A Survey of Reinforcement Learning Approaches for Tuning Particle Swarm Optimization

Bogdan Milićević[1,2,3*], Vladimir Milovanović[2]

[1] Institute for Information Technologies, University of Kragujevac, Jovana Cvijića bb, 34000 Kragujevac, Serbia; e-mail: bogdan.milicevic@uni.kg.ac.rs
[2] Faculty of Engineering, University of Kragujevac, Sestre Janjić 6, 34000 Kragujevac, Serbia; e-mail: vlada@kg.ac.rs
[3] Bioengineering Research and Development Center (BioIRC), Prvoslava Stojanovića 6, 34000 Kragujevac, Serbia;

* *Corresponding author*

DOI: 10.46793/ICCBIKG25.198M

Abstract: Particle Swarm Optimization (PSO) remains a popular, simple, and strong baseline for numerical optimization, yet its performance depends critically on a small set of hyper-parameters (e.g., inertia weight w and cognitive and social coefficients c_1, c_2) and on structural design choices (e.g., topology, velocity clamps). Over the last decade, reinforcement learning (RL) has emerged as a principled, data-driven way to adapt these design choices online—either by directly controlling parameters, reshaping swarm interactions, selecting variation operators, or transferring control policies across runs. This survey systematizes RL–for–PSO tuning along four families: (1) direct parameter control, (2) topology/structure control, (3) operator/strategy selection, and (4) cross-run memory and transfer. We highlight representative methods—including tabular Q-learning, Deep Q-Networks (DQN), deterministic policy gradients (DDPG), and hybrid RL–PSO schemes—summarize empirical evidence, and distill practical design patterns (state, action, reward, and training protocols). We conclude with open challenges in stability, sample efficiency, safety-constrained control, and reproducible benchmarking.

Keywords: particle swarm optimization, reinforcement learning, parameter tuning

1. Introduction

PSO models a population of particles whose positions are updated by linearly combining momentum, cognitive, and social coefficients, originally introduced by Kennedy and Eberhart [1]. The introduction of an inertia weight in 1998 provided an explicit lever to trade exploration and exploitation, catalyzing a large body of parameter-control work [2]. Time-varying and adaptive inertia-weight designs (e.g., linear decrease, fuzzy/adaptive rules, and per-particle adjustments) became standard tools. Classic representatives include AIW-PSO (per-dimension adaptive weights using an

Individual Search Ability measure) and later adaptive variants that maintain diversity and stabilize convergence [3,4]. While these designs encode expert heuristics, reinforcement learning now offers an alternative: learn how to select parameters and strategies from feedback during the run or even before the run via pretraining.

2. Methodology

RL formalizes parameter/strategy choice as sequential decision-making: at each iteration the controller observes a state summarizing the swarm's search progress, selects an action (e.g., updated inertia weight, cognitive and social coefficients or a topology/operator), and receives a reward that typically encodes improvement in the best objective value. Early studies integrated Q-learning into PSO, and subsequent work adopted deep RL (DQN, DDPG) for richer, continuous control [5–7]. One of the particularly influential approaches is RLAM-RLPSO: a DDPG-based controller pre-trained to output PSO coefficients from features such as iteration progress, non-improving-iteration ratio, and swarm diversity; the learned policy is then deployed online [8].

3. Results and Discussion

Reinforcement learning methods applied to particle swarm optimization can be broadly divided into four categories: direct parameter control, topology and structure control, operator or strategy selection, and cross-run transfer of learned behaviors. Each category emphasizes a different mechanism by which RL augments or replaces traditional heuristic rules.

Direct Parameter Control. The most common application of RL within PSO is the adjustment of key parameters such as the inertia weight and the cognitive and social coefficients. Early approaches integrated tabular Q-learning, in which states were defined by simple progress indicators such as iteration count or swarm diversity, and actions corresponded to discrete parameter settings. This led to QLPSO-style variants that outperformed static schedules across continuous optimization problems [5]. More recent studies extended the idea with function-approximation, for example through deep Q-networks, enabling self-tuning PSO that reacts to state signals modeled as a partially observable Markov decision process [7]. Moving beyond discretization, continuous-action RL methods such as DDPG allow direct adjustment of parameters at every iteration. The RLAM–RLPSO framework exemplifies this line, where an actor–critic network outputs inertia and acceleration coefficients based on progress, diversity, and stagnation indicators, producing statistically significant improvements over fuzzy logic and Q-learning baselines on the CEC benchmark suite [8].

Topology and Structure Control. A second family of approaches focuses on adapting the communication topology of the swarm. Since the rate and structure of information exchange among particles directly influence the balance of exploration and exploitation, learning which neighbors to connect to becomes a natural RL problem. One representative work uses Q-learning to adjust the swarm topology online, showing that learned communication patterns improve scalability and robustness compared with static ring or star topologies [9].

Operator and Strategy Selection. Another direction is the use of RL to decide which velocity update rule or mutation operator to apply at a given iteration. Instead of committing to a single operator throughout the run, the swarm alternates strategies under the guidance of a learned policy. For instance, NRLPSO introduces a neighborhood differential mutation operator alongside classical velocity updates, while reinforcement learning is employed to select between them adaptively, yielding stronger performance across CEC benchmarks [10]. This mechanism positions RL as an adaptive strategy selector, comparable to hyper-heuristics in evolutionary computation.

Cross-Run Memory and Transfer. Finally, reinforcement learning has been leveraged to transfer useful control behaviors across optimization runs, particularly in cyclic or repetitive domains. In the context of wastewater treatment optimization, RLPSO maintains an elite network of successful particle actions, which is then reused to guide velocity predictions in subsequent cycles. This approach significantly reduces computational cost compared with multi-objective baselines while maintaining effluent quality and energy efficiency [11]. Such cross-run reuse highlights RL's potential not only for within-run control but also for meta-level learning across repeated problem instances.

4. Conclusion

Reinforcement learning has proven to be a powerful framework for enhancing particle swarm optimization, offering systematic ways to adjust parameters, restructure swarm topologies, select among competing operators, and even transfer experience across repeated optimization tasks. Across these approaches, common design patterns emerge: states are usually derived from progress and diversity indicators, actions correspond to parameter values or operator choices, and rewards measure improvements in objective value or feasibility. Continuous-action methods such as DDPG offer finer-grained control compared to discrete Q-learning, while pretraining strategies improve stability and sample efficiency relative to purely online learning.

Empirical studies consistently show that RL-tuned PSO variants outperform static and heuristic parameter adaptation, with notable successes in both benchmark functions and real-world applications such as cyclic wastewater treatment. At the same time, open challenges remain in balancing sample efficiency with robustness, designing rewards that handle constraints gracefully, and ensuring reproducibility of RL–PSO hybrids. Looking forward, the integration of transfer learning and meta-RL holds promise for making PSO adaptation more generalizable across problem domains.

Acknowledgment

This research was supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia, contract number [451-03-136/2025-03/200378 (Institute of Information Technologies, University of Kragujevac) and 451-03-136/2025-03/200107 (Faculty of Engineering, University of Kragujevac)].

References

[1] J. Kennedy, R. Eberhart, Particle Swarm Optimization, Proceedings of the IEEE International Conference on Neural Networks (ICNN'95), 4 (1995) 1942–1948.

[2] Y. Shi, R.C. Eberhart, Parameter Selection in Particle Swarm Optimization, Proceedings of the 1998 IEEE International Conference on Evolutionary Computation (ICEC'98), (1998).

[3] Z. Qin, L. Huang, X. Ding, Adaptive Inertia Weight Particle Swarm Optimization, in: ICAISC 2006, LNAI 4029, (2006) 450–459.

[4] S. Kessentini, D. Barchiesi, Particle Swarm Optimization with Adaptive Inertia Weight, International Journal of Machine Learning and Computing, 5 (2015) 368–373.

[5] Y. Liu, H. Lu, S. Cheng, Y. Shi, An Adaptive Online Parameter Control Algorithm for Particle Swarm Optimization Based on Reinforcement Learning, Proc. IEEE Congress on Evolutionary Computation (CEC), (2019) 815–822.

[6] R. Olivares, F. Jaramillo, A Learning-Based Particle Swarm Optimizer for Solving Continuous Optimization Problems, Algorithms, 12(7) (2023) 643.

[7] O. Aoun, Deep Q-Network-Enhanced Self-Tuning Control of Particle Swarm Optimization, Modelling, 5(4) (2024) 1709–1728.

[8] S. Yin, M. Jin, H. Lu, G. Gong, W. Mao, G. Chen, W. Li, Reinforcement-Learning-Based Parameter Adaptation Method for Particle Swarm Optimization, Complex & Intelligent Systems, 9 (2023) 5585–5609.

[9] Y. Xu, D. Pi, A Reinforcement-Learning-Based Communication Topology in Particle Swarm Optimization, Neural Computing and Applications, 32 (2020) 10007–10032.

[10] W. Li, P. Liang, B. Sun, Y. Sun, Y. Huang, Reinforcement Learning-Based Particle Swarm Optimization with Neighborhood Differential Mutation Strategy (NRLPSO), Swarm and Evolutionary Computation, (2023) 101274.

[11] L. Lu, H. Zheng, J. Jie, M. Zhang, R. Dai, Reinforcement Learning-Based Particle Swarm Optimization for Sewage Treatment Control, Complex & Intelligent Systems, 7 (2021) 2199–2210.