

Review Article

Tuna Swarm Optimization: A Novel Swarm-Based Metaheuristic Algorithm for Global Optimization

Lei Xie¹, Tong Han,¹ Huan Zhou,¹ Zhuo-Ran Zhang,² Bo Han,³ and Andi Tang¹

¹Aeronautics Engineering College, Air Force Engineering University, Xi'an 710038, China

²Unit 95806 of People's Liberation Army of China, Beijing, China

³Unit 93525 of People's Liberation Army of China, Beijing, China

Correspondence should be addressed to Andi Tang; 418932433@qq.com

DWfHW # (EVfW TVd\$` \$#- 3UWfW \$` EVfW TVd\$` \$#- BgT fZw \$` A UfaTVd\$` \$#

Academic Editor: Ahmed Mostafa Khalil

Copyright © 2021 Lei Xie et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, a novel swarm-based metaheuristic algorithm is proposed, which is called tuna swarm optimization (TSO). The main inspiration for TSO is based on the cooperative foraging behavior of tuna swarm. The work mimics two foraging behaviors of tuna swarm, including spiral foraging and parabolic foraging, for developing an effective metaheuristic algorithm. The performance of TSO is evaluated by comparison with other metaheuristics on a set of benchmark functions and several real engineering problems. Sensitivity, scalability, robustness, and convergence analyses were used and combined with the Wilcoxon rank-sum test and Friedman test. The simulation results show that TSO performs better compared to other comparative algorithms.

1. Introduction

Real-world optimization problems have become more challenging, which requires more efficient solution methods. Different scholars have studied various approaches to solve these complex and difficult problems from the real world. A part of researchers solve these optimization problems using traditional methods such as quasi-Newton, conjugate gradient, and sequential quadratic programming methods. However, owing to the nonlinear, nonproductivity characteristics of most real-world optimization problems and the involvement of multiple decision variables and complex constraints, these traditional algorithms are difficult to be solved effectively [1, 2]. The metaheuristic algorithm has the advantages of not relying on the problem model, not requiring gradient information, having strong search capability and wide applicability, and can achieve a good balance between solution quality and computational cost [3].

Therefore, the metaheuristic algorithms have been proposed to solve real-world optimization problems, such as image segmentation [4, 5], feature selection [6, 7], mission planning [8, 9], parameter optimization [10, 11], job shop scheduling [12, 13], etc.

Metaheuristic algorithms are usually classified into three categories [14]: evolution-based algorithms, physical-based algorithms, and swarm-based algorithms. The evolution-based algorithm is inspired by the laws of evolution in nature. Genetic algorithm (GA) [15], inspired by Darwin's theory of superiority and inferiority, is a well-known evolution-based algorithm. With the popularity of GA, several other widely used evolution-based algorithms have been proposed, including differential evolution (DE) [16], genetic programming (GP) [17], evolutionary strategies (ES) [18], and evolutionary programming (EP) [19]. In addition, several new evolution-based algorithms have been proposed, such as artificial algae algorithm (AAA) [20], biogeography-based optimization (BBO) [21], and monkey king evolutionary (MKE) [22]. The physical-based algorithms are inspired by various laws of physics. One of the most famous algorithms of this category is simulated annealing (SA) [23]. SA is inspired by the law of thermodynamics in which a material is heated up and then cooled slowly. There are other physical-based algorithms proposed, including gravitational search algorithm (GSA) [24], nuclear reaction optimization (NRO) [25], water cycle algorithm (WCA) [26], and sine cosine algorithm (SCA) [27]. The swarm-based algorithms

