mathematics

MDPI

Article

Smart Exploration of Lentic Cyanobacterial Water Bodies Supported by Model-Based Simulation, Autonomous Surface Vehicles and Evolutionary Algorithms

Gonzalo Carazo-Barbero *, Eva Besada-Portas, José Antonio López-Orozco and José Luis Risco-Martín

Department of Computer Architecture and Automatics, University Complutense of Madrid, Avenida de Séneca 2, 28040 Madrid, Spain; ebesada@ucm.es (E.B.-P.); jalopez@ucm.es (J.A.L.-O.); jlrisco@ucm.es (J.L.R.-M.)
* Correspondence: gocarazo@ucm.es

Abstract

Cyanobacterial blooms in lakes and reservoirs pose significant environmental and public health risks. This paper presents an effective exploration strategy to detect them from Autonomous Surface Vehicles (ASVs) equipped with probes, whose sensing trajectories are optimized by an AI-based planner that considers the 3D spatial-temporal evolution of the cyanobacteria concentration obtained by a multiphysics model. The planner, simultaneously working on the AI decision-making and robotic domains, optimizes the surface displacement of the ASV and the depth of its probe by solving a constrained multi-objective optimization problem that minimizes the mission duration and trajectory length, maximizes the possibilities of the probe to overpass areas with high concentration of cyanobacteria, and satisfies operational constraints (such as ASV velocity or acceleration limits, and obstacle avoidance). The optimization is supported by two well-known versions of the Non-Sorted Generic Algorithm (NSGA-II and NSGA-III) and by encoding the trajectories with spline curves whose number of control points can be fixed, progressively increased, or freely manipulated by the algorithm. The performance of different configurations of the planner is tested against six scenarios obtained from different simulations of the multiphysics model (which couples water dynamics and temperature, light transmission, daily vertical migration of the cyanobacteria and their growth). The statistical analysis of the planner results determines that NSGA-III working with variable-length chromosomes and NSGA-II with the progressive increment of spline points as the best configurations for maximizing cyanobacteria detection, and minimizing mission duration and trajectory length.

Keywords: cyanobacterial blooms; Autonomous Surface Vehicles; trajectory planning; non-dominated sorting genetic algorithm; multiphysics modeling

MSC: 68W50; 76-10; 92-10; 68T40

Check for updates

Academic Editor: Jüri Majak

Received: 14 March 2026
Revised: 8 May 2026
Accepted: 20 May 2026
Published: 24 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The bloom (i.e., massive growth) of cyanobacteria (also known as blue-green algae) in lentic water bodies (e.g., lakes and reservoirs) often becomes a problem for the environment and public health [1,2]. Their environmental effects are notable, commonly depleting the waters of different resources (e.g., light, dissolved oxygen, phosphorous and nitrates) and negatively impacting the survival of other microorganisms. The toxins released by some species and strains of cyanobacteria also cause the death of large animals (to the point of representing an ecological disaster) and provoke in humans different health problems (e.g.,

allergic reactions, skin rashes, and serious neurological, respiratory, gastrointestinal and hepatic issues) [3,4]. The negative economic impact of their troubles (aggravated by the costs associated with water treatments required when the cyanobacteria levels increase in reservoirs designated for human consumption) cannot be neglected either.

All in all, thanks to an increasing awareness of the water authorities about the concerns regarding cyanobacteria blooms, the international regulations related to their presence in water are hardening. The new laws [5,6] indicate that cyanobacteria blooms should be monitored more frequently at multiple locations to be able to anticipate them and improve the management of the water bodies where they appear. However, it is still common to find reservoirs used for human consumption or recreational activities that have developed significant cyanobacterial populations. Hence, it is necessary to implement monitoring and predicting approaches that allow anticipating cyanobacterial blooms, and provide early warnings and reaction time for applying preventive measures (e.g., the closure of recreational areas, changes of water treatments or alternative locations of the origin of water supplies).

There are different methods for monitoring water quality and cyanobacterial state, including taking water samples manually [7], automatically measuring data from stations placed at fixed location [8,9], and collecting information from sensors onboard Autonomous Surface Vehicles (ASVs, a type of robotized ship, ref. [10]). In the first approach, an expert free to take samples at any location could provide good quality data of representative areas, if the time requirements to perform these kinds of missions with the necessary frequency were not prohibitively expensive. The second strategy allows continuous measurements taken automatically from stations placed at fixed areas, but prevents the observation of important cyanobacteria developments in other locations of the water body. Although this problem can be mitigated by distributing multiple stations (e.g., in form of buoys) within the water body, the spatial resolution of the collected data will be low, unless the number of stations (and therefore the cost of the approach) is significantly increased [11]. Finally, equipping ASVs with sensors tackles the main limitations and reduces the costs of the previous methods: the lack of an expert within the monitoring process makes possible to carry it out at the required frequency, and since ASVs are not bound to a specific location, they can gather information from any area of the domain. For this reason, this work focuses on the detection of cyanobacteria blooms using ASVs as mobile monitoring stations.

Often, there is a disparity between the area that can be covered by an ASV in a single mission and the size of the water domains under observation. On one hand, the lakes or reservoirs dedicated to recreational use and water consumption that we are interested to monitor are relatively big (covering areas of tens of km^2). On the other hand, the duration of ASV's mission (around 3 h) is bounded by its battery life, connection range and allocated mission time. The small size and payload of our ASVs also limit their speed (rarely exceeding 1 m/s, making the maximal distance traveled during a mission around 10 km). As the situation of cyanobacteria blooms can change drastically in short distances (since water currents segregate some areas from others), covering the whole surface of the water bodies with segments spaced apart (e.g., 50 m) requires a trajectory of hundreds of km, well above the capabilities of our ASVs. Therefore, instead of carrying out the previous exhaustive inspection of the water domain, this paper proposes to perform a *smart* exploration, supported by information of the areas where the cyanobacteria blooms are most likely to happen, ASVs and Artificial Intelligence (AI)-based path planning techniques.

The information of the cyanobacterial distribution can come from different sources. An option consists of using expert opinions [12,13]. However, this strategy is not valid for sites that have not been deeply studied. In addition, conclusions drawn from past experiences may not apply under changing climatological conditions. The alternative,

consisting of predictions by models based on data [14] is also problematic, since it requires large sets of historical data and does not generalize well either to changing environments. To overcome both issues, this work uses predictive models based on processes [15,16], since they require minimal data (collectable as part of a short preliminary study) and the outputs of their simulations can be modified by changing their initial conditions and inputs.

In particular, we build a physical predictive model that takes into account the water dynamics, the light transmission, the water temperature, the daily vertical migration of the cyanobacteria and their growth. Simulations of this model with the appropriate initial conditions and inputs let us determine the plausible three-dimensional and timely distribution of the cyanobacteria within the water body. This information, in turn, is used by an AI-based planner that optimizes the surface displacement of the ASV and the depth of its probe with the purpose of reducing the mission duration and the 3D length of the ASV+probe trajectory, while making the probe overpass areas where it is expected to find high concentrations of cyanobacteria. In order to handle also the water domain and ASV+probe physical constraints (e.g., finite navigation space, and maximal velocities of the ASV and of the probe) while optimizing the former conflicting objectives, the planner is supported by well-grounded multi-objective constrained evolutionary strategies of the Non-Sorted Genetic Algorithms (NSGA, refs. [17–19]), which have been already used in autonomous vehicle path planning problems [20–27]. Besides, the 3D trajectory is compactly encoded by 3D cardinal splines [28], which consists of consecutive polynomial segments defined by control points, whose number is handled by our planner with three different strategies in order to balance the search space exploration and exploitation efforts of NSGA. Specifically, our planner can work with a fixed number of control points [29–31], increase them progressively [32], or simultaneously manipulate solutions with different number of control points in order to smartly optimize their values and ideal number [33,34].

In order to determine which option (algorithm and control point management strategy) is the best, the stochastic nature of NSGA requires to statistically analyze the results obtained by the planner in different scenarios [35,36]. Commonly used non-parametric methods such as Friedman test [37] only provide partial insights to our multi-objective constrained problem, since they are not directly applicable to population sets containing unfeasible Pareto fronts, or studies in which the result groups have different sizes or variances. Therefore, we also present an innovative statistical analysis framework that combines the capabilities of the Skillings-Mack [38], Nemenyi [39] and Mann-Whitney U [40] statistical tests to compare the results obtained during the entire optimization processes of the different variants of our planner.

The main contributions of the paper, which combines the robotic and decision-making application domains of AI, are the following:

- *A physical model that can simulate the evolution and distribution of cyanobacteria in a three-dimensional environment*, by coupling submodels that replicate fluid dynamics, varying water properties, light propagation in water, vertical migration of cyanobacteria driven by buoyancy changes, passive transport of cyanobacteria colonies, and growth dynamics of cyanobacteria populations.
- *An AI-based multi-objective constrained planner that minimizes the mission time and the length of the 3D ASV+probe trajectory, while maximizing the observation of areas which are likely to have high concentration of cyanobacteria (according to the simulations)*. To this end, we formulate a multi-objective constrained problem and solve it with two NSGA variants that use several strategies for managing the number of control points of 3D-splines that encode the ASV+probe trajectories.
- *The comparison of the performance of the optimizer variants under 6 simulated scenarios*, which have been carefully set up to impose different difficulties to the planner. This

can be done thanks to *a novel statistical analysis framework* that has been developed *to analyze the results of multi-objective constrained optimization algorithms throughout the optimization process*, regardless of the population sets having differing sizes and variances due to the experimental design or the production of unfeasible solutions in the early optimization stages.

These contributions build upon our previous work in [32], which introduced a baseline planner for optimizing a 3D ASV trajectory using a dynamic model of cyanobacterial blooms. The most relevant improvements introduced in the current version of the planner, which is presented in this paper, are:

- The physical model includes a new biological growth model and adds water pressure, density, and temperature variables to simulate how the thermocline and pycnocline affect cyanobacterial transport and buoyancy.
- The optimization engine of the planner incorporates NSGA-III to better handle the multi-objective search space and objective space covering.
- The trajectory encoding introduces variable-length chromosomes supported by new proportional crossover and mutation operators, allowing the planner to simultaneously optimize the number of spline segments and their parameters.
- The trajectory refinement step includes a kernel-based damping function that distributes the new trajectory control nodes taking into account the expected cyanobacterial distribution and the location of the existing nodes of the trajectory.
- The evaluation protocol of the planner performance features a new non-parametric statistical framework combining the Skillings-Mack, Nemenyi, and Mann-Whitney U tests to properly analyze populations with unfeasible solutions and differing sizes throughout the entire optimization process.
- The testing scope is expanded to six distinct simulated scenarios designed to impose different spatial and temporal difficulties on the planner.

This paper is structured as follows. Section 2 introduces the state of the art on prediction models for cyanobacteria and on autonomous vehicle AI-based planners. Next, Section 3 presents the mathematical model of cyanobacterial growth and transport, and Section 4 sets up and simulates different scenarios. Afterwards, Section 5 describes all the characteristics of our AI-based 3D trajectory planner, and Section 6 presents an statistical analysis of the results obtained for the different scenarios. Finally, Section 7 draws the conclusions of this work and presents some future lines of research.

2. State of the Art

As mentioned in the introduction, our strategy comprises two key components: a predictive model to simulate the growth and movement patterns of cyanobacteria, and a trajectory planner to optimize the 3D trajectory that determines the best surface displacement of the ASV and depth of its probe for validating the cyanobacteria forecasts. The relevant state of the art for both components is respectively presented in Sections 2.1 and 2.2.

2.1. Prediction of Cyanobacteria

Although the interest in cyanobacteria research is reflected in the development of many partial models that predict specific cyanobacterial behaviors [41–46], only a few of them [47,48] capture their full three-dimensional spatio-temporal dynamics, due to the inherent complexity of modeling together all their governing processes (which include biological mechanisms that vary among species and multitude of physical processes that interact with them) and to the uncertainty introduced by all the information needed to tune their parameters. These findings are backed by the systematic literature reviews presented in [16,49], which highlight that many models lack crucial cyanobacterial behaviors (such

as their vertical migration) or fail to account for significant environmental effects (such as water temperature), affecting gravely the quality of their predictions.

In the following two sections, we further analyze different aspects of the two main approaches used for cyanobacteria prediction.

2.1.1. Data-Based Predictions

Numerous data-driven prediction models have been proposed, although they often limit dimensionality and focus on specific regions.

Some of them use classical approaches. For instance, ref. [50] uses wavelet threshold filters, clustering, and regression models to predict chlorophyll-a from temperature, pH and dissolved oxygen; ref. [51] couples genetic algorithms with model trees to show a strong correlation between the appearance of blooms and the temperature; ref. [52] takes a direct statistical approach and uses canonical correspondence analysis to identify driving factors in the growth of six cyanobacterial species. Focused on particular aspects of the problem, they cannot determine the spatio-temporal distribution of the bloom.

Recently, there has been a shift towards Artificial Neural Network (ANN) models and deep learning [53]. For instance, refs. [54,55] predict the concentration of chlorophyll-a using an ANN+support vector machine and a convolutional neural network with an attention module; while [56] locally predicts the concentration of phycocyanin (a better proxy than chlorophyll-a to predict the presence of cyanobacteria) with a Long Short-Term Memory (LSTM) network. Again, the locality of the approaches makes them invalid for us.

The combination of ANNs with other statistical or process-based models to improve the predictions is also explored. In this regard, refs. [57,58] show that the predictions of their LSTM networks are improved when combined with a process-based predictor for hydrodynamic data and with an auto-regressive integrated moving average module. Although some of the shortcomings of isolated ANNs are resolved, these approaches bring a new level of complexity, while maintaining shortcomings related to the scarcity of data or their inability to predict extreme scenarios accurately.

Satellite hyperspectral imagery is often used for monitoring the blooms [59–63] and sometimes also in prediction models. For instance, ref. [64] uses them to feed a Gaussian Process, while [65] combines the cyanobacterial concentrations estimated from satellite data with a hydrodynamic model. The low resolution of the images, the low frequency of the observations, and the inaccuracy in the vertical component of the measurements reduce the spatio-temporal precision of these models (especially for small inland water bodies and to determine the cyanobacteria vertical migration).

2.1.2. Process-Based Predictions

An alternative is to describe the physical phenomena through mathematical (often differential) equations. This approach allows to model specific behaviors without relying on big amounts of (often unavailable) data. Hence, we follow this strategy, which is next analyzed with higher detail.

Many of these models focus on cyanobacterial vertical migration. The review in [66] concludes that particle-tracking methods provide better predictions than continuous methods. Some of them [42,43,67] consider the effect of light irradiance as a driver of cellular density regulation, while others [68,69] assume constant daily cycle velocities (lacking accuracy and the ability to adapt to the seasonal lighting conditions). The former 1-dimensional (in depth) models are independent of other factors such as horizontal transport, temperature variations, domain geometry or nutrient availability.

A light transmission model helps to determine the light dynamics at different depths, because it is easier to measure the irradiance at the water surface than inside the water at

different depths. This allows to use data from available weather stations and models, since the scattering effects of the atmosphere are almost negligible. The simplicity and accuracy of the Beer-Lambert law [70] makes it a widely used model for this purpose.

There are other relevant physical phenomena that affect the location of cyanobacteria. In this regard, ref. [71] analyzes the contributions of different phenomena to the transport of cyanobacteria using specialized satellite data. Vertically, it shows that, not only does the irradiance contribute to cyanobacteria buoyancy, but also the temperature plays a significant role, as their migration velocity is higher in summer than in winter. Horizontally, it shows that, while the majority of the transport is caused by hydrodynamics, the wind also plays a significant effect. For this reason, it is important to include as many of these behaviors as possible in our final model.

Hydrodynamics affect the cyanobacterial transport, and also distribute nutrients and temperature to different areas. As they are ruled by the difficult-to-solve Navier-Stokes equations [72], different assumptions are often made to simplify their computation. A common practice is to reduce the dimensionality of the domain. However, one-dimensional longitudinal models focused on river flow and sediment transport [73,74] do not capture vertical stratification and lateral variations required for comprehensive cyanobacteria studies. Alternatively, one-dimensional vertical models [75,76] excel in simulating thermal structure and mixing, but ignore horizontal processes that affect the cyanobacteria distribution. Furthermore, since two-dimensional models simulating longitudinal and vertical dynamics [77,78] assume lateral homogeneity, they overlook critical cyanobacteria transport mechanisms in wider domains. Finally, three-dimensional models [79–83] offer detailed simulations of hydrodynamic processes, but are computationally intensive and require setting up many initial conditions and input data. The accuracy of their simulations is also affected by their assumptions regarding pressure, turbulence, temperature transport and density changes.

These shortcomings can be palliated by using calibration standards [84,85] based on factors like pressure and temperature to adjust the water density. We exploit this strategy to improve the accuracy of the water density and, consequently, of the cyanobacterial vertical migration in our model.

Using an accurate model of cyanobacteria growth is also important to obtain robust predictions, since there are several biological factors (beyond those driving vertical migration) that affect the formation and bloom of the cyanobacteria colonies [86]. Similarly to cellular density changes, light incidence on the colony also plays a crucial role in their growth. Consequently, several growth models (e.g., [87,88] and ours) consider the relationship among irradiance, the photosynthetic capability of the colonies [44] and their size.

There are also attempts to combine many of the previous factors. They usually consist in coupling modular submodels [47,48]. Due to their complexity, they have such a big number of tuning parameters that is challenging to obtain enough data to determine all their values. In an attempt to become practically feasible, they also limit the dimensionality of the problem [89] or miss important dynamics such as growth [90] or accurate vertical migration induced by cellular density changes [82].

To make this gap explicit, Table 1 summarizes which physical and biological models are simultaneously considered by representative process-based references, showing in the first columns the publications grouped in the same row for sharing properties and in each of the remaining columns the different types of process that can be included in the model (and that are included in the models of each row when marked with ✓).

Table 1. Comparison of representative process-based cyanobacteria models, indicating with a check mark the processes that are included in each model.

Publications	Irradiance	Vertical Migration	3D Hydro. Transport	Temperature	Pressure	Water Density	Growth
[42,43,67]	✓	✓					
[68]	✓	✓					✓
[69]	✓	✓		✓			
[87,88]	✓						✓
[44]	✓			✓			✓
[45,46]							✓
[47,89]	✓	✓		✓			✓
[90]	✓	✓					
[48]	✓	✓	✓	✓			✓
[82]	✓		✓	✓	✓		✓
Ours	✓	✓	✓	✓	✓	✓	✓

Considering the state of the art and the different behavior under study in Table 1 our process-based model stands out for already encompassing the most representative physics, including horizontal and vertical transport, irradiance, cell density changes, colony growth, and effects related to the water temperature and pressure. Being backed up by particle-based and discretized models, it can be improved in the future to include other phenomena such as the effects of the wind in the distribution of cyanobacteria near the water surface [91,92] and of the competition for the nutrients of different species of microorganisms in the cyanobacteria growth [45,46,93].

2.2. Optimization

Many methods in the state of the art developed for planning the paths of robots and autonomous vehicles can be used to determine the optimal 3D trajectory of the ASV+probe.

Classical approaches like A* [94] and Dijkstra's algorithm [95] are grid-based methods focused on finding the shortest path. However, since the best trajectory is not necessarily the shortest one in those applications that have to fulfill other optimization criteria, those classical algorithms are less used in more complex problems [96]. In addition, the limitations of these methods in high dimensional spaces justify the development of sampling-based trajectory planners such as Probabilistic Roadmaps (PRM, ref. [97]) and Rapidly-exploring Random Trees (RRT, ref. [98]). Although these new approaches address the dimensionality problem, the resulting trajectories still fail to consider other objectives and generate suboptimal paths lacking smoothness.

For these reasons, trajectory planning in autonomous vehicle missions often relies on optimization methods that can handle multiple objectives, restrictions and heuristics. Some planners of this type use methods customized to the problem [29,99–104]. Many others are supported by well-known optimization approaches such as: Graph-based Optimization [105], Particle Swarm Optimization [106–110], ant colony optimization [111–115], Simulated Annealing [116,117], or Tabu Search [118]. More references of this type can be found in [119,120]. Other popular options are Evolutionary Algorithms, including Differential Evolution [121–123], and Multi-objective Evolutionary Algorithms such as NSGA-II [20–24,124] or NSGA-III [26,27], among others [125–127], which are able to efficiently handle multiple objectives and constraints using Pareto fronts. Our approach for planning the 3D ASV+probe trajectory falls within this last group, as NSGA-II showed really promising results in the previous version of our approach [32].

The trajectory encoding also plays an important role in the optimization process, as different encoding of the solutions provide different opportunities to the planner and different characteristics to the trajectory. Planners based on sequences of instructions or on the concatenation of trajectory primitives [100,111,128] introduce global changes whenever an early section of the trajectory is modified, which can slow down the convergence of the optimization process. Waypoint-based encodings [20,21,107,113,116–118,121,129] represent the path as a sequence of discrete points in space that the agent must pass through, joined by straight lines or leaving the navigation from one to the next to the onboard software. This method is intuitive and easy to understand but can result in rough trajectories and lack of fine-grained control. Alternatively, spline-based encoding [29–31,122] offer a continuous and smooth representation of the trajectory using only a few parameters, while maintaining the flexibility by introducing a path definition given by piece-wise functions. Moreover, as splines can easily be adjusted to accommodate various constraints and objectives during the trajectory (which is not straightforward on other representations [124]), we use this encoding for the previous and current version of our planner.

Splines often used in path planning, include Bézier splines, B-splines, cubic Hermite splines and other non-standard spline representations. Planners supported by Bézier splines [29,102] intuitively manipulate the control points but not always provide the necessary smoothness and continuity. B-splines offer more flexibility and local control to the planners [30,103,122–124,130], being more suitable for complex path planning scenarios. Cubic Hermite splines provide a good balance between smoothness and control to the planners [31,104], with explicit representation of the tangent vectors at the endpoints, which is particularly useful in dynamic environments that require implicit constraints related to the endpoints. In addition, Hermite splines have the advantage of being interpolating splines, which simplifies and speeds up constraint checking. For these reasons, our planner uses a cardinal spline [28] that is essentially a particularization of cubic Hermite splines where the node tangents are controlled through a tensioning parameter. By setting this parameter to zero, we enforce C^2 continuity while retaining control of the endpoints, both in position and velocity.

Regardless of the encoding, during the optimization process the number of decision variables can remain fixed [22,121,122,128,130] or change [21,23,24,33,34,123]. In the latter case, the length of the chromosomes of the evolutionary algorithms can be incremented cyclically [23,24] or considered as a variable by the optimizer [21,33,34,123] to let it automatically decide the number of instructions or waypoints that define the trajectory. The approach in this paper can handle the three possibilities.

Finally, and focusing on ASV path planning for water quality monitoring, the final solutions provided by the planner are shaped through the objective functions of the missions. For instance, when there is no information about the pollutant distribution, the problem is set up as a coverage problem [126,127]. Otherwise, when the information in the regions of interest is relevant, the problem has been handled as an information gain problem [108,114]. Alternative options consists on optimizing the visiting ordering of the water monitoring points provided by an operator [115] or trajectories of multiple ASVs according to the water quality uncertainty modeled with a Gaussian Process and updated with the new ASVs measurements [109,110,125]. The approaches in this paper and in [25,32,131] consider the pollutant or cyanobacteria concentration and distribution obtained by a biological and physical simulator to simultaneously optimize (1) the pollutant or cyanobacteria concentration that is expected to be observed with the ASV probes, (2) the ASV+probe trajectory length, and (3) the mission duration.

3. Physical Model

Physical models are representations of reality that, through the use of mathematical equations or algorithms, help us to understand nature and make predictions about it. In this paper, we are interested in predicting the displacements and growth of cyanobacteria. These two behaviors, in reality, depend on numerous factors that make modeling the system extremely difficult. Nevertheless, we can approximate the model including the most representative parts of the physical phenomena related to them. Next, we describe all the elements of our model and the relationships among them.

3.1. Architecture

The coupling of the current submodels of our approach is reflected in Figure 1, where the submodels are represented as boxes, the inputs and outputs to other submodels as arrows, and the states of the particles associated to the colonies of the cyanobacterial bloom, which are updated and queried by several submodels, as ellipses. Besides, some elements are highlighted in green and blue, which respectively indicate if the element has been added since the version of our planner in [32], or if it has been improved. A special case is the biological growth model, which has been extensively upgraded to the point of being considered almost new.

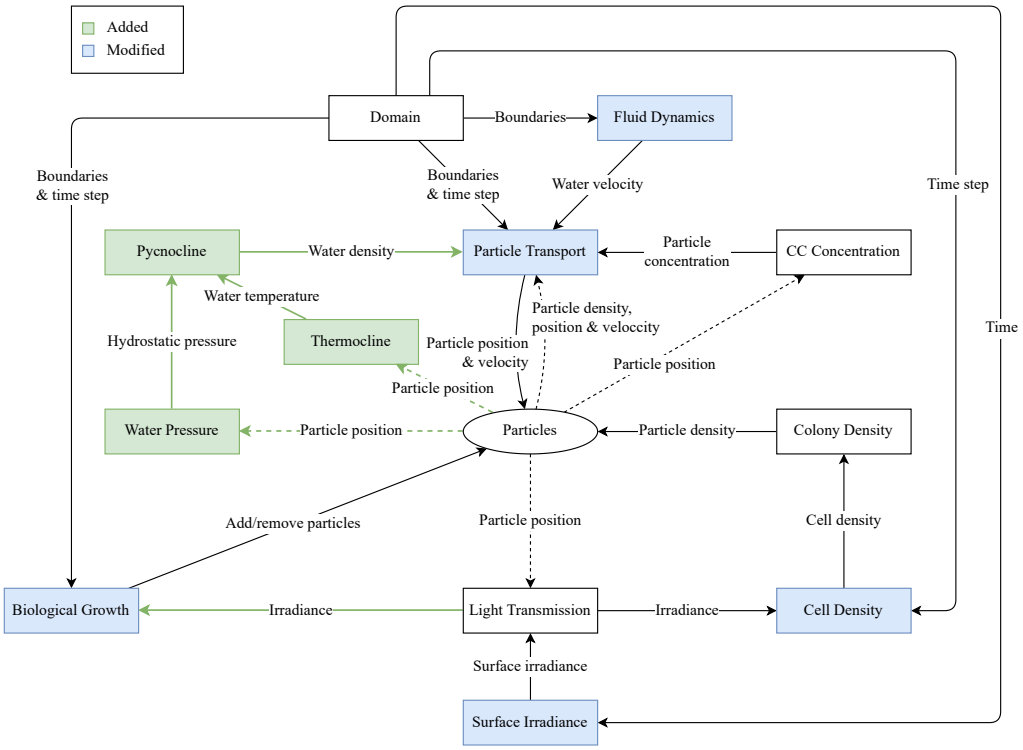


Figure 1. Block diagram of the model architecture.

Before going into further details, the spatial and temporal domain of the model is used in the fluid dynamics, the cell density, the biological growth and the particle transport models. Besides, the solution of the fluid dynamics is used in the particle transport model, which updates the position and velocity of each particle. The transport model also takes as inputs the particle concentrations and densities, as well as the water density (which is a function of the water temperature and pressure) at the particle positions. The particle density, which represents the density of the cyanobacterial colony, is determined by the density of the cells forming it, which vary due to incident irradiance fluctuations caused by changes in position or time of day. Finally, the irradiance also influences the biological

growth, which creates or destroys particles in order to model the reproduction and death biological processes.

Finally, it is worth noting that the model of the fluids dynamics is implemented in COMSOL [132], while the remaining models and the interactions among all of them are implemented in MATLAB [133].

3.2. Domain

The time domain $\mathcal{T}$ of the simulations is defined by its starting t_0 and final points t_f, while its physical domain Ω is the three-dimensional space matching the volume of the water body. We build Ω in Blender [134], using an unstructured 3D polygonal mesh defined from satellite imagery for the shore line and from bathymetric data for its depth. Its boundaries are divided into the surface of the water body Γ_{surf}, its bed Γ_{bed}, its N_{in} inlets Γ^i_{in}, and its N_{out} outlets Γ^j_{out}. To simplify the implementation of other submodels, we make Γ_{surf} coplanar to the $z = 0$ plane.

The discretization of the time domain is a streamlined even division in sub-intervals of size Δt. Due to the Lagrangian nature of the majority of the submodels, the spatial domain only needs to be discretized for the fluid dynamics, which benefits from using a free tetrahedral mesh with two layers of thin hexahedral elements around Γ_{bed}, in order to maintain the smoothness and increase the convergence of its simulations.

3.3. Fluid Dynamics

The model of the fluid dynamics uses Ω as an input and produces the fluid velocity vector field $\boldsymbol{u}_{\text{ss}}$ at any cell of the tetrahedral mesh as an output. The model is defined by the Navier-Stokes equations for incompressible fluids [72] applied as a laminar steady state system (i.e., removing the time variability and without introducing extra turbulence terms). These assumptions are valid (and supported by the simulations) given the expected inlet and outlet flows, the slow speed of the water and the large scale of Ω.

The boundary conditions enforce no slip on Γ_{bed}; free slip but no traversal on Γ_{surf}; and a fixed laminar volume flow in inlets and outlets, the first adding a volume of water to Ω and the second subtracting it.

This model is solved using the PARDISO linear solver of COMSOL, embedded on a relaxed double-dogleg non-linear solver with parameter easing applied to the inlet and outlet flows in ten steps of increasing size in order to achieve convergence. Further details on how to set the model and solver can be found in COMSOL's documentation [135].

3.4. Water Pressure

The slow velocity of the fluid in our lentic water bodies lets us start approximating its pressure field p_{ss} by the hydrostatic pressure p_{hs} at any depth z, which is easy to compute with the equation for constant fluid density

$$p_{\text{hs}}(z) = -\rho_{\text{F,atm}}|\boldsymbol{g}|z + p_0, \tag{1}$$

where $\rho_{\text{F,atm}}$ is the fluid density at atmospheric pressure, $|\boldsymbol{g}|$ the module of the gravity acceleration, and p_0 the atmospheric pressure.

Note that for higher water velocities, p_{ss} should be computed at any point of Ω by interpolating the pressure field that is internally calculated by the hydrodynamic model for all the cells of the tetrahedral mesh used to discretize Ω. Our approach is simpler and computationally lighter.

3.5. Thermocline

In a stratified domain, the thermocline is defined as the water layer in which the greatest temperature variation is located, going from the warmest near the surface to the coldest near the bed. This thermal variations of the water domain as depth increases can be modeled with the sigmoid function

$$T(z) = T_0 + \frac{T_\mathrm{f} - T_0}{1 + \exp\left(\frac{\psi_T}{|z_{T,\mathrm{f}} - z_{T,0}|}\left(z - \frac{z_{T,0}+z_{T,\mathrm{f}}}{2}\right)\right)}, \tag{2}$$

where $T(z)$ is the temperature of water at depth z; T_0 and T_f their hypothetical respective temperatures at $z = \infty$ and $z = -\infty$; ψ_T the tension parameter for tuning the temperature gradient; and $z_{T,0}$ and $z_{T,\mathrm{f}}$ the start and final depth of the thermocline. Note that, although not exactly, T_0 and T_f approximate the water temperature just over and just below the thermocline.

3.6. Pycnocline

The pycnocline is the layer of water where the density of the fluid changes most rapidly as depth increases. It usually coincides roughly with the thermocline because the latter partly causes the former, and this happens specially in fresh water where salinity does not influence water density. As mentioned before, the density variation of the fluid plays an important role in regulating the vertical migration of the cyanobacteria colonies, which move by changing their cell density. The water density $\rho_\mathrm{F}(z)$ at depth z can be computed according to regressions based on experimental data [85], and to the values of $p_\mathrm{hs}(z)$ and $T(z)$ respectively obtained with Equation (1) and Equation (2), using

$$\rho_\mathrm{F}(z) = \rho_\mathrm{F,atm}(T(z)) \cdot (1 + \kappa_T(T(z)) \cdot (p_\mathrm{hs}(z) - p_0)), \tag{3}$$

where $\rho_\mathrm{F,atm}(\cdot)$ and $\kappa_T(\cdot)$ can be computed as

$$\begin{aligned}\rho_\mathrm{F,atm}(T) = 999.84847 + 6.337563 \cdot 10^{-2} T - 8.523829 \cdot 10^{-3} T^2 + \\ 6.943248 \cdot 10^{-5} T^3 - 3.821216 \cdot 10^{-7} T^4,\end{aligned} \tag{4}$$

$$\begin{aligned}\kappa_T(T) = 50.83101 \cdot 10^{-8} - 3.68293 \cdot 10^{-9} T + 7.263725 \cdot 10^{-11} T^2 - \\ 6.597702 \cdot 10^{-13} T^3 + 2.87767 \cdot 10^{-15} T^4.\end{aligned} \tag{5}$$

3.7. Surface Irradiance

We model the irradiance at the surface of the domain $I_\mathrm{surf}(t)$ as a truncated sinusoidal, which simulates the day/night cycle according to the time t of the day, providing an expected amount of light that ranges from 0 to the maximal irradiance I_max. In particular, we use the following function

$$I_\mathrm{surf}(t) = \max\left\{0,\ (I_\mathrm{max} - I_\mathrm{off}) \cdot \sin\left(\frac{2\pi(t - \Delta t_\mathrm{noon})}{\Delta t_\mathrm{day}} + \frac{\pi}{2}\right) + I_\mathrm{off}\right\}, \tag{6}$$

where I_off is the mean of the un-truncated sine wave and controls the duration of the irradiated period, Δt_day is the duration of a day, and Δt_noon is the time difference between t_0 and the moment when the sun is highest in the sky.

This mathematical formulation inherently supports seasonal solar geometry, as its parameters, such as the duration of the irradiated period, the maximum irradiance, and the time of noon, can be adjusted to reflect different times of the year.

Note that when weather data are available, other models will be better.

3.8. Light Transmission

The light transmission model is responsible for describing the quantity of the surface irradiance $I_{\text{surf}}(t)$ that is transmitted down to a certain depth z. To this end, we use the Beer-Lambert law [70] for our context

$$I(z,t) = I_{\text{surf}}(t)\mathrm{e}^{K_I \cdot z}, \tag{7}$$

where K_I is the attenuation coefficient.

3.9. Cyanobacterial Cell Density

The density model of the cells that form each colony of cyanobacteria within the bloom is critical for modeling their vertical migration. It takes as input the irradiance at the depth of the particle k that is associated to each colony and computes the timely variation of the density ρ^k_{cell} of its cells. The model distinguish between the light and dark regime, dependent on whether the quantity of incident light $I^k = I(z^k(t), t)$ surpasses or not the irradiance threshold I_{thres}. Mathematically, the model is described as

$$\frac{\partial}{\partial t}\rho^k_{\text{cell}} = \begin{cases} \beta I^k \mathrm{e}^{-I^k / I_{\text{lim}}} + \gamma, & I^k \geq I_{\text{thres}}, \\ -\alpha\left(\rho^k_{\text{cell}} - \rho_{\text{cell,min}}\right)\mathrm{H}\left(\rho^k_{\text{cell}} - \rho_{\text{cell,min}}\right), & I^k < I_{\text{thres}}, \end{cases} \tag{8}$$

where β is a normative factor, I_{lim} the photoinhibition limit, γ the rate of change in cell density in the dark, α the decay rate of cell density in dark conditions, $\rho_{\text{cell,min}}$ the minimum possible cell density, and $\mathrm{H}(\cdot)$ the Heaviside step function. In order to enforce the continuity of the model between both regimes, we compute β and γ as

$$\beta = \frac{\mathrm{d}\rho_{\text{max}}}{I_{\text{lim}}\mathrm{e}^{-1} - I_{\text{thres}}\mathrm{e}^{-I_{\text{thres}}/I_{\text{lim}}}}, \tag{9}$$

$$\gamma = -\beta I_{\text{thres}}\mathrm{e}^{-I_{\text{thres}}/I_{\text{lim}}}, \tag{10}$$

where $\mathrm{d}\rho_{\text{max}}$ is the maximum rate of cell density change.

The low values of the derivatives of this model lets us integrate it with the Explicit Euler time-stepping method.

3.10. Cyanobacterial Colony Density

A cyanobacterial colony is composed by cells holding small gaseous vacuoles and by a gelatinous substance called mucilage. According to [90], all these parts have to be considered when computing the density ρ^k_{p} of the particle corresponding to each colony of the bloom, using

$$\rho^k_{\text{p}} = \rho^k_{\text{cell}} n_{\text{cell}}(1 - n_{\text{gas}}) + \rho_{\text{muc}}(1 - n_{\text{cell}}), \tag{11}$$

where n_{cell} is the volume percentage of cells within the colony, n_{gas} is the volume percentage of gas within the cells, and ρ_{muc} is the mucilage density.

3.11. Concentration of Cyanobacterial Colonies

Since the simulated particles are placed at specific locations $\boldsymbol{l}_{\mathrm{p}}^{k} = [l_{\mathrm{p},x}^{k}, l_{\mathrm{p},y}^{k}, l_{\mathrm{p},z}^{k}]$, the continuous probabilistic concentration of particles $c_{\mathrm{p}}(\boldsymbol{l}, t)$ at any point $\boldsymbol{l} = [l_x, l_y, l_z]$ in Ω and time $t \in \mathcal{T}$ is computed as a sum of Gaussian bells centered at the position of each particle, with

$$c_{\mathrm{p}}(\boldsymbol{l}, t) = \sum_{k \in \mathcal{K}(t)} A_{\mathrm{c}} \exp\left(\sum_{d \in \{x,y,z\}} -\frac{\left(l_d - l_{\mathrm{p},d}^{k}\right)^2}{2\sigma_{c,d}^2} \right), \tag{12}$$

where $\mathcal{K}(t)$ is the list of indices with colonies alive at t, and A_{c} is given by

$$A_{\mathrm{c}} = \frac{1}{(2\pi)^{3/2} \prod_{d \in \{x,y,z\}} \sigma_{c,d}} \tag{13}$$

to ensure that given a spread σ_{c}, the volumetric integral of the contribution of each particle to $c_{\mathrm{p}}(\boldsymbol{l}, t)$ is equal to 1, to represent the probability of finding each particle at a given point in space.

3.12. Particle Transport

This model integrates information coming from other models to determine how the particles (and therefore the colonies) are moving. In particular, it uses the water velocity obtained with the fluid dynamics, the density of the water and the density of the particles to compute the particle velocity and position. More specifically, the velocity $\boldsymbol{u}_{\mathrm{p}}^{k}$ of particle k, is given by

$$\left(1 + \frac{1}{2}\frac{\rho_{\mathrm{F}}^{k}}{\rho_{\mathrm{p}}^{k}}\right)\frac{\mathrm{d}\boldsymbol{u}_{\mathrm{p}}^{k}}{\mathrm{d}t} = \left(1 - \frac{\rho_{\mathrm{F}}^{k}}{\rho_{\mathrm{p}}^{k}}\right)\boldsymbol{g} + \frac{18\nu\rho_{\mathrm{F}}^{k}}{d_{\mathrm{p}}^{2}\rho_{\mathrm{p}}^{k}}\left(\boldsymbol{u}_{\mathrm{ss}}^{k} - \boldsymbol{u}_{\mathrm{p}}^{k}\right) + \frac{3}{2}\frac{\rho_{\mathrm{F}}^{k}}{\rho_{\mathrm{p}}^{k}}\frac{\mathrm{D}\boldsymbol{u}_{\mathrm{ss}}^{k}}{\mathrm{D}t} + D_{\mathrm{p}}\boldsymbol{r}_{D}^{k}, \tag{14}$$

where ρ_{F}^{k} is the density of the fluid at the depth of particle k, d_{p} the diameter of the particles, $\boldsymbol{u}_{\mathrm{ss}}^{k}$ the water velocity at the position of particle k (obtained by interpolating the water velocity $\boldsymbol{u}_{\mathrm{ss}}$ computed by COMSOL for the cells of the discretization of Ω), D_{p} the diffusion coefficient, $\boldsymbol{r}_{D}^{k}$ a 3D random vector whose components follow the normal distribution, $\frac{\mathrm{d}}{\mathrm{d}t} = \frac{\partial}{\partial t} + \left(\boldsymbol{u}_{\mathrm{p}}^{k} \cdot \nabla\right)$ computes the time derivative along the trajectory of the particle, and $\frac{\mathrm{D}}{\mathrm{D}t} = \frac{\partial}{\partial t} + \left(\boldsymbol{u}_{ss}^{k} \cdot \nabla\right)$ computes the convective derivative along the trajectory of the fluid at the position of the particle. Note that, as in [136], the diffusion is added as a random walk in order to address the lack of a turbulence model in a sub-discretization level.

The previous model obtains the acceleration $\frac{\partial \boldsymbol{u}_{p}^{k}}{\partial t}$ of each particle. Integrating these values twice using the Explicit Euler time-stepping method, we obtain the velocities and positions of the cyanobacteria colonies.

Finally, note that on the boundaries of Ω, the colonies locations are driven by special behaviors. In particular, on Γ_{surf} the particles are allowed to slide, on Γ_{bed} they get stuck but are allowed to float back up freely without sliding, and on the outlets $\Gamma_{\mathrm{out}}^{j}$ they are destroyed as they exit the domain.

3.13. Biological Growth

The biological growth is the rate at which the colonies grow by multiplying their cells to eventually form new colonies or by dying. We combine both phenomena, using data fitting for reproduction [88] and adding a linear relationship that has not been previously

contemplated in [87,88] to account for the shrink of the colonies. In particular, the total growth rate μ^k of colony k, dependent on the irradiance I^k reaching it, is given by

$$\mu^k = \frac{2\mu_{\max}(1+\eta)\left(\frac{I^k}{I_{\lim}}\right)}{\left(\frac{I^k}{I_{\lim}}\right)^2 + 2\eta\left(\frac{I^k}{I_{\lim}}\right) + 1} - \xi I^k - \mu_0, \tag{15}$$

where $\mu_{\max}$ is the reproduction rate under ideal conditions, η the attenuation coefficient for photo-inhibition in reproduction, ξ the correlation coefficient between irradiance and death rate, and μ_0 the basal metabolic growth rate.

As the colonies are represented as particles, their biological growth or shrink must be represented by adding or removing an integer amount of particles. To implement it, we start computing the integer number of colonies N_μ^k to add or remove per each particle and time step with

$$N_\mu^k = \text{trunc}\left(\mu^k \Delta t\right) + \text{sign}\left(\mu^k \Delta t\right) \text{J}\left(r_\mu^k < \text{frac}\left(\mu^k \Delta t\right)\right), \tag{16}$$

where r_μ^k is sampled from the uniform distribution $\mathcal{U}([0,1])$, $\text{J}(\cdot) = 1$ if the input inequality is true and $\text{J}(\cdot) = 0$ otherwise, $\text{sign}(\cdot)$ is the sign function that returns -1 if the input is less than 0 and 1 otherwise, $\text{trunc}(\cdot)$ is the truncation function that removes the decimal part of the input, and $\text{frac}(\cdot) = |\cdot| - \text{trunc}(|\cdot|)$ the fractional function that returns the decimal part of the input.

Afterwards, if $N_\mu^k > 0$, new N_μ^k particles are added around particle k, keeping the same velocity and density as particle k, and each of them with a new position sampled from a Gaussian distribution centered at $\boldsymbol{l}^k$ with standard deviation σ_μ. If any of the sampled locations is outside Ω, it is value is re-sampled until a valid location is found. If $N_\mu^k < 0$, the particle is removed as well as its closest $N_\mu^k - 1$ particles. If $N_\mu^k = 0$, no action is taken.

4. Physical Simulations

In order to simulate the dynamics of cyanobacteria in a particular water body, we have to define the domains of the simulation (Ω and $\mathcal{T}$), and set up the initial conditions and parameters of all the models presented in Section 3. Afterwards, we completely compute the fluid dynamics with COMSOL version 5.3 to obtain the velocity of the water in the defined domains. Finally, we iterate in MATLAB R2023b the rest of the models for each time-step, until reaching the final time in $\mathcal{T}$. In this section, we show intermediate and final results of this process, setting up six different scenarios over two different water bodies.

4.1. Common Model Parameters

The six simulation scenarios presented in this work, not tailored to fit specific events, are set up to pose challenges to the AI-based planner that we present in Section 5. For simplicity, they share the values of the majority of the physical parameters, which are listed in Table 2, organized according to the first model where they appear. The parameters written in black are commonly known constants, those in blue are rough estimations provided by experts, while those in magenta (the majority related to the vertical migration) have been optimized to fit observations as described in Appendix A. In this regard, it is important to highlight that many of these parameters are unique to specific species and, in some cases, they cannot be measured and will also have to be determined by fitting the models to experimental data.

Table 2. Common parameters of the models for the simulations of this work .

Main Model	Variables and Values
Water pressure	$\boldsymbol{g} = (0, 0, -9.81)^\top \,\mathrm{m\,s^{-2}}$, $p_0 = 101.325\,\mathrm{kPa}$
Thermocline	$T_0 = 25\,\mathrm{C}$, $T_\mathrm{f} = 22\,\mathrm{C}$, $\psi_T = 10$, $z_{T,0} = -2\,\mathrm{m}$, $z_{T,\mathrm{f}} = -5\,\mathrm{m}$.
Surface Irradiance	$I_\mathrm{max} = 800\,\mathrm{W\,m^{-2}}$, $I_\mathrm{off} = 200\,\mathrm{W\,m^{-2}}$, $\Delta t_\mathrm{day} = 24\,\mathrm{h}$, $\Delta t_\mathrm{noon} = 6\,\mathrm{h}$.
Light Transmission	$K_I = 0.5\,\mathrm{m^{-1}}$.
Cell Density	$I_\mathrm{lim} = 338\,\mathrm{W\,m^{-2}}$, $I_\mathrm{thres} = 25.5\,\mathrm{W\,m^{-2}}$, $\alpha = 8.96 \times 10^{-6}\,\mathrm{s^{-1}}$, $\rho_\mathrm{cell,min} = 1010\,\mathrm{kg\,m^{-3}}$, $\mathrm{d}\rho_\mathrm{max} = 8.32 \times 10^{-4}\,\mathrm{kg\,m^{-3}\,s^{-1}}$
Colony Density	$n_\mathrm{cell} = 10\,\%$, $n_\mathrm{gas} = 5\,\%$, $\rho_\mathrm{muc} = 998\,\mathrm{kg\,m^{-3}}$.
Cyanobacteria Concentration	$\boldsymbol{\sigma}_\mathrm{c} = (25, 25, 0.5)^\top\,\mathrm{m}$.
Particle Transport	$\nu = 1 \times 10^{-6}\,\mathrm{m^2\,s^{-1}}$, $d_\mathrm{p} = 2.96 \times 10^{-3}\,\mathrm{m}$, $D_\mathrm{p} = 0.2\,\mathrm{m\,s^{-2}}$.
Biological Growth	$\mu_\mathrm{max} = 1.15\,\mathrm{day^{-1}}$, $\eta = 0.78$, $\xi = -10^{-3}\,\mathrm{m^2\,J^{-1}}$, $\mu_0 = -0.2\,\mathrm{day^{-1}}$, $\boldsymbol{\sigma}_\mu = (2, 2, 0.2)^\top\,\mathrm{m}$.

4.2. Invariant Simulations of the Physical Models

In this section we show the solutions of those model that do not depend on the initial conditions of each scenario. Their results, analyzed below, are summarized in Figure 2.

4.2.1. Water Pressure, Thermocline and Pycnocline

The thermocline and the pycnocline, respectively simulated with the models in Sections 3.5 and 3.6, are shown in Figure 2a. The blue line shows the thermocline as a graphic where the temperature T (in the bottom horizontal axis) depends on the depth z (in the vertical axis). The black dashed horizontal lines show the start and end of the thermocline, $z_{T,0}$ and $z_{T,\mathrm{f}}$. The red line shows the pycnocline as a graphic where the fluid density ρ_F (in the top horizontal axis) also depends on the depth z. Since both models are coupled, the change in temperature as the water depth increases also produces a sudden change in the fluid density. In addition, while the water temperature remains constant below the pycnocline, the water density ρ_F keeps on increasing, because it also depends on the hydrostatic pressure, which increases with the depth according to the model in Section 3.4. Although the density difference under 5 m is not huge, it is enough to affect organisms whose density changes are also small.

4.2.2. Surface Irradiance and Light Transmission

Figure 2b summarizes the irradiance changes throughout one day. The plot at the top represents the surface irradiance $I_\mathrm{surf}(z)$ (at the vertical axis) against the time of the day (at the horizontal axis). The plot at the right shows the maximum irradiance (at the horizontal axis) reaching each depth (at the vertical axis) at any given time of the day. The color plot in the center represents, according to color scale at the bottom, the irradiance $I(z, t)$ reaching each depth (in the vertical axis) at any time of the day (in the horizontal axis). It is easy to observe how the irradiance decreases with the depth and increases at the middle hours of the day. In addition, the dashed red line within the color plot represents $I_\mathrm{thres} = 25.5\,\mathrm{Wm^{-2}}$, which the cell density model in Section 3.9 uses to distinguish when the cyanobacteria cells transition from the light to the dark regime.

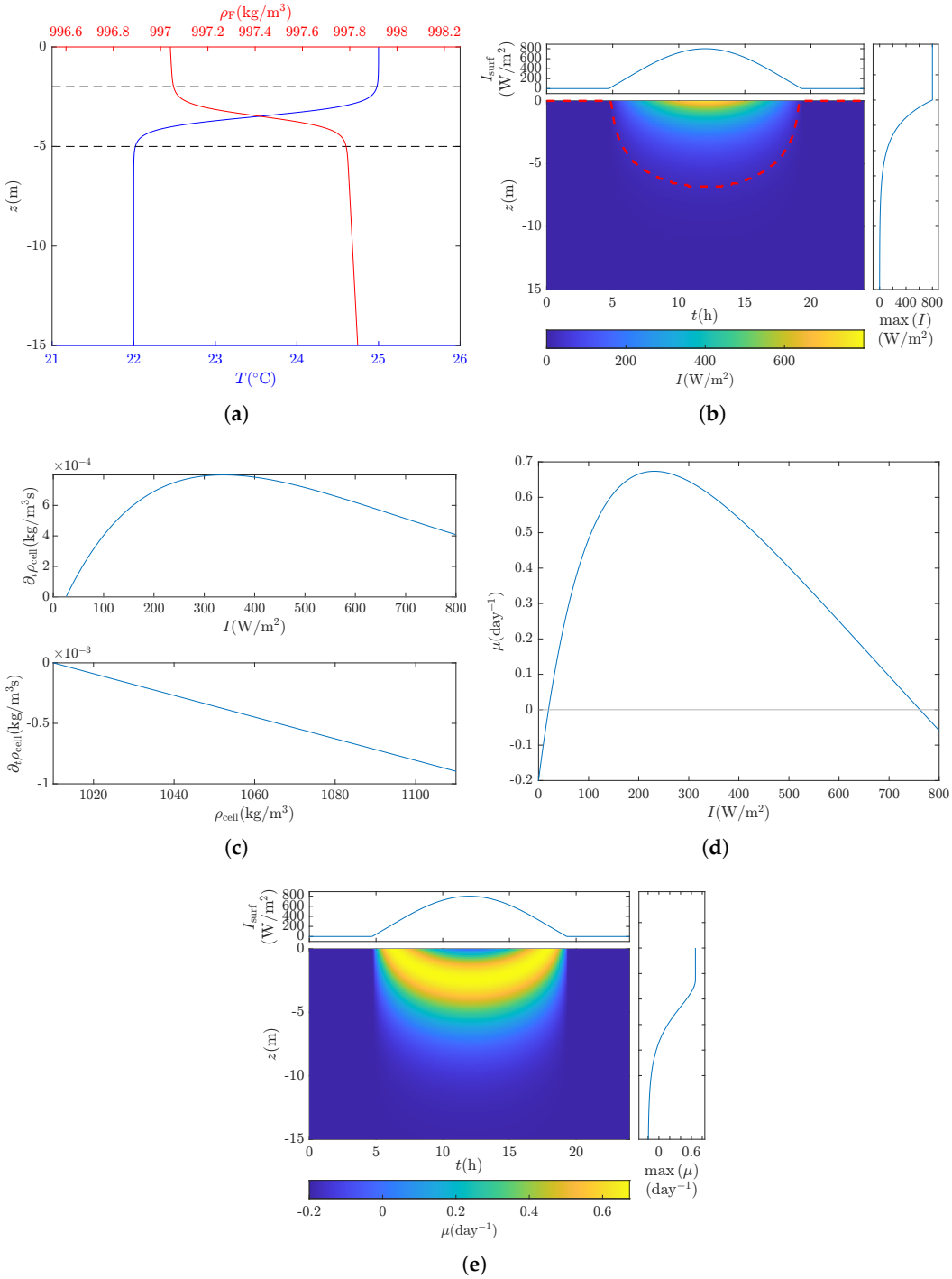


Figure 2. Invariant simulations of several submodels. (**a**) Thermocline and pycnocline. (**b**) Irradiance over time and depth. (**c**) Cell density changes in light and dark conditions. (**d**) Growth rate. (**e**) Growth rate over time and depth.

4.2.3. Cell Density

Figure 2c shows the rate of cell density change. On one hand, the top plot, corresponding to the light regime, represents the rate of cell density change (in the vertical axis) against the irradiance $I(z,t)$ reaching the cells. Although the cell density always increases in light conditions (because the rate is positive), the greatest increase happens at $I_{\text{lim}} = 338\,\text{Wm}^{-2}$. Beyond this point, the photo-inhibition of the cyanobacteria starts, slowing down the rate of cell density increments. On the other hand, the bottom plot, corresponding to the dark regime, represents the rate of cell density change (in the vertical axis) against the cell density

ρ_{cell}. At the minimal cell density $\rho_{\text{cell,min}} = 1010\,\text{kgm}^{-3}$ there are no cell density changes. The rate of cell density loss linearly increases with the number of cells, up to reaching the cells with the greatest density where the greatest density loses occur.

4.2.4. Biological Growth

Figure 2d shows the rate of growth of cyanobacteria μ (in the vertical axis) against the irradiance $I(z,t)$ at the location of each colony, according to the model of Section 3.13. The basal metabolic growth rate $\mu_0 = -0.2\,\text{day}^{-1}$ occurs when no illumination reaches the colony. As the irradiance increases, the combination of reproduction and death rates makes the growth rate positive until reaching a maximum. Beyond that point, the photo-inhibition takes place, reducing the growth rate until becoming negative.

Figure 2e shows the rate of growth of cyanobacteria μ at different depths along the day. The plot at the top represents again the surface irradiance $I_{\text{surf}}(z)$ (at the vertical axis) against the time of the day (at the horizontal axis). The plot at the right shows the maximum growth rate (at the horizontal axis) reaching each depth (at the vertical axis) at any given time of the day. The color plot in the center represents, according to color scale at the bottom, the rate of growth of cyanobacteria reaching each depth (in the vertical axis) at any time of the day (in the horizontal axis). We can observe how the evolution of the irradiance throughout the day makes the ideal depth change, even forcing cyanobacteria to die if they stay near the surface during the brightest hours of the day.

4.3. Simulations of the Complete Model

This section presents the simulation of the complete model as a whole, which depends on the relationships among all the submodels, on their parameters and on their initial conditions. After defining the domains of the simulation, we will simulate the fluid dynamics of two water bodies and the distribution of cyanobacteria in six scenarios (name SC1-SC6 hereafter).

4.3.1. Domain Definition

The time domain $\mathcal{T}$ is set up using the same start time t_0 and time step Δt the for all the scenarios, and a different end time t_f for each of them. Their values are presented in Table 3.

Table 3. Time domain $\mathcal{T}$ for the different scenarios.

Scenario	SC1	SC2	SC3	SC4	SC5	SC6
$t_0(\text{h})$			0			
$t_f(\text{h})$	68	57	9	50	60	30
$\Delta t(\text{h})$			1.667×10^{-2}			

Regarding the space domain Ω, the first water body (named WB1 hereafter and used in scenarios SC1, SC2 and SC3) is inspired in Lake Mendota (Madison, WI, USA). We have removed its shallower portions and scale it down to showcase better the capabilities of our trajectory planner (for a small and slow ASV). Its dimensional properties are summarized in Table 4 and its geometry is shown in Figure 3a, using the horizontal and vertical axes for the x and y coordinates, and the color scale at the right for the z coordinate of its water bed. Thus, red is associated to the shallower areas and blue to the deepest. Its non-convex shape is well-connected and easily accessible, from the point of view of the planner, for the ASV.

Table 4. Properties of the space domain Ω for the different scenarios.

Water Body	Scenarios	Volume (m^3)	Area of Γ_{surf} (m^2)	Mean z at Γ_{bed} (m)	Minimal z at Γ_{bed} (m)
WB1	SC1, SC2 & SC3	6.5×10^7	1.43×10^6	−44.58	−70.82
WB2	SC4, SC5 & SC6	2.77×10^6	3.63×10^5	−7.61	−19.77

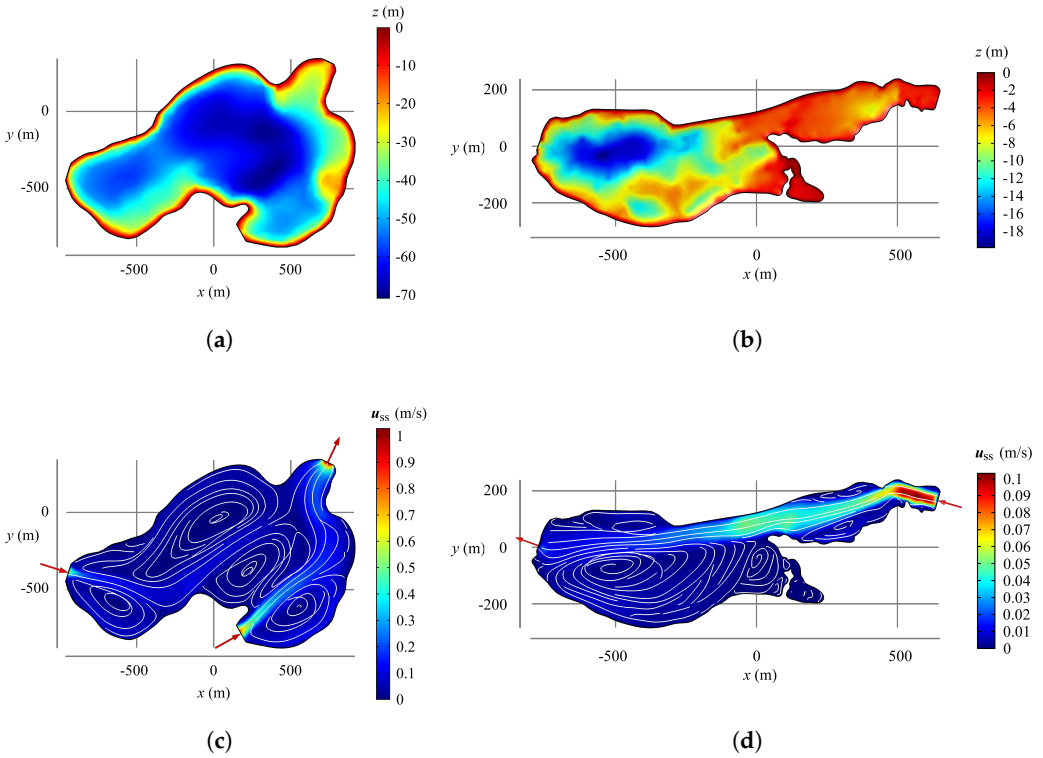


Figure 3. Bathymetry and fluid dynamics simulation of both water bodies. (**a**) Bathymetry of WB1. (**b**) Bathymetry of WB2. (**c**) Fluid dynamics of WB1. (**d**) Fluid dynamics of WB2.

The second water body (named WB2 and used in scenarios SC4, SC5 and SC6) is El Campillo Lagoon (Rivas-Vaciamadrid, Madrid, Spain). Its dimensional properties are summarized in Table 4 and its geometry is represented in Figure 3b. It consists of three distinct areas: the middle to the west main body of greatest depth which is safely navigable by the ASV, a shallower narrow long unsafer area in the north-east, and a really shallow area in the south-east which can also harbor the growth of cyanobacterial blooms.

4.3.2. Fluid Dynamics

Figure 3c shows the solution of the fluid dynamics for WB1, using the horizontal and vertical axes to represent the x and y coordinates; the color scale to show the magnitude of the water velocity in steady state $\boldsymbol{u}_{SS}$ at the water surface Γ_{surf}; the inwards and outward red arrows to mark the inlets (with a mass flow of 20.99 m^3/s and 79 m^3/s) and outlet (with a mass flow of 100 m^3/s); and the white flow lines to indicate the projection onto the horizontal plane of the water path. WB1 geometry connects the inlets and outlets by two fast-moving currents, with maximal speed near the inlets and outlets. Several vortices (i.e., masses of water moving in cyclic paths) of reduced fluid speed are also developed, becoming regions capable of concentrating great amounts of cyanobacteria dragged towards them by the currents.

Figure 3d shows the fluid dynamics solution for WB2. Again, a main current connects the inlet on the east and the outlet on the west, both with a mass flow of $10\,m^3/s$. Vortices are also created between the current and the shores. Since the inlet is placed in a narrow area, its speed is maintained (and slightly increased) in the strait region that connects the inlet to the rest of WB2. There is a vortex at the south of the main current that covers most of the water body, and smaller and less stable vortices towards the north of the current, within the inlet area and at the east part of the main lagoon area. Finally, in the south-east peninsula, the water is almost still.

4.3.3. Cyanobacteria Evolution

This section presents the setup of the six scenarios under study and discusses their cyanobacteria evolution, represented in two distinct kinds of plots, placed at the top and bottom rows of graphics of Figures 4 and 5.

To set up the scenarios, we initially sample 1000 particles, according to the Gaussian distributions of Table 5, where $\mathcal{N}(\boldsymbol{a}, \boldsymbol{b})$ stands for a Gaussian distribution of mean $\boldsymbol{a}$ and variance $\boldsymbol{b}$. The initial velocity of the particles is equal to the speed of the fluid at the initial location of the particle, and the initial cell densities are computed to make the colony densities equal to $997\,kg\,m^{-3}$ (a value that provides a reasonable floatability to the particle given their rough location and vertical migration cycle point).

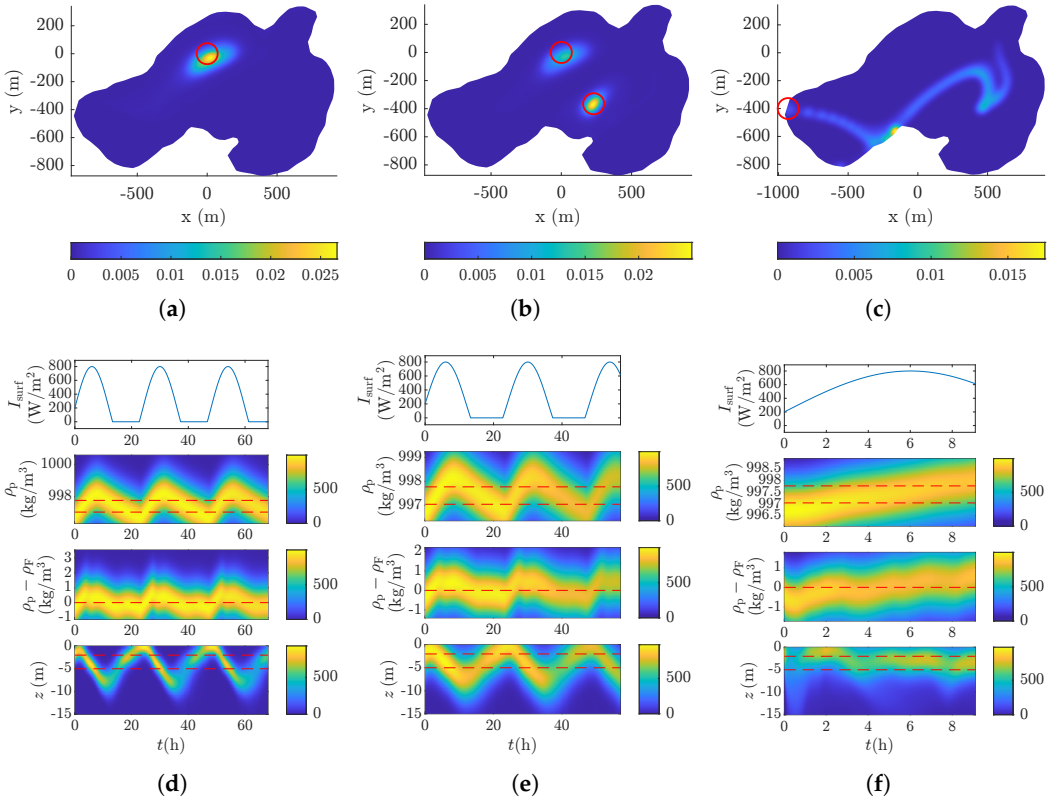


Figure 4. Cyanobacterial evolution in the scenarios of WB1. (**a**) Horizontal transport in SC1. (**b**) Horizontal transport in SC2. (**c**) Horizontal transport in SC3. (**d**) Vertical migration in SC1. (**e**) Vertical migration in SC2. (**f**) Vertical migration in SC3.

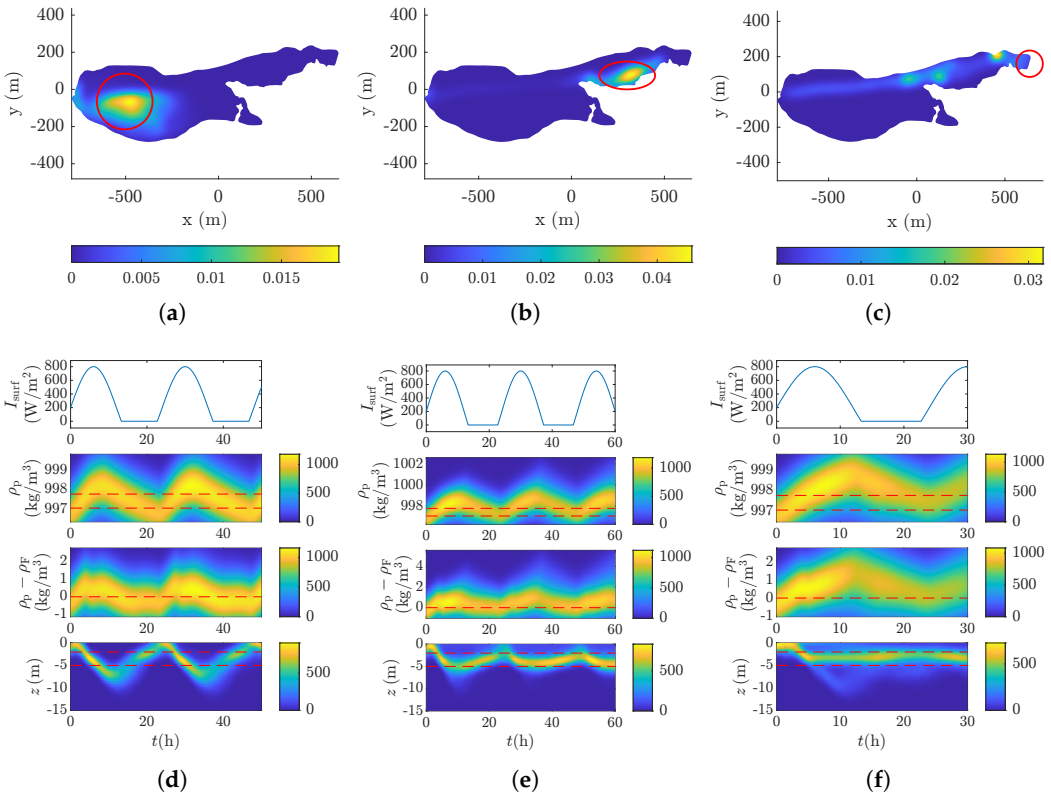


Figure 5. Cyanobacterial evolution in the scenarios of WB2. (**a**) Horizontal transport in SC4. (**b**) Horizontal transport in SC5. (**c**) Horizontal transport in SC6. (**d**) Vertical migration in SC4. (**e**) Vertical migration in SC5. (**f**) Vertical migration in SC6.

Table 5. Initial distributions of particles.

Scenario	Distribution
SC1	$P_{SC1}(\boldsymbol{l}_0^k) = \mathcal{N}([0, 0, -2], \text{diag}([25^2, 25^2, 0.5^2]))$
SC2	$P_{SC2}(\boldsymbol{l}_0^k) = 0.5 \cdot P_{SC1}(\boldsymbol{l}_0^k) + 0.5 \cdot \mathcal{N}([230, 365, -2], \text{diag}([25^2, 25^2, 1^2]))$
SC3	$P_{SC3}(\boldsymbol{l}_0^k) = \mathcal{N}([-930, -400, -1], \text{diag}([25^2, 25^2, 0.5^2]))$
SC4	$P_{SC4}(\boldsymbol{l}_0^k) = \mathcal{N}([-505, -65, -1], \text{diag}([50^2, 50^2, 1^2]))$
SC5	$P_{SC5}(\boldsymbol{l}_0^k) = \mathcal{N}([300, 75, -1], \text{diag}([50^2, 25^2, 1^2]))$
SC6	$P_{SC6}(\boldsymbol{l}_0^k) = \mathcal{N}([640, 160, -1], \text{diag}([25^2, 25^2, 1^2]))$

The first type of plots, corresponding to Figures 4a–c and 5a–c, focuses on the horizontal transport of cyanobacteria, and uses the horizontal and vertical axes for the x and y coordinates, and the color scale at their bottom for the time-averaged particle concentration field, integrated over the vertical axis. Hence, darker regions of each water body will have, on average, lower concentration of cyanobacteria than brighter regions, where cyanobacteria accumulate for a significant amount of time. Finally, the red ellipses are centered at the mean and span 6 times the standard deviations of the Gaussian distributions used to initially sample the particles from.

The second kind of plots, corresponding to Figures 4d–f and 5d–f, focuses on the vertical migration of cyanobacteria and consists of four plots. The first one represents the irradiance reaching the surface (in the vertical axis) against the simulation time (in the horizontal axis). We include it to highlight the correlation between the cyanobacteria vertical migration and the irradiance. The second plot shows the concentration of cyanobacteria (using the color scale in its right) against the colony density ρ_p (in the vertical axis) and time (in the horizontal axis); and highlights the fluid density ρ_F at the start and end of the

pycnocline (with two red dashed lines). This plot provides an idea of the density variations in the colonies, but it does not indicate their floatability, since the fluid density changes with the different depths where the cyanobacteria are placed. Thus, the third plot shows the cyanobacteria concentration against the difference between the particles and the fluid densities (in the vertical axis) and the time (in the horizontal axis) around them, and the point where the particles are neutrally buoyant (using a dashed red line). Regions above this line correspond to sinking colonies, while regions below are associated to floating ones. Finally, the last plot shows the cyanobacteria concentration against the z coordinate and the time (in the vertical and horizontal axes), and the start and end depths of the pycnocline (with the red dashed lines). Hence, regions with brighter colors correspond to depths where the cyanobacteria are concentrating. Generally, the results of the simulations, shows that the changes in the density and floatability of the colonies, make the cyanobacteria vertical migration occur every 24 h, reaching the surface during early morning and sinking to reach the lowest point during the late afternoon, when they start their migration up, which lasts for the whole night. Note that this pattern is species dependent and can be modified using the parameters of the simulation.

Finally, we provide some insight related to each scenario.

Scenario SC1

Figure 4a shows that the particles are concentrated near the greatest vortex of the lake, where the fluid velocity is slowest. Figure 4d shows that the vertical migration is cohesive, with most particles having a similar depth at the same time.

Scenario SC2

Figure 4b shows that the particles are initially distributed in two vortices, where they remain concentrated. Figure 4e shows that the vertical migration is also well-defined, but that the particles are distributed more widely in the vertical direction due to the flow difference of different areas.

Scenario SC3

Figure 4c shows how the particles, initially sampled near one of the inlets, are dragged by the strong currents connecting the inlets and the outlet. Figure 4f shows an uncoordinated vertical migration due to the negligible accelerations produced by the density changes. In addition, the length of this simulation, limited by the time required for the particles to traverse WB1, does not permit a clear view of any long term pattern formation.

Scenario SC4

Figure 5a shows that particles starting in a slow moving area around a vortex surrounded by unstructured currents end up widely spread. Figure 5d shows that the vertical migration is cohesive, as has happened in other scenarios with particles located in still waters.

Scenario SC5

Figure 5b shows that particles, originally sampled around a small vortex located in the north-east, remain in the same area due to the vortex inward currents and to the geometric isolation of the area. Figure 5e shows a reduced vertical migration, due the shallow waters in the area.

Scenario SC6

Figure 5c shows that particles, originally sampled over the current that traverses the north-east area, are spread along a line covering the north of WB2, with zones of higher concentration when particles are attached to the water bed. Figure 5f shows a negligible vertical migration, since the accelerations due to currents are greater than those due to floatability.

5. Trajectory Planner

In order to validate the predictions of the model and raise early warnings about the different uses of water, our AI-based approach makes use of ASVs equipped with multi-parametric probes capable of determining the presence of cyanobacteria (e.g., measuring phycocyanin concentration [56]) and automatically driven according to the trajectories provided by a planner supported by evolutionary algorithms.

In particular, the planner, presented in this section, aims to optimize the trajectory of the ASV, taking into account the results of the simulation of a given scenario, multiple objectives (related not only with predicted cyanobacteria concentration but also to the mission time and length of the trajectory) and constraints (related to the water body and the ASV). To this end, we define a multi-objective constrained optimization problem, and solve it with the different variants of evolutionary algorithms supported by the planner. Besides, our planner optimizes a 3D continuous trajectory, working under the assumption that measurements will be taken along the path traced by a probe that has to adapt its depth to the cyanobacteria vertical migration and whose horizontal localization is determined by the ASV displacement.

It is worth noting the following two facts before detailing our planner. On one hand, tour planner is designed to run offline before the start of the monitoring mission. Real-time global trajectory optimization is computationally prohibitive and unnecessary for this application, given that cyanobacteria vertical migration follows a slow 24-hour cycle and the ASV navigates at low speeds. Hence, any immediate local obstacle avoidance required during the mission is left to the onboard navigation software of the vehicle. On the other hand, as the results of the 3D cyanobacteria evolution simulation are used by the planner to optimize the ASV trajectories, when the simulations are predicting a wrong distribution of the cyanobacteria, the ASV will be guided to wrong locations. If that is the case, the ASV can either complete its trajectory to collect data that can be useful to validate the model later or resume it, and wait for a new trajectory obtained by the planner from a new simulation. Alternatively, the ASV could also be used to gather data, from regions of interest, in order to restart the particles of the simulation from regions where the ASV has detected cyanobacteria. At this point, it is important to understand that the remainder of this paper considers that the simulations are correct and therefore useful to guide the ASV to the correct regions of the water body, and what-to-do when they are not, falls outside the scope of this paper.

In the following two sections we (1) describe the optimization problem (which was already presented in [32]) solved by our AI-based planner, and (2) detail its current supporting evolutionary optimization algorithms.

5.1. Optimization Problem

This section starts presenting the encoding of the trajectory used by our planner, and continues detailing the three objective and seven constrain functions that are optimized/enforced by its supporting evolutionary algorithm.

5.1.1. Encoding

Since the planner has to determine the best 3D ASV+probe *continuous smooth* trajectory, we have decided to encode it with a spline [137], a piece-wise polynomial whose design can meet different requirements, and whose shape, controlled by several nodes, can fit well the flow lines of water (and therefore, the distribution of cyanobacteria in the water body).

In particular, we have decided to encode it as a 3D cardinal spline [28] with a null tensioning parameter for the following reasons. First, this type of cubic Hermite spline defines a C^2 continuous curve for the position, velocity and acceleration of the ASV. Second, it is a local spline, which ensures that when the planner changes the value of a particular node, it only affects its surroundings (and does not modify the entire trajectory). Third, as the curve is interpolating, it makes the trajectory go through the nodes that define the spline and simplifies the computation of the domain constraint. Finally, it also allows setting velocity values for the start and end nodes (in our case, to stop the ASV at the start and end of the trajectory).

In particular, we represent the ASV+probe trajectory with $N_S + 1$ nodes, providing the values of its three spatial coordinates (x and y for the ASV surface location, and z for the probe depth) and of its time t (to allow the planner adjust the time mission and speed of the ASV). However, to let the user decide the start and end locations of the ASV, the planner fixes the location of the extreme nodes of the spline. The initial time is also excluded, as the planner considers, without loss of generality, that the trajectory always starts at $t = \tau_0$. Thus, the total number of continuous decision variables that the planner has to determine, for a 3D spline $\boldsymbol{S}(t)$ of N_S segments, is $4(N_S - 1) + 1$.

Finally, it is worth noting that for the evaluation of the objective and constraint functions, when required, we use the definition of the spline $\boldsymbol{S}(t)$ and not only its control nodes. Besides, the spline is discretized with a time step of 1 s to evaluate OF_3 and CF_2 to CF_6, and through the global adaptive quadrature from MATLAB's `integral` to evaluate OF_2 and OF_7.

5.1.2. Objective Functions

The planner determines the suitability of each trajectory according to the following three Objective Functions (OF), already used in [32] and related to the mission duration, the ASV trajectory length and the expected concentration of cyanobacteria to observe during the mission.

OF_1 Minimizing Mission Duration

It makes the resulting best trajectory temporally efficient, prolonging the lifetime of the ASV and allowing to run more missions in a given amount of time. Particularly, OF_1 is defined as

$$\min \mathrm{OF}_1 = \tau_\mathrm{f} - \tau_0, \tag{17}$$

where τ_f is the time coordinate of the last node of the spline $\boldsymbol{S}$. Note that, neither the start and end time of the trajectory, τ_0 and τ_f, have to equal the start and end time of the simulation, t_0 and t_f.

OF_2 Minimizing Trajectory Length

Its aim is to reduce the distance navigated by making the trajectory traverse the interest regions without unnecessary detours. In conjunction with the previous objective, it increases the safety and efficiency of the solution. Mathematically, OF_2 is formulated as

$$\min \mathrm{OF}_2 = L_S = \int_{\tau_0}^{\tau_\mathrm{f}} \left\| \frac{\partial \boldsymbol{S}(t)}{\partial t} \right\| \mathrm{d}t \tag{18}$$

OF_3 Maximizing Probe Observations in Areas of High Cyanobacterial Concentration

It focuses on maximizing, while the ASV+probe are moving, the observations in the regions of the water body with a high expected concentration of cyanobacteria. To this end, Equation (19) generalizes the concept of setting a detection radius around the probe and counting the number of particles within it at any point of the trajectory [25], by maximizing the maximal contributions of each particle to the cyanobacteria concentration along $\boldsymbol{S}(t)$. Hence, OF_3 incentivizes $\boldsymbol{S}(t)$ to pass through the areas with expected high concentration of cyanobacteria, by taking advantage of the fact that the exponential functions in $c_p(\boldsymbol{l}, t)$ contribute their maximal value when $\boldsymbol{S}(t)$ passes through the particle and smoothly decrease as it moves further away from it.

$$\max \mathrm{OF}_3 = \sum_{k \in \mathcal{K}(t)} \sup_{t \in [\tau_0, \tau_\mathrm{f}]} \left[A_\mathrm{c} \exp \left(\sum_{d \in \{x,y,z\}} - \frac{\left(S_d(t) - l^k_{\mathrm{p},d}(t) \right)^2}{2\sigma^2_{c,d}} \right) \right] \tag{19}$$

Finally, note that maximizing the integral or the time average of the concentration $c_\mathrm{p}(\boldsymbol{l}, t)$ along $\boldsymbol{S}(t)$ is not appropriate, as those options tend to favor paths that reach the highest concentration area as soon as possible and remain there during the rest of the mission.

5.1.3. Constraint Functions

The mission is bound by restrictions related to the physical domain and to the capabilities of the ASV and the probe. They are enforced by the seven Constraint Functions (CF) presented in this section, which are handled by the optimization algorithm as explained in the description of its Recombination step within Section 5.2.4.

CF_1. Limited Mission Duration

The mission duration is constrainable by the battery life or user-imposed decisions. Mathematically, we express it as

$$\mathrm{CF}_1 = \tau_\mathrm{f} - \tau_0 - \Delta\tau_\mathrm{max} \leq 0, \tag{20}$$

where $\Delta\tau_\mathrm{max}$ is the maximal mission duration.

CF_2. Limited ASV Velocity

The ASV speed is constrained by its maximal velocity $v_\mathrm{H,max}$. Since the water also moves, the constraint is applied to the ASV velocity (measurable in $\boldsymbol{S}_\mathrm{H}(t) \triangleq \left(S_x(t), S_y(t)\right)^\top$) with respect to the water speed at the ASV location ($\boldsymbol{u}_\mathrm{ss,H}(t) \triangleq \left(u_{\mathrm{ss},x}(\boldsymbol{w}_\mathcal{S}(t)), u_{\mathrm{ss},y}(\boldsymbol{w}_\mathcal{S}(t))\right)^\top$, where $\boldsymbol{w}_\mathcal{S}(t) \triangleq \left(S_x(t), S_y(t), 0\right)^\top$). CF_2 is mathematically expressed as

$$\mathrm{CF}_2 = \sup_{t \in [\tau_0, \tau_\mathrm{f}]} \left(\left\| \frac{\partial \boldsymbol{S}_\mathrm{H}(t)}{\partial t} - \boldsymbol{u}_\mathrm{ss,H}(t) \right\| \right) - v_\mathrm{H,max} \leq 0. \tag{21}$$

CF_3. Limited ASV Acceleration

The acceleration of the ASV, measured with respect to the water, is also constrained to a maximal value $a_\mathrm{H,max}$, using

$$\mathrm{CF}_3 = \sup_{t \in [\tau_0, \tau_\mathrm{f}]} \left(\left\| \frac{\partial^2 \boldsymbol{S}_\mathrm{H}(t)}{\partial t^2} - \frac{\mathrm{D}\boldsymbol{u}_\mathrm{ss,H}(t)}{\mathrm{D}t} \right\| \right) - a_\mathrm{H,max} \leq 0. \tag{22}$$

CF_4. Limited Curvature of the ASV Trajectory

Constraining the maximal curvature of $S_H(t)$ to $\kappa_{max,H}$ is necessary to ensure its physical viability according to the ASV minimal turning radius. Thus, CF_4 equation is

$$CF_4 = \sup_{t\in[\tau_0,\tau_f]} \left(\frac{\left| \frac{\partial S_x(t)}{\partial t} \frac{\partial^2 S_y(t)}{\partial t^2} - \frac{\partial S_y(t)}{\partial t} \frac{\partial^2 S_x(t)}{\partial t^2} \right|}{\left(\left(\frac{\partial S_x(t)}{\partial t} \right)^2 + \left(\frac{\partial S_y(t)}{\partial t} \right)^2 \right)^{3/2}} \right) - \kappa_{H,max} \leq 0. \tag{23}$$

CF_5. Limited Probe Velocity

The multi-parametric probe is displaced vertically by a mechanical pulley system with constrained speed $v_{V,max}$. For safety, CF_5 constrains the vertical speed of $\boldsymbol{S}(t)$ as

$$CF_5 = \sup_{t\in[\tau_0,\tau_f]} \left(\left| \frac{\partial S_z(t)}{\partial t} \right| \right) - v_{V,max} \leq 0. \tag{24}$$

CF_6. Limited Probe Acceleration

The vertical component of the acceleration of $\boldsymbol{S}(t)$ is also restricted to the maximal probe acceleration $a_{V,max}$ with

$$CF_6 = \sup_{t\in[\tau_0,\tau_f]} \left(\left| \frac{\partial^2 S_z(t)}{\partial t^2} \right| \right) - a_{V,max} \leq 0. \tag{25}$$

CF_7. Limited Spatial Domain

We also restrict the areas within Ω where the ASV can navigate safely, avoiding shores, shallow waters and forbidden areas. The navigable zones are defined by volume $F(\boldsymbol{S}) \leq 0$, where $F(\boldsymbol{S})$ is a signed distance field [138] or, for simplicity, a composition of discontinuous boolean functions. Finding the intersections between $\boldsymbol{S}(t)$ and $F(\boldsymbol{S}) \leq 0$ by solving $F(\boldsymbol{S}(t)) = 0$ for $t \in [\tau_0, \tau_f]$, allows determining if the former is located within the latter. Assuming that the initial and final nodes of the spline are always defined within the volume by the user, the number N_{inter} of intersections between $\boldsymbol{S}(t)$ and the navigable zone boundaries is even and defined by their time coordinates τ^b. After computing, with the parallelized bisection method proposed in [25], these multiple intersections, CF_7 accounts for the length of $\boldsymbol{S}(t)$ that are outside the allowed volume with

$$CF_7 = \sum_{b=1}^{N_{inter}/2} \int_{\tau^{2b-1}}^{\tau^{2b}} \left\| \frac{\partial \boldsymbol{S}(t)}{\partial t} \right\| dt \leq 0. \tag{26}$$

5.2. Optimization Algorithm

The complexity and irregularities of the constrained multi-objective optimization problem proposed in Section 5.1 are handled by two well proven and versatile evolutionary algorithms: the second and third versions of the Sorting Genetic Algorithms, NSGA-II [17] and NSGA-III [18,19]. The number of nodes of the spline is controlled by three approaches: maintaining them fixed, increasing them progressively, or letting the optimizer determine it by manipulating solutions of different lengths. In the following, we overview some properties of both NSGA versions; discuss the refinement mechanisms for controlling the number of nodes; describe the planner algorithmically; and detail the steps, procedures and operators that support them.

5.2.1. NSGA Overview

The different versions of NSGA [17–19,139] are multi-objective optimization iterative methods based on the principles of genetic algorithms. In short, each iteration $q \in \mathcal{Q}$ corresponds to a generation and improves its population of solutions which is evolved through survival of the fittest, crossover and mutation. They determine the fitness of the solutions in the population sorting them in different Pareto fronts [140] according to their corresponding objective and constraints values and to a two level domination mechanism, which is detailed in the description of the Recombination step of Section 5.2.4.

NSGA-II (sometimes shortened as A2) is an approach successfully and widely used in many real-world constrained multi-objective problems with numerous decision variables, including ours [32]. It benefits from an elitist perspective that introduces the crowding distance to sort the last partial fronts that will be included in the newest generation and that relies on a fast non-dominated sorting algorithm to form the Pareto fronts.

NSGA-III (A3) extends the capabilities of NSGA-II and of the previous version of our planner [32] by focusing on objective space covering. This is achieved replacing the crowding-distance mechanism in NSGA-II by the use of reference points, evenly distributed over the search front, in order to ensure the diversity of the solutions of the new generation. Its main advantage is observed in problems with a high dimensionality in the objective space.

5.2.2. Refinement for Controlling the Number of Nodes of the Spline

The number of nodes of the spline used to encode the AVS+probe trajectories affects the possible shapes they can handle. In turn, the 3D spline $\boldsymbol{S}(t)$ with $N_S + 1$ control points used by our planner requires $4(N_S - 1) + 1$ continuous decision variables. Hence, although incrementing N_S can facilitate the adaptation of $\boldsymbol{S}(t)$ to different water flows and cyanobacteria behaviors, its corresponding number of decision variables can slow down the convergence of the optimizer. To overcome this problem, our planner supports three different refinement mechanisms to control the number of decision variables used to encode $\boldsymbol{S}(t)$. Although the first two were already introduced in [32], in this paper the second is modified and the third is introduced.

The first type (named R1 hereafter) consists on maintaining N_S fixed. Hence, all the solutions of the population $\mathcal{P}$ have the same number of nodes (and decision variables) during the whole optimization (i.e., $N_S^0 = N_S^q = N_S^f$, where the super-index stands for the generation number). Being the easiest method to implement, it requires the user to preselect its value.

The second type (R2#) consists on starting the optimization process with a reduced number of nodes and incrementing it progressively. Thus, $N_S^0 = N_{S,\min} \leq N_S^q \leq N_{S,\max} = N_S^f$, where $N_{S,\min}$ and $N_{S,\max}$ are the minimal and maximal values that the user defines. The increment can be done every certain number of generations Q_{refine} as in [32] (named R2A) or when the refinement conditions, which are detailed in the Refinement paragraph within Section 5.2.4, are met (R2B). This new option is a relaxed convergence condition that introduces a new node in all the solutions of the population, when the algorithm is not improving them any longer.

The third type (R3) is based on variable-length chromosomes (VLC, refs. [33,34]) and lets the algorithm simultaneously optimize the number of nodes of the trajectory and their values. This is achieved by incorporating specialized crossover and mutation operators to the algorithm. The former mixes the information from different control points of the parents, while the latter adds or deletes control points randomly. Section 5.2.4 presents further details.

5.2.3. Algorithmic Description of the Planner

The implementation of our planner is summarized in Algorithm 1, which has two differentiated sections. In the first part, from line 1 to 8, it initializes parameters and the population, while in the second, from line 9 onward, it performs the optimization loop. The steps that are only applied in certain variants are indicated within a parenthesis at the end of the step. In essence, our planner follows the classical structure of an evolutionary algorithm: after initializing and evaluating the first population, it sequentially performs, within the optimization loop, the parent selection, crossover (with three possibilities, X1 for R1 and R2#, and X2 and X3 for R3), mutation, generation of immigrants, evaluation and recombination steps. The additional final step within the control loop is required to increment the number of nodes in the population for R2 variants. Note that, some input parameters to the functions are variant dependent and therefore only used when necessary.

Algorithm 1: Planner.

Input: Problem Parameters: $\boldsymbol{x}_{\mathrm{p}}^{k}$, A_{c}, $\boldsymbol{\sigma}_{c}$, $\boldsymbol{u}_{\mathrm{ss}}$, $\Delta\tau_{\max}$, $v_{\mathrm{H,max}}$, $a_{\mathrm{H,max}}$, $\kappa_{\mathrm{H,max}}$, $v_{\mathrm{V,max}}$, $a_{\mathrm{V,max}}$, $\mathrm{F}(\cdot)$
Input: Encoding parameters: N_S (R1), $N_{S,\min}$ (R2#, R3), $N_{S,\max}$ (R2#, R3)
Input: Mission parameters: τ_0, $\boldsymbol{x}_{S,0}$, $\boldsymbol{x}_{S,\mathrm{f}}$
Input: Planner parameters: A#, R#, X#, N_{gen}, N_{pop}, N_{child}, P_{xover}, $\boldsymbol{\sigma}_{\mathrm{mut}}$, P_{var} (R3), σ_{ref} (R2#, R3), $\varepsilon_{\mathrm{ref}}$ (R2C), $N_{\mathrm{gen,ref}}$ (R2C)
Output: *pop*, Population of solutions, sorted in Pareto fronts
Output: *val*, OF and CF values of *pop*

1 Generate HVE Samples (R2C)
2 Compute reference points (A3)
3 $X\# \leftarrow$ X1 (R1, R2#)
4 $N_{\mathrm{imm}} \leftarrow N_{\mathrm{pop}} - N_{\mathrm{child}}$
5 $N_S \leftarrow N_{S,\min}$ (R2#)
6 $q_{\mathrm{ref}} \leftarrow 0$ (R2C)
7 $pop \leftarrow \mathrm{PopGeneration}(N_{\mathrm{pop}}, \mathrm{F}(\cdot), \mathrm{R\#}, N_S, N_{S,\min}, N_{S,\max}, \mathrm{MissionParam})$
8 $val \leftarrow \mathrm{Evaluation}(pop, \mathrm{ProblemParam})$
9 **for** $q \in \{1, ..., N_{\mathrm{gen}}\}$ **do**
10 $\quad parents \leftarrow \mathrm{ParentSelection}(pop, val, N_{child}, \mathrm{A\#})$
11 $\quad children \leftarrow \mathrm{Crossover}(parents, , P_{\mathrm{xover}}, \mathrm{X\#})$
12 $\quad children \leftarrow \mathrm{Mutation}(children, \boldsymbol{\sigma}_{\mathrm{mut}}, \mathrm{R\#}, P_{\mathrm{var}}, \sigma_{\mathrm{ref}})$
13 $\quad imm \leftarrow \mathrm{PopGeneration}(N_{\mathrm{imm}}, \mathrm{F}(\cdot), \mathrm{R\#}, N_S^g, N_{S,\min}, N_{S,\max}, \mathrm{MissionParam})$
14 $\quad newpop \leftarrow \{children, imm\}$
15 $\quad newval \leftarrow \mathrm{Evaluation}(newpop, \mathrm{ProblemParameters})$
16 $\quad [pop, val] \leftarrow \mathrm{Recombination}(pop, newpop, val, newval, N_{\mathrm{pop}}, A\#, \mathrm{RefPoints})$
17 $\quad [pop, val, N_S] \leftarrow \mathrm{Refinement}(pop, val, N_S, q, R\#, \mathrm{HVESamples}, \sigma_{\mathrm{ref}}, \varepsilon_{\mathrm{ref}}, N_{\mathrm{gen,ref}})$ (R2#)
18 **end**

5.2.4. Steps and Operators of the Planner

This section details the relevant properties of the different steps and genetic operators of our planner.

Generation of Initial Population and Immigrants

New N_{pop} and N_{imm} trajectories are respectively created as members of the initial population *pop* and as immigrants *imm*, in lines 7 and 13 of Algorithm 1. Depending on the R# variant, the generation strategy differs. In R1 and R2#, the number of spline nodes $N_S + 1$ (and therefore the required number of decision variables) is known and common for all the trajectories (of the same generation). In R1, this number is fixed, while R2# initializes it in line 5 with its lower limit. In R3, since each trajectory may have a different amount of nodes, the number of segments in each of them is sampled from a uniform integer distribution within $[N_{S,\min}, N_{S,\max}]$. The spatial and time values of the corresponding nodes are chosen from several uniform continuous distributions. For the spatial coordinates the ranges of the distributions are the extremes of $\mathrm{F}(\cdot)$, since this function simultaneously encodes the

navigable domain and the probe reach. The time coordinates are sampled from the uniform distribution within $[0, \Delta\tau_{\max}]$, sorted (to make them monotonically increasing) and shifted by τ_0 (to make their values fix within the mission domain $[\tau_0, \tau_0 + \Delta\tau_{\max}]$).

Evaluation of the Trajectories

This step, performed in lines 8 and 15 of Algorithm 1, uses the OFs and the CFs to evaluate all the trajectories in populations *pop* and *newpop*. It starts computing the spline coefficients from the node coordinates, to be able to evaluate $\boldsymbol{S}(t)$ and its derivatives when needed. Next, it evaluates the CFs. Finally and only if the trajectory is feasible (i.e., all its $\mathrm{CF}_{\#} \leq 0$), it computes the OFs.

Selection of Parents

This step, performed in line 10 of Algorithm 1, selects N_{child} *parents* (which will be mated in pairs) from the current population *pop*. All parents are sampled using binary tournament, uniformly sampling two integer values from $[1, N_{pop}]$ and selecting the best solution (i.e., trajectory) among them. If both trajectories are in the same Pareto front, for NSGA-II we select the trajectory with the greatest crowding distance and for NSGA-III, we choose any of them with the same probability. Both methods are equivalent to those implemented in the original versions of NSGA-II [17] and NSGA-III [18,19], since the Pareto front ordering of our planner (detailed in the Recombination paragraph within Section 5.2.4) takes into account the constraint violation.

Crossover

This operator, used in line 11 of Algorithm 1 starts choosing which pairs of parents will mate with a crossover probability P_{xover}. Unselected pairs directly become *children*, while the selected pairs undergo two-point crossover, which consists in selecting two control nodes of the trajectories of the parents and forming two children, one with the first and last segment of one parent and the middle segment of the other, and another with the opposite segments. To ensure that the time coordinates of the nodes are monotonically increasing and unrepeated, the time coordinates are sorted and any coincident time is averaged between the offending time coordinate and the previous one. The first type of crossover (named X1 and used in R1 and R2#, where the trajectories of the parents have the same number of control nodes) divides the parents by the same crossing points (sampled, independently for each mating pair, from a uniform integer distribution over the number of nodes of the spline, ensuring that there is always one node left in each of the three sections in which the spline is divided) to yield equally sized children. For R3, where the trajectories of the mating parents can have different numbers of control nodes, we propose two additional strategies. In X2, the crossover points of each parent are sampled independently, being able to build children of different number of nodes than their parents (including some with 3 nodes and others with the remaining, and ensuring that all the nodes of both parents are used in one of the children). In X3, two random numbers r^1_{xover} and r^2_{xover} are chosen from a uniform continuous distribution in the range $[0, 1]$, proportionally mapped to the genome of each parent by computing $r^1_{\text{xover}} \cdot N^{q,m}_{p,\boldsymbol{S}}$ and $r^2_{\text{xover}} \cdot N^{q,m}_{p,\boldsymbol{S}}$ (where $N^{q,m}_{p,\boldsymbol{S}}$ is the number of nodes in the m parent of the q generation), and rounded to the nearest integer to determine the corresponding crossover point in each parent. We also ensure that there is at least one node in each section of each trajectory, resampling r^1_{xover} and r^2_{xover} when it is not. This yields to trajectory sections that are proportionally equal in both parents, indirectly controlling the growth and shrinking of the trajectories. It is worth noting that X3 behaves as X1, when both parents have the same number of control nodes.

Mutation

This operator, used in line 12 of Algorithm 1, introduces variability in the population by changing the values of some decision variables of all the children. It determines if a decision variable has to be mutated by checking if a random variable sampled from a uniform within $[0, 1]$ is smaller to the mutation probability $P_{\text{mut}}^{q,m} = 1/(4(N_{c,S}^{q,m} - 1) + 1)$, where $N_{c,S}^{q,m}$ is the number of nodes in the m children of the q generation. To modify the values of the mutating decision variables, a random offset, sampled from a Gaussian distribution with zero mean and σ_{mut} standard deviation, is added. Afterwards, it reorders the time coordinates that are placed out of order by the mutation and shifts the whole time vector (to make it start in τ_0). R3 supports two additional types of mutation, include and remove new control points, with a uniform probability P_{var}. The inclusion of nodes is performed according to the refinement procedure explained in the refinement paragraph of this section. Their removal, only performed when $N_{c,S}^{q,m} > 2$, is carried out by selecting any of the control points with equal probability.

Recombination

This operator, used in line 16 of Algorithm 1, sorts the trajectories of the old and new populations into Pareto fronts to select the best trajectories among both for forming the population of N_{pop} candidates of the next generation. The process starts by determining the *Pareto dominance* among all pairs of solutions within the population formed by the old and new populations. To this end, we start comparing the feasibility of the pairs and only if both solutions are feasible, compare the values of their objective functions applying the usual Pareto dominance definition (i.e., a candidate dominates the other if the values of the OFs of the candidate are all equal or better than those of the other, ref. [140]). We also make feasible solutions dominate unfeasible solutions (diverging from the standard NSGA implementation in order to encourage the convergence of the planner to feasible solutions). If both candidates are unfeasible, we compute the unfeasibility degree of each CF as $max(0, CF_{\#})$ and apply the usual dominance definition to those values. Hence, the dominance definition only distinguish those CFs which are violated and those solutions that are closer to become feasible can dominate those that are further away. While determining the Pareto dominance among all pairs of solutions, we also count the number of dominating candidates for each trajectory in the whole population, and save the list of the candidates that they dominate. Afterwards, we can easily perform the *Fast Non-Dominated Sort* by iteratively adding those solutions with count equal to zero to the current front and decreasing the counter of all the solutions in their dominated list. Hence, the first Pareto front stores all the non-dominated solutions, while the remaining contain solutions with are dominated by solutions in earlier fronts and which dominate solutions in later fronts. Finally, we conform the resulting population pop with the highest ranked solutions, taken in order according to the fronts, until reaching the maximal number N_{pop} of trajectories. A2 and A3 use two different criteria to select the solutions of the last front that is used to fill the resulting population. In particular, A2 selects those with the highest crowing distance (computed as the sum of the distances between the candidate and its neighbors for each OF) within the front; while A3 pre-computes (in line 2 of Algorithm 1) a group of reference points of the OFs space, and uses them to cluster solutions (which are nearest to the line that joins them with the origin) and to select solutions associated to reference points that cluster the least amount of solutions (in the population and in the candidate front).

Refinement

This operator adds a new control node (and therefore segment) to the spline trajectory. It is used within line 12 of Algorithm 1 to let the mutation increases the number of nodes

of the trajectory for R3 and in line 17 to let R2# increment progressively the number of nodes of the whole population (from $N_{S,\min}$ to $N_{S,\max}$, with the purpose of speeding up the convergence by reducing the complexity of the search space initially). The time coordinate of the new node is obtained, as Equation (27) states, maximizing a function that weights the cyanobacterial concentration $c_p(\boldsymbol{S}^{q,m}(t), t)$ measured by the probe at any time of the m feasible trajectory $\boldsymbol{S}^{q,m}$ of the q generation (which ends at time $\tau_f^{q,m}$) with kernels centered at the time coordinate $\tau_{S,s}^{q,m}$ of the existing nodes of the trajectory. These kernels, represented in Figure 6 using the horizontal axis for the time distance to the node in units of the function's spread parameter σ_{ref} and the vertical axis for the value of the kernel, are equal to zero at the time step of each node and tend towards 1 the further they get from it. Hence, the new node might be added near other nodes as long as the concentration there is sufficiently high, and not only to the node of highest concentration that is further enough of time of the existing control nodes as in [32]. The term ε_{ref} is added to $c_p(\boldsymbol{S}^{q,m}(t), t)$, because the latter is only computed for feasible trajectories and because for the unfeasible ones we want to determine the new node location purely from the placement of other nodes. The new node location is obtained, as Equation (28) states, from the spline at the optimal time $\tau_{S,\text{new}}^{q,m}$. The new node is inserted in the trajectory ensuring that the time coordinates remain sorted.

$$\tau_{S,\text{new}}^{q,m} = \sup_{t\in[\tau_0,\tau_f^{q,m}]} \left((c_p(\boldsymbol{S}^{q,m}(t)) + \varepsilon_{\text{ref}}) \prod_{s\in[1,N_S^{q,m}]} \left(\frac{-1}{1 + \frac{|t-\tau_{S,s}^{q,m}|}{\sigma_{\text{ref}}}} + 1 \right) \right) \tag{27}$$

$$\boldsymbol{x}_{S,\text{new}}^{q,m} = \boldsymbol{S}^{q,m}(\tau_{S,\text{new}}^{q,m}) \tag{28}$$

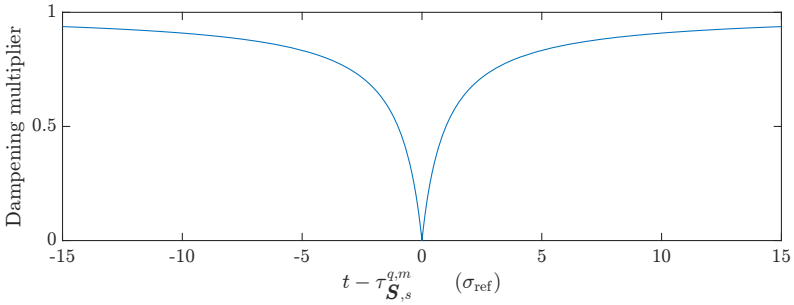


Figure 6. Damping function for refinement.

After having explained how the refinement of a trajectory is performed, we detail some additional operations within the Refinement operator in line 17 of Algorithm 1. After adding the new node, the OFs and CFs are also recomputed. Besides, the operation is only performed at some generations g. For configuration P (Periodic), we pre-compute as in [32] how often the Refinement of the whole population must take place rounding up the result of $N_{\text{gen}}/(N_{S,\max} - N_{S,\min} + 1)$. For configuration C (Convergence), the refinement of the entire population is controlled by four different conditions. The first one enforces that at least $N_{\text{gen,ref}}$ generations have elapsed between one refinement and the next. The second condition checks if the dominated HyperVolume Estimate (HVE) improvement of the current generation HVE^q with respect the one of the one $N_{\text{gen,ref}}$ generations early (i.e., $\text{HVE}^{q-N_{\text{gen,ref}}}$) is smaller than a given tolerance ϵ_{ref}. We follow [141] for computing HVE as the percentage of N_{HVE} samples that 1) are randomly obtained from the objective space (using the uniform distributions $[0, \Delta\tau_{\max}]$, $[0, v_{\text{H,max}}\Delta\tau_{\max}]$ and $[0, A_c|\mathcal{K}|]$ respectively for OF_1, OF_2 and OF_3, where $|\mathcal{K}|$ is the cardinality of the $\mathcal{K}$ set) and that 2) are dominated by the feasible solutions of the first Pareto front of the current population. Note that the sample points are generated only once, in line 1 of Algorithm 1, in order to stabilize this metric. The third condition blocks the addition of nodes until at least the first Pareto front

contains feasible solutions, while the fourth stops the refinement once the required number of spline segments $N_{S,\max}$ has been reached.

Stop Condition

The planner stops after reaching the maximum amount of generations N_{gen}, with remains unchanged for all the executions of a variant.

6. Trajectory Planner Results

This section analyzes the performance of several variants of the planner under six different scenarios, which are defined by one of the simulations computed in Section 4, by other problem parameters (e.g., the limits of the ASV constraints) and by mission criteria (i.e., start time, and start and end location of the ASV+probe). To this end, Section 6.1 introduces the planner variants under analysis and their parametrization, Section 6.2 details the testing scenarios, Section 6.3 describes the analysis methodology presented in the paper to characterize and compare the performance of the different variants of the planner and against two baseline strategies, Section 6.4 discusses their results, and Section 6.5 provides some solution trajectories to each scenario as examples.

6.1. Planner Variants

With the different algorithms (NSGA-II and NSGA-III), codification and operators presented in Section 5.2, we set up the 10 variants of the planner specified in the rows of Table 6, where the first column shows the label we have assigned to each variant, encoding in the two first characters the algorithm (further detailed in the second column of the table), in the next three characters the refinement type (third and fourth column), and in the last two characters the crossover operator (fifth column). The last column reminds which variants introduce or delete control points of the spline defining the ASV+probe trajectory within the mutation operator.

Table 6. Summary of planner variants under analysis, indicating with a check mark if the corresponding feature is included in the variant.

Variant	Algorithm	Refinement		Crossover	Mutation
		Type	Control	Points	Add&Rem
A2R1-X1	NSGA-II	None	-	Common	
A2R2PX1	NSGA-II	Incremental	Periodic	Common	
A2R2CX1	NSGA-II	Incremental	Convergence	Common	
A2R3-X2	NSGA-II	VLC	-	Independent	✓
A2R3-X3	NSGA-II	VLC	-	Proportional	✓
A3R1-X1	NSGA-III	None	-	Common	
A3R2PX1	NSGA-III	Incremental	Periodic	Common	
A3R2CX1	NSGA-III	Incremental	Convergence	Common	
A3R3-X2	NSGA-III	VLC	-	Independent	✓
A3R3-X3	NSGA-III	VLC	-	Proportional	✓

Hence, the variants under analysis are defined by means of the planner parameters A#, R# and X#. Other planner parameters used in this paper for analyzing the performance of the planner are presented in Table 7. The encoding parameters which define the number of segment of the spline depends on the refinement type. For R1 (without refinement), the number of segments is set to $N_S = 11$. For R2#, it starts at $N_S^0 = 4$, and progressively increases up to $N_S^f = 11$. For R3, the minimum and maximum number of segments of the splines in the initial population and immigrants is also set to $N_{S,\min} = 4$ and $N_{S,\max} = 11$ (and may go out of these bounds during the optimization due to crossover and mutation).

Note that all these parameters have been selected after performing multiple tests over different scenarios. The remaining parameters of the planner are explained in the following sections, since they are related to each problem and mission.

Table 7. Planner parameters.

Variable	Value	Variable	Value
N_{gen}	1000	P_{var} (R3)	1%
N_{pop}	100	σ_{ref} (R2#, R3)	150 s
N_{child}	90	ε_{ref} (R2C)	0.05%
P_{xover}	80%	$N_{\text{gen,ref}}$ (R2C)	30
$\boldsymbol{\sigma}_{\text{mut}}$	$\text{diag}([200\,\text{m}, 200\,\text{m}, 2\,\text{m}, 2000\,\text{s}])$		

6.2. Planning Scenarios

The six planning scenarios under analysis in this paper are defined by the parameters of each problem and mission. The problem parameters are those needed to evaluate the OFs and the CFs that do not depend on the spline, while the mission parameters particularize each problem to the start time τ_0 of the operation, and to the start and end location, $\boldsymbol{l}_{S,0}$ and $\boldsymbol{l}_{S,\text{f}}$, of the ASV+probe trajectory.

Since each planning scenario is associated to one of the simulation scenarios presented in Section 4.3.3 (which is provided to the planner through the cyanobacteria particles evolution $\boldsymbol{l}_{\text{p}}^{k}$, the amplitude A_{c} and spread $\boldsymbol{\sigma}_{c}$ of the concentration Gaussian function, and the steady state water flow $\boldsymbol{u}_{\text{ss}}$), they inherit their labels. Besides, the problem parameters presented Table 8 are the same for all the scenarios, because they are directly related to the physical capabilities of the ASV and of the probe. Additionally, as the navigable domain is dependent on the water body where each simulation was carried out (WB1 or WB2), two different $\text{F}(\cdot)$ are used in this paper. In particular, the navigable zone of WB1 (used in SC1, SC2 and SC3) is the water domain Ω reduced by (1) a margin of 50 m around the shores (to ensure the ASV safety), by (2) a semi-elliptical area in the east of the domain, and by (3) the deepest restriction of the probe $z_{\text{min}} = -15\,\text{m}$. Similarly, the navigable zone of WB2 (used in SC4, SC5 and SC6) is bounded by a security margin 15 m inside the shores, and the same deepest restriction of the probe $z_{\text{min}} = -15\,\text{m}$. Finally, the mission parameters of each scenario are displayed in Table 9.

Table 8. Problem parameters.

Variable	$\Delta\tau_{\text{max}}$	$v_{\text{H,max}}$	$a_{\text{H,max}}$	$\kappa_{\text{H,max}}$	$v_{\text{V,max}}$	$a_{\text{V,max}}$
Value	3 h	$1\,\text{m/s}$	$0.1\,\text{m/s}^2$	$0.2\,\text{m}^{-1}$	$1\,\text{m/s}$	$0.1\,\text{m/s}^2$

Table 9. Mission parameters.

Scenario	τ_0	$\boldsymbol{l}_{S,0}$	$\boldsymbol{l}_{S,f}$
SC1	65 h	$(-750, -600, -0.1)^{\top} m$	$\boldsymbol{l}_{S,0}$
SC2	54 h	$(380, 100, -0.1)^{\top} m$	$(-100, -450, -0.1)^{\top} m$
SC3	6 h	$(300, -550, -0.1)^{\top} m$	$\boldsymbol{l}_{S,0}$
SC4	47 h	$(60, -115, -0.1)^{\top} m$	$\boldsymbol{l}_{S,0}$
SC5	56 h	$(60, -115, -0.1)^{\top} m$	$\boldsymbol{l}_{S,0}$
SC6	21 h	$(60, -115, -0.1)^{\top} m$	$\boldsymbol{l}_{S,0}$

Figure 7 summarizes some characteristics of the planning scenarios. In particular, the color map represents the mean concentration of cyanobacteria projected onto the horizontal plane during the mission (i.e., only from τ_0 and $\tau_0 + \Delta\tau_{\text{max}}$ and not during the whole simulation as happens in Figures 4 and 5), the white lines the bounds of the

navigable zone (note that the south-east peninsula in SC4, SC5 and SC6 is not navigable due to its shallow depth), and the red x and dot the starting and ending location of the ASV (which coincide in all scenarios except SC2). The particularities and challenges of each scenario are the following.

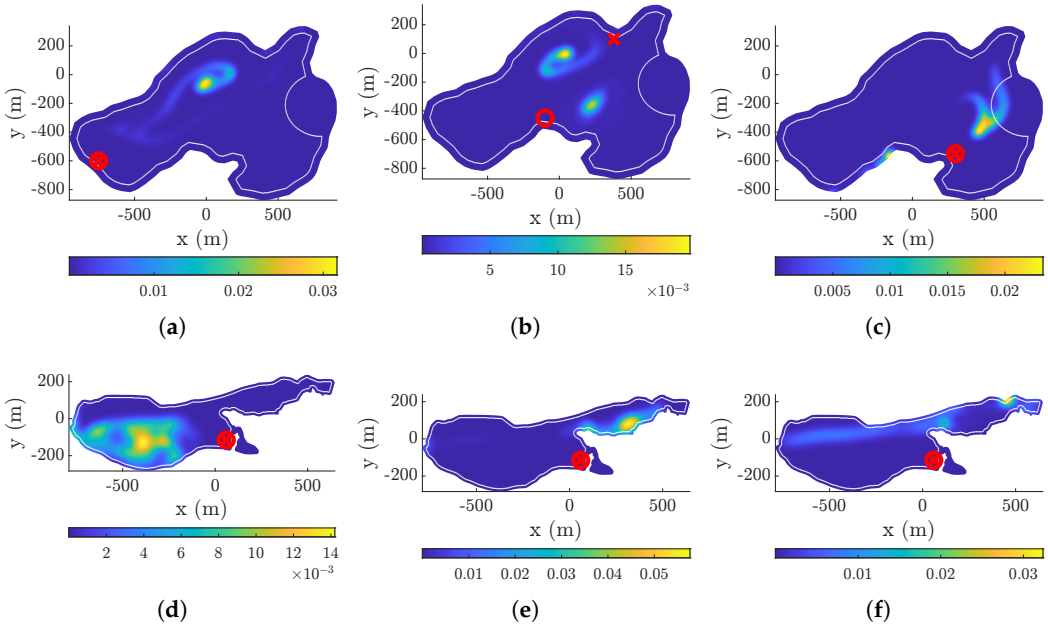


Figure 7. Planning scenarios. (**a**) SC1. (**b**) SC2. (**c**) SC3. (**d**) SC4. (**e**) SC5. (**f**) SC6.

Scenario SC1 (Figure 7a)

It has a high concentration area with low fluid velocities and a trail of cyanobacteria leading to it. Hence, SC1 allows us to test the capabilities of the planner in a simple scenario, and see if it is able to fine-tune a trajectory that traverse the areas with presence of cyanobacteria, including those with the highest concentration.

Scenario SC2 (Figure 7b)

It has two high concentration areas and a restricted mission time to reach them. Ideally, it should find an efficient trajectory that traverses both.

Scenario SC3 (Figure 7c)

The high cyanobacterial concentration areas move fast dragged by the main currents of the lake. The challenge is to find a trajectory that coordinates the ASV with the moving cyanobacteria and allows the ASV to get measurements in the area of interest while going with or against the currents.

Scenario SC4 (Figure 7d)

Although the simulation started with particles close to a vortex as in SC1, the currents have distributed the cyanobacteria over a greater area. Hence, the planner needs to find a trajectory that covers a wider region without focusing in any particular spot.

Scenario SC5 (Figure 7e)

The highest concentration area is placed in a zone of difficult access. Since a higher number of spline segments should facilitate the maneuverability through the strait that gives access to area of interest, SC5 put a strain on the planner refinement methods.

Scenario SC6 (Figure 7f)

Cyanobacteria are distributed in a straight line close to the north shore. The planner has to find a safe trajectory that, if possible, goes also inside the difficult-to-access area.

6.3. Comparison Methodology and Results

This section presents the methodology that we propose to characterize and compare the results obtained by the 10 variants of the planner for the six scenarios (yielding a total of $10 \times 6 = 60$ combinations) and by 2 baseline deterministic strategies. Due to the stochastic nature of NSGA, we run each planner variant 50 times (yielding a total of $60 \times 50 = 3000$ executions). During these executions, run from MATLAB R2023b [133] inside an Intel Core i7-6700HQ CPU (2.60 GHz) with 16 GB of RAM, the computation time and decision variables of the first Pareto front of all the generations of the planner are saved for its subsequent analysis. This analysis, carried out (also in MATLAB) in Section 6.4, consists of the perspectives described in the following sections.

6.3.1. Computational Requirements and Feasibility

The computational requirements of each variant are characterized by (1) representing, against the planner computation time and for each scenario, the number of active executions of each variant (in order to see how their executions are ending), and (2) by computing the mean execution time per generation of the variants for each scenario and for all of them.

The feasibility is characterized representing, against the computation time and for each scenario, the number of executions where the first Pareto front contains feasible solutions.

The results of these analyses are shown in Figure 8 and Table 10. In particular, Figure 8 contains a different subfigure per scenario and within it, two different graphics. The top one represents the evolution of the active executions (i.e., those that have not reached $N_{\text{gen}} = 1000$) of each variant with a different colored line. The bottom one represents the evolution of the number of executions where the first Pareto front is feasible, using a different row per variant, green to identify the variants where all the executions have a first feasible Pareto front, a red-yellow scale to indicate how the number of executions with a first feasible Pareto front grows, and white to remind that all the executions have ended. Otherwise, Table 10 stores the mean execution times per generation, using a row for each variant and a column for each scenario (except the last one, that is for the mean of all of them), highlighting in green the best mean execution time of each column, and in red the worst.

Table 10. Mean execution times (in seconds) per generation, indicating in red and green the worst and best execution times, respectively.

Variant	SC1	SC2	SC3	SC4	SC5	SC6	All
A2R1-X1	16.68	**13.67**	9.21	**13.45**	16.06	**10.67**	13.36
A2R2PX1	13.41	9.50	4.86	11.24	11.77	7.07	9.68
A2R2CX1	13.13	9.39	4.99	11.20	12.15	7.12	9.71
A2R3-X2	13.50	9.75	4.80	11.28	11.08	6.98	9.61
A2R3-X3	13.23	9.68	4.77	10.24	10.48	6.94	9.27
A3R1-X1	**17.08**	13.37	**9.61**	13.42	**16.81**	10.01	**13.44**
A3R2PX1	13.05	9.27	4.96	11.25	11.42	7.49	9.61
A3R2CX1	13.02	**9.17**	4.98	11.49	11.97	7.52	9.73
A3R3-X2	11.93	9.44	4.78	10.95	10.96	**6.65**	9.16
A3R3-X3	**11.88**	9.42	**4.75**	**10.01**	**10.42**	6.76	**8.91**

The higher execution times observed for the R1 variants occur because the evaluation of the objective and constraint functions takes longer when the number of spline segments is permanently fixed at N_S =11 (creating a search space of 41 decision variables) for all trajectories in the population. R2/R3 variants benefit, from the computational time point of view, for an increasing/adaptable search space. The dependency of the computation time with the refinement type is a direct observation on how the number of control points impacts the mathematical complexity of the optimization algorithm.

Without going in further details, it is easy to observe that while some variants need some time to find feasible solutions, others end sooner. In order to fairly perform the remaining comparisons of the paper, we will only use information from feasible solutions/fronts and, after finishing each of the executions of the planner, from its final best Pareto front.

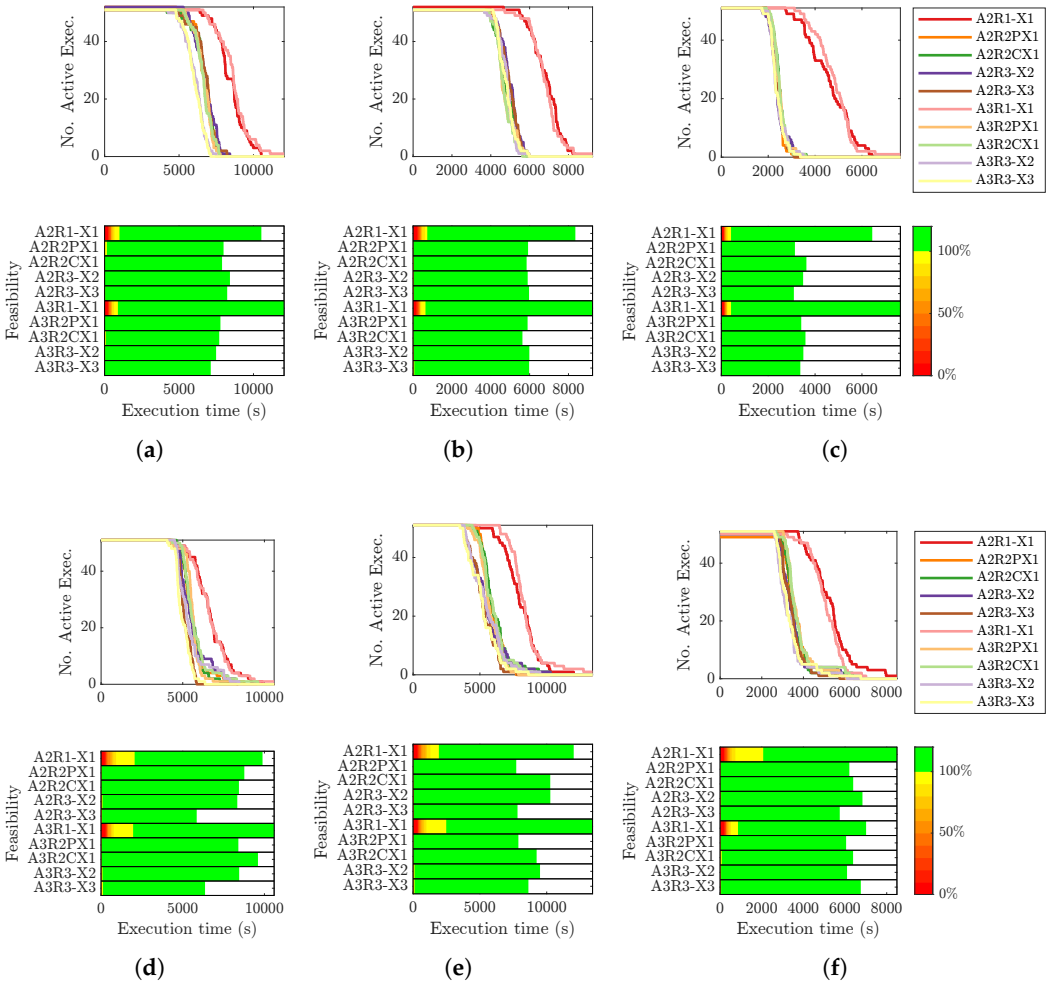


Figure 8. Evolution of the number of active executions and of feasible first Pareto fronts. (**a**) SC1. (**b**) SC2. (**c**) SC3. (**d**) SC4. (**e**) SC5. (**f**) SC6.

6.3.2. Extreme Solutions of the First Feasible Pareto Front

It is also interesting to see how the values of the OFs evolve. In this case, we will focus on the analysis of the evolution of the mean and 95% confidence intervals, for the runs of each variant, of the best values of each OF. We have selected only the values of the extreme elements of the first Pareto front (i.e., the best of each OF of the best available solutions) to distinguish the variability among executions from the variability within the best Pareto front.

The results of this analysis are represented in Figure 9 and Table 11. In particular, Figure 9 presents 18 graphics, organized row-wise by scenarios and column-wise by OFs. Within each graphic, the mean and 95% interval of the corresponding OF and scenario are represented with different colored lines and envelopes for each variant. For comparison purposes, lower is best for OF_1 and OF_2 (which have to be minimized), while higher is best for OF_3 (which has to be maximized). Besides, as many lines and envelopes overlap, we group them, according to a careful inspection of each graphic in Figure 9, in up to three clusters, which are identifiable by the green (best group), yellow (intermediate group) and

red (worst group) cells of Table 11, where the variants are organized row-wise, and the OFs and scenarios column-wise.

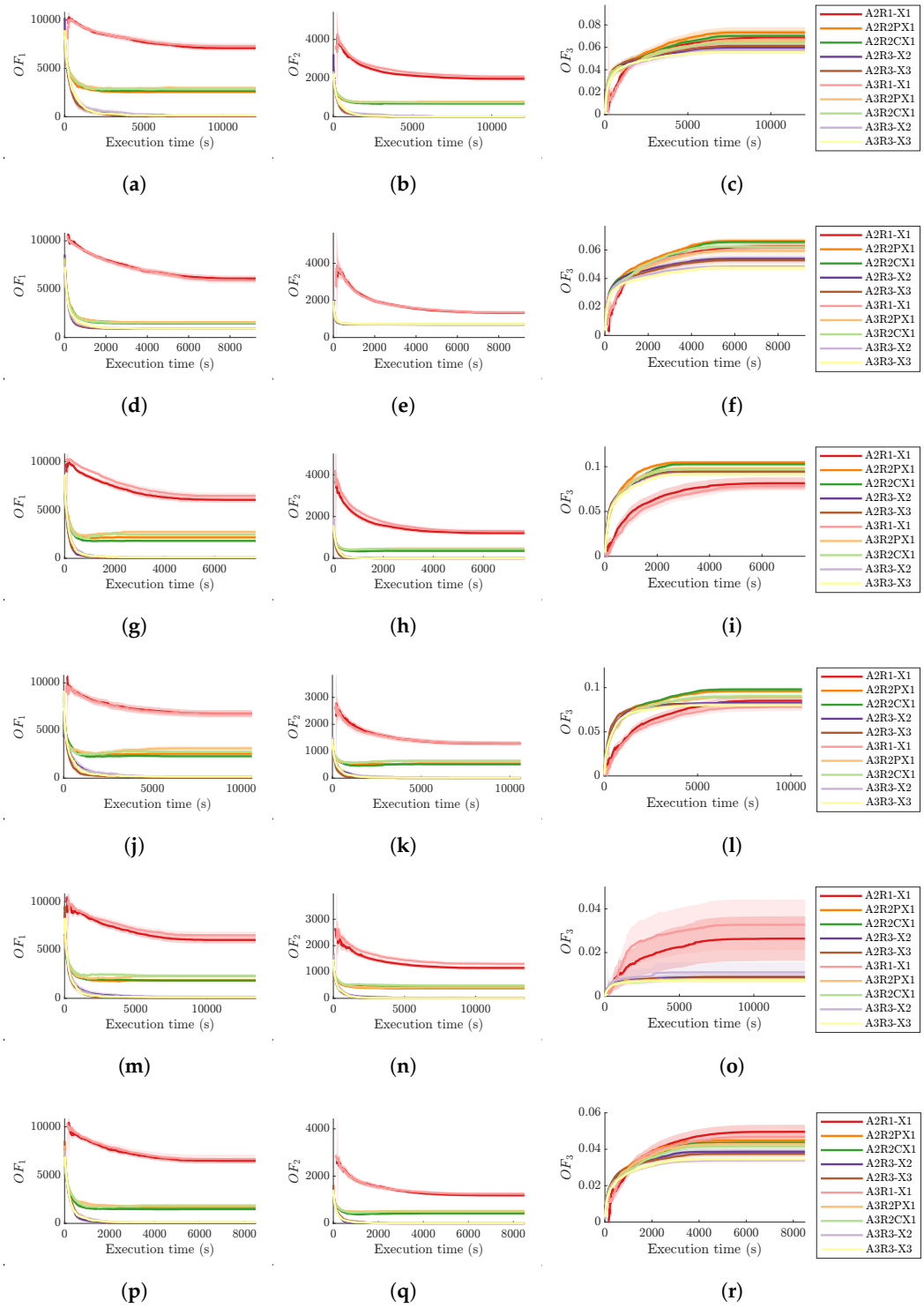


Figure 9. Evolution of the best values of the OFs per scenario. (**a**) OF_1 in SC1. (**b**) OF_2 in SC1. (**c**) OF_3 in SC1. (**d**) OF_1 in SC2. (**e**) OF_2 in SC2. (**f**) OF_3 in SC2. (**g**) OF_1 in SC3. (**h**) OF_2 in SC3. (**i**) OF_3 in SC3. (**j**) OF_1 in SC4. (**k**) OF_2 in SC4. (**l**) OF_3 in SC4. (**m**) OF_1 in SC5. (**n**) OF_2 in SC5. (**o**) OF_3 in SC5. (**p**) OF_1 in SC6. (**q**) OF_2 in SC6. (**r**) OF_3 in SC6.

Table 11. Summary of the comparison of extreme solutions of the first Pareto front, indicating in red, yellow and green the worst, intermediate and best clusters, respectively.

Variant	OF_1 and SC#						OF_2 and SC#						OF_3 and SC#					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
A2R1-X1																		
A2R2PX1																		
A2R2CX1																		
A2R3-X2																		
A2R3-X3																		
A3R1-X1																		
A3R2PX1																		
A3R2CX1																		
A3R3-X2																		
A3R3-X3																		

6.3.3. Statistical Comparison of the Best Feasible Pareto Fronts

The previous analysis ignores the information of the non-extreme solutions of the best Pareto front. While standard multi-objective metrics like hypervolume and 3D Pareto front visualizations are common to overcome this issue, they present significant drawbacks for our specific evaluation. Visualizing 3D Pareto fronts for 10 variants yields highly cluttered plots that are difficult to interpret. Furthermore, standard metrics like hypervolume (since they require arbitrary penalty assignments for constraint violations) are problematic to fairly evaluate populations that contain unfeasible solutions, which are frequent in the early stages of our optimization process. Therefore, we propose using a non-parametric statistical framework, which is fully capable of handling population sets with differing sizes, unfeasible solutions, and varying variances without relying on arbitrary penalties.

More specifically, our test compares the information of the best Pareto fronts of each execution and variant based on their rankings. We determine if there is a significant difference among the ranks of solutions within the best Pareto front using the Skillings-Mack [38] omnibus test (or the Mann-Whitney U test [40] when only two variants are comparable, because the solutions of the remaining variants are still unfeasible), and, when the test turns out to be positive, we perform a pair-wise post-hoc Nemenyi test [39] to determine which variants differ. The first test is selected because is valid for unbalanced block designs, a property especially useful in our case, since the number of feasible solutions within the best Pareto front at a given computation time can differ among the executions of the variants of the planner. In addition, unlike more common statistical tests, this test does not assume normality or homogeneity of variance between the variants. The second test is performed as it provides, with a given confidence level, the pairs of groups that are statistically different and the ranks of the groups according to the mean rank within each group. Hence, it allows us to identify the best, worst and intermediate variants. We carry out the tests with different rankings of the solutions in the first Pareto fronts in order to characterize the planners from three complementary perspectives.

The objective of the first ranking is to extend the analysis of the evolution of the OF of the extreme points of the best Pareto front (performed in previous section) to all their solutions. To this end, for each scenario and computation time under analysis, all the solutions within the best feasible Pareto front of all the executions of all the variants are ranked according to each OF independently. The statistical analyses of these rankings are summarized in the graphics of Figure 10, respectively organized row-wise and column-wise by scenario and OF. Within each graphic, we represent the evolution of this analysis

along the computation time, using a different row per variant. We highlight in white when there were no still feasible solutions in any of the executions of the variant. We highlight in green/red the best/worst variants (i.e., those where the Skillings-Mack or the Mann-Whitney U tests succeed with $p \leq 0.05$), as well as those variants that do not significantly differ from them (i.e., those with $p \leq 0.05$ according to the Nemenyi or Mann-Whitney U tests). Finally, we highlight in gray the remaining variants.

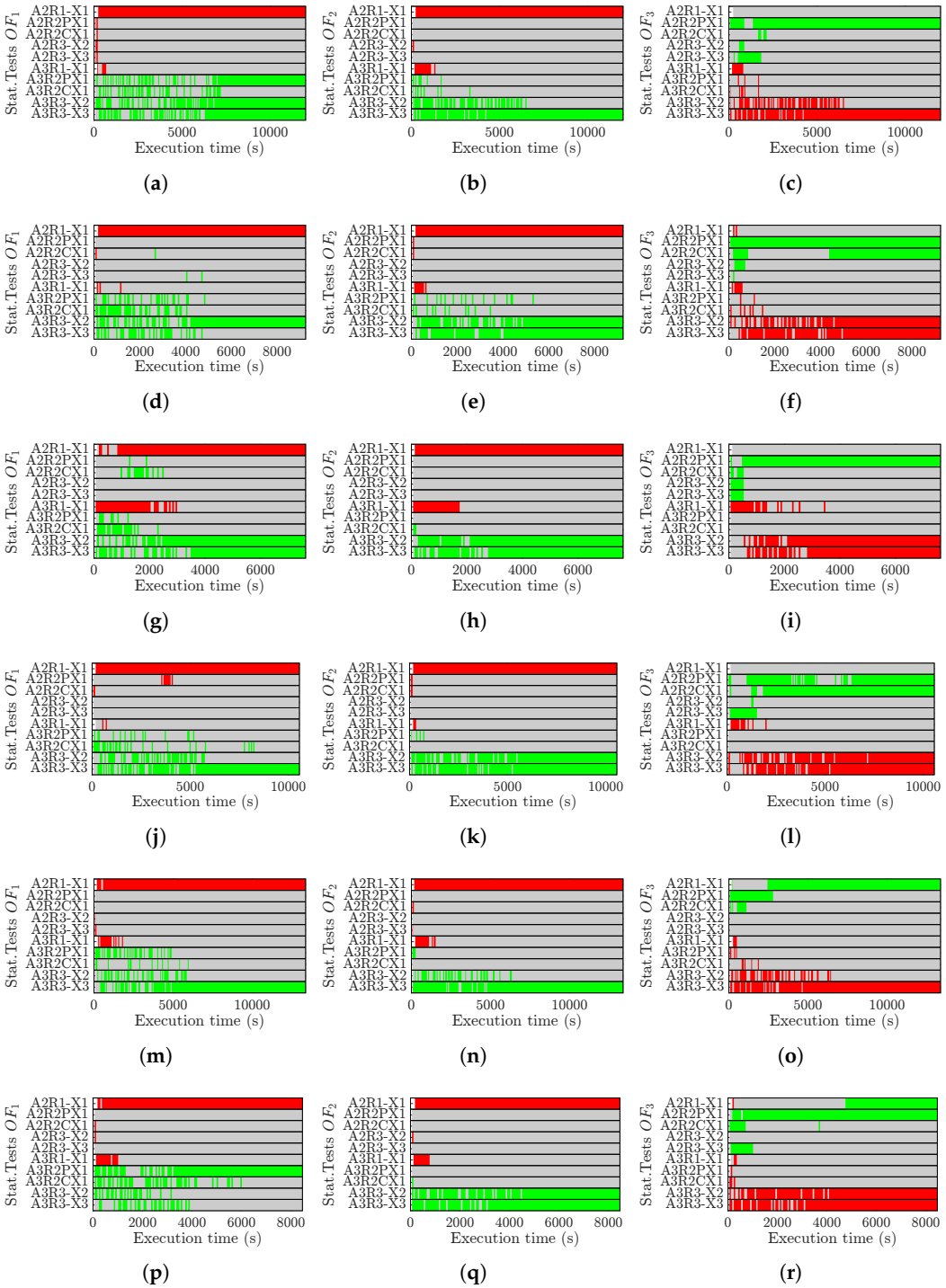


Figure 10. Statistical analysis of the OFs values in the best Pareto fronts per scenario. (**a**) OF_1 in SC1. (**b**) OF_2 in SC1. (**c**) OF_3 in SC1. (**d**) OF_1 in SC2. (**e**) OF_2 in SC2. (**f**) OF_3 in SC2. (**g**) OF_1 in SC3. (**h**) OF_2 in SC3. (**i**) OF_3 in SC3. (**j**) OF_1 in SC4. (**k**) OF_2 in SC4. (**l**) OF_3 in SC4. (**m**) OF_1 in SC5. (**n**) OF_2 in SC5. (**o**) OF_3 in SC5. (**p**) OF_1 in SC6. (**q**) OF_2 in SC6. (**r**) OF_3 in SC6.

The objective of the second ranking is to determine the best variants to optimize, for each scenario, all the OFs simultaneously (and therefore exploit better NSGA parent selection and recombination mechanisms). To this end, for each scenario and computation time under analysis, all the solutions within the best feasible Pareto front of all the executions of all the variants are sorted in Pareto fronts and tied ranked according to them. The statistical analyses of these rankings are summarized in the graphics of Figure 11. Although the row-wise and color interpretation of these graphics are similar to the ones in the previous analyses, in this case we determine which variants obtain better or worse solutions in their best feasible Pareto fronts.

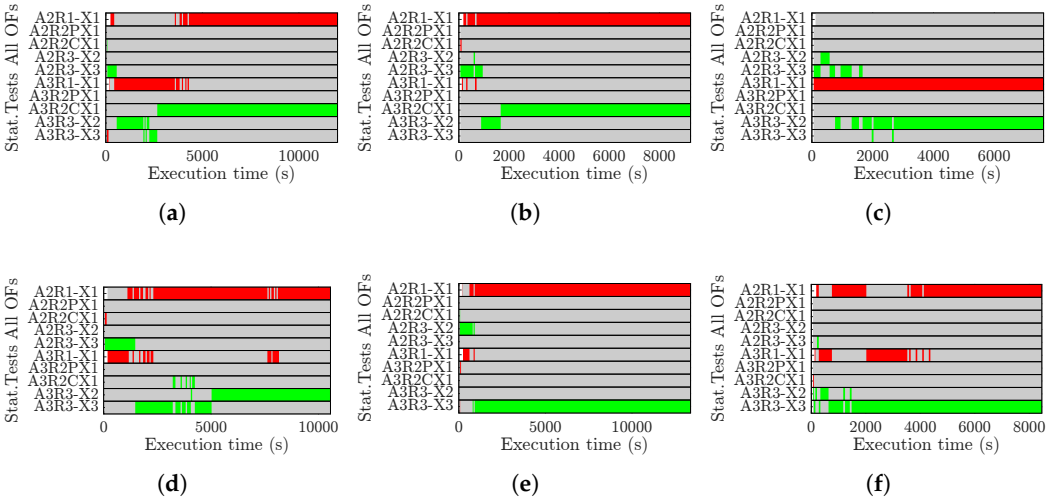


Figure 11. Simultaneous statistical analysis of all OFs in the best Pareto fronts per scenario. (**a**) SC1. (**b**) SC2. (**c**) SC3. (**d**) SC4. (**e**) SC5. (**f**) SC6.

The objective of the third approach is to combine the 24 previous analysis, performed for 6 scenarios with 4 ranking criteria (OF_1, OF_2, OF_3 and Pareto front belonging), in only 4 studies (according to each ranking criterion). To this end, each scenario is treated as an experimental block, whose ranking (according to the selected criterion), can be combined with the ranking (according to the same criterion) of the others, using the error adjustments of Skillings-Mack, Mann-Whitney and Nemenyi tests. The statistical analysis of the results of the combined rankings are summarized in Figure 12, using a different graphic for each ranking criterion. Although the color interpretation of these graphics are similar to the ones in the previous analyses, in this case we are determining which variant has obtained, for each of the 4 criterion under analysis, better or worse results for all the scenarios under study.

Additionally, we want to highlight that when comparing variants considering several OFs, it is common to treat them independently and group them in blocks, following the same strategy that we use for grouping the results of several scenarios [142], or to use performance metrics that combine them in some other way [143]. However, we have decided to use the organization in Pareto fronts as the ranking criteria for analyzing all the OFs simultaneously, because their values are correlated within the front, and our strategy preserves this correlation. The use of blocks to group the different scenarios is supported by the fact that the planner obtains incomparable results from different scenarios.

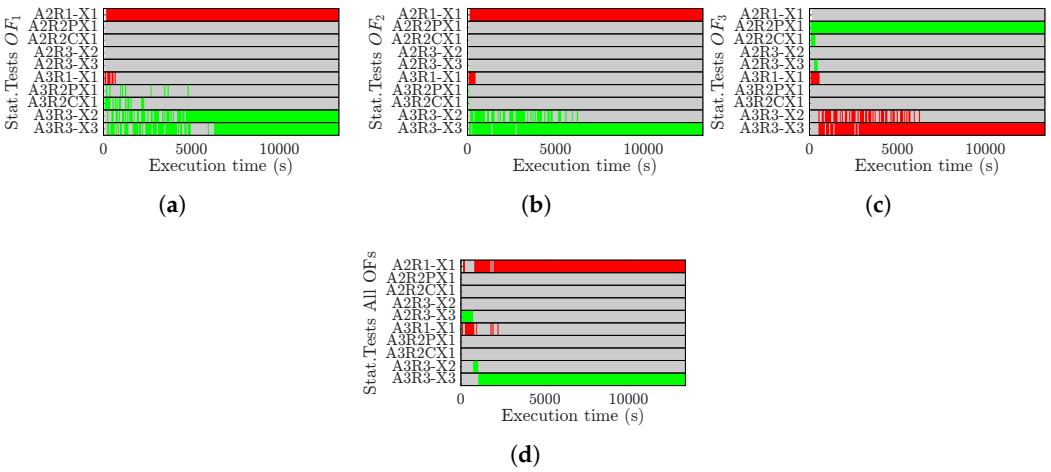


Figure 12. Combined statistical analysis over all the scenarios. (**a**) OF_1. (**b**) OF_2. (**c**) OF_3. (**d**) All OFs in all scenarios.

Finally, Table 12 summarizes the main differences among these four statistical analyses, which can be associated with the way the solutions are ranked and grouped as part of the non-parametric statistical tests, where each variant represents a treatment and each scenario, if several are considered within the same test, a block. Regarding the rankings, they consider the feasible solutions within the first Pareto front of all the executions and variants under study. Whenever each objective is considered independently, the ranking indicates the order of the solution within the combined population, and whenever all objectives are considered simultaneously, the ranking indicates in which combined Pareto front the solutions falls into. Regarding the groups, all the solutions of the same scenario belong to the same block of the statistical test, and the solutions are ranked in each of the different scenarios independently, returning the characterization of each scenario on its own, or the characterization of all the scenarios together considering the correlation effects within the blocks.

Table 12. Summary of statistical tests.

		Ranking	
		Comparing Single Objectives OF_i	**Comparing All Objectives [OF_1, OF_2, OF_3] Through Pareto Dominance**
Scenario	Individual scenario	Analysis 1, Figure 10	Analysis 2, Figure 11
Groups	All the scenarios	Analysis 3, Figure 12a–c	Analysis 4, Figure 12d

6.3.4. Comparison with Baseline Strategies

The results of the proposed planner are compared against two baseline strategies: A LawnMower (LM) coverage and a Greedy Best-First Search (GBFS). Both are deterministic (making unnecessary a statistical analysis of their results) and evaluated according to the same OFs and (when possible) enforcing the same CFs that our planner uses to enable a fair comparison.

The LM mission is configured to cover areas of high cyanobacterial concentration, which are manually set up for each scenario, both in the horizontal plane and vertically. In practice, to ensure the automatic generation of the mission as part of an autonomous planner, the area of the lawnmower mission would need to be determined by an optimizer, but this is out of the scope of this work. In the manually defined area, the lawnmower

trajectory follows parallel rows across the domain separated by 50 m, which corresponds to $2\sigma_c$ in the horizontal plane, to ensure that the probe is always at most σ_c away from any cyanobacteria particle in the same plane. The points of the lawnmower trajectory outside the navigable domain are discarded but, since the areas are manually adjusted, no obstacle avoidance strategies are introduced. Consequently, the constraint CF_7, which ensures the safe navigation of the ASV within the Ω is typically met but not explicitly enforced. The ASV and probe are configured to always move at the maximum allowable speed ($v_{H,max}$ horizontally and $v_{V,max}$ vertically), implicitly enforcing constraints CF_2 and CF_5. The mission is truncated when the maximal duration $\Delta\tau_{max}$ is reached, enforcing also CF_1, although no truncation was needed in the areas manually configured for the six scenarios. The trajectory starts by traveling from the initial location $\boldsymbol{l}_{S,0}$ to the start of the lawnmower sweeps, and ends by traveling to the final location $\boldsymbol{l}_{S,f}$ from the position of the ASV and probe at the end of the lawnmower sweeps (or from the point where the path is truncated). The time needed to reach the start and end locations are considered within the mission duration. Since a lawnmower pattern changes its direction drastically at the end of its row, the acceleration at these points becomes infinite, and acceleration constraints CF_3, CF_4 and CF_6 cannot be met. Nevertheless, this could be solved, by substituting the abrupt changes of directions of the pattern by arcs of curvature that can be followed by our ASV.

The GBFS mission is configured to start at the initial location $\boldsymbol{l}_{S,0}$, to iteratively sample neighboring locations of a 3×3 grid of points separated by 10 cm around the current ASV+probe location, to select the new location that maximizes the expected cyanobacterial concentration (given by Equation (12)), and to terminate when the maximal duration $\Delta\tau_{max}$ is reached by traveling to the final location $\boldsymbol{l}_{S,f}$, in order to enforce CF_1. Locations that are previously visited in the trajectory or outside the navigable domain are discarded from the search space. As in the lawnmower planner, the ASV and probe travel at their maximal speed, enforcing CF_2 and CF_5. In this case, no acceleration constraints CF_3, CF_4 and CF_6 are handled, while the domain restriction CF_7 (Equation (26)) is not explicitly enforced, but usually upheld by following the gradient, discarding points outside the navigable domain and taking small steps.

The results of the baseline strategies and their comparison against our planner can be found in Tables 13 and 14. As our planner returns multiple solutions within the first Pareto Front and the baseline strategies return only one, to perform the comparisons of this section, we have selected the trajectory with the best value of OF_3 of the last generation of each planner execution and variant. We have prioritized OF_3 as the primary selection criterion because the goal of the ASV monitoring mission is to identify the bloom's location, and therefore, maximizing the probe observations in areas of expected high cyanobacterial concentration is crucial.

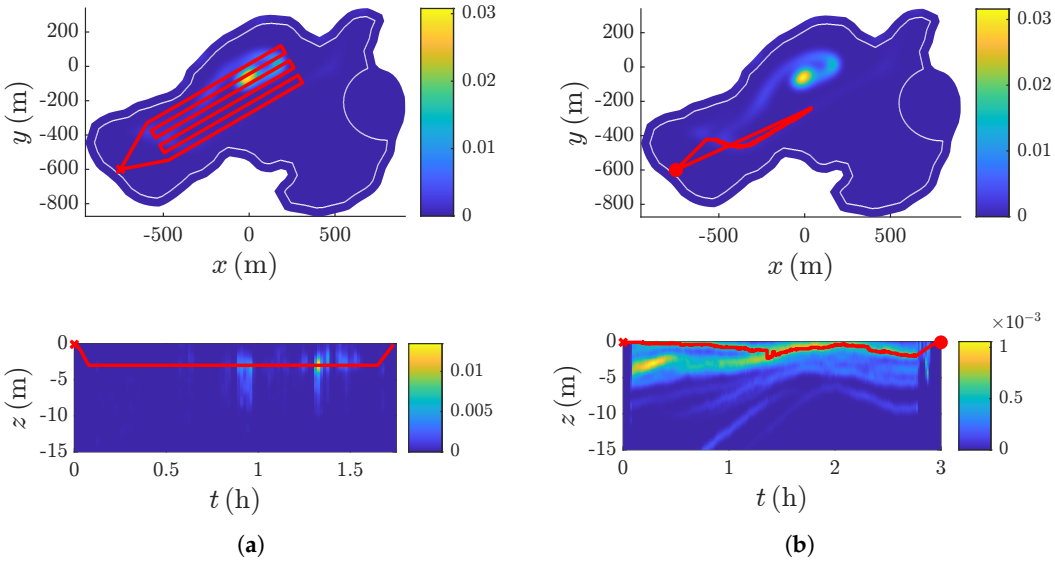
For the comparison in Table 13, we compute the mean values of each OF for the selected solutions for three different cases: across *all* the variants and executions, across the executions of the A3R3-X3 variant (which will be identified in the analysis performed in Section 6.4.3 as the overall best planner), and across the executions of variant A2R2PX1 (which will be identified in the analysis performed in Section 6.4.2 as the best planner regarding OF_3). We report these mean values (accompanied, within parentheses, by the 95% confidence interval half-range) in the last three columns of the table, and the OF values associated to the unique solution returned by the two baseline strategies in the previous two columns. The values of each OF are grouped in consecutive rows, which are associated to the scenario indicated in the second column of the table. Within each row (scenario and OF), the best/worst values are respectively highlighted in green/red. Note that the values of SC5 for the baseline strategies are missing because they have not found a path within the water domain that reaches the location of the cyanobacteria.

Table 13. OF comparison of the baseline planners and our planner in each scenario, indicating in red and green the worst and the best, respectively.

	Scenario	Baselines		Ours		
		LM	GBFS	All	A3R3-X3	A2R2PX1
OF_1 (h)	SC1	1.74	3.00	2.33 (±0.04)	2.54 (±0.09)	2.05 (±0.12)
	SC2	1.19	3.00	1.38 (±0.05)	0.79 (±0.08)	1.33 (±0.11)
	SC3	0.69	3.00	1.91 (±0.05)	2.02 (±0.15)	1.42 (±0.17)
	SC4	1.07	3.00	2.47 (±0.02)	2.33 (±0.10)	2.58 (±0.07)
	SC5	–	–	2.08 (±0.04)	2.13 (±0.12)	1.84 (±0.13)
	SC6	0.90	3.00	1.91 (±0.05)	1.77 (±0.21)	1.94 (±0.18)
OF_2 (m)	SC1	6263	10,800	2366 (±32)	2194 (±36)	2219 (±51)
	SC2	4288	10,800	1632 (±46)	1160 (±47)	1484 (±56)
	SC3	2513	10,800	1087 (±40)	719 (±22)	912 (±34)
	SC4	3852	10,800	1905 (±31)	1577 (±16)	1860 (±51)
	SC5	–	–	1349 (±37)	1022 (±98)	1282 (±59)
	SC6	3254	10,800	1202 (±36)	910 (±36)	955 (±31)
OF_3 ($\times 10^{-3}$)	SC1	59.35	3.70	63.85 (±0.88)	56.05 (±1.83)	73.74 (±3.43)
	SC2	55.14	2.32	57.03 (±0.61)	46.63 (±1.25)	66.19 (±1.32)
	SC3	46.00	86.43	93.36 (±0.93)	91.46 (±1.18)	103.98 (±1.71)
	SC4	70.97	18.89	86.47 (±0.72)	80.04 (±0.61)	96.41 (±1.41)
	SC5	–	–	12.06 (±1.08)	7.27 (±0.36)	8.79 (±0.39)
	SC6	14.67	26.23	40.85 (±0.53)	32.50 (±1.06)	44.73 (±0.91)

For the comparison in Table 14, we include the solution provided by each baseline strategy within the three groups of selected solutions (i.e., the selected solutions for all the variants and executions, for the executions of A3R3-X3 and for the execution of A2R2PX1) and rank the solutions within each group according to their Pareto dominance and according to each of the OF individually. The values of each group (all, A3R3-X3, A2R2PX1) are reported in consecutive rows, which are associated to the scenario indicated in the second column of the table. The remaining columns are grouped by the baseline result which is included in the comparison (LM or GBFS), and within each group of columns, the first column shows information related to the Pareto ranking reported as *the front where the solution of the baseline solution falls / the total number of fronts (the size of the Pareto front where the solution of the baseline planner falls)*, and the consecutive three columns show the rank of the baseline solution within that front for each of the OFs independently. The cases where the baseline solutions fall in the first/last Pareto fronts are highlighted in green/red. No solutions are reported for SC5, because the baseline strategies have not produced a feasible solution to be used in this comparison.

Finally, as an example, Figure 13 shows the ASV+probe trajectories of the baseline planners in SC1. Each sub-figure is composed of two images. The upper ones represent in the color scale the cyanobacterial concentration $c_p(\cdot)$ projected onto the horizontal plane, with a white line the limits of the navigable volume $F(\cdot)$, and with a red line the resulting ASV trajectory. In the bottom images, the horizontal axes represent the time span from the start of the mission until its end, and the vertical ones the depth of the probe (starting from the water Γ_{surf} and reaching to its maximum depth). The color scales represent the cyanobacteria concentration $c_p(\cdot)$ projected into the vertical axis z, and the red lines represent the probe spline trajectory.



(**a**) (**b**)

Figure 13. Trajectories of baseline strategies for SC1. (**a**) Lawnmower planner. (**b**) GBFS planner.

Table 14. Pareto ranking of each baseline and their within-front rank for each OF, indicating in red and green the last and first fronts, respectively.

	Scenario	LM				GBFS			
		Front	OF_1	OF_2	OF_3	Front	OF_1	OF_2	OF_3
All	SC1	5/14 (45)	2	45	27	15/15 (1)	1	1	1
	SC2	9/15 (24)	1	24	19	16/16 (1)	1	1	1
	SC3	4/17 (44)	1	44	44	15/17 (9)	9	9	1
	SC4	1/12 (69)	1	69	66	13/13 (1)	1	1	1
	SC5	–	–	–	–	–	–	–	–
	SC6	7/13 (47)	1	47	47	14/14 (1)	1	1	1
A3R3-X3	SC1	1/6 (14)	2	14	8	7/7 (1)	1	1	1
	SC2	3/4 (14)	12	14	2	5/5 (1)	1	1	1
	SC3	2/6 (12)	1	12	12	5/6 (3)	3	3	1
	SC4	1/5 (9)	1	9	9	6/6 (1)	1	1	1
	SC5	–	–	–	–	–	–	–	–
	SC6	3/4 (14)	1	14	14	5/5 (1)	1	1	1
A2R2PX1	SC1	3/4 (7)	1	7	6	5/5 (1)	1	1	1
	SC2	5/5 (2)	1	2	2	6/6 (1)	1	1	1
	SC3	3/4 (7)	1	7	7	5/5 (1)	1	1	1
	SC4	1/5 (21)	1	21	21	6/6 (1)	1	1	1
	SC5	–	–	–	–	–	–	–	–
	SC6	4/4 (2)	1	2	2	5/5 (1)	1	1	1

6.4. Discussion of the Planner Performance

In this section we discuss the results summarized in the figures and tables presented in the previous one.

6.4.1. Computational Requirements and Feasibility

Table 10 and the top axis of each graphic in Figure 8 show that A2*R1*-X1 and A3*R1*-X1 are significantly computationally slower than the rest. This happens because the evaluation of the OFs and CFs in R1 variants takes longer, since the number of splines segments in all trajectories of the populations of the planner is $N_S = 11$. Although the rest of the variants are more homogeneous regarding execution time, there is a slight positive bias towards A3 variants and more specifically A3R3-X3.

The bottom axis of each graphic in Figure 8 show that the convergence rate to feasible fronts of both R1 variants is also significantly slower.

Note that non-feasible solutions are faster to compute since the OFs can be skipped, so the computation time difference is even more remarkable in the case of R1. Also note that a green bar that ends sooner in Figure 8 does not necessarily mean that the algorithm is on average faster, as shown in Table 10. The bar ends only whenever the last execution of a variant is finished. Such discrepancy can be observed in scenario SC3, in which the faster variant on average is A3R3-X3, but A2R3-X3 is the first variant to finish all its executions.

6.4.2. Extreme Solutions of the First Feasible Pareto Fronts

The first and second column of graphics in Figure 9 show that the evolution of the mean and standard deviation of the best values of OF_1 and of OF_2 within the first feasible Pareto front (computed over the 50 executions of each variant) are tied to three groups. In addition, Table 11 shows that while R3 variants always belong to the best (green) group, R1 variants always conform the worst (red) group. The sensibility of OF_1 and OF_2 to the number of spline segments justifies this behavior: R3 variants benefit from their capability to work with trajectories with small number of nodes, while R1 variants struggle to reduce the duration and length of trajectories randomly generated with a high number of segments.

The third column of graphics in Figure 9 show that the mean and standard deviation of the best values of OF_3 within the first feasible Pareto front are tied to two or three groups. In this case, Table 11 shows that the variants belonging to each group are highly dependent on the scenario; that R3 variants are never among the best group; that R1 and R2 variants can belong to the three groups; and that A2R2PX1 is the best option. We believe that A2R2PX1 specially benefits from the periodic addition of control points in the spline over regions with high concentration of cyanobacteria, while R1 variants take advantage sometimes of the fact that longer trajectories can observe more areas with high concentration of cyanobacteria.

Figure 9 also shows that the standard deviation of R1 variants is often the widest. This happens because R1 variants handle a big number of decision variables, making it harder to reach solutions of similar quality at different executions. In contrast, the incremental refinement mechanisms in R2 and the VLC encoding of R3 help to overcome this issue successfully.

6.4.3. Statistical Comparison of the Best Feasible Pareto Fronts

The previous analysis only considers the values of the OFs of the extreme solutions, assuming that they are independent on each other and Gaussian. Beyond both unratified assumptions, the remaining values of the OFs in the best feasible Pareto front are ignored, overlooking the benefits of the simultaneous optimization of all of them that NSGA provides.

Figure 10 overcomes the last two issues by summarizing, for each scenario, the statistical analysis of the evolution of *the ranking of all the values within the best Pareto front* of the different OFs. Regarding OF_1 and OF_2, NSGA-III R3 variants are usually among the best, while A2R1-X1 is always within the worst. Besides, for OF_1, NSGA-III R2 variants are also sometimes useful. Regarding OF_3, A2R2PX1 is usually among the best, while NSGA-III R3 variants are always among the worst. Exceptionally, for SC5, the dominant variant on OF_3 is A2R1-X1 (since the area with the highest cyanobacterial concentration is secluded, and variants that produce longer trajectories from the start are better to explore the domain). Figure 12a–c aggregate the rankings of the different scenarios for each of the OFs, letting us conclude that A3R3-X2 and A3R3-X3 are the best options for optimizing

OF_1 (while optimizing also OF_3), and that A2R2PX1 is the best option for optimizing OF_3 (while optimizing OF_1 and OF_2).

The previous analysis still overlooks the correlation that exists among the OFs of the solutions in the first Pareto front. This question is tackled by the statistical analysis summarized in Figure 11, where for each scenario *the ranking of the Pareto fronts in which the solutions within the best Pareto front can be organized* is analyzed. In this case, the worst options are associated to R1 variants, (especially to A2R1-X1), while the best options usually belong to NSGA-III, more specifically to A3R2CX1 in SC1 and SC2, and to both R3 variants (i.e., A3R3-X2 or A3R3-X3) in the remaining scenarios. Moreover, when analyzing the solutions as a whole, using the Pareto front definition that is exploited by our planner, Figure 12d shows that, after aggregating the results of all the scenarios, the worst option is the naivest A2R1-X1, while the best one is A3R3-X2, which is backed up by the novelties introduced in this work: NSGA-III recombination mechanisms, the VLC encoding and the proportional crossover.

6.4.4. Comparison with Baseline Strategies

The comparison with baseline strategies highlights the advantages of setting up a multi-objective optimization problem and choosing a NSGA-based optimizer to solve it.

The objective functions are not considered in the baseline planners, which results in missions that, despite moving at the maximal velocity, are longer than the alternatives proposed by our planner. In the case of LM, since the area to cover is manually adjusted, the temporal efficiency (OF_3/OF_1) is higher, but in the GBFS the full mission time is spent without focusing on the most affected areas. Also note that both baselines are less spatially efficient (OF_3/OF_2) than our approach. In the LM example shown in Figure 13a, the probe covers the designated space, but although it was manually defined, it spends most of the trajectory in areas that are unaffected by cyanobacteria. In the GBFS solution displayed in Figure 13b, the probe follows the cyanobacterial concentration gradient to find itself in a local minimum chasing a group of particles across the domain without being able to reach the areas of globally high cyanobacterial concentration. Despite enforcing a constant maximum velocity, the gradient dictates a back and forth around the local optima that results in an effectively slower and redundant trajectory.

These drawbacks are characterized across all the scenarios in Table 13, which let us compare independently the values of the OF of the solutions returned by the baseline strategies with the mean value of each of the OFs for the solutions with the best OF_3 across all executions and variants, across the executions of the overall best variant, and across the executions of the variant that is the best suited to optimize OF_3. Generally, the baseline strategies usually obtain significantly worse OF_2 and OF_3 values than our NSGA-based approach. While the LM strategy always produces short trajectories in time, the distance covered is always greater and the concentration observation is usually lower than in NSGA. This happens because the LM strategy always uses the maximal speed, which can become a problem for the accuracy of the sensors. In contrast, the GBFS planner does not consider the distance or duration of trajectory beyond the truncation of the mission, which causes OF_1 and OF_2 to reach their feasible maxima, which in a minimization problem becomes the worst possible value. Both baseline strategies usually under-perform in OF_3 when compared against our planner. While the LM planner has an advantage in OF_1 and OF_2, for OF_3, depending on the scenario, the obtained result is better in the LM or GBFS planner, with GBFS being better in scenarios where the trajectory is initialized near the global maximum of concentration. However, neither of them is able to improve upon the results obtained by NSGA. In addition, due to the simplicity and limitations of the baseline

strategies, no feasible solutions are produced in SC5, while NSGA is able to find a solution in this scenario in all executions and variants.

Comparing only the mean values reported in Table 13 for the groups of solutions associated to our planner, we can observe that A3R3-X3 and A2R2PX1 usually outperform the combined population of all the selected solutions. The first of these variants, since it was selected considering all objectives simultaneously, is more balanced and performs better in OF_2; while A2R2PX1, since it was selected based on OF_3 performance, usually provides better concentration observations, while still being competitive in the other objectives. Between these two variants, the performance in OF_1 depends on the scenario, which highlights the quality of the overall results and the effectiveness of using a multi-objective approach during the optimization. Extraordinarily, due to the characteristics of SC5, the selected variants under-perform with respect to the overall mean with regards to OF_3. This happens because the cyanobacteria are located in a hard-to-reach area, which favors variants that are generally less efficient and produce longer and more winding trajectories.

The results of Table 14 show, using a Pareto perspective, the advantage of our planner against both baseline strategies. In particular, GBFS is consistently the worst among all the solutions provided by the multiple executions of our planner, regardless of considering the full results, or only the solutions of a selected variant. In contrast, the LM planner, adjusted manually, usually produces a solution that is contained in one of the middle Pareto fronts, although depending on the scenario and the selected solutions of our planner, it can also be categorized in the last or first front. However, inspecting the individual OF ranking within that front, we can observe that the LM solution always represents an extreme or nearly extreme solution of the front, usually with a good OF_1 value, but with a significantly bad OF_2 and OF_3 values, which implies that it is placed in the first Pareto front because it obtains really competitive results with respect to the duration of the mission, while sacrificing the other objectives.

Finally, we want to highlight that there is a significant difference in the computational search effort of the (deterministic single-pass operational) baseline strategies and of our (offline multi-objective EA-based) planner. Nevertheless, as the planner is to be used offline and its computation time is appropriate for our purpose, the comparison of this section is set up 1) to draw the reader's attention to the main objective of the monitoring mission by selecting those solutions with the best value of OF_3, and 2) to analyze the performance of each of the OFs independently and of all of them simultaneously. Furthermore, the most competitive of the two baselines, LM, has been adjusted manually, which requires both human intervention, and a time and knowledge effort that can be avoided with our proposal. It is worth noting that the manual operations required before running the LM strategy can be avoided by substituting the decision variables of our planner with those required to set up the LM strategy manually, and then using this strategy to obtain the trajectory that is evaluated by the planner. The price to pay in this case, is that the new planner will also have to be run for the time required to achieve convergence to overall good solutions, making this approximation similar to ours at the expense of losing the flexibility that our spline encoding brings to the problem.

6.5. Examples of Trajectories Returned by the Planner

Figure 14 shows an example of the ASV+probe trajectories generated by the planner for each scenario. In particular, we show, for each scenario, the solutions with the best value of OF_3 obtained by A3R3-X3, which is the overall best option according to the previous statistical tests. We maintain the representation elements and colors used in Figure 13, incorporating the spline control nodes as red points in the trajectory.

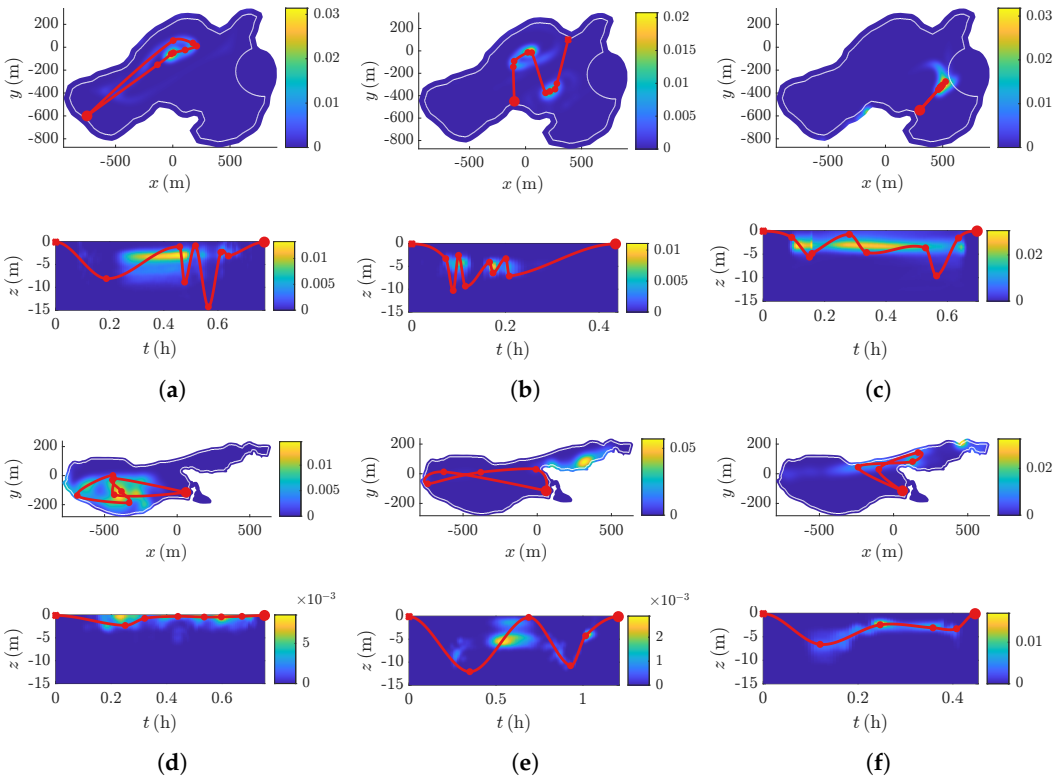


Figure 14. Example solutions of each scenario using variant A3R3-X3. (**a**) SC1. (**b**) SC2. (**c**) SC3. (**d**) SC4. (**e**) SC5. (**f**) SC6.

In scenario SC1 (Figure 14a), the ASV trajectory quickly moves to the area of interest following one of the cyanobacteria trails and finely adjust itself to cover the areas of high cyanobacteria concentration. Vertically, and especially in areas of high concentration, the probe is lowered and risen at the appropriate times in order to observe the cyanobacteria at different depths.

In SC2 (Figure 14b), the ASV approaches the first vortex following a trail, adjusts its trajectory to cover it, travels to the second vortex while capturing other trails, and covers it precisely. In the way from and to the vortices, the probe slowly moves vertically to be at the right depth when it reaches the area of interest. Once there, it moves up and down more quickly around the depths of high cyanobacterial concentration.

For SC3 (Figure 14c), the ASV quickly moves to the area of interest and then slows down to observe the particles that move fast towards the outlet. It focuses on the area where the particles change direction, which is the easiest to manage. The probe moves vertically throughout the entire trajectory to observe the greatest amount of cyanobacteria.

In SC4 (Figure 14d), where the cyanobacteria concentration is spread, the ASV moves as it was solving a covering problem, following a trajectory that adheres to the southern shore and covers the areas of high concentration without getting caught up in them. The probe depth is adjusted below the water surface, where cyanobacteria are concentrated.

The geometric constraints of Scenario SC5 (Figure 14e) are so challenging that the ASV cannot find the entrance to the north-eastern region of the lagoon and navigate back out without running into the shores of the strait. So, the ASV moves up to the outlet of the domain where a lower cyanobacteria concentration area is placed.

Finally, in SC6 (Figure 14f), the ASV ventures into the secluded section of the domain aided by the cyanobacteria trial, observing particles while approaching and returning, and skillfully avoiding the shores. Vertically, the probe adjusts its depth to the cyanobacteria distribution throughout the ASV trajectory.

7. Conclusions

This paper presents an approach for smartly exploring lentic water bodies with cyanobacteria blooms, using model-based simulations, ASVs and Evolutionary Algorithms. In particular, it presents a complex physical three-dimensional model that predicts the cyanobacteria distribution, taking into account the fluid dynamics, the water pressure, the thermocline and pycnocline, the irradiance, and the cyanobacteria buoyancy and growth. In turn, the results of its simulations are used in an AI-based path planner that optimizes the trajectory of an ASV and the vertical displacement of its probe, taking into account three objective functions (related to the effectiveness of the exploration) and seven restrictions (related to the viability of the path from the ASV and safety perspective). The planner is backed up by NSGA-II and NSGA-III, and three different strategies for determining the number of spline nodes in the trajectory (fixed, progressively increased, and variable). A robust statistical comparison method, based on the Skillings-Mack, the Nemenyi and the Mann-Whitney U tests, is also proposed to identify those configurations of the planner that are better under different conditions. Finally, we perform six different simulations to pose different types of challenges for the planner, and analyze the performance of 50 executions of 10 variants of the planner using the proposed statistical method. The results show that, when aggregating the results of all scenarios and objectives, the best variant is NSGA-III with VLC and proportional crossover (A3R3-X3). However, for specific objectives or scenarios the results vary. In particular, NSGA-III with VLC and independent crossover version (A3R3-X2) is also advantageous for obtaining shorter (in distance and time) trajectories, while the cyanobacterial concentration along the trajectory is better optimized with NSGA-II and incremental periodic refinement (A2R2PX1).

While the current models provide a solid baseline for ASV trajectory planning, the dynamic nature of cyanobacterial blooms is highly sensitive to environmental factors. In order to improve our planner further and validate its field detection capabilities, future work will focus on: validating the multiphysics model outputs against field measurements; performing Monte Carlo simulations to account for the uncertainty of the model inputs; performing a comprehensive sensitivity analysis to assess the model's generalizability and robustness to diverse environmental parameters (including surface irradiance and attenuation); coupling a climate model to a turbulent time-dependent fluid dynamic model to improve the accuracy of the thermocline; and improving the performance of the planner by fine-tuning its parameters, exploring other encodings or extending its support to other population-based muti-objective optimizers.

Author Contributions: Conceptualization, E.B.-P., J.A.L.-O. and G.C.-B.; methodology, G.C.-B.; software, G.C.-B.; validation, G.C.-B.; formal analysis, G.C.-B.; investigation, G.C.-B.; resources, J.A.L.-O., E.B.-P. and J.L.R.-M.; data curation, G.C.-B.; writing—original draft preparation, G.C.-B.; writing—review and editing, E.B.-P., J.A.L.-O. and J.L.R.-M.; visualization, G.C.-B.; supervision, E.B.-P., J.A.L.-O. and J.L.R.-M.; project administration, J.A.L.-O. and E.B.-P.; funding acquisition, J.A.L.-O. and E.B.-P. All authors have read and agreed to the published version of the manuscript.

Funding: Different parts of this work have been supported by the Research Projects IA-GES-BLOOM-CM (Y2020/TCS6420) from the Synergy program of the Comunidad de Madrid (Spain); BEST-WATER (PID2024-156033OB-C31) funded by MICIU/AEI/10.13039/501100011033 and by ERDF/EU; and INSERTION (PID2021-127648OB-C33) from the Knowledge Generation Projects program of the Spanish Ministry of Science and Innovation of Spain. The work by Gonzalo Carazo-Barbero is specifically supported by the 2020 Pre-Doctoral for Training Research Personnel Program (CT82/20) of the University Complutense of Madrid and the Santander Bank.

Data Availability Statement: The simulation scenarios, datasets, and detailed parameters supporting the findings of this study are available from the corresponding author upon reasonable request. The source code is part of an ongoing project and is not publicly available at this time.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A. Parameter Fitting

The calibration of this process-based model requires parameters that are not directly measurable under field conditions, especially those linked to buoyancy regulation and light-driven migration. Hence, we estimate them by matching the simulated cyanobacterial positions over time to observed vertical behavior. The fitting procedure is designed for a black-box stochastic simulator where each candidate parameter set runs a full particle simulation, the resulting mean particle depth is compressed into a smooth proxy function, and that proxy is compared against a target curve built from observations.

In the following we detail the methodology to optimize the parameters, and its application to the simulations set up in this paper.

Appendix A.1. Methodology

Let $\mathbf{\Psi}$ be the vector of physical parameters to calibrate, and let $\mathrm{Model}(\mathbf{\Psi})$ denote the simulated particle positions over time. Instead of comparing raw particle clouds directly, we compare two low-dimensional proxy functions: a target proxy $\mathrm{Y}_{\mathrm{target}} = \mathrm{Y}(\mathbf{\Psi}'_{\mathrm{target}})$ (from the observations) and a fitted proxy $\mathrm{Y}_{\mathrm{fit}} = \mathrm{Y}(\mathbf{\Psi}'_{\mathrm{fit}})$ (from the simulation output), both adjusted through an input fitting parameter vector. The outer optimization problem is

$$\begin{aligned} \mathbf{\Psi}^* &= \arg\min_{\mathbf{\Psi}} \left\| \mathrm{Y}_{\mathrm{target}} - v(\mathrm{Model}(\mathbf{\Psi})) \right\| \\ &\text{s.t. any applicable constraints,} \end{aligned} \tag{A1}$$

where $v(\cdot)$ is the operator that fits a proxy to the simulated depths. This two-level optimization reduces noise from particle-level variability and makes the calibration more stable for long simulations, where direct point-wise matching is highly sensitive. The approach is robust in the sense that each simulation and each proxy fit returns a valid objective value inside the admissible search region. However, because the simulator is nonlinear and non-convex, the solutions of the outer problem can be captured in local minima.

Appendix A.2. Application

For the simulations presented in this paper, the observations correspond to an expert-defined day-night vertical migration pattern of cyanobacteria (ascent at night, near-surface residence in early morning, descent during daytime). We make $\mathbf{\Psi} = [I_{\mathrm{lim}}, I_{\mathrm{thres}}, \alpha, \mathrm{d}\rho_{\mathrm{max}}, d_{\mathrm{p}}]$. Bounds are imposed component-wise (Table A1), together with $I_{\mathrm{lim}} > I_{\mathrm{thres}}$.

Table A1. Parameter values used in the parameter fitting.

Variable	Initial Value	Lower Limit ($\mathbf{\Psi}_{\mathbf{lower}}$)	Upper Limit ($\mathbf{\Psi}_{\mathbf{upper}}$)	Final Value ($\mathbf{\Psi}^*$)
I_{lim} ($\mathrm{W\,m^{-2}}$)	150	100	800	338
I_{thres} ($\mathrm{W\,m^{-2}}$)	5	1	100	25.5
α ($\mathrm{s^{-1}}$)	2×10^{-5}	10^{-10}	10^{-4}	8.96×10^{-6}
$\mathrm{d}\rho_{\mathrm{max}}$ ($\mathrm{kg\,m^{-3}\,s^{-1}}$)	10^{-5}	10^{-10}	10^{-3}	8.32×10^{-4}
d_{p} (m)	8×10^{-4}	10^{-4}	5×10^{-3}	2.96×10^{-3}

Both target and fitted proxies use the same truncated sinusoidal:

$$\Upsilon(\boldsymbol{\Psi}') = \min\left\{0,\ \Psi'_1 \sin\left(\frac{2\pi}{\Delta t_{\text{day}}} t + \Psi'_2\right) + \Psi'_3\right\}, \tag{A2}$$

where $\boldsymbol{\Psi}' = [\Psi'_1, \Psi'_2, \Psi'_3]$ are the fitting parameters of the function. The target parameters $\boldsymbol{\Psi}'_{\text{target}}$ are selected so that Υ_{target} follows the expected migration cycle. For each simulation, proxy parameters $\boldsymbol{\Psi}'_{\text{fit}}$ are obtained by minimizing mean absolute error between particle depths $l^k_{\text{p},z}\ \forall k \in \mathcal{K}(t)$ and candidate proxy values:

$$\begin{aligned} \boldsymbol{\Psi}'_{\text{fit}} &= \arg\min_{\boldsymbol{\Psi}'} \ \text{MAE}\left(l^k_{\text{p},z},\ \Upsilon(\boldsymbol{\Psi}')\right) \\ &\text{s.t. } \boldsymbol{\Psi}'_{\text{lower}} \leq \boldsymbol{\Psi}' \leq \boldsymbol{\Psi}'_{\text{upper}}. \end{aligned} \tag{A3}$$

The proxy-fit settings (initial guess, lower/upper limits, and fitted values) are reported in Table A2. The outer objective compares target and fitted proxies through the Euclidean distance metric between the vectors of proxy parameters $\boldsymbol{\Psi}'_{\text{target}}$ and $\boldsymbol{\Psi}'_{\text{fit}}$.

Table A2. Parameter values of the proxy functions.

Variable	Initial/Target Value (Ψ'_{target})	Lower Limit (Ψ'_{lower})	Upper Limit (Ψ'_{upper})	Initial Fitted Value (Ψ'_{fit})	Final Fitted Value (Ψ'_{fit})
Ψ'_1	2.5	0	2π	2.7742	1.9480
Ψ'_2	2.5	0	10	1.0505	2.8052
Ψ'_3	-2	-10	10	2.4422	-2.1548

Both the inner proxy fit and the outer parameter search are solved with MATLAB's `fmincon` function, using up to 1000 objective evaluations in each inner optimization and up to 100 in the outer optimization. To keep computational cost manageable, each outer objective call runs a shortened simulation (72 h, $\Delta t = 120$ s, 50 particles). The domain is a closed rectangular prism with null background flow, and initial particle locations are sampled from a Gaussian cloud around the origin. Shared model settings are taken from Table 2.

References

1. 2030 Agenda for Sustainable Development. 2016. Available online: https://sdgs.un.org/goals (accessed on 19 October 2024).
2. Meriluoto, J.; Spoof, L.; Codd, G. *Handbook of Cyanobacterial Monitoring and Cyanotoxin Analisis*; Wiley: Hoboken, NJ, USA, 2017.
3. Falconer, I.R. Potential impact on human health of toxic cyanobacteria. *Phycologia* **1996**, *35*, 6–11. [CrossRef]
4. Fernando Bellém, S.N.; Morais, M. Cyanobacteria Toxicity: Potential Public Health Impact in South Portugal Populations. *J. Toxicol. Environ. Health Part* **2013**, *76*, 263–271. [CrossRef] [PubMed]
5. United States Environmental Protection Agency. Fourth Unregulated Contaminant Monitoring Rule. 2016. Available online: https://www.epa.gov/dwucmr/fourth-unregulated-contaminant-monitoring-rule (accessed on 14 October 2024).
6. Comission, E. Directive (EU) 2020/2184 of the European Parliament and of the Council. 2020. Available online: https://eur-lex.europa.eu/eli/dir/2020/2184/oj (accessed on 14 October 2024).
7. Srivastava, A.; Singh, S.; Ahn, C.Y.; Oh, H.M.; Asthana, R.K. Monitoring Approaches for a Toxic Cyanobacterial Bloom. *Environ. Sci. Technol.* **2013**, *47*, 8999–9013. [CrossRef]
8. Le Vu, B.; Vinçon-Leite, B.; Lemaire, B.J.; Bensoussan, N.; Calzas, M.; Drezen, C.; Deroubaix, J.F.; Escoffier, N.; Dégrés, Y.; Freissinet, C.; et al. High-frequency monitoring of phytoplankton dynamics within the European water framework directive: Application to metalimnetic cyanobacteria. *Biogeochemistry* **2011**, *106*, 229–242. [CrossRef]
9. Miller, T.R.; Tarpey, W.; Nuese, J.; Smith, M. Real-Time Monitoring of Cyanobacterial Harmful Algal Blooms with the Panther Buoy. *ACS ES&T Water* **2022**, *2*, 1099–1110. [CrossRef]
10. Hitz, G.; Pomerleau, F.; Garneau, M.E.; Pradalier, C.; Posch, T.; Pernthaler, J.; Siegwart, R.Y. Autonomous Inland Water Monitoring: Design and Application of a Surface Vessel. *IEEE Robot. Autom. Mag.* **2012**, *19*, 62–72. [CrossRef]

11. Rogalus, M.K.; Watzin, M.C. Evaluation of sampling and screening techniques for tiered monitoring of toxic cyanobacteria in lakes. *Harmful Algae* **2008**, *7*, 504–514. [CrossRef]
12. Smith, V.H. Eutrophication of freshwater and coastal marine ecosystems: A global problem. *Environ. Sci. Pollut. Res. Int.* **2003**, *10* 126–139. [CrossRef] [PubMed]
13. Moreira, C.; Vasconcelos, V.; Antunes, A. Cyanobacterial Blooms: Current Knowledge and New Perspectives. *Earth* **2022**, *3*, 127–135. [CrossRef]
14. Kim, S.; Kim, S.; Mehrotra, R.; Sharma, A. Predicting cyanobacteria occurrence using climatological and environmental controls. *Water Res.* **2020**, *175*, 115639. [CrossRef] [PubMed]
15. Steuer, R.; Knoop, H.; Machné, R. Modelling cyanobacteria: From metabolism to integrative models of phototrophic growth. *J. Exp. Bot.* **2012**, *63*, 2259–2274. [CrossRef]
16. Ranjbar, M.H.; Hamilton, D.P.; Etemad-Shahidi, A.; Helfer, F. Individual-based modelling of cyanobacteria blooms: Physical and physiological processes. *Sci. Total Environ.* **2021**, *792*, 148418. [CrossRef]
17. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
18. Deb, K.; Jain, H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints. *IEEE Trans. Evol. Comput.* **2014**, *18*, 577–601. [CrossRef]
19. Jain, H.; Deb, K. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. *IEEE Trans. Evol. Comput.* **2014**, *18*, 602–622. [CrossRef]
20. Lucas, C.; Hernadez-Sosa, D.; Caldeira, R. Multi-Objective Four-Dimensional Glider Path Planning using NSGA-II. In Proceedings of the IEEE/OES Autonomous Underwater Vehicle Workshop (AUV), Porto, Portugal, 6–9 November 2018.
21. Ghambari, S.; Golabi, M.; Lepagnot, J.; Brévilliers, M.; Jourdan, L.; Idoumghar, L. An Enhanced NSGA-II for Multiobjective UAV Path Planning in Urban Environments. In Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI), Baltimore, MD, USA, 9–11 November 2020.
22. Singh, M.K.; Choudhary, A.; Gulia, S.; Verma, A. Multi-objective NSGA-II optimization framework for UAV path planning in an UAV-assisted WSN. *J. Supercomput.* **2023**, *79*, 832–866. [CrossRef]
23. Perez-Carabaza, S.; Besada-Portas, E.; Lopez-Orozco, J.A.; de la Cruz, J.M. A Real World Multi-UAV Evolutionary Planner for Minimum Time Target Detection. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Denver, CO, USA, 20–24 July 2016.
24. Pérez-Carabaza, S.; Besada-Portas, E.; Lopez-Orozco, J.A.; Pajares, G. Minimum Time Search in Real-World Scenarios Using Multiple UAVs with Onboard Orientable Cameras. *J. Sens.* **2019**, *2019*, 7673859. [CrossRef]
25. Carazo-Barbero, G.; Besada-Portas, E.; Girón-Sierra, J.M.; López-Orozco, J.A. EA-Based ASV Trajectory Planner for Pollution Detection in Lentic Waters. In Proceedings of the Applications of Evolutionary Computation (EvoApplications), Virtual, 7–9 April 2021.
26. Liu, J.; Wang, W.; Li, X.; Wang, T.; Bai, S.; Wang, Y. Solving A Multi-objective Mission Planning Problem for UAV Swarms with An Improved NSGA-III Algorithm. *Int. J. Comput. Intell. Syst.* **2018**, *11*, 1067–1081. [CrossRef]
27. Bai, H.; Fan, T.; Niu, Y.; Cui, Z. Multi-UAV Cooperative Trajectory Planning Based on Many-Objective Evolutionary Algorithm. *Complex Syst. Model. Simul.* **2022**, *2*, 130–141. [CrossRef]
28. Schoenberg, I.J. *Cardinal Spline Interpolation*; SIAM: Philadelphia, PA, USA, 1973.
29. Zong, X.; Sun, Q.; Yao, D.; Du, W.; Tang, Y. Trajectory planning in 3D dynamic environment with non-cooperative agents via fast marching and Bézier curve. *Cyber-Phys. Syst.* **2019**, *5*, 119–143. [CrossRef]
30. Usenko, V.; von Stumberg, L.; Pangercic, A.; Cremers, D. Real-time trajectory replanning for MAVs using uniform B-splines and a 3D circular buffer. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017.
31. Dubeň, A.; Pěnička, R.; Saska, M. Information Gathering Planning with Hermite Spline Motion Primitives for Aerial Vehicles with Limited Time of Flight. In Proceedings of the Modelling and Simulation for Autonomous Systems, Palermo, Italy, 29–31 October 2019.
32. Carazo-Barbero, G.; Besada-Portas, E.; Risco-Martín, J.L.; López-Orozco, J.A. EA-based ASV Trajectory Planner for Detecting Cyanobacterial Blooms in Freshwater. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Lisbon, Portugal, 15–19 July 2023.
33. Gao, X.G.; Fu, X.W.; Chen, D.Q. A genetic-algorithm-based approach to UAV path planning problem. In Proceedings of the WSEAS International Conference on Simulation, Modelling and Optimization (SMO), Corfu Island, Greece, 17–19 August 2005.
34. Qiongbing, Z.; Lixin, D. A new crossover mechanism for genetic algorithms with variable-length chromosomes for path optimization problems. *Expert Syst. Appl.* **2016**, *60*, 183–189. [CrossRef]

35. Besada-Portas, E.; de la Torre, L.; Moreno, A.; Risco-Martín, J.L. On the performance comparison of multi-objective evolutionary UAV path planners. *Inf. Sci.* **2013**, *238*, 111–125. [CrossRef]
36. Doerr, B.; Qu, Z. A First Runtime Analysis of the NSGA-II on a Multimodal Problem. *IEEE Trans. Evol. Comput.* **2023**, *27*, 1288–1297. [CrossRef]
37. Friedman, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [CrossRef]
38. Skillings, J.H.; Mack, G.A. On the Use of a Friedman-Type Statistic in Balanced and Unbalanced Block Designs. *Technometrics* **1981**, *23*, 171–177. [CrossRef]
39. Nemenyi, P.B. Distribution-Free Multiple Comparisons. Ph.D. Thesis, Princeton University, Princeton, NJ, USA, 1963.
40. Mann, H.B.; Whitney, D.R. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *Ann. Math. Stat.* **1947**, *18*, 50–60. [CrossRef]
41. Guven, B.; Howard, A. A review and classification of the existing models of cyanobacteria. *Prog. Phys. Geogr.* **2006**, *30*, 1–24. [CrossRef]
42. Kromkamp, J.; Walsby, A.E. A computer model of buoyancy and vertical migration in cyanobacteria. *J. Plankton Res.* **1990**, *12*, 161–183. [CrossRef]
43. Visser, P.; Passarge, J.; Mur, L. Modelling vertical migration of the cyanobacterium Microcystis. *Hydrobiologia* **1997**, *349*, 99–109. [CrossRef]
44. Gao, K.; Ai, H. Relationship of growth and photosynthesis with colony size in an edible cyanobacterium, GE-XIAN-MI NOSTOC (Cyanophyceae). *J. Phycol.* **2004**, *40*, 523–526. [CrossRef]
45. Chujo, M.; Li, J.; Datta, T.; Amano, Y.; Machida, M. A Competitive Growth Model for the Simulation of Cyanobacterial Blooms Under Eutrophic Conditions. *Environ. Eng. Sci.* **2021**, *38*, 15–23. [CrossRef]
46. Ducobu, H.; Huisman, J.; Jonker, R.R.; Mur, L.R. Competition between a prochlorophyte and a cyanobacterium under various phosphorus regimes: Comparison with the Droop model. *J. Phycol.* **1998**, *34*, 467–476. [CrossRef]
47. Hipsey, M.; Romero, J.; Antenucci, J.; Hamilton, D. *Computational Aquatic Ecosystem Dynamics Model: CAEDYM v2*; Contract Research Group, Centre for Water Research, University of Western Australia: Nedlands, Australia, 2006; Volume 90.
48. Liu, H.; Zheng, Z.C.; Young, B.; Harris, T.D. Three-dimensional numerical modeling of the cyanobacterium Microcystis transport and its population dynamics in a large freshwater reservoir. *Ecol. Model.* **2019**, *398*, 20–34. [CrossRef]
49. Rousso, B.Z.; Bertone, E.; Stewart, R.; Hamilton, D.P. A systematic literature review of forecasting and predictive models for cyanobacteria blooms in freshwater lakes. *Water Res.* **2020**, *182*, 115959. [CrossRef]
50. Zheng, L.; Hu, B.; Ding, A.Z. A multi-factor data-driven prediction model for cyanobacteria blooms in lakes and reservoirs. *Desalin. Water Treat.* **2020**, *189*, 207–216. [CrossRef]
51. Ostfeld, A.; Tubaltzev, A.; Rom, M.; Kronaveter, L.; Zohary, T.; Gal, G. Coupled Data-Driven Evolutionary Algorithm for Toxic Cyanobacteria (Blue-Green Algae) Forecasting in Lake Kinneret. *J. Water Resour. Plan. Manag.* **2015**, *141*, 04014066. [CrossRef]
52. Zhao, C.; Shao, N.; Yang, S.; Ren, H.; Ge, Y.; Feng, P.; Dong, B.; Zhao, Y. Predicting cyanobacteria bloom occurrence in lakes and reservoirs before blooms occur. *Sci. Total Environ.* **2019**, *670*, 837–848. [CrossRef] [PubMed]
53. Echard, J.S. A Review of Harmful Algal Bloom Prediction Models for Lakes and Reservoirs. Master's Thesis, Utah State University, Logan, UT, USA, 2021.
54. Park, Y.; Lee, H.K.; Shin, J.K.; Chon, K.; Kim, S.; Cho, K.H.; Kim, J.H.; Baek, S.S. A machine learning approach for early warning of cyanobacterial bloom outbreaks in a freshwater reservoir. *J. Environ. Manag.* **2021**, *288*, 112415. [CrossRef]
55. Pyo, J.; Cho, K.H.; Kim, K.; Baek, S.S.; Nam, G.; Park, S. Cyanobacteria cell prediction using interpretable deep learning model with observed, numerical, and sensing data assemblage. *Water Res.* **2021**, *203*, 117483. [CrossRef] [PubMed]
56. Fournier, C.; Fernandez-Fernandez, R.; Cirés, S.; López-Orozco, J.A.; Besada-Portas, E.; Quesada, A. LSTM networks provide efficient cyanobacterial blooms forecasting even with incomplete spatio-temporal data. *Water Res.* **2024**, *267*, 122553. [CrossRef]
57. Lin, S.; Pierson, D.C.; Mesman, J.P. Prediction of algal blooms via data-driven machine learning models: an evaluation using data from a well-monitored mesotrophic lake. *Geosci. Model Dev.* **2023**, *16*, 35–46. [CrossRef]
58. Li, H.; Qin, C.; He, W.; Sun, F.; Du, P. Improved predictive performance of cyanobacterial blooms using a hybrid statistical and deep-learning method. *Environ. Res. Lett.* **2021**, *16*, 124045. [CrossRef]
59. Kutser, T.; Metsamaa, L.; Strömbeck, N.; Vahtmäe, E. Monitoring cyanobacterial blooms by satellite remote sensing. *Estuar. Coast. Shelf Sci.* **2006**, *67*, 303–312. [CrossRef]
60. Reinart, A.; Kutser, T. Comparison of different satellite sensors in detecting cyanobacterial bloom events in the Baltic Sea. *Remote Sens. Environ.* **2006**, *102*, 74–85. [CrossRef]
61. Wynne, T.T.; Stumpf, R.P.; Tomlinson, M.C.; Dyble, J. Characterizing a cyanobacterial bloom in Western Lake Erie using satellite imagery and meteorological data. *Limnol. Oceanogr.* **2010**, *55*, 2025–2036. [CrossRef]
62. Mishra, S.; Stumpf, R.P.; Schaeffer, B.A.; Werdell, P.J.; Loftin, K.A.; Meredith, A. Measurement of Cyanobacterial Bloom Magnitude using Satellite Remote Sensing. *Sci. Rep.* **2019**, *9*, 18310. [CrossRef]

63. Drozd, A.; de Tezanos Pinto, P.; Fernandez, V.; Bazzalo, M.; Bordet, F.; Ibañez, G. Hyperspectral remote sensing monitoring of cyanobacteria blooms in a large South American reservoir: High-and medium-spatial resolution satellite algorithm simulation. *Mar. Freshw. Res.* **2019**, *71*, 593–605. [CrossRef]
64. Nguyen, H.Q.; Ha, N.T.; Pham, T.L. Inland harmful cyanobacterial bloom prediction in the eutrophic Tri An Reservoir using satellite band ratio and machine learning approaches. *Environ. Sci. Pollut. Res.* **2020**, *27*, 9135–9151. [CrossRef]
65. Wynne, T.T.; Stumpf, R.P.; Tomlinson, M.C.; Schwab, D.J.; Watabayashi, G.Y.; Christensen, J.D. Estimating cyanobacterial bloom transport by coupling remotely sensed imagery and a hydrodynamic model. *Ecol. Appl.* **2011**, *21*, 2709–2721. [CrossRef]
66. Overman, C.; Wellsd, S. Modeling Cyanobacteria Vertical Migration. *Water* **2022**, *14*, 953. [CrossRef]
67. Wallace, B.B.; Hamilton, D.P. The effect of variations in irradiance on buoyancy regulation in Microcystis aeruginosa. *Limnol. Oceanogr.* **1999**, *44*, 273–281. [CrossRef]
68. Belov, A.P.; Giles, J.D. Dynamical model of buoyant cyanobacteria. *Hydrobiologia* **1997**, *349*, 87–97. [CrossRef]
69. Overman, C.C.M. Modeling Vertical Migration of Cyanobacteria and Zooplankton. Master's Thesis, Portland State University, Portland, OR, USA, 2019.
70. Bouguer, P. *Essai D'Optique sur la Gradation de la Lumière [Optics Essay on the Attenuation of Light]*; Claude Jombert: Paris, France, 1729. (In French)
71. Xue, K.; Ma, R.; Shen, M.; Wu, J.; Hu, M.; Guo, Y.; Cao, Z.; Xiong, J. Horizontal and vertical migration of cyanobacterial blooms in two eutrophic lakes observed from the GOCI satellite. *Water Res.* **2023**, *240*, 120099. [CrossRef] [PubMed]
72. Navier, C.L.M.H. Mémoire sur les lois du mouvement des fluides. *MéM. L'Acad. Des. Sci. L'Inst. Fr.* **1822**, *6*, 389–440.
73. Brunner, G.W. *HEC-RAS, River Analysis System Hydraulic Reference Manual*; US Army Corps of Engineers: Washington, DC, USA, 2016. Available online: https://www.hec.usace.army.mil/software/hec-ras/documentation/HEC-RAS%205.0%20Reference%20Manual.pdf (accessed on 6 June 2024).
74. Chapra, S.C.; Pelletier, G.J. QUAL2K: A Modeling Framework for Simulating River and Stream Water Quality. 2004. Available online: https://api.semanticscholar.org/CorpusID:126914883 (accessed on 8 July 2024).
75. Imberger, J.; Patterson, J.C. A Dynamic Reservoir Simulation Model—DYRESM: 5. In *Transport Models/Inland & Coastal Waters*; Fischer, H.B., Ed.; Academic Press: Cambridge, MA, USA, 1981.
76. Hipsey, M.R.; Bruce, L.C.; Boon, C.; Busch, B.; Carey, C.C.; Hamilton, D.P.; Hanson, P.C.; Read, J.S.; de Sousa, E.; Weber, M.; et al. A General Lake Model (GLM 3.0) for linking with high-frequency sensor data from the Global Lake Ecological Observatory Network (GLEON). *Geosci. Model Dev.* **2019**, *12*, 473–502. [CrossRef]
77. DHI. MIKE 21 Flow Model FM, Hydrodynamic and Transport Module, Scientific Documentation. 2024. Available online: https://manuals.mikepoweredbydhi.help/latest/Coast_and_Sea/MIKE_21_Flow_FM_Scientific_Doc.pdf (accessed on 6 June 2024).
78. Wells, S.A. *CE-QUAL-W2: A Two-Dimensional, Laterally Averaged, Hydrodynamic and Water Quality Model, Version 4.5*; Portland State University: Portland, OR, USA, 2023. Available online: https://scholar.google.com.hk/citations?view_op=view_citation&hl=en&user=Xx8E1DMAAAAJ&sortby=pubdate&citation_for_view=Xx8E1DMAAAAJ:0KyAp5RtaNEC (accessed on 6 June 2024).
79. Burchard, H.; Bolding, K.; Villarreal, M.R. Three-dimensional modelling of estuarine turbidity maxima in a tidal estuary. *Ocean Dyn.* **2004**, *54*, 250–265. [CrossRef]
80. DHI. MIKE 3 Flow Model FM, Hydrodynamic and Transport Module, Scientific Documentation. 2024. Available online: https://manuals.mikepoweredbydhi.help/latest/Coast_and_Sea/MIKE_3_Flow_FM_Scientific_Doc.pdf (accessed on 6 June 2024).
81. Hodges, B.; Dallimore, C. *Estuary, Lake and Coastal Ocean Model: ELCOM. v2.2 User Manual*. Centre for Water Research, University of Western Australia: Crawley, WA, USA, 2007. Available online: https://oceanexpert.org/downloadFile/7467 (accessed on 6 June 2024).
82. DSI LLC. *EFDC+ Theory, Version 11*; DSI LLC: Edmonds, WA, USA, 2023. Available online: https://downloads.eemodelingsystem.com/technical-papers/2023-10-18-EFDC_GOTM_Whitepaper.pdf (accessed on 6 June 2024).
83. Deltares Systems. Delft3D-FLOW. Simulation of Multi-Dimensional Hydrodynamic Flows and Transport Phenomena, Including Sediments. User Manual. Hydro-Morphodynamics. Deltares Systems. 2023. Available online: https://content.oss.deltares.nl/delft3d4/Delft3D-FLOW_User_Manual.pdf (accessed on 6 June 2024).
84. Kell, G.S. Density, thermal expansivity, and compressibility of liquid water from 0.deg. to 150.deg.. Correlations and tables for atmospheric pressure and saturation reviewed and expressed on 1968 temperature scale. *J. Chem. Eng. Data* **1975**, *20*, 97–105. [CrossRef]
85. Jones, F.E.; Harris, G.L. ITS-90 Density of Water Formulation for Volumetric Standards Calibration. *J. Res. Natl. Inst. Stand. Technol.* **1992**, *97*, 335–340. [CrossRef]
86. Muhetaer, G. Effects of Different Light Intensity on the Growth and Stress Responses of Paeudanabaena Galeata (Cyanobacteria). Ph.D. Thesis, Saitama University, Saitama, Japan, 2020.

87. Jassby, A.D.; Platt, T. Mathematical formulation of the relationship between photosynthesis and light for phytoplankton. *Limnol. Oceanogr.* **1976**, *21*, 540–547. [CrossRef]
88. Peeters, J.C.H.; Eilers, P. The relationship between light intensity and photosynthesis—A simple mathematical model. *Hydrobiol. Bull.* **1978**, *12*, 134–136. [CrossRef]
89. Fadel, A.; Sharaf, N.; Siblini, M.; Slim, K.; Kobaissi, A. A simple modelling approach to simulate the effect of different climate scenarios on toxic cyanobacterial bloom in a eutrophic reservoir. *Ecohydrol. Hydrobiol.* **2019**, *19*, 359–369. [CrossRef]
90. Aparicio-Medrano, E. Physical Aspects Explaining Cyanobacteria Scum Formation in Natural Systems. Ph.D. Thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2014.
91. George, D.G. Wind-induced water movements in the South Basin of Windermere. *Freshw. Biol.* **1981**, *11*, 37–60. [CrossRef]
92. Wang, H.; Zhang, Z.; Liang, D.; du, H.; Pang, Y.; Hu, K.; Wang, J. Separation of wind's influence on harmful cyanobacterial blooms. *Water Res.* **2016**, *98*, 280–292. [CrossRef] [PubMed]
93. Agawin, N.S.R.; Rabouille, S.; Veldhuis, M.J.W.; Servatius, L.; Hol, S.; van Overzee, H.M.J.; Huisman, J. Competition and facilitation between unicellular nitrogen-fixing cyanobacteria and non—nitrogen-fixing phytoplankton species. *Limnol. Oceanogr.* **2007**, *52*, 2233–2248. [CrossRef]
94. Hart, P.E.; Nilsson, N.J.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
95. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]
96. LaValle, S.M. *Planning Algorithms*; Cambridge University Press: Cambridge, UK, 2006.
97. Kavraki, L.; Svestka, P.; Latombe, J.C.; Overmars, M. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. [CrossRef]
98. LaValle, S.M.; Kuffner, J.J. Rapidly-exploring random trees: Progress and prospects. *Algorithmic Comput. Robot.* **2001**, *5*, 293–308.
99. Kalakrishnan, M.; Chitta, S.; Theodorou, E.; Pastor, P.; Schaal, S. STOMP: Stochastic trajectory optimization for motion planning. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011.
100. Xu, W.; Wei, J.; Dolan, J.M.; Zhao, H.; Zha, H. A real-time motion planner with trajectory optimization for autonomous vehicles. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011.
101. Wang, Z.; Zhou, X.; Xu, C.; Gao, F. Geometrically Constrained Trajectory Optimization for Multicopters. *IEEE Trans. Robot.* **2022**, *38*, 3259–3278. [CrossRef]
102. Manyam, S.G.; Casbeer, D.W.; Weintraub, I.E.; Taylor, C. Trajectory Optimization For Rendezvous Planning Using Quadratic Bézier Curves. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021.
103. Tang, L.; Wang, H.; Liu, Z.; Wang, Y. A real-time quadrotor trajectory planning framework based on B-spline and nonuniform kinodynamic search. *J. Field Robot.* **2021**, *38*, 452–475. [CrossRef]
104. Neto, A.A.; Campos, M.F.M. A path planning algorithm for UAVs with limited climb angle. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, MO, USA, 11–15 October 2009.
105. Rösmann, C.; Feiten, W.; Wösch, T.; Hoffmann, F.; Bertram, T. Efficient trajectory optimization using a sparse model. In Proceedings of the European Conference on Mobile Robots, Barcelona, Spain, 25–27 September 2013.
106. Kim, J.J.; Lee, J.J. Trajectory Optimization with Particle Swarm Optimization for Manipulator Motion Planning. *IEEE Trans. Ind. Inform.* **2015**, *11*, 620–631. [CrossRef]
107. Yu, Z.; Si, Z.; Li, X.; Wang, D.; Song, H. A Novel Hybrid Particle Swarm Optimization Algorithm for Path Planning of UAVs. *IEEE Internet Things J.* **2022**, *9*, 22547–22558. [CrossRef]
108. Xiong, C.; Zhou, H.; Lu, D.; Zeng, Z.; Lian, L.; Yu, C. Rapidly-Exploring Adaptive Sampling Tree*: A Sample-Based Path-Planning Algorithm for Unmanned Marine Vehicles Information Gathering in Variable Ocean Environments. *Sensors* **2020**, *20*, 2515. [CrossRef]
109. Kathen, M.J.T.; Flores, I.J.; Reina, D.G.; Córdoba, A.T. Autonomous monitoring system for water resources based on PSO and Gaussian process. In Proceedings of the IEEE Congress on Evolutionary Computation, Kraków, Poland, 28 June–1 July 2021.
110. Kathen, M.J.T.; Flores, I.J.; Reina, D.G. An informative path planner for a swarm of ASVs based on an enhanced PSO with Gaussian surrogate model components intended for water monitoring applications. *Electronics* **2021**, *10*, 1605. [CrossRef]
111. Lazarowska, A. Ship's Trajectory Planning for Collision Avoidance at Sea Based on Ant Colony Optimisation. *J. Navig.* **2015**, *68*, 291–307. [CrossRef]
112. Cekmez, U.; Ozsiginan, M.; Sahingoz, O.K. Multi colony ant optimization for UAV path planning with obstacle avoidance. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016.
113. Yang, H.; Qi, J.; Miao, Y.; Sun, H.; Li, J. A New Robot Navigation Algorithm Based on a Double-Layer Ant Algorithm and Trajectory Optimization. *IEEE Trans. Ind. Electron.* **2019**, *66*, 8557–8566. [CrossRef]
114. Xiong, C.; Chen, D.; Lu, D.; Zeng, Z.; Lian, L. Path planning of multiple autonomous marine vehicles for adaptive sampling using Voronoi-based ant colony optimization. *Robot. Auton. Syst.* **2019**, *115*, 90–103. [CrossRef]

115. Xia, G.; Han, Z.; Zhao, B.; Liu, C.; Wang, X. Global Path Planning for Unmanned Surface Vehicle Based on Improved Quantum Ant Colony Algorithm. *Math. Probl. Eng.* **2019**, *2019*, 2902170. [CrossRef]
116. Meng, H.; Xin, G. UAV route planning based on the genetic simulated annealing algorithm. In Proceedings of the IEEE International Conference on Mechatronics and Automation, Xi'an, China, 4–7 August 2010.
117. Behnck, L.P.; Doering, D.; Pereira, C.E.; Rettberg, A. A Modified Simulated Annealing Algorithm for SUAVs Path Planning. In Proceedings of the IFAC Conference on Embedded Systems, Computer Intelligence and Telematics (CESCIT), Maribor, Slovenia, 22–24 June 2015.
118. Balan, K.; Luo, C. Optimal Trajectory Planning for Multiple Waypoint Path Planning using Tabu Search. In Proceedings of the IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 8–10 November 2018.
119. Abdulsaheb, J.A.; Kadhim, D.J. Classical and Heuristic Approaches for Mobile Robot Path Planning: A Survey. *Robotics* **2023**, *12*, 93. [CrossRef]
120. Tan, C.S.; Mohd-Mokhtar, R.; Arshad, M.R. A Comprehensive Review of Coverage Path Planning in Robotics Using Classical and Heuristic Algorithms. *IEEE Access* **2021**, *9*, 119310–119342. [CrossRef]
121. Zhou, Z.; Duan, H.; Li, P.; Di, B. Chaotic differential evolution approach for 3D trajectory planning of unmanned aerial vehicle. In Proceedings of the IEEE International Conference on Control and Automation (ICCA), Hangzhou, China, 12–14 June 2013.
122. MahmoudZadeh, S.; Powers, D.M.W.; Yazdani, A.M.; Sammut, K.; Atyabi, A. Efficient AUV Path Planning in Time-Variant Underwater Environment Using Differential Evolution Algorithm. *J. Mar. Sci. Appl.* **2018**, *17*, 585–591. [CrossRef]
123. Yu, X.; Li, C.; Zhou, J. A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios. *Knowl.-Based Syst.* **2020**, *204*, 106209. [CrossRef]
124. Ahmed, F.; Deb, K. Multi-objective path planning using spline representation. In Proceedings of the IEEE International Conference on Robotics and Biomimetics, Phuket, Thailand, 7–11 December 2011.
125. Peralta, F.; Reina, D.G.; Toral, S. Water quality online modeling using multi-objective and multi-agent Bayesian Optimization with region partitioning. *Mechatronics* **2023**, *91*, 102953. [CrossRef]
126. Arzamendia, M.; Gregor, D.; Reina, D.G.; Toral, S.L.; Gregor, R. Evolutionary Path Planning of an Autonomous Surface Vehicle for Water Quality Monitoring. In Proceedings of the International Conference on Developments in e-Systems Engineering, Liverpool, UK, 31August–2 September 2016.
127. Arzamendia, M.; Gregor, D.; Reina, D.; Toral, S. An evolutionary approach to constrained path planning of an autonomous surface vehicle for maximizing the covered area of Ypacarai Lake. *Soft Comput.* **2019**, *23*, 1723–1734. [CrossRef]
128. Paranjape, A.A.; Meier, K.C.; Shi, X.; Chung, S.J.; Hutchinson, S. Motion primitives and 3D path planning for fast flight through a forest. *Int. J. Robot. Res.* **2015**, *34*, 357–377. [CrossRef]
129. Li, Y.; Eslamiat, H.; Wang, N.; Zhao, Z.; Sanyal, A.K.; Qiu, Q. Autonomous waypoints planning and trajectory generation for multi-rotor UAVs. In Proceedings of the Workshop on Design Automation for CPS and IoT (DESTION), Montreal, QC, Canada, 15 April 2019.
130. Kano, H.; Fujioka, H. B-Spline Trajectory Planning with Curvature Constraint. In Proceedings of the 2018 Annual American Control Conference (ACC), Milwaukee, WI, USA, 27–29 June 2018,
131. González-Calvin, A.; García-Perez, L.; Risco-Martín, J.L.; Besada-Portas, E. Simulation, Optimization and Control of Trajectories of ASVs Performing Hacbs Monitoring Missions in Lentic Waters. In Proceedings of the Winter Simulation Conference (WSC), San Antonio, TX, USA, 10–13 December 2023.
132. Webpage. 1998. Available online: https://www.comsol.com/ (accessed on 19 October 2024).
133. Webpage. 1994. Available online: https://www.mathworks.com/ (accessed on 19 October 2024).
134. Community, B.O. *Blender—A 3D Modelling and Rendering Package*; Blender Foundation: Amsterdam, The Netherlands, 2018.
135. Webpage. 1998. Available online: https://doc.comsol.com/ (accessed on 19 June 2024).
136. Soontiens, N.; Binding, C.; Fortin, V.; Mackay, M.; Rao, Y.R. Algal bloom transport in Lake Erie using remote sensing and hydrodynamic modelling: Sensitivity to buoyancy velocity and initial vertical distribution. *J. Great Lakes Res.* **2019**, *45*, 556–572. [CrossRef]
137. Ravankar, A.; Ravankar, A.; Kobayashi, Y.; Hoshino, Y.; Peng, C. Path Smoothing Techniques in Robot Navigation: State-of-the-Art, Current and Future Challenges. *Sensors* **2018**, *18*, 3170. [CrossRef]
138. Jones, M.; Baerentzen, J.; Sramek, M. 3D distance fields: A survey of techniques and applications. *IEEE Trans. Vis. Comput. Graph.* **2006**, *12*, 581–599. [CrossRef]
139. Srinivas, N.; Deb, K. Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evol. Comput.* **1994**, *2*, 221–248. [CrossRef]
140. Jones, D.F.; Florentino, H.O. Multi-Objective Optimization: Methods and Applications. In *The Palgrave Handbook of Operations Research*; Springer International Publishing: Berlin/Heidelberg, Germany, 2022; pp. 181–207. [CrossRef]

141. Bader, J.; Deb, K.; Zitzler, E. Faster Hypervolume-Based Search Using Monte Carlo Sampling. In *Proceedings of the Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems*; Ehrgott, M., Naujoks, B., Stewart, T.J., Wallenius, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2010.
142. Abdollahzadeh, B.; Gharehchopogh, F.S. A multi-objective optimization algorithm for feature selection problems. *Eng. Comput.* **2022**, *38*, 1845–1863. [CrossRef]
143. Gaspar-Cunha, A.; Costa, P.; Monaco, F.; Delbem, A. Many-Objectives Optimization: A Machine Learning Approach for Reducing the Number of Objectives. *Math. Comput. Appl.* **2023**, *28*, 17. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.