Comparative Analysis of the Starfish Optimization Algorithm

Rochester Institute of Technology

Nicholas Boulos | nb4509@rit.edu

Introduction

The starfish optimization algorithm (SFOA) is a novel bio-inspired search technique that models the **cooperative preying behavior of starfish**.

- The algorithm is derived from particle-swarm optimization and operates through two phases: exploration and exploitation.
- Each starfish will randomly choose between **exploring** and **exploiting**.
- **Dimensional awareness** is a key feature of SFOA:
  - For D >= 5, starfish explore **five directions** and follow either **sine or cosine trajectories at random**
  - For D < 5, **a single direction** is used
- During Exploitation, solutions are refined by starfish moving along **a vector defined by two directions pointing towards the global optimum**.
- The regeneration phase removes starfish furthest from the global optimum to **prevent stagnation**

The purpose of this project is to benchmark the technique's performance and answer the following research questions:

- **RQ1: How does SFOA compare to industry-standard algorithms in combinatorial optimization problems and hyperparameter optimization?**
- **RQ2: Are dimensionally-aware techniques the future of search algorithms?**

Methodology

- **Algorithms for comparison**:
  - Particle-Swarm Optimization (PSO): PSO is the progenitor of SFOA so it would provide a good baseline for performance.
  - Differential Evolution (DE): DE is a metaheuristic algorithm that works better for low-dimension problems so it would provide a good comparison for D < 5 SFOA search.
- **Combinatorial Optimization (CO) Problems**:
  - Sphere Function: Smooth, unimodal, function where the objective is to get as close to 0 as possible.
  - Wireless Sensor Network (WSN): Optimizing sensor node coverage represented by circular discs over a square area. Tests multi-modal and spatial applications.
- **Classifiers**:
  - SVM: Support-Vector Machines are a hyperparameter sensitive classifier with a low amount of hyperparameters to optimize.

CO comparisons took place over 30 trials with random seeds 0-29 for reproducibility. While classifier measurements took place over 1 trial using the Iris Flower data set.

Results

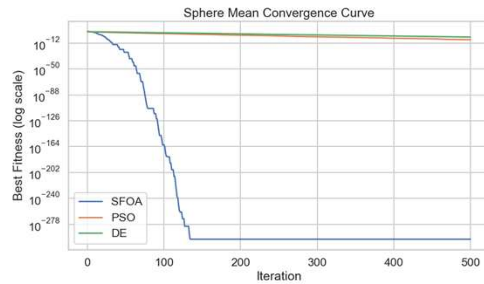
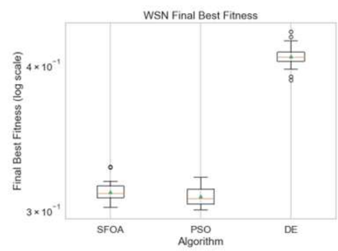


Table A. Sphere Significance

	Raw P-Value	Holm P-Value	Significant
SFOA vs PSO	1.86E-09	1.67E-02	Yes
SFOA vs DE	1.86E-09	2.50E-02	Yes
PSO vs DE	1.86E-09	5.00E-02	Yes

The above figures show the results for the sphere function benchmarks. In this problem, **SFOA greatly outperformed DE and PSO**. SFOA was the only technique to consistently converge on the theoretical minimum and did so **over 300 iterations faster** than PSO and DE. This is reinforced by the difference in solutions **being statistically significant.**



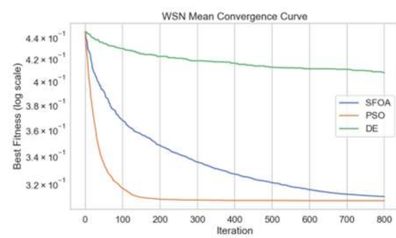


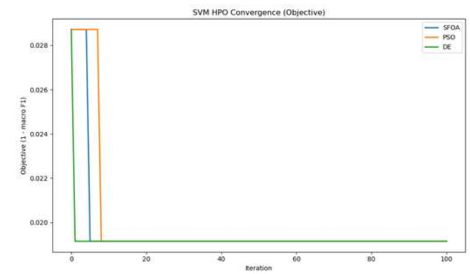
Table B. WSN Success Rate (Within 0.02 of best run)

	Success	Failure
SFOA	28	2
PSO	30	0
DE	0	30

Table C. WSN Significance

	Raw P-Value	Holm P-Value	Significant
SFOA vs DE	1.73E-06	1.67E-02	Yes
PSO vs DE	1.73E-06	2.5E-02	Yes
SFOA vs PSO	1.18E-01	5.0E-02	No

The above figures show the benchmarking for the WSN problem. In this problem, **performance between SFOA and PSO was closer**. While the solutions that SFOA and PSO found **were not significantly different**, the speed **PSO converged on that solution was approximately 100 iterations faster than SFOA**, and all 30 PSO trials were within 0.02 of the best solution while SFOA failed twice to stay within that interval. This may be because the exploration phase of PSO covers more ground quicker than SFOA which has large impacts for spatial problems.



All three algorithms achieved a 1.00 accuracy from a baseline of 0.95 accuracy. However, DE was much faster at converging on the best parameters to achieve this while SFOA only outperformed PSO. This could be due to the **low-dimensionality of SVM optimization compared to the CO problems.**

Discussion

Overall, **SFOA could perform equally if not better** than the algorithms used to compare in this project. Smooth mathematical functions appear to be its strongest application. PSO has a faster exploration technique for spatial problems and DE performs better on low-dimensional problems. However, what makes SFOA unique is its dimensional awareness. While it does not outperform either algorithm in what those techniques were designed for, it can perform comparably while PSO or DE struggle on problems they are not designed for. While dimensionally-aware algorithms may not be as performant on problems they are not explicitly designed for, their use could still be justified due to the simplicity of general purpose search algorithms. Due to time constraints, this project was unable to test on a large suite of CO problems, algorithms, and classifiers with varying amounts of hyperparameters. Further research would involve testing against larger test suite and testing the parameter sensitivity of SFOA.

References

- Ahmad, M. F., Isa, N. A. M., Lim, W. H., & Ang, K. M. (2022). Differential evolution: A recent review based on state-of-the-art works. *Alexandria Engineering Journal*, 61(5), 3831–3872. https://doi.org/10.1016/j.aej.2022.07.036
- Alkhalifa, A. K., Aljebreen, M., Alanazi, R., Ahmad, N., Alrusaini, O., Aljehane, N. O., Alqazzaz, A., & Alkhiri, H. (2025). Leveraging hybrid deep learning with starfish optimization algorithm-based secure mechanism for intelligent edge computing in smart cities environment. *Scientific Reports*, 15, 33069. https://doi.org/10.1038/s41598-025-11608-4
- Sakpere, W., Yisa, F. I., Salami, A., & Olaniyi, G. A. (2025). Particle swarm optimization and benchmark functions: An extensive analysis. *International Journal of Engineering Research in Computer Science and Engineering, 12*(1), 1–12.
- UCI Machine Learning Repository. (n.d.). *Iris species* [Data set]. Kaggle. https://www.kaggle.com/datasets/uciml/iris
- Zhong, C., Li, G., Meng, Z., Li, H., Yildiz, A. R., & Mirjalili, S. (2025). Starfish optimization algorithm (SFOA): A bio-inspired metaheuristic algorithm for global optimization compared with 100 optimizers. *Neural Computing and Applications*, 37, 3641–3683. https://doi.org/10.1007/s00521-024-10694-