DPSO: A Hybrid Approach for Load Balancing using Dragonfly and PSO Algorithm in Cloud Computing Environment

Subasish Mohapatra¹, Subhadarshini Mohanty¹, Hriteek Kumar Nayak¹, Milan Kumar Mallick¹, Janjhyam Venkata Naga Ramesh², Khasim Vali Dudekula³

¹ School of Computer Sciences, Odisha University of Technology and Research, Bhubaneswar, India
² Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur Dist, Andhra Pradesh, 522302, India
³ School of Computer Science & Engineering, VIT-AP University, Andhra Pradesh, India

Abstract

Load balancing is one of the promising challenges in cloud computing system. For solving the issues, many heuristic, meta-heuristic, evolutionary and hybrid algorithms have been proposed by the researchers. Still, it is under way of research for finding optimal solution in dynamic change in behaviour of task as well as computing environments. Attempts have been made to develop a hybrid framework to balance the load in cloud environment by obtain the best fitness value. To achieve an optimal resource for load balancing, the proposed framework integrates Dragonfly (DF) and Particle Swarm Optimization (PSO) algorithm. The performance of the proposed method is compared with PSO and Dragonfly algorithm. The performance is evaluated in different measures such as best fitness value, response time by varying the user base and response time. The user bases are varied from 50, 100, 500, and 1000. Similarly, the population size has been varied to observe the performance of the algorithm. It is observed that the proposed method outperforms the other approached for load balancing. The statistical analysis and standard testing also validate the relative superiority of PSO a considerable Dragonfly Algorithm. The hybrid approach provides better response time.

Keywords: Resource Allocation, Load Balancing, Cloud Computing, Dragonfly Algorithm, PSO, Hybrid model

Received on 21 October 2023, accepted on 03 January 2024, published on 11 January 2024

Copyright © 2024 S. Mohapatra et al., licensed to EAI. This is an open access article distributed under the terms of the CC BY-NC-SA 4.0, which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/eetiot.4826

*Corresponding author Email: smohapatra@outr.ac.in

1. Introduction

Optimization problems arise in numerous fields, such as engineering, economics, logistics, and data analysis. The quest for finding optimal solutions to these problems has driven researchers to explore various optimization techniques. Nature-inspired algorithms have emerged as a powerful tool in this context, as they draw inspiration from the behaviour of biological systems to solve complex problems efficiently. In this paper, we propose a hybrid approach, called Hybrid Dragonfly-Particle Swarm Optimization (DPSO), that combines the strengths of PSO and DA to create a more robust and efficient optimization algorithm [1]. The motivation behind this hybridization is to exploit the complementary features of both algorithms, aiming to improve the overall performance in different measures. DPSO algorithm operates by maintaining two separate populations of particles, one following the PSO behaviour and the other following the DA behaviour. These populations interact and exchange information during the search process, allowing them to benefit from each other's strengths. The PSO component facilitates effective exploitation, while the DA component enhances exploration capabilities. Furthermore, we introduce adaptive mechanisms within DPSO to dynamically adjust the interaction between the PSO and DA components. This adaptively enables DPSO to fine-tune its behaviour based on the characteristics of the optimization problem being solved, thus enhancing its performance and
adaptability. To evaluate the effectiveness of DPSO, we conduct extensive experiments on a set of benchmark optimization problems. Comparative analyses are performed against PSO, DA, and other state-of-the-art optimization algorithms. The experimental results demonstrate that DPSO achieves competitive or superior performance in terms of convergence speed and solution quality across various problem domains. This paper is structured as follows: The Literature Survey is in Section 2. Methodology of the algorithm is introduced in Section 3. The proposed model of the algorithm is introduced in 4th Section. Results are summarized in Section 5 and section 6 describes the Conclusion and Future work.

2. Related Work

Authors proposed a new binary Dragonfly Algorithm (BDA) which uses a variety of techniques to update coefficients' values to solve the feature selection problem [3]. Dragonfly algorithm has been compared to three versions of BDA with different measures. Results indicate that the proposed updated mechanism has a significant impact on algorithm performance. A dragonfly algorithm with an improved opposition-based learning methods and adaptive steps for exponential functions has been developed [4]. By replacing the original random step with the proposed algorithm, the algorithm compares favourably with the original dragonfly algorithm both in terms of convergence speed and convergence accuracy. An article proposes Adaptive Dragonfly algorithm (ADA) is used in a novel load-balancing algorithm [5]. A dragonfly algorithm (DA) and a firefly algorithm are combined in the ADA. Various metrics are used to evaluate the performance of the proposed methodology. As compared to other approaches, the proposed approach achieves better load balancing results. Similarly, the author has presented a more comprehensive find out more about DA's performance in comparison to other swarm intelligence algorithms, as well as its applications across a range of domains [6]. They have also analysed the hybrid of DA and compared the performance and limitations with original DA. The authors of a novel Memory-based Hybrid Dragonfly Algorithm (MHDA) for solving numerical optimization problems have spoken in [7]. To direct the process of conducting a search for potential candidates, conventional DA supplemented with the p-best and g-best concepts of Particle Swarm Optimization (PSO). Testing on the CEC 2014 test functions and the fundamental unrestricted benchmark function validates the MHDA's effectiveness. According to the findings, MHDA performs better than traditional DA and PSO. Ghosh et al. (2023) embarked on a comprehensive study to assess water quality through predictive machine learning. Their research underscored the potential of machine learning models in effectively assessing and classifying water quality. The dataset used for this purpose included parameters like pH, dissolved oxygen, BOD, and TDS. Among the various models they employed, the Random Forest model emerged as the most accurate, achieving a commendable accuracy rate of 78.96%. In contrast, the SVM model lagged behind, registering the lowest accuracy of 68.29%[13]. Alenezi et al. (2021) developed a novel Convolutional Neural Network (CNN) integrated with a block greedy algorithm to enhance underwater image dehazing. The method addresses color channel attenuation and optimizes local and global pixel values. By employing a unique Markov random field, the approach refines image edges. Performance evaluations, using metrics like UCIQE and UIQM, demonstrated the superiority of this method over existing techniques, resulting in sharper, cleaner, and more colourful underwater images [14]. Sharma et al. (2020) presented a comprehensive study on the impact of COVID-19 on global financial indicators, emphasizing its swift and significant disruption. The research highlighted the massive economic downturn, with global markets losing over US $6 trillion in a week in February 2020. Their multivariate analysis provided insights into the influence of containment policies on various financial metrics. The study underscores the profound effects of the pandemic on economic activities and the potential of using advanced algorithms for detection and analysis [15].

3. Methodology

3.1. Dragonfly Algorithm

Mirjalili proposed the Dragonfly Algorithm in 2016. Static and dynamic phases like exploration and exploitation in meta heuristic optimization make up dragonfly optimization. Dragonflies form Sub-swarms and fly over various locations during the static phase. [8] Dragonflies are fly in larger swarms and in a single line during the dynamic phase. Reynolds published a study in 1987 titled "Flocks, Herds and Schools: A Distributed Behavioural Model" in which the following three fundamental stages describe swarms:

1. Separation: refers to reducing static collisions between nearby entities.
2. Alignment: refers to entities that match other local entities speeds in term of speed.
3. Cohesion: refers to a group of entities propensity to gravitate toward the neighborhood’s center.

The separation between two adjacent dragonflies is:

\[ S_i = - \sum_{j=1}^{N} X_i - X_j \]  

(1)

The alignment of dragonflies:

\[ A_i = \frac{\sum_{j=1}^{N} V_j}{N} \]  

(2)

Where, \( V_j \) describe the velocity of \( J^{th} \) neighboring individual.

We can derive the cohesion as:
\[ C_i = \frac{\sum_{j=1}^{N} V_j}{N} - X \]  

(3)

Where \( X \) demonstrates the current person's position, \( P_i \) indicates connecting individual, The number of nearby individuals is \( N \).

We compute the attraction toward the food using this equation:

\[ F_i = X^+ - X \]  

(4)

Where, \( X^+ \) describe the position of the food source.

We can determine the Distraction from the enemy by:

\[ R_i = X^- - X \]  

(5)

Where, \( X^- \) denotes the natural enemy.

Two vectors are taken into account in order to replicate dragonfly motion and update their position in a search space: Step vector (\( \Delta X \)) and position vector (\( X \)).

\[ \Delta X_i + 1 = (sS_i + aA_i + cC_i + fF_i + rR_i) + \omega \Delta X_i \]  

(6)

Where \( s, a, c, f \) are weights for the phases: separation, alignment, cohesion, attraction, and distraction. Also, \( \omega \) presents the inertia weight, and \( \varepsilon \) is the iteration counter.

The position vector is easily determined with the help of the step vector by:

\[ X_i + 1 = X_i + \Delta X_i + 1 \]  

(7)

4. Proposed Model

The hybrid dragonfly-PSO algorithm intends model is an algorithm for meta heuristic optimization combines the advantages of particle swarm optimization (PSO) and dragonfly optimization [9]. The dragonfly algorithms are inspired by nature optimization algorithm that is based regarding behavior of dragonflies in search of prey. It is characterized by the social and swarming behavior of dragonflies to find and catch their prey. The hybrid dragonfly-PSO algorithm proposed model takes advantage of the strengths of both algorithms by incorporating the PSO component into the dragonfly algorithm [10]. This integration allows the algorithm to take advantage of the exploration ability of the dragonfly algorithm and the exploitation ability of the PSO algorithm to find optimal solutions efficiently. In the hybrid dragonfly-PSO algorithm proposed model, the PSO component is added to the dragonfly algorithm by updating the position and velocity of each dragonfly based on the PSO equations [11]. The PSO move is added to the moves towards the best dragonfly, the nearest dragonflies, and the random move to update the position of the dragonfly where the dragonflies are updated based on their distance to the other dragonflies, their attraction to the best dragonfly, their attraction to the nearest dragonflies, and their random movement. Overall, the hybrid dragonfly-PSO algorithm proposed model is a powerful optimization algorithm that can efficiently search for optimal solutions in a wide range of optimization problems [12]. It combines the exploration and exploitation abilities of both the dragonfly and PSO algorithms, which makes it suitable for solving complex problems with multiple local optima.

Proposed DPSO algorithm

1. Initialize the dragonfly’s population \( X_i (i = 1, 2, ..., n) \)
2. Initialize step vectors \( \Delta X_i (i = 1, 2, ..., n) \)
3. Define the maximum number of hyper spheres (segments)
4. Define the archive size
5. while the end condition is not satisfied
   1. Calculate the objective values of all dragonflies
   2. Find the non-dominated solutions
   3. Update the archive with respect to the obtained non-dominated solutions
   4. if the archive is full
   5. Run the archive maintenance mechanism to emit one of the current archive members
   6. Add the new solution to the archive
   7. end if
   8. if any of the new added solutions to the archive is located outside the hyper spheres
   9. Update and re-position all the hyper spheres to cover the new solution(s)
   10. end if
11. Select a food source from archive: \( X^+ = \text{SelectFood(archive)} \)
12. Select an enemy from archive: \( X^- = \text{SelectEnemy(archive)} \)
13. Update step vectors using \( \Delta X_{i+1} = (sS_i + aA_i + cC_i + fF_i + rR_i) + \omega \Delta X_i \)
14. Update position vectors using \( X_{i+1} = X_i + \Delta X_{i+1} \)
15. Check and correct the new positions based on the boundaries of variables
16. end while

5. Simulation and Result Discussion

The simulation was carried out using Cloud sim simulator. The user base was varied between 100 to 10,000 and the response time is accordingly depicted in figure 2. Similarly, to assess the effectiveness of competitive approaches, the best fitness value is also calculated and shown in figure 3. The proposed algorithm measured up PSO and Dragonfly Algorithm and the observation is that DPSO algorithm performs better than other approaches.
6. Conclusion and Future Scope

Due to the search agents' straightforward interactions, which result in a global intelligent behaviour, algorithms for swarm intelligence well-liked metaheuristic algorithms for resolving summon optimization issues. Both static and dynamic fly horoscopes had an impact on a recent swarm intelligence algorithm called the dragonfly algorithm. In several applications, it was discovered to perform better than many other swarm intelligence algorithms. Consider DA’s application to the traveling sales-man problem because it has been demonstrated to outperform multiple swarm intelligence algorithms in a variety of applications, making it a well-known discrete optimization problem with numerous real-world applications. This paper come up with a dragonfly algorithm that is optimized by hybridizing with PSO, where 2 parameters like the number of iterations and population area are being focused on. Different values are applied as these parameters to discover out the greatest suitable condition to dragonfly algo. It is found that the
best result was shown by the hybrid model when population size and number of iterations, are 50 and 100 respectively.

Figure 6. Best fitness value at population size is 20.

Figure 7. Best fitness value at population size is 50.

Figure 8. Best fitness value at population size is 100.

The algorithm has several parameters that can be examined to achieve better performance. Overall, the dragonfly algorithm has shown promising results in solving optimization problems, especially in multi-objective optimization and real-world applications. Since this algorithm is not being explored a lot, global optimization is challenging. Researchers can study and explore more about this algorithm. Also, its performance can be improved by building hybrid models.

References
