Optimization with Atomic Orbital Search (AOS)

Julia Sánchez Esquivel - julia.sanchez.esquivel@alumnos.upm.es

Abstract

This project aims to implement the Atomic Orbital Search (AOS) metaheuristic for solving mathematical optimization problems and enhance its performance using parameter optimization through Artificial Bee Colony (ABC). The project will compare AOS's performance with Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) using metrics such as convergence speed, computational time, and solution accuracy. Simulations will be carried out in Python, with results visualized as short videos to demonstrate algorithm behavior dynamically.

1 Introduction

Optimization problems are central to numerous fields, ranging from engineering to artificial intelligence. Metaheuristic algorithms, inspired by natural and physical processes, have proven effective for solving these problems. Among these, the Atomic Orbital Search (AOS) algorithm is a novel approach inspired by quantum mechanics, which models the movement of electrons around an atomic nucleus to explore the search space. This project explores the potential of AOS by implementing it for well-known benchmark problems and comparing its performance with other popular algorithms.

2 Proposal

The primary goal of this project is to evaluate the effectiveness of AOS in solving optimization problems. To achieve this, AOS will be implemented in Python and tested on benchmark functions, such as Himmelblau's and Rastrigin's functions. Additionally, AOS parameters will be optimized using Artificial Bee Colony (ABC) to enhance its convergence speed and accuracy. Comparative analysis will be performed against PSO and GA to establish AOS's competitiveness.

The simulations will emphasize dynamic visualizations of the algorithms in action, showing their search behavior and convergence.

3 Methodology

The implementation of the AOS algorithm will follow the mathematical model described in the original paper by M. Azizi (2021) [1]. The search space will represent the electron cloud, and candidate solutions will be updated iteratively based on quantum principles of energy absorption and emission. The parameters of AOS, including the photon rate (PR), will be optimized using Artificial Bee Colony (ABC) from `NiaPy` [3] framework. ABC will search for parameter configurations that yield the best performance on selected benchmark problems.

Simulations will be conducted in Python, leveraging libraries such as `numpy` for numerical operations. For the comparative analysis, PSO will be implemented using the `pyswarms` [2] library, while GA will use the `geneticalgorithm` [4] library. Each algorithm will be tested on the benchmark functions over multiple independent runs. The convergence behavior of AOS, PSO, and GA will be visualized dynamically. These visualizations will illustrate how each algorithm explores and converges toward optimal solutions.

4 Expected Outcomes

This project aims to demonstrate the efficiency of AOS as a competitive optimization algorithm. By integrating ABC for parameter optimization, the project expects to show improvements in the algorithm's convergence rate and robustness. Visualizations will highlight the dynamic search process of AOS compared to PSO and GA.

The results will include:

- Short videos demonstrating the iterative optimization process of AOS, PSO, and GA.
- Convergence plots for fitness values over iterations.
- A comparison of computational times and solution for each algorithm.

Ultimately, this project aims to highlight AOS's potential for solving complex optimization problems and its role as an innovative tool in the field of metaheuristics.

References

[1] Mahdi Azizi. Atomic orbital search: A novel metaheuristic algorithm. *Applied Mathematical Modelling*, 93:657–683, 2021.

[2] Lester M. Miranda. A python-based particle swarm optimization library. https://pyswarms.readthedocs.io, 2018.

[3] MIT. Micro framework for building nature-inspired algorithms. https://niapy.org.

[4] Ahmed Rashed. A python library for genetic algorithms. https://pypi.org/project/geneticalgorithm/, 2018.