$ equals the negative $S_{ji}(t)$.

The curl term is defined as:

$A(x,t)$ equals $S(t)$ multiplied by $G(x,t)$.

Its properties are analyzed. Considering the inner product:

The transpose of $G(x,t)$ multiplied by $A(x,t)$

equals the transpose of G multiplied by S multiplied by G .

Since S is an antisymmetric matrix,

The transpose of G multiplied by S multiplied by G equals 0.

Therefore, $A(x,t)$ and $G(x,t)$ are orthogonal.

This property implies that, in the first-order approximation, the curl term does not change the descent direction projection of the function value, but it does change the direction of motion, causing the trajectory to deflect. If G is considered as the radial component, then A is equivalent to the tangential component. Their superposition forms a spiral trajectory.

Mathematically, this structure is equivalent to transforming the original gradient system into:

Point x equals the negative gradient $f(x)$ plus S multiplied by the gradient $f(x)$.

The part induced by S generates a rotating flow.

3.3 Topological Backflow Term

To construct a continuous memory mechanism, a set of historical good solutions M is introduced. Let the elements in M be m_k , where k ranges from 1 to K .

Define the backflow kernel function:

$R(x,t)$ equals

summing k from 1 to K

$\omega_k(t)$ multiplied by

$\exp(\text{negative } \|x - m_k\|^2 \text{ divided by } \sigma_k)$

multiplying by $(m_k - x)$

where $\|x - m_k\|^2$ squared represents the squared Euclidean distance.

This expression can be interpreted as an attractive field formed by the superposition of multiple Gaussian kernels. If x is close to a certain m_k , the corresponding exponential term approaches 1, and the attractive force increases; if the distance is greater, the exponential term decays rapidly.

Unlike a simple jump to m_k , this mechanism creates a continuous attractive flow in space, causing individuals to naturally flow back when approaching historically favorable regions.

3.4 Total Induced Flow Field

Combining the gradient approximation, curl term, and recirculation term, the total flow field is defined as:

$F(x,t)$ equals

negative $G(x,t)$

plus $A(x,t)$

plus $R(x,t)$

Expanding this, we get:

$F(x,t)$ equals

negative $G(x,t)$

plus $S(t)$ multiplied by $G(x,t)$

plus summation of k from 1 to K

$\omega_k(t)$ multiplied by $\exp(\text{squared of negative } \|x - m_k\| \text{ divided by } \sigma_k)$

multiplied by $(m_k - x)$

Due to the presence of the $S(t)$ term and the recirculation term, this vector field generally does not satisfy the conservatism condition. In other words, its curl is usually not zero.

From a dynamical system perspective, this flow field can exhibit the following behaviors:

First, descent along the negative gradient direction;

Second, circulation around equipotential surfaces;

Third, formation of local circulation in historically favorable regions.

Therefore, the system no longer simply converges to the nearest minimum, but may traverse complex potential wells through rotational and recirculation structures.

IV. Curvature Refraction Mechanism

In complex non-convex optimization problems, the local geometry of the objective function has a decisive influence on the algorithm's behavior. If the curvature is large in a certain direction, the function value changes drastically along that direction; if the curvature is small, the function is relatively flat in that direction. If curvature information is ignored and a uniform step size is used, oscillations, overshoot, or even numerical instability are likely to occur in high-curvature directions; while convergence may be too slow in low-curvature directions.

Therefore, it is necessary to introduce a curvature modulation mechanism based on the flow field to adaptively match the search behavior with the local geometry.

4.1 Local Diagonal Curvature Approximation

Consider the second-order difference approximation of the function in the j -th coordinate direction. Let δ be a small positive number, and the j -th unit vector be e_j . Then, define:

$h_j(x)$ equals

$(f(x + \delta e_j) - 2f(x) + f(x - \delta e_j))$

divided by the square of δ .

This expression is the second-order central difference approximation of the function in the j -th direction. When δ is sufficiently small, $h_j(x)$ is approximately equal to the second-order partial derivative in that direction.

If the absolute value of $h_j(x)$ is large, it indicates that the curvature in that direction is large; if $h_j(x)$ is close to zero, it indicates that the direction is approximately linear or flat.

Constructing the Curvature Vector

$H_{diag}(x)$ equals $(h_1(x), h_2(x), \dots, h_d(x))$

This vector characterizes the curvature intensity of the current point in each coordinate direction.

4.2 Construction of the Refraction Factor

To avoid excessive displacement in high curvature directions, a refraction factor is introduced.

$T_j(x, t)$ equals

$1 \text{ divided by } (1 + \text{abs}(h_j(x)))$

This function satisfies the following properties:

First, when $\text{abs}(h_j(x))$ is large, T_j approaches 0;

Second, when $\text{abs}(h_j(x))$ approaches 0, T_j approaches 1;

Third, T_j always lies between 0 and 1.

Therefore, T_j acts as a scaling factor, automatically reducing the step size in high curvature directions and maintaining a larger step size in low curvature directions.

Define the refraction modulation vector:

$T(x, t)$ equals $(T_1(x, t), T_2(x, t), \dots, T_d(x, t))$

4.3 Flow field after refraction

The original flow field is $F(x,t)$. Introducing a dimension-by-dimensional modulation mechanism, the flow field after refraction is defined as:

$F_{\text{star}}(x,t)$ equals

$T(x,t)$ multiplied dimension by $F(x,t)$

That is, the j -th dimension component is

$F_{\text{star}_j}(x,t)$ equals $T_j(x,t)$ multiplied by $F_j(x,t)$

This structure is geometrically similar to the phenomenon of light refraction in different media. If $F(x,t)$ is considered as the incident direction, then the modulation of $T(x,t)$ is equivalent to changing the propagation speed according to the local curvature.

From a stability perspective, the refraction mechanism is equivalent to introducing a diagonal matrix $D(x,t)$, whose diagonal elements are $T_j(x,t)$, thus transforming the system into:

F_{star} equals $D(x,t)$ multiplied by $F(x,t)$

Since the spectral radius of D is less than or equal to 1, the system update step size is naturally constrained, helping to avoid oscillations.

V. Second-Order Dynamical Systems

Most optimization algorithms use a first-order update form, where position is directly determined by the current state. However, in real physical systems, motion typically follows second-order dynamics, where position change is determined by velocity, and velocity is determined by force.

To enhance the system's traversal capability, a velocity variable $v_i(t)$ is introduced.

5.1 Velocity Update

The velocity update formula is defined as:

$v_i(t+1)$ equals

μ multiplied by $v_i(t)$

plus η multiplied by $F_{\text{star}}(x_i(t), t)$

Where μ is the inertia coefficient, typically ranging from 0 to 1; η is the flow field coupling strength.

If μ equals 0, the system degenerates into a first-order model; if μ is close to 1, the system has strong inertia.

5.2 Position Update

The position update is:

$x_i(t + 1)$ equals

$x_i(t)$

plus $v_i(t + 1)$

The entire system constitutes a discrete second-order nonlinear dynamic system.

5.3 Dynamic Interpretation

Writing the above update formula in single-variable form, we get:

$x_i(t + 1) - x_i(t)$

equals

μ multiplied by $(x_i(t) - x_i(t - 1))$

plus η multiplied by $F_{\text{star}}(x_i(t), t)$

This expression shows that the current position is determined by both historical displacement and the current flow field. If a downward trend persists in a certain direction, the velocity will gradually accumulate, thus enabling the system to cross narrow potential wells.

The second-order structure also gives the system oscillatory and penetrating characteristics, which are significant in complex multimodal functions.

VI. Energy Evolution and Conservation Violation

To further construct a non-equilibrium system, an individual energy variable $E_i(t)$ is introduced. Energy is used to regulate individual activity and the overall exploration intensity of the system.

6.1 Individual Energy Update

Let the velocity modulus be

$V_i(t)$ equal to the norm of $F_{\text{star}}(x_i(t), t)$

Define the energy update formula:

$E_i(t + 1)$ equals

ρ multiplied by $E_i(t)$

minus β multiplied by $V_i(t)$

plus γ multiplied by $(f(x_i(t - 1)) - f(x_i(t)))$

Where:

ρ is the energy retention coefficient, less than 1 indicates dissipation;

β controls flow consumption;

γ is the improvement reward coefficient.

If an individual obtains a function value improvement in the current iteration, then $f(x_i(t - 1)) - f(x_i(t))$ is positive, thus increasing energy.

6.2 Total Energy and Energy Conservation Violation

Define the total energy of the system

Total_E(t) equals

Sum of $E_i(t)$ from 1 to N

Under the dissipative mechanism, the total energy usually decreases over time. If Total_E(t) is too low, the system may fall into a frozen state.

Therefore, an energy injection mechanism is introduced.

If

Total_E(t) is less than θ_E multiplied by the initial total energy

Then execute

$E_i(t+1)$ equals

$E_i(t+1)$

Add λ multiplied by (initial total energy minus $Total_E(t)$)

This mechanism breaks energy conservation, making the system an externally driven non-equilibrium system.

From a statistical physics perspective, this system is similar to a dissipative structure, maintaining a dynamic evolution far from equilibrium through energy input.

VII. Entropy-Driven Population Splitting

In the later stages of optimization, the population may concentrate in a certain region, leading to a decrease in diversity. To quantify the uniformity of distribution, positional entropy is introduced.

7.1 Entropy Definition

The search space is divided into several grids, with the proportion of individuals in the k -th grid being p_k .

Define positional entropy:

$S(t)$ equals

the negative sum of k multiplied by p_k , $\log(p_k)$.

If all individuals are concentrated in a single region, $S(t)$ is close to 0; if uniformly distributed, $S(t)$ is larger.

7.2 Splitting Condition

When

$S(t)$ is less than θ_S

and

the global optimum has not significantly improved within τ generations,

then a split operation is performed.

7.3 Splitting and Merging

The population is divided into several subgroups using a clustering method, and each subgroup evolves independently. The independent stage can last for L generations.

Then the subgroups are merged, retaining the best individuals from each subgroup.

This process can be represented as follows:

The overall population

splits into a set of subgroups

each subgroup updates independently

they merge into a new overall population

This periodic structure increases diversity and avoids getting trapped in a single extreme region.

VIII. Algorithm Flow

Step 1: Initialize $x_i(0)$, $v_i(0)$, $E_i(0)$

Step 2: Calculate the gradient approximation G

Step 3: Calculate the curl term A

Step 4: Calculate the backflow term R

Step 5: Construct F equal to negative G plus A plus R

Step 6: Calculate curvature h_j and refractive factor T

Step 7: Calculate F_{star} equal to T multiplied by F dimension by dimension

Step 8: Update velocity v_i

Step 9: Update position x_i

Step 10: Update energy E_i

Step 11: Determine Total_E and perform energy injection

Step 12: Calculate position entropy $S(t)$

Step 13: Split if the condition is met

Step 14: Update the historical memory set

Step 15: Determine the termination condition

IX. Stability Analysis

Ignoring curl and backflow terms, the system is approximated as:

$$v(t+1) = \mu v(t) - \eta T(x) G(x)$$

If μ If η is less than 1 and sufficiently small, the system can be considered a discrete damped oscillator system, locally stable.

Adding a curl term transforms the system into a non-conservative system. However, since

the transpose of G multiplied by S multiplied by G equals 0,

the curl term does not change the decreasing trend of the function value under the first-order approximation.

The refraction factor T_j lies between 0 and 1, making the update step bounded.

The energy dissipation coefficient ρ is less than 1, ensuring that the individual energy does not grow indefinitely.

Therefore, within a reasonable parameter range, the system is a bounded dissipative system.

X. Time Complexity Analysis

For each individual, each generation requires the following calculations:

Gradient approximation: d times

Curvature approximation: d times

Curvature matrix multiplication: approximately d^2

Therefore, the time complexity per generation is:

$$O(N \times d^2)$$

If the total number of iterations is T , then the total complexity is:

$O(T \times N \times d^2)$

XI. Discussion of Method Characteristics

First, the curl structure makes the flow field non-conservative;

Second, curvature refraction improves geometric adaptability;

Third, second-order dynamics enhances traversal capability;

Fourth, the recirculation kernel achieves continuous memory;

Fifth, the entropy-driven mechanism maintains diversity;

Sixth, energy violation constructs a non-equilibrium evolution system.

XII. Conclusion

This paper systematically constructs a rainbow trout optimization algorithm based on a non-conservative flow field, curvature refraction, and energy non-equilibrium mechanisms. The algorithm breaks the traditional conservative descent model through curl coupling and recirculation structure, and enhances the system's exploration capability with the help of second-order dynamics and energy evolution mechanisms. Theoretical analysis shows that this method belongs to the dissipative nonequilibrium structure at the dynamical system level, possessing good stability and expressive power. Further research can focus on its continuous-time limiting form and rigorous convergence proof.

References

1. Ali, E. S., Abd Elazim, S. M., & Abdelaziz, A. Y. (2017). Ant Lion Optimization Algorithm for optimal location and sizing of renewable distributed generations. *Renewable Energy*, 101, 1311-1324.
2. Hussien, A. G., Hassanien, A. E., Houssein, E. H., Bhattacharyya, S., & Amin, M. (2018). S-shaped binary whale optimization algorithm for feature selection. In *Recent trends in signal and image processing: ISSIP 2017* (pp. 79-87). Singapore: Springer Singapore.
3. Wang, J., Du, P., Niu, T., & Yang, W. (2017). A novel hybrid system based on a new proposed algorithm — Multi-Objective Whale Optimization Algorithm for wind speed forecasting. *Applied energy*, 208, 344-360.
4. Pilanci, M., & Wainwright, M. J. (2017). Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence. *SIAM Journal on Optimization*, 27(1), 205-245.
5. Xiong, G., Zhang, J., Shi, D., & He, Y. (2018). Parameter extraction of solar photovoltaic models using an improved whale optimization algorithm. *Energy conversion and management*, 174, 388-405.

6. Bull, A. D. (2011). Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(10).
7. Ishaque, K., Salam, Z., Shamsudin, A., & Amjad, M. (2012). A direct control based maximum power point tracking method for photovoltaic system under partial shading conditions using particle swarm optimization algorithm. *Applied Energy*, 99, 414-422.
8. Zhao, X., Zhang, C., Yang, B., & Li, P. (2011). Adaptive knot placement using a GMM-based continuous optimization algorithm in B-spline curve approximation. *Computer-Aided Design*, 43(6), 598-604.
9. Wu, J., Wang, H., Li, N., Yao, P., Huang, Y., Su, Z., & Yu, Y. (2017). Distributed trajectory optimization for multiple solar-powered UAVs target tracking in urban environment by Adaptive Grasshopper Optimization Algorithm. *Aerospace Science and Technology*, 70, 497-510.
10. Zhang, J. Forest Rattlesnake Optimization Algorithm. <https://doi.org/10.5281/zenodo.18184545>
11. Nouri, M., Bekrar, A., Jemai, A., Niar, S., & Ammari, A. C. (2018). An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem. *Journal of Intelligent Manufacturing*, 29(3), 603-615.
12. Elymany, M. M., Enany, M. A., & Elsonbaty, N. A. (2024). Hybrid optimized-ANFIS based MPPT for hybrid microgrid using zebra optimization algorithm and artificial gorilla troops optimizer. *Energy Conversion and Management*, 299, 117809.
13. Borousan, F., & Hamidan, M. A. (2023). Distributed power generation planning for distribution network using chimp optimization algorithm in order to reliability improvement. *Electric Power Systems Research*, 217, 109109.
14. MiarNaeimi, F., Azizyan, G., & Rashki, M. (2021). Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems. *Knowledge-Based Systems*, 213, 106711.
15. Bakır, H. (2024). Dynamic fitness-distance balance-based artificial rabbits optimization algorithm to solve optimal power flow problem. *Expert Systems with Applications*, 240, 122460.
16. Zamani, H., Nadimi-Shahraki, M. H., & Gandomi, A. H. (2021). QANA: Quantum-based avian navigation optimizer algorithm. *Engineering Applications of Artificial Intelligence*, 104, 104314.
17. Snášel, V., Rizk-Allah, R. M., Izci, D., & Ekinici, S. (2023). Weighted mean of vectors optimization algorithm and its application in designing the power system stabilizer. *Applied Soft Computing*, 136, 110085.
18. Gezici, H., & Livatyali, H. (2022). Chaotic Harris hawks optimization algorithm. *Journal of Computational Design and Engineering*, 9(1), 216-245.
19. Wen, C., Jia, H., Wu, D., Rao, H., Li, S., Liu, Q., & Abualigah, L. (2022). Modified remora optimization algorithm with multistrategies for global optimization problem. *Mathematics*, 10(19), 3604.
20. Niu, Q., Zhang, L., & Li, K. (2014). A biogeography-based optimization algorithm with mutation strategies for model parameter estimation of solar and fuel cells. *Energy conversion and management*, 86, 1173-1185.
21. Mavrovouniotis, M., Li, C., & Yang, S. (2017). A survey of swarm intelligence for dynamic optimization: Algorithms and applications. *Swarm and Evolutionary Computation*, 33, 1-17.
22. Song, J., Wang, J., & Lu, H. (2018). A novel combined model based on advanced optimization algorithm for short-term wind speed forecasting. *Applied energy*, 215, 643-658.

23. Nazari-Heris, M., Mohammadi-Ivatloo, B., & Gharehpetian, G. B. (2018). A comprehensive review of heuristic optimization algorithms for optimal combined heat and power dispatch from economic and environmental perspectives. *Renewable and Sustainable Energy Reviews*, 81, 2128-2143.
24. Sahu, B. K., Pati, S., Mohanty, P. K., & Panda, S. (2015). Teaching – learning based optimization algorithm based fuzzy-PID controller for automatic generation control of multi-area power system. *Applied Soft Computing*, 27, 240-249.
25. Komaki, G. M., & Kayvanfar, V. (2015). Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time. *Journal of computational science*, 8, 109-120.
26. Devabalaji, K. R., Ravi, K., & Kothari, D. P. (2015). Optimal location and sizing of capacitor placement in radial distribution system using bacterial foraging optimization algorithm. *International Journal of Electrical Power & Energy Systems*, 71, 383-390.
27. Devabalaji, K. R., & Ravi, K. (2016). Optimal size and siting of multiple DG and DSTATCOM in radial distribution system using bacterial foraging optimization algorithm. *Ain Shams Engineering Journal*, 7(3), 959-971.
28. Shakarami, M. R., & Davoudkhani, I. F. (2016). Wide-area power system stabilizer design based on grey wolf optimization algorithm considering the time delay. *Electric Power Systems Research*, 133, 149-159.
29. Ali, E. S., & Abd-Elazim, S. M. (2011). Bacteria foraging optimization algorithm based load frequency controller for interconnected power system. *International Journal of Electrical Power & Energy Systems*, 33(3), 633-638.
30. Liao, W. H., Kao, Y., & Li, Y. S. (2011). A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks. *Expert Systems with Applications*, 38(10), 12180-12188.
31. Zhang, J. Arboreal Monitor Lizard Optimization Algorithm. <https://doi.org/10.5281/zenodo.18184577>
32. Yildiz, A. R. (2012). A comparative study of population-based optimization algorithms for turning operations. *Information Sciences*, 210, 81-88.
33. Kumar, K. S., & Jayabarathi, T. (2012). Power system reconfiguration and loss minimization for an distribution systems using bacterial foraging optimization algorithm. *International Journal of Electrical Power & Energy Systems*, 36(1), 13-17.
34. Niknam, T., Taheri, S. I., Aghaei, J., Tabatabaei, S., & Nayeripour, M. (2011). A modified honey bee mating optimization algorithm for multiobjective placement of renewable energy resources. *Applied energy*, 88(12), 4817-4830.
35. Zhang, J. Bird-of-Paradise Optimization Algorithm. <https://doi.org/10.5281/zenodo.18184522>
36. Niu, Q., Zhang, L., & Li, K. (2014). A biogeography-based optimization algorithm with mutation strategies for model parameter estimation of solar and fuel cells. *Energy conversion and management*, 86, 1173-1185.
37. Ting, C. J., & Chen, C. H. (2013). A multiple ant colony optimization algorithm for the capacitated location routing problem. *International journal of production economics*, 141(1), 34-44.
38. Cuevas, E., Echavarría, A., & Ramírez-Ortegón, M. A. (2014). An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation. *Applied intelligence*, 40(2), 256-272.
39. Jain, H., & Deb, K. (2013). An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and

- extending to an adaptive approach. *IEEE Transactions on evolutionary computation*, 18(4), 602-622.
40. Zhang, J. Water Monitor Lizard Optimization Algorithm. <https://doi.org/10.5281/zenodo.18184604>
41. Sharma, Y., & Saikia, L. C. (2015). Automatic generation control of a multi-area ST–Thermal power system using Grey Wolf Optimizer algorithm based classical controllers. *International Journal of Electrical Power & Energy Systems*, 73, 853-862.
42. Lynn, N., & Suganthan, P. N. (2017). Ensemble particle swarm optimizer. *Applied Soft Computing*, 55, 533-548.
43. ben Oualid Medani, K., Sayah, S., & Bekrar, A. (2018). Whale optimization algorithm based optimal reactive power dispatch: A case study of the Algerian power system. *Electric Power Systems Research*, 163, 696-705.
44. Amaran, S., Sahinidis, N. V., Sharda, B., & Bury, S. J. (2016). Simulation optimization: a review of algorithms and applications. *Annals of Operations Research*, 240(1), 351-380.
45. Zheng, X. L., & Wang, L. (2016). A collaborative multiobjective fruit fly optimization algorithm for the resource constrained unrelated parallel machine green scheduling problem. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(5), 790-800.
46. Fathy, A. (2016). A reliable methodology based on mine blast optimization algorithm for optimal sizing of hybrid PV-wind-FC system for remote area in Egypt. *Renewable energy*, 95, 367-380.
47. Mann, Z. Á. (2015). Allocation of virtual machines in cloud data centers—a survey of problem models and optimization algorithms. *Acm Computing Surveys (CSUR)*, 48(1), 1-34.