Journal of *Marine Science and Engineering*

MDPI

Article

Path Planning for Underwater Information Gathering Based on Genetic Algorithms and Data Stochastic Models

Matteo Bresciani [1,2,*], Francesco Ruscio [1,2], Simone Tani [1,2], Giovanni Peralta [1,2], Andrea Timperi [1,3], Eric Guerrero-Font [4], Francisco Bonin-Font [4], Andrea Caiti [1,2,3] and Riccardo Costanzi [1,2,3]

1 Dipartimento di Ingegneria dell'Informazione, Università di Pisa, 56122 Pisa, Italy; francesco.ruscio@phd.unipi.it (F.R.); simone.tani@ing.unipi.it (S.T.); giovanni.peralta@phd.unipi.it (G.P.); andrea.timperi@ing.unipi.it (A.T.); andrea.caiti@unipi.it (A.C.); riccardo.costanzi@unipi.it (R.C.)
2 Interuniversity Center of Integrated Systems for the Marine Environment (ISME), 16145 Genova, Italy
3 Centro di Ricerca "E. Piaggio", Università di Pisa, 56122 Pisa, Italy
4 Systems Robotics and Vision Group, University of the Balearic Islands (UIB), 07122 Palma de Mallorca, Spain; ericguerrerofont@gmail.com (E.G.-F.); francisco.bonin@uib.es (F.B.-F.)
* Correspondence: matteo.bresciani@phd.unipi.it; Tel.: +39-34-7008-5307

Abstract: Recent technological developments have paved the way to the employment of Autonomous Underwater Vehicles (AUVs) for monitoring and exploration activities of marine environments. Traditionally, in information gathering scenarios for monitoring purposes, AUVs follow predefined paths that are not efficient in terms of information content and energy consumption. Informative Path Planning (IPP) represents a valid alternative, defining the path that maximises the gathered information. This work proposes a Genetic Path Planner (GPP), which consists in an IPP strategy based on a Genetic Algorithm, with the aim of generating a path that simultaneously maximises the information gathered and the coverage of the inspected area. The proposed approach has been tested offline for monitoring and inspection applications of Posidonia Oceanica (PO) in three different geographical areas. The a priori knowledge about the presence of PO, in probabilistic terms, has been modelled utilising a Gaussian Process (GP), trained on real marine data. The GP estimate has then been exploited to retrieve an information content of each position in the areas of interest. A comparison with other two IPP approaches has been carried out to assess the performance of the proposed algorithm.

Keywords: genetic algorithm; path planning; Gaussian process; AUVs; optimal sampling; Posidonia Oceanica; maritime monitoring and inspection task

Citation: Bresciani, M.; Ruscio, F.; Tani, S.; Peralta, G.; Timperi, A.; Guerrero-Font, E.; Bonin-Font, F.; Caiti, A.; Costanzi, R. Path Planning for Underwater Information Gathering Based on Genetic Algorithms and Data Stochastic Models. *J. Mar. Sci. Eng.* **2021**, *9*, 1183. https://doi.org/10.3390/jmse9111183

Academic Editors: Marco Cococcioni and Pierre Lermusiaux

Received: 30 September 2021
Accepted: 23 October 2021
Published: 27 October 2021

1. Introduction

In the last years, sea protection and monitoring have been central topics for the underwater community. However, because of the difficulties and risks involved in underwater missions, most part of the marine environment still remains unexplored. Thanks to recent developments in technology and automation, *Autonomous Underwater Vehicles* (AUVs) can now play a key role in underwater exploration. Indeed, AUVs represent effective tools for very different purposes, such as glacier inspection [1], monitoring of marine habitats [2] and climate changes effects [3], and for seafloor mapping [4]. In particular, the exploitation of AUVs for monitoring purposes makes data collection faster and certainly safer [5].

Traditionally, in *Information Gathering* (IG) operations, where underwater exploration entails environmental data collection, AUVs follow predefined geometric routes, typically with lawnmower patterns. However, this approach is not efficient as the AUV moves along a path which can lead to areas of low interest, wasting time and energy. On the contrary, optimal path planning algorithms generate paths that optimise an objective function representing specific criteria, such as energy consumption, survey duration or distance travelled [6]. Moreover, by benefiting from the measurements coming from the

sensors on-board the vehicle, the planner can employ online replanning strategies to avoid collisions with potential obstacles along the way. Among the various optimal planning strategies, *Informative Path Planning* (IPP) focuses on increasing the knowledge about environmental variables of interest while respecting some constraints on the robot and path features [7].

A model of the environmental variable of interest, based on the *a priori* knowledge about the area to be inspected, is crucial to achieve efficient informative paths and a successful completion of missions. An efficient way for modelling the environment is applying a *Gaussian Process* (GP) [8]. GPs are powerful nonparametric techniques that can handle a large variety of problems, having the ability to learn spatial correlation with noisy measured data [9]. The key feature of GPs for IG scenarios is their capability to handle both data uncertainty and data incompleteness effectively. Thus, they allow to perform dense estimation of environmental variables for coverage-based exploration purposes, giving a clear idea of which are the places that need to be inspected to reduce model uncertainty. However, the weakness of these techniques is related to their high computational time. Some works overcome this issue by proposing the use of a sparse version of the GP [10,11], or local maps fusion using Bayesian Committee Machine (BCM) [12], to reduce computational costs and enable online execution.

The estimate of the variable of interest provided by the GP, in terms of mean and covariance, can be exploited to quantitatively describe the informativeness associated with each position within the target area, using the so called information function. In particular, higher values on the covariance map indicate points where the GP estimates are more uncertain and, thus, more informative: collecting data in those areas means reducing the uncertainty of the model, and thus increasing knowledge about the environmental variable of interest. While the authors of [13] propose the use of an interpolated version of the GP model predicted variance, most of the methods use either *Differential Entropy* (DE) or *Mutual Information* (MI) as information function. Using DE results in lower computation time and higher uncertainty reduction than using MI, when computed from GP predictions of stationary setups [14]. Despite the submodularity property of MI [15], the computational time of such information function results prohibitive in many applications.

The IPP approaches used for IG purposes can be categorised into two groups: myopic and non-myopic. Myopic strategies only plan one step forward and they work following greedy heuristics [16]. Conversely, non-myopic strategies look several steps ahead, providing paths that may result better on the long run, and are usually based on graphs [10,17,18], random trees [14,19–22], or evolutionary algorithms [23–25]. In the framework of non-myopic strategies, a Rapidly-exploring Information Gathering (RIG) tree [20] approach was presented by Hollinger et al. in 2014, where the vertices in RIG-tree represent a tuple of location, cost, and information. In 2019, Viseras et al. improved the RIG approach in [14], proposing a two-stage planning process. In the first step, the planner finds the target position providing the highest information spot under a budget constraint through Rapidly-exploring Random Tree (RRT) [26]. In the second step, it provides an optimised path to the goal. Among the category of the evolutionary algorithms are the *Genetic Algorithms* (GA), which are commonly employed in classical cost-optimisation planning. GA are inspired by genetics and survival concepts [27,28], ensuring flexibility in multiple scenarios, such as terrestrial [29], aerial [30], and marine [31,32]. However, the above applications of GAs suffer from some problems, such as a high computational cost, the need to discretise the environment [33] or the necessity to have a fixed end point [34]. Furthermore, GAs are rarely used in IPP strategies for IG purposes in underwater environments.

The innovation introduced by the work presented in this paper consists in the application of a GA-based IPP algorithm, renamed *Genetic Path Planner* (GPP), with the aim of generating a path that simultaneously maximises the information gathered and the coverage of the inspected area. In particular, the analysed scenario is the monitoring of *Posidonia Oceanica* (PO), which is an endemic low-growing Mediterranean sea grass with a high ecological value. Indeed, it is essential for the stability of coastal ecosystems, currents

and waves energy attenuation, and carbon absorption, and it is a source of food and refuge for numerous animal species. Many studies show how human activity is causing the continuous decline of this species [35], and how monitoring the ecosystems that they form is essential to plan strategies for its control and maintenance. The authors' final goal is to employ an AUV capable of finding and replanning autonomously the best path to carry out PO monitoring activities. In this paper, a GP trained on data describing the presence of PO, in probabilistic terms, has been utilised to create a model of the PO meadow extension and shape in the area of interest, through Equations (A2) and (A3) [11] (details in Appendix A.1). To obtain the dataset, used for the training of the GP, a collection of images, acquired by the AUV in previous missions over the target area, were segmented using a pre-trained *Convolutional Neural Network* (CNN) [36]. The uncertainty of the estimate provided by the GP has then been used as input of the GPP algorithm to compute the informativeness of a given position within the inspected area. As a further step towards the final goal, the new data collected by the AUV along the planned path will be exploited to retrain the GP, and hence update the environmental model. This way, the GPP method will be able to re-plan a path according to the updated covariance map of the GP prediction.

Within this work, the proposed GPP algorithm has been tested offline using GPs trained on real marine data relative to three different geographical areas covered by PO. Moreover, a comparison with other two IPP strategies has been carried out.

The manuscript is organised as follows. The proposed path planner, together with the alternative planning strategies and the metrics used for the comparative analysis, are presented in Section 2. Results analysis and comparative evaluation are described in Section 3. Section 4 reports a discussion regarding the results reported in the previous Section. Finally, paper conclusions and possible future works are presented in Section 5.

2. Materials and Methods

2.1. Genetic Path Planner

This section describes the architecture of the genetic planner designed for autonomous IG applications in marine environment. The main purpose of the proposed GPP is to find the optimal path to be followed by the AUV by maximising a performance function, called 'utility function', which involves the path informativeness and other variables assessing the path performance. Within the specific application, the planner has to generate collision-free trajectories, compliant with the AUV manoeuvrability, to carry out PO inspection and monitoring tasks. A GP is employed to model the presence of PO and the resulting uncertainty in its estimate is used to compute the information acquired along the path. Typically, high uncertainties will be obtained in borders between PO meadow and sand, and zones not yet explored by the robot. Therefore, the planner should focus on those uncertain areas, where the robot can acquire data to increase the knowledge about the environment. In designing the GPP, the following assumptions were made:

1. **The presence of PO is considered time-invariant during all the planning task**. The assumption is reasonable as the typical endurance of an AUV is limited to few hours while the variations in PO meadows usually occur over several months.
2. **The boundaries of the area to be explored and the presence of obstacles are considered a priori known**. The assumption results appropriate as PO monitoring is carried out periodically in the same geographical areas to assess its spread and health state.
3. **A two-dimensional scenario is considered** to represent a path using North and East coordinates referred to the NED frame. The vertical coordinate is beyond the aim of the proposed planner. The considered application of PO monitoring is usually based on optical cameras. The vertical coordinate is thus usually controlled to ensure the AUV to maintain a desired altitude with respect to the sea bottom to guarantee good acquisition conditions.

According to assumptions 2 and 3, $X_{free} \subset \mathbb{R}^2$ represents the space free from obstacles of the area in which the vehicle can move, expressed in North and East coordinates. The AUV paths generated by the GPP strategy are described using chromosomes and genes, essential elements of general genetic algorithms (more details in Appendix A.2). In particular, a chromosome represents a path, and it is composed of m genes, each of which describes a step of the path. Within this work, a gene has been characterised as a tuple $\eta = (\alpha, d)$, where α represents the increment of the AUV heading angle and d the length of the step of the path. The α angle follows a Gaussian distribution $\alpha \sim \mathcal{N}(0, \sigma_\alpha^2)$ with zero mean and with standard deviation σ_α, while d is sampled over an uniform distribution with a range $[d_{min}, d_{max}]$ scaling with the size of the area to inspect. Figure 1 shows an example of a chromosome along with its genes.

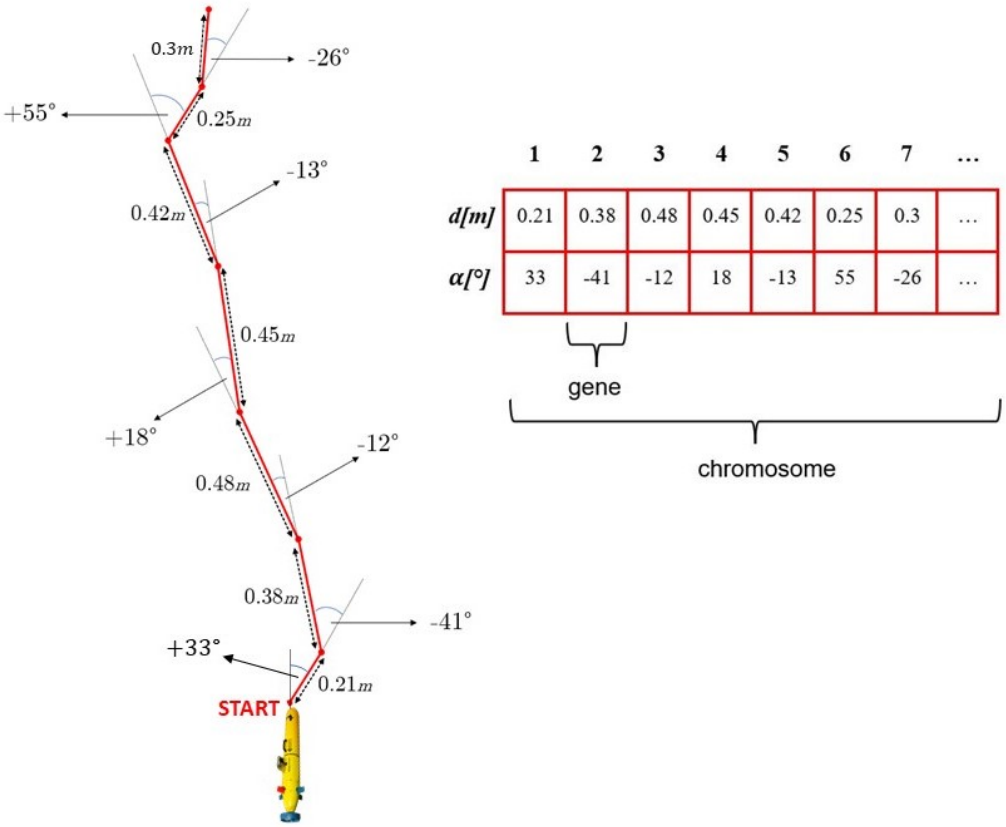


Figure 1. Example of a GPP chromosome.

Note that paths composing of genes characterised by small values for the angle α better approximate smooth paths, which are crucial to reduce energy consumption and survey duration. Indeed, they can be performed by the AUV without stopping. This is related to the AUV manoeuvrability, which mostly depends on its speed and shape. Typically, AUVs can be categorised, according to their geometry, such as high-manoeuvrable or torpedo-shape. On the one hand, the high-manoeuvrable AUVs are usually stocky and provided with several motors to move and rotate in almost all directions [37,38]. This category of underwater vehicles is able to perform rapidly big changes in direction by rotating on the spot, but their manoeuvrability reduces with the increase of their speed. On the other hand, torpedo-shaped vehicles are slender and usually equipped with a propeller on the stern and a rudder, or lateral thrusters, to make turns [39,40]. These vehicles can achieve higher speed, but the steering capabilities of those equipped with a rudder considerably decrease when working at low speed, while for those provided with lateral thrusters the manoeuvrability reduction occurs at high speed. According to the

aforementioned considerations, the values of the standard deviation σ_α and the range of the uniform distribution of d are chosen considering the specific AUV employed for the missions, so to obtain smoother paths for the robot.

Figure 2 shows the flowchart of the designed GPP algorithm. The "Parameters Setup" block provides the parameters used to initialise the process, including initial population size n, chromosome length m, and all those parameters affecting the evolutionary operations of the algorithm. The initial population of chromosomes is generated in the "Global Initialisation" phase depicted in the green box. On the other hand, the blue box reports the iterative part of the algorithm, in which the genetic operations take place. In order to execute the planner on board of an AUV, without issues on the global computation resources, the duration of the iterative phase is limited to t_{max}. In the following, more details will be given for all the phases of the algorithm, except for the "Ranking" phase, which is a sorting and removing operation to maintain only the k best chromosomes in the population. Details about the Consistency check operation, which ensures path generation within X_{free}, are also provided in a dedicated subsection.

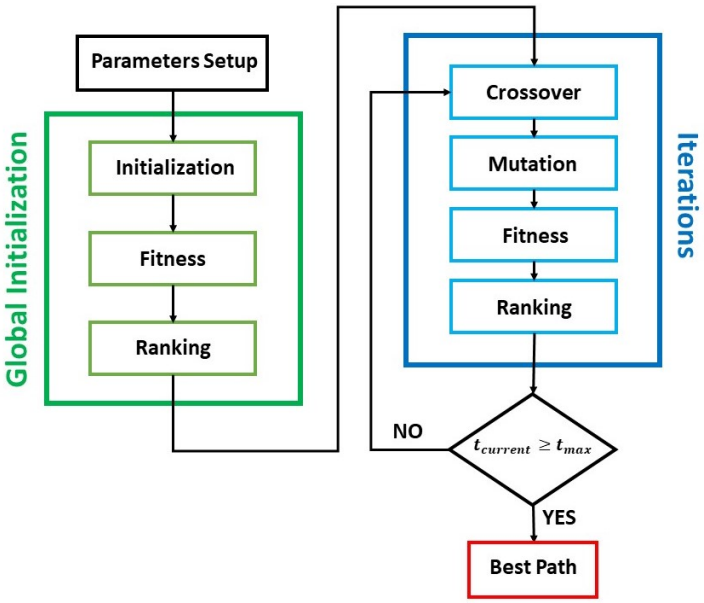


Figure 2. Flowchart of the Genetic Path Planner.

2.1.1. Consistency Check

It is necessary to ensure that each generated path is within the mission area and does not intersect possible obstacles. Therefore, a consistency check is made every time a new gene is created or the content (α or d) of an existing one changes. This guarantees that all the locations x of the nodes composing a path belong to X_{free}. Thanks to the *a priori* knowledge about obstacles and boundaries of the mission area, such verification results to be a trivial task. If the verification fails, meaning that the position x resulting from the new generated gene is either outside the target area or inside the obstacle area, then the gene is deleted and replaced with a new one. However, due to the limitation on the standard deviation σ_α, imposed to obtain routes compliant with the AUV manoeuvrability capabilities, it might happen that no consistent genes could be generated. A similar scenario could occur near the borders of the target and obstacle areas, especially in the corners, where any steering angle α sampled from the Gaussian distribution will result to be insufficient to generate points in X_{free}, with a consequent impossibility to carry on the algorithm. The strategy adopted by the authors in such circumstances is structured as follows. First, the algorithm attempts to generate a new consistent gene sampling the elements of its tuple from the respective distributions, for a limited number of trials C_g. If this first attempt fails, then the new gene is generated adopting an uniform distribution with range $[0°, 360°)$

for the α angle. That is done with the aim of straying from the borders of the target and obstacle areas, thus increasing the chances to obtain a consistent gene. In the case that also this approach does not provide useful results, after C_u attempts, the last resort is to delete the last C_n nodes (genes) of the path, if any, and replace them with new ones. Deleting the nodes allows to exit from the condition which generates inconsistent genes, but does not guarantee that such condition will not recur. Moreover, the proposed strategy could sacrifice the conformity of the route to the manoeuvrability of the AUV in order to always obtain a viable path and avoid algorithm interruption, even for the aforementioned critical scenarios.

2.1.2. Global Initialisation

The "Global Initialisation" phase consists in the generation of the initial population of chromosomes and the subsequent best chromosomes selection. The initial population G is composed of n chromosomes defined as P_i, with $i \in [1, n]$, so it can be described as $G = [P_1, \ldots, P_n]$. Each chromosome P_i is composed of m genes represented by η_{ij}, with $j \in [1, m]$. Therefore, a chromosome P_i can be expressed as $P_i = [\eta_{i1}, \ldots, \eta_{im}]$. The distance d_{ij} and the angle α_{ij} composing the gene η_{ij} are randomly sampled according to their respective distributions. Each gene is, thus, univocally associated to a location x_{ij} in the target area, defined as:

$$x_{ij} = \begin{bmatrix} x_{north_{ij}} \\ x_{east_{ij}} \end{bmatrix} = \begin{cases} x_{north_0} + \sum_{j=1}^{m} d_{ij} cos(\alpha_{ij}) \\ x_{east_0} + \sum_{j=1}^{m} d_{ij} sin(\alpha_{ij}) \end{cases} \quad (1)$$

where i is the index of the considered chromosome, while x_{north_0} and x_{east_0} are input parameters for the algorithm representing the North and East coordinates of the origin point of the path, respectively. Furthermore, as explained in Section 2.1.1, it is always verified that each x_{ij} satisfies the consistency check, and thus it belongs to X_{free} . Each gene has an information content that is retrieved from the covariance map of the GP estimate according to the corresponding position $x_{ij} \in X_{free}$. In the "Fitness" phase, the information content of each gene of a chromosome is exploited as input of an objective function to quantitatively assess the utility of the path (chromosome). More details about the specific functions employed in this work during the "Fitness" phase can be found in Section 2.1.5. Finally, the "Ranking" phase orders the initial population according to the utility value obtained in the "Fitness" phase and then selects the first k chromosomes. Such k chromosomes are used as starting population of the following iterative part of the algorithm. It is worth highlighting that n, m, and k are design parameters and can be chosen according to the performance of the computer on which the algorithm is running. In fact, higher values of these parameters, although advantageous for providing variety to the population, will result in a higher computational cost of the algorithm, and so into a smaller number of iterations for a fixed value of t_{max}.

2.1.3. Crossover

Inspired by the classical Boolean approaches [27,41], crossover strategies are employed to mix the genetic content of pairs of chromosomes. Their goal is to generate offspring whose utility could result to be better than the one associated to the parents. Within this work, four crossover strategies were evaluated (details in Appendix A.3), although only one of these is used during the "Crossover" phase of the GPP. Moreover, in each of these approaches prior to crossover operation, the w best chromosomes, in terms of utility value, are duplicated and queued to the current chromosomes population. This way, the best w chromosomes are preserved for future generations and used to generate offspring as well. Note that w may vary between 2 and 3 depending on whether the population size k after the 'Ranking' phase is even or odd, respectively. This is done to maintain an even number of chromosomes on which the following genetic operations are applied, as all the implemented crossover strategies work with couples of chromosomes.

2.1.4. Mutation

The mutation helps to explore the entire X_{free}, maintaining population diversity [41] and avoiding local minima. For a binary representation of a gene, a simple mutation consists in inverting each gene value with a small probability, changing 0 to 1, and vice versa [41]. Conversely, in the GPP algorithm proposed here, genes are represented by a tuple of value $\eta = (\alpha, d)$. Thus, the mutation phase selects the mutating genes by generating a binomial distributed random array with probability q and dimension m. The resulting array, composed of 0 and 1 elements, can be used as a mask to select the mutating genes of the chromosome, wherever the value of the random array is 1. Moreover, just a random subset of the chromosomes will mutate at each iteration according to a binomial distributed random variable with probability z. Note that the probability q is referenced as *gene mutation rate* in the following, while the probability z as *chromosome mutation rate* , and both parameters are defined by the user in the "Parameters Setup" block. Once a gene is selected for the mutation, its values α and d are discarded and sampled once again from their respective distributions. This way, the genetic content of the gene will change, thus the path described by the mutating chromosome will be modified as well. Figure 3 describes a mutation operation with a gene mutation rate q of 0.25, in which the mutating genes are depicted in yellow. For the sake of image clarity, the mutation operation is shown here with only 8 genes containing only heading α information.

Figure 3. Example of mutation operation.

2.1.5. Fitness

The fitness operation consists in assigning a performance value to each chromosome through the computation of the utility function, which involves as contributions the informativeness of the path and a variable assessing the coverage of the area to inspect. The informative content of a chromosome P_i representing a path is evaluated using an information function $I(P_i)$, which utilises as input the covariance map obtained by the GP estimate. In particular, the information function employed within this work is an averaged version of the DE function $h(x)$, which depends only on the variance of the GP evaluated in the generic point x within the target area. Considering each point in the path as a test location x^*, the predicted distribution over it is given by $f_* \sim \mathcal{N}(\bar{f}_*, \mathbb{V}[f_*])$ and the corresponding DE is [42]

$$h(x^*) = \frac{1}{2} log(2\pi e \mathbb{V}[f_*]) \tag{2}$$

where $\mathbb{V}[f_*]$ is the variance obtained from the GP estimate in the test location x^*, defined as in Appendix A.3. The information function $I(P_i)$ utilises Equation (2) to evaluate how much informative a specific path is, by averaging the DE, as follows:

$$I(P_i) = \frac{\sum_{x_{ij} \in P_i} h(x_{ij})}{m} \tag{3}$$

where P_i is a specific chromosome, m is the number of genes in a chromosome and j is the gene's index.

The information value obtained by $I(P_i)$ is exploited as input of the utility function $U(P_i)$, together with the distance between the initial and the final points d_{o2e} of the path. In fact, d_{o2e} has been utilised as indicator of the extension, over the area of interest, of

the paths generated by the GPP. Thus, the formulation of the utility function can then be formalised as:

$$U(P_i) = g(I(P_i), d_{o2e}, \beta) = \beta I(P_i) + (1-\beta)\frac{d_{o2e}}{c_d} \quad (4)$$

where c_d is a scaling factor used to normalise the distance with respect to the maximum reachable distance for the specific area under inspection and $\beta \in [0,1]$ is a parameter which tunes the trade-off between informativeness and coverage.

2.2. Comparative Planners

In order to assess the performance of the proposed GPP algorithm, a comparative analysis with two other IPP strategies has been carried out. In particular, the two planners used as benchmark are as follows.

1. **Sampling-Based planner (S-B)**
   This planner corresponds to the implementation of the planning method proposed in [14], which is considered, to the extent of the authors' knowledge, as one of the main state-of-the-art IPP approaches for IG operations. The planner is composed of two phases: The first one is called *Station Searching* and applies a RRT algorithm to find the most informative point within the target area, named as *Station*. In the second phase, called *Informative Path Planner*, the planner searches for the optimal path according to a trade-off between the DE of path and its length. With this aim, a RRT* algorithm is utilised to find the best path which connects the start point to the *Station*, if any, respecting a budget constraint on the travelled distance. In case the RRT* algorithm fails to find a feasible solution, the path obtained during the *Station Searching* phase is used.

2. **Random Planner**
   A random planner has been evaluated to assess whether the genetic operations in the iterative part of the GPP algorithm provide an improvement on the performance of the solutions. Otherwise, they would result to be just time consuming operations which can be avoided. Therefore, this planning strategy consists in an extended global initialisation phase, identical to the one implemented in the GPP algorithm, which generates random chromosomes for all the available planning time. Afterwards, using the fitness and ranking operators, the best path among all the generated ones is selected. Although a similar effect on the chromosomes generation could be achieved using high mutation probabilities z and q in the GPP approach, the authors decided to implement the random planner to not perform any kind of optimisation phase in the algorithm. This allows to evaluate the contribution of the optimisation performed by the genetic operations in the GPP algorithm.

2.3. Metrics

To quantitatively compare the proposed GPP method with ones presented in Section 2.2, some metrics are introduced. In particular, the three metrics exploited within this work are meant to evaluate the informativeness of the path and its capacity to cover a wider portion of the target area.

- **Distance Origin to End (02E)**
  This metric coincides with the variable adopted as contribution in the GPP utility function, indicated as d_{o2e}. It specifies how far the final point is from the starting one, providing a sense of how spread the path is over the target area.

- **Path Length (PL)**
  PL is the total length of a path, computed by adding all the distances between consecutive nodes. It is an indicator of how extended the path is, which has to be considered together with the O2E to avoid misleading conclusions. In particular, in those cases where the O2E results to be small, a high value of PL could mean an extended path

and, thus, a wider coverage of the area to inspect. On the contrary, for the same O2E value, a small value of PL entails a short path.

- **Mean Entropy (ME)**
  ME calculates the line integral of the DE of the points along the whole path. This is done refining Equation (3) and sampling the path with a resolution r. The ME calculation is defined as

$$ME = \frac{\int_P h(x)dx}{PL/r}[bits] \quad (5)$$

  where P is the evaluated path and PL represents the path length. In this work, the resolution of the path sampling r is set to $0.2m$. Note that, generally r can differs from d_{min}, and it is recommendable to always have $r < d_{min}$ to increase the accuracy in the computation of the metric ME. Moreover, it is important to highlight that the computation of the metric ME is performed offline. Thus, r can be set according to the desired accuracy of the ME metric without influencing the computational cost of the planners. Therefore, the authors decided to maintain r equal for all the target areas.

Considering IPP strategies in the IG scenario, the ME represents a decisive indicator about the planner performance, as it provides the amount of information acquired along a path. However, this metric alone could not be sufficient to assess the best path as the goal of the planner proposed in this work is to maximise both information and coverage of the inspected area. Therefore, PL and O2E are utilised to support the evaluation and are used as area coverage indexes. Higher PL and O2E values correspond to a wider coverage of the target area.

3. Results

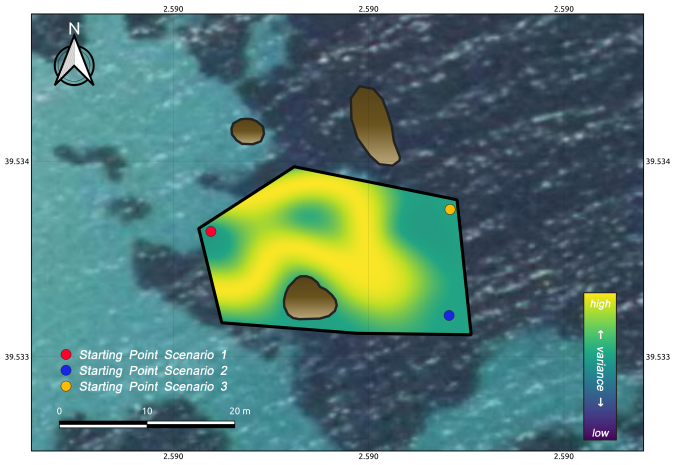
This section reports the results obtained by applying the proposed GPP approach on three different areas of interest for PO monitoring. After a description of the analysed scenarios, the GPP parameters configuration selected to obtain the results of the specific algorithm are presented. Finally, the section reports the offline results obtained for the comparative analysis among the three IPP strategies introduced in Section 2.2, exploiting the metrics presented in Section 2.3.

3.1. Test Scenario Description

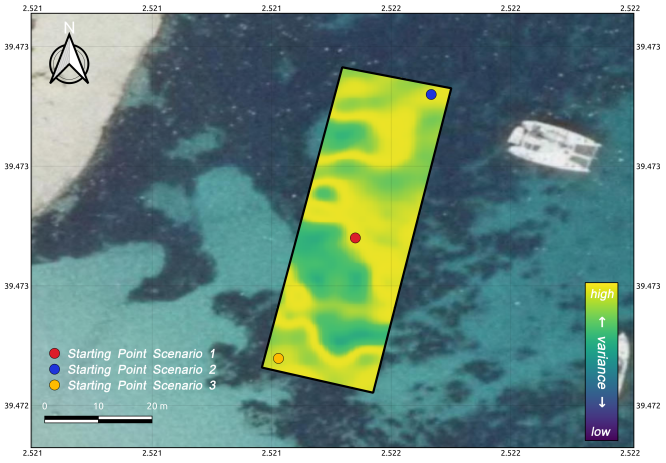
To execute the offline tests, conducted to produce the results presented in this paper, a Dell XPS 13 7390 with a Intel(R) Core(TM) i7-10510U CPU @ 1.80 GHz–2.30 GHz and 16GB of RAM was used. In particular, the tests were performed working on data relative to three different geographical areas located in Mallorca Island (Spain). All the areas considered in this work are coastal regions with an average distance from the coast of about 30m and they can be classified as very shallow waters, as the depth within them ranges between 3 m to 15 m. However, note that the GPP algorithm has been designed on the assumption of working in a 2D scenario, as the slope of areas where PO uses to grow is usually low. Thus, the bathymetry of the inspected area does not impact the behaviour of the algorithm nor the execution of the mission. In any case, the AUV is equipped with an altitude sensor and it is controlled to maintain a fixed distance from the sea bottom during all the missions. This allows the AUV to maintain good acquisition conditions about the images used to disclaim the presence of PO.

In each area, three different scenarios were considered by varying the origin point of the path. The extension and shape of PO meadow in the environment was modelled through a GP trained on data describing, in probabilistic terms, the presence of PO in a specific location of the target area. Such data were produced by segmentation of images through a pre-trained CNN. The training of the GP was carried out using GpFlow [43], library of Tensorflow [44], in Python environment. Figure 4 shows a satellite view of the three areas of inspection. In each of them, the variance map obtained by the GP prediction is represented as a colormap, where high variance regions are depicted in yellow, while the

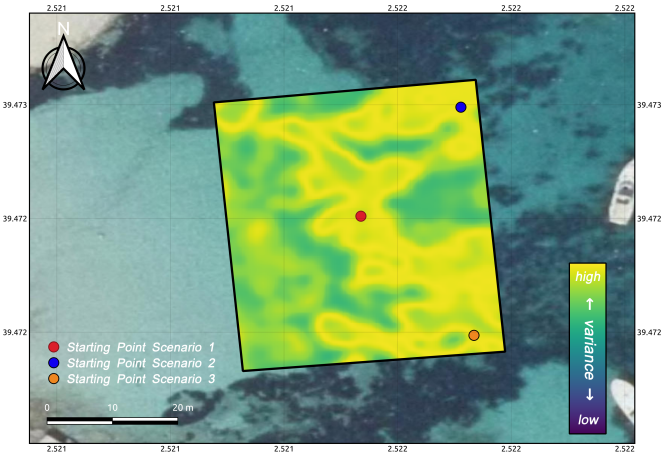
areas with low variance are displayed in blue . The a priori known obstacles are depicted in brown and the different starting points are shown on each map, respectively, in red, blue, and orange, according to their order.



(**a**)



(**b**)



(**c**)

Figure 4. Aerial view of the three geographic areas of interest. (**a**) Starting points Area 1; (**b**) Starting points Area 2; (**c**) Starting points Area 3.

3.2. GPP Parameter Selection

The proposed GPP method utilises several parameters which need to be defined by the user prior to the execution of the algorithm. In order to select the parameters configuration which allows to produce the best path in terms of ME, O2E, and PL, a sensitivity analysis was conducted. Such investigation was performed on the data relative to the first geographical area, represented in Figure 4a. Afterwards, the selected parameters configuration was validated on the other two areas considered in this work, which significantly differ from Area 1 for dimension, shape and covariance map of the GP. The parameters involved in the analysis are the gene mutation rate q, the chromosome mutation rate z, the tuning parameter for the utility function β, and the crossover method. Varying the values of these parameters, several possible combinations are obtained. Then, 40 offline executions of the GPP algorithm were computed for each set of parameters, calculating the metrics ME, PL, and O2E for each generated path. Finally, the configuration providing the best result in terms of mean and standard deviation of the considered metrics was selected. The resulting values for the parameters considered in the sensitivity analysis are reported in Table 1 together with, for the sake of completeness, a summary for all the other parameters adopted for the offline tests reported in the comparative analysis in Section 2.2. Parameters in the table are divided according to the phase of the algorithm they affect and a brief description of each of them is provided. Furthermore, the 'Single-Point' method (details in Appendix A.3) resulted to be the crossover methods which provides the best routes, among the implemented ones. It must be stressed that the values d_{min} and d_{max} are chosen according to the size of Area 1, thus for the other target areas these values have been scaled by the ratio between the size of the target area and the size of Area 1, which result to be 1.6 and 2.45, respectively, for Area 2 and Area 3. Thus, the values for d_{min} and d_{max} are 0.32 m and 1.28 m for Area 2, while for Area 3 they are 0.49 m and 1.96 m. It is worth noticing that limits in the manoeuvrability could occur if the vehicle moves at the same speed and d_{min} decreases. Therefore, if a target area presents some peculiarity or the AUV has different manoeuvrability capabilities (or a different speed), the parameter d_{min} may differ from the one presented in Table 1. Moreover, the parameter σ_α have been set considering the manoeuvrability capabilities of the AUV utilised for the collection of the images exploited to train the GP, which is a SPARUS II AUV [39] of property of the University of the Balearic Islands (UIB).

Table 1. Summary table of the GPP parameters value utilised within this work.

Symbol	Description	Value
	Global initialisation	
n	# of chromosomes in the initial population	200
m	# of genes per chromosome	40
σ_α	Std. Dev. of the Gaussian distribution of α	20°
d_{min}	Lower bound of the uniform distribution of d	0.2 m
d_{max}	Higher bound of the uniform distribution of d	0.8 m
	Genetic operations	
k	# of chromosomes maintained after the "Ranking" phase	10
q	Gene mutation rate	5%
z	Chromosome mutation rate	40%
t_{max}	Duration limit of the iterative phase	15 s
	Utility function	
β	Tuning parameter for the trade-off in the utility function	0.95
	Consistency check	
C_g	Attempts limit to generate α from the Gaussian distributions	10
C_u	Attempts limit to generate α from the uniform distribution	15
C_n	# of genes to delete to exit from inconsistent conditions	5

3.3. Comparative Analysis

A comparative offline analysis has been carried out to assess the performance of the three considered IPP strategies. For each scenario in each geographic area, 40 tests have been conducted, evaluating the metrics detailed in Section 2.3. The planning time has been set to 25 s for all the three planners, in which the duration of the initialisation phase for both GPP and S-B has been set to 10 s.

Note that the algorithm proposed in [14] is executed for 15 s (5 of initialisation and 10 of optimisation). Here, both the phases have been extended not to risk to limit it in its performance, also in compliance with the underwater scenario that, in view of typically slowly dynamics with respect to a ground one, may allow longer planning time. Tables 2–4 report the results, in terms of the metrics mean and three times the standard deviation, for each planner in all the scenarios of Areas 1–3, respectively. The scenario name is referred to the starting point selected in the area of interest.

Table 2. Results, in terms of mean and three times the std. dev., of the 3 planners in 3 scenarios of Area 1.

AREA 1 (Figure 4a)				
Scenario	**Planner**	**ME [bits]**	**PL [m]**	**O2E [m]**
First	Genetic	0.58 ± 0.01	20.2 ± 3.39	14.3 ± 2.56
	Random	0.58 ± 0.01	20.2 ± 3.28	15.1 ± 5.20
	S-B	0.53 ± 0.02	18.1 ± 12.7	10.7 ± 1.29
Second	Genetic	0.57 ± 0.01	20.6 ± 3.15	18.0 ± 3.31
	Random	0.57 ± 0.01	20.3 ± 3.38	17.6 ± 3.86
	S-B	0.55 ± 0.01	26.7 ± 6.74	19.3 ± 1.00
Third	Genetic	0.54 ± 0.01	20.9 ± 2.68	17.0 ± 6.79
	Random	0.54 ± 0.01	20.6 ± 3.00	16.6 ± 4.80
	S-B	0.52 ± 0.01	21.5 ± 13.5	12.4 ± 1.39

Table 3. Results, in terms of mean and three times the std. dev., of the 3 planners in 3 scenarios of Area 2.

AREA 2 (Figure 4b)				
Scenario	**Planner**	**ME [bits]**	**PL [m]**	**O2E [m]**
First	Genetic	0.51 ± 0.02	31.7 ± 6.59	21.5 ± 20.9
	Random	0.50 ± 0.02	30.3 ± 5.07	13.1 ± 7.13
	S-B	0.48 ± 0.15	13.4 ± 25.1	8.27 ± 10.7
Second	Genetic	0.51 ± 0.01	33.6 ± 5.89	27.7 ± 9.77
	Random	0.50 ± 0.01	32.0 ± 4.82	26.3 ± 11.5
	S-B	0.50 ± 0.03	22.3 ± 19.8	17.7 ± 17.4
Third	Genetic	0.51 ± 0.01	28.7 ± 5.85	14.2 ± 5.24
	Random	0.50 ± 0.01	30.3 ± 5.08	13.1 ± 7.13
	S-B	0.42 ± 0.12	26.5 ± 22.5	20.3 ± 16.0

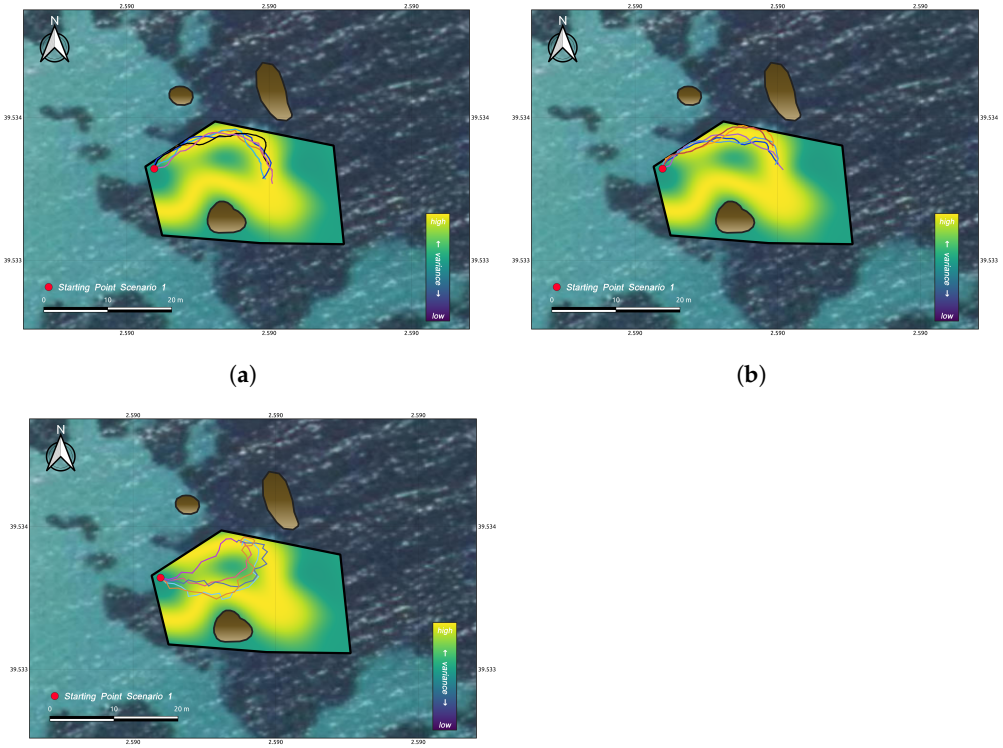
Table 4. Results, in terms of mean and three times the std. dev., of the 3 planners in 3 scenarios of Area 3.

AREA 3 (Figure 4c)				
Scenario	**Planner**	**ME [bits]**	**PL [m]**	**O2E [m]**
First	Genetic	0.51 ± 0.01	46.6 ± 7.41	30.4 ± 10.4
	Random	0.51 ± 0.02	47.0 ± 8.96	28.4 ± 11.1
	S-B	0.49 ± 0.11	10.8 ± 16.3	8.19 ± 11.9
Second	Genetic	0.51 ± 0.01	46.4 ± 8.42	26.8 ± 23.3
	Random	0.51 ± 0.03	47.9 ± 7.14	25.6 ± 27.0
	S-B	0.45 ± 0.16	35.1 ± 21.6	28.5 ± 20.6
Third	Genetic	0.51 ± 0.01	47.9 ± 6.37	37.0 ± 14.1
	Random	0.51 ± 0.02	47.7 ± 7.23	30.6 ± 26.2
	S-B	0.49 ± 0.07	21.7 ± 28.0	16.9 ± 21.1

Analysing the percentage of times each planner performs better than the other ones in terms of ME only, the results are reported in Table 5. Figures 5–7 show a subset of the paths generated by each investigated planner in the three areas of inspection. In fact, although 40 optimal paths have been generated for each trial, just five of them have been randomly selected and represented, in order to maintain a good readability of the figures. Instead, Figures 8–10 display the plots of the ME, PL, and O2E values, respectively, over 40 trials for each planner in all the three geographic areas. For the sake of readability, only the plots of the first scenario for all the 3 areas of interest are reported. In fact, the other scenarios provide similar plots, without any anomalies in the results. Last, Figure 11 shows the trend of the utility value over each iteration of the GPP's optimisation phase for 40 trials (Area 2-Scenario 1). In particular, the values represented at the initial time are the ones obtained at the end of the initialisation phase of each trial. Then, each iteration of the GPP algorithm tries to improve the utility value, which is therefore optimised over time. Moreover, the mean value of the utility at each iteration is shown in red to represent the mean trend obtained thanks to the optimisation procedure.

Table 5. Performance comparison of the ME metric of the 3 path planners over the 3 areas of interest with different starting points.

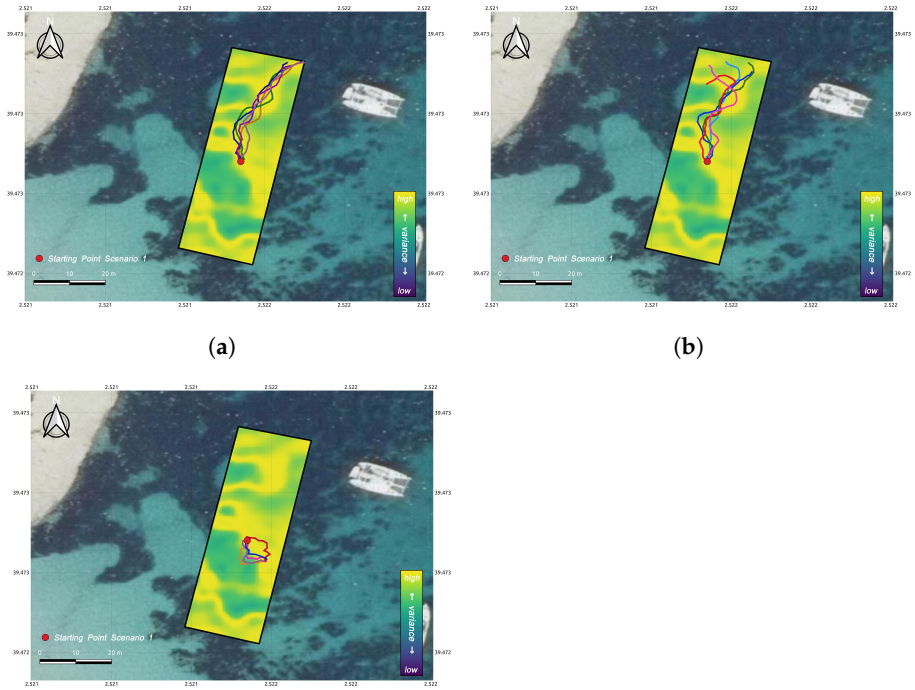
AREA 1		
Scenario 1	**Scenario** 2	**Scenario** 3
Genetic: 85% Random: 15% S-B: 0%	Genetic: 67.5% Random: 27.5% S-B: 5%	Genetic: 82.5% Random: 15% S-B: 2.5%
AREA 2		
Scenario 1	**Scenario** 2	**Scenario** 3
Genetic: 50% Random: 0% S-B: 50%	Genetic: 52.5% Random: 12.5% S-B: 35%	Genetic: 85% Random: 15% S-B: 0%
AREA 3		
Scenario 1	**Scenario** 2	**Scenario** 3
Genetic: 65% Random: 20% S-B: 15%	Genetic: 70% Random: 22.5% S-B: 7.5%	Genetic: 80% Random: 12.5% S-B: 7.5%



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

(**c**)

Figure 5. Five best paths Area 1-Scenario 1. (**a**) Genetic; (**b**) Random; (**c**) S-B.



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

(**c**)

Figure 6. Five best paths Area 2-Scenario 1. (**a**) Genetic; (**b**) Random; (**c**) S-B.

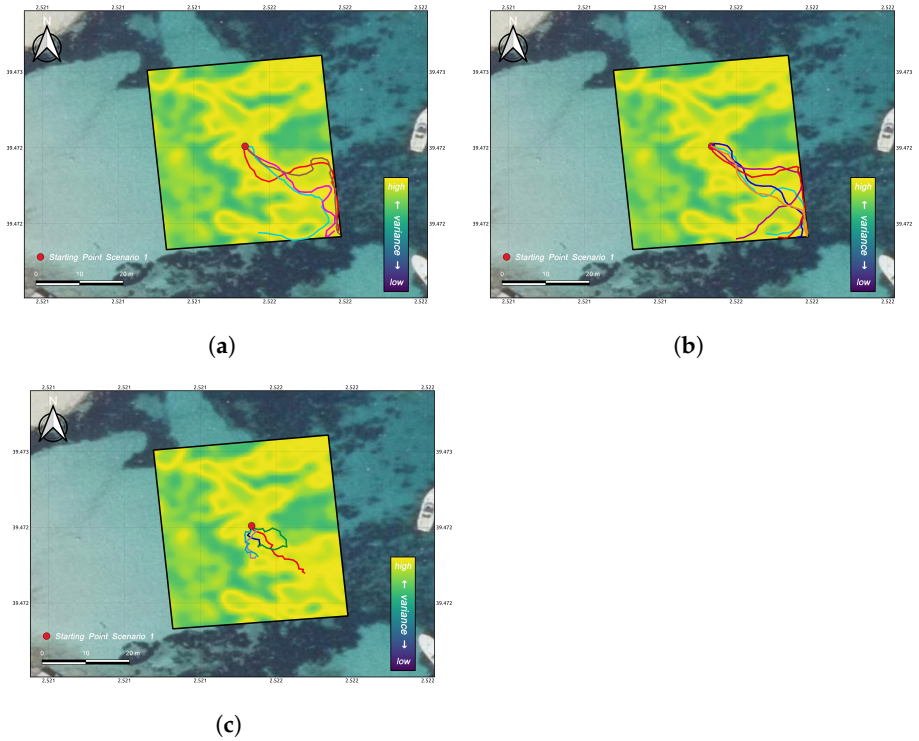


Figure 7. Five best paths Area 3-Scenario 1. (**a**) Genetic; (**b**) Random; (**c**) S-B.

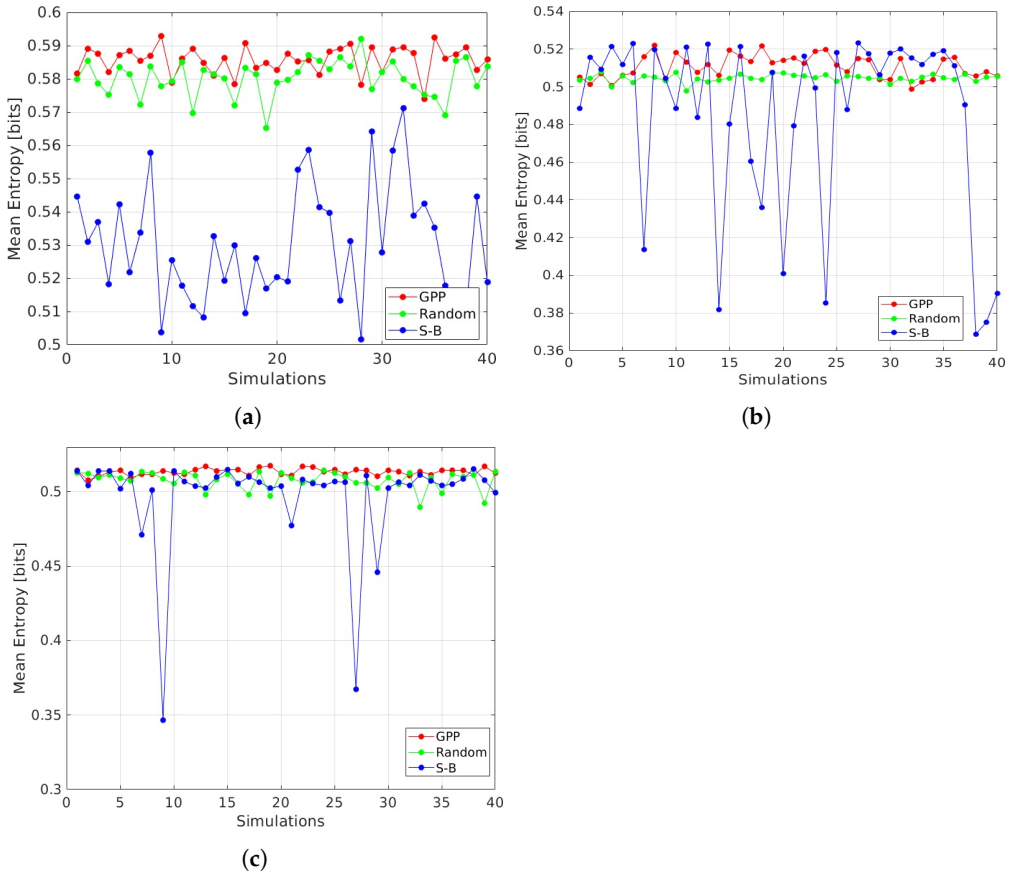


Figure 8. Mean Entropy over 40 offline executions in scenario 1. (**a**) Area 1; (**b**) Area 2; (**c**) Area 3.

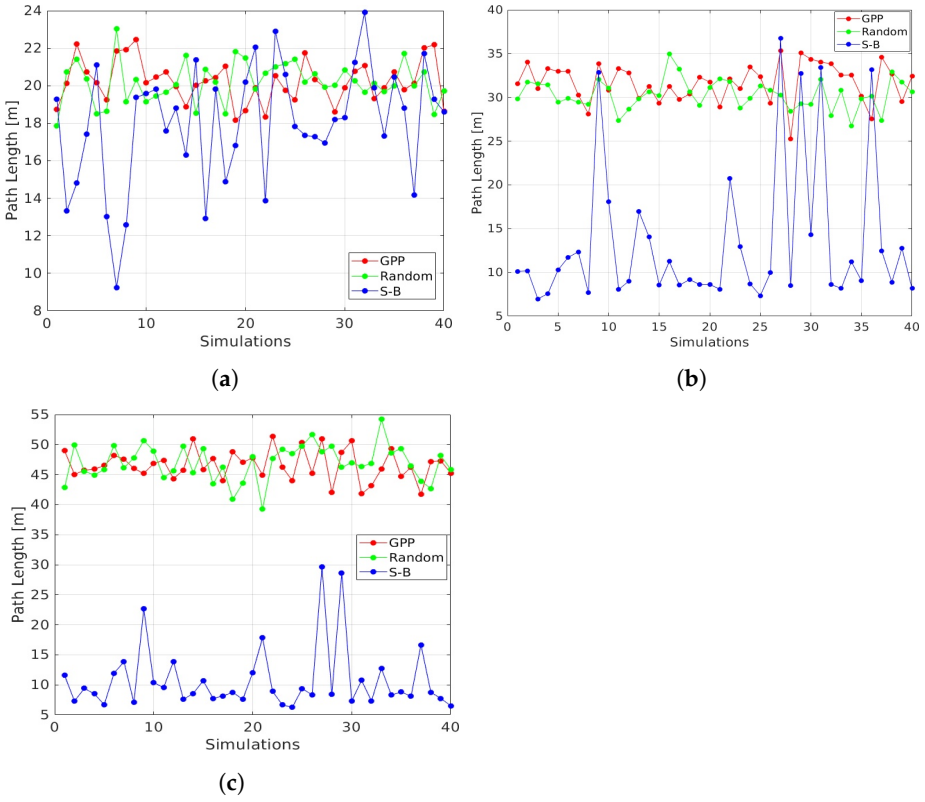


Figure 9. Path Length metric over 40 offline executions in scenario 1. (**a**) Area 1; (**b**) Area 2; (**c**) Area 3.

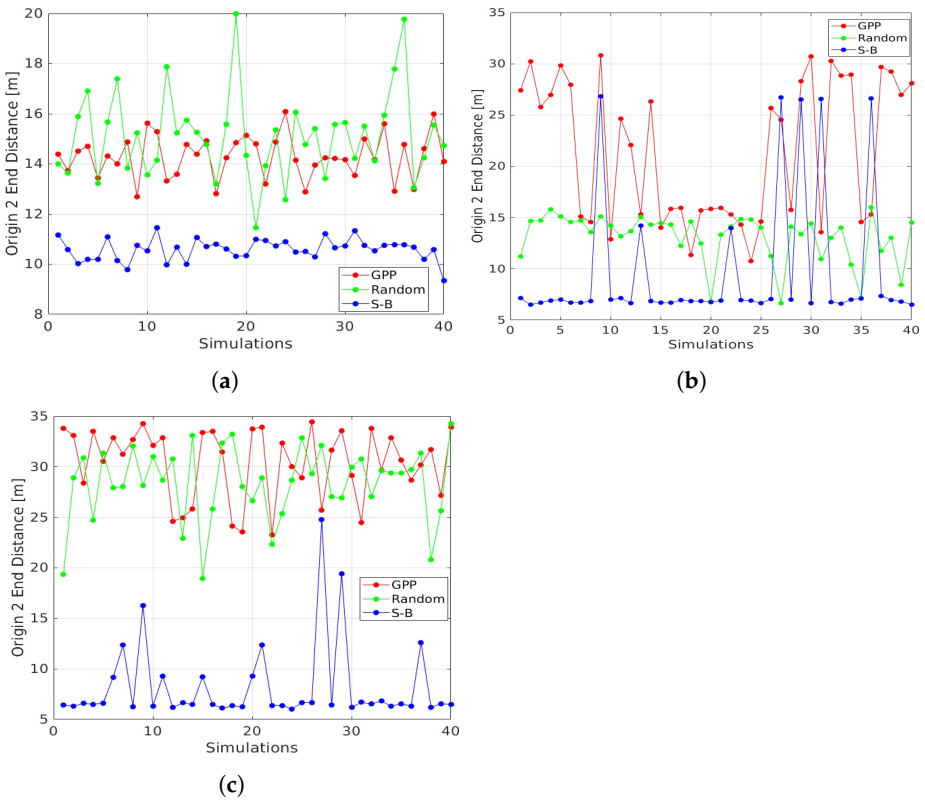


Figure 10. Origin to End distance of the path over 40 offline executions in scenario 1.

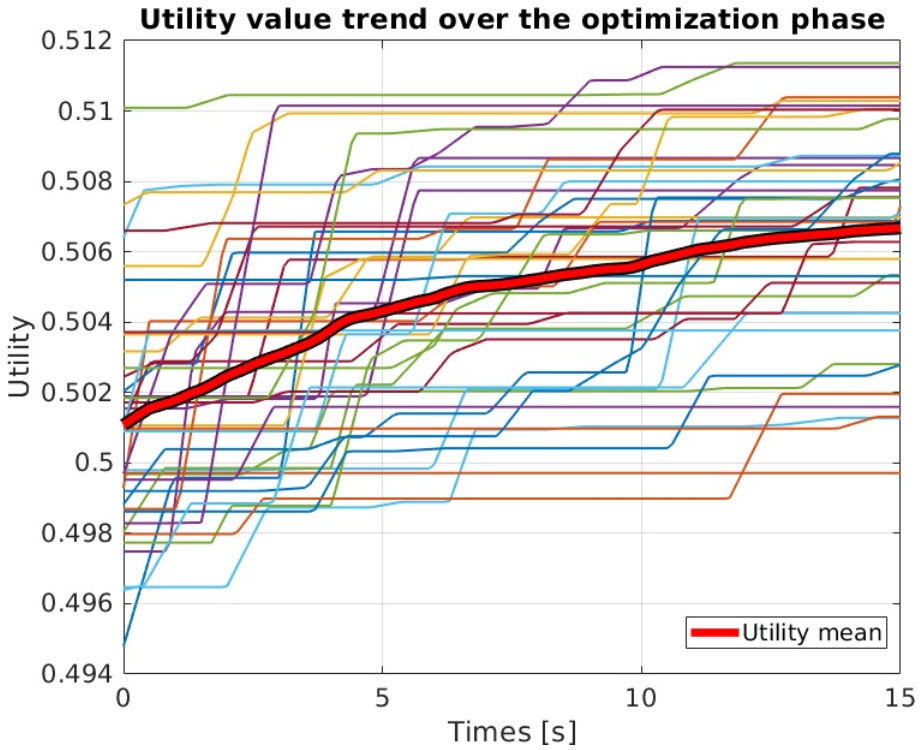


Figure 11. Utility value trend over the optimisation phase of the GPP algorithm. (**a**) Area 1; (**b**) Area 2; (**c**) Area 3.

4. Discussion

The results of the comparative study show that the proposed GPP generally performs better than the S-B and Random planners. In terms of ME, the GPP algorithm statistically outperforms the others for all the studied cases (area and scenario). It is worth discussing the case of Area 2-Scenario 1 and 2, in which the percentage of times the S-B planner performs better than the other approaches results to be greater than the other areas, as represented in Table 5 and Figure 8b. This is because the starting point of the first and second scenarios is located in an area characterised by a high variance of the GP. Therefore, the S-B planner is likely to find its *Station* point close to the starting point, resulting in a short path with high information content. This situation could occur whenever the most informative point is located near the starting point, despite the amount of target area investigated by the *Station Search* phase of the S-B planner. Consequently, the ME metric could result in a higher value for the S-B planner in similar scenarios, potentially leading to misleading considerations if not associated to the other two metrics. In fact, the standard deviation of the ME for the results of S-B planner in Area 2-Scenario 1 is much higher than the one obtained by GPP, as reported in Table 3 and Figure 8b, and that highlights a statistic inconsistency in the S-B planner results. On the contrary, the other two approaches generate paths with a higher mean value of ME and lower standard deviation over the 40 offline executions.

Moreover, the proposed GPP algorithm results to generate paths with a good coverage of the area under inspection. In particular, as shown in Figures 9 and 10, the PL and O2E metrics obtained by the GPP strategy result to be significantly better than the ones of the S-B planner, allowing to explore a wider portion of the target area as well as gathering more useful data about PO along the path. The PL and O2E metrics result comparable between the GPP and the Random planner. This is reasonable since the two planners exploit the same distributions and parameters for the chromosome generation. However, as reported in Table 5, the ME of the GPP approach is statistically higher than that of the Random planner, thus justifying the use of the iterative part of the GPP algorithm. Furthermore, Figure 11 shows the improvement obtained thanks to the optimisation phase

performed in the GPP algorithm, and not included in the Random planner instead. The mean trend of the utility value, represented in red in the plot, is strictly increasing but does not reach convergence in the defined planning time t_{max}. Thus, the choice of utilising a stop condition based on time become more relevant, since the convergence of the algorithm could require an amount of time which will make the algorithm not exploitable online. An analogous result is reported also in [14], which shows the evolution of the utility value over an extended planning time of 180 s for the S-B planner. The reader is there addressed, as a S-B characterisation is beyond the scope of the authors' manuscript.

Last, it is worth pointing out that the results obtained by the GPP algorithm, utilising the parameters configuration discussed in Section 3.2, have similar performance over the three different target areas, although obtained through optimisation on a specific area (Area 1). This confirms that the considered set of parameters has a validity independent from the specific area and scenario.

5. Conclusions

This work presents a novel IPP algorithm based on GA theory, renamed GPP, to address the path planning problem in underwater environments. The strategy relies on the concepts of genetic operations, namely crossover and mutation, to generate the optimal path in terms of informativeness and coverage of the inspected area. The proposed approach has been tested offline, exploiting real marine data, within the context of PO monitoring, where the GPP aims at generating an optimal path for an AUV to investigate the presence of PO in a target area. To this aim, the GPP algorithm exploits the uncertainty of a GP prediction, which is employed to provide an a priori knowledge about the presence of PO in the area of interest. The algorithm uses such uncertainty to compute the DE in the points belonging to each generated path, which is adopted to represent their information content. Afterwards, the GPP method retrieves the path that maximises an utility function, consisting of an averaged version of the DE along the path and a geometric index representing the coverage of the inspected area. A comparative study with other two IPP strategies has been carried out to evaluate the performance of the proposed GPP method. The first one is a S-B algorithm, which is considered, to the extent of the authors' knowledge, as one of the main state of the art IPP approaches for IG operations. The second strategy is instead a Random planner, which has been taken into account to assess whether the genetic iterative operations performed in the GPP algorithm provide improvements to the path performance, or they can be avoided to reduce planning time. Thus, this planner implements only the first phase of the GPP method, generating random chromosomes for all the available planning time, without applying any crossover or mutation transformations, and in the end the best chromosome in terms of utility value among all the generated ones is selected.

The comparative analysis has been performed over 3 different areas, located in Mallorca Island (Spain), and considering 3 starting points for each area. The result of the tests showed that the GPP generally performs better than the S-B planner in terms of all the metrics used for the comparison: ME, PL, and O2E. The performance of GPP and Random planner result similar in terms of PL and O2E, but the GPP usually generates more informative paths (higher ME).

As a future work, the issue of retraining the GP exploiting the data acquired by the AUV along the planned path will be addressed. This will allow the AUV to update the model of the environment and to replan its path according to the newly acquired information. Moreover, alternative methods to perform the consistency check on the genes of the chromosome will be investigated, e.g., a cost factor in the utility function could be introduced to penalise those genes which are outside X_{free}. Furthermore, strategies to avoid the issue of overlapping path sections will be studied. Indeed, the presence of loops within a path implies that the same point will be visited multiple times. Therefore, the information acquired by the AUV along such path will be conditioned by the redundant points, which will contribute more than once to the global information content of the

path. This issue cannot be addressed by updating the GP model every time a new data is available, due to the computational cost of this operation. However, the problem could be solved imposing some penalties in the utility function when a path has overlapping sections. Finally, the proposed GPP algorithm will be implemented on an AUV and tested on field experiments to assess its performance in a real marine scenario.

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

Funding: Work partially supported by MCIN/AEI/ 10.13039/501100011033, grant PID2020-115332RB-C33, by ERDF "A way of making Europe", by Government of the Balearic Islands, grant FPI/2031/2017 (CAIB) and by the Italian Ministry of Education and Research (MIUR) in the framework of the Cross-Lab project (Departments of Excellence).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AUV	Autonomous Underwater Vehicle
CNN	Convolutional Neural Network
DE	Differential Entropy
GA	Genetic Algorithm
GP	Gaussian Process
GPP	Genetic Path Planner
IG	Information Gathering
IPP	Informative Path Planning
ME	Mean Entropy
MI	Mutual Information
O2E	Distance Origin to End
PL	Path Length
PO	Posidonia Oceanica
RIG	Rapidly-exploring Information Gathering
RRT	Rapidly-exploring Random Tree
S-B	Sampling-Based Planner
S-E	Squared-Exponential

Appendix A. Background

Appendix A.1. Gaussian Process

A GP is a stochastic process formed by a collection of random variables that have a multivariate normal distribution. In this work, the random variables represent the value of a function $f(x) \sim GP(m(x), k(x, x'))$ that gives the probability of having PO at location x. Such GP is characterised by the mean $m(x)$ and the covariance $k(x, x', \theta)$ between any two given positions x and x'. The Squared-Exponential (SE) covariance kernel, defined as $k(x, x', \theta) = \sigma_f^2 exp(-\frac{||x-x'||^2}{2l^2})$, is utilised to describe the covariance value between two locations. In the SE covariance expression, the parameters are indicated as $\theta = [\sigma_f^2, l]$, where σ_f^2 represents the maximum allowable covariance, and l is the kernel *length scale* that defines the kernel smoothness [11]. The joint distribution of the observed training values $y = \{y_i\}_{i=1,\dots,N}$ and the test values f_* is given by [8]

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right) \tag{A1}$$

where σ_n^2 is the variance of the noise fluctuations, $X = \{x_i \mid i = 1, \cdots, N\}$ are training locations, and $X_* = \{x_i^* \mid i = 1, \cdots, N_*\}$ represent testing locations. Applying standard rules for conditioning Gaussian functions, given the training locations, the predictive distribution over the test locations is given by $f_* \sim \mathcal{N}(\bar{f}_*, \mathbb{V}[f_*])$. The resulting expressions of mean and variance are

$$\bar{f}_* = K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} y \tag{A2}$$

$$\mathbb{V}[f_*] = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*) \tag{A3}$$

where $\mathbb{V}[f_*]$ represents the prediction uncertainty in the query locations X_*, and $\bar{f}_*$ the mean probability of having PO in X_*.

Appendix A.2. Genetic Algorithm

GA creates a population following basic evolutionary concept to solve the path finding problem in parallel. Every path in a population is called *chromosome* with a node representation, called *gene*. Through *crossover* operators, parents genes are mixed together to generate *offsprings*. To avoid local minima, through *mutation* operation, random genes are mutated in every chromosome. This operation is characterised by a *mutation rate* index which regulates the percentage of genes to be mutated in each chromosome. GAs evaluate chromosomes environment adaptability in terms of the *fitness* and *ranking* operators value. Fitter individuals selection is the foundation of algorithm convergence [45].

Appendix A.3. Crossover Methods

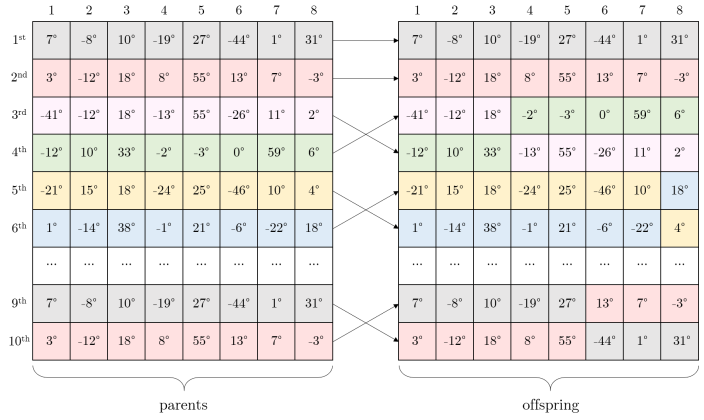
GAs exploit crossover methods to generate offspring from a set chromosome ranked as best. Several crossover strategies are applicable to mix the genetic content of pairs of chromosomes. In the following, we describe four of these techniques.

- Single-Point Crossover
  This method selects pairs of chromosomes with m genes according to their ranking order and samples from a uniform distribution with range $[1, m]$ the index of mutation i_m. Then, the genes from i_m to the end are switched between the pair of chromosomes. An illustration of this crossover approach is given in the Figure A1a.

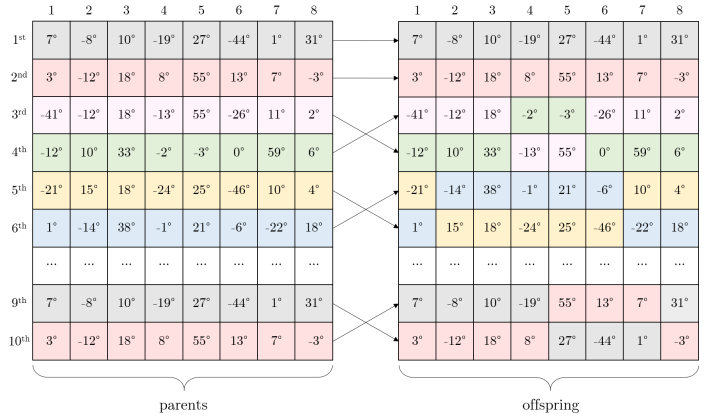
- Double-Point Crossover
  Similar to the Single-Point Crossover, this method also selects a pair of chromosomes according to their ranking, but differs in that it samples two mutation indices. The first, i_{m1} is sampled from a uniform distribution with range $[1, m]$; the second, i_{m2} is also sampled from a uniform distribution but with range $[i_{m1}, m]$, so that $i_{m2} > i_{m1}$. Then, the genes in the range $[i_{m1}, i_{m2}]$ are switched between the pair of chromosomes. Figure A1b shows an example of this crossover method.

- Uniform Crossover
  This method generates a random array with dimension m (as the number of genes in each chromosomes) sampling from an uniform distribution with range $[0, 1]$. The pair of chromosomes doing crossover, which are selected according to their ranking position, switch their genes where the value of the random array is higher or equal than a probability p. A visual example is provided in the Figure A1c, where $p = 0.5$.

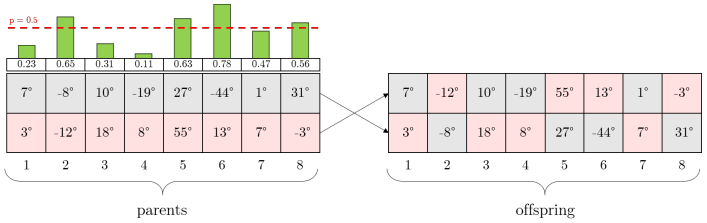
- Single-Random Crossover
  Working almost as the Single-Point Crossover, this method differs only in how the pair of chromosomes are selected. In fact, differently form the previous approaches, which are based on the ranking order, in this one the chromosomes are randomly paired up. Figure A1d shows an example of this approach, where the randomly generated pairs are 3rd–9th, 4th–6th, and 5th–10th.



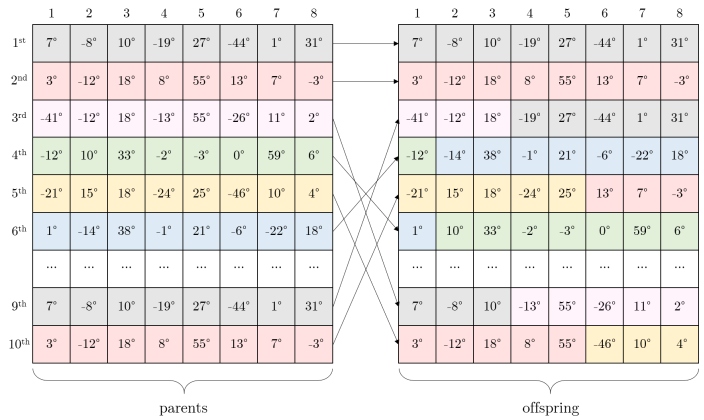
(**a**)



(**b**)



(**c**)



(**d**)

Figure A1. Crossover approaches. (**a**) Single-Point Crossover; (**b**) Double-Point Crossover; (**c**) Uniform Crossover; (**d**) Single-Random Crossover.

References

1. McFarland, C.J.; Jakuba, M.V.; Suman, S.; Kinsey, J.C.; Whitcomb, L.L. Toward ice-relative navigation of underwater robotic vehicles under moving sea ice: Experimental evaluation in the arctic sea. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 1527–1534.
2. Williams, S.B.; Pizarro, O.R.; Jakuba, M.V.; Johnson, C.R.; Barrett, N.S.; Babcock, R.C.; Kendrick, G.A.; Steinberg, P.D.; Heyward, A.J.; Doherty, P.J.; et al. Monitoring of benthic reference sites: Using an autonomous underwater vehicle. *IEEE Robot. Autom. Mag.* **2012**, *19*, 73–84.
3. Schofield, O.; Ducklow, H.W.; Martinson, D.G.; Meredith, M.P.; Moline, M.A.; Fraser, W.R. How do polar marine ecosystems respond to rapid climate change? *Science* **2010**, *328*, 1520–1523.
4. Mayer, L. Frontiers in sea floor mapping and visualization. *Mar. Geophys. Res.* **2006**, *27*, 7–17.
5. Furlong, M.E.; Paxton, D.; Stevenson, P.; Pebody, M.; McPhail, S.D.; Perrett, J. Autosub long range: A long range deep diving AUV for ocean monitoring. In Proceedings of the 2012 IEEE/OES Autonomous Underwater Vehicles (AUV), Southampton, UK, 24–27 September 2012; pp. 1–7.
6. Raja, P.; Pugazhenthi, S. Optimal path planning of mobile robots: A review. *Int. J. Phys. Sci.* **2012**, *7*, 1314–1320.
7. Binney, J.; Krause, A.; Sukhatme, G.S. Informative Path Planning for an Autonomous Underwater Vehicle. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, Alaska, 3–8 May 2010; pp. 4791–4796.
8. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*; The MIT Press: Cambridge, MA, USA; 2006.
9. Krause, A.; Singh, A.; Guestrin, C. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.* **2008**, *9*, 235–284.
10. Ma, K.C.; Liu, L.; Heidarsson, H.K.; Sukhatme, G.S. Data-driven learning and planning for environmental sampling. *J. Field Robot.* **2018**, *35*, 643–661.
11. Guerrero-Font, E.; Bonin-Font, F.; Martin-Abadal, M.; Gonzalez-Cid, Y.; Oliver-Codina, G. Sparse Gaussian process for online seagrass semantic mapping. *Expert Syst. Appl.* **2021**, *170*, 114478.
12. Ghaffari Jadidi, M.; Valls Miro, J.; Dissanayake, G. Sampling-based incremental information gathering with applications to robotic exploration and environmental monitoring. *Int. J. Robot. Res.* **2019**, *38*, 658–685.
13. Hollinger, G.A.; Pereira, A.A.; Sukhatme, G.S. Learning uncertainty models for reliable operation of autonomous underwater vehicles. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 5593–5599.
14. Viseras, A.; Shutin, D.; Merino, L. Robotic active information gathering for spatial field reconstruction with rapidly-exploring random trees and online learning of Gaussian processes. *Sensors* **2019**, *19*, 1016.
15. Krause, A.; Guestrin, C. Submodularity and its applications in optimized information gathering. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 1–20.
16. Marchant, R.; Ramos, F. Bayesian optimisation for intelligent environmental monitoring. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Algarve, Portugal, 7–12 October 2012; pp. 2242–2249.
17. Binney, J.; Sukhatme, G.S. Branch and bound for informative path planning. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MI, USA, 14–18 May 2012; pp. 2147–2154.
18. Kodgule, S.; Candela, A.; Wettergreen, D. Non-myopic planetary exploration combining in situ and remote measurements. *arXiv* **2019**, arXiv:1904.12255.
19. Yang, K.; Keat Gan, S.; Sukkarieh, S. A Gaussian process-based RRT planner for the exploration of an unknown and cluttered environment with a UAV. *Adv. Robot.* **2013**, *27*, 431–443.
20. Hollinger, G.A.; Sukhatme, G.S. Sampling-based robotic information gathering algorithms. *Int. J. Robot. Res.* **2014**, *33*, 1271–1287.
21. Schmid, L.; Pantic, M.; Khanna, R.; Ott, L.; Siegwart, R.; Nieto, J. An efficient sampling-based method for online informative path planning in unknown environments. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1500–1507.
22. 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.
23. Hitz, G.; Galceran, E.; Garneau, M.È.; Pomerleau, F.; Siegwart, R. Adaptive continuous-space informative path planning for online environmental monitoring. *J. Field Robot.* **2017**, *34*, 1427–1449.
24. Popović, M.; Vidal-Calleja, T.; Chung, J.J.; Nieto, J.; Siegwart, R. Informative Path Planning for Active Field Mapping under Localization Uncertainty. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 31 October 2020; pp. 10751–10757.
25. 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.
26. LaValle, S.M. *Rapidly-Exploring Random Trees: A New Tool for Path Planning*; The Annual Research Report; 1998. Available online: https://www.semanticscholar.org/paper/Rapidly-exploring-random-trees-%3A-a-new-tool-for-LaValle/d967d9550f831a8b3f5cb00f8835a4c866da60ad (accessed on 10 October 2021).
27. Yu, X.; Gen, M. *Introduction to Evolutionary Algorithms*; Springer: Berlin/Heidelberg, Germany, 2010.
28. Katoch, S.; Chauhan, S.S.; Kumar, V. A review on genetic algorithm: past, present, and future. *Multimed. Tools Appl.* **2020**, *80*, 8091–8126.

29. Kapanoglu, M.; Alikalfa, M.; Ozkan, M.; Parlaktuna, O. A pattern-based genetic algorithm for multi-robot coverage path planning minimizing completion time. *J. Intell. Manuf.* **2012**, *23*, 1035–1045.
30. Roberge, V.; Tarbouchi, M.; Labonté, G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Trans. Ind. Informatics* **2012**, *9*, 132–141.
31. Alvarez, A.; Caiti, A.; Onken, R. Evolutionary path planning for autonomous underwater vehicles in a variable ocean. *IEEE J. Ocean. Eng.* **2004**, *29*, 418–429. doi:10.1109/JOE.2004.827837.
32. Tao, W.; Yan, S.; Pan, F.; Li, G. AUV Path Planning Based on Improved Genetic Algorithm. In Proceedings of the 2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE), Dalian, China, 19–20 September 2020; pp. 195–199.
33. Tanakitkorn, K.; Wilson, P.A.; Turnock, S.R.; Phillips, A.B. Grid-based GA path planning with improved cost function for an over-actuated hover-capable AUV. In Proceedings of the 2014 IEEE/OES Autonomous Underwater Vehicles (AUV), Oxford, MI, USA, 6–9 October 2014; pp. 1–8.
34. Li, Y.; Huang, Z.; Xie, Y. Path planning of mobile robot based on improved genetic algorithm. In Proceedings of the 2020 3rd International Conference on Electron Device and Mechanical Engineering (ICEDME), Suzhou, China, 1–3 May 2020; pp. 691–695.
35. Marbà, N.; Duarte, C.M.; Cebrián, J.; Gallegos, M.E.; Olesen, B.; Sand-Jensen, K. Growth and population dynamics of Posidonia oceanica on the Spanish Mediterranean coast: Elucidating seagrass decline. *Mar. Ecol. Prog. Ser.* **1996**, *137*, 203–213.
36. Martin-Abadal, M.; Guerrero-Font, E.; Bonin-Font, F.; Cid, Y.G. Deep Semantic Segmentation in an AUV for Online Posidonia Oceanica Meadows Identification. *IEEE Access* **2018**, *6*, 60956–60967.
37. Ridolfi, A.; Costanzi, R.; Fanelli, F.; Monni, N.; Allotta, B.; Bianchi, S.; Conti, R.; Gelli, J.; Gori, L.; Pugi, L.; et al. FeelHippo: A low-cost autonomous underwater vehicle for subsea monitoring and inspection. In Proceedings of the 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC), Florence, Italy, 7–10 June 2016; pp. 1–6. doi:10.1109/EEEIC.2016.7555607.
38. Gelli, J.; Meschini, A.; Monni, N.; Pagliai, M.; Ridolfi, A.; Marini, L.; Allotta, B. Development and Design of a Compact Autonomous Underwater Vehicle: Zeno AUV. In Proceedings of the 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles (CAMS 2018), Opatija, Croatia, 11–12 September 2018. doi:10.1016/j.ifacol.2018.09.463.
39. Carreras, M.; Candela, C.; Ribas, D.; Palomeras, N.; Magií, L.; Mallios, A.; Vidal, E.; Pairet, È.; Ridao, P. Testing SPARUS II AUV, an Open Platform for Industrial, Scientific and Academic Applications. *CoRR*. 2018. Available online: https://arxiv.org/abs/1811.03494 (accessed on 10 October 2021).
40. Allen, B.; Stokey, R.; Austin, T.; Forrester, N.; Goldsborough, R.; Purcell, M.; von Alt, C. REMUS: A Small, Low Cost AUV; System Description, Field Trials and Performance Results. 1997; Volume 2, pp. 994–1000. Available online: https://www.semanticscholar.org/paper/REMUS%3A-a-small%2C-low-cost-AUV%3B-system-description%2C-Allen-Stokey/e93b548c13c28548bfd22d9147abb656e435a04b (accessed on 10 October 2021).
41. Sivanandam, S.; Deepa, S. Genetic algorithms. In *Introduction to Genetic Algorithms*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 15–37.
42. Cover, T.M. *Elements of Information Theory*; John Wiley & Sons: Hoboken, NJ, USA, 1999.
43. Matthews, A.G.d.G.; van der Wilk, M.; Nickson, T.; Fujii, K.; Boukouvalas, A.; León-Villagrá, P.; Ghahramani, Z.; Hensman, J. GPflow: A Gaussian process library using TensorFlow. *J. Mach. Learn. Res.* **2017**, *18*, 1–6.
44. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: tensorflow.org (accessed on 10 October 2021).
45. Shivgan, R.; Dong, Z. Energy-Efficient Drone Coverage Path Planning using Genetic Algorithm. In Proceedings of the 2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR), Newark, NJ, USA, 11–14 May 2020; pp. 1–6.