Research Article

Rider Optimization Algorithm (ROA): An optimization solution for engineering problem

Kumar Rahul[a], Rohitash Kumar[b], Banyal[c]

[a]Department of Basic and Applied Science, NIFTEM, Sonipat 131028, India
[b]Department of Computer Science and Engineering, Rajasthan Technical University, Kota 324010, India,
kumarrahul.niftem@gmail.com, rkbayal@gmail.com

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 23 May 2021

Abstract: Optimization is a process and instrument applied to solve an engineering problem. Rider optimization algorithm (ROA) is a new optimization technique used to find the target location through different components like bypass rider, follower, overtake, and attackers' parameters. An optimization problem can be solved through an objective function. The inspirational ROA development is based on the riders who proceed towards target locations to win the race. Any optimization problem is either nature-inspired or artificial intelligence techniques. Some of the nature-inspired algorithms are firefly algorithm (FA), ant colony optimization (ACO), particle swarm optimization (PSO), grey wolf optimization (GWO), spider monkey optimization (SMO), and so on. For solving engineering problems, multi-objective optimizations are used for engineering design, scientific experiments and decision making, etc. This paper introduces ROA, its application in industries, and usability in engineering and industrial problem. The purpose of this paper is to identify the importance and understanding of the optimization-related problem.

Keywords: optimization, rider, overtaker, attacker, convergence analysis

1. Introduction

Problem formulation of any optimization problem encompasses variables, constraint, objective functions, algorithm selections, design, and optimal solutions (Formulation & Optimization, n.d.). ROA is defined through four riders such as bypass rider, follower, overtake, and attackers. Each of the riders moves towards the target location with some strategies (Binu & Kariyappa, 2019): -

1) Bypass rider having an objective to reach the target location by bypassing leading riders.
2) The follower will follow the leading rider and will not disobey the leading rider.
3) Overtaker follows its path to reach the target location keeping in mind the leading rider.
4) Attackers follow the maximum speed concept and move to the target location.

The ROA was developed in 2018 by D. Binu. ROA defined through an algorithm consisting of the following steps: rider parameter optimization, finding the success rate, leading rider, position update of rider (through update process of bypass rider, follower, overtaker, attacker), rider parameter update (through activity counter, steering angle, gear, accelerator, and brake) (Binu & Kariyappa, 2019). It would cover up through group initialization, denoted as given equation as per (Binu & Kariyappa, 2019).

$$A_t = \{A_t(l, m)\}; 1 \leq l \leq R; 1 \leq m \leq Q \quad (1)$$

Here, A_t is group initialization, and R is the number of riders. Rider R can be calculated as count the total number of riders available in each group. $A_t(l, m)$ is represented as $l\ th$ rider with time instant $t. Q$ defined as coordinates or dimensions. So, it would be as (Binu & Kariyappa, 2019):

$$R = P + L + T + K \quad (2)$$

Where P defined as a bypass rider (in numbers), L defined as a follower (in numbers), T defined as overtaker (in numbers), and K defined as an attacker (in numbers). Therefore, the relationship exists in the form of (Binu & Kariyappa, 2019):

$$P = L = T = K = R/4 \quad (3)$$

ROA success depends on the number of participants, their possible matches, and in case of non-availability of matches, they stop their participation (Aydin, Gokasar, & Kalan, 2020). For dynamic route-finding in traffic congestion, the ROA is used to match people and similar routes, schedule the route on short notice, etc. (Aydin et al., 2020). Based on the relationship between bypass rider, follower, overtaker, and attacker, the rider position defined in the form of $[A_i, A_{R/4}]$, $[A_{\frac{R}{4}+1}, A_{R/2}]$, $[A_{\frac{R}{2}+1}, A_{3R/4}]$, and $[A_{\frac{3R}{4}+1}, A_R]$ respectively. The steering angle S of a rider within time t can be defined as (Binu & Kariyappa, 2019):

$$S_t = \{S_t^{i,j}\} \qquad 1 \leq i \leq R; 1 \leq j \leq Q \quad (4)$$

Where, $\{S_t^{i,j}\}$ is the steering angle of the i^{th}rider's vehicle. The initial steering angle is given as (Binu & Kariyappa, 2019):

$$S_{i,j} = \begin{cases} \theta_i & if\ j = 1 \\ S_{i,j-1} + \varphi & if\ j \neq 1\ and\ S_{i,j-1} + \varphi\ \leq 360 \\ S_{i,j-1} + \varphi - 360 & otherwise \end{cases} \quad (5)$$

Here, $\theta_i = i * \frac{360^o}{R}$ and $\varphi = \frac{360}{Q}$, R is number of riders, φ is coordinate angles and θ_i is the positioning angle of the i^{th} rider vehicle. The gear, accelerator, and brake defined in the form of (Binu & Kariyappa, 2019):

$$G = \{G_i\};\ \ where,\ \ 1 < i < R \quad (6)$$

$$A = \{A_i\};\ \ where,\ \ 1 < i < R \quad (7)$$

$$B = \{B_i\};\ \ where,\ \ 1 < i < R \quad (8)$$

Here, G_i is the gear of the i^{th}rider, which values exist between [0,1,2,3,4]. Similarly, A_i is the accelerator of the i^{th}rider, which values exist between [0,1] and B_i is the accelerator of the i^{th} rider, which values exist between [0,1]. The maximum speed, an i^{th}rider can drive will be calculated as (Binu & Kariyappa, 2019):

$$X_{max}^i = \frac{M_D^i - M_E^i}{T_{off}} \quad (9)$$

The rider has a maximum and minimum value. The rider uses a particular speed (weight) to reach the target. Here, M_D^i is the maximum value of the i^{th}rider, and M_E^i is the minimum value of the i^{th}rider, and T_{off} is the maximum time, which is allotted the rider to reach the target position. The gear speed limit for the i^{th}rider can be calculated as (Binu & Kariyappa, 2019):

$$X_i^E = \frac{X_{max}^i}{|G|} \quad (10)$$

Here, X_{max}^i is the maximum speed of the i^{th}rider and $|G|$ is counted as the number of gears. The success rate is another critical parameter in finding the success of the i^{th}rider by evaluating the differences between the rider's position and target position.

The success rate will be calculated as (Binu & Kariyappa, 2019):

$$S_r = \frac{1}{||X_i - P_t||} \quad (11)$$

Here, X_i represented as i^{th}rider position and P_t is represented as the target position. The success rate determines the leading rider, who is close to the target location. The leading rider reaches the target location as compared to the others. Any rider can become a leading rider after that the position of the leading rider change. This leads to update the rider's position, bypass rider, follower, overtaker, and attackers. Bypass rider update defines in such a way, where the bypass rider does not follow the standard path and leading rider as well. Its positions are random and calculated as (Binu & Kariyappa, 2019):

$$A_{t+1}^{B}\ (l, m) = \delta[(A_t(d, m) * \beta(m) + A_t(\xi, m) * [1 - \beta(m)]] \quad (12)$$

Here, (l, m) represents l^{th}rider with time instant $t + 1$ with m^{th}coordinate, and $1 < \delta < 1, 1 < d < R, 1 < \xi < R, 0 < \beta < 1$, and $\delta, d, \xi\ and\ \beta$ are a random number.

Followers updated his positions following the rider's positions as per given equations where coordinate selector becomes important. It will be calculated as (Binu & Kariyappa, 2019):

$$A_{t+1}^{F}\ (l, k) = A^E(E, k) + \cos[(\mathrm{T}_{l,k}^{t}) * A^E(E, k) * d_l^t] \quad (13)$$

Here, k is coordinate selector, A^E is the leading rider (position), rider index represented through E. $\mathrm{T}_{l,k}^{t}$ is the steering angle of a l^{th}rider in the k^{th}coordinate along with distance traveled by the l^{th}rider, which calculated as the multiplication of rider velocity and off-time (rate) (Binu & Kariyappa, 2019), i.e.

$$d_l^t = v_l^t * (\frac{1}{T_{off}}) \quad (14)$$

Here, d_l^t is representing distance has to cover by l^{th} rider and v_l^t representing the velocity of the l^{th} rider. Overtaker updated process depends on the coordinate selector, direction indicator $D_t^I(i)$, and relative success rate (Binu & Kariyappa, 2019).

$$A_{t+1}^{o}(l,k) = A_t(l,k) + [D_t^I(l) * A^E(E,k)] \tag{15}$$

Here, $A_t(l,k)$ represents the position of the lth rider in the k^{th} coordinate and $D_t^I(l)$ represents direction indicator. Similarly, attackers try to take the leading position in the same fashion of followers with coordinates update, and it will be expressed through the equations mentioned below (Binu & Kariyappa, 2019).

$$A_{t+1}^{A}(l,m) = A^E(E,m) + \left[\cos\left(T_{l,m}^t\right) * A^E(E,m)\right] + d_l^t \tag{16}$$

2. Related Work

Optimization is a technology of process by which the value of variables brings optimal a function's optimal solutions (Cavazzuti, 2013). The stochastic problem are categorized in the form of simulated annealing(SA), particle swarm optimization (PSO), evolutionary algorithm (EA), game theory-based optimization (GTBO), and genetic algorithm (GA) (Cavazzuti, 2013). Stochastic optimizations include randomness that is important for the search procedure (Cavazzuti, 2013). The optimization problem can be understood through an optimization flowchart, where it starts from identifying the problem, input parameter and variables, optimization loop, and soon as per given in (Cavazzuti, 2013). Stochastic optimization is a powerful tool used to manage enterprise resources. Some factors affecting optimizations are complicating optimization factors, constrained versus unconstrained optimization, differential calculus, the process of differentiation, etc. (Seifi & Sepasian, 2011). The optimization search process leads to a stochastic concept, called stochastic optimization problem, where randomness exists. It is classified into various segments comprised of simulated annealing (SA), particle swarm optimization (PSO), evolutionary algorithm (EA), genetic algorithm (GA), etc., as per (Cavazzuti, 2013). Stochastic problems are single objective or multi-objective problems. ROA has been improved with fitness function. Fitness-oriented ROA (FO-ROA) algorithm developed by S Ravikumar, H Vennila, and R Deepak (Ravikumar, Vennila, & Deepak, 2020)**.**

For achieving a higher and optimized control system for power generation, the ROA algorithm has been improved and modified (Ravikumar et al., 2020). FO-ROA is used in renewable energy technologies to enhance performances through the hybrid power generating system (Ravikumar et al., 2020). In-plant disease detection, ROA is integrated with the Cuckoo Search Algorithm (CSA), where the image is used as preprocessing to the removal of noise before proceeding to segmentation using fuzzy logic (Cristin, Kumar, Priya, & Karthick, 2020).

With the implementation of ROA-CSA methodologies, food production enhancement was achieved and reduced wastages in organizations (Cristin et al., 2020). It decreases disease spread among plant and predict diseases detection. Plant diseases spread badly affect the potential growth of plant within an agricultural-based product, which affect the economic development of human in developing countries (Cristin et al., 2020). ROA-CSA is implemented through a deep belief network (DBN) which is used for plant disease detection (Cristin et al., 2020). The ROA-CSA technologies (algorithm) have been used for identifying plant disease detections through image processing (Cristin et al., 2020).

Similarly, Bypass linked attacker updated (BLA) is a metaheuristics algorithm. It is used to find the shortest path considering various points, including transmission delay, the distance between nodes, packet ratio losses, etc. (Aditya Sai Srinivas & Manivannan, 2020). BLA-ROA have been discussed for optimizing the deep belief network (DBN) by the authors (Aditya Sai Srinivas & Manivannan, 2020).

An experimental analysis has been carried out, where the proposed BLA-ROA algorithm has a higher constant function (28.1%) than the dragonfly deer hunting optimization algorithm (D-DHOA) (Aditya Sai Srinivas & Manivannan, 2020). BLA-ROU applies to IoT-based applications where comprehensive data is managed through cloud services and protected through the HELLO flood attack (Aditya Sai Srinivas & Manivannan, 2020). In IoT-enabled devices and engineering devices, the network flooded with the HELLO packet. HELLO, the packet is used to check the vulnerability in communications. It sent routinely in the open shortest path, so the HELLO flood attack is defended and detected by the novel approach of the BLA-ROA algorithm (Aditya Sai Srinivas & Manivannan, 2020).

Improved ROA is used to enhanced medical image compression through segmentation, image compression, encoding methods, merged Huffman coding (Saini, Sayal, & Rawat, 2019). Improvement in steering and gear (ISG-ROA) methodologies proposed with enhancement in steering angle and gear with $S_c^{\tau+1}(i) = 1$ and $S_c^{\tau+1}(i) = 0$ for best solutions. Similarly, document clustering is an important activity in data science where voluminous data being generated. Document segmentation carried out through ROA and moth search algorithm

(ROA-MSA), which are further evaluated by precision, accuracy, recall, F score etc.(Yarlagadda, Gangadhara Rao, & Srikrishna, 2019).

The proposed ROA-MSA algorithm consists of preprocessing (where redundant data removes from the text documents), semantic level feature extraction, feature selection followed by document clustering through moth rider optimization (MRO) (Yarlagadda et al., 2019). An optimal location is identified to improve power quality through a ROA (Gaddala & Raju, 2020). ROA is merged with a gravitational search called the gravitational search strategy ROA (GSSROA) developed and designed by the authors(Wang, Yuan, & Guo, 2019).

3. Materials And Methods

A problem is solved by the (ROA) and its association with another algorithm or improved algorithm. Plant disease detection in the agricultural domain starts from preprocessing of removing noise from the images taken and classifications of images. Deep Believe Network (DBN) based ROA-CSA (Cuckoo Search Algorithm) proposed and solved the plant disease problems (Cristin et al., 2020). The pseudo-code for ROA is given as per:

Table 1. The ROA (Binu & Kariyappa, 2019)

Steps	Descriptions
Step 1:	Input: Population initialization along with rider's location X_i
Step 2:	Output: Leading rider X^l
Step 3:	Rider parameter initialization like steering angle S ,gear G , accelerator A, and brake B
Step 4:	Find out the $S_r = \frac{1}{\|\|X_i - P_t\|\|}$
Step 5:	While $i < T_{off}$
Step 6:	For $t = 1\ to\ R$
Step 7:	Update the position of the Bypass rider through equation (12)
Step 8:	Update the position of Followers through equation (13)
Step 9:	Update the position of Overtaker through equation (15)
Step 10:	Update the position of Attacker through equation (16)
Step 11:	Ranking of the rider based on S_r
Step 12:	Find the rider where $S_r = maximum\ value$ consider as leading rider
Step 13:	Update $S, G, A\ and\ B$
Step 14:	Return X^l and $t = t + 1$
Step 15:	End For
Step 16:	End While
Step 17:	Exit

4. Implementation And Discussion

ROA implemented successfully in healthcare sector where medical image compressions and segmentations used effectively (Saini et al., 2019). The ROA is implemented for route splitting to remove traffic congestion given by the authors (Aydin et al., 2020). Rider matching algorithm is used to maximize travel distance saving, this is one of the other ROA applications discussed in (Aydin et al., 2020). The ROA can be implemented in MATLAB 2018a, 2018b, or 2019a versions, and results would be compared with the other optimization process. In the case of power generations, the convergence analysis results evaluated through MATLAB 2018a and compared with genetic algorithm (GA), firefly algorithm (FA), grey wolf optimization (GWO) as mentioned in (Ravikumar et al., 2020).

5. Conclusions

The performance of ROA is calculated in terms of convergence analysis, confusion parameter analysis, input voltage, and input frequency. Through ROA, sensitivity analysis and promptness analysis can be explored for various optimization and engineering problems. ROA can be merged with the other optimization techniques to show better results and provide efficiency in evaluating and establishing centroid selection in clustering and other data analysis processes. This ROA would be one solution for optimization, cluster analysis in any analytical problem under an application-based project.

6. Acknowledgment

This study has been carried out in NIFTEM to adhere to data cleaning in the big data-based application. The authors are grateful for the institution and university (Rajasthan Technical University, India) to use resources in all possible manner. The authors are having no conflict of interest.

References

1. Aditya Sai Srinivas, T., & Manivannan, S. S. (2020). Prevention of hello flood attack in IoT using combination of deep learning with improved rider optimization algorithm. Computer Communications. https://doi.org/10.1016/j.comcom.2020.03.031
2. Aydin, O. F., Gokasar, I., & Kalan, O. (2020). Matching algorithm for improving ridesharing by incorporating route splits and social factors. PLoS ONE, 15(3), 1–23. https://doi.org/10.1371/journal.pone.0229674
3. Binu, D., & Kariyappa, B. S. (2019). RideNN: A New Rider Optimization Algorithm-Based Neural Network for Fault Diagnosis in Analog Circuits. IEEE Transactions on Instrumentation and Measurement, 68(1), 2–26. https://doi.org/10.1109/TIM.2018.2836058
4. Cavazzuti, M. (2013). Optimization methods: From theory to design scientific and technological aspects in mechanics. Optimization Methods: From Theory to Design Scientific and Technological Aspects in Mechanics, 1–262. https://doi.org/10.1007/978-3-642-31187-1
5. Cristin, R., Kumar, B. S., Priya, C., & Karthick, K. (2020). Deep neural network based Rider-Cuckoo Search Algorithm for plant disease detection. Artificial Intelligence Review, (0123456789). https://doi.org/10.1007/s10462-020-09813-w
6. Formulation, P., & Optimization, O. F. (n.d.). CHAPTER 4. 87–122.
7. Gaddala, K., & Raju, P. S. (2020). Optimal location of UPQC for power quality improvement: novel hybrid approach. Journal of Engineering, Design and Technology. https://doi.org/10.1108/JEDT-04-2019-0113
8. Ravikumar, S., Vennila, H., & Deepak, R. (2020). Hybrid power generation system with Total Harmonic Distortion minimization using improved Rider Optimization Algorithm: Analysis on converters. Journal of Power Sources, 459(March), 228025. https://doi.org/10.1016/j.jpowsour.2020.228025
9. Saini, H. S., Sayal, R., & Rawat, S. S. (2019). Innovations in Computer Science and Engineering (Vol. 32). https://doi.org/10.1007/978-981-10-8201-6
10. Seifi, H., & Sepasian, M. S. (2011). Optimization Techniques. Power Systems, 49, 15–30. https://doi.org/10.1007/978-3-642-17989-1_2
11. Wang, G., Yuan, Y., & Guo, W. (2019). An Improved Rider Optimization Algorithm for Solving Engineering Optimization Problems. IEEE Access, 7, 80570–80576. https://doi.org/10.1109/ACCESS.2019.2923468
12. Yarlagadda, M., Gangadhara Rao, K., & Srikrishna, A. (2019). Frequent itemset-based feature selection and Rider Moth Search Algorithm for document clustering. Journal of King Saud University - Computer and Information Sciences, (xxxx). https://doi.org/10.1016/j.jksuci.2019.09.002