



UNIVERSIDAD NACIONAL  
DE EDUCACIÓN A DISTANCIA

Escuela Técnica Superior de  
Ingeniería Informática



UNIVERSIDAD  
COMPLUTENSE DE MADRID

Facultad de  
Informática

# PLANIFICACIÓN DE TRAYECTORIAS DE EMBARCACIONES AUTÓNOMAS PARA LOCALIZAR CONTAMINANTES BIOLÓGICOS MEDIANTE ALGORITMOS GENÉTICOS

Diego Vela Calcerrada

Director/a: Eva Besada Portas

Co-director/a: José Antonio López Orozco

Trabajo de Fin de Máster

Máster Universitario  
en Ingeniería de sistemas y de control

Curso académico 2023-2024, convocatoria de septiembre

# Autorización

Autorizamos a la Universidad Complutense y a la UNED a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firmado: Diego Vela Calcerrada.

## Resumen

Este trabajo consiste en la elaboración de distintos métodos de exploración de un embalse basados en la realización de trayectorias generadas mediante algoritmos genéticos, con la finalidad de comprobar el estado de las aguas en cuanto a cianobacterias se refiere. Para ello, partiendo de una simulación del movimiento y proliferación de las cianobacterias, y teniendo en cuenta la dinámica del fluido del embalse, que hace que éstas se desplacen, se optimiza la trayectoria de un vehículo de superficie para que en su recorrido se maximice la observación de zonas con elevada concentración de dicha bacteria. Todo ello con el fin de ofrecer un optimizador de trayectorias realista, que tenga en cuenta factores tanto externos al vehículo, como el fluido en el que se encuentra, como factores provenientes de la propia dinámica del vehículo. Los resultados obtenidos en este trabajo para distintos escenarios de prueba indican que las propuestas presentadas permiten generar soluciones factibles y de buena calidad para que el vehículo autónomo de superficie realice medidas en las zonas donde el simulador de cianobacterias ha predicho una mayor concentración.

**Palabras clave:** Algoritmos genéticos, Vehículos autónomos, USV, NSGA-II

# Abstract

This work involves the development of different methods for exploring a reservoir based on trajectories generated through genetic algorithms, with the aim of checking the water quality concerning bacteria. To achieve this, starting from a simulation of the movement and proliferation of cyanobacteria, and considering the fluid dynamics of the reservoir, which causes these bacteria to move, the trajectory of a surface vehicle is optimized to maximize the observation of areas with high concentrations of the bacteria. The goal is to offer a realistic trajectory optimizer that takes into account both external factors, like the fluid in which the vehicle is located, and factors from the vehicle's own dynamics. The results obtained in this work for different test scenarios indicate that the presented proposals allow for generating feasible and high-quality solutions for the surface autonomous vehicle to take measurements in the areas where the cyanobacteria simulator has predicted a higher concentration.

**Keywords:** Genetic algorithms, autonomous vehicles, USVs, NSGA-II

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	2
1.2. Antecedentes . . . . .	2
1.3. Propuesta y objetivos . . . . .	5
1.4. Estructura del documento . . . . .	6
<b>2. Fundamentos y metodología</b>	<b>9</b>
2.1. Modelo predictivo de cianobacterias . . . . .	9
2.1.1. Dominio del embalse . . . . .	10
2.1.2. Dinámica del fluido . . . . .	10
2.1.3. Dinámica del organismo . . . . .	11
2.1.4. Simulaciones . . . . .	14
2.2. Descripción formal del problema de planificación de trayectorias . . . . .	15
2.3. Planificadores de trayectorias . . . . .	18
2.3.1. Planificador de trayectorias 3D . . . . .	20
2.3.2. Planificador de trayectorias 2D . . . . .	20
2.3.3. Planificador con puntos de paso preseleccionados . . . . .	22
2.3.4. Codificación con tiempos absolutos frente a tiempos relativos . . . . .	22
2.4. Algoritmo NSGA-II . . . . .	23
2.4.1. Concepto de Dominancia . . . . .	23
2.4.2. Frente pareto . . . . .	23
2.4.3. Calculo Crowding-Distance . . . . .	24
2.4.4. Concepto de cruce y mutación . . . . .	24
2.4.5. Funcionamiento del algoritmo . . . . .	25
2.5. Análisis de resultados . . . . .	26
2.5.1. Método de análisis de los resultados . . . . .	27
2.5.2. Ejemplo de análisis de resultados . . . . .	28
<b>3. Resultados</b>	<b>31</b>
3.1. Modificación de los parámetros del AG . . . . .	31
3.1.1. Problema en tres dimensiones . . . . .	32

3.1.2.	Problema en dos dimensiones . . . . .	35
3.1.3.	Problema con puntos de paso predefinidos . . . . .	37
3.2.	Modificación con tiempos relativos . . . . .	39
3.2.1.	Problema en tres dimensiones . . . . .	40
3.2.2.	Problema en dos dimensiones . . . . .	41
3.2.3.	Problema en dos dimensiones con puntos de paso predefinidos . . . . .	42
3.3.	Modificación del número de tramos . . . . .	43
3.3.1.	Problema en tres dimensiones . . . . .	44
3.3.2.	Problema en dos dimensiones . . . . .	45
3.3.3.	Problema en dos dimensiones con puntos de paso predefinidos . . . . .	46
3.4.	Funcionamiento del sistema ante distintos escenarios simulados . . . . .	48
3.4.1.	Problema en tres dimensiones . . . . .	48
3.4.2.	Problema en dos dimensiones . . . . .	50
3.4.3.	Problema en dos dimensiones con puntos de paso predefinidos . . . . .	51
<b>4.</b>	<b>Conclusiones y trabajos futuros</b>	<b>53</b>
4.1.	Conclusiones . . . . .	53
4.2.	Trabajo futuro . . . . .	55
	<b>Bibliografía y referencias</b>	<b>57</b>

# Índice de figuras

1.1. Vehículo autónomo para detección de hiervas, imagen extraída de Bechar and Vigneault [2016] . . . . .	3
1.2. Vehículos autónomos desarrollados por el grupo ISCAR para exploración de entornos marítimos . . . . .	4
1.3. Vehículo de superficie Boyi para detección de concentraciones de cianobacterias desarrollado por ISCAR . . . . .	5
2.1. Lake Mendota, Wisconsin, EE. UU. Imagen de Google Earth . . . . .	11
2.2. Representación 2D Lake Mendota . . . . .	11
2.3. Solución de la simulación de COMSOL . . . . .	11
2.4. Intensidad de luz en función del tiempo y profundidad . . . . .	12
2.5. Escenarios de simulación . . . . .	14
2.6. Ejemplo de calculo de número de puntos de paso . . . . .	20
2.7. Calculo Crowding-Distance, Deb et al. [2002] . . . . .	24
2.8. Proceso de combinación de publicaciones de NSGA-II . . . . .	26
2.9. Análisis del hipervolumen 2D . . . . .	28
2.10. Ejemplo aplicado del método de montecarlo . . . . .	28
2.11. Ejemplo hipervolumen en tres dimensiones . . . . .	28
2.12. Ejemplo montecarlo en 3 dimensiones . . . . .	28
2.13. Ejemplo de análisis de resultados . . . . .	29
3.1. Análisis de convergencia del problema en tres dimensiones ante cambios en el valor de los parámetros del optimizador . . . . .	33
3.2. Trayectorias del problema en tres dimensiones obtenidas con la configuración 1	34
3.3. Trayectorias del problema en tres dimensiones obtenidas con la configuración 2	34
3.4. Trayectorias del problema en tres dimensiones obtenidas con la configuración 3	34
3.5. Análisis de convergencia del problema en dos dimensiones ante cambios en el valor de los parámetros del optimizador . . . . .	35
3.6. Trayectorias del problema en dos dimensiones obtenidas con la configuración 1	36
3.7. Trayectorias del problema en dos dimensiones obtenidas con la configuración 2	36
3.8. Trayectorias del problema en dos dimensiones obtenidas con la configuración 3	37

3.9. Análisis de convergencia del problema en dos dimensiones con puntos de paso preseleccionados, ante cambios en el valor de los parámetros del optimizador	38
3.10. Trayectorias del problema en dos dimensiones con puntos de paso predefinidos obtenidas con la configuración 1	38
3.11. Trayectorias del problema en dos con puntos de paso predefinidos dimensiones obtenidas con la configuración 2	39
3.12. Trayectorias del problema en dos dimensiones con puntos de paso predefinidos obtenidas con la configuración 3	39
3.13. Análisis de convergencia del problema en tres dimensiones comparando la respuesta con tiempos absolutos frente a tiempos relativos	40
3.14. Trayectorias del problema 3D con una codificación de tiempo basada en tiempos relativos	41
3.15. Análisis de convergencia del problema en dos dimensiones comparando la respuesta con tiempos absolutos frente a tiempos relativos	41
3.16. Trayectorias del problema 2D con una codificación de tiempo basada en tiempos relativos	42
3.17. Análisis de convergencia del problema en dos dimensiones con puntos de paso preseleccionados, comparando la respuesta usando tiempos absolutos frente a tiempos relativos	43
3.18. Trayectorias del problema 2D con una codificación de tiempo basada en tiempos relativos	43
3.19. Análisis de convergencia del problema en tres dimensiones añadiendo tramos a la trayectoria	44
3.20. Trayectorias en dos dimensiones con 7 tramos	45
3.21. Análisis de convergencia del problema en dos dimensiones añadiendo tramos a la trayectoria	46
3.22. Trayectorias en dos dimensiones con 7 tramos	46
3.23. Análisis de convergencia del problema en dos dimensiones con puntos de paso predefinidos, añadiendo tramos a la trayectoria	47
3.24. Trayectorias en dos dimensiones con puntos de paso predefinidos con 7 tramos	47
3.25. Análisis de convergencia del problema en tres dimensiones para cada uno de los escenarios de simulación	49
3.26. Trayectorias generadas en el escenario B para el problema en tres dimensiones	49
3.27. Trayectorias generadas en el escenario C para el problema en tres dimensiones	49
3.28. Análisis de convergencia del problema en dos dimensiones para cada uno de los escenarios de simulación	50
3.29. Trayectorias generadas en el escenario B para el problema en dos dimensiones	50
3.30. Trayectorias generadas en el escenario C para el problema en dos dimensiones	51

3.31. Análisis de convergencia del problema en dos dimensiones con puntos de paso predefinidos, para cada uno de los escenarios de simulación . . . . .	51
3.32. Trayectorias generadas en el escenario B para el problema en dos dimensiones con puntos de paso predefinidos . . . . .	52
3.33. Trayectorias generadas en el escenario C para el problema en dos dimensiones con puntos de paso predefinidos . . . . .	52



# Índice de tablas

2.1. Parámetros del modelo ajustados . . . . .	14
2.2. Codificación de las variables de decisión del problema en 3D . . . . .	20
2.3. Codificación de las variables de decisión del problema en 2D . . . . .	21
2.4. Codificación de las variables de decisión del problema en 2D con puntos de paso predefinidos . . . . .	22
3.1. Configuraciones para el algoritmo NSGA-II . . . . .	31
3.2. Parámetros del sistema . . . . .	32



# Capítulo 1

## Introducción

Las cianobacterias son organismos que se encuentran comúnmente en masas de agua, como son los embalses. Los agrupamientos masivos de cianobacterias se denominan "*floraciones de agua*" o "*blooms algales*" Giannuzzi [2016]. Debido al cambio climático, el aumento de las temperaturas de las aguas hace que estas bacterias aumenten sus tasas de crecimiento, generando grandes floraciones que pueden producir altas concentraciones de unas toxinas dañinas para el ser humano Pérez et al. [2008]. Ante esta crítica situación, cabe la posibilidad de tratar los embalses con el fin de eliminar estas concentraciones de bacterias evitando la contaminación del agua. En el caso de no ser tratada a tiempo, ese agua no podría ser usada para el consumo ya que es perjudicial para el ser humano. Este hecho junto con las cada vez mas frecuentes sequías, hace que el análisis de las aguas de nuestros embalses sea una tarea indispensable para aprovechar el agua en su totalidad.

En la actualidad, la tarea de analizar el agua de los embalses se realiza en determinados puntos de medida. Esto provoca que en muchas ocasiones cuando se detecta un *bloom*, ya es demasiado tarde para tratarlo, haciendo que todo el agua contaminada no pueda ser empleada para su consumo hasta eliminar las toxinas. Es por ello que se deben explorar distintas alternativas para poder actuar a tiempo. Sin embargo, el comportamiento de estas bacterias dificulta mucho su localización ya que no siempre se encuentran en la superficie, donde se podrían llegar a localizar a simple vista.

En este trabajo se ofrece una alternativa para el seguimiento del estado del agua en grandes embalses, realizando medidas tanto superficiales como a determinadas profundidades con una embarcación automatizada. Las medidas se realizan en zonas donde se prevé una alta concentración de cianobacterias, a partir de una simulación que predice donde se encuentran estas grandes floraciones. Esto permite tanto validar la simulación realizada, como realizar un seguimiento del estado del agua, permitiendo así que se pueda tratar a tiempo.

## 1.1. Motivación

El agua es una de las materias primas más importantes para la supervivencia del ser humano, es por lo que es necesario cuidar la calidad del agua potable. Durante los últimos años este mensaje ha cobrado cada vez más importancia, todo ello debido al cambio climático, un problema contra el que todo el mundo debe luchar. El cambio climático trae consigo muchos problemas para la naturaleza y para el ser humano, entre ellos, el aumento de la temperatura del planeta y largos periodos de sequía, como se ha comprobado durante los últimos años. El aumento de las temperaturas hace que organismos como las cianobacterias proliferen en las grandes masas de agua como los embalses, y con ello su contaminación. Si a este hecho se le suman largas temporadas de sequía, hace que el agua para el consumo disponible disminuya, y en ocasiones no se pueda emplear por estar contaminada. Es por ello que en este trabajo se propone estudiar una alternativa para poder actuar a tiempo antes de que estos organismos dañinos para el agua de consumo, contaminen los embalses.

La alternativa que se detalla en este trabajo cumple una segunda función. Conociendo como es el comportamiento físico de los organismos que se encuentran dentro de grandes masas de agua, se puede crear un modelo predictivo que localice grandes concentraciones de bacterias, nutrientes etc. Este es un área de estudio muy relevante ya que con un buen modelo y un estudio previo de un embalse, se puede desarrollar un modelo con el cual tener en todo momento monitorizadas las grandes floraciones de cianobacterias, permitiendo así que se actúe antes de que estos *blooms* ocurran. Sin embargo, para poder desarrollar un modelo preciso es necesario tomar muchas medidas que verifiquen los resultados de los modelos. Es aquí donde entra la segunda función de este trabajo, ofrecer una herramienta capaz de tomar medidas que permitan validar un determinado modelo predictivo.

## 1.2. Antecedentes

Dada la necesidad de estudiar alternativas para prevenir las consecuencias del cambio climático y con vista a la agenda 2030 para el Desarrollo Sostenible de las Naciones Unidas United Nations [n.d.], se abre la puerta a la incorporación de nueva tecnología, como los vehículos autónomos con el fin de mejorar la calidad de las aguas.

Los vehículos autónomos han tomado una gran relevancia como campo de estudio en los últimos años, debido a la amplia gama de ámbitos en los que estos pueden ser usados. Dentro de la agricultura, estos vehículos se pueden emplear para siembra, riego o pulverización de pesticidas tal y como se recoge en Bechar and Vigneault [2016] con el vehículo de la figura 1.1. En exploración espacial, los vehículos autónomos cobran una especial importancia ya que realizan tareas de exploración de larga duración. En el ámbito de la navegación marítima tanto en superficie como submarina, pueden ser empleados para realizar cartografía

submarina, vigilancia costera, exploración o tareas de búsqueda y rescate. En general, los vehículos autónomos son muy útiles para realizar tareas repetitivas y que ponen en riesgo la vida del ser humano que la pudiera desempeñar. A su vez, el continuo desarrollo de tecnologías como la inteligencia artificial o el desarrollo de hardware y software cada vez más potente hace que las tareas que estos vehículos pueden realizar, sean cada vez más complejas. Esto sumado a los avances en la tecnología MEMS <sup>1</sup>, que hacen que los sistemas de navegación sean más asequibles, al reducir los costes de producción de estos sistemas, aumenta la viabilidad de incorporar estos vehículos en más sectores de la industria.



Figura 1.1: Vehículo autónomo para detección de hiervas, imagen extraída de Bechar and Vigneault [2016]

El grupo de investigación en Ingeniería de Sistemas, Control, Automatización y Robótica (ISCAR) de la Universidad Complutense de Madrid (UCM), ha estado trabajando en la automatización de diversos tipos de vehículos de superficie durante los últimos 25 años. En sus inicios, los esfuerzos de ISCAR en este campo se centraron en desarrollar controladores que garanticen el correcto funcionamiento de los ASVs, a pesar de las perturbaciones provocadas por factores externos como las mareas o el viento. Posteriormente, el grupo se ha enfocado en el desarrollo de ASVs y los algoritmos necesarios para su aplicación en distintos campos, como la búsqueda y rescate de naufragos, la contención de cianobacterias, o la monitorización de la calidad del agua. De entre los vehículos desarrollados por el grupo de investigación se encuentran los mostrados en las figuras 1.2, ambas imágenes extraídas de la web del grupo de investigación <http://www.dacya.ucm.es/area-isa/media/imagenes.html>.

El grupo de investigación ISCAR realizó un vehículo enfocado especialmente en la exploración de embalses para realizar medidas relacionadas con la concentración de cianobacterias. Este vehículo es el presentado en la figura 1.3 cuyo guiado y control se expone en el trabajo González Calvin [2023], donde se propone como método de guiado el método GVF (guidance vector field). En paralelo, este grupo de investigación, desarrolla algoritmos de planificación de trayectorias con los fines anteriormente mencionados.

---

<sup>1</sup>MEMS (MicroElectroMechanical Systems), se trata de una tecnología que incorpora elementos mecánicos móviles y elementos electrónicos a escala microscópica

Hasta la actualidad, los estudios realizados acerca de la planificación trayectorias siguen dos caminos diferenciados. Por un lado, el estudio se ha centrado en realizar trayectorias con el objetivo de maximizar la exploración de una determinada zona, tal y como se realiza en el artículo Arzamendia et al. [2019], donde se trata de resolver el problema conocido como el “Problema del Vendedor Viajero” (TSP<sup>2</sup>), con la diferencia de que en este caso se trata de maximizar el área recorrida por el vehículo dentro de un lago. Por otro lado, también se han realizado estudios donde se plantea una planificación de trayectorias con un conocimiento previo del entorno, en Xiong et al. [2019] desarrolla una nueva técnica de optimización por colonia de hormigas que se basa en los diagramas de Voronoi para encontrar trayectorias óptimas que maximicen la recolección de datos. Dentro de este último grupo de estudio, cabe destacar la aportación realizada en Carazo-Barbero et al. [2023] y González-Calvin et al. [2023]. En el primer artículo, se plantean trayectorias en tres dimensiones para una embarcación autónoma, el algoritmo empleado permite modificar de forma dinámica, valores de la configuración de la trayectoria, conociendo previamente las concentraciones de cianobacterias en el embalse, todo ello con el fin de maximizar la medida referente a la concentración de estas bacterias. Esto mejora notablemente la trayectoria generada, sin embargo, aumenta el coste computacional. El segundo artículo, se plantea un planificador en dos dimensiones que realiza una medida a una profundidad fija, salvo en los puntos de paso de la trayectoria donde realiza un perfil de medida. Ambos artículos emplean el algoritmo de optimización multiobjetivo NSGA-II, pero para simular el comportamiento de la dinámica del fluido emplean distintos simuladores. Empleando las restricciones, funciones objetivo y las simulaciones de fluido presentadas en Carazo-Barbero et al. [2023], en este TFM se estudiarán nuevos enfoques y se tratará de unificar estos dos últimos trabajos, con el fin de obtener un planificador de trayectorias capaz de crear soluciones óptimas con menor carga computacional, reduciendo así el tiempo de ejecución.

---

<sup>2</sup>Este problema consiste en encontrar la ruta más corta posible que permita a un viajero visitar un conjunto de puntos y regresar al punto de origen



Figura 1.2: Vehículos autónomos desarrollados por el grupo ISCAR para exploración de entornos marítimos



Figura 1.3: Vehículo de superficie Boyi para detección de concentraciones de cianobacterias desarrollado por ISCAR

Reducir el tiempo de ejecución de la tarea de planificación es una tarea muy importante, ya que cuanto menor sea el tiempo necesario desde que se realiza una trayectoria con sus respectivas medidas, y la siguiente trayectoria, permitirá a la empresa que gestione el embalse mejorar el seguimiento del estado del agua, mejorando de ese modo la calidad del agua destinada al consumo humano.

### 1.3. Propuesta y objetivos

El trabajo propuesto consiste en desarrollar un planificador de trayectorias, basado en algoritmos genéticos, para un vehículo de superficie, que partiendo de una simulación de la dinámica del movimiento y generación de cianobacterias en un embalse, sea capaz de realizar una trayectoria que pase sobre las zonas donde se estima máxima la concentración. A su vez, el planificador debe ser lo más realista posible, de modo que, además de la dinámica del fluido y de las cianobacterias debe tener en cuenta restricciones de movimiento del vehículo que desempeña las tareas de navegación.

Con el fin de obtener un buen planificador de trayectorias, se estudiarán distintas soluciones al problema, ofreciendo comparativas entre distintas versiones del planificador para poder así discernir cual es la mejor alternativa al método de muestreo del agua tradicional.

Con la propuesta que se realiza para este trabajo se establecen los siguientes objetivos:

1. Estudio y comprensión de la dinámica de partículas que se emplea como modelo predictivo de cianobacterias.
2. Definir el problema de optimización multi-objetivo para la planificación de trayectorias del vehículo de superficie.

3. Estudio y aplicación del algoritmo genético para problemas multi-objetivo NSGA-II.
4. Resolver el problema de optimización empleando trayectorias en dos y tres dimensiones.
5. Resolver el problema de optimización empleando puntos de paso asignados con anterioridad a la ejecución del algoritmo.
6. Analizar y comparar los resultados obtenidos con los métodos desarrollados.
7. Analizar cómo los cambios en los parámetros del algoritmo genético afectan a los resultados.

Para realizar este trabajo se partirá de las aportaciones realizadas en Carazo-Barbero et al. [2023], donde se presenta un planificador de trayectorias en tres dimensiones para realizar medidas de concentración de cianobacterias basándose en un modelo de comportamiento de las mismas. Sin embargo, este planificador está programado en MATLAB, esto hace que no sea posible integrarlo en algunos sistemas. Por otro lado, se encuentra el trabajo realizado en González-Calvin et al. [2023], el cual ofrece un nuevo enfoque para resolver el problema en dos dimensiones. Este último está desarrollado en Python empleando la librería Klein et al. [2021] para resolver el problema de optimización, haciendo que sea posible su incorporación en cualquier sistema, pero este cuenta con un alto coste computacional. La idea consiste en unificar estos dos trabajos empleando las herramientas de python presentadas por González-Calvin et al. [2023] con las restricciones y funciones objetivo presentadas en Carazo-Barbero et al. [2023], replicando una de las resoluciones del problema en tres dimensiones y redefiniendo el concepto del problema en dos dimensiones, aportando a su vez un nuevo método para incorporar puntos de paso obligatorios en el planificador de dos dimensiones.

## 1.4. Estructura del documento

Este trabajo comienza con estudio teórico del modelo dinámico de partículas que se emplea como modelo predictivo de concentraciones de cianobacterias. Teniendo en cuenta los desplazamientos de las partículas provocados por las propias cianobacterias en función de la irradiación solar, y por el propio fluido, lo que hace que estas partículas se desplacen en tres dimensiones dentro del embalse, y considerando la proliferación de estas partículas, este modelo es capaz de, dadas unas condiciones iniciales (dinámica del fluido, densidad y ubicación inicial de las cianobacterias) ubicar las partículas en el espacio-tiempo, permitiendo así calcular donde se encuentran las altas concentraciones de las cianobacterias.

Conocido el comportamiento dinámicos de los elementos básicos que dan forma al problema, en la sección 2.2, se detallan distintas descripciones y planteamientos para el caso propuesto. Todo problema que se pretende resolver, tiene múltiples formas de afrontarlo,

teniendo cada una de ellas sus ventajas y desventajas. Es por ello que en esta sección se explicarán las ventajas e inconvenientes de cada una de las descripciones propuestas. Dado que el problema se resolverá mediante el uso de un algoritmo genético, en esta sección se expondrá la definición formal del problema de optimización, explicando las funciones objetivo y restricciones empleadas.

Con el problema ya definido, en la sección 2.4 se explica el funcionamiento del algoritmo genético empleado para la resolución del problema, el algoritmo NSGA-II. Con el fin de evaluar las soluciones generadas mediante este algoritmo, en la sección 2.5.1 se detallan los métodos para el análisis de las estas soluciones.

Una vez se expuestos los conocimientos teóricos necesarios para la descripción del problema y para su resolución, en el capítulo 3 se presentan los resultados obtenidos para cada uno de los problemas propuestos, también, se realizarán comparativas entre las distintas descripciones del problema y se analizará como afectan a estas soluciones las modificaciones de los parámetros que intervienen en el algoritmo genético empleado.

Para finalizar el trabajo, en el capítulo 4 se presentan las conclusiones extraídas del trabajo y de los resultados. Con el fin de abrir la vía de colaboración en la investigación, se propondrá una línea de trabajo por la que continuar este proyecto.



# Capítulo 2

## Fundamentos y metodología

Dada la importancia que tiene realizar un correcto y recurrente seguimiento del estado del agua, se desea mediante el uso de algoritmos genéticos, generar trayectorias que maximicen la posibilidad de realizar medidas en zonas de elevada concentración de cianobacterias, partiendo de un modelo predictivo de concentraciones de cianobacterias y la dinámica del fluido del que se compone el embalse. En este capítulo se detallan las bases teóricas necesarias para abordar el problema y comprender los resultados obtenidos, y las herramientas necesarias para obtener dichos resultados. Como material teórico se describirá el modelo predictivo de cianobacterias. También se dará una descripción detallada del problema que se desea resolver. Se finalizan con las herramientas que nos permiten resolverlo y los métodos utilizados para analizarlo.

### 2.1. Modelo predictivo de cianobacterias

El primer paso para crear un buen optimizador de trayectorias, es comprender como es la dinámica de los organismos que se desean seguir. El comportamiento de las cianobacterias no es un problema que resulte fácil de analizar, ya que la posición de estos organismos se ve afectada por distintos factores. En primer lugar, se ven desplazados por las corrientes internas de los embalses, lo que les proporciona un movimiento en tres dimensiones. También, se ven afectadas por la intensidad de luz que recibe el agua, necesaria para la supervivencia y extensión de la colonia de cianobacterias, lo que provoca que las partículas se desplacen en el plano vertical del embalse. A su vez, debe tenerse en cuenta el crecimiento biológico de dicho organismo, que depende, entre otros factores, de la luminosidad que recibe el embalse, haciendo que una colonia aumente o disminuya su población. Esto hace que la concentración de cianobacterias dependa de cuatro variables, posición en los ejes  $x$ ,  $y$ ,  $z$  y el tiempo, ecuación (2.1).

$$C_c = f(x, y, z, t) \quad (2.1)$$

Todos los detalles del modelo predictivo que se presentan en el resto de esta sección para comprender el comportamiento de las cianobacterias, así como las simulaciones de las cianobacterias utilizadas para evaluar los resultados del planificador, forman parte del trabajo realizado por el doctorando Gonzalo Carazo para su tesis, publicado parcialmente en Carazo [2023].

### 2.1.1. Dominio del embalse

El primer elemento que debe ser descrito es el dominio físico del embalse donde de va a desarrollar toda la actividad y movimiento de las cianobacterias. Entendemos como dominio físico las dimensiones del embalse, en los casos evaluados en este TFM es el Lago Mendota, Wisconsin, EEUU, que se encuentra representado en la Figura 2.1. Para obtener datos de las dimensiones del embalse se emplean herramientas CAD<sup>1</sup>, obteniéndose un área de superficie de  $1,43 \times 10^6 \text{ m}^2$ , un volumen de  $6,61 \times 10^7 \text{ m}^3$  y una profundidad máxima de  $70,79 \text{ m}$ . Como no se tienen datos reales de las profundidades de todo el embalse, se simplificará el modelo usando un fondo plano igual a la profundidad máxima. Esto no será un problema ya que las cianobacterias no se suelen situar por debajo de la termo-clima que suele encontrarse a unos 15 metros, Hernández-Becerril and Pastén-Miranda [2015]. Teniendo los vértices que componen los límites del embalse, se obtiene la representación 2D de la figura 2.2. Cabe destacar que en la representación 2D se tiene en cuenta las dimensiones del embalse limitado a la zona donde es seguro navegar, es decir, no tiene en cuenta las zonas donde no es seguro navegar con el vehículo o simplemente aquellas zonas que carecen de interés para ser exploradas.

### 2.1.2. Dinámica del fluido

El siguiente elemento a tener en cuenta es la dinámica del fluido del embalse. Para ello se emplea el software COMSOL Multiphysics COMSOL. Este software, partiendo del modelo CAD del embalse y la tasa de flujo laminar de entrada y salida del embalse, permite obtener las velocidades y aceleraciones del fluido en el embalse, figura 2.3. Donde se puede apreciar el movimiento del fluido en la superficie, representando la velocidad del fluido en cada punto con distintos tonos de colores, y las entradas y salidas de agua con las flechas de color rojo.

---

<sup>1</sup>Computer-Aided Design: Conjunto de herramientas software para diseño asistido por ordenador

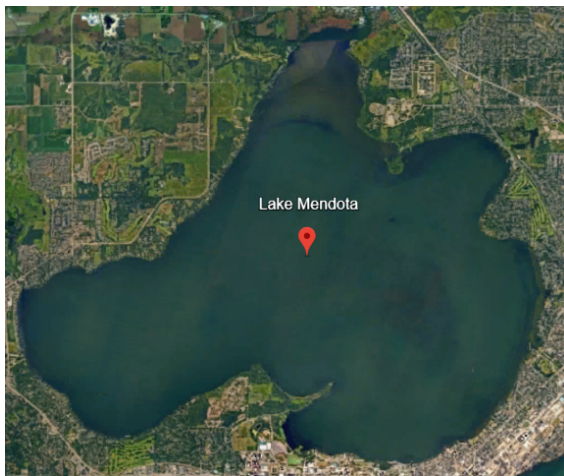


Figura 2.1: Lake Mendota, Wisconsin, EE. UU. Imagen de Google Earth

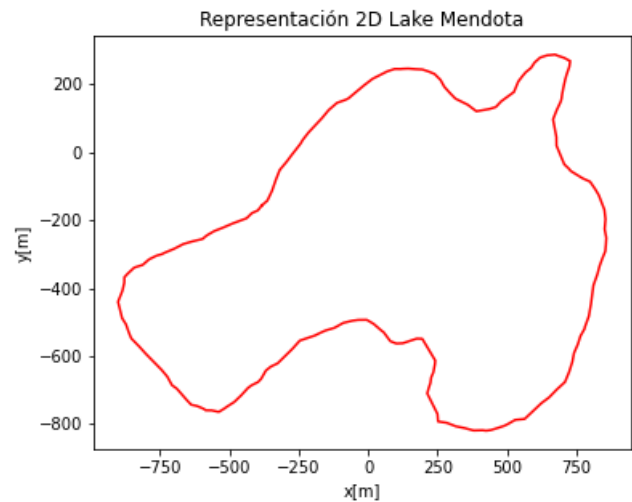


Figura 2.2: Representación 2D Lake Mendota

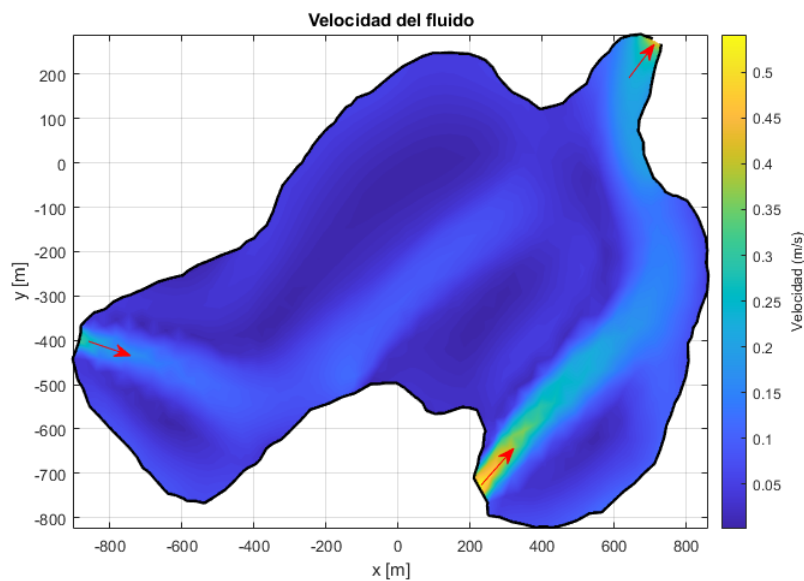


Figura 2.3: Solución de la simulación de COMSOL

### 2.1.3. Dinámica del organismo

Tras haber obtenido la información 3D de la velocidad del agua, es el momento de explicar el modelo predictivo de la evolución espacio temporal de los afloramientos de cianobacterias. Este modelo se basa en un modelo de partículas, donde las partículas son las cianobacterias, que tiene en cuenta los siguientes elementos:

- Densidad de la colonia de cianobacterias
- Crecimiento biológico de la colonia

- Transporte del organismo
- Concentración de la colonia de cianobacterias

En primer lugar, se describe el modelado de la densidad de la colonia de cianobacterias, que depende de la intensidad de luz, por lo tanto, es necesario dar una descripción del valor de intensidad de luz en profundidad  $z$  y en tiempo  $t$ . El modelo utilizado considera que la intensidad de la luz aumenta hasta el medio día, donde alcanza un valor máximo  $I_{max}$ , y que una vez pasa el medio día este valor decrece. Una forma de modelar este comportamiento sería empleando un seno con una amplitud  $I_{max}$ , una frecuencia de  $\frac{\pi}{15}$  y un desfase de  $-\frac{\pi}{3}$ . Sin embargo, la intensidad de luz no toma valores negativos, y se ve afectada también por la profundidad dentro del agua, pudiendo ser representado esto último con como una exponencial negativa. El modelo utilizado se muestran en la ecuación (2.2), los resultados de su simulación son representados en la figura 2.4.

$$I(t, z) = \begin{cases} I_{max} \sin\left(\frac{\pi}{15}t - \frac{\pi}{3}\right) e^{0,5z} & \text{para } 24n + 5 \leq t \leq 24n + 20 \text{ con } n = 0, 1, 2, \dots \\ 0 & \text{c.c} \end{cases} \quad (2.2)$$

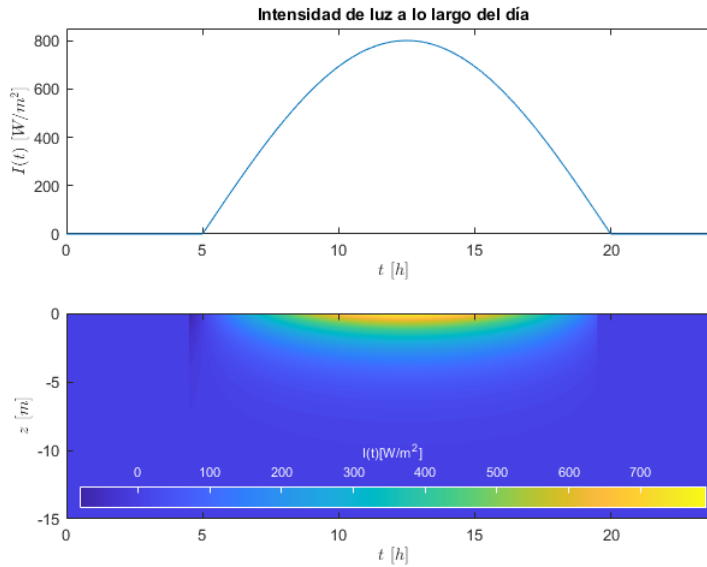


Figura 2.4: Intensidad de luz en función del tiempo y profundidad

La variación de la densidad de las células que componen la colonia en función de la intensidad de la luz se modela con la ecuación (2.3)

$$\frac{\partial}{\partial t} \rho_{cell}^k(t) = \begin{cases} \beta I^k e^{-\frac{k}{I_0}} + \gamma & \text{para } I(z, t) \geq I_{lim} \\ -\alpha \Delta_\rho H(\Delta_\rho) & \text{para } I(z, t) < I_{lim} \end{cases} \quad (2.3)$$

donde  $\beta$  es el factor normativo de la densidad de las células,  $I_0$  el límite de foto-inhibición,  $\gamma$  es el cambio medio en condiciones de oscuridad,  $\alpha$  el factor de decrecimiento de la densidad de las células,  $H(\cdot)$  la función escalón,  $\Delta_\rho = \rho_{cell}^k - C_p$ , con  $C_p$  siendo el mínimo de densidad de una célula y  $k$  identifica la partícula. Como se puede apreciar, se distinguen dos situaciones, una situación de oscuridad siendo  $I(t, z) < I_{lim}$  donde la densidad de las células decrece rápidamente, y una situación de iluminación  $I(t, z) \geq I_{lim}$  donde la densidad aumenta.

Con la densidad de las células calculadas, se puede calcular la densidad de las colonias de cianobacterias siguiendo la expresión (2.4).

$$\rho_p^k(t) = \rho_{cell}^k(t)n_{cell}(1 - n_{gas}) + \rho_{muc}(1 - n_{cell}) \quad (2.4)$$

donde  $n_{cell}$  es la proporción de la colonia formada por células,  $n_{gas}$  la proporción de la colonia formada por gases,  $\rho_{muc}$  la densidad de mucílago, una sustancia gelatinosa que se encuentra alrededor de la célula, y el índice  $k$  identifica la partícula.

El siguiente paso es determinar el crecimiento biológico de estos organismos, el cual se modela con la expresión (2.5), donde  $r_\Delta^k$  es un número aleatorio con una distribución gaussiana con una media igual al crecimiento por unidad de tiempo esperado y una varianza inversa al tiempo necesario para renovar completamente la población, y  $s_\Delta^k \in [0, 1]$  un número aleatorio con distribución uniforme. Como resultado se obtendrá el número de partículas generadas por el crecimiento biológico, o en su defecto (si el valor es negativo), las partículas que fallecen en la colonia.

$$N_{new}^k = trunc(r_\Delta^k) + sign(r_\Delta^k)H(s_\Delta^k < mod(|r_\Delta^k|, 1)) \quad (2.5)$$

Dado que los organismos no se mantienen estáticos dentro del embalse, también se debe modelar su movimiento dentro del fluido, para ello se emplea la expresión (2.6). Este modelo da información acerca de la aceleración de cada partícula en los ejes  $(x, y, z)$  provocada por las corrientes y el cambio de densidad de la colonia, con esta información, mediante integración, se pueden obtener las velocidades y las posiciones de las partículas.

$$\left(1 + \frac{1}{2} \frac{\rho_f}{\rho_p^k}\right) \frac{d\mathbf{u}_p^k}{dt} = \left(1 + \frac{\rho_f}{\rho_p^k}\right) \mathbf{g} + \frac{18\nu\rho_f}{d_p^2\rho_p^k}(\mathbf{u}_{ss} - \mathbf{u}_p^k) + \frac{3}{2} \frac{\rho_f}{\rho_p^k} \frac{d\mathbf{u}_{ss}}{dt} + D_0 r_D \quad (2.6)$$

donde  $\rho_f$  es la densidad del agua,  $\rho_p$  es la densidad de la célula,  $\frac{d\mathbf{u}_p^k}{dt}$  es la aceleración de la partícula,  $\mathbf{g}$  es la gravedad como vector  $[0, 0, g]$ ,  $d_p$  el diámetro de la partícula,  $u_{ss}$  la velocidad del fluido,  $u_p$  la velocidad de la partícula,  $\frac{d\mathbf{u}_{ss}}{dt}$  la aceleración del fluido,  $D_0$  el coeficiente de difusión,  $r_D$  un vector de tres dimensiones aleatorio con media  $[0, 0, 0]$  y varianza  $[1, 1, 1]$ , y  $\nu$  la viscosidad cinemática del fluido.

Conocidas las posiciones de cada una de las partículas, se puede calcular la concentración de cianobacterias en cualquier punto de la masa de agua en el espacio-tiempo recurriendo a la

expresión 2.7, donde  $A_c = \frac{1}{(2\pi)^{3/2}\sigma_{c,x}\sigma_{c,y}\sigma_{c,z}}$ . Como se puede apreciar, la concentración se modela como la contribución de cada partícula en una determinada posición en un determinado tiempo.

$$C_c(x, y, z, t) = \sum_{k=1}^{N_p(t)} A_c \exp \left( - \sum_{d \in \{x,y,z\}} \frac{(d - p_d^k(t))^2}{2\sigma_{c,d}^2} \right) \quad (2.7)$$

### 2.1.4. Simulaciones

Todas las expresiones y simulaciones mostradas parten del artículo Carazo-Barbero et al. [2023], donde se emplean los parámetros de la tabla 2.1 para obtener las simulaciones de la figura 2.5. Para lograr cada uno de los escenarios se han generado de forma aleatoria un determinado número de partículas, 500 partículas dispersas según una gaussiana de desviación estándar [25, 25, 0'5] en torno al punto (0, 0, -2) para el escenario A, 250 en torno al punto (230, -365, -2) con una desviación estándar [25, 25, 1] para el escenario B, y 500 centradas en (-930, -400, -1) con una desviación estándar [25, 25, 0'5] para el escenario C.

Tabla 2.1: Parámetros del modelo ajustados

Parámetro	Valor	Parámetro	Valor
$\alpha$	$4,7 \times 10^{-4} s^{-1}$	$n_{gas}$	5 %
$\beta$	$1,5 \times 10^{-3} s^2/m^3$	$\rho_{muc}$	$998 kg/m^3$
$\gamma$	$-8,3 \times 10^{-3} kg/m^3 s$	$\rho_f$	$1000 kg/m^3$
$C_p$	$1010 kg/m^3$	$v$	$2,5 \times 10^{-4} m^2/s$
$I_{lim}$	$5,75 W/m^2$	$d_p$	$8 \times 10^{-4} m$
$I_0$	$146,43 W/m^2$	$D_0$	$1 m/s^2$
$n_{cell}$	10 %	$\sigma_c$	[25, 25, 0,5]m

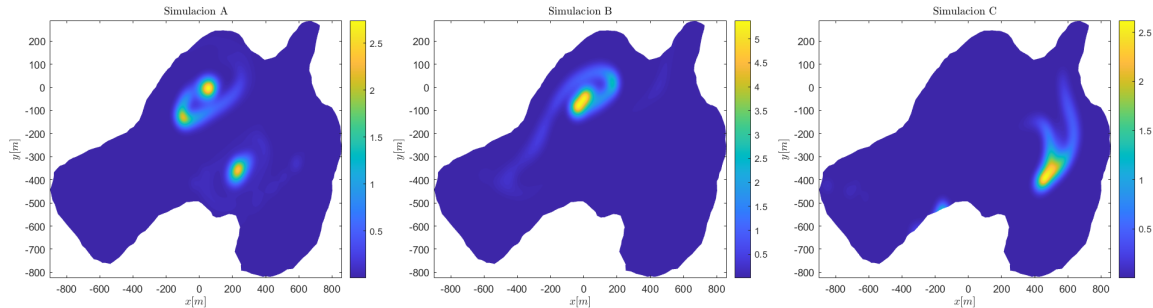


Figura 2.5: Escenarios de simulación

Para obtener esta representación de la figura 2.5, se calcula la concentración en cada punto del embalse proyectando el eje z sobre el plano de la superficie del agua. De este modo se puede representar el embalse en dos dimensiones y añadir una tercera dimensión, la concentración, que se representa con distintos colores en función de la cantidad de concentración en

cada punto. En este trabajo, todas las gráficas que muestran el dominio del embalse tienen unidades de metros en sus ejes x e y.

## 2.2. Descripción formal del problema de planificación de trayectorias

En esta sección se dará una descripción matemática del problema a resolver. En primer lugar se plantea el problema de forma general, mostrando las restricciones y funciones objetivo que se van a emplear para optimizar la trayectoria del vehículo de superficie. Dadas las restricciones y funciones objetivo, se explicarán distintas variantes o enfoques del problema, lo que dará lugar a nuevas descripciones, todo ello con la finalidad de obtener el mejor optimizador posible para determinadas situaciones.

El optimizador, parte de la información del modelo de cianobacterias explicado en la sección 2.1.3. Con este modelo se obtiene la concentración de las cianobacterias en toda la masa de agua en cada instante de tiempo. Con este conocimiento, se desea realizar una trayectoria para que la sonda situada en el vehículo de superficie tome medidas en localizaciones con concentraciones de cianobacterias significativas. Sin embargo, el vehículo tiene limitaciones derivadas de la duración de la batería, por lo que debe realizar la trayectoria de longitud reducida en un tiempo breve. Debido a las limitaciones por batería se debe exigir un límite de tiempo de duración de la misión, pero además, existen limitaciones provocadas por la dinámica del vehículo, como son la velocidad, la aceleración y la curvatura que puede trazar. Por otro lado, el sistema motorizado que se encarga de subir y bajar la sonda, tiene sus propias limitaciones en velocidad, aceleración y profundidad.

Las restricciones y funciones objetivo detalladas en esta sección parten del trabajo Carazo-Barbero et al. [2023]. A partir de estas restricciones y funciones objetivo, se plantearán modificaciones sobre su planteamiento del problema y se empleará conceptos extraídos de González-Calvin et al. [2023], con el fin de unificar estos dos trabajos y lograr un planificador que reduzca el coste computacional.

### Restricción en tiempo de misión

Una de las restricciones necesarias para el correcto funcionamiento del vehículo durante toda la trayectoria es la duración, que se va a limitar debido al consumo de la batería.

$$CF_1 = \sup(0, t_e - t_0 - T_{max}) \quad (2.8)$$

donde  $t_e$  es el tiempo de duración de la misión,  $t_0$  el tiempo de inicio y  $T_{max}$  el tiempo máximo permitido de misión.

### Restricción en la velocidad horizontal del vehículo

Debido a la características del vehículo, este podrá desplazarse a una velocidad máxima. Además, puede ser interesante limitar esta velocidad por mejorar la duración de la batería. Dado que las corrientes internas del embalse pueden mover la sonda, no conviene que el vehículo se desplace a altas velocidades por el desplazamiento del agua, es por ello que se añade también la velocidad del fluido.

$$CF_2 = \sup(0, \sup(\|\dot{s}_H(t_j) - u_{ss,H}(t_j)\|) - v_{max,H}) \quad (2.9)$$

donde  $\dot{s}_H(t_j)$  es la velocidad horizontal  $(x, y)$  del vehículo,  $u_{ss,H}(t_j)$  es la velocidad horizontal  $(x, y)$  del fluido, y  $v_{max,H}$  es la velocidad máxima permitida del vehículo.

### Restricción en la aceleración del vehículo

De nuevo debido a las características del vehículo, la aceleración puede verse limitada, pero también es interesante tener controlada la aceleración que sufre, debido a la propia aceleración del fluido que pueden hacer que el vehículo no sea estable, una aceleración del vehículo junto con la aceleración del agua puede hacer que este vuelque.

$$CF_3 = \sup\left(0, \sup\left(\left\|\ddot{s}_H(t_j) - \frac{du_{ss,H}(t_j)}{dt}\right\|\right) - a_{max,H}\right) \quad (2.10)$$

donde  $\ddot{s}_H$  es la aceleración horizontal  $(x, y)$  del vehículo,  $\frac{Du_{ss,H}}{Dt}$  la aceleración horizontal  $(x, y)$  del fluido y  $a_{max,H}$  la aceleración máxima permitida.

### Restricción en la curvatura de la trayectoria

La curvatura que puede realizar el vehículo puede estar también limitada, es por ello que se añade esta restricción, con el fin de que el vehículo pueda trazar correctamente las trayectorias designadas. Un vehículo estilo catamarán permite girar alrededor de su propio eje, sin embargo, un vehículo con motor fueraborda tiene un ángulo de giro limitado.

$$CF_4 = \sup\left(0, \sup\left(\frac{|\dot{s}_x(t_j)\ddot{s}_y(t_j) - \dot{s}_y(t_j)\ddot{s}_x(t_j)|}{(\dot{s}_x^2(t_j) + \dot{s}_y^2(t_j))^{3/2}}\right) - k_{max}\right) \quad (2.11)$$

donde  $k_{max}$  es la curvatura máxima que puede tener la trayectoria para que sea realizable por el vehículo.

### Restricción en la velocidad vertical de la sonda

La limitación en la velocidad vertical de la sonda esta causada por las características del motor que se emplea para subir y bajar la sonda, este es un factor importante si se desea

realizar una medida continua a lo largo de la trayectoria. A su vez, es interesante tener controlada la velocidad de subida y de bajada de la sonda en el caso de realizar un perfil en profundidad de la sonda en unos puntos determinados de la masa de agua, ya que es necesario bajar a una determinada velocidad, por lo que debe estar limitada.

$$CF_5 = \sup(0, \sup(|\dot{s}_z(t_j)|) - v_{max,V}) \quad (2.12)$$

donde  $\dot{s}_z$  es la velocidad vertical de la sonda y  $v_{max,V}$  la velocidad máxima permitida de la sonda.

### Restricción en la aceleración vertical de la sonda

Por la propia construcción del sistema de subida/bajada de la sonda, este valor se verá limitado, es por ello que se emplea esta restricción.

$$CF_6 = \sup(0, \sup(|\ddot{s}_z(t_j)|) - a_{max,V}) \quad (2.13)$$

donde  $a_{max,V}$  es la aceleración vertical máxima permitida para el desplazamiento de la sonda.

### Maximización de la medida de concentración de cianobacterias

Esta es la función objetivo más importante ya que será la que provoque que la trayectoria pase por los puntos de máxima concentración de cianobacterias. Sin embargo se plantean una serie de problemas, entre ellos que se trata de un problema dinámico, donde las partículas están en constante movimiento, lo que quiere decir que el vehículo debe ser capaz de moverse allá donde estas concentraciones se muevan sin estar midiendo continuamente la misma zona o el mismo conjunto de partículas. Es por ello que se plantea esta función objetivo, donde se evalúa la aportación de cada partícula en cada instante de tiempo de modo que al optimizar, se toma aquel momento en el que la aportación es máxima, es así como se consigue no medir la misma concentración.

$$\max OF_1 = \sum_{k=1}^{N_p(t)} \sup_{t_j \in [t_0:T_s:t_e]} \left[ A_c \exp \left( - \sum_{d \in \{x,y,z\}} \frac{(s_d(t_j) - p_d^k(t_j))^2}{2\sigma_{c,d}^2} \right) \right] \quad (2.14)$$

donde  $N_p(t)$  es el número total de partículas a lo largo del tiempo,  $A_c = \frac{1}{(2\pi)^{3/2} \sigma_{c,x} \sigma_{c,y} \sigma_{c,z}}$ ,  $s_d(t_j)$  es la posición del vehículo,  $p_d^k$  es la posición de la partícula  $k$ ,  $\sigma_{c,d} = [25, 25, 0, 5]$ ,  $t_0$  el tiempo inicial desde el que parte la trayectoria,  $T_s$  el periodo de muestreo de la simulación y  $t_e$  el tiempo final en el que termina la trayectoria.

### Minimización de la longitud de la trayectoria

Para maximizar la concentración de cianobacterias medida, se podría realizar un barrido de todo el embalse, de forma que se midiese toda la concentración existente. Sin embargo, esto no es útil ya que estos organismos se concentran en determinadas posiciones del embalse, y el hacer un barrido de todo el embalse es costoso en tiempo. Por ello se plantea como función objetivo la longitud de la trayectoria con el fin de hacerla siempre lo más corta posible (siempre y cuando, se optimicen los otros objetivos).

$$\min OF_2 = \int_{t_0}^{t_e} \|\dot{s}(t)\| dt \quad (2.15)$$

donde  $t_e$  es el tiempo de duración de la trayectoria.

### Minimización del tiempo de la misión

Dada la importancia de reducir el consumo de las baterías, y con el objetivo de no obtener unas trayectorias con un tiempo de duración más largo de lo necesario, se plantea esta función objetivo.

$$\min OF_3 = (t_e - t_0) \quad (2.16)$$

## 2.3. Planificadores de trayectorias

La idea del planificador de tres dimensiones, consiste en realizar la trayectoria sin detenerse mientras que se realizan medidas con la sonda de forma continuada. Esto implica modificar la profundidad a la que se encuentra la sonda mientras el vehículo se desplaza a lo largo de la trayectoria. Este planteamiento permite explorar las zonas con más precisión ya que el optimizador tratará de llegar al punto de máxima concentración y no la zona de máxima concentración como ocurre con los siguientes métodos. Para realizar esta labor, se emplean todas las funciones objetivo y restricciones comentadas anteriormente ya que están diseñadas para la resolución en tres dimensiones. Para reducir la complejidad computacional que supone el planificador de Carazo-Barbero et al. [2023], se plantea emplear una codificación no dinámica del problema, de forma más específica, no se altera el número de variables de decisión mientras se realiza la optimización.

Otra forma de resolver el problema pasa por emular los métodos que se emplean en la actualidad para estimar la concentración de cianobacterias en un embalse. Este método consiste en trasladarse a distintos puntos del embalse en los cuales se realiza un perfil vertical de medidas referentes a la concentración de cianobacterias. Esta labor puede ser sustituida por el vehículo realizando trayectorias como en el caso anterior, pero deteniéndose en los puntos de paso, donde se realiza el perfil vertical de medidas. Tal y como se realiza en el

trabajo González-Calvin et al. [2023], con la diferencia de que en este caso solo se considerarán las medidas tomadas en los puntos especificados y no a lo largo de la trayectoria.

Se va a presentar una modificación sobre el planificador 2D presentado en el caso anterior, de modo que antes de realizar la optimización, se da opción al operador de introducir en el programa una lista de puntos de paso por los que es imperativo que el vehículo pase. Este enfoque es muy interesante, ya que en muchos embalses se tiene un histórico donde puede darse el caso en el que exista una determinada zona donde habitualmente hay una gran concentración de cianobacterias, es por ello que a pesar de que el modelo de cianobacterias determine que en dichas zonas no existe una alta concentración, es necesario tenerlas vigiladas. Para tener en cuenta estos datos históricos, se modifica la definición de las variables de decisión del planificador 2D.

Todos los planteamientos de planificadores de trayectorias presentados emplean la codificación de trayectorias mediante splines cúbicos. Esta es una técnica de interpolación muy utilizada para planificación de trayectorias para vehículos autónomos. La construcción de un spline cúbico implica dividir un conjunto de datos, en intervalos con puntos de paso. En cada intervalo, se define un polinomio cúbico que no solo se ajusta a los puntos de paso, sino que también asegura que la transición entre intervalos sea suave y continua. Este método presenta la ventaja de un bajo coste computacional, pero no se puede controlar la curvatura de la trayectoria planificada, como si se puede hacer empleando el método de las curvas de Hermite. Sin embargo, se escoge este método ya que en el propio planificador ya se contempla limitada la curvatura gracias a la restricción presentada en la sección 2.2.

Para definir un spline cúbico solo es necesario definir los puntos de paso y el tiempo en el que son alcanzados. Para cada intervalo entre dos puntos de paso consecutivos, se construye un polinomio cúbico. Si los puntos de paso son  $(x_i, y_i)$  y  $(x_{i+1}, y_{i+1})$  con tiempos  $t_i$  y  $t_{i+1}$ , entonces para el intervalo  $[t_i, t_{i+1}]$  la función spline  $S_i(t)$  es un polinomio cúbico en  $t$  el cual se puede apreciar en la ecuación 2.17.

$$S_i(t) = a_i(t - t_i)^3 + b_i(t - t_i)^2 + c_i(t - t_i) + d_i \quad (2.17)$$

donde  $a_i$ ,  $b_i$ ,  $c_i$  y  $d_i$  son los coeficientes que deben determinarse.

También se debe asegurar que el spline pase por cada punto de paso, lo que significa que  $S_i(t_i) = y_i$  y  $S_i(t_{i+1}) = y_{i+1}$ . La pendiente de la spline debe ser continua en los puntos de paso, lo que implica que la primera derivada de  $S_i(t)$  en  $t_{i+1}$  debe coincidir con la primera derivada de  $S_{i+1}(t)$  en  $t_{i+1}$ . Por último, la curvatura de la spline también debe ser continua en los puntos de paso, es decir, la segunda derivada de  $S_i(t)$  en  $t_{i+1}$  debe coincidir con la segunda derivada de  $S_{i+1}(t)$  en  $t_{i+1}$ . El spline queda definido cuando se resuelven los coeficientes  $a_i$ ,  $b_i$ ,  $c_i$  y  $d_i$ , mediante sistema de ecuaciones basado en las condiciones de continuidad.

### 2.3.1. Planificador de trayectorias 3D

Las variables de decisión de este problema son los puntos de paso de la trayectoria y los tiempos en los que estos puntos deben ser alcanzados, pero desconocemos cuantas variables son necesarias para definir el problema. Teniendo en cuenta que se desea realizar una trayectoria con  $n$  tramos, se deben definir  $n + 1$  puntos de paso como se aprecia en la figura 2.6.

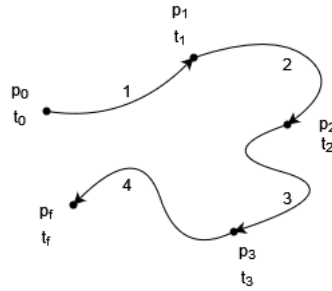


Figura 2.6: Ejemplo de calculo de número de puntos de paso

Dado que cada punto de paso esta compuesto por sus componentes  $x, y, z$  y en cada punto se debe definir también el tiempo en el que es alcanzado  $t$ , el número de variables queda definido por la expresión 2.18.

$$\#var = 4(n + 1) \quad (2.18)$$

De la expresión 2.18, se deben eliminar las condiciones iniciales  $x_0, y_0, z_0, t_0$  y las condiciones finales  $x_f, y_f, z_f$ , el tiempo final no debe definirse ya que es parte de la optimización. De este modo del numero de variables queda definido por la ecuación 2.19.

$$\#var = 4(n + 1) - 7 \quad (2.19)$$

Con el número de variables definido, se debe plantear la codificación de variables empleada para el problema mostrado en la tabla 2.2.

Tabla 2.2: Codificación de las variables de decisión del problema en 3D

$x_1$	$x_2$	...	$x_{n-1}$	$y_1$	$y_2$	...	$y_{n-1}$	$z_1$	$z_2$	...	$z_{n-1}$	$t_1$	$t_2$	...	$t_n$
-------	-------	-----	-----------	-------	-------	-----	-----------	-------	-------	-----	-----------	-------	-------	-----	-------

Con las funciones objetivo, las restricciones y las variables de decisión definidas, queda definido formalmente el problema de optimización a resolver.

### 2.3.2. Planificador de trayectorias 2D

Para resolver el problema con este enfoque es necesario en primer lugar modificar algunas de las funciones objetivo y restricciones.

- Se eliminan las restricciones referentes al movimiento vertical de la sonda. Esto se debe a que no se van a realizar medidas en profundidad en toda la trayectoria, solo en los puntos de paso donde se realizará un perfil de medida que recogerá toda la concentración en  $z$  del punto de paso situado en  $(x, y)$ .
- Se modifica la función objetivo empleada, eliminando el componente en  $z$ . Esto implica que la aportación realizada por cada partícula en un punto  $(x', y')$  sea la misma para todo  $z$ . De este modo, el optimizador tratará de buscar las zonas de máxima concentración sin importar en que  $z$  se encuentre ya que con el perfil de medida se alcanzará dicho punto.

$$\max OF_3 = \sum_{k=1}^{N_p(t)} \sup_{t_j \in [t_0, T_p, t_e]} \left[ A_c \exp \left( \sum_{d \in \{x, y\}} - \frac{(s_d(t_j) - p_d^k(t_j))^2}{2\sigma_{c,d}^2} \right) \right] \quad (2.20)$$

donde en este caso  $t_j$  se trata de un vector con los tiempos en los que se alcanza cada punto de paso. De esta forma no se tiene en cuenta la medida en toda la trayectoria, solo se tiene en cuenta en los puntos de paso.

Con las restricciones y las funciones objetivo definidas, es momento de determinar las variables de decisión. El razonamiento es similar al caso en tres dimensiones, se desea realizar una trayectoria de  $n$  tramos, de modo que se obtendrán  $n + 1$  puntos de paso. En este caso son tres las variables necesarias para ubicar este punto en el espacio-tiempo,  $x, y, t$  de modo que se obtendrá un total de  $\#var = 3(n + 1)$ . Como en el caso anterior, se deben eliminar las condiciones iniciales y finales ya que se debe definir  $x_0, y_0, t_0, x_f, y_f$  de modo que la expresión que define el número de variables de decisión en función del número de tramos es la reflejada en la ecuación 2.21.

$$\#var = 3(n + 1) - 5 \quad (2.21)$$

Con el número de variables definido, se debe plantear la codificación de variables empleada para el problema mostrado en la tabla 2.3.

Tabla 2.3: Codificación de las variables de decisión del problema en 2D

$x_1$	$x_2$	...	$x_{n-1}$	$y_1$	$y_2$	...	$y_{n-1}$	$t_1$	$t_2$	...	$t_n$
-------	-------	-----	-----------	-------	-------	-----	-----------	-------	-------	-----	-------

Con las restricciones, las funciones objetivo y las variables de decisión ya definidas, queda definido el problema de optimización a resolver.

### 2.3.3. Planificador con puntos de paso preseleccionados

El planteamiento inicial de las variables de decisión es el mismo que en el caso de dos dimensiones. Se desea realizar una trayectoria de  $n$  tramos, por lo que serán necesarios  $n + 1$  puntos de paso, de los cuales al quitar las condiciones iniciales y finales se obtienen  $\#var = 3(n + 1) - 5$ . Hasta aquí la codificación es idéntica al caso anterior, sin embargo, es necesario eliminar los  $m$  puntos de paso que se han pre-establecido, obteniendo  $\#var = 3(n + 1) - 5 - 2m$ . Por último, se van a añadir a las variables de decisión el índice dentro de la lista de puntos de paso en el que se debe ubicar cada punto de paso definido por el operador, es decir  $m$  variables, de modo que se obtiene la expresión de la figura 2.22.

$$\#var = 3(n + 1) - 5 - m \quad (2.22)$$

Con el número de variables definido, se debe plantear la codificación de variables empleada para el problema mostrado en la tabla 2.4.

Tabla 2.4: Codificación de las variables de decisión del problema en 2D con puntos de paso predefinidos

$x_1$	$x_2$	...	$x_{n-1-m}$	$y_1$	$y_2$	...	$y_{n-1-m}$	$t_1$	$t_2$	...	$t_n$	$i_1$	$i_2$	...	$i_m$
-------	-------	-----	-------------	-------	-------	-----	-------------	-------	-------	-----	-------	-------	-------	-----	-------

Siendo  $n$  el número de tramos de la trayectoria,  $m$  el número de puntos de paso predefinidos e  $i$  el índice donde intercalar cada uno de estos puntos.

### 2.3.4. Codificación con tiempos absolutos frente a tiempos relativos

En las secciones anteriores se ha descrito como se codifican las variables de decisión en cada uno de los planteamientos del problema. En todos los planteamientos, se usa como variables de decisión el tiempo que se tarda en alcanzar cada punto de paso. Estos valores permiten modificarse para considerar cada una de las variables referentes al tiempo como absolutas o como relativas. Una codificación con tiempos absolutos implica que la variable de decisión de tiempo en alcanzar el punto  $p$  es el tiempo con respecto al inicio  $t_0$ . Una codificación con tiempos relativos implica que la variable de decisión de tiempo en alcanzar ese mismo punto es el tiempo con respecto al tiempo en alcanzar el punto anterior, es decir, el tiempo que se tarda de ir de un punto al siguiente.

Estas codificaciones presentan un comportamiento distinto en los resultados, es por ello que en este trabajo se comparará la respuesta de estas dos codificaciones con el fin de comprobar cual ofrece una mejor respuesta en cada uno de los planteamientos propuestos para resolver el problema.

## 2.4. Algoritmo NSGA-II

El algoritmo NSGA-II (Non-dominated Sorting Genetic Algorithm) es un algoritmo genético de optimización diseñado para resolver problemas multi-objetivo desarrollado por Deb et al. [2002]. Es el método utilizado para optimizar las rutas de nuestro planificador, al ser un algoritmo que se ha utilizado con éxito en alguna de las variantes ya mencionadas del problema de optimización en el que se centra este TFM. Para explicar este método, es necesario revisar una serie de conceptos fundamentales, definición de dominancia, frente pareto, distancia de apilamiento y los conceptos de cruce y mutación de empleados en los algoritmos genéticos.

### 2.4.1. Concepto de Dominancia

Se dice que una solución  $s_1$  domina a una solución  $s_2$  si se cumplen las siguientes condiciones:

1. La solución  $s_1$  no es peor que  $s_2$  en todos los objetivos.
2. La solución  $s_1$  es estrictamente mejor que  $s_2$  en por lo menos un objetivo.

Si alguna de estas condiciones no se cumpliera, la solución  $s_1$  no dominaría a la solución  $s_2$ , Flórez et al. [2008].

Este concepto se puede extender a  $n$  soluciones para encontrar una serie de soluciones que no sean dominadas. En el algoritmo NSGA-II, la dominancia se emplea para clasificar las soluciones en frentes pareto.

### 2.4.2. Frente pareto

El siguiente concepto necesario para comprender el funcionamiento del algoritmo NSGA-II es el concepto de frente pareto. Si se consideran  $n$  soluciones, con cada una  $m$  funciones objetivo (funciones para optimizar), el frente pareto es el conjunto de soluciones no dominadas que se puede calcular del siguiente modo:

1. Tomar la solución  $x_i$  y comprobar dominancia con cada una de las soluciones  $x_j$  con  $i \neq j$ .
2. Si  $x_i$  es dominado, pasar a la siguiente solución y volver al paso 1.
3. Si no es dominado se marca esta solución como no dominada, pasar a la siguiente solución y volver al paso 1.

Con todas las soluciones evaluadas, se toma el conjunto de valores no dominado, este será el frente pareto.

Un aspecto importante de este algoritmo es la definición de distintos niveles de frente pareto. El frente pareto explicado pertenece al primer nivel, si estas soluciones no se toman en cuenta y se realiza de nuevo el proceso se obtiene un frente pareto de segundo nivel. Esto se puede repetir mientras que el número de soluciones sea no nulo.

### 2.4.3. Calculo Crowding-Distance

El método de cálculo de distancia de apilamiento (crowding distance) utilizado en el algoritmo NSGA-II es fundamental para mantener la diversidad en las soluciones de este algoritmo multiobjetivo. Mediante este cálculo se desea conocer la densidad de soluciones alrededor de una solución del frente pareto.

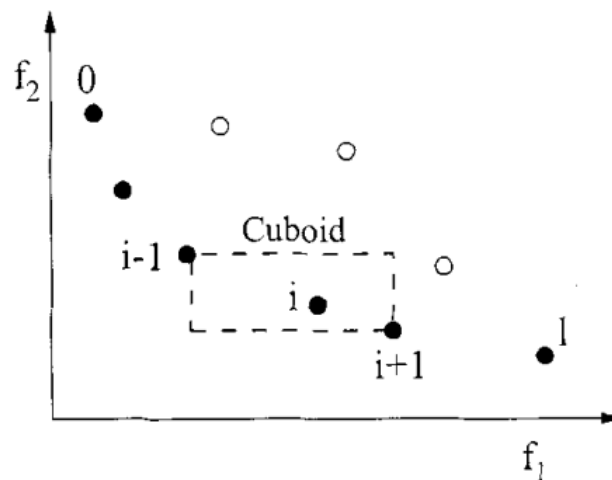


Figura 2.7: Calculo Crowding-Distance, Deb et al. [2002]

La idea simplificada de este método consiste en evaluar la distancia a la que se encuentra cada solución de la solución  $x_i$ , siguiendo la expresión 2.23.

$$d_i^m = \frac{f_{i+1}^m - f_{i-1}^m}{f_{max}^m - f_{min}^m} \quad (2.23)$$

Donde  $f_{i+1}^m, f_{i-1}^m$  son los valores del objetivo  $m$  para las soluciones adyacentes y  $f_{max}^m, f_{min}^m$  son los valores máximo y mínimo del objetivo  $m$  en el frente pareto.

### 2.4.4. Concepto de cruce y mutación

El cruce es un operador de los algoritmos genéticos que combina la información de dos soluciones (individuos) para crear nuevas soluciones. Este operador combina las características de los padres para producir descendientes que hereden sus características. El método

de cruce empleado en este trabajo es el método Simulated Binary Crossover (SBX). En el cual se establece un parámetro de distribución  $\eta_{cross}$ , que controla la distribución de los descendientes alrededor de los padres y el parámetro  $\rho_{cross}$ , el factor de cruce. Un valor alto de  $\eta_{cross}$  hace que los descendientes sean más similares a los padres, mientras que un valor bajo aumenta la variabilidad en los descendientes.

La mutación es otro operador de los algoritmos genéticos que trata de evitar el estancamiento de las soluciones en óptimos locales. Su objetivo es evitar que el algoritmo converja demasiado pronto y así mantiene la diversidad dentro de la población de soluciones. Esto permite explorar el espacio de soluciones de manera más eficaz. El método empleado para este trabajo es el Polynomial Mutation (PM), donde se definen el parámetro  $\rho_{mut}$  siendo este la probabilidad de mutación de una solución, y  $\eta_{mut}$  la distribución de los valores mutados. Un valor alto de  $\eta_{mut}$  significa que los valores mutados estarán cerca del valor original, mientras que un valor bajo aumenta la variabilidad en el valor mutado.

#### 2.4.5. Funcionamiento del algoritmo

En este algoritmo inicialmente se crea una población inicial de padres  $P_0$ . Esta población inicial puede escogerse o seleccionarse de forma aleatoria, esto afecta directamente a las soluciones encontradas. El siguiente paso consiste en evaluar la población generada según las funciones objetivo establecidas, la población se ordena en frentes pareto, siendo las mejores soluciones las pertenecientes al primer frente pareto. Mediante la selección por torneo, el cruce y la mutación crea la población de descendientes  $Q_0$ . Con todo el sistema inicializado, se sigue el siguiente proceso:

1. Se combinan las poblaciones de padres y descendientes, con ello se evalúan las funciones objetivo y se determinan los niveles de frente pareto.
2. Se realiza el ordenamiento en niveles de frente pareto
3. Se calcula la distancia de apilamiento (crowding-distance) dentro de cada frente pareto.
4. Empezando por el primer frente se comienzan a añadir las soluciones a la siguiente generación.
5. Cuando a la siguiente generación no se le puede añadir un frente completo, se añaden las soluciones según la distancia de apilamiento, con el fin de conservar diversidad en las soluciones.
6. Todas las soluciones que no pasan a la siguiente generación son eliminadas.
7. La nueva generación pasa a ser los nuevos padres, se vuelve al paso 1 y se repite el proceso.

Este proceso queda representado en la figura 2.8, extraída del artículo Deb et al. [2002] y se puede apreciar el funcionamiento de dicho proceso con el algoritmo 1.

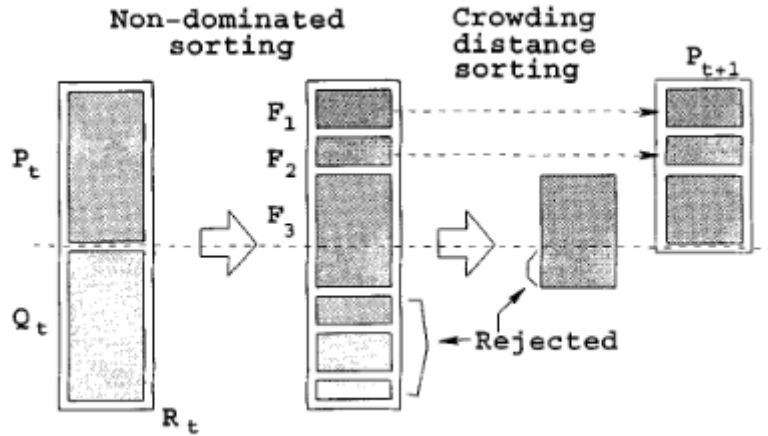


Figura 2.8: Proceso de combinación de poblaciones de NSGA-II

---

#### Algorithm 1 NSGA-II

---

```

 $R_t = P_t \cup Q_t$ 
 $\mathcal{F} = \text{fast-non-dominated-sort}(R_t)$ 
while  $|P_{t+1}| + |\mathcal{F}_i| \geq N$  do
    crowding-distance-assignment( $\mathcal{F}_i$ )
     $P_{t+1} = P_{t+1} \cup \mathcal{F}_i$ 
     $i = i + 1$ 
end while
Sort( $\mathcal{F}, \prec_n$ )
 $P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1 : (N - |P_{t+1}|)]$ 
 $Q_{t+1} = \text{make-new-pop}(P_{t+1})$ 

```

```

 $t = t + 1$ 

```

---

## 2.5. Análisis de resultados

Esta sección inicia con una explicación detallada del procedimiento que se emplea para llevar a cabo un análisis exhaustivo de la calidad de los resultados obtenidos. Se describe paso a paso cada uno de los métodos utilizados para evaluar las soluciones obtenida. Finalmente, la sección concluye con la presentación de un ejemplo concreto en el que se pone en práctica el procedimiento descrito, demostrando cómo se utiliza para obtener conclusiones acerca del funcionamiento del algoritmo.

### 2.5.1. Método de análisis de los resultados

Uno de los métodos empleados para analizar el comportamiento del algoritmo se basa en el análisis del hipervolumen de las soluciones Bader and Zitzler [2011]. Este método trata de evaluar la calidad de las soluciones en el primer frente pareto generado por el algoritmo. El hipervolumen es una medida que calcula el volumen del espacio objetivo dominado por las soluciones del frente pareto, este volumen estará limitado por unas referencias que sean dominadas por todo el frente pareto.

Una forma sencilla de comprender el funcionamiento de este método consiste en emplear un problema ficticio con dos funciones, al ser un problema con dos dimensiones, no se analiza el volumen sino el área. En primer lugar se deben establecer las referencias, con ellas se podrá calcular el área (hipervolumen) de cada solución. Supondremos entonces un problema con dos funciones objetivo  $f_1$  y  $f_2$ , de las cuales, se desea minimizar el primer objetivo y maximizar el segundo. Planteando las limitaciones se obtiene la figura 2.9, donde también se muestra el área ocupada por una única solución.

En este ejemplo, dado que se desea minimizar la función  $f_1$  y maximizar  $f_2$  el área parte de la esquina inferior derecha. En ambos casos se determina el área con respecto al peor de los casos de cada función objetivo,  $f_{1lim}$  para  $f_1$  y 0 para  $f_{2lim}$ . Con ello se puede calcular el área del hipervolumen ocupada por nuestra solución, de modo que cuantifica como de buena es esta solución.

Este problema en 3 dimensiones ya implica analizar un volumen en lugar de un área 2.11, pero cabe preguntarse lo siguiente ¿Como determinamos cuantitativamente este volumen? Una forma de determinar el área o el volumen es mediante integración por el método de montecarlo, por el cual se genera una serie de números aleatorios, se evalúa cuales de estos números quedan por debajo de la solución, dividiendo el número de elementos por debajo de la solución entre el número de elementos generados, se obtiene el área o volumen que ocupa la solución, tal y como se muestra en la figura 2.10 y en la figura 2.12.

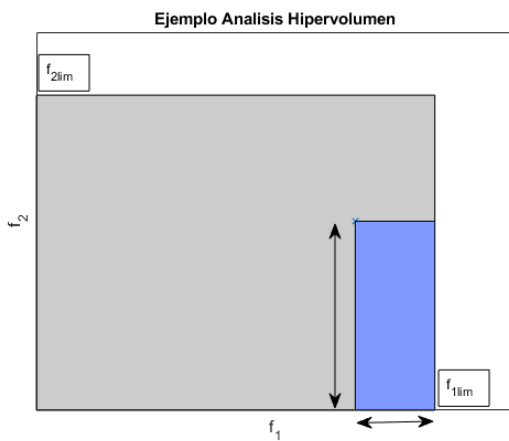


Figura 2.9: Análisis del hypervolumen 2D

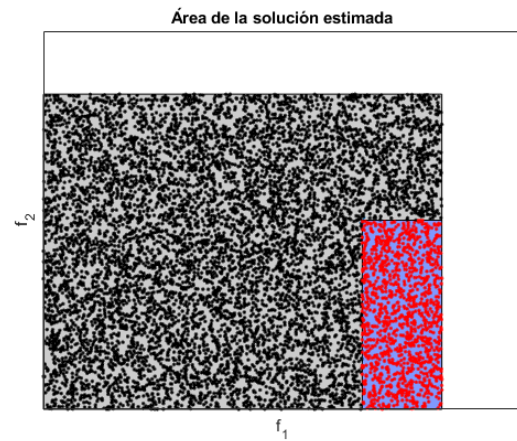


Figura 2.10: Ejemplo aplicado del método de montecarlo

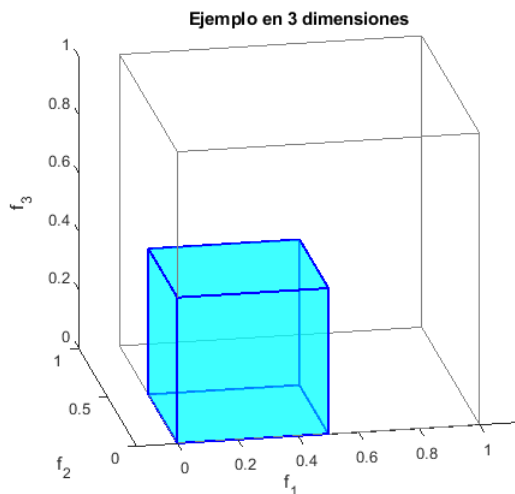


Figura 2.11: Ejemplo hipervolumen en tres dimensiones

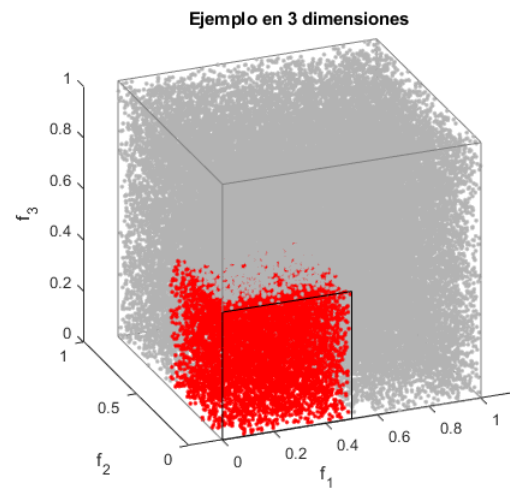


Figura 2.12: Ejemplo montecarlo en 3 dimensiones

## 2.5.2. Ejemplo de análisis de resultados

Para comprender como se realiza el análisis sobre el problema que se plantea en este trabajo, se va a presentar en esta sección un ejemplo de análisis de convergencia basado en hipervolumen y en funciones objetivo. Para realizar este análisis es necesario ejecutar el algoritmo de optimización repetidas veces almacenando los resultados del primer frente pareto, los valores de las funciones objetivo asociadas a dicho frente pareto, y el número de soluciones que cumplen las restricciones planteadas en cada generación. Estos datos serán los necesarios para realizar el análisis de las soluciones obtenidas. Para este caso, en particular, se emplearán 30 ejecuciones del algoritmo. Para el calculo del valor del HVE y de las funciones objetivo, se calculará la media y la desviación estándar de las 30 ejecuciones para todas las iteraciones del algoritmo. Esta información se dibuja a lo largo de todas las generaciones

del algoritmo dando como resultado la figura 2.13. En esta figura, en las cuatro gráficas superiores, se puede apreciar como la media de las 30 ejecuciones del hipervolumen y de las funciones objetivos tiende a un valor estacionario, lo que es una buena señal ya que quiere decir que el algoritmo ha encontrado una solución que no mejora drásticamente con la siguiente generación, lo que indica que el algoritmo converge. Por otro lado, en la imagen inferior de la figura se representa el porcentaje de ejecuciones con soluciones que cumplen las restricciones en cada generación, pudiéndose observar a medida que pasan las generaciones, hay más soluciones capaces de cumplirlas. Si el color rojo se propagase por las generaciones, indicaría que el algoritmo no logra resolver el problema sin saltarse alguna de las restricciones.

Haciendo uso de estas herramientas se caracterizará el funcionamiento de las variantes de planificador planteadas con distintas configuraciones.

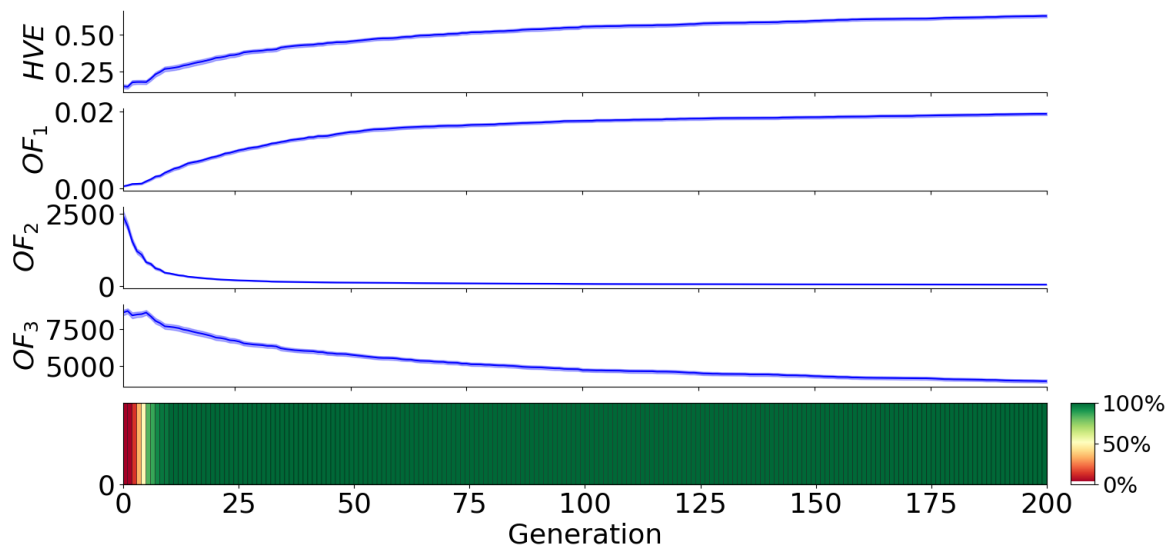


Figura 2.13: Ejemplo de análisis de resultados



# Capítulo 3

## Resultados

En este capítulo se pondrán a prueba todos los algoritmos y planteamientos propuestos para resolver el problema planteado. Con el fin de determinar cual es la configuración más adecuada para su resolución, se propondrán distintas modificaciones tanto en los parámetros clave de los algoritmos genéticos (como son el cruce y la mutación), como en la definición de las variables de decisión. Una vez puesto a prueba cada una de las variantes mediante la herramienta mostrada en la sección 2.2 del capítulo 2, se determinará cual de las configuraciones es más adecuada para el problema, y la variante más adecuada en cada caso es utilizada como punto de partida para analizar las variaciones posteriores.

### 3.1. Modificación de los parámetros del AG

En esta sección, se analizarán las soluciones obtenidas empleando el algoritmo genético NSGA-II mediante el análisis del hipervolumen y analizando también algunas de las soluciones del frente pareto obtenido. Como modificaciones a analizar se realizarán optimizaciones alterando los valores  $\rho_{mut}$  y  $\rho_{cross}$  se tratan de las probabilidades de mutación y cruce respectivamente, de cada individuo. Se tomarán las siguientes combinaciones mostradas en la tabla 3.1

Tabla 3.1: Configuraciones para el algoritmo NSGA-II

NSGA-II											
Configuración 1				Configuración 2				Configuración 3			
$\rho_{mut}$	0.9	$\rho_{cross}$	0.9	$\rho_{mut}$	$\frac{1}{\#var}$	$\rho_{cross}$	0.8	$\rho_{mut}$	$\frac{1}{\#var}$	$\rho_{cross}$	0.7
$\eta_{mut}$	0.15	$\eta_{cross}$	0.15	$\eta_{mut}$	0.15	$\eta_{cross}$	0.15	$\eta_{mut}$	0.15	$\eta_{cross}$	0.15

Estas modificaciones se emplearán en los distintos planteamientos del problema, donde la nomenclatura es la siguiente:

- Problema M: Problema en tres dimensiones.

- Problema N: Problema en dos dimensiones.
- Problema L: Problema en dos dimensiones con puntos de paso predefinidos.

Para resolver estos problemas se deben definir los parámetros y el valor de las restricciones del sistema, estos parámetros quedan reflejados en la tabla 3.2.

Tabla 3.2: Parámetros del sistema

<b>Definiciones del problema</b>			
$P_{init}$	[380, 100] m	$a_{Hmax}$	0,1 $m/s^2$
$P_{fin}$	[380, 100] m	$v_{Vmax}$	1 $m/s$
$N_{tramos}$	5	$a_{Vmax}$	0,1 $m/s^2$
$v_{Hmax}$	1 $m/s$	$\kappa$	0,2 $m^{-1}$
	$t_{max}$		3 h

Todos estos valores se han escogido tomando como referencia experimentos con un vehículo autónomo de superficie acuático real, diseñado y construido por el grupo de investigación de Ingeniería de Sistemas, Control, Automática y Robótica de la Universidad Complutense de Madrid.

### 3.1.1. Problema en tres dimensiones

En primer lugar se analizará la respuesta obtenida por el planificador de tres dimensiones, realizando una comparación entre las distintas configuraciones mostradas en la tabla 3.1. La optimización de la trayectoria se realizará sobre el escenario A, ya que es el escenario que presenta más zonas de alta concentración. Con esto se podrá comprobar si el algoritmo es capaz de explorar distintas zonas. Como se ha explicado en la sección 2.5.1, el método de análisis empleado es el análisis del hipervolumen formado por las soluciones, también, para comprobar como de buenas son las soluciones obtenidas, se analizarán los valores optimizados con el fin de comprobar si estos valores convergen y ver a que valor tienden. Con este análisis se pretende escoger cual de las configuraciones del problema es la más adecuada en este contexto. El resultado obtenido se muestra en la figura 3.1, donde se puede apreciar que las soluciones arrojadas por las tres configuraciones logran converger en las 200 generaciones empleadas, dominando la configuración 1 sobre las demás. El hecho de que se obtengan mejores resultados con la primera configuración se puede apreciar también en las gráficas donde se muestran los valores objetivo en cada generación, la configuración 1 es la que tarda más en llegar a un valor estacionario pero es el que ofrece una respuesta mejor. Por último es necesario observar el numero de generaciones necesarias para lograr que el 100 % de las ejecuciones con soluciones cumplan las restricciones en una generación dada. Como se puede apreciar, en el caso de la configuración 1 es la primera en lograrlo, esto ocurre por las probabilidades de cruce y mutación que son mayores a las usadas en el resto de

configuraciones. Un valor más elevado de estas probabilidades hace que las soluciones del algoritmo exploren con mayor facilidad el espacio de variables de decisión.

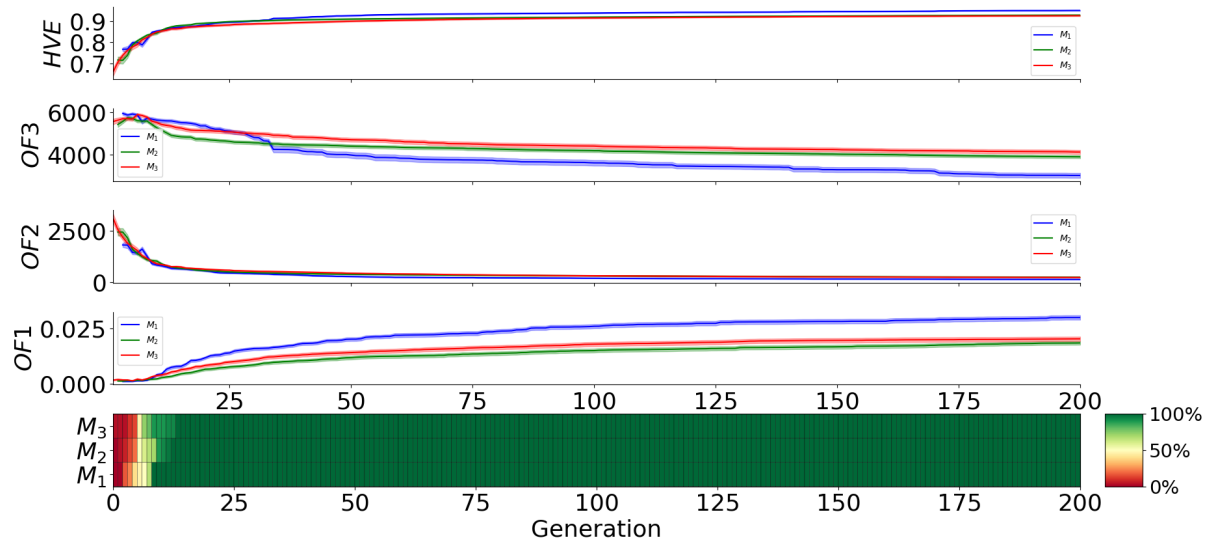


Figura 3.1: Análisis de convergencia del problema en tres dimensiones ante cambios en el valor de los parámetros del optimizador

Una vez se han analizado las soluciones obtenidas por las distintas configuraciones del algoritmo, se mostrarán algunos de los resultados obtenidos para cada solución. Dos ejemplos ilustrativos de cada configuración se muestran en las figuras 3.2, 3.3 y 3.4, donde se puede apreciar que las soluciones de la primera configuración aunque se quedan aisladas en una región, es la que más zonas de concentración explora haciendo buenas trayectorias, sin excederse en longitud. La segunda configuración alcanza también varias zonas de altas concentraciones, pero las trayectorias trazan curvas algo innecesarias. Analizando las trayectorias generadas por la tercera configuración, se puede apreciar que dos de los puntos de paso de la trayectoria parecen quedarse fijos. Esto puede deberse a que el algoritmo no es capaz de explorar adecuadamente el espacio de soluciones.

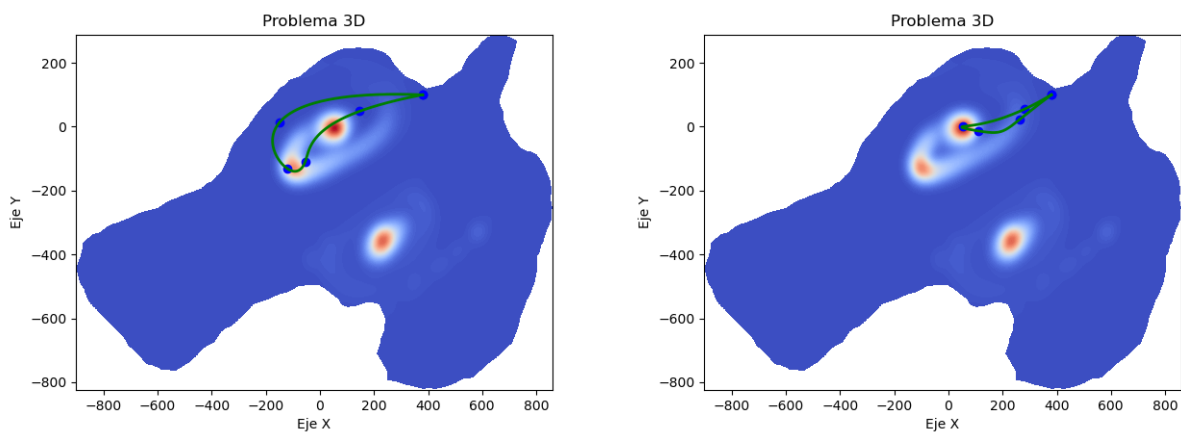


Figura 3.2: Trayectorias del problema en tres dimensiones obtenidas con la configuración 1

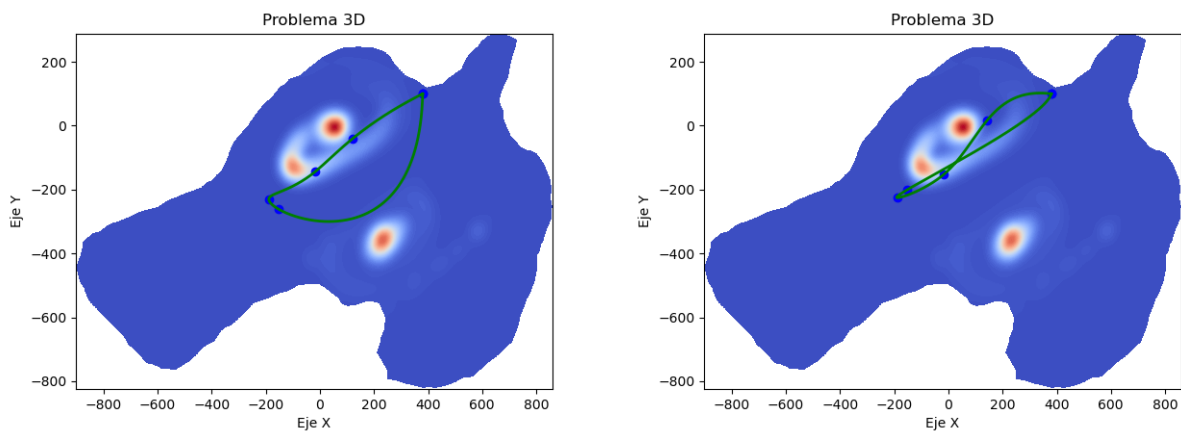


Figura 3.3: Trayectorias del problema en tres dimensiones obtenidas con la configuración 2

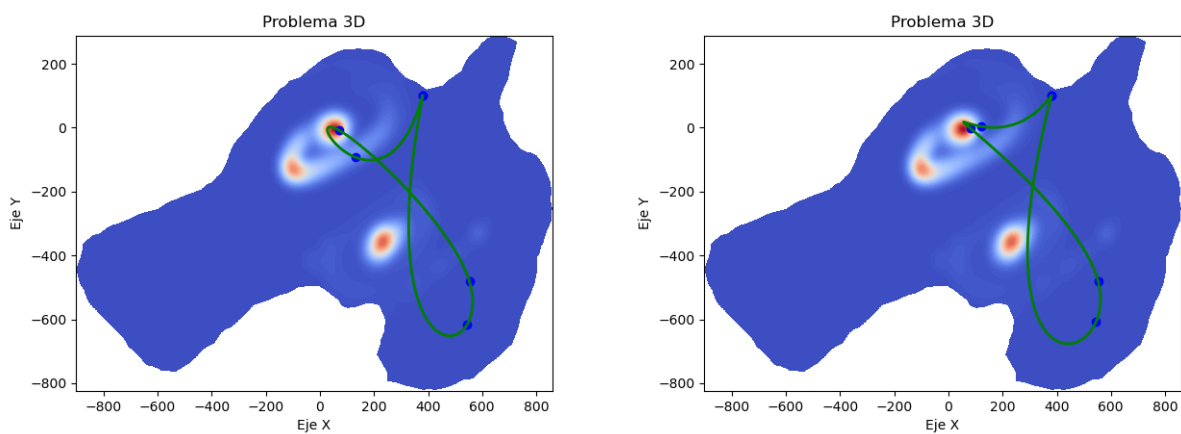


Figura 3.4: Trayectorias del problema en tres dimensiones obtenidas con la configuración 3

### 3.1.2. Problema en dos dimensiones

En esta sección se analizará la respuesta obtenida con el planteamiento presentado del planificador de dos dimensiones. De nuevo, como en el caso anterior se debe analizar el hipervolumen formado por las soluciones obtenidas, las funciones objetivo y el porcentaje de ejecuciones con soluciones que cumplen las restricciones por generación. Este análisis se encuentra en la figura 3.5, donde se puede apreciar que en este caso, de nuevo, la mejor respuesta se obtiene con la configuración 1, que tiene un porcentaje de hipervolumen superior al resto. Sin embargo, el valor al que tiende el hipervolumen y la primera función objetivo es menor que en el problema en tres dimensiones, ya que a diferencia del caso anterior, el planteamiento en dos dimensiones solo mide, realizando un perfil en profundidad, en los puntos de paso mientras que el planteamiento en tres dimensiones mide a lo largo de toda la trayectoria. Por otro lado, el 100% de soluciones que cumplen las restricciones para la configuración 1 se obtiene con menos generaciones que el resto. Esto demuestra que tal y como ocurre con el problema en tres dimensiones, la configuración 1 es la mejor en estos casos.

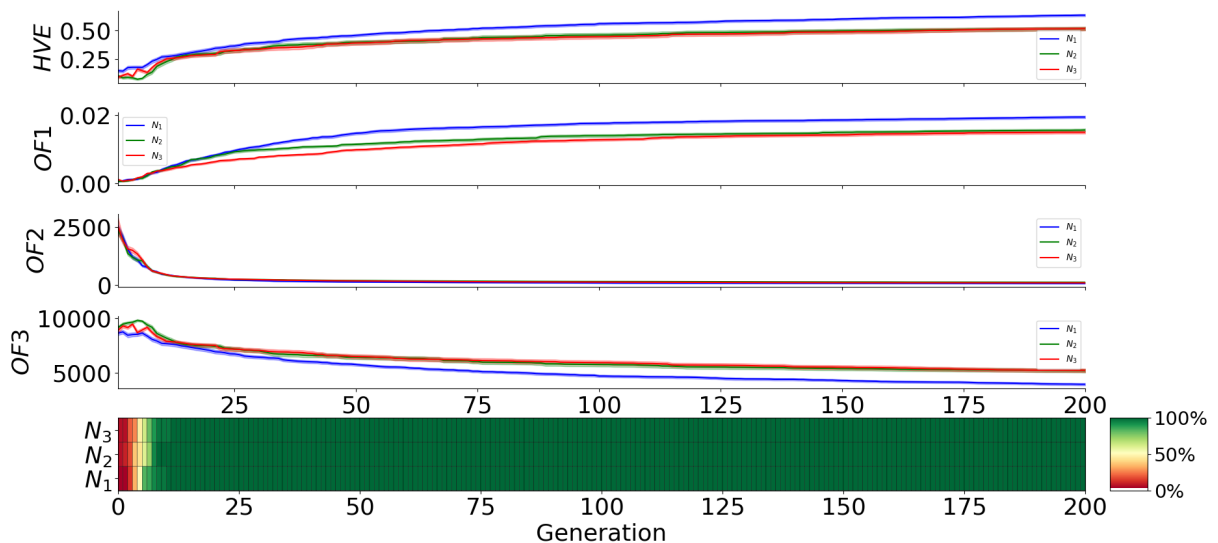


Figura 3.5: Análisis de convergencia del problema en dos dimensiones ante cambios en el valor de los parámetros del optimizador

Una vez analizado el funcionamiento del algoritmo de forma estadística, se mostrarán algunas de las soluciones generadas por el algoritmo. Ejemplos representativos de estas soluciones se muestran en las figuras 3.6, 3.7 y 3.8. En estas figuras se puede apreciar que el planteamiento presentado resuelve adecuadamente el problema ya que la gran mayoría de puntos de paso se encuentran en zonas de máxima concentración, donde se debería realizar un perfil de medida. Comparando las trayectorias y las ubicaciones de los puntos de medida, es razonable pensar que las mejores soluciones se obtienen con la primera configuración ya

que las trayectorias cubren una mayor extensión del embalse o simplemente van a uno de los puntos de medida y regresan. En este caso, la configuración 3 parece dar una mejor respuesta que la configuración 2, ya que es capaz de medir en las dos zonas de alta concentración ligeramente separadas de la parte superior del embalse.

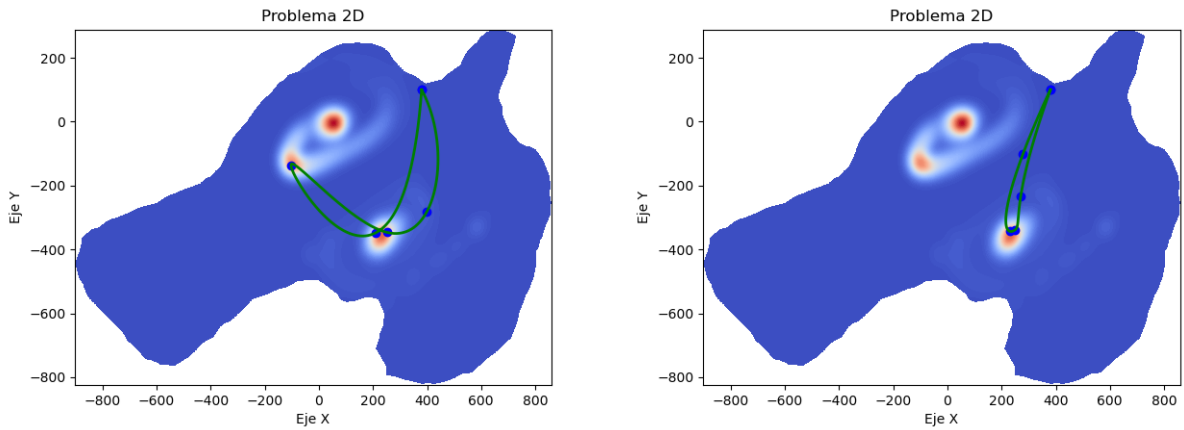


Figura 3.6: Trayectorias del problema en dos dimensiones obtenidas con la configuración 1

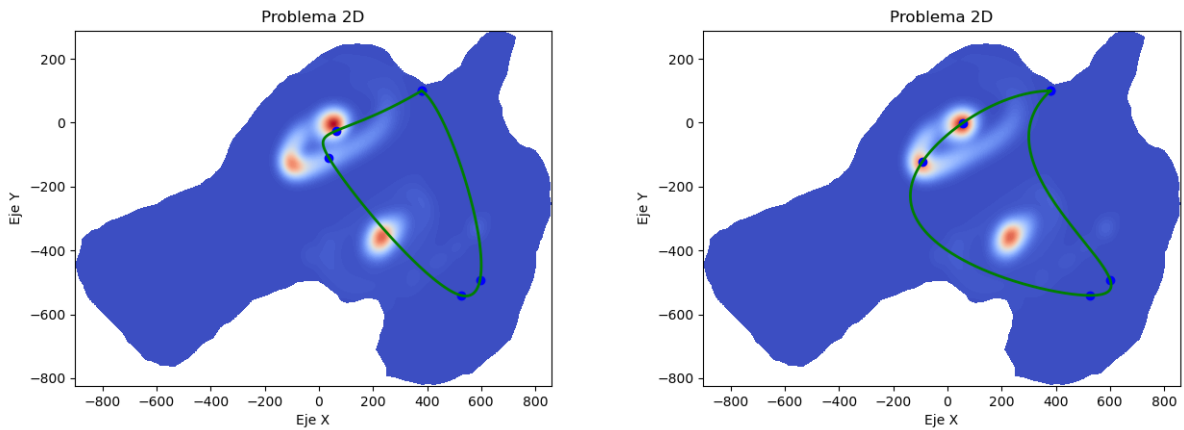


Figura 3.7: Trayectorias del problema en dos dimensiones obtenidas con la configuración 2

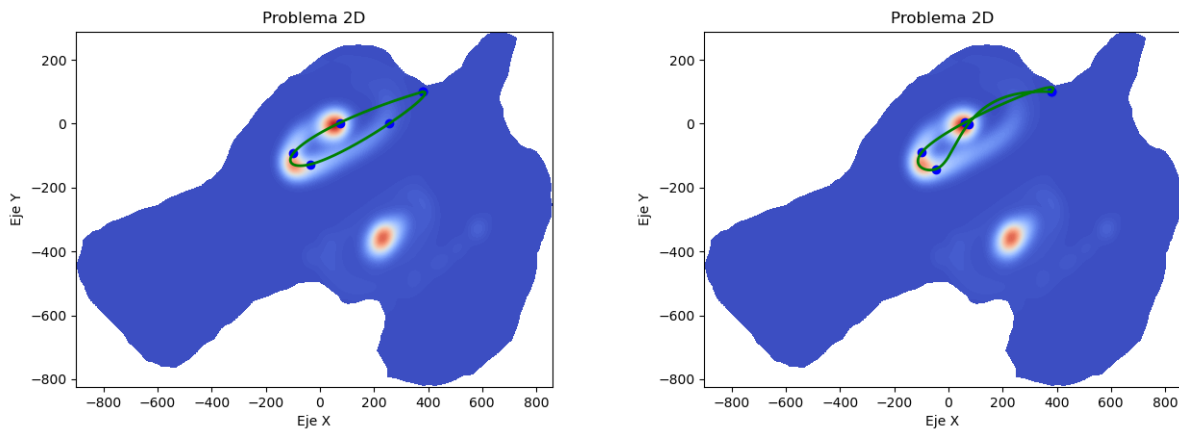


Figura 3.8: Trayectorias del problema en dos dimensiones obtenidas con la configuración 3

### 3.1.3. Problema con puntos de paso predefinidos

Este caso es comparte las funciones objetivo del planteamiento en dos dimensiones, aunque difiere del mismo al permitir que el usuario sitúe unos puntos de paso obligatorios. Por lo tanto, cabe esperar que los valores del hipervolumen y de las funciones objetivo queden limitados ya que se fuerza la ubicación de los puntos de paso que son previamente definidos. Al solo medir en los puntos de paso, para cualquier solución habrá puntos que se encontrarán en la misma posición pero que pueden ser alcanzados en distintos instantes de tiempo, para mejorar el resultado, el optimizador debe ser capaz de colocarlos en el mejor momento. El análisis se encuentra representado en la figura 3.9, donde se puede apreciar que el algoritmo ha conseguido obtener en el hipervolumen de la configuración 1 un valor similar al obtenido para el caso de dos dimensiones, siendo este superior a los hipervolumenes de las configuraciones 2 y 3. También se puede comprobar que la configuración 1 ofrece mejores resultados analizando las funciones objetivo dadas ya que en cada uno de los casos mejora al resto de las configuraciones. Por último, analizando el porcentaje de ejecuciones con soluciones factibles en cada generación, el 100 % de soluciones que cumplen todas las restricciones se obtiene antes para la configuración 3, pero no son muchas más que las necesarias par el resto de casos, denotando de nuevo que la configuración adecuada para este problema es la primera.

Ejemplos representativos de las trayectorias obtenidas en cada configuración se muestran en las figuras 3.10, 3.11 y 3.12. En primer lugar se debe destacar el hecho de que todas las trayectorias tienen un punto en común, el definido antes de la optimización. Se supondrá que se conoce por un registro histórico que en esa zona existe una alta concentración de cianobacterias por lo que la embarcación autónoma debe realizar las medidas en dicha posición. Como se puede apreciar, el hecho de que el hipervolumen sea similar al caso de dos dimensiones para la primera configuración, se debe a que las medidas se acumulan en la misma zona de alta concentración, mientras que en el resto de configuraciones, se explora el resto del embalse tomando medidas de concentración en la parte superior del embalse. En

este caso el hecho de no optimizar las soluciones tanto como la configuración 1 hace que la trayectoria pase por más zonas antes de regresar al punto de partida.

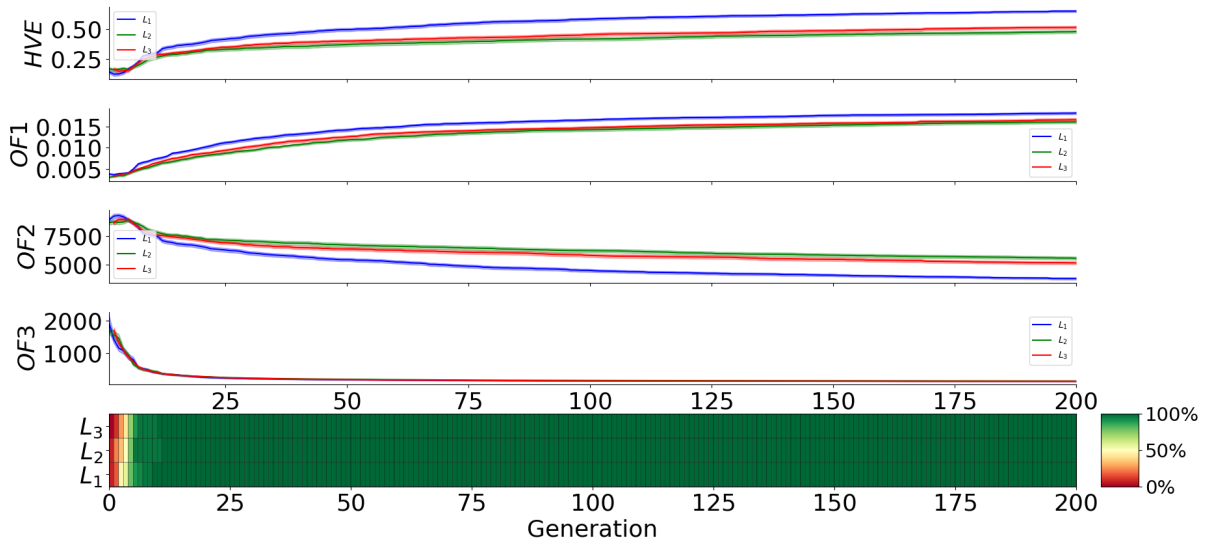


Figura 3.9: Análisis de convergencia del problema en dos dimensiones con puntos de paso preseleccionados, ante cambios en el valor de los parámetros del optimizador

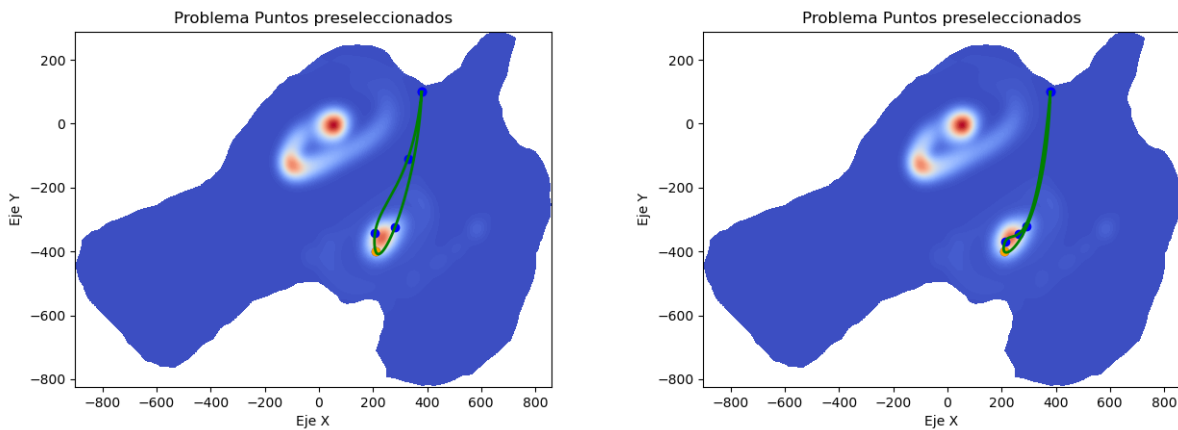


Figura 3.10: Trayectorias del problema en dos dimensiones con puntos de paso predefinidos obtenidas con la configuración 1

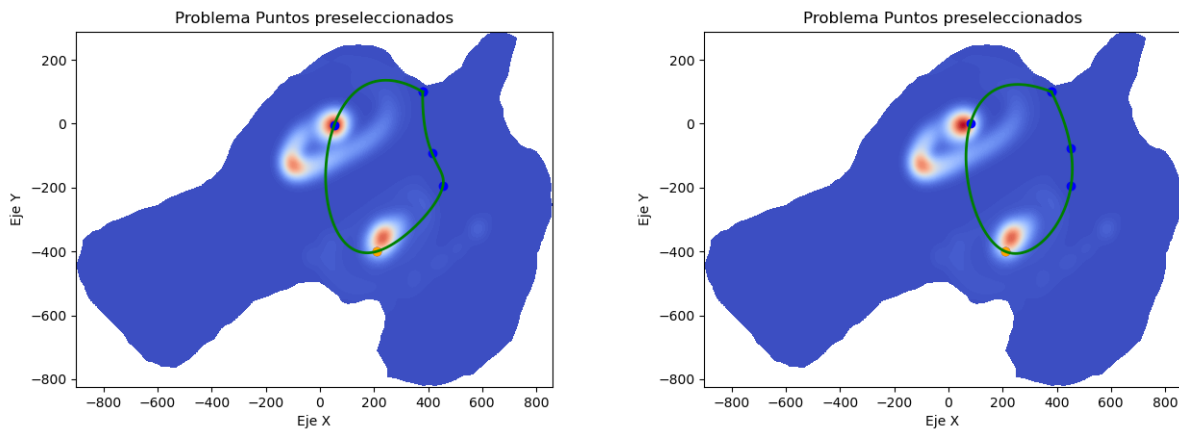


Figura 3.11: Trayectorias del problema en dos con puntos de paso predefinidos dimensiones obtenidas con la configuración 2

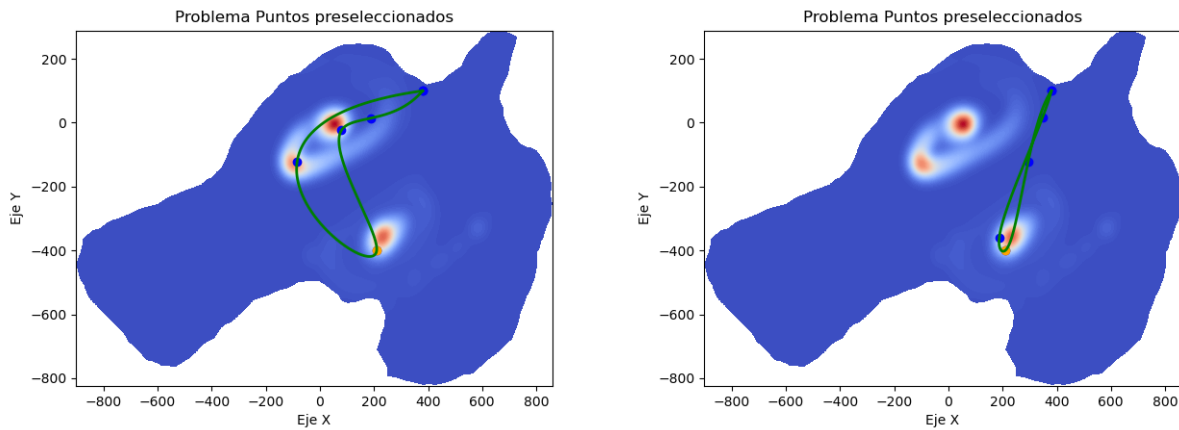


Figura 3.12: Trayectorias del problema en dos dimensiones con puntos de paso predefinidos obtenidas con la configuración 3

## 3.2. Modificación con tiempos relativos

Una vez se ha analizado cual es la configuración más adecuada del algoritmo para cada uno de los planteamientos del problema, se tomarán estas configuraciones para el algoritmo y se le aplicarán más cambios. En esta sección se analizará la respuesta de los algoritmos empleando una codificación con tiempos relativos, explicados en la sección 2.3.4. Una vez analizadas las respuestas se seleccionará cual de las codificaciones es más adecuada para este problema y se empleará para la codificación de las trayectorias.

### 3.2.1. Problema en tres dimensiones

En este apartado se compararán los hipervolumenes obtenidos por las soluciones, empleando una codificación con tiempos absolutos, cuyos resultados se obtienen del análisis de la sección 3.1, con la configuración 1, frente a los resultados obtenidos empleando una codificación con tiempos relativos. El resultado de este análisis se encuentra en la figura 3.13, donde se puede apreciar como la codificación con tiempos absolutos logra un hipervolumen superior a la configuración con tiempos relativos. A su vez, los tiempos absolutos permiten medir una concentración de cianobacterias mayor a la obtenida con tiempos relativos. Todo esto a costa de necesitar más generaciones para lograr un 100 % de ejecuciones con soluciones factibles. Un hecho destacable son las fluctuaciones que sufren tanto las funciones objetivo como el hipervolumen de la codificación con tiempos relativos, estas se deben a que las variaciones en el tiempo necesario para un único tramo condicionan todos los tiempos de todos los tramos posteriores haciendo que la trayectoria ante un cambio leve sea muy distinta a la original.

Una vez analizados los comportamientos de los hipervolumenes y de las funciones objetivo, en la figura 3.14 se presentan las trayectorias creadas con la codificación con tiempos relativos. Como se puede apreciar, esta codificación también resuelve el problema, sin embargo, se selecciona como codificación los tiempos absolutos debido a que los resultados son ligeramente mejores.

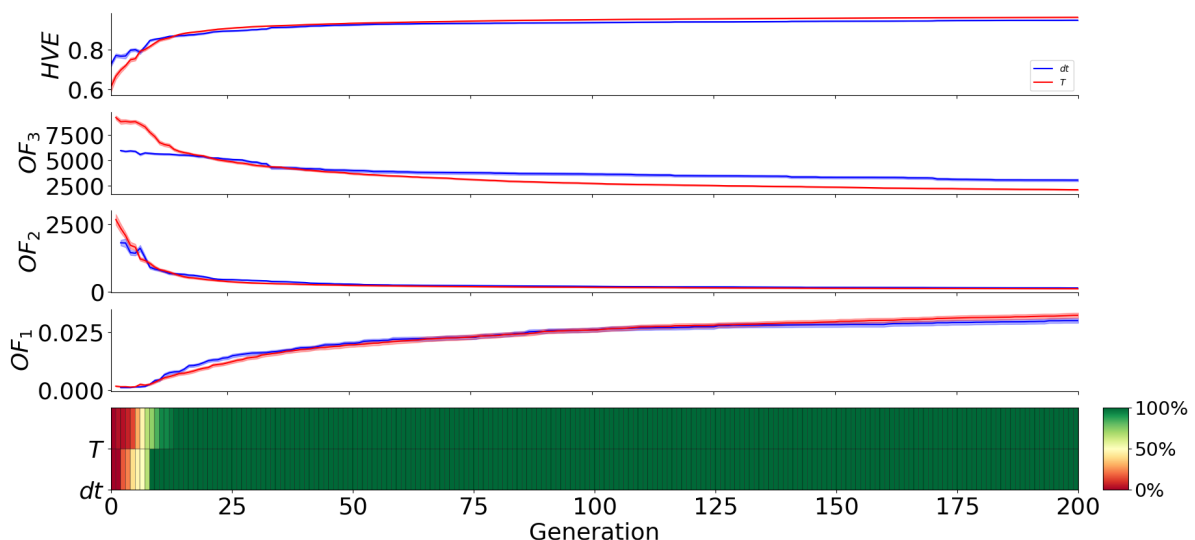


Figura 3.13: Análisis de convergencia del problema en tres dimensiones comparando la respuesta con tiempos absolutos frente a tiempos relativos

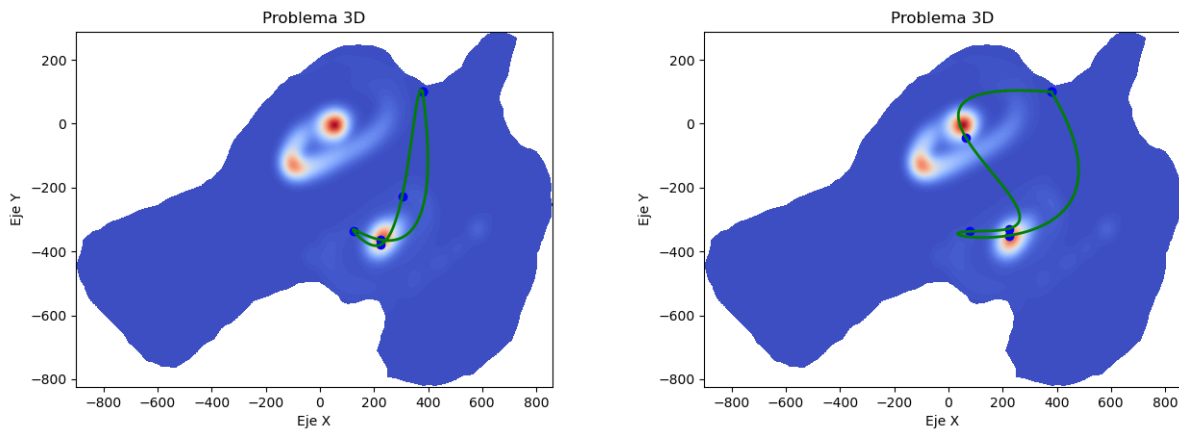


Figura 3.14: Trayectorias del problema 3D con una codificación de tiempo basada en tiempos relativos

### 3.2.2. Problema en dos dimensiones

La codificación de tiempos basada en tiempos relativos también es una opción para el planteamiento del problema en dos dimensiones. En la figura 3.15 se presentan los hipervolumenes ocupados por las soluciones presentadas con las dos codificaciones de tiempos. En este caso, como en el anterior, se obtiene un mayor hipervolumen con la codificación de tiempos absolutos, pero no por mejorar la función objetivo referente a la concentración de cianoobacterias, sino, por mejorar la función objetivo de tiempo de misión, donde es notoria la mejora que presenta emplear una codificación de tiempos absolutos. Por otro lado, de nuevo, son necesarias más generaciones para obtener un 100 % de ejecuciones con soluciones factibles frente a la codificación basada en tiempos relativos.

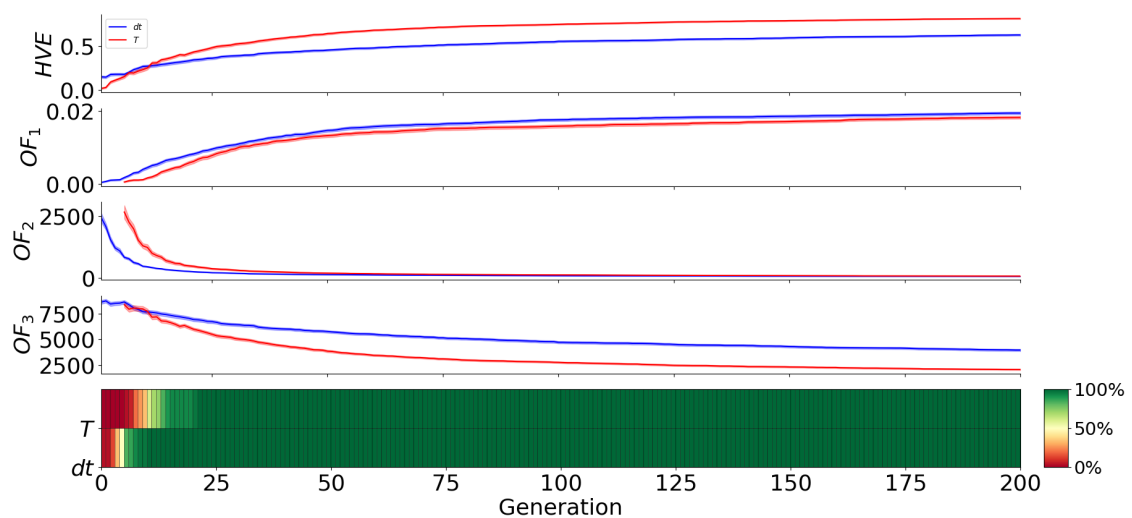


Figura 3.15: Análisis de convergencia del problema en dos dimensiones comparando la respuesta con tiempos absolutos frente a tiempos relativos

Las trayectorias obtenidas para esta codificación se presentan en la figura 3.16, donde se puede apreciar que los puntos de paso se encuentran muy próximos los unos a los otros. Esto da sentido a la mejora en la medida de la de concentración de cianobacterias, ya que los perfiles de medición se concentran durante más tiempo sobre una zona de alta concentración.

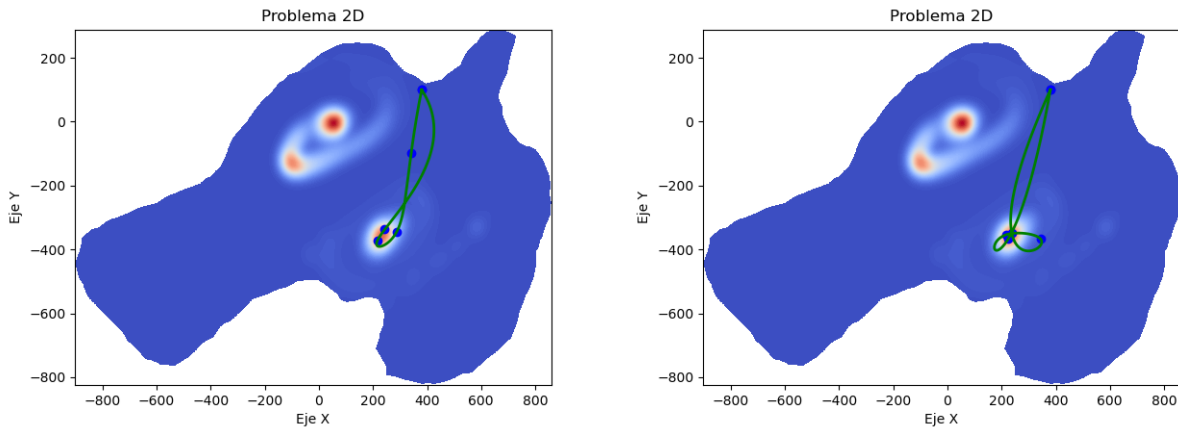


Figura 3.16: Trayectorias del problema 2D con una codificación de tiempo basada en tiempos relativos

### 3.2.3. Problema en dos dimensiones con puntos de paso predefinidos

En la figura 3.17 se presenta el análisis del hipervolumen y funciones objetivo de la codificación de tiempos basada en tiempos absolutos frente a relativos. En esta figura se puede apreciar como el hipervolumen de la codificación con tiempos absolutos es superior a la codificación con tiempos relativos, a pesar de empeorar ligeramente el objetivo referente a la concentración de cianobacterias. Esto ocurre ya que el tiempo de misión de las soluciones obtenidas por el planificador con tiempos absolutos es notablemente inferior al de las soluciones que se obtienen con la codificación de tiempos relativos.

A continuación se presenta en la figura 3.18, algunas de las soluciones obtenidas con la codificación con tiempos relativos, donde se puede apreciar que con esta nueva codificación, las trayectorias exploran dos de las zonas de alta concentración realizando una trayectoria más larga y por tanto empleando más tiempo que en las soluciones obtenidas con la codificación de tiempos absolutos.

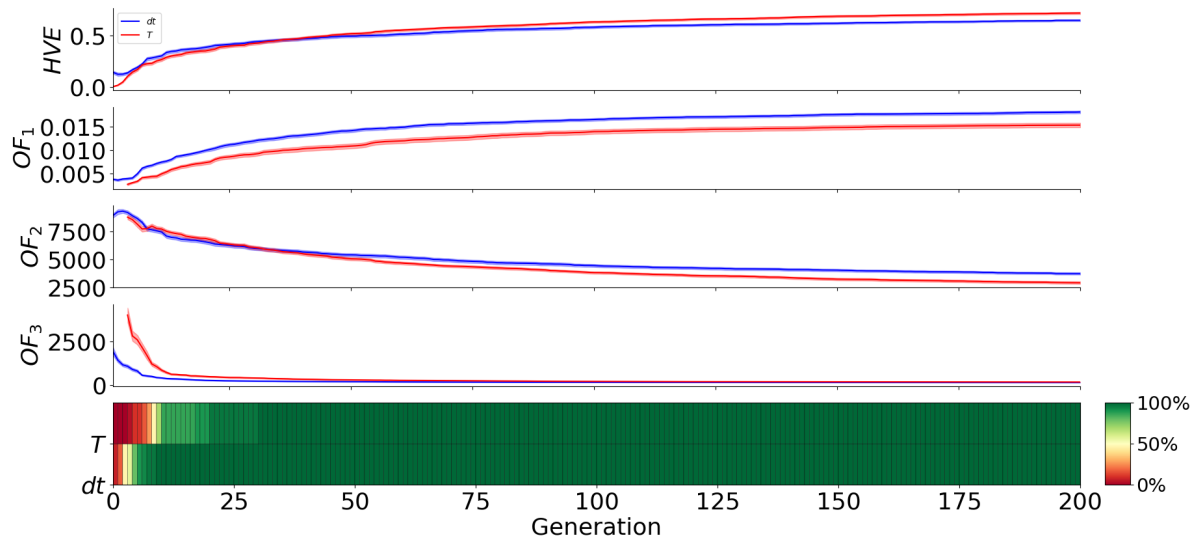


Figura 3.17: Análisis de convergencia del problema en dos dimensiones con puntos de paso preseleccionados, comparando la respuesta usando tiempos absolutos frente a tiempos relativos

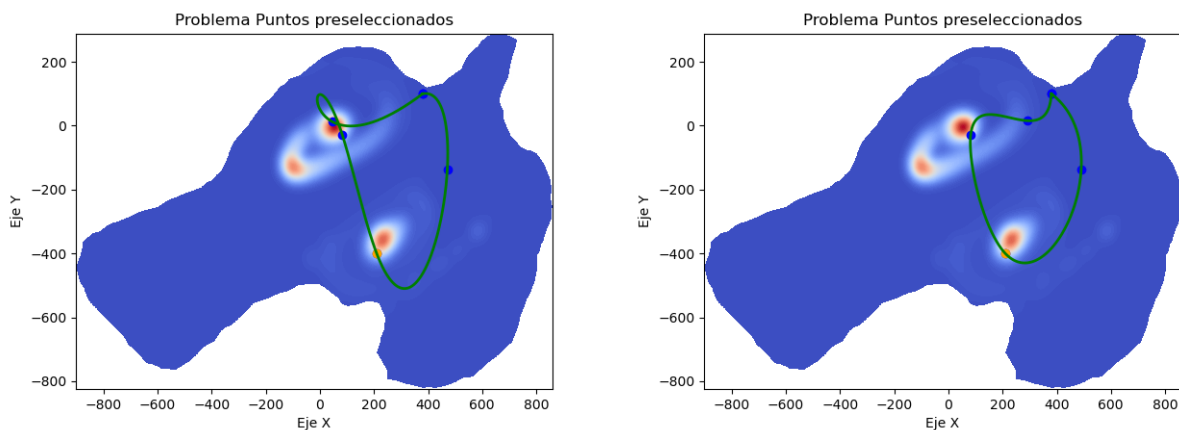


Figura 3.18: Trayectorias del problema 2D con una codificación de tiempo basada en tiempos relativos

### 3.3. Modificación del número de tramos

Según la codificación de la trayectoria presentada en la sección 2.3, donde se explica que la trayectoria se compone de un número de tramos, se desea modificar uno de los parámetros del algoritmo con el fin de que la trayectoria generada tenga más tramos, lo que implica que la curva sea más compleja. En las secciones anteriores se han empleado 5 tramos para cada trayectoria, en esta sección se emplearán 7. Al añadir más tramos, se añaden un número adicional de variables de decisión. En esta sección se comprobará como afecta este cambio al resultado en cada uno de los planteamientos del problema.

### 3.3.1. Problema en tres dimensiones

En esta sección se comparará y analizará los hipervolumenes de las soluciones del planificador en tres dimensiones con trayectorias de 5 y 7 tramos. En la figura 3.19 se puede apreciar que el hipervolumen ocupado por las soluciones para 7 tramos es inferior. Esto se debe a que los valores de las funciones objetivo de tiempo y longitud de trayecto son inferiores para 5 tramos. Por otro lado, en el caso de 7 tramos se obtiene un valor superior en la función objetivo 1, pero se obtiene un valor estacionario muy rápido por lo que puede indicar que se quede estancado en un óptimo local no deseado. En la gráfica donde se muestra el porcentaje de soluciones factibles en cada generación, se puede apreciar que son necesarias muchas generaciones para lograr un 100% de ejecuciones con soluciones que cumplan todas las restricciones.

Para completar el análisis se muestran en la figura 3.20 las trayectorias generadas por el planificador para 7 tramos, donde se puede apreciar que si bien la trayectoria pasa por una de las zonas de alta concentración, la trayectoria del vehículo de superficie no es adecuada ya que se extiende innecesariamente.

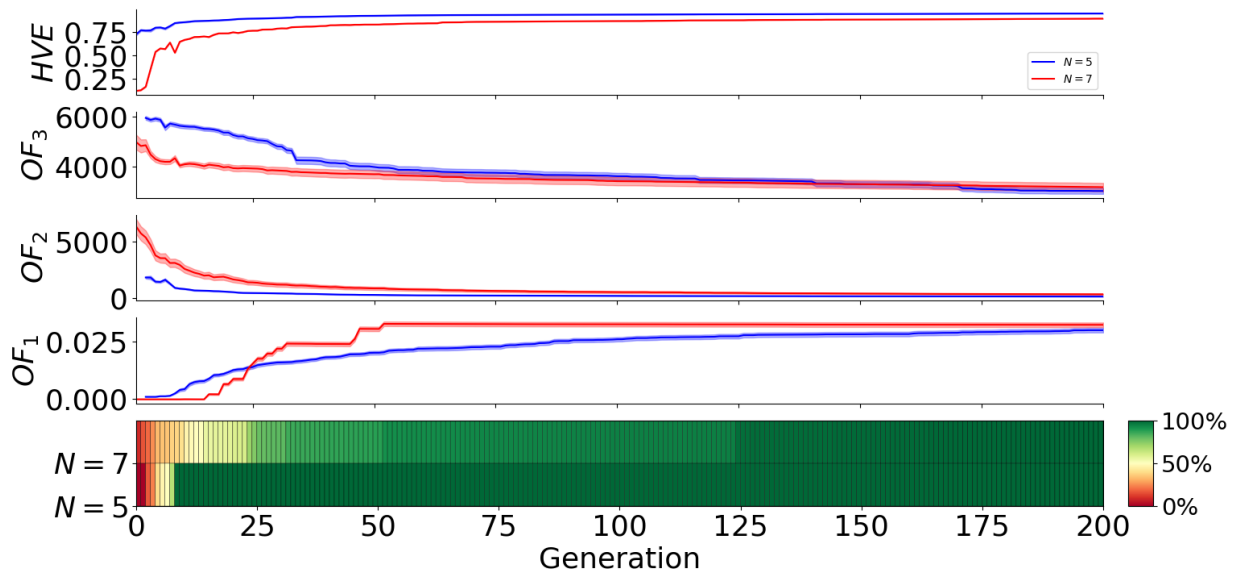


Figura 3.19: Análisis de convergencia del problema en tres dimensiones añadiendo tramos a la trayectoria

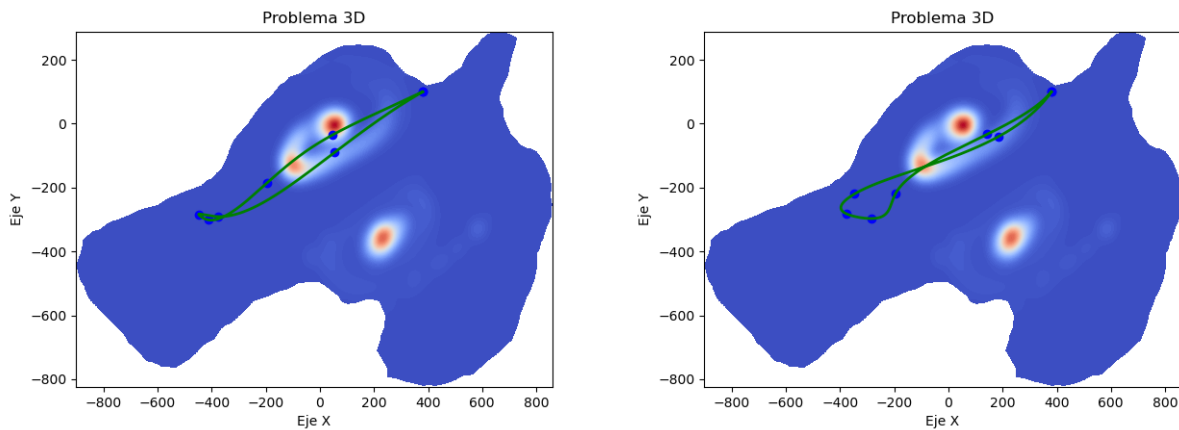


Figura 3.20: Trayectorias en dos dimensiones con 7 tramos

### 3.3.2. Problema en dos dimensiones

En esta sección se comparará y analizará los hipervolumenes de las soluciones obtenidas del planificador en dos dimensiones con trayectorias de 5 y 7 tramos. Este análisis se resume en la figura 3.21, donde se puede apreciar que el porcentaje del hipervolumen ocupado por las soluciones del planificador con trayectorias de 5 tramos es mayor a la de 7 tramos. Esto se debe a los valores de las funciones objetivo de distancia y tiempo, ya que el algoritmo con 5 tramos le es más sencillo realizar trayectos más cortos mientras que con 7 tramos, algoritmo debe optimizar más para lograr reducir estas distancias y tiempos. Cabe destacar dentro de la función objetivo 1, referente a la medida de las cianobacterias, que la trayectoria con 7 tramos es capaz de medir más concentración, pero esto solo es debido a que al tener más tramos tiene más puntos de paso y por tanto, mas puntos de medida. Por último, cabe destacar el elevado numero de generaciones necesarias para lograr un 100% de ejecuciones con soluciones factibles, frente al numero de generaciones necesarias para el caso de 5 tramos. Esto se debe a que al aumentar el numero de tramos aumenta considerablemente el número de variables de decisión haciendo que sea más complejo para el algoritmo encontrar soluciones que cumplan todas las restricciones.

Una vez analizadas las soluciones, se muestran en la figura 3.22 algunas de las trayectorias generadas con el planificador de dos dimensiones con 7 tramos. Como se puede apreciar, el añadir tramos favorece también la exploración de más zonas de alta concentración del embalse, aunque parecen concentrarse en torno a una de ellas.

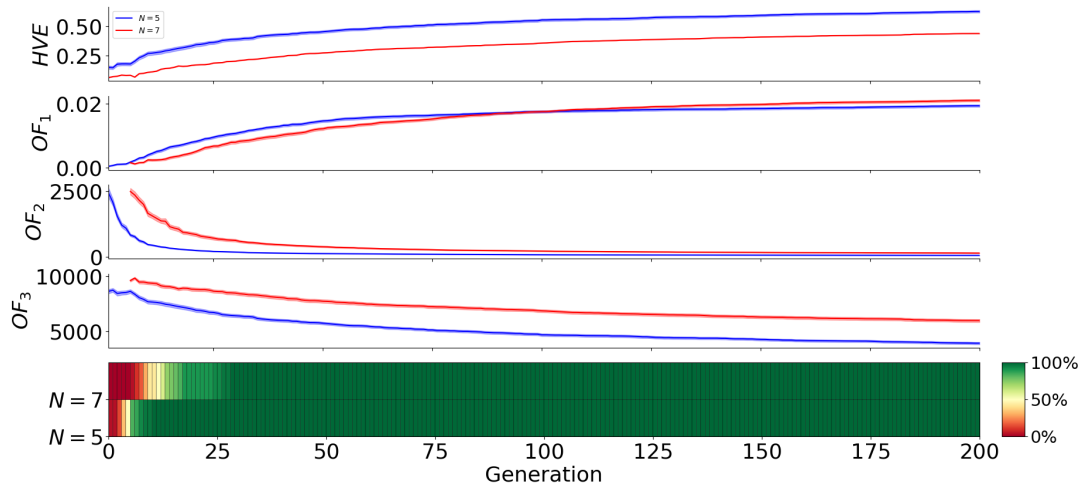


Figura 3.21: Análisis de convergencia del problema en dos dimensiones añadiendo tramos a la trayectoria

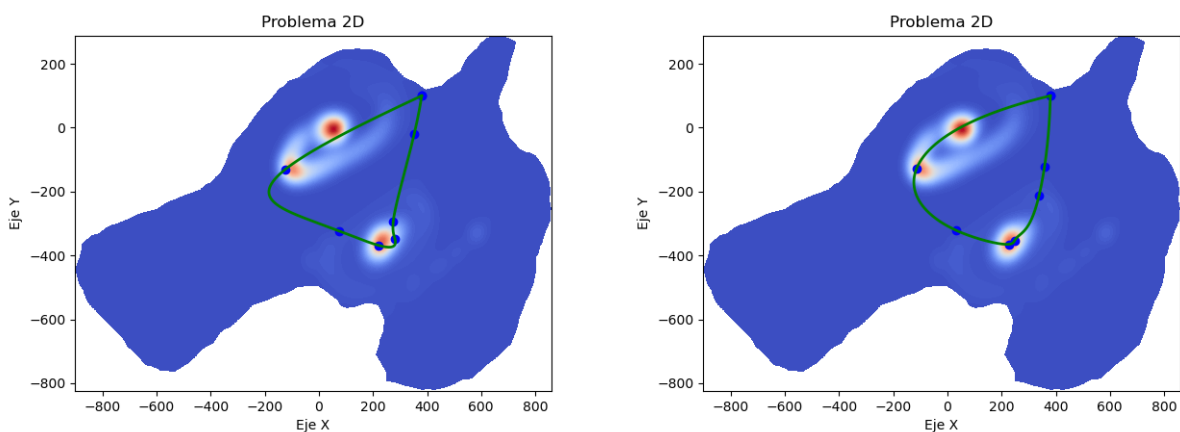


Figura 3.22: Trayectorias en dos dimensiones con 7 tramos

### 3.3.3. Problema en dos dimensiones con puntos de paso predefinidos

En esta sección se realiza el análisis de las soluciones obtenidas con trayectorias de 5 y 7 tramos, resumido en la figura 3.23, donde se puede apreciar de nuevo que el hipervolumen ocupado por las soluciones del planificador con 5 tramos es mayor que el de las soluciones obtenidas con 7 tramos. Este hecho sigue el mismo razonamiento que en el planificador anterior, menos tramos suele implicar menos longitud de trayectoria y menos tiempo de misión. En lo referente a la concentración de cianobacterias, al tener mas puntos de medida se obtienen valores más altos. Por último, el hecho de aumentar el número de tramos hace

que sea más complicado encontrar soluciones factibles, como se puede apreciar en el gráfico de porcentaje de ejecuciones con soluciones factibles en cada generación, donde el 100 % de soluciones factibles para el planificador con 7 tramos necesita más generaciones para lograr que todas las soluciones cumplan las restricciones.

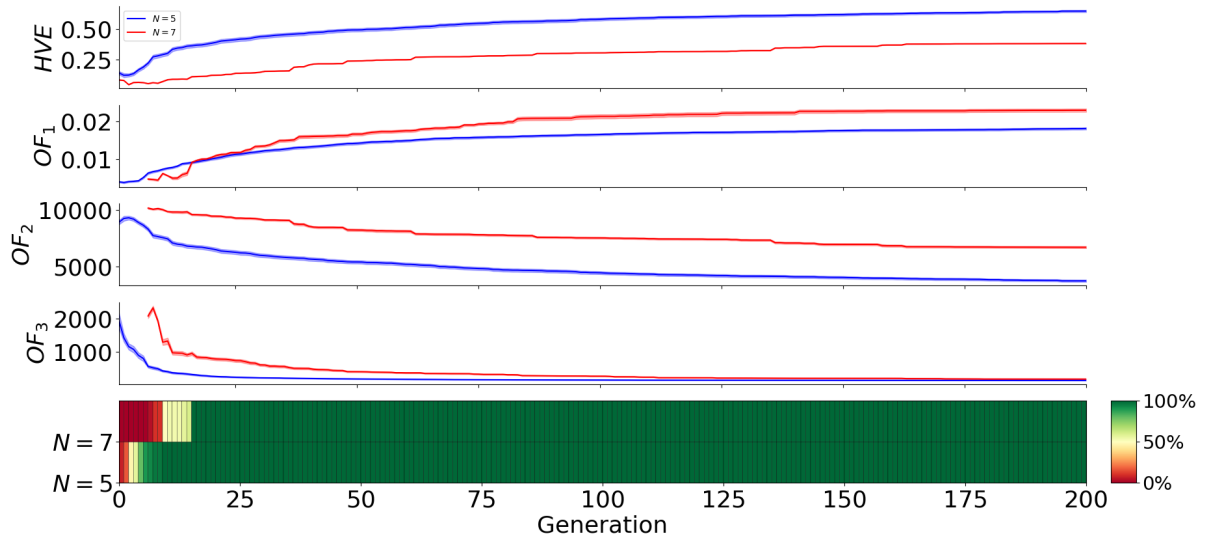


Figura 3.23: Análisis de convergencia del problema en dos dimensiones con puntos de paso predefinidos, añadiendo tramos a la trayectoria

El siguiente paso es analizar las trayectorias generadas por el planificador. En la figura 3.24 se puede apreciar que las trayectorias generadas exploran adecuadamente las distintas zonas de alta concentración de cianobacterias del embalse, pasando además por el punto de paso predefinido.

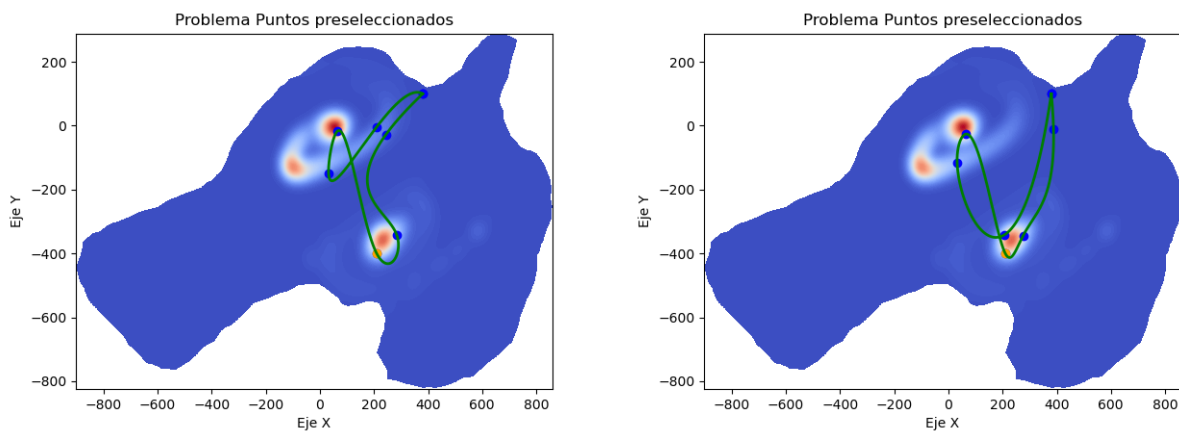


Figura 3.24: Trayectorias en dos dimensiones con puntos de paso predefinidos con 7 tramos

## 3.4. Funcionamiento del sistema ante distintos escenarios simulados

En esta sección se pondrán a prueba los tres planteamientos del problema ante las 3 simulaciones presentadas en la figura 2.5 del capítulo anterior. Esta última sección tiene como fin comprobar si las configuraciones escogidas en las secciones anteriores sirven para resolver para diferentes escenarios o lugares en los que se ubican las zonas de alta concentración de cianobacterias.

### 3.4.1. Problema en tres dimensiones

Se analizará el hipervolumen ocupado por las soluciones en los tres escenarios, con el fin de comprobar la calidad de los resultados obtenidos. En este caso no se va a realizar una comparativa ya que los escenarios no guardan relación más allá de tratarse del mismo embalse. El hecho de que en un escenario se obtengan mejores resultados se debe a las características intrínsecas del escenario. Tal y como se puede apreciar en la figura 3.25, en todos los escenarios, los hipervolumenes convergen mostrando que el planificador está dando unas soluciones adecuadas. En el caso del escenario B, la varianza en la función objetivo 1, relacionada con la medida de cianobacterias es mayor a la mostrada en el resto de escenarios. Esto puede significar que las soluciones obtenidas, y por lo tanto, las trayectorias son más distintas entre sí.

Con el análisis realizado, en las figuras 3.26 y 3.27 se encuentran algunas de las trayectorias obtenidas con el planificador en tres dimensiones dentro del escenario B y C respectivamente. Como se puede apreciar en el escenario B, las trayectorias, a pesar de estar sobre la zona de alta concentración, son muy distintas. Esto cobra sentido tras analizar la desviación en la función objetivo 1. Las trayectorias del escenario C no tienen este problema.

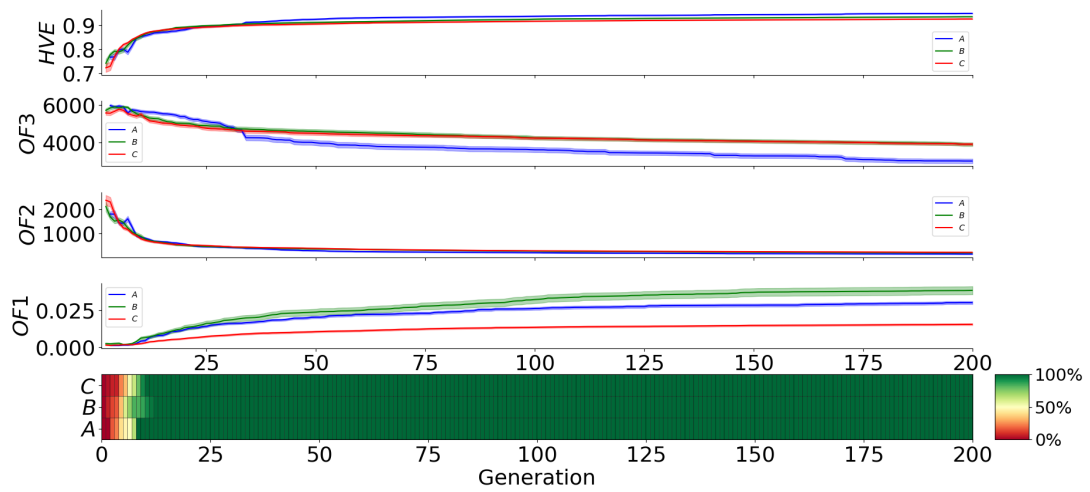


Figura 3.25: Análisis de convergencia del problema en tres dimensiones para cada uno de los escenarios de simulación

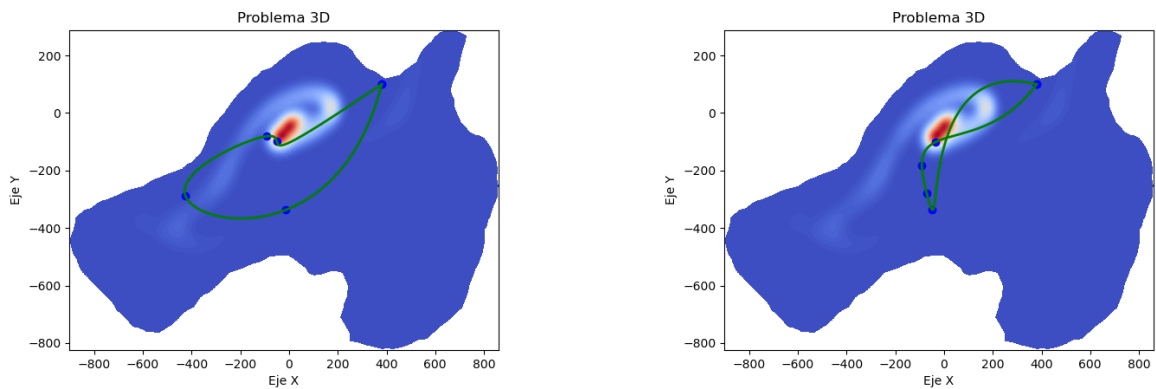


Figura 3.26: Trayectorias generadas en el escenario B para el problema en tres dimensiones

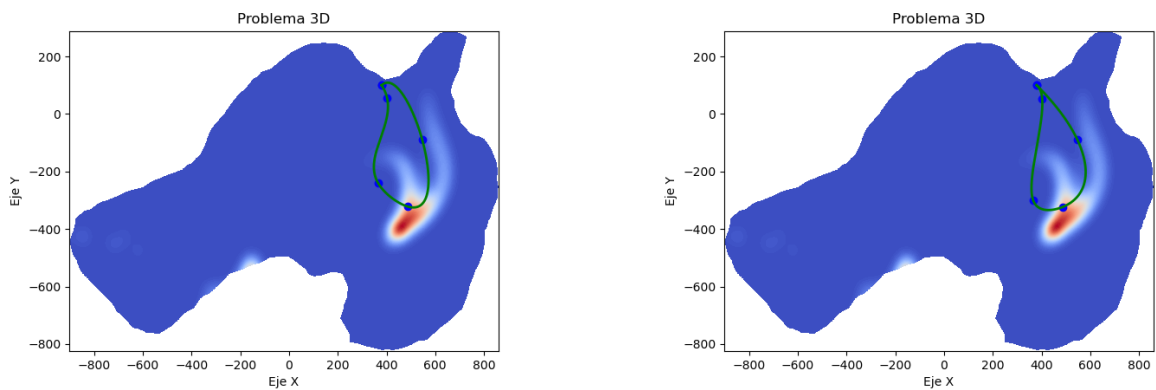


Figura 3.27: Trayectorias generadas en el escenario C para el problema en tres dimensiones

### 3.4.2. Problema en dos dimensiones

Una vez analizados los resultados del problema en tres dimensiones, se analizan las soluciones obtenidas en distintos escenarios con el planteamiento en dos dimensiones. En la figura 3.28 se puede apreciar que las soluciones convergen adecuadamente a un valor. A su vez, en cada escenario son relativamente pocas generaciones las necesarias para alcanzar el 100 % de soluciones factibles, siendo el escenario B el que más tarda en lograrlo.

El resultado obtenido en los nuevos escenarios se puede apreciar en las figuras 3.29 y 3.30 donde se muestran las trayectorias obtenidas para el escenario B y C respectivamente. En ambos casos se puede apreciar que los puntos de paso tienden a concentrarse en la única zona de alta concentración de cianobacterias, en estos escenarios donde solo existe una única zona de alta concentración, estas trayectorias son válidas ya que no necesitan explorar más zonas.

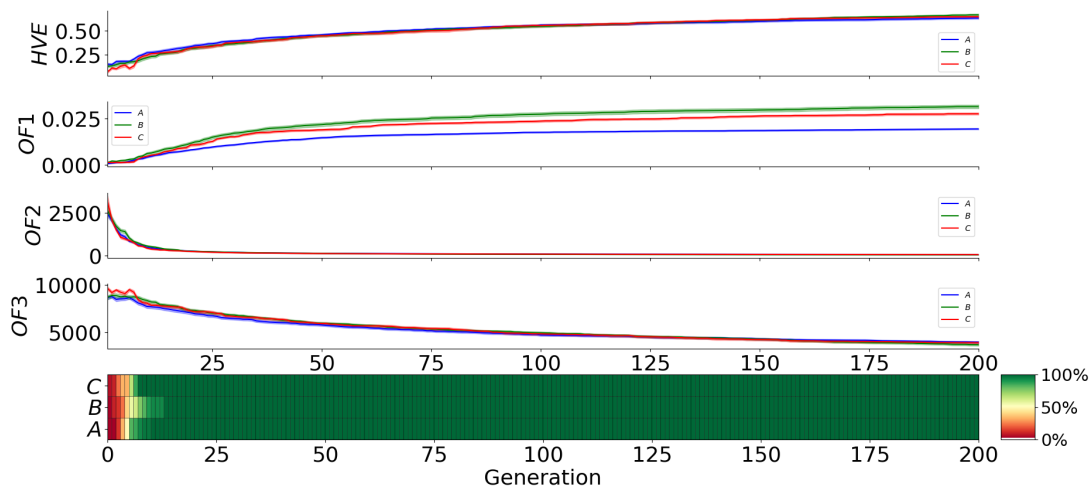


Figura 3.28: Análisis de convergencia del problema en dos dimensiones para cada uno de los escenarios de simulación

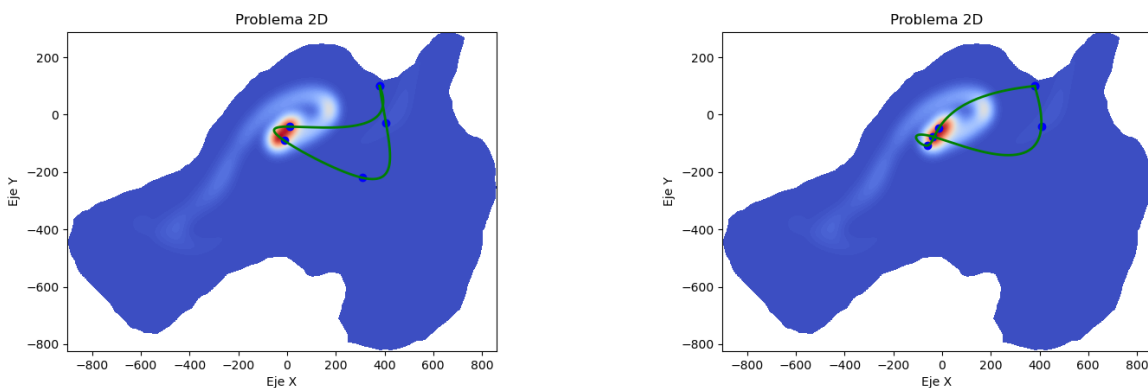


Figura 3.29: Trayectorias generadas en el escenario B para el problema en dos dimensiones

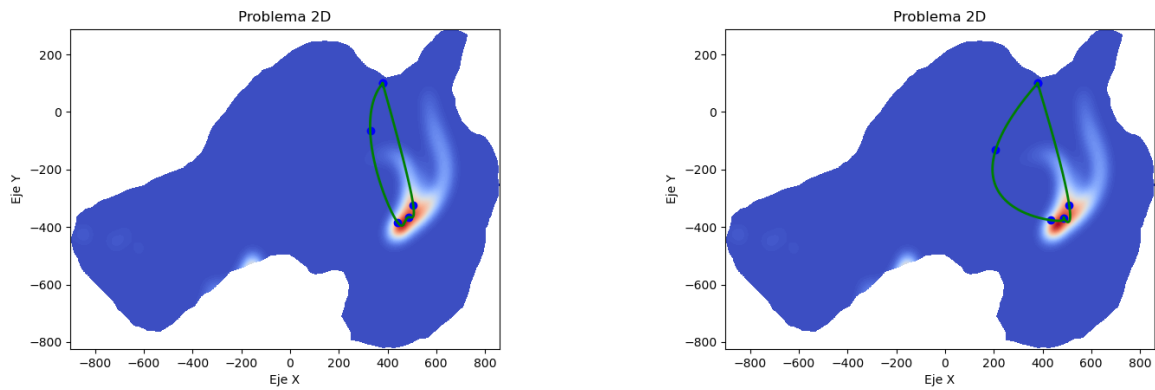


Figura 3.30: Trayectorias generadas en el escenario C para el problema en dos dimensiones

### 3.4.3. Problema en dos dimensiones con puntos de paso predefinidos

En este último caso, se analiza el comportamiento del algoritmo ante distintos escenarios pero se mantiene el punto de paso obligatorio haciendo que la trayectoria pase por una zona que no tiene una alta concentración de cianobacterias. Esto dará una visión de lo que ocurriría en la realidad si la zona donde históricamente existe una alta concentración no la hubiera. En la figura 3.31 se puede apreciar que las soluciones convergen adecuadamente y de nuevo el número de generaciones necesarias para lograr el 100% de ejecuciones con soluciones factibles es bajo.

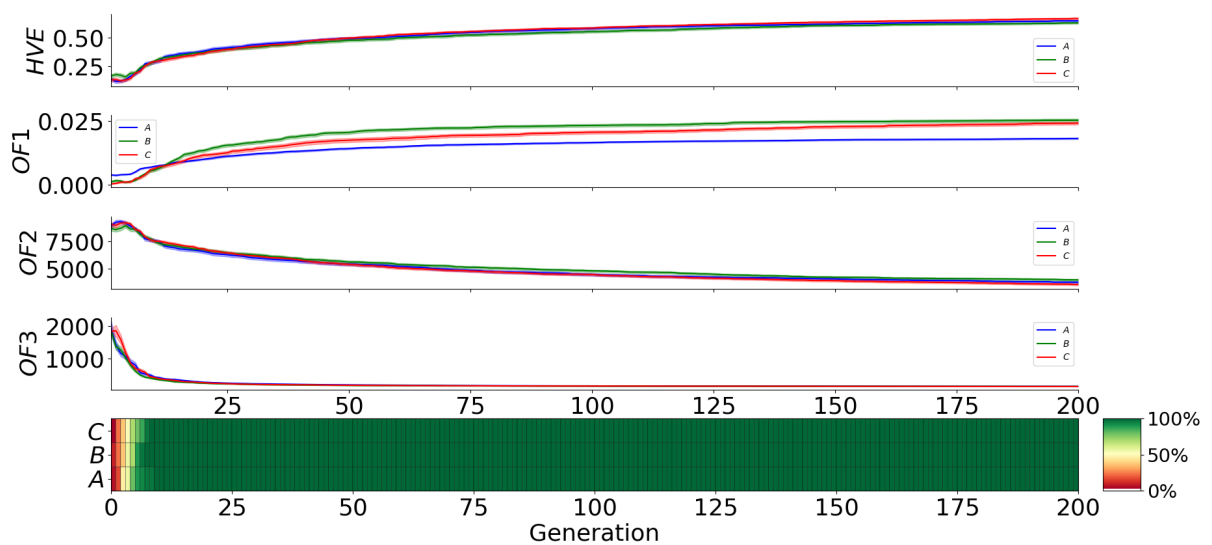


Figura 3.31: Análisis de convergencia del problema en dos dimensiones con puntos de paso predefinidos, para cada uno de los escenarios de simulación

Los resultados obtenidos se muestran en las figuras 3.32 y 3.33, donde se puede apreciar que al introducir un punto fuera de las zonas de alta concentración, la trayectoria se alarga, pero aún así el planificador logra colocar puntos de paso en estas zonas.

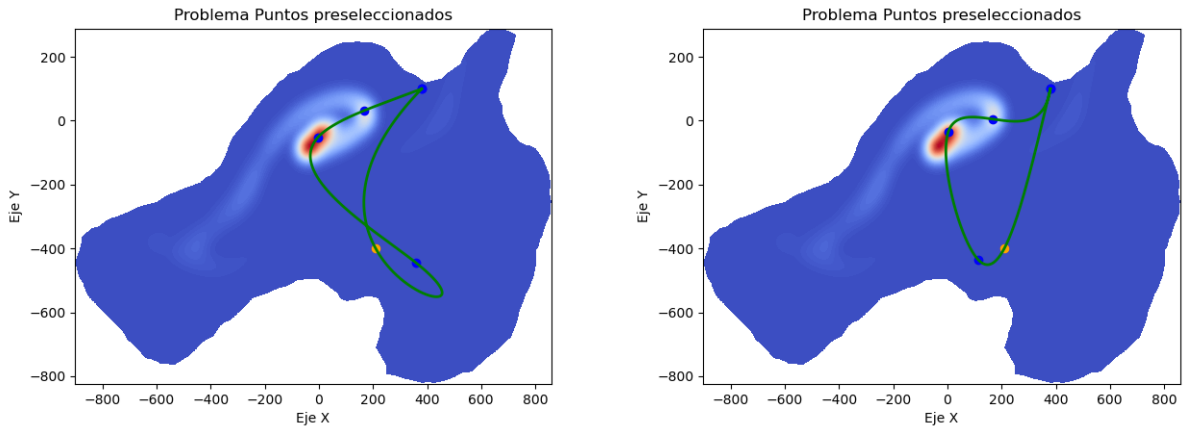


Figura 3.32: Trayectorias generadas en el escenario B para el problema en dos dimensiones con puntos de paso predefinidos

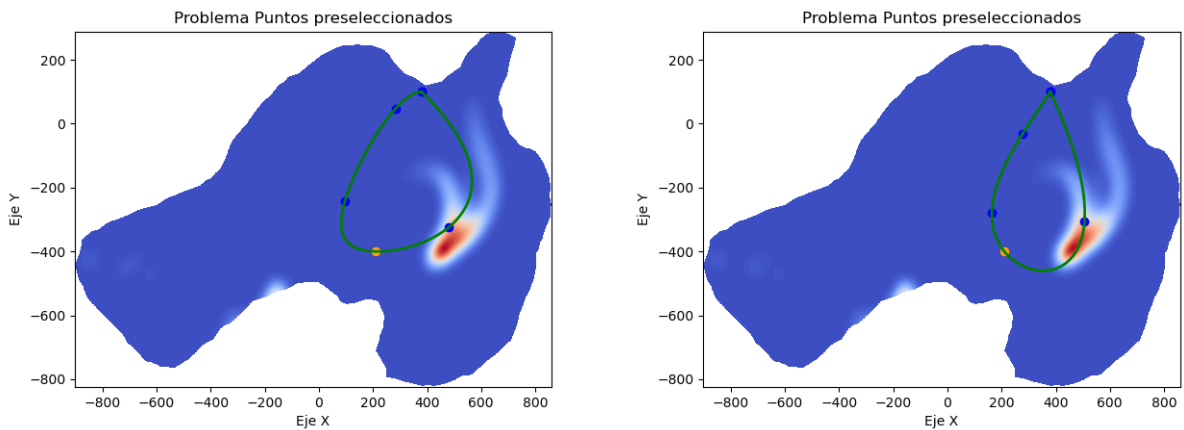


Figura 3.33: Trayectorias generadas en el escenario C para el problema en dos dimensiones con puntos de paso predefinidos

# Capítulo 4

## Conclusiones y trabajos futuros

### 4.1. Conclusiones

En este trabajo fin de máster se han presentado tres planteamientos para optimizar las trayectorias que debe seguir una embarcación (en un tipo de misiones de monitorización de la situación de las cianobacterias en un embalse) empleando algoritmos genéticos. El primer planteamiento emplea trayectorias en tres dimensiones, en las cuales, dos dimensiones se corresponden con el movimiento horizontal  $(x, y)$  realizado por la embarcación, y la tercera dimensión está asociada al movimiento vertical  $z$  realizado por la sonda. La sonda se encuentra realizando medidas continuamente, pero se puede ver afectada por las corrientes internas del embalse haciendo que la sonda se desplace. Como se ha podido apreciar en el análisis de las soluciones, este planteamiento del problema permite medir la mayor concentración de cianobacterias, pero las trayectorias no logran explorar todas las zonas de alta concentración dejando puntos de interés sin medir.

El segundo planteamiento consiste en un planificador de trayectorias en dos dimensiones. Como se ha explicado, tal y como se codifican las trayectorias, los puntos de paso de estas, serán los puntos de medida de cianobacterias. En estos puntos la embarcación se detendrá y realizará un perfil de profundidad midiendo la concentración. En este planteamiento, la sonda al bajar también se ve desplazada por corrientes internas del embalse, pero un único perfil dará las medidas de concentración necesarias para que en un determinado radio entorno a la embarcación, carezca de interés realizar otra medida. Esto hace que el enfoque en dos dimensiones cobre fuerza a la hora de seleccionar un método ya que en el punto de medida, el movimiento de la sonda estará limitado. Además, el procedimiento es análogo al realizado actualmente por los responsables de la calidad del embalse, que se trasladan a unos puntos determinados en una embarcación para realizar un perfil en profundidad. Tras realizar el análisis de las soluciones de este planteamiento, se observa que las trayectorias generadas tienen sus puntos de paso sobre las zonas de alta concentración de cianobacterias, haciendo viable la medida solo mediante perfiles en lugar de hacerlas de forma continua. Esto además

dará la posibilidad de aumentar el tiempo de misión ya que como se ha comentado en secciones anteriores, el tiempo en el que se consumen las baterías es crítico, y un sistema que realiza medidas de forma continua consumirá más que un sistema que realiza medidas de forma puntual. Si bien la concentración de cianobacterias medida es inferior a la obtenida con el planteamiento en tres dimensiones, esto no es más que por el hecho de estar realizando medidas durante toda la trayectoria. Esto hace de este método una opción especialmente interesante para abordar el problema.

Por último, se ha presentado una modificación del problema en dos dimensiones que permite al operador introducir puntos de paso donde es imperativo que el vehículo se sitúe para realizar un perfil de medidas. Este planteamiento parte de tener un registro histórico de las zonas donde suele ser alta la concentración de cianobacterias. Con una adecuada codificación de las variables del problema se ha logrado introducir estos puntos de medida sin afectar al funcionamiento del algoritmo creado para el planteamiento del problema en dos dimensiones.

Todos estos análisis vienen acompañados de distintas modificaciones que buscan mejorar la respuesta del optimizador. Si bien son varias las modificaciones que se han realizado, en este campo de estudio son necesarias más configuraciones y más pruebas sobre distintos escenarios. Sin embargo, dado que el análisis estadístico de los algoritmos genéticos requiere un elevado número de horas de procesado y tiene una gran carga computacional, no es posible evaluar todo el abanico de posibles modificaciones para obtener el mejor optimizador. A pesar de ello, con las variantes analizadas se intenta mostrar el funcionamiento y las respuestas que se obtienen al cambiar ciertos parámetros del algoritmo así como las codificaciones de las variables de decisión del problema.

Además, es importante destacar que se han alcanzado todos los objetivos propuestos al inicio del proyecto. Más concretamente.

1. Estudio del modelo predictivo de cianobacterias. En la sección 2.1.3, se ha detallado el funcionamiento del simulador de partículas capaz de predecir donde se ubicarán las grandes concentraciones de cianobacterias, describiendo las ecuaciones que definen su dinámica.
2. Definir el problema de optimización multi-objetivo. En las secciones 2.2 y 2.3 se describen los planteamientos del problema así como las funciones objetivo y restricciones empleadas en el optimizador. Con la descripción de cada uno de los planteamientos así como la codificación de trayectorias empleada, queda definido el problema que se desea resolver.
3. Estudio y aplicación del algoritmo genético NSGA-II. La sección 2.4 detalla el funcionamiento del algoritmo así como las herramientas y métodos que este usa para su

ejecución.

4. Resolver los planteamientos del problema en tres dimensiones y el planteamiento en dos dimensiones con la posibilidad de asignar puntos de paso personalizados. Empleando el método NSGA-II se ha resuelto el problema de optimización para todos los planteamientos presentados, mostrando sus resultados en el capítulo 3.
5. Analizar y comparar los resultados. Con los métodos de análisis de la sección 2.5.1, se analizan en el capítulo 3, todos los resultados arrojados por cada uno de los planificadores, con sus distintas variantes.

## 4.2. Trabajo futuro

Este trabajo se centra en resolver el problema de optimización de trayectorias de embarcaciones autónomas con distintos enfoques para realizar medidas de concentración de cianobacterias en un embalse, pero tal y como se ha podido apreciar en algunos de los resultados obtenidos, las soluciones proporcionadas por el algoritmo no siempre es capaz de recorrer todas las zonas donde es máxima la concentración. Esto puede deberse a las propias limitaciones de la embarcación, que no tiene la suficiente velocidad para explorar dichas zonas, o que las zonas se encuentran muy distanciadas entre si o con respecto a la posición inicial de la embarcación. Con vistas a solucionar este problema se plantea como trabajo futuro modificar los algoritmos presentados, de modo que como resultado se obtengan  $n$  trayectorias, siendo  $n$  el número de vehículos disponibles para realizar la misión. Con las modificaciones adecuadas, se puede tratar de favorecer la exploración de todo el embalse con una flota de embarcaciones autónomas.

Una de las posibles formas de abordar este problema, es aumentar el número de embarcaciones que realizan medidas simultáneamente en el embalse. El embalse se puede dividir en celdas, del tamaño que sea necesario (por ejemplo, tan grandes como el radio entorno a la embarcación donde una vez realizada la medida carece de interés medir) la idea consiste en añadir una nueva función objetivo que minimice el número de celdas no exploradas por las embarcaciones. Cada embarcación realizará una trayectoria distinta y cada celda explorada tomará un valor 0, mientras que las celdas no exploradas toman un valor 1. El optimizador deberá minimizar la suma de la función que indica si la celda ha sido explorada o no. También es necesario añadir restricciones para evitar la colisión de los vehículos, lo que implica que dos vehículos no pueden encontrarse en la misma posición  $(x, y)$  en el mismo instante de tiempo  $t$ .

Este es un posible enfoque, aunque recurriendo a la literatura, se pueden encontrar muchos trabajos relacionados con la planificación de trayectorias que pueden servir como base,

como es el caso de Besada-Portas et al. [2010], donde ya se trabaja el uso de optimizadores para realizar las trayectorias de un enjambre de drones.

Dado que uno de los objetivos de realizar estas medidas es validar el modelo dinámico de cianobacterias, una posible forma enfocada empleando sistemas multiagente, se basa en el artículo Acuaviva et al. [2023], donde se presenta una solución para localizar la fuente, o máximo, de un campo escalar desconocido usando un enjambre de robots móviles. Para este caso, se puede emplear el enjambre de robots para realizar las medidas referentes a la concentración de cianobacterias, cotejando estas medidas con los datos arrojados por el modelo de cianobacterias.

Estos son algunos de los enfoques que se le pueden dar al problema con el fin de mejorar los resultados presentados en este trabajo. Con estas propuestas planteadas, se abre la puerta a la colaboración de la comunidad científica para lograr que se le pueda dar el máximo aprovechamiento a un recurso tan importante para los seres vivos como es el agua.

# Bibliografía

- Avital Bechar and Clément Vigneault. Agricultural robots for field operations: Concepts and components. *Biosystems Engineering*, 149:94–111, 2016.
- Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.
- Leda Giannuzzi. Cianobacterias y cianotoxinas. In *V Congreso Internacional sobre Cambio Climático y Desarrollo Sustentable (La Plata, 14, 15 y 16 de septiembre de 2016)*, 2016.
- Denisa Soledad Pérez, Alejandro Luis Soraci, and Maria Ofelia Tapia. Cianobacterias y cianotoxinas: Rol de las microcistinas en la salud humana y animal y su detección en muestras de agua. 2008.
- United Nations. 2030 agenda for sustainable development, n.d. URL <https://sdgs.un.org/goals>. Accessed on 15/10/2020.
- Alfredo González Calvin. Monitorización automática de aguas embalsadas mediante vehículos autónomos de superficie, jun 2023.
- Mario Arzamendia, Derlis Gregor, Daniel Gutierrez Reina, and Sergio Luis Toral. An evolutionary approach to constrained path planning of an autonomous surface vehicle for maximizing the covered area of ypacarai lake. *Soft Computing*, 23:1723–1734, 2019.
- Chengke Xiong, Danfeng Chen, Di Lu, Zheng Zeng, and Lian Lian. Path planning of multiple autonomous marine vehicles for adaptive sampling using voronoi-based ant colony optimization. *Robotics and Autonomous Systems*, 115:90–103, 2019. ISSN 0921-8890. doi: <https://doi.org/10.1016/j.robot.2019.02.002>. URL <https://www.sciencedirect.com/science/article/pii/S0921889018304469>.
- Gonzalo Carazo-Barbero, Eva Besada-Portas, José Luis Risco-Martín, and José Antonio López-Orozco. Ea-based asv trajectory planner for detecting cyanobacterial blooms in freshwater. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23*, page 1321–1329, New York, NY, USA, 2023. Association for Computing

- Machinery. ISBN 9798400701191. doi: 10.1145/3583131.3590484. URL <https://doi.org/10.1145/3583131.3590484>.
- Alfredo González-Calvin, Lía García-Perez, José L Risco-Martín, and Eva Besada-Portas. Simulation, optimization and control of trajectories of asvs performing hachbs monitoring missions in lentic waters. In *2023 Winter Simulation Conference (WSC)*, pages 910–921. IEEE, 2023.
- R. Klein, J. Lemke, A. Deutz, L. Hennig, and M. Wagner. pymoo: Multi-objective optimization in python, 2021. URL <https://pymoo.org/>. Version 0.7.0.
- David Uriel Hernández-Becerril and Norberto Pastén-Miranda. Abundancia y distribución de la cianobacteria picoplanctónica synechococcus en bahía de la paz y cuenca carmen, golfo de california (junio, 2001). *Hidrobiológica*, 25(3):357–364, 2015.
- COMSOL. COMSOL Multiphysics Webpage. <https://www.comsol.com/>. Accessed on 18/05/2024.
- Carlos Adrián Correa Flórez, Ricardo Andrés Bolaños Ocampo, and Alexánder Molina Cabrera. Algoritmo multiobjetivo nsga ii aplicado al problema de la mochila. *Scientia et technica*, 2(39):206–211, 2008.
- Johannes Bader and Eckart Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary computation*, 19(1):45–76, 2011.
- Eva Besada-Portas, Luis de la Torre, Jesús M. de la Cruz, and Bonifacio de Andrés-Toro. Evolutionary trajectory planner for multiple uavs in realistic scenarios. *IEEE Transactions on Robotics*, 26(4):619–634, 2010. doi: 10.1109/TRO.2010.2048610.
- Antonio Acuaviva, Jesus Bautista, Weijia Yao, Juan Jimenez, and Hector Garcia de Marina. Resilient source seeking with robot swarms. *arXiv preprint arXiv:2309.02937*, 2023.