Wiebke Frenkel and Christian Renner. 2025. Hyperparameter Optimizer for PSO-Based Energy-Aware Path Planning for AUV Swarms. 2025 International Conference on Architecture of Computing Systems (ARCS), Kiel, Germany.

Hyperparameter Optimization for PSO-Based Energy-Aware Path Planning for AUV Swarms

Wiebke Frenkel[1,2][0000−0001−8030−6963] and
Bernd-Christian Renner[1,3][0000−0002−6936−6444]

[1] Hamburg University of Technology
Institute for Autonomous Cyber-Physical Systems,
Harburger Schloßstr. 28, 21079 Hamburg, Germany
https://www.tuhh.de/acps/
[2] wiebke.frenkel@tuhh.de
[3] christian.renner@tuhh.de

Abstract. In missions using autonomous underwater vehicles (AUVs), reaching predefined waypoints is essential, e.g., for seabed mapping and infrastructure monitoring. The primary objective is to minimize energy consumption to reduce the risk of failure, avoid long (re)charging times, or maximize the number of successfully visited waypoints. If the AUVs can visit the waypoints in any order, we can frame this as a Travelling Salesperson Problem (TSP). However, finding the global optimal solution to the TSP becomes computationally infeasible even for relatively small sets of waypoints. To address this challenge, we utilize Particle Swarm Optimization (PSO) as a lightweight alternative, allowing us to obtain energy-efficient routes that closely approximate the global solution. Choosing the corresponding hyperparameters is fundamental, as poor selections can lead to local optima. Our analysis shows significant differences in the sensitivity of hyperparameters between the two PSO-based approaches, which only differ in initialization. However, there is a consistent range of hyperparameters where both methods yield comparable results. We identify this range by optimizing hyperparameters to improve solution quality. Simulative evaluations in real-world-inspired scenarios demonstrate that optimized hyperparameter selection improves the energy efficiency of the AUV swarm and ensures reliable mission execution using a minimal number of AUVs.

Keywords: Particle Swarm Optimizer · Hyperparameter Optimization · Travelling Salesperson Problem · AUV Swarm

1 Introduction

Autonomous underwater vehicles (AUVs) are a common choice for a plethora of applications and use cases, e.g., ocean exploration or environmental monitoring. In particular, they have become an essential tool in scientific research, where institutions employ AUVs to gather critical data, as noted in [7]. AUV designs vary based on specific applications, e.g., [20] mentions the deployment of different AUVs for various missions. In [1], AUVs are used for polar and marine research, operating in challenging environments such as the Arctic Ocean.

They can estimate underground oil plumes and concentrations, facilitating long-range pollution monitoring, as described in [5]. Similarly, AUVs are utilized for inspecting offshore infrastructure [14] and dam inspections [18]. Moreover, they can assist in predicting environmental disasters using underwater measurements, as highlighted in [18].

In the examples discussed, AUVs collect data simultaneously or over time and navigate to various waypoints. As the number of waypoints increases, deploying a swarm of AUVs to work together can be an effective strategy to enhance operation time and robustness. Mission planning involves determining which AUV will handle each task, including the waypoints and their order, to maximize mission duration and ensure success. We previously addressed this challenge in [9] by framing it as a Traveling Salesperson Problem (TSP) and optimizing the route using a Particle Swarm Optimizer (PSO).

The PSO is inspired by the natural dynamics of flocks of birds or schools of fish and uses self-organized swarm behavior to find an energy-optimal solution. Due to its stochastic nature, PSO offers advantages over other optimization techniques, such as ease of implementation and efficient exploration of large solution spaces. It can deliver near-optimal solutions quickly, making it ideal for real-time applications and resource-limited situations. While PSO provides a practical and flexible optimization routine, its results depend on the choice of configuration parameters, such as population size or number of generations, as we have already discussed in [9]. These parameters influence convergence speed and solution quality, making their sensible choice crucial. Traditional empirical tuning is often time-consuming and suboptimal, especially when dealing with complex, real-time applications. To address this challenge, we adopt a systematic approach by treating these configuration variables as hyperparameters and applying hyperparameter optimization (HPO) techniques widely used in machine learning to enhance algorithmic performance. By leveraging HPO, we aim to improve the robustness and efficiency of PSO-based mission planning for AUVs. In particular, we have identified the following research questions that we address in the paper:

- **How can hyperparameter optimization enhance the PSO-based AUV mission planning?**
  We systematically apply HPO to optimize PSO parameters for swarm-based AUV mission planning. Unlike empirical tuning, where parameters are selected based on manual adjustments or heuristics, our method accelerates the process while improving robustness by considering energy constraints and mission success rates.
- **How does hyperparameter sensitivity affect the stability and energy efficiency of PSO-based AUV mission planning?**
  We conduct an extensive statistical analysis to determine the impact of hyperparameter configurations on PSO performance. This allows us to establish well-defined boundary conditions for stable and energy-efficient mission execution.

- **How can energy-aware optimization reduce AUV swarm size while ensuring mission feasibility?**
  Our framework optimizes mission plans to reduce the number of AUVs required while ensuring energy-efficient waypoint allocation. This leads to lower operational costs and improved mission feasibility in resource-limited environments.
- **How does the optimized PSO-based planning perform under simulated real-world constraints?**
  We evaluate our optimized PSO-based planning approach through simulations that mimic real-world constraints, demonstrating its practical applicability in marine operations such as environmental monitoring and underwater infrastructure inspection.

We review PSO-based AUV path planning in section 2, highlighting strengths and the need for systematic HPO. To address this, we integrate energy-aware AUV allocation into HPO. We formulate the mission planning problem as an asymmetric TSP in section 3, optimizing waypoints and routes under energy constraints. Our cascaded framework in section 4 uses HPO to tune PSO hyperparameters for better convergence and energy efficiency. We evaluate the approach in section 5, analyzing convergence, energy use, and robustness across missions. Finally, section 6 summarizes findings and future work.

2 State of the Art

PSO is a widely used tool in ongoing research on optimization algorithms, which is also described in [11]. The authors use the PSO to solve the discrete optimization problem of the TSP, focusing on a balanced behavior in exploration and exploitation. Classical PSO is effective but often gets stuck in local optima. To improve this, [13] created an enhanced PSO for three-dimensional path planning for AUVs. This version features adaptive nonlinear weight adjustments and a modified velocity function that uses the average of particle positions to improve movement and reduce dispersion. Additionally, it incorporates Darwin's "survival of the fittest" concept to boost convergence speed and optimization performance. The updated parameters were empirically determined and validated through simulations. The article [16] presents an AUV path-planning strategy using a combination of constrained sampling A* and an adaptive quantum-behaved PSO to minimize energy consumption during single AUV diving processes in the presence of ocean currents. The algorithm's effectiveness is demonstrated through validation and comparison with other algorithms using six test scenarios in a simulation. However, some parameters, such as population size and generations, are determined empirically.

The improvements make the original PSO more complex and thus contradict the idea of [10], who introduced the algorithm in 1995, of a simple mathematical description that can be implemented with just a few lines of code.

In all of the mentioned applications, the PSO's hyperparameters, such as population size or generation, were determined empirically. Setting these hyper-

parameters is anything but trivial and often involves a large number of time-consuming simulation runs, especially with a large number of interrelated hyperparameters. A better way than an empirical setting is to use an optimization technique to find the best set of these parameters. In machine learning, the optimization of hyperparameters has already been established in practice. As [21] pointed out, the direct impact on the model's performance makes appropriate optimization methods to determine the best model parameters essential to avoid time-consuming trial and error. We can also define hyperparameters for the PSO, which we can then determine using a suitable HPO. In addition to standard PSO parameters such as population size or generations, we can also set the number of AUVs as a hyperparameter. Compared to [9], we map the number of AUVs to the waypoints instead of setting them empirically and hoping that the energetically minimum route can be navigated with the selected number.

3 Problem Statement

A fully autonomous swarm of AUVs can enhance underwater exploration, environmental monitoring, and infrastructure inspection. We propose a multi-layered system architecture consisting of a base station, the AUV swarm, and the environment, as illustrated in fig. 1. Using environmental data like nautical charts, the base station is responsible for the global, coarse-grained mission planning, which includes preliminary path planning and assignment of waypoints. The AUV swarm takes charge of executing the mission autonomously. The swarm members optimize their local paths and adjust to real-time environmental conditions, such as dynamic obstacles and unknown changes in terrain, using their onboard sensors. Real-time decision-making is essential for modifying mission parameters, like altering routes or reassigning waypoints, based on immediate sensor data and predefined mission goals. Additionally, ad-hoc communication between AUVs is crucial for coordinating tasks, sharing important information, e.g., obstacles and changes in conditions, and ensuring efficient swarm behavior while avoiding collisions and redundancy.

This paper addresses the resource-efficient operation of near-optimal path planning at the base station. Therefore, we pre-plan routes at the base station and communicate them to the AUVs, allowing for local adjustments based on real-time factors like energy consumption and environmental changes. We break down mission planning and execution into two main tasks to achieve high-quality solutions with computational efficiency:

1. Assign waypoints to AUVs: Distribute N waypoints among M AUVs, ensuring each waypoint is visited once while balancing the energy consumption.
2. Determine energy-optimal routes: Solve the TSP for each AUV to find the most energy-efficient route for its assigned N_i waypoints, considering environmental factors.

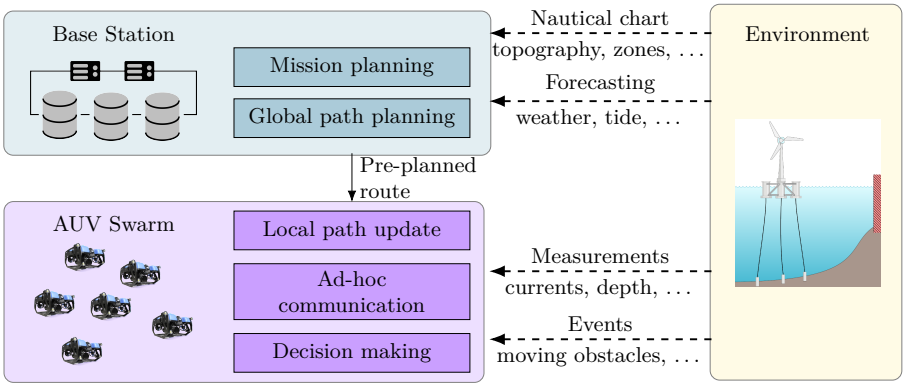


Fig. 1. Overview of the AUV swarm mission setup.

3.1 Energy Model

AUVs are limited by their energy supply, typically from Li-ion batteries. While advancements in battery technology have improved operational duration, energy remains a finite resource that impacts mission feasibility. We use the energy consumption model from [9], which incorporates location-dependent factors like ocean currents, hydrodynamic drag, and vehicle dynamics, demonstrating that energy consumption is influenced by more than just distance, as described in [8].

In fig. 2, we present the AUV's energy consumption model, consisting of mechanical and electrical components. As noted in [9], AUVs can improve energy efficiency by utilizing natural ocean currents. We define a cost function, where Θ^* represents the set of optimal parameters derived through the optimization process, given by

$$f(\Theta^*) = \sum_{i=0}^{N-1} E_i \quad \textbf{with } E_i = \mathbf{F} \cdot |\mathbf{x}| \tag{1}$$

that we use to minimize energy consumption by considering the interaction between the AUV and its environment. The differential displacement $\mathbf{x} = \begin{bmatrix} \mathrm{d}x & \mathrm{d}y & \mathrm{d}z \end{bmatrix}^{\mathrm{T}}$ and the force $\mathbf{F}$, which includes environmental factors like ocean currents and hydrodynamic resistance, are both included in the cost function. The energy consumption is adjusted based on the AUV's relative velocity $\mathbf{v}_r = \mathbf{v} - \mathbf{v}_c$, where $\mathbf{v}$ is the velocity of the AUV from a global perspective, and $\mathbf{v}_c$ is the velocity of the ocean current. The energy use is reduced when the AUV moves with the current and increased when moving against it.

A Li-ion battery provides the electrical power to drive the propulsion unit. Similar to [9], we make a statement about mission fulfillment with the estimation of the time derivative of the state of charge SoC

$$\dot{\sigma}(t) = -\frac{1}{C_{\mathrm{bat}}} \cdot i(t) \tag{2}$$

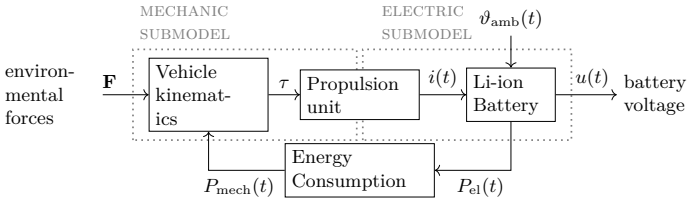


Fig. 2. Energy consumption model of an AUV.

with the capacity C_{bat} and the electric current $i(t)$ of the battery using the nonlinear differential equation system as proposed in [17]. The authors use an electrical equivalent circuit model to describe the electrochemical behavior of the battery, which influences the system dynamics.

3.2 Complexity

As noted in [6], we tackle the problem by converting it into an asymmetric TSP, where energy consumption, influenced by direction-dependent external forces, replaces distance. For an asymmetric TSP with more than two cities, the maximum number of valid tours is $(N-1)!$, leading to an exponentially growing search space that makes complete searches impractical, even for a few cities.

The brute force approach [15], which determines all possible permutations of routes, calculates their associated costs, and selects the lowest-cost route, is an exact method. The number of possible solutions is given by $\sum_{i=0}^{M-1}(N_i-1)!$ for a uniform distribution of locations. While theoretically sound, this method becomes computationally expensive for larger problem instances.

In practice, metaheuristics such as PSO, genetic algorithms, or simulated annealing are often approximated to find near-optimal solutions more efficiently. In the case of PSO, the search space is limited to $M \cdot P \cdot G \cdot \frac{N}{M}$, determined by the population size P and the number of generations produced G. In practice, these methods are efficient because they can achieve valuable results faster than exact algorithms, such as the brute force approach, by iteratively improving a population of solutions.

4 Optimization Process

We structure the entire optimization process in fig. 3 into a cascaded framework, where HPO serves as the outer optimization, and the PSO-based TSP solver functions as the inner optimization. As a result, the HPO provides the hyperparameters, which serve as the configuration for the PSO. We embedded the energy model of the AUV in an underlying simulation, which provides values for the object function of the HPO. We solve the AUV swarm's underlying TSP in each iteration step and simulate the energy consumption and battery dynamics.

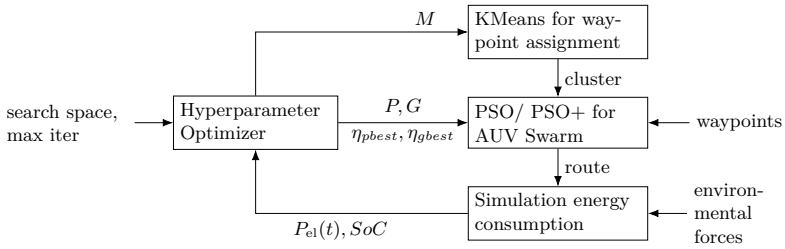


Fig. 3. Block diagram of the cascaded optimization process.

4.1 PSO and PSO+ for TSP

The PSO, introduced by [10], combines elements of genetic algorithms and evolutionary computation. Inspired by flocking behavior in nature, PSO simulates decentralized systems where simple individual actions lead to collective intelligence. Each particle represents a potential solution, moving in an n-dimensional search space and adjusting its trajectory based on personal experiences and the swarm's collective knowledge. The optimal solution is determined by modifying the distances between particles through a stochastic process. The new position of a particle

$$X_i^{k+1} = V_i^k + X_i^k \tag{3}$$

is determined from the current position X_i^k and the corresponding velocity calculating with the discrete differential equation

$$V_i^{k+1} = \omega \cdot V_i^k + \eta_1 \cdot \mathrm{R} \cdot (P_{\mathrm{best}}^k - X_i^k) + \eta_2 \cdot \mathrm{R} \cdot (P_{\mathrm{global}}^k - X_i^k) \ , \tag{4}$$

whereby ω is the inertia factor and $\eta_1 \cdot R$ weights the best position of the particle P_{best} and $\eta_2 \cdot R$ weights the global best position P_{global}. We define the optimization problem

$$P_{\mathrm{global}}^k = \arg\min_{X_i^k} f(\Theta^*), \tag{5}$$

by minimizing the cost function eq. (1) with the position of the particles X_i^k in the search space.

In [19], the PSO was adapted for discrete problems like the TSP. We initialize each AUV_i with $i \in \{0, \ldots, M-1\}$, assigning a particle swarm P_i to solve the TSP, with each particle representing a randomized route. Instead of distances between waypoints, we use energy levels based on the energy consumption required for an AUV to travel between points. This energy consumption depends on the distance traveled, the AUV's speed, and the water current, see section 3.1, while we assume a constant speed for our problem. As particles exchange information, the best global solution with the lowest energy level is identified.

To enhance PSO for the TSP, the authors in [12] introduce a greedy initialization for each particle, called PSO+. Instead of random initialization, PSO+ selects the next point with the lowest energy level from a starting point. This method improves convergence speed but may also lead to getting stuck in a local optimum.

4.2 Hyperparameter Optimizer

Selecting hyperparameters for PSO-based algorithms is challenging as their values significantly affect convergence speed and solution quality. The complex interactions between hyperparameters, such as population size and inertia weight, make manual tuning time-consuming and error-prone. To find an optimal configuration, we use HPO. Open-source libraries like Hyperopt for Python [4] facilitate HPO, employing methods such as Random Search (RS) and the Tree-structured Parzen Estimator (TPE). RS tests hyperparameters randomly and is more efficient than grid search, as shown in [3]. In comparison, in [2], TPE models the conditional probability distribution of hyperparameters based on observed target values by separating the result space into areas with high and low probabilities of valuable outcomes. It approximates separate densities for the best and worst results and aims to maximize the ratio of these densities.

Impact of the Hyperparameters We analyzed the influence of the different hyperparameters on the quality of the result. For this purpose, we conducted a case study with $N \in \{3, 4, \ldots, 10\}$ waypoints in which we configure the PSO and PSO+ with varying hyperparameters $\lambda \in \{G, P\}$, which differ in the initialization. We compare the results of the algorithms against a global solution obtained with brute force methods (BF) through

$$d(P^k_{\text{global}}) = \frac{|P^k_{\text{global,alg}} - P^k_{\text{global,BF}}|}{P^k_{\text{global,BF}}} \text{ with alg} \in \{\text{PSO, PSO+}\} \tag{6}$$

as the normalized difference.

In fig. 4, we illustrate the influence of the hyperparameters on the result. The PSO+ algorithm performs significantly better than the PSO. We recognize a correlation between the number of generations produced and the population size increases exponentially as the number of waypoints rises. This illustrates that selecting hyperparameters is not a simple task. It requires a careful balance between the number of generations, the population size, and the number of waypoints. While a higher number of generations can improve the chances of finding a global solution, it also increases computation time. Therefore, keeping these values as small as possible is crucial without compromising the search for a global solution. Both tested algorithms, particularly the PSO algorithm, can achieve high accuracy within a certain range despite variations from the global solution. Notably, their minimum plateaus are similar, indicating that a common set of hyperparameters can lead to similarly effective results for both PSO-based algorithms.

Choice of HPO-Algorithm We conduct an additional analysis to determine which of the possible algorithms, RS or TPE, is most suitable for our use case. We define the following hyperparameters: population size P, number of generations G, and the probabilities for the best local solution η_{pbest} and the best global

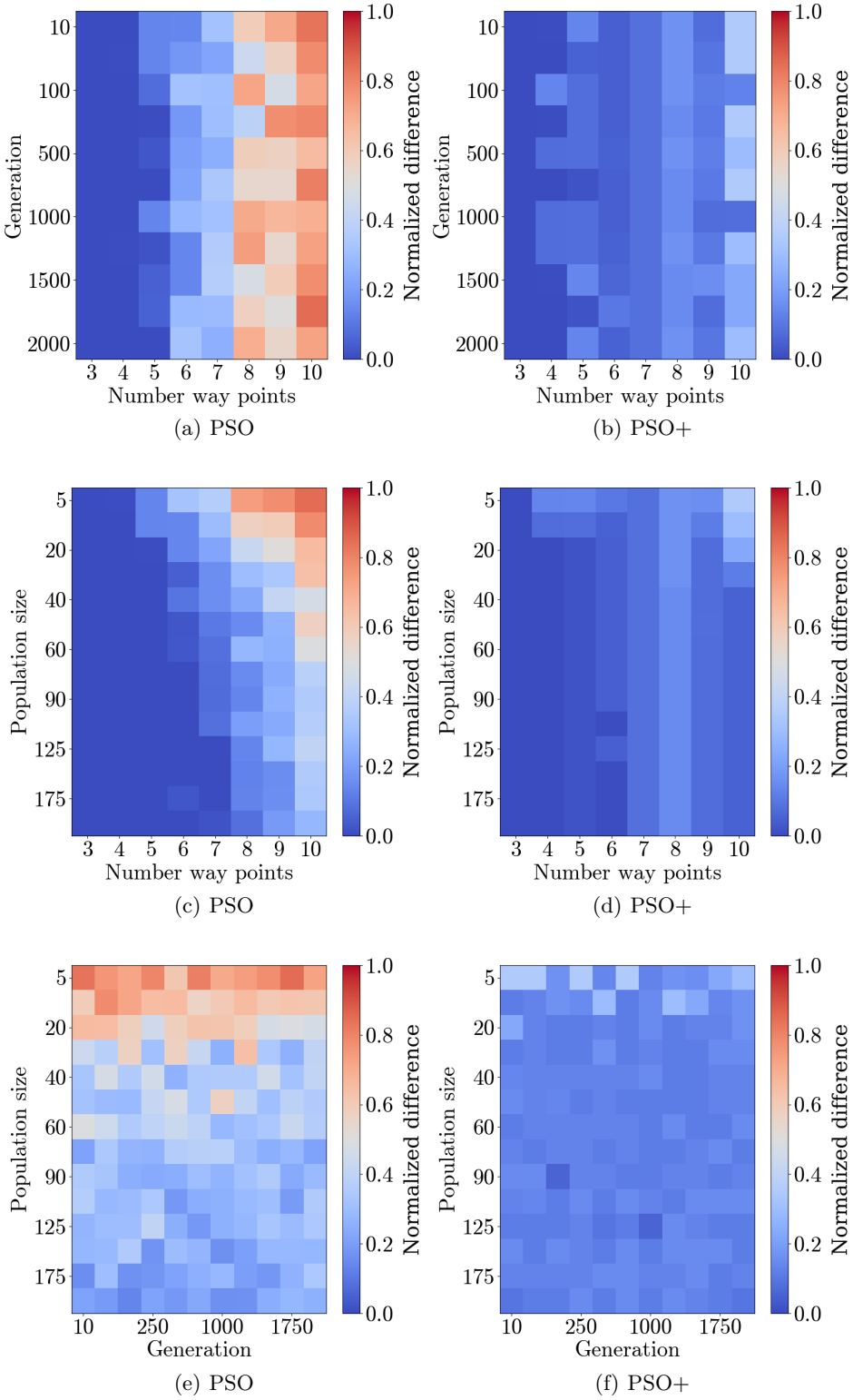


Fig. 4. Dependency of the hyper-parameter for the PSO-based algorithms.

solution η_{gbest}. Additionally, we are interested in the number of AUVs, which effectively corresponds to the number of waypoints assigned per AUV for a given total number of points.

To address this, we extend our set of hyperparameters, where M denotes the number of AUVs,

$$\lambda = \{P, G, \eta_{pbest}, \eta_{gbest}, M\} \ , \tag{7}$$

allowing us to formulate the optimization problem as follows:

$$\lambda^{(*)} \equiv \arg\min_{\lambda \in \Lambda} \Psi(\lambda) \ . \tag{8}$$

We aim to minimize the hyperparameter response function over the search space Λ defined as $\lambda \in \Lambda$. With the information presented in section 4.2, we select the configuration such as the search space Λ shown in table 1. We optimize the parameters for different sets of waypoints and missions to ensure the optimized hyperparameters apply beyond a specific problem. This approach enhances the robustness against the sensitivity of the number of waypoints that each AUV must navigate.

We define the objective function $\Psi(\lambda)$ as

$$\Psi(\lambda) = P_{\mathrm{el}}(t) + \omega_1 \cdot M + \omega_2 \cdot \overline{\sigma} + \omega_3 \cdot \sigma_{scale} \tag{9}$$

the simulated electric power $P_{\mathrm{el}}(t)$ used from the Li-ion battery, the mean of the state-of-charge (SoC) at the end of the mission, and σ_{scale} is a scaled version of the SoC

$$\sigma_{scale} = \sum_{0}^{N-1} \sum_{0}^{M-1} \frac{\sigma_i - min(\sigma_i)}{max(\sigma_i) - min(\sigma_i)} \ . \tag{10}$$

Since the individual components of the objective function have different units and magnitudes, we apply weighting factors ω_1, ω_2, and ω_3 to balance their contributions. These factors ensure that each term appropriately influences the optimization process, preventing any single term from dominating due to scale differences.

The choice of these weighting factors is motivated by the need to prioritize specific aspects of the mission, depending on the application context. For instance, in specific scenarios, minimizing the number of AUVs (i.e., reducing energy usage) might be more critical, while in others, maximizing the state-of-charge (SoC) could be the key objective. The weightings thus serve as a control parameter that allows for flexible optimization based on the mission's priorities, which we can adjust according to the specific needs of the task.

5 Evaluation

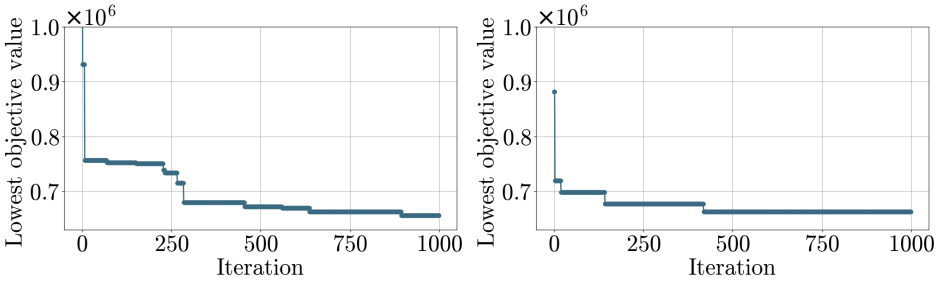
When applying HPO to both PSO-based algorithms, the maximum number of iterations alongside the objective function was necessary. This is crucial because, on the one hand, it allows for a more accurate approximation of the global optimum. On the other hand, it helps to limit iteration and avoid unnecessary

	HPO	RS-PSO	TPE-PSO	RS-PSO+	TPE-PSO+
Max. iter (HPO)		500			
number of waypoints	$N \in \{80, 100\}$	$N \in \{70, 75, 80, 85, 90, 95, 100, 105\}$			
number of missions	$K = 2 \cdot N_{AUV}$	$800 \rightarrow 100$ per setup			
	Search space	Optimal hyperparameter			
P	rand(5, 250)	7	72	103	172
G	rand(100, 500)	207	271	286	301
η_{gbest}	uniform(0.7, 1.0)	0.83	0.82	0.94	0.93
η_{pbest}	uniform(0.7, 1.0)	0.73	0.88	0.9	0.96
M	rand(5, 20)	12	10	10	11
		Computational time T_{comp}			
mean(T_{comp}) [s]		0.899	1.917	2.28	3.149
var(T_{comp}) [s^2]		0.243	0.403	0.459	0.486
$\text{var}^2(T_{\text{comp}})$ [s^4]		0.059	0.163	0.211	0.236

Table 1. Overview of the HPO configuration, optimal hyperparameters, and computational time for each algorithm variant, including the number of waypoints, missions, and key parameters.

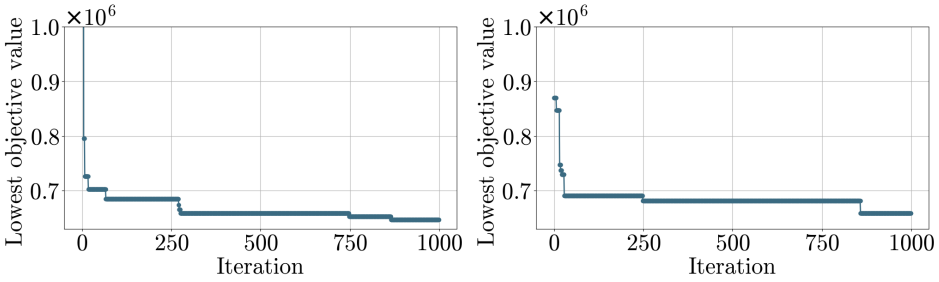
resource consumption. The convergence plot in fig. 5(a-d) shows the minimum objective function value at each iteration across 1000 iterations of the HPO. This allows for a visual assessment of how quickly the optimization approaches the global optimum. The advantage of the PSO+ approach becomes apparent since its greedy initialization enables the HPO algorithm to start with a lower objective function value. Although TPE-PSO begins with the highest initial cost, it achieves the lowest objective function value by the end of the optimization process. After approximately 500 iterations, further improvements stagnate, indicating diminishing returns in prolonged optimization. However, since each iteration incurs a computational cost, and we are considering future applications where PSO may serve as a local planner for AUVs, we need to evaluate whether achieving results with fewer iterations, which approach the minimum, is sufficient.

To evaluate this more precisely, we use an internal simulation routine. We consider the AUV failures and the SoC as a function of the maximum iterations, shown in fig. 5(e-f). To test against uncertainties and the result's robustness, we generated 100 missions with $N \in 70, \ldots, 105$ waypoints. Since we aim to determine hyperparameters that generalize beyond a specific problem instance, we deliberately varied the number of waypoints. This allows us to assess the robustness of the selected hyperparameters across different mission sizes. Specifically, we aim to test the transferability of the results by expanding the range of waypoints from 70 to 105 while we initially determine the HPOs for missions with 80 and 100 waypoints. Both plots confirm the observations made in the convergence analysis. Overall, the failure rate is lower for PSO than for PSO+. However, this is mainly due to RS-PSO+, which performs significantly worse than the other variants. In contrast, TPE-PSO+ delivers results comparable to



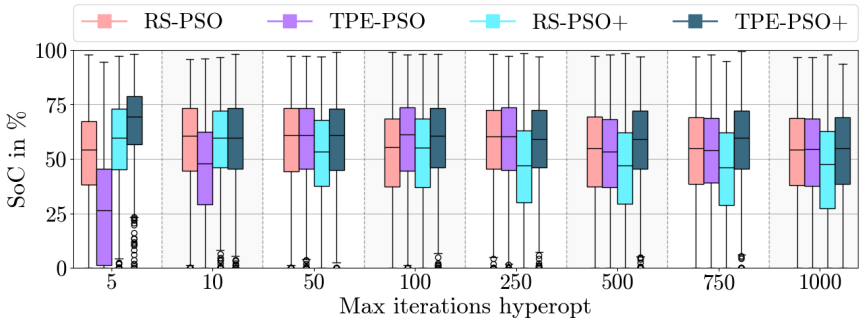
(a) RS-PSO: init value = $21.60 \cdot 10^5$; best value = $6.56 \cdot 10^5$

(b) RS-PSO+: init value = $8.82 \cdot 10^5$; best value = $6.63 \cdot 10^5$

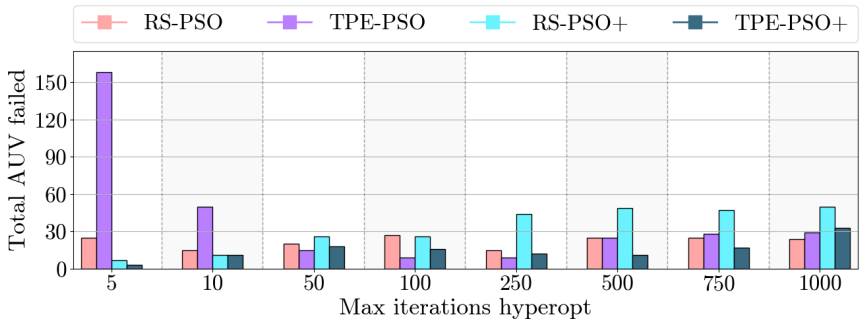


(c) TPE-PSO: init value = $30.90 \cdot 10^5$; best value = $6.47 \cdot 10^5$

(d) TPE-PSO+: init value = $8.7 \cdot 10^5$; best value = $6.59 \cdot 10^5$

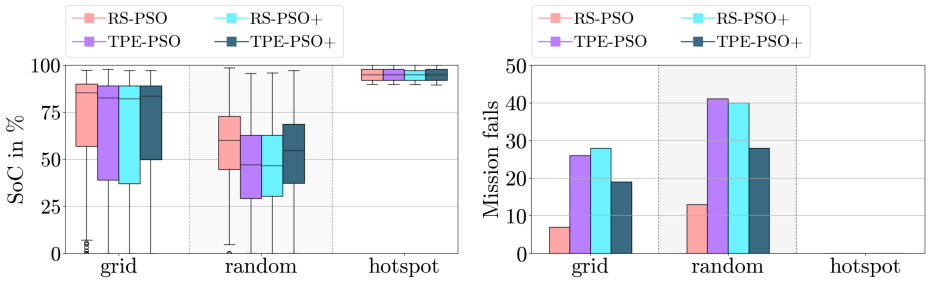


(e) State of Charge per AUV



(f) Failed AUVs in total

Fig. 5. Results of the HPO: (a-d) Outer Optimization, Convergence plot of the different algorithms as a function of the maximum iterations; (e-f) Inner Optimization, AUV's *SoC* and total number of failure depending on the chosen maximal iterations.



(a) SoC per test fields (b) Mission fails per test case

Fig. 6. *SoC* and mission failure rate for all algorithm variants across three test scenarios with HPO over 500 iterations.

TPE-PSO. Additionally, as the maximum number of iterations increases, the failure rate of AUVs remains lower for PSO, and their *SoC* stays higher than in PSO+, indicating more efficient energy utilization. However, improvements stagnate after approximately 500 iterations, suggesting diminishing returns from further optimization.

Considering the reduced computation time, we now aim to conduct a more in-depth evaluation to assess how well this configuration performs across different test scenarios under different distributions:

1. Grid: An arrangement typical to pattern tracing methods, where the coordinates are evenly spaced in a square field.
2. Random: Randomly distributed coordinates in the field that follow a regular distribution.
3. Hotspot: Coordinates distributed randomly around three centers.

We test the determined hyperparameters for robustness against deviations in the number of waypoints and the type of distribution in the field. The evaluation plot shown in fig. 6 presents the outcomes for the *SoC* and mission failures across three test cases using the hyperparameters at 500 iterations for the HPO. We define a mission as failed if any AUVs do not complete their assigned tasks.

Furthermore, we averaged the computational time across all simulation test scenarios, as shown in table 1. The computational times were measured on a Dell Latitude 7430 laptop, equipped with an Intel Core i7 processor and 16 GB of RAM, running Ubuntu 22.04. We perform the experiments by running each scenario 240 times to ensure statistical significance. The results indicate that all algorithms achieve similar solutions with few mission failures. RS-PSO delivers the best results in mission fulfillment and energy consumption, with the smallest population and fewest generations, leading to the lowest computational time, making it ideal for resource-limited applications. However, it requires the highest number of AUVs, increasing costs. In contrast, RS-PSO+ and TPE-PSO+ have higher computational times but use fewer AUVs, making them more cost-effective when AUVs are limited.

In the hotspot field scenario, shorter routes result in surplus energy, suggesting that fewer AUVs would be sufficient. This highlights a potential improvement for HPO by incorporating waypoint distribution into the optimization process.

In conclusion, with optimized parameters, traditional PSO performs as well as the enhanced versions, suggesting that improvements may not be necessary. Despite using fewer iterations than indicated by the convergence plot, the results remain robust. A good strategy would involve many iterations for global path planning and fewer for local adjustments, thereby reducing computational effort.

6 Conclusion and Outlook

In this work, we investigated how HPO can improve the robustness and resource utilization of PSO for energy-aware AUV mission planning. Our study focused on determining optimal hyperparameters to balance exploration and exploitation, ensuring that PSO delivers energy-optimal solutions for waypoint assignment and path planning. Defining PSO configuration parameters as hyperparameters allows for a systematic optimization process rather than relying on trial-and-error tuning. Through real-world-inspired simulations, we validated that our HPO-driven approach enhances mission success rates, reduces energy consumption, and enables a more precise estimation of the minimum required number of AUVs for a given mission.

A key finding from our simulations is that with optimized hyperparameters, the traditional PSO performs almost as well as the improved PSO+ variant. This suggests that, under the proper parameter settings, the added complexity of PSO+ may not always be necessary. Additionally, while a high number of iterations is beneficial for determining the global path at the base station, significantly fewer iterations are sufficient for local adjustments on the AUVs. This insight can help reduce computational effort in real-world setups.

This work enhances energy-efficient path planning and lays the groundwork for future autonomous swarms. Adaptive configurations for local path planning allow the system to respond dynamically to environmental changes. Additionally, swarm intelligence strategies could enable AUVs to redistribute waypoints in case of individual failures, improving mission robustness. Validation through real-world missions is essential to assess practical applicability and refine optimization based on actual ocean conditions.

References

1. Alfred-Wegener-Institut, Helmholtz Zentrum für Polar- und Meeresforschung: Tiefsee-Ökologie und -technologie (2024), `https://www.awi.de/forschung/biowissenschaften/tiefsee-oekologie-und-technologie/technologie.html`
2. Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for Hyper-Parameter Optimization. Advances in Neural Information Processing Systems **24** (2011)

3. Bergstra, J., Bengio, Y.: Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research **13**(2) (2012)
4. Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., Cox, D.D.: Hyperopt: A Python Library for Model Selection and Hyperparameter Optimization. Computational Science & Discovery **8**(1), 014008 (2015)
5. Conmy, R.N., DiPinto, L., Kukulya, A., Garcia-Pineda, O., Graettinger, G., Sundaravadivelu, D., Gloekler, M., Hall, A., Fischell, E., Gomez-Ibanez, D.: Advances in Underwater Oil Plume Detection Capabilities. In: International Oil Spill Conference. vol. 2021, p. 1141330 (2021)
6. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT press (2022)
7. FONA - Forschung für Nachhaltigkeit: Unterwasserfahrzeuge der deutschen Meeresforschung (2024), `https://www.fona.de/de/massnahmen/forschungsinfrastrukturen/unterwasserfahrzeuge.php`
8. Fossen, T.I.: Handbook of Marine Craft Hydrodynamics and Motion Control. John Wiley & Sons (2011)
9. Frenkel, W., Renner, B.C.: Towards Energy-Aware Path Planning for AUV Swarms. In: 2024 18th ICARCV. pp. 1106–1111. IEEE (2024)
10. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: ICNN'95. vol. 4, pp. 1942–1948. ieee (1995)
11. Kuehner, I.: Swarm Intelligence for Solving a Traveling Salesman Problem. eKNOWN (2020)
12. Liu, X., Su, J., Han, Y.: An Improved Particle Swarm Optimization for Traveling Salesman Problem. In: ICIC 2007, Qingdao, China, August 21-24, 2007. Proceedings 3. pp. 803–812. Springer (2007)
13. Lu, C., Yang, J., Leira, B.J., Chen, Q., Wang, S.: Three-Dimensional Path Planning of Deep-Sea Mining Vehicle Based on Improved Particle Swarm Optimization. Journal of Marine Science and Engineering **11**(9), 1797 (2023)
14. Martin-Abadal, M., Oliver-Codina, G., Gonzalez-Cid, Y.: Real-Time Pipe and Valve Characterisation and Mapping for Autonomous Underwater Intervention Tasks. Sensors **22**(21), 8141 (2022)
15. Messon, D., Verma, D., Rastogi, M., Singh, A.: Comparative Study of Time Optimization Algorithms for Traveling Salesman Problem. In: AICTC 2021, pp. 555–566. Springer (2022)
16. Qu, N., Chen, G., Shen, Y.: A Three-Dimensional Path Planning System for AUV Diving Process Considering Ocean Current and Energy Consumption. In: OCEANS 2021: San Diego–Porto. pp. 1–7. IEEE (2021)
17. Reuter, J., Mank, E., Aschemann, H., Rauh, A.: Battery State Observation and Condition Monitoring Using Online Minimization. In: 2016 21st MMAR. pp. 1223–1228. IEEE (2016)
18. Villwock, A., Kersten, C.: From the Deep Sea to The Atmosphere: GEOMAR Helmholtz Centre for Ocean Research Kiel (2015)
19. Wang, K.P., Huang, L., Zhou, C.G., Pang, W.: Particle Swarm Optimization for Traveling Salesman Problem. In: Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE cat. no. 03ex693). vol. 3, pp. 1583–1585. IEEE (2003)
20. Woods Hole Oceanographic Institution: AUVs (2025), `https://www.whoi.edu/what-we-do/explore/underwater-vehicles/auvs/`
21. Yang, L., Shami, A.: On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice. Neurocomputing **415**, 295–316 (2020)