2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

An Ensemble Method for Applying Particle Swarm Optimization Algorithms to Systems Engineering Problems

Ken Hampshire
School of Engineering and Applied Science
George Washington University
Washington, DC
ken.hampshire@gwu.edu

Thomas Mazzuchi, D.Sc.
School of Engineering and Applied Science
George Washington University
Washington, DC
mazzu@gwu.edu

Shahram Sarkani, PhD
School of Engineering and Applied Science
George Washington University
Washington, DC
sarkani@gwu.edu

***Abstract*—As a subset of metaheuristics, nature-inspired optimization algorithms such as particle swarm optimization (PSO) have shown promise both in solving intractable problems, and in their extensibility to novel problem formulations due to their general approach requiring few assumptions. Unfortunately, a given algorithm requires detailed tuning of parameters and cannot be proven to be best suited to a particular problem class on account of the "no free lunch" (NFL) theorems. Using these algorithms in real-world problems requires exquisite knowledge of the many approaches and applying them based upon intuition. This research aims to present a unified view of PSO-based approaches from the perspective of relevant systems engineering problems, with the purpose to then elicit the best solution for any problem formulation in an ensemble learning approach. The central hypothesis of the research is that using the PSO algorithms found in literature to solve real-world optimization problems requires a general ensemble-based method for all problem formulations but a single implementation and solution for any instance. The main results will be a problem-based literature survey and a general method to find more globally optimal solutions for any systems engineering optimization problem.**

Keywords—particle swarm optimization, ensemble learning, swarm intelligence

I. Background

Particle swarm optimization (PSO) is one of the most promising global search algorithms used to solve optimization problems because the technique is simple to code and implement, has few parameters, and its flexibility allows for easy modification or hybridization with other metaheuristic approaches [1]. The technique was presented by Kennedy & Eberhart in 1995, and first initializes randomly generated particle (candidate solution) positions and velocities in the search space [2]. Then, these positions are evaluated relative to one another to determine the particle set best solution, and subsequent iterations move particles towards this best solution by updating each particle's position and velocity. For a given particle, the tradeoff between searching around itself and moving towards the set best is controlled by social and cognitive damping coefficients. This tradeoff is often referred to in literature as the balance between exploration and exploitation.

Despite its simplicity, PSO suffers from two drawbacks: premature convergence to local minima, and the inability to guarantee finding a global minimum. The latter problem is common to all metaheuristic approaches. Usually, searching stops after prescribed iteration or time limits. Much of the recent literature has been dedicated to improving the algorithm through three main approaches: identifying appropriate parameter settings, hybridizing PSO with other approaches, or using multiple particle sets (swarms) [1]. The last approach often turns to biology for inspiration in new ways to balance exploration and exploitation [3].

Novel PSO approaches in a paper usually highlight the inspiration, the algorithm approach, test the approach on benchmark optimization function sets such as CEC2005, compare performance to other PSO approaches, then illustrate the approach on one or more real-world problem formulations [4]. Yet Wolpert and Macready's "no free lunch theorems for optimization" (NFL) state that a given algorithm's performance on one class of problems is offset by its performance on another class [5]. This suggests that any two algorithms become computationally equivalent when averaged over all classes in a domain, creating a need for PSO approaches matched to problem classes.

The NFL theorems notwithstanding, a gap in the research literature is taking a problem-first approach to the study of PSO, specifically in the class of systems engineering problem formulations. Novel PSO papers often highlight their approach's relevance to real-world problems, which some review papers have aggregated when surveying and reconciling the various approaches [1], [6], [7]. Yet it remains difficult to apply the many PSO approaches to a particular systems engineering problem because doing so requires a unique understanding of those approaches and the ability to tune parameters for the given problem, while the NFL theorems suggest that relative performance is somewhat arbitrary. In short, most of the PSO literature is from the computational science perspective and assumes detailed optimization knowledge. While this is necessary, it is not sufficient to scale its adoption for use in systems engineering on problems such as scheduling, allocation, network ontologies, or system design.

We hypothesize that systems engineering would benefit from systematically cataloging which PSO approaches have been used on what class of problems and providing a general method for their use when solving optimization problems in systems engineering. The general method is essential to account for the NFL theorems of optimization and defines a repeatable way to leverage PSO techniques to improve optimization in systems engineering.

II. Method

A. Literature Survey

Our literature survey will examine papers from journals which publish the preponderance of PSO research on systems engineering, including: *IEEE Transactions on Systems, Man, and Cybernetics; Expert Systems with Applications; IEEE Journal of Automatica Sinica; Complex & Intelligent Systems.* The goal of reviewing each paper is to understand and classify the PSO approach, then extract and categorize the sample problem formulations where that approach was used, especially as applied to systems engineering problems. There are two outputs of the literature survey: a catalog of PSO approaches and the types of problems they have been applied to as in Fig. 1, as well as a review paper which presents common systems engineering problem formulations and describes the PSO approaches applied to them from a problem-centric perspective.

B. Ensemble Optimization

The general method draws upon the PSO algorithm catalog produced by the literature survey to select from the most promising algorithms given knowledge of the problem at hand. Within a given class of problem, the method involves iteratively solving using each algorithm and returning the best solution from the set of candidate algorithms, similar to the ensemble learning approach found in machine learning.

Because PSO is a type of stochastic optimization, the best solution usually varies from run to run. Therefore, the method will be repeated for each problem multiple times (m runs) to ensure statistically significant results. The Friedman ranking test will be used to compare the relative performance of the various algorithms as in Table I.

Finding solutions through repeated iterations of multiple approaches is computationally expensive and not practical as the basis of a general method. Therefore, we hypothesize that there are two ways to construct the general method. First, if the relative performance of the algorithms is statistically insignificant, the bucket of algorithms will be used only as a diversity of solutions for a naïve problem formulation. Second, if the relative performance of the algorithms is statistically significant, the general ensemble method will be weighted to bias proportionally towards the best algorithms for a one-time solution to a naïve problem formulation.

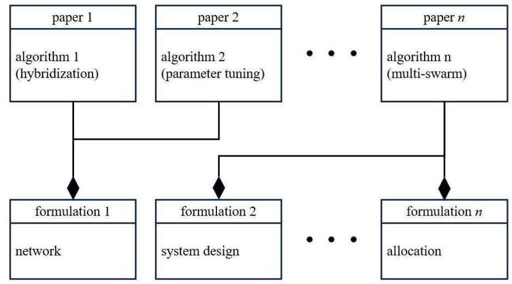

Fig. 1. Sample literature survey categorgization schema.

TABLE I. Data Organization Schema.

run		algorithm 1	algorithm 2	…	algorithm n
1	***mean***				
	std				
2	***mean***				
	std				
…					
m	***mean***				
	std				
rank					

References

[1] T. Shami et al., "Particle swarm optimization: a comprehensive survey," IEEE Access, vol. 10, pp. 10031-10061, 2022.

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN '95 – International Conference on Neural Networks, vol. 4, pp. pp. 1942-1948, November 1995.

[3] H. Minh, T. Sang-To, G. Theraulaz, M. Abdel Wahab, and T. Cuong-Le, "Termite life cycle optimizer," Exerpt Sys. Appl., vol. 213, p. 119211, March 2023.

[4] P. Suganthan et al., "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," KanGAL report, vol. 2005005, no. 2005, p. 2005, 2005.

[5] D. Wolpert and W. Macready, "No free lunch theorems for optimization," IEEE Trans. Evol. Comput., vol. 1, no. 1, pp. 67–82, April 1997.

[6] J. Nayak, H. Swapnarekha, B. Naik, G. Dhiman, and S. Vimal, "25 Years of Particle Swarm Optimization: Flourishing Voyage of Two Decades," Arch. Comput. Methods Eng., vol. 30, no. 3, pp. 1663–1725, April 2023.

[7] J. Tang, G. Liu, and Q. Pan, "A Review on Representative Swarm Intelligence Algorithms for Solving Optimization Problems: Applications and Trends," IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 10, pp. 1627–1643, October 2021