Programa Oficial de Doctorado en Tecnologías de la Información y las Comunicaciones

Técnicas de Inteligencia Artificial Aplicadas al Sector de las Aeronaves Pilotadas por Control Remoto

Técnicas de Intelixencia Artificial Aplicadas ao Sector das Aeronaves Pilotadas por Control Remoto

Artificial Intelligence Techniques Applied to the Remote-Controlled Aircraft Sector

Autor:

Alejandro Puente Castro

Directores:

Enrique Fernández Blanco

Daniel Rivero Cebrián

UNIVERSIDADE DA CORUÑA

Dept. de Ciencias de la Computación y Tecnologías de la Información

UNIVERSIDADE DA CORUÑA

Dr. Daniel Rivero Cebrián, profesor en el área de Ciencias de la Computación e Inteligencia Artificial, perteneciente al Departamento de Ciencias de la Computación y Tecnologías de la Información, Facultad de Informática, Universidade da Coruña.

Y

Dr. Enrique Fernández Blanco, profesor en el área de Ciencias de la Computación e Inteligencia Artificial, perteneciente al Departamento de Ciencias de la Computación y Tecnologías de la Información, Facultad de Informática, Universidade da Coruña.

HACEN CONSTAR QUE:

La memoria titulada «Técnicas de Inteligencia Artificial Aplicadas al Sector de las Aeronaves Pilotadas por Control Remoto» ha sido realizada por D. Alejandro Puente Castro, bajo su dirección en el Departamento de Ciencias de la Computación y Tecnologías de la Información y constitute la Tesis Doctoral que presenta para obtar al Grado de Foctor en Informática de la Universidade da Coruña.

En A Coruña, a 11 de septiembre de 2023.

Fdo.: Enrique Fernández Blanco

Fdo.: Daniel Rivero Cebrián

Agradecimientos

Me gustaría empezar expresando mi gratitud a mis directores el Dr. Enrique Fernández Blanco y el Dr. Daniel Rivero Cebrián por todas sus enseñanzas. También, por su dedicación y paciencia a lo largo de todo este camino. Sin ellos, toda esta investigación y sus resultados no hubiesen sido posibles.

Quiero expresar mis agradecimientos al Dr. Alejandro Pazos Sierra por darme la oportunidad y acogerme en el grupo RNASA-IMEDIR, del que estoy orgulloso de pertenecer.

Dentro del grupo RNASA-IMEDIR quiero dar las gracias a todos mis compañeros. No solo por apoyarme, sino también por su compañía y amistad. He aprendido mucho de ellos y espero haber sido de ayuda para ellos.

También quiero mostrar mis agradecimientos a los compañeros de la Universidade da Coruña con los que tengo el placer de compartir mi día a día. Quiero agradecer toda su ayuda y todas sus facilidades para poder compaginar mi trabajo con mi investigación.

A mi familia quiero agradecerles el haber llegado hasta aquí. Especialmente a mis padres. Sin los cuidados y apoyo de todos, esto hubiese sido más difícil. Tal vez no entiendan nada de lo que hago, pero saben que es importante para mí.

A todos mis amigos quiero darles las gracias por su compañía. Gracias a ellos por escucharme hablar todos los días de la tesis. Tal vez, ellos tampoco entienden nada de lo que hago, pero, tras todos estos años, ya se han acostumbrado.

En general, quiero expresar mi agradecimiento a todas las personas que me han acompañado en todo este proceso. Algunas me siguen acompañando y otras ya no, pero, de un modo u otro, han hecho de todo este tiempo una historia interesante e inolvidable.

Finalmente, quiero dar las gracias a toda persona a la que esta investigación sea de interés o sirva de inspiración. Esta memoria no es el fin de una investigación, sino el fin de un comienzo. En ella, se muestran las bases para construir un largo camino en el que invito a todo el mundo a recorrer y descubrir.

Resumen

El principal objetivo de esta Tesis Doctoral es estudiar el uso de técnicas para el control de enjambres heterogéneos de Aeronaves Pilotadas Remotamente (RPA o UAV, por sus siglas en inglés), coloquialmente conocidos como drones. Esta tesis está apoyada por tres publicaciones científicas indexadas en el sistema Journal Citation Report. Uno de ellos es el estudio de la aplicación de estas y otras técnicas en el ámbito de los enjambres de UAV. Los dos restantes proponen modelos para su aplicación en mapas simulados sin obstáculos y con obstáculos fijos.

La importancia del estudio de estas técnicas para el control de enjambres de UAV demuestra que emplear un grupo heterogéneo de UAV con total libertad de movimiento permiten realizar las tareas de manera más rápida que empleando solo uno. Además, las técnicas de Aprendizaje por Refuerzo demuestran que son capaces de adaptarse a la situación del entorno y a sus obstáculos. El Aprendizaje por Refuerzo es un conjunto de técnicas de la Inteligencia Artificial que buscan resolver ciertos tipos de tareas basándose en la interacción con un entorno. Todo esto es realizado basándose en la recompensa o refuerzo que provoca realizar diferentes acciones en dicho entorno. Así, si una acción es la correcta, el refuerzo es positiva y, de ser incorrecta, el refuerzo es negativo. Al poder emplear un único sistema para el control de los UAV, se reduce la necesidad de tener un operador por cada aeronave, reduciendo los costes asociados a la operación.

Para una mejora en la capacidad de estas técnicas, se han empleado Redes de Neuronas Artificiales por su capacidad de extraer conocimiento a partir de patrones. Así, se consigue mejorar la capacidad de adaptación de los modelos propuestos a los diferentes entornos en los que es probado.

Resumo

O principal obxectivo desta Tese Doctoral é estudar o uso de técnicas para o control de enxames heteroxéneos de Aeronaves Pilotadas Remotamente (RPA ou UAV, polas súas siglas en inglés) , coloquialmente coñecidos como drons. Esta tese está apoiada por tres publicacións científicas indexadas no sistema Journal Citation Report. Un deles é o estudo da aplicación destas e outras técnicas no ámbito dos enxames de UAV. Os dous restantes propoñen modelos para a súa aplicación en mapas simulados sen obstáculos e con obstáculos fixos.

A importancia do estudo destas técnicas para o control de enxames de UAV demostra que empregar un grupo heteroxéneo de UAV con total liberdade de movemento permiten realizar as tarefas de maneira máis rápida que empregando só un. Ademais, as técnicas de Aprendizaxe por Reforzo demostran que son capaces de adaptarse á situación da contorna e aos seus obstáculos. A Aprendizaxe por Reforzo é un conxunto de técnicas da Intelixencia Artificial que buscan resolver certos tipos de tarefas baseándose na interacción cunha contorna. Todo isto é realizado baseándose na recompensa ou reforzo que provoca realizar diferentes accións na devandito contorna. Así, se unha acción é a correcta, o reforzo é positiva e, de ser incorrecta, o reforzo é negativo. Ao poder empregar un único sistema para o control dos UAV, redúcese a necesidade de ter un operador por cada aeronave, reducindo os custos asociados á operación.

Para unha mellora na capacidade destas técnicas, empregáronse Redes de Neuronas Artificiais pola súa capacidade de extraer coñecemento a partir de patróns. Así, conséguese mellorar a capacidade de adaptación dos modelos propostos ás diferentes contornas nos que é probado.

Abstract

The main objective of this Doctoral Thesis is to study the use of the techniques for the control of heterogeneous swarms of Remotely Piloted Aircraft (RPA) or Unmanned Aerial Vehicles (UAV), colloquially known as drones. This thesis is supported by three scientific publications indexed in the Journal Citation Report system. One of them is the study of the application of these and other techniques in the field of UAV swarms. The remaining two propose models for their application in simulated maps without obstacles and with fixed obstacles.

The importance of the study of these techniques for UAV swarm control demonstrates that using a heterogeneous group of UAVs with full freedom of movement allows tasks to be performed faster than using only one. In addition, Reinforcement Learning techniques prove that they are able to adapt to the environmental situation and its obstacles. Reinforcement Learning is a set of Artificial Intelligence techniques that seek to solve certain types of tasks based on interaction with an environment. All this is done based on the reward or reinforcement caused by performing different actions in that environment. Thus, if an action is correct, the reinforcement is positive and, if it is incorrect, the reinforcement is negative. By being able to use a single system to control UAVs, the need for one operator per aircraft is reduced, reducing the costs associated with the operation.

To improve the capability of these techniques, Artificial Neural Networks have been used for their ability to extract knowledge from patterns. Thus, it is possible to improve the adaptability of the proposed models to the different environments in which they are tested.

Estructura de la Tesis Doctoral

Esta Tesis Doctoral está dividida en dos partes. La primera de ellas está centrada en la introducción al problema, la descripción de todos los aspectos necesarios para la experimentación, los resultados obtenidos y posibles líneas de investicación futuras. Posteriormente, la segunda parte consta de las publicaciones científicas resultantes del trabajo desarrollado.

La primera parte consta de siete capítulos. En el primer capítulo se introduce el problema y se presenta la justificación. En el segundo capítulo se especifica la hipótesis de partida los objetivos específicos. En el tercer capítulo se revisan todos los conceptos básicos del campo de la aplicación de los algoritmos propuestas aplicadas al dominio del problema. El cuarto capítulo hace una descripción de todos los materiales y métodos empleados para la realización para el proceso de investigación. En el quinto capítulo se discuten los resultados obtenidos. En el sexto capítulo se exponen las conclusiones obtenidas durante el desarrollo de esta Tesis Doctoral. Finalmente, en el séptimo capítulo se muestran alguna las líneas de investigación futuras que parten de esta Tesis Doctoral.

La segunda parte, presenta en ANEXOS las publicaciones obtenidas del trabajo experimental desarrollado en toda la investigación. Éstas son las siguientes:

1. Alejandro Puente-Castro, Daniel Rivero, Alejandro Pazos, Enrique Fernandez-Blanco. **A review of artificial intelligence applied to path planning in UAV swarms**. Neural Computing and Applications 34, 153–170 (2022). https://doi.org/10.1007/s00521-021-06569-4

2. Alejandro Puente-Castro, Daniel Rivero, Alejandro Pazos, Enrique Fernandez-Blanco. **UAV swarm path planning with reinforcement learning for field prospecting**. Applied Intelligence 52, 14101–14118 (2022). https://doi.org/10.1007/s10489-022-03254-4

3. Alejandro Puente-Castro, Daniel Rivero, Eurico Pedrosa, Artur Pereira, Nuno Lau, Enrique Fernandez-Blanco. **Q-Learning based system for path planning with unmanned aerial vehicles swarms in obstacle environments**. Expert Systems with Applications, 121240 (2023). https://doi.org/10.1016/j.eswa.2023.121240

Índice general

Índice de Figuras

Índice de Términos y Abreviaturas

Primera Parte

Capítulo I

Introducción

Se conoce como dron o Aeronave Pilotada Remotamente (RPA, por sus siglas en inglés), o también conocido en inglés como *Unmanned Aerial Vehicle* (UAV), a toda aeronave que no es operada por personal que se encuentra a bordo (Nex y Remondino 2014). Este tipo de aeronaves tiene sus orígenes en el ámbito militar, pero se ha ido extendiendo a otros ámbitos debido a su potencial (Gonzalez-Aguilera y Rodriguez-Gonzalvez 2017) y se utilizan ampliamente en diferentes aplicaciones, tanto profesionales como recreativas. Los UAV representan una de las tecnologías más desafiantes y con mayor potencial de la actualidad. Inicialmente limitados a usos militares (Buckley y Buckley 1999), ahora se están extendiendo a diferentes sectores comerciales e industriales (Akhloufi et al. 2019).

El principal motivo de su potencial es su capacidad para adaptarlos a las necesidades del momento. Su estructura, configuración y equipamiento varían en función de la tarea a realizar (Hassanalian et al. 2015). Poder disponer de diferentes tipos de configuraciones y equipamiento implica una mejora en términos de consumo eléctrico, tiempo de funcionamiento o reducción de los riesgos de seguridad en la operación. Por lo tanto, se traduce en una reducción de costes gracias a la mejora de la eficiencia de la operación.

Actualmente, destacan muchas aplicaciones de los UAV como la fotografía o el rescate aéreo. Sin embargo, con la creciente popularidad y uso de UAV para aplicaciones de consumo, el número de incidentes con UAV está aumentando drásticamente (Majd et al. 2018). Este aumento de los accidentes aéreos se debe al incremento de este tipo de tráfico aéreo y al desconocimiento de su uso por parte de algunos usuarios. Principalmente, porque muchos vehículos aéreos no tripulados pueden adquirirse sin licencia ni pruebas de aptitud.

Gran parte de las tareas arriesgadas o laboriosas suelen requerir el uso de múltiples UAV simultáneamente. A estas agrupaciones sin formación se las conoce como

enjambres de UAV y pueden estar constituidas por aeronaves iguales o de diferentes características, lo que se conoce como enjambres homogéneos o heterogéneos. Su uso se debe, en particular, a la gran cantidad de tiempo necesario para su funcionamiento y a la limitada autonomía de estos pequeños vehículos. Cuando están trabajando, los vehículos disponibles asumen la función de aquellos que fallan y no pueden seguir con su labor. Así, al emplear enjambres, la tarea se desarrolla en paralelo y el tiempo necesario se acorta en comparación con cuando se utiliza cada UAV uno por uno. Por ello, cada vez más sectores están sustituyendo el uso individual de los UAV por su uso colectivo (Tahir et al. 2019), puesto que permiten reducir los problemas asociados, como la poca autonomía que poseen.

Esta forma colectiva de afrontar el problema está sujeta a diversos retos. En la mayoría de los países, un operario solo puede manejar un vehículo, por lo que un aumento en el número de operarios supone el aumento de costes. Sin embargo, al tratarse de vehículos autónomos en enjambre, se necesitan de mecanismos o sistemas que controlen sus rutas y combinen sus datos de manera autónoma, aumentando el coste de ingeniería (Albani et al. 2017). A pesar de ello, implica una mejora con respecto a emplear diferentes operarios, puesto que el coste de energía solo es necesario cuando se desarrolla el sistema y se reutiliza en cada vuelo, mientras que los operarios son necesarios cada vez que se requiera dicho vuelo.

Una ventaja adicional importante del uso de enjambres es que estos pueden ayudar a solucionar el mayor y más conocido inconveniente de este tipo de tecnología, que es la poca autonomía que ofrece. Esto hace que se requieran de múltiples baterías o de múltiples paradas para recargar y puede prolongar la operación en el tiempo (Lin et al. 2018). Aumentando, así, los costes en gran medida.

Estos sistemas deben tener unas capacidades de operación rápidas y precisas. Tales capacidades implican tener en cuenta diversos factores que el sistema debe considerar (Sharma et al. 2022). Por ello, deben ser capaces de percibir y gestionar perfectamente el entorno. Además, los sistemas deben conocer las limitaciones de interacción con el entorno. Hacer que el sistema gestione estos factores implica que le sean descritos con precisión, por lo que puede ser un reto altamente complejo que un programador lo haga explícitamente.

Una posible ayuda puede ser el uso de Inteligencia Artificial (IA). Según Stuart Russell, reconocido investigador y profesor en el campo, la IA se refiere al estudio de agentes inteligentes, es decir, modelos matemáticos que pueden percibir su entorno y tomar acciones para maximizar sus posibilidades de éxito en el logro de sus objetivos (Russell y Norvig 2021). De esta forma, se ahorra tiempo de análisis del problema y de la solución, a la vez que se obtienen resultados más precisos. Ante estas capacidades, la IA ofrece un conjunto de herramientas atrayentes debido a la flexibilidad y precisión

de sus técnicas.

La IA se divide en diferentes ramas. La elección de cada una de ellas depende del problema a resolver y de los posibles datos a manejar. Una rama muy extendidas es el Aprendizaje Automático o *Machine Learning* (ML). Se trata de una rama de la IA que se enfoca en el desarrollo de algoritmos y modelos que permiten a las computadoras aprender y mejorar automáticamente a partir de datos y experiencias previas, sin ser programadas explícitamente (Bishop y Nasrabadi 2006). Un ejemplo de uso, es la clasificación o la combinación de los datos que capturan los UAV de un enjambre. De esta manera, se consigue la reducción de tiempo del uso de múltiples UAV junto con la reducción de costes al necesitar un único operario.

Dentro del ML se puede encontrar un amplio abanico de paradigmas, cada una con un comportamiento diferente. Entre ellas destaca el Aprendizaje por Refuerzo o *Reinforcement Learning* (RL), que se basa en la interacción de un agente con un entorno para aprender a tomar decisiones y acciones óptimas. El agente aprende mediante la experiencia de prueba y error, recibiendo retroalimentación en forma de recompensas o castigos según las acciones que realiza (Sutton y Barto 2018). El uso de estas técnicas ya ha sido empleado para el control de UAV, tanto en uso individual (Wang et al. 2019), como en su uso colectivo (Liu et al. 2020) en diferentes ámbitos. Sin embargo, estos trabajos están limitados a la navegación de UAV individuales o en formación y con necesidad de mucha información del mapa adicional a los límites del polígono que delimitan el terreno. Así mismo, han demostrado una gran flexibilidad con aplicaciones que van desde la seguridad y el rescate al estudio topográfico. Un ejemplo muy claro de alto potencial de la aplicación de UAV en enjambres es la agricultura.

El continuo crecimiento de la población mundial, junto con la disminución de los recursos disponibles por el calentamiento global, plantea el problema del uso inteligente de los recursos (Maja y Ayano 2021). Esto es de gran importancia, especialmente, en el campo de la producción de alimentos agrícolas y la explotación óptima del suelo. Para ello, surge la Agricultura de Precisión, que procura la maximización de la producción en base a la información recabada además de reducir costes y recursos asociados (Delavarpour et al. 2021). En los últimos años, el uso de técnicas autónomas para inspeccionar el estado de la producción en la agricultura y recabar información es un factor determinante (Lytridis et al. 2021). La robótica es introducida en este campo aportando soluciones interesantes y eficaces a cada una de las fases de la producción agrícola (Oliveira et al. 2021).

Dentro del campo de la robótica agrícola, los UAV han supuesto una revolución en cuanto a la captura de datos de manera rápida, precisa y de costes reducidos (Tsouros et al. 2019). En comparación con la tecnología satelital, el uso de UAV es muy eficaz debido a que pueden ofrecer a los agricultores una visión a vista de pájaro de sus

campos, lo que les permite realizar evaluaciones más precisas (Gago et al. 2015). En particular, la popularidad el uso de UAV da la oportunidad de obtener un estudio a nivel multiespectral de la zona. Poder recuperar datos de diferente índole de manera conjunta y en un formato visual de fácil interpretación permite hacer un mejor uso del tiempo de los agricultores, en lugar de, simplemente, tener que hacer un estudio a simple vista del cultivo (Deng et al. 2018). Todo esta información permite al agricultor conocer el estado del cultivo en cualquiera de sus fases sin hacerle caminar a ciegas por un campo que puede ser más alto que su cabeza en búsqueda de posibles problemas en sus cultivos (Tripicchio et al. 2015). A pesar de todas estas bondades, pocas son las opciones con técnicas de RL aplicadas en el caso de la agricultura. Por otra parte, el uso de las técnicas en enjambres de UAV para agricultura está experimentando un gran crecimiento (Aslan et al. 2022).

Este trabajo explora el uso de técnicas de RL para navegación autónoma que permita el control simultáneo de enjambres de UAV sin formación grupal de vuelo predefinida en diferentes mapas. El uso paralelo y autónomo de estos dispositivos permitiría minimizar el tiempo de ejecución total y el uso de la batería particular. Así, las técnicas de RL podrían permitir la navegación de enjambres de UAV de forma óptima mediante el cálculo de sus rutas, manteniendo un reducido consumo energético en base a la interacción de cada aeronave con el entorno que la rodea sin necesitar modelar el comportamiento de forma explícita. Con esto, se consigue un sistema adaptativo que aprenda de casos nuevos y se actualice para ofrecer siempre los mejores resultados posibles. También hay que destacar que al no tener que mantener las restricciones de una formación de vuelo, podrá obtener mejores rutas ya que no se pierden capacidades de movimiento individual. Es decir, de tener que conservar una formación, el movimiento de cada UAV se ve condicionado por los movimientos de los demás UAV del enjambre. Además, no sería necesario establecer las leyes que rijan la formación de este grupo y tampoco habría que definir un mecanismo para indicar qué UAV sería el empleado como referencia para los demás para mantener la formación durante el vuelo.

Capítulo II

Objetivos

Este capítulo explica la hipótesis de partida que motiva esta Tesis Doctoral. Además, se listan los objetivos específicos para abordar el problema propuesto.

2.1 Hipótesis de Partida

Como hipótesis de partida se plantea la posibilidad del uso de técnicas de RL para realizar en el control de enjambres de UAV para tareas colaborativas que impliquen recorrer la totalidad de un terreno, como la prospección y captura de datos de terrenos u otras superficies, sin necesidad de aportar información adicional del mapa para dirigir el cálculo de las rutas, como balizas o puntos de destino (Zhou et al. 2021). Es decir, en vez de emplear estas técnicas para establecer rutas con un origen y un destino, se procura que los UAV recorran todo el mapa maximizando la región de datos capturada. Todo esto, manteniendo la máxima eficiencia posible de cada ruta. Es decir, en el menor tiempo posible y con el menor número de pasos. De este modo, se optimizaría el cálculo automático de las mejores rutas posibles para para cada UAV según el mapa y, así, optimizando los costes de tiempo y personal.

Si bien ya existen trabajos que actualmente ya han unido los campos del manejo autónomo de enjambres de UAV con técnicas de RL, todos necesitan aportar información adicional del mapa como puntos de control o mapas de calor con las distancias a los obstáculos. Esto implica tener conocimiento adicional más allá de la propia topología del mapa. Además, establecer información adicional supone incrementar el riesgo de que se generen sesgos a la hora de determinar las rutas de vuelo.

2.2 Objetivos Específicos

Para alcanzar la hipótesis de partida, previamente descrita, se propone el cumplimiento de los siguientes objetivos específicos:

- Análisis del estado de la cuestión basado en una revisión bibliográfica en el dominio de aplicación del problema. Así, se analizará el conjunto de aplicaciones de la IA para el control de enjambres de UAV aplicados a distintos dominios como el rescate, la agricultura o el despliegue de redes. En la selección de este conjunto debe tenerse en cuenta la novedad y la relevancia de las publicaciones de manera que esté formado por aquellas de mayor impacto en los últimos cinco años. Todo esto permitirá conocer las diferentes aproximaciones empleadas para resolver diferentes problemas de control de UAV.
- Exploración del uso de técnicas de RL utilizadas en el control colaborativo de robots. Poder conocer sus fortalezas y debilidades permite desarrollar posteriores técnicas de RL combinadas con otros métodos de ML que permitan el control de enjambres de UAV sin una estructura grupal predefinida probadas en experimentos controlados y reproducibles que ayuden a determinar las mejores para su aplicación en el dominio de los enjambres de UAV.
- Validación de las técnicas y algoritmos en conjuntos de pruebas definidos. Implica la definición de un conjunto de mapas con los que evaluar el rendimiento, tanto en correctitud como velocidad, del sistema. Permitirá analizar el posible comportamiento en entornos controlados y, además, conocer las bondades y limitaciones de cada modelo ante cada situación. El análisis estadístico de los modelos para la asistencia de enjambres de UAV probados permitirá analizar los efectos de las diferentes configuraciones de los modelos y en qué medida afectan a las rutas obtenidas.

Capítulo III

Estado de la Cuestión

Este capítulo describe el estado de la cuestión actual. Se realiza una revisión de las publicaciones existentes sobre el uso de técnicas de Inteligencia Artificial (IA) aplicadas a la robótica colaborativa para resolver problemas de planificación de rutas.

3.1 Enjambres de UAV

Muchas son las aplicaciones de las técnicas de Inteligencia Artificial (IA) que se puede encontrar en el campo de los enjambres de UAV. Todas estas técnicas tienen en cuenta las capacidades y las limitaciones de los propios UAV. Por ejemplo, tienen la ventaja de que estas aeronaves permiten gran libertad de movimiento, pero eso supone mayor complejidad a la hora de necesitar movimientos precisos. Por ello, se exponen las publicaciones agrupadas por el tipo de técnica que emplean.

3.1.1 Aprendizaje por Refuerzo

Comenzando con las técnicas del Aprendizaje por Refuerzo (RL), se aprecia el ejemplo de Hung y Givigi (2016) donde gestionan un tipo de enjambres de formación fija llamados bandadas de UAV (*UAV flocks* en inglés). Estas aeronaves son del tipo de ala fija (Figura III.1) y en el grupo hay un líder y un conjunto de seguidores. Así, consiguen grupos de UAV autónomos que se mueven de forma sincronizada mediante técnicas de *Q-Learning* (Watkins y Dayan 1992), similar a una bandada de pájaros durante los movimientos migratorios (Figura III.2). El uso de una aeronave líder mejora el tiempo de cálculo, ya que es importante mejorar la trayectoria líder y las demás se derivarían de ella. Sin embargo, en caso de fallo del UAV líder, sería más costoso recalcular todas las trayectorias porque habría que determinar cuál sería el nuevo líder. Si no hubiera dependencia entre las trayectorias o formación fija, solo se verían afectados los UAV más cercanos a la aeronave caída, mientras que los demás podrían seguir con la operación. Además, el uso de UAV de ala fija limita mucho su aplicación debido a su

control más complejo y a su menor capacidad de vuelo estacionario (Beard et al. 2014).

Figura III.1 Ejemplo de UAV de ala fija. Estos sistemas presentan una estructura formada por dos o más alas que no son móviles y que, además, su vuelo es similar al de un avión tradicional (Beard et al. 2014).

Figura III.2 Ejemplo de una bandada de pájaros volando en formación. Estas aves procuran mantener su formación durante el vuelo para mejorar la eficiencia de su vuelo (Heppner et al. 2009).

Si bien *Q-Learning* ha demostrado sus capacidades en el dominio, existen muchos trabajos que proponen variaciones del algoritmo con el fin de mejorar sus capacidades. Un ejemplo es el trabajo de Hafez et al. (2017), en el que se combinan Sistemas Difusos (Zadeh 2023) con *Q-Learning* para el control de enjambres de UAV de uso militar. Su método muestra robustez ante fallos. De este modo, puede recuperarse en caso de caída de un UAV. En el artículo, los UAV tienen que mantener una formación, lo que condiciona el cálculo de las trayectorias. Además, se han probado en entornos cerrados de interior, por lo que no contemplan cambios en la dirección del viento ni obstáculos dinámicos como pájaros. También, combinado con Sistemas Difusos, Su et al. (2016) hacen uso de matrices difusas como función de recompensa para calcular trayectorias de grupos de drones. En este caso, el uso de técnicas de agrupación o *clustering* (Omran et al. 2007) para la distribución inicial del terreno lo hace menos dependiente de los parámetros de inicialización que en otros trabajos. Por lo tanto, se debe hacer un buen estudio previo de los parámetros para que puedan ser utilizados en una variedad de entornos reales. Siguiendo con la computación difusa, en 2020, Yang et al. (2020) también proponen combinarla con *Q-Learning*. Un punto muy importante de su proyecto es que es uno de los pocos que tienen en cuenta el nivel de batería de cada aeronave.

También en 2020, Chen et al. (2020) proponen un sistema *Q-Learning* multiagente basado en acciones restringidas. Así, facilitan la toma de decisiones autónoma en vuelo teniendo en cuenta la incertidumbre de la localización de cada punto de referencia en el mapa. Además, su sistema se probó con un número diferente de UAV. Demostraron que, a medida que aumenta el número de UAV, se incrementa la tasa de fallos en las tareas, debido a un mayor número de aeronaves, pero que no afecta a la completitud del objetivo.

De todo el abanico de posibilidades, la variante que más destaca dentro del *Q-Learning* es el *Deep Q-Learning*, también conocido como DQN (Fan et al. 2020), por su poder de generalización y control (Mnih et al. 2015). Uno de los enfoques más importantes es el propuesto por Roudneshin et al. (2019) en el que emplean Redes Neuronales Artificiales o RR.NN.AA. (Krogh 2008) para controlar enjambres compuestos por UAV y robots heterogéneos. Este trabajo, de carácter militar, no presenta un trabajo puramente en UAV sino que añade robots terrestres conocidos como o *Unmanned Ground Vehicle* o UGV (Gage 1995) (Figura III.3). Sin embargo, se trata de un problema de planificación de trayectorias de enjambres con mayor dificultad que utilizando únicamente UAV. Dicho aumento en la dificultad del problema se debe a las diferentes limitaciones que presentan los vehículos aéreos y terrestres. Así, un vehículo terrestre puede encontrarse con obstáculos no geográficos y tiene movimientos más limitados. Como utilidad práctica, exponen las capacidades para su uso en misiones de búsqueda y rescate. También se desenvuelve en condiciones de emergencia o rescate el trabajo de Baldazo et al. (2019) donde proponen un modelo de DQN para coordinar múltiples UAV para la monitorización de inundaciones y, a la vez, minimizar los costes de los daños. En este trabajo se elige muy bien el tipo de UAV para la vigilancia de inundaciones. Los UAV de ala fija son la solución más eficiente para los desplazamientos de larga distancia debido a su mayor velocidad. Como tienen menos capacidad de vuelo estacionario que otras configuraciones, los UAV de ala fija requieren trayectorias de vuelo suaves, sin cambios bruscos. Si se aplican en escenarios reales, las trayectorias calculadas deberían disponer de mecanismos para suavizar las curvas debido a posibles cambios bruscos en caso de obstáculos.

Otro algoritmo muy utilizado dentro del RL en el dominio es SARSA (Rummery y Niranjan 1994, Sutton y Barto 2018). Si bien es de conocida popularidad y conocidas bondades, tiene menos representación que *Q-Learning*. Además, con el algoritmo SARSA se encontraron aproximaciones menos recientes. Sin embargo, muestran resultados satisfactorios en diferentes casos. Luo et al. (2018) probaron su algoritmo *Deep-SARSA* en entornos dinámicos, donde los obstáculos pueden cambiar. Dicho trabajo ofrece un método eficiente ante entornos dinámicos, lo que demuestra su capacidad en entornos cambiantes y, asimismo, refuerza su utilidad en el mundo real. Sin embargo, el modelo

Figura III.3 UGV cortacésped doméstico. Los UGV se caracterizan por estar limitados a la topología del terreno, por lo que tienen una reducida libertad de movimiento pero con gran capacidad de adaptabilidad a las necesidades de la operación (Gage 1995).

requiere una fase de preentrenamiento, lo que puede limitar su aplicación en entornos nuevos debido al tiempo necesario para el preentrenamiento y a que las condiciones del entorno pueden cambiar cuando se valide el sistema tras el entrenamiento. Speck y Bucci (2018) combinan el aprendizaje centrado en el objeto con el algoritmo SARSA para mejorar el propio algoritmo. Este trabajo presenta un enfoque descentralizado muy eficiente en términos de generalización. La capacidad de generalización puede verse limitada cuando se trata de UAV de ala fija por las mismas razones que los trabajos citados anteriormente. Así, la configuración del UAV limita el rango de aplicación del sistema a los casos en los que es óptimo utilizar UAV de ala fija. El artículo de Zhao et al. (2019) muestra un nuevo método para la coordinación de enjambres de UAV en redes en malla. Estas redes son muy importantes en zonas de catástrofe para mantener las comunicaciones (Figura III.4). Asimismo, su enfoque contempla las limitaciones de las comunicaciones en estos casos. A pesar de contemplar dichas limitaciones, las redes en malla no siempre pueden desplegarse si el entorno es accidentado o de muy difícil acceso. Por ello, hay que considerar la posibilidad de limitar el número de rutas de vuelo necesarias para que sea lo más viable posible.

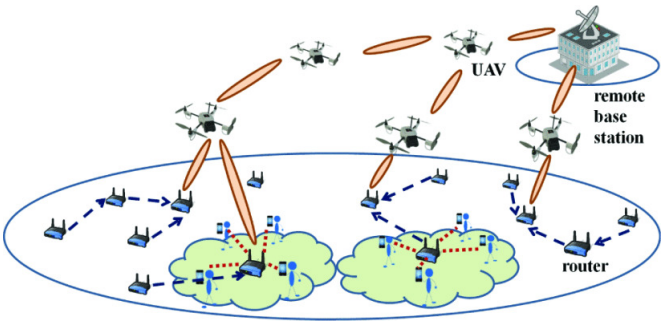


Figura III.4 Diagrama del uso de enjambres de UAV en redes de malla (Zhao et al. 2019). En estas redes, los rúteres forman nodos de una malla con el fin de garantizar la máxima cobertura.

3.1.2 Computación Evolutiva

Otra de las grandes familias de técnicas empleadas en el dominio de los enjambres de UAV es la Computación Evolutiva o *Evolutionary Computation* (EC) (Holland 1992). Por ejemplo, Olson et al. (2020) diseñaron GA para sistemas multi-UAV. En su caso, buscan crear mapas 3D de regiones donde las telecomunicaciones están comprometidas utilizando múltiples UAV. Para ello simplifican el área de vuelo a un mapa 2D. Una vez creado, su sistema busca rutas que maximicen la cobertura y permitan reducir el tiempo de vuelo. El uso del tiempo de vuelo en EC también es utilizado por otros autores, como Huang et al. (2020). En su trabajo tienen en cuenta el tiempo que tarda cada UAV en encontrar un objetivo. Un gran punto a destacar de su trabajo es que es uno de los pocos que tienen en cuenta los atributos de los propios UAV. Otros autores tienen en cuenta el tiempo de vuelo de todo el enjambre en función de la tarea a realizar pero no su arquitectura. Como en el caso de Ye et al. (2020) donde buscan minimizar el tiempo total de vuelo del enjambre. Así, puede ser más eficiente en términos globales minimizar el tiempo de un UAV aunque no se minimice el tiempo de otro. El tiempo se calcula utilizando el modelo de Dubins (Dubins 1957). Este modelo suele referirse a la curva más corta que conecta dos puntos en el plano euclidiano bidimensional. Puede no ser la forma más óptima de obtener trayectorias y tiempos de UAV porque sólo considera curvas.

En lo trabajos de Ramirez-Atencia, Bello-Orgaz, R-Moreno y Camacho (2017) y Ramirez-Atencia, R-Moreno y Camacho (2017) utilizan variaciones del Algoritmo Genético Multiobjetivo para la planificación de misiones con múltiples UAV. En su trabajo realizan una evaluación exhaustiva de su sistema y muestran la evolución de los resultados a medida que aumenta la complejidad de los experimentos. En ambos trabajos falta detalle en la descripción de los conjuntos de datos que utilizan, por lo que no queda claro si estos cambios en la complejidad se interpretan correctamente. Cekmez et al. (2016) encuentran puntos de control en el terreno utilizando agrupaciones por *K-means* (Ahmed et al. 2020) para, luego, calcular las rutas de vuelo necesarias. A continuación, un algoritmo genético paralelo resuelve el problema de planificación de trayectorias multi-UAV de cada subconjunto de puntos de control. La ventaja de este algoritmo genético es su implementación en CUDA (Garland et al. 2008), que permite una experimentación más rápida. El uso de la agrupación por *K-means* puede ser limitante para la partición de áreas. Se sabe que muchos algoritmos de agrupación, como *K-means* o *K-medians*, dependen en gran medida de los parámetros de inicialización (Moshkovitz et al. 2020). Por lo tanto, esta partición debe probarse en una gran cantidad de entornos diferentes hasta conseguir parámetros satisfactorios.

Cimino et al. (2016) emplean la Evolución Diferencial para enjambres de UAV para detectar objetivos de forma colaborativa. La principal diferencia entre la Evolución

Diferencial y otros algoritmos de EC, es que depende en mayor medida del operador de mutación que del operador de cruce (Storn y Price 1997). Así, un descendiente puede ser la mutación exclusiva de un padre. Al depender menos de un tipo de operador que del otro, es más difícil encontrar nuevos individuos en la población. Por lo tanto, puede ser más costoso encontrar el camino óptimo. En el trabajo de Zhou et al. (2020) se hace que múltiples UAV vuelen sobre una porción de terreno en presencia de objetivos dinámicos. Para ello, utilizan el Algoritmo Genético Inmune. El inconveniente de este método es la necesidad de suavizar la trayectoria, lo que requiere de mayor coste computacional y de tiempo de espera.

Todas estas técnicas pueden ir combinadas con otras técnicas. Cabe destacar que, mayoritariamente, estas técnicas se encuentran combinadas con otras técnicas para, así, poder obtener mayores capacidades. Por ejemplo, Sathyan et al. (2016) combinan un Algoritmo Genético (Forrest 1996) con Lógica Difusa (Zadeh 2023) para mejorar la precisión durante la planificación de trayectorias. En esta publicación se aborda el problema desde un punto de vista muy interesante, ya que interpretan los objetivos en el mapa como polígonos (Figura III.5), por lo que su tamaño y forma afecta a las rutas de vuelo. Así, resuelven rápidamente el problema de que cada UAV vuelva al punto de partida al final de la operación como si fuera parte de la propia trayectoria. Su principal inconveniente es que no tienen en cuenta el consumo de combustible ni las posibles colisiones. De este modo, las trayectorias pueden tener grandes longitudes o cambios bruscos de dirección que la autonomía no puede soportar. Además, dichas trayectorias pueden estar tan próximas entre sí que los UAV pueden colisionar.

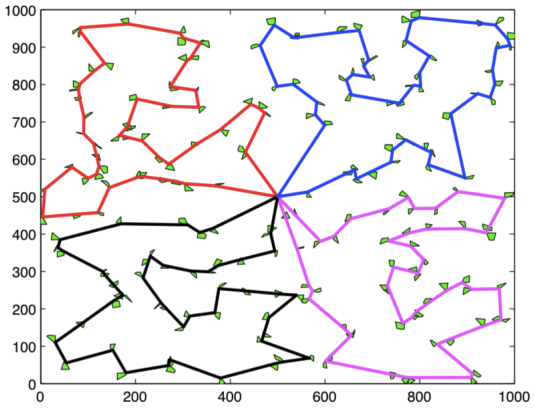


Figura III.5 Ejemplo de mapa con los objetivos que dirigen la ruta marcados como polígonos (Sathyan et al. 2016). Su posición y tamaño ayudan a dirigir el cálculo de la ruta.

.

En el 2021, Kusyk et al. (2021) combinan RL con técnicas de Teoría de Juegos (Owen 2013). Ese mismo año, Pan et al. (2021) combinaron técnicas de EC con técnicas de Aprendizaje Profundo o, como suele conocerse, *Deep Learning* (LeCun et al. 2015)

para calcular rutas óptimas para múltiples UAV para capturar datos de múltiples nodos. Gracias a esta combinación, mejoran los resultados respecto al uso de técnicas de EC puras en caso de tener numerosos nodos.

3.1.2.1 Inteligencia de Enjambre

Dentro del paradigma de la EC se encuentran las técnicas de Inteligencia de Enjambre o *Swarm Intelligence* (SI) (Yang 2015). Estas técnicas se consideran un campo dentro de la EC por centrarse, principalmente, en la coordinación y la auto-organización de grupos de individuos bioinspirados (Dorigo et al. 2007). Además, es otro de los grupos de técnicas más empleados en enjambres de UAV. Cekmez et al. (2018) utilizan la Optimización por Colonia de Hormigas para planificar trayectorias óptimas de UAV evitando obstáculos complejos como radares. En su trabajo, implementan una versión paralela del algoritmo para GPU que les permite realizar más iteraciones de dicho algoritmo al mismo tiempo. Esto permite acercarse más a la solución óptima. Además, consideran una velocidad de vuelo constante, por lo que los giros a realizar para cada UAV pueden no ser las más eficientes ya que no se puede reducir dicha velocidad. Perez-Carabaza et al. (2018) también utilizan esta técnica para planificar trayectorias de vuelo de forma que múltiples UAV puedan encontrar objetivos en entornos desconocidos en el mínimo tiempo posible. El uso de su heurística es muy preciso, debido a que la velocidad de cálculo que consigue permite mejores ajustes sin perjudicar mucho el tiempo necesario para llegar a soluciones óptimas. Además, una heurística correctamente definida puede reducir el coste computacional. Como afirman los autores, las trayectorias deben suavizarse o se limitaría a un cierto número de tipos de UAV que permitan cambios de dirección más abruptos. Otra aproximación a esta técnica es su uso en la planificación cooperativa de misiones de búsqueda-ataque para múltiples UAV (Zhen et al. 2018). En estos casos, suele tratarse de encontrar un objetivo y acercarse para realizar un ataque a dicho objetivo. En particular, suelen enfrentarse a más cambios de trayectoria porque los objetivos cambian con frecuencia su dirección. En este trabajo, también consideran velocidades de vuelo constantes. Si son velocidades altas, las curvas trazadas pueden no ser factibles.

3.2 Otros Vehículos

Todas las técnicas anteriormente descritas no son exclusivas de los enjambres de vehículos aéreos. Otros muchos tipos de vehículos autónomos se benefician de ellos, aunque en menor medida debido a su menor popularidad.

3.2.1 Enjambres de USV

En entornos acuáticos se pueden encontrar los conocidos como vehículos de superficie no tripulados o *Unmanned Surface Vehicle* (USV) (Figura III.6). Estos vehículos presentan un menor abanico de movimientos comparados con los UAV debido a que se su ámbito se limite a la superficie de masas de agua que tienen corrientes que arrastran los vehículos, especialmente aquellas que surgen de manera repentina debido a la climatología (Ding et al. 2018). Por lo tanto, no son necesarios movimiento que supongan controlar la altura pero sí tener control de la propia deriva de la nave por el factor de las corrientes. Sin embargo, han demostrado sus capacidades en diferentes ámbitos (Liu et al. 2016).

Figura III.6 Ejemplo se USV. Estos pequeños vehículos, similares a barcos, presentan un comportamiento limitado a la superficie de entornos acuáticos, lo que limita sus movimientos.

Dentro del RL existen trabajos como los de Xie et al. (2021) donde proponen el uso de estas técnicas para el control de enjambres de USV. Un restricción importante en su trabajo es la necesidad de mantener la formación en "V" inversa. Esta restricción implica una limitación en movimientos, pues se podría perder la formación. Además, dentro de dicha formación hay un líder, que condiciona los movimientos de los demás. Esta pérdida de libertad puede complicar la obtención de resultados satisfactorios. A estas mismas restricciones de formación se enfrentaron Monaco et al. (2021) donde emplean técnicas de SI para poder controlar lo que ellos llaman "bandadas" de USV.

Siguiendo con EC, Xia et al. (2021) emplean Algoritmos Genéticos para el control de enjambres de USV. Un punto a tener muy en cuenta es que consideran la morfología del casco del vehículo, por lo que su modelo sabe interpretar el ángulo y su efecto en la toma de decisiones.

3.2.2 Enjambres de UUV

Si se profundiza más en los entornos acuáticos es necesario el uso de vehículos submarinos no tripulados o *Unmanned Underwater Vehicle* (UUV) (Figura III.7). Si bien se añade un nivel más de complegidad al tener que controlar la profundidad, entre las

mayores limitaciones de estos vehículos se encuentran las comunicaciones y que dependen de las presiones que tienen que soportar bajo el agua para conocer sus posiciones (Yildiz et al. 2009). Existen gran cantidad de aplicaciones en este tipo de vehículos submarinos (Sands 2020).

Figura III.7 Ejemplo de UUV sobre la superficie del agua. Estos vehículos submarinos tienen un gran abanico de formas y son muy empleados en diferentes ámbitos gracias a que permiten incorporar diferentes sensores (Wang et al. 2020).

En los enjambres de UUV también se encuentran aplicaciones con técnicas de RL y de SI. Por ejemplo, (Wu et al. 2021) combinan estas técnicas para tareas de búsqueda y rescate. En su trabajo, realizan dichas operaciones autónomas en diferentes fases, por lo que dividen el problema en diferentes subproblemas.

3.2.3 Enjambres de UGV

Cambiando al ámbito terrestre se encuentran los anteriormente mencionados vehículos terrestres no tripulados o *Unmanned Ground Vehicle* (UGV) (Figura III.3). Estos vehículos, al igual que los USV realizan menor número de movimientos ya que se le limitan a las superficies, en este caso la terrestre. Si bien no tienen que preocuparse de la altura o la profundidad, sí que tienen una gran dependencia de cómo es el relevo por el que se mueven, siendo las escaleras o las grandes pendientes sus mayores dificultades (Nguyen-Huu et al. 2009). Normalmente, y como se citaron previamente, se suele combinar su uso con UAV, ya que aportan mucha información cuando se emplean juntos. Por ejemplo, (Nguyen et al. 2019) emplean técnicas de RL donde un UAV asiste en la toma de decisiones para guiar enjambres de UGV.

Capítulo IV

Metodología

Este capítulo describe todos los materiales, métodos y metodologías necesarios para la realización de la investigación. En primer lugar, se describe el problema a resolver. Posteriormente, se describen todos todos los aspectos necesarios para su resolución.

4.1 Problemas de Planificación de Rutas

Se conocen como problemas de planificación de rutas (*Path Planning* en inglés) a todos en los que se procura encontrar rutas óptimas para vehículos o robots con el fin de conseguir diferentes objetivos (Zhuang et al. 2012). Este cálculo puede hacerse de forma global o de forma local, según la información que necesita el vehículo o robot (Marin-Plaza et al. 2018). En general, los problemas de planificación de rutas (ec. IV.1) pueden describirse del siguiente modo (Contreras-Cruz et al. 2015): dado un robot o vehículo en un entorno de trabajo u operación, se debe buscar una función (f) que obtenga la secuencia de acciones que formen una ruta conocida como el conjunto de acciones A) desde un estado inicial i hasta un estado objetivo f, ambos pertenecientes al conjunto de posibles estados S, de acuerdo con un determinado criterio de rendimiento (conocida como la función C). Así, se puede entender que hay tres aspectos importantes a la hora de afrontar los problemas de planificación de rutas (Zhang et al. 2018): el modelado del entorno, el método de obtención de la ruta y, finalmente, el criterio de optimización.

$$f : C \times S \times S \longrightarrow A \tag{IV.1}$$

Principalmente, su modo de operación de las técnicas de planificación de rutas se compone de dos pasos (Giesbrecht 2004): el primero, obtener el modelo del entorno; y, el segundo, calcular las rutas siguiendo un criterio de optimización.

En el primer paso, es importante recopilar toda la información disponible en un entorno de manera efectiva y suficiente para poder obtener un modelo interno de manera

precisa y calcular rutas teniendo en cuenta las capacidades del robot o del vehículo. Si el entorno es real, los vehículos o robots deben contar con diferentes tipos de sensores y éstos deben ser correctamente calibrados (Borenstein et al. 1997). Sin embargo, si el entorno es artificial, deben tenerse en cuenta todos los aspectos físicos subyacentes que puedan afectar a las rutas.

En el segundo paso, es crucial utilizar una técnicas de búsqueda lo más adecuadas posible para encontrar la mejor ruta en el espacio determinado previamente. Para ello, debe tenerse en cuenta toda la información del entorno, tanto la relativa al vehículo o robot, como la del entorno de vuelo.

En esta Tesis Doctoral, es necesario establecer y modelar entornos de vuelo adecuados para, luego, poder desarrollar las técnicas adecuadas para poder controlar las rutas independientes de los UAV de un enjambre en entornos simulados. Todo esto, teniendo en cuenta los criterios de optimización adecuados y convenientes.

4.1.1 Modelado del Entorno de Vuelo

El entorno de vuelo es el espacio en el que vuelan los UAV del enjambre. Dicho entorno determinará el espacio de búsqueda de soluciones. Es decir, a mayor entorno, mayor es la búsqueda para encontrar rutas óptimas.

Se puede utilizar un entorno continuo o un entorno discreto para describir el mundo para la planificación de rutas (Giesbrecht 2004). En un entorno continuo, el mundo se modela como un conjunto infinito de estados posibles representando las posibles combinaciones de los infinitos puntos. En consecuencia, la planificación de rutas implica la identificación de una ruta continua que se mueva a través de este espacio. La planificación suele ser más difícil en un espacio de estados continuo ya que es necesario tener un mayor nivel de precisión y las posibilidades de movimiento son infinitas, lo que hace que el espacio de búsqueda sea infinito. Esta gran cantidad de combinaciones implica que haya un gran número de mínimos y máximos locales que dificultan la búsqueda.

Sin embargo, cuando se emplea un entorno discreto, el espacio se divide en un número finito de estados por los que un vehículo puede desplazarse y describe cualquier número de atributos de estado, como la ubicación y la pose de un vehículo o la elevación de esa celda en el entorno. Cada uno de estos estados discretos tiene una función de transición de estado que identifica los demás estados directamente accesibles (s') desde el estado actual (s) para construir un plan. Luego, un algoritmo recorre el espacio de estados en busca de un conjunto de estados cuya secuencia discreta determine la ruta. Dicha ruta será optimizada desde el estado inicial hasta el estado final deseado en base a toda la información necesaria y a los criterios de optimización. Por lo tanto, los entornos discretos son muy utilizados debido a se simplifica la obtención de una buena solución

al aplicarle una discretización del entorno de operación. Hay que resaltar que dicha discretización debe ser capaz de representar cualquier entorno real o supondría una gran pérdida de información que puede afectar a los resultados finales.

Existen tres tendencias principales a la hora de descomponer un mapa en un entorno discreto:

- **Mapas de celdas:** es la técnica más utilizada. El mapa se divide en un conjunto de zonas representativas denominadas celdas (Fernández-Blanco 2010, Debnath et al. 2021). En cada una de las celdas es necesario establecer información adicional, como la presencia de obstáculos, para mejorar la precisión de la ruta (Figura IV.1). Es importante tener equilibrio en el número de celdas, Demasiadas, inducirán a una complejidad innecesaria debido al excesivo incremento de estados posibles (Bellman 1966); muy pocas, harán perder precisión en el mapa y supondrían falta de precisión en la resolución de los problemas.

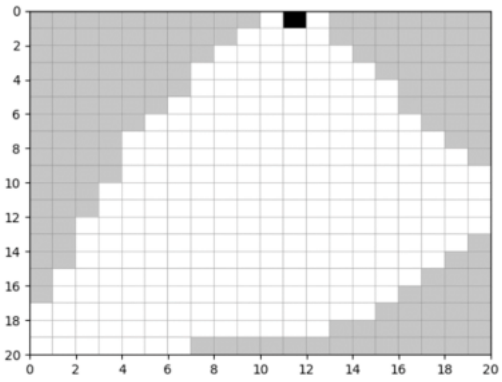


Figura IV.1 Ejemplo de mapa dividido en celdas (Puente-Castro et al. 2022*b*). En gris se representan las celdas sobre las que no se puede volar. En blanco se presentan las celdas sobre las que se puede volar. En negro se presenta el punto de partida del vehículo.

- **Mapas de carreteras:** este tipo de mapa intenta describir el mundo en términos de cómo llegar de un lugar de origen a un lugar de destino, teniendo en cuenta el coste de desplazarse entre ellos. Su elaboración es mucho más difícil y lenta que la de los mapas anteriores ya que supone crear un esqueleto de un mapa en forma de grafo (Debnath et al. 2021). Su descomposición se basa, principalmente, en dos alternativas: grafos de visibilidad (Figura IV.2) (O'Sullivan y Turner 2001) y mapas de Voronoi (Figura IV.3) (Gold et al. 1996). Como ventaja, son más rápidos de procesar una vez creados puesto que se conoce de antemano el coste de pasar por las diferentes regiones del mapa.
- **Campos de potenciales o gravedades:** cada vehículo se representa como un objeto bajo la influencia de valores numéricos que forman un campo de potenciales creado por objetivos y obstáculos del mundo real (Figure IV.4). Dichos valores numéricos, conocidos como potenciales o gravedades, indican cómo puede llegar

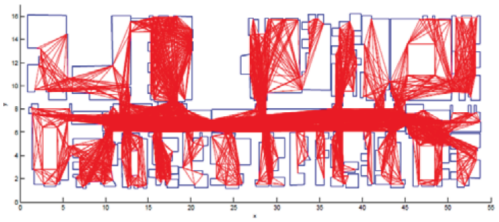


Figura IV.2 Ejemplo de grafo de visibilidad (Kaluđer et al. 2011). En rojo se ve el grafo que representa las partes del mapa que puede percibir por sus sensores el vehículo desde distintos puntos del mapa y teniendo en cuenta las limitaciones de los sensores. En azul se ve el mapa real de referencia.

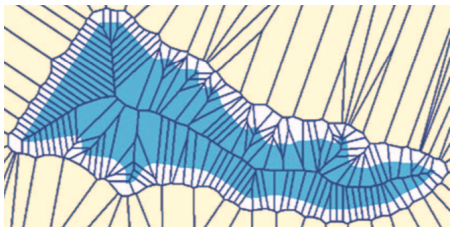


Figura IV.3 Ejemplo de mapa de Voronoi (Bhattacharya y Gavrilova 2008). La región se divide en polígonos que cumplan una condición dada. Luego el mapa de carretera se construye en base a la intersección de las líneas que dividen la región. Estas líneas son las líneas de visión o de percepción resultantes si el vehículo estuviese en ese punto.

a afectar el posible obstáculo en ese punto del mapa. Los potenciales influyen en el vehículo como si fuera una magnitud física (Hwang et al. 1992). Este método se ha utilizado sobre todo para evitar obstáculos locales en vehículos y robots móviles, pero también puede contribuir a una planificación eficaz de la trayectoria. Es importante calcular bien dichos potenciales para reducir posibles errores en el cálculo de las rutas.

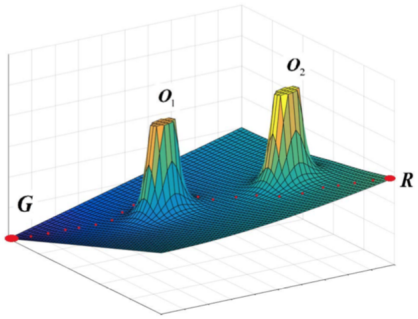


Figura IV.4 Ejemplo de mapa por campos de potenciales (Sun et al. 2020). Los obstáculos son representados como montículos que cuantifican la intensidad y la amplitud de su potencial. Estos montículos se conocen como campos de potenciales.

Para esta Tesis Doctoral se ha elegido emplear mapas de celda al ser la técnica más empleadas. Además, se tienen representaciones simplificadas que reducen el tiempo de obtención para el cálculo de las rutas. Incluso, simplifica el cálculo de la evaluación de

las rutas obtenidas y facilita la incorporación de obstáculos para evaluar los algoritmos empleados.

Hoy en día, el vuelo sobre zonas del mundo real puede suponer un coste elevado o problemas de disponibilidad. Además de esto, en muchos países falta legislación para los vuelos experimentales, es muy restrictiva o, incluso, puede ser muy compleja. Este hecho puede, incluso, que acabe dilatando la operación (Alrubaye y Miyauchi 2023). Ante esta situación, la mayoría de los autores optan por utilizar entornos de vuelo artificiales, sean estos totalmente artificiales o simulaciones del mundo real donde pueden sortear las restricciones impuestas por la ley (Puente-Castro et al. 2022*a*). Éstas pueden ser aplicaciones comerciales, como el conocido AirSim (Shah et al. 2018), o pueden ser propias (Walter et al. 2006). Así, las pruebas de los algoritmos puede realizarse en entornos virtuales que imitan las condiciones físicas que limitan el problema a resolver. De este modo, solo es necesario el equipo informático pero no es necesario tener UAV y áreas de vuelo delimitadas.

4.1.2 Cálculo de Rutas de Vuelo

Muchas son las técnicas de cálculo de rutas de vuelo disponibles (Aggarwal y Kumar 2020). Si bien son de diferente naturaleza, el objetivo es el mismo: obtener la mejor ruta posible teniendo en cuenta las limitaciones impuestas por el vehículo y por el entorno. Para ello, han de cumplir lo siguientes puntos (Giesbrecht 2004):

- **Longitud total de la ruta:** cuanto más corta sea la ruta, más óptima será. Si una ruta es más corta que otra y conecta los mismos puntos, significa que tiene menos vueltas y menos curvas, lo que la hace más eficiente energéticamente, ya que se reducen los movimientos.
- **Evasión de obstáculos:** el sistema debe ser capaz de permitir a los UAV evitar cualquier obstáculo que aparezca durante los vuelos. Ya sean dinámicos o estáticos.
- **Zonas de acceso restringido:** el sistema debe ser capaz de controlar que los UAV no sobrevuelen zonas prohibidas para, así, no exponer al usuario a situaciones comprometidas de riesgo legal.
- **Tolerancia a fallos:** es un punto crítico en enjambres ya que el sistema debe ser capaz de reorganizar las trayectorias de todos los UAV en caso de que uno falle.
- **Exhaustividad:** es necesario que el sistema pueda satisfacer un criterio de exhaustividad de acuerdo con la tarea asignada. Es decir, que sean capaces de cumplir con el objetivo en su totalidad.
- **Configuración de los UAV:** el sistema debe ser capaz de adaptar la trayectoria a

los límites de las capacidades de cada UAV.

- **Otros factores externos:** otro reto es poder tener en cuenta condiciones adicionales que influyen en el cálculo de la trayectoria óptima. Factores como el viento, los pájaros, la lluvia o las tormentas solares son obstáculos en las trayectorias.

Como ha sido mencionado anteriormente, las técnicas de cálculo de rutas de vuelo están sujetas a gran cantidad de variables que van desde la información del mapa a las necesidades del objetivo a cumplir (Zheng et al. 2005). Por ello, las técnicas empleadas deben poder hacer uso de toda esta información con el fin de alcanzar el objetivo final.

4.1.3 Inteligencia de Enjambre

Cuando el cálculo de rutas se aplica en grupos de vehículos o robots se lo conoce como Inteligencia de Enjambre, o *Swarm Intelligence* (SI). El concepto fue originalmente introducido por (Beni y Wang 1993) aplicado a lo que se conocían como sistemas robóticos celulares, estos agentes del enjambre aprenden y actúan de forma autónoma basándose en un entorno. De este modo, cada agente es capaz de abstraer conocimientos de alto nivel sin estar explícitamente programado. A menudo el conocimiento es difícil de representar en su totalidad debido a su complejidad o a su amplia gama de casos debido a que es la información que percibe cada agente.

El objetivo principal es que los comportamientos deseados en el enjambre no se codifican explícitamente con una estructura jerárquica donde uno o varios integrantes del grupo ejercen mando o control sobre los demás. En cambio, la interacción es un comportamiento emergente de la interacción de los agentes entre ellos y con su entorno (Beni 2004). Este tipo de comportamiento para la resolución de problemas se inspira en el comportamiento colectivo de grupos sociales biológicos como las colonias de insectos y otras sociedades animales (Sharkey y Sharkey 2006). En general, los agentes del grupo utilizan reglas sencillas para realizar sus acciones y, mediante las interacciones individuales de todos los agentes del grupo, el enjambre alcanza sus objetivos (Liu y Passino 2000).

Cabe destacar que todos los agentes de un enjambre abstraen conocimiento a partir de la información obtenida de su entorno. La gran ventaja de las técnicas de SI es que los agentes pueden ser diferentes entre sí dentro de un grupo. Por tanto, el conocimiento abstraído por cada agente puede obtenerse de forma diferente, lo que aumenta la dificultad a la hora de calcular las rutas. Por ello, las técnicas de cálculo de rutas de vuelo en enjambres tienen que considerar los mismos aspectos que para el cálculo de rutas para un único vehículo y, además, la relación entre datos de diferente índole y las inteerferencias entre otros miembros del grupo (Zheng et al. 2005). Todo ello, sin la necesidad de ser programado explícitamente.

4.1.4 Técnicas de Inteligencia Artificial

Las técnicas de Inteligencia Artificial (IA) (Russell y Norvig 2021) son aquellas técnicas donde se pretende poder exhibir un comportamiento inteligente sobre una tarea de manera similar a cómo lo haría un ser humano. Es por ello que se tratan de técnicas muy populares dentro de SI.

La IA engloba un conjunto de técnicas que se pueden englobar en diferentes ramas. Todas ellas son útiles ante diferentes necesidades (Wang et al. 2021), pero capaces de generalizar comportamientos sin necesidad de ser explícitamente indicado. De este modo, se reduce el riesgo de errores y sesgo a la hora de modelar comportamientos en diferentes situaciones (Sethu et al. 2022).

4.1.4.1 Técnicas de Aprendizaje por Refuerzo

Dentro del mundo de la IA existe un conjunto de técnicas conocido como técnicas de Aprendizaje por Refuerzo o *Reinforcement Learning* (RL) (Sutton y Barto 2018) donde un agente (que puede ser un vehículo o un robot) aprende a moverse e interactuar en un entorno en base a un sistema de recompensa o refuerzo. Es decir, el agente realiza acciones en un entorno y aprende cuáles son las mejores de manera autónoma. Es por ello que se considera que estas técnicas constan de dos tipos de enfoques de los que hace uso en diferentes medidas para alcanzar su objetivo: enfoque de exploración, donde el agente aprende a interactuar con el entorno probando nuevas acciones que le permiten descubrir nuevas posibilidades; y enfoque de explotación, donde el agente usa su conocimiento para interactuar con el entorno pero que no le permiten descubrir nuevas posibilidades. Por lo que es importante saber determinar cuánto debe explorar y explotar su conocimiento el agente con el fin de que aprenda a moverse de manera óptima (Yogeswaran y Ponnambalam 2012). Así, si el agente pasa la mayor parte del tiempo explorando puede ser que converja en un comportamiento errático o que no pueda converger en un comportamiento. Sin embargo, si pasa la mayor parte explotando su conocimiento puede hacer que no aprenda posibles comportamientos más óptimos.

A pesar de que todas las técnicas de RL siguen la estructura común anteriormente descrita, existen diferentes variaciones. La principal diferencia es la estrategia de aprendizaje que adopta cada técnica. Es decir, la manera que tienen de decidir cuándo explorar y cuándo explotar el conocimiento. Existen varios tipos de estas estrategias las cuales siguen políticas diferentes que les permiten enfrentarse a problemas distintos. Además, las propias acciones tomadas por el agente pueden modificar el entorno, por lo que las estrategias elegidas deben ser capaces de adaptarse a estos cambios (Van Hasselt y Wiering 2007).

4.1.4.2 Algoritmo Q-Learning

Una de las técnicas de RL más empleadas en el cálculo de rutas es el *Q-Learning* (Watkins y Dayan 1992). Estas técnicas sigue una estrategia libre de modelos (*model free strategy*) (Gläscher et al. 2010), por lo que adquiere su conocimiento siguiendo una política de por ensayo-error en el que solo se mira el efecto de las acciones tomadas en el entorno. Se basa en el aprendizaje *off-policy* (Hausknecht et al. 2016). Es decir, se sigue una política para explorar y, así, aprende y mejora una política óptima diferente que permite a los agentes explotar su experiencia para determinar qué acciones tomar (Sutton et al. 1998). La función óptima de aprendizaje clásico de *Q-Learning* para calcular los valores de la tabla Q, o *Q-table*, (Q(s, a)), los cuales determinan cuál es el mejor movimiento, se basa en la Ecuación de Bellman (Ec. IV.2) (Clifton y Laber 2020, Bellman 1966). Dicha tabla contiene los posibles valores de las recompensas futuras (Q-valores o *Q-values*) de tomar cada posible acción (a) en cada estado (s). Generalmente, elegir la acción del valor más alto en cada estado debería llevar a la mejor solución posible.

$$Q(s,a) \leftarrow r + \gamma \times \arg\max_{a'}(Q(s^{'},a^{'})) \tag{IV.2}$$

La Ecuación de Bellman consta de una serie de elementos fundamentales. Si S es el conjunto de todos los posibles estados, donde s es el estado actual y $s^{'}$ el siguiente estado a éste, y el conjunto A el conjunto de todas las posibles acciones que el agente puede tomar, donde a es una acción del determinada del conjunto y $a^{'}$ es cada una de las acciones que puede tomar el agente para un estado dado, se define como $Q(s,a)$ la función que calcula el valor Q o *Q-value* para el estado actual s y para una acción dada a. La recompensa de acción realizada en ese estado es la variable r es calculada mediante la función de recompensa $R(s,a)$. γ es el factor de descuento y decide cuánta importancia debe darse a la recompensa futura acumulada en comparación con la recompensa instantánea. La función $\arg\max_{a'}(Q(s^{'},a^{'}))$ determina cuál es la acción con el valor Q máximo calculado de cada par $(s^{'},a^{'})$, representado como $Q(s^{'},a^{'})$. Dicho par $(s^{'},a^{'})$ es el par donde $s^{'}$ es el siguiente estado al actual y las posibles acciones $(a^{'}$ que se puede tomar en él. El siguiente estado $s^{'}$ es calculado mediante la función de transición $T(s,a)$ que devuelve el estado resultante de realizar la acción seleccionada a en el estado actual s.

Si bien emplear la Ecuación de Bellman ha demostrado sus bondades, tiene limitaciones (Fan et al. 2020). Una de ellas es el tamaño de la tabla Q, que tiene un gran crecimiento si el número de estados es muy elevado y es necesario tener los valores almacenados. Esto es muy limitante si no se conoce el número de posibles estados con anterioridad. Para alcanzar mejores resultados, el cálculo de los valores de cada acción para cada estado se realizar mediante otras técnicas regresión del campo de la IA, como

pueden ser modelos de Aprendizaje Máquina o *Machine Learning* (Bishop y Nasrabadi 2006) debido a que tienen mejor capacidad de generalización y no es necesario almacenar todos los valores previamente calculados.

Las técnicas más comúnmente empleadas en el Estado de la Cuestión son los modelos de Aprendizaje Profundo conocidos como Redes de Neuronas Artificiales (RR.NN.AA.) (Rosenblatt 1958) debido a su gran capacidad de aplicación y generalización. Esto es lo que se conoce como *Deep Q-Learning* (DQN) (Mnih et al. 2015). Las RR.NN.AA. se caracterizan por estar formados por nodos interconectados entre sí y que están dispuestos en capas (Figura IV.5).

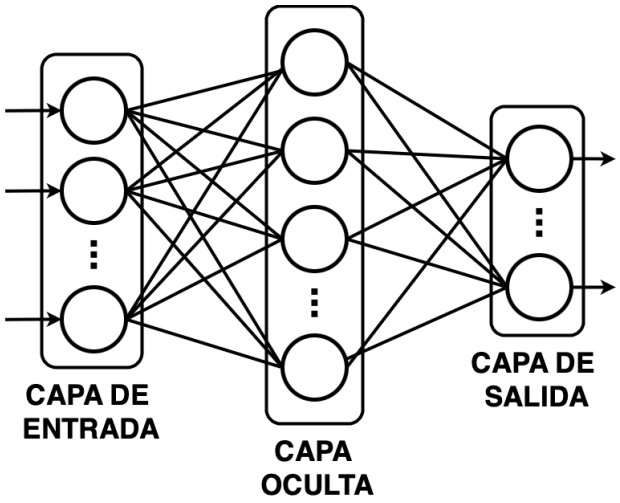


Figura IV.5 Diagrama de las capas clásicas de las RR.NN.AA. donde se puede ver cada una de las capas. Todas están formadas por: una capa de entrada, que es la que recibe los datos de los que hace uso la red; una capa de salida, que es la que devuelve el resultado o decisión finales; y, una o más capas ocultas, que son aquellas donde se extrae el conocimiento y permiten a la red aprender las relaciones complejas que forman los datos.

El funcionamiento de las RR.NN.AA. recae en la interacción entre los nodos que las conforman. Estos nodos son conocidos como neuronas artificiales (McCulloch y Pitts 1943). Estas neuronas realizan, cada una, una operación matemática que sigue misma estructura de la forma $\hat{y} = f(X \cdot W + b)$ pero cuyos valores numéricos varían entre neuronas (Figura IV.6). Dicha ecuación calcula la salida $\hat{y}$ para un vector de entradas X que contiene los datos separados en valores numéricos. Para ello, ajustan los pesos o *weights* (W) de sus conexiones durante el entrenamiento, que indican la importancia de cada conexión en la predicción final. Además, para ajustar mejor la salida, se añade un sesgo o *bias* (b) que desplaza la función resultante en el espacio de valores. Cabe destacar que la fórmula $X \cdot W + b$ se trata de una función lineal simple. Como se dijo anteriormente, los datos presentan un alta complejidad que la red debe aprende. Es por ello que la neurona necesita una función de activación o *activation function* ($f()$) que calcule la salida de la neurona a partir de resultado de la fórmula $X \cdot W + b$ con el fin

de poder obtener resultados que se adapten a la complejidad del dominio en el que se aplican las RR.NN.AA.. Durante el entrenamiento, es necesario que los valores de W y b sean los mejores posibles para que la salida $\hat{y}$ sea lo más similar posible a la del mundo real y, así, reducir el error que la propia red pueda generar.

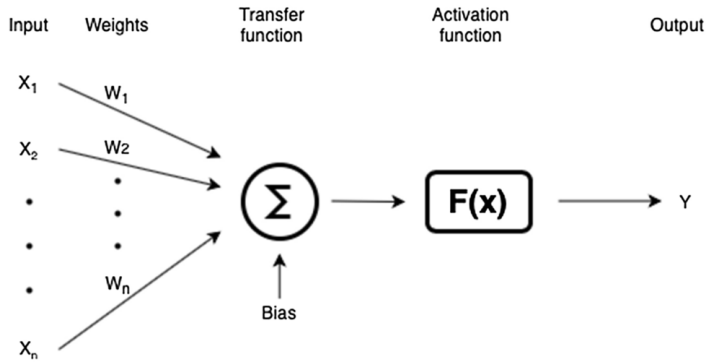
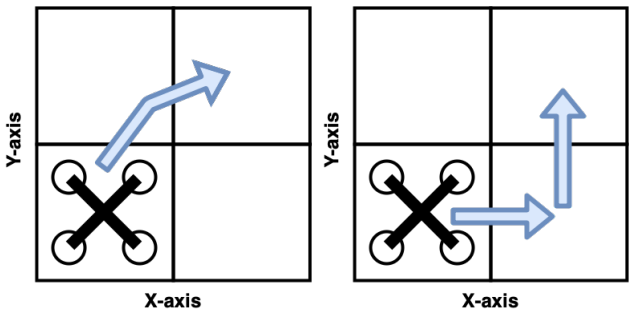


Figura IV.6 Esquema general de una neurona artificial (Puente-Castro et al. 2022*a*). Inicialmente, el vector X que contiene las entradas o *inputs* x_i se multiplica por el vector de pesos o *weights* W que contiene los pesos w_i correspondientes a la conexión de la neurona que contiene cada entrada y se suman con el sesgo o *bias* (b). Posteriormente, este resultado se utiliza como entrada de la función de activación o *activation function* ($f(x)$). La salida de la neurona es el resultado de esta función.

Para definir cualquier acción a dentro del conjunto de acciones A se ha tenido en cuenta la complejidad de movimiento de los UAV y las necesidades de la operación según la representación en celdas del mapa. A pesar de las capacidades de vuelo de estas aeronaves citadas anteriormente, un movimiento en curva implica combinar, en diferentes grados, diferentes tipos de movimientos más sencillos (Susanto et al. 2021), lo que es difícil de computar y, además, pueden aumentar el consumo energético debido a la gran combinación de acciones a tomar y a sus efectos en el aire (Thibbotuwawa et al. 2019). Además, movimientos complejos como curvas suponen que no se garantiza que un UAV sobrevuele una celda en su totalidad, por lo que puede que no perciba toda la información de dicha celda (Figura IV.7).

Siguiendo con la tendencia de muchos autores mencionados en el capítulo dedicado al estudio del Estado de la Cuestión, se ha decidido establecer movimientos básicos simplificados a un vecindario de von Neumann (Toffoli y Margolus 1987) en lugar de uno de Moore (Sharma et al. 2013) (Figura IV.8), similares a los que usan los protocolos de comunicación por radiofrecuencia. Estos protocolos reciben instrucciones de movimientos simples que luego combinan automáticamente en movimientos complejos y los traducen en acciones por parte de los rotores (Koubâa et al. 2019).

Para poder dirigir el aprendizaje del sistema, hay que definir las recompensas para cada tipo de acción. Para forzar que los UAV de un enjambre prioricen desplazarse a zonas no visitadas y descubran regiones nuevas, la recompensa debe ser la mayor de



(a) Movimiento en curva (b) Movimiento en ángulo

Figura IV.7 Comparación de movimientos donde se ve que un movimiento en curva no tiene por qué pasar totalmente sobre una celda, por lo que se pueden perder datos o información de dicha celda (Puente-Castro et al. 2022*b*).

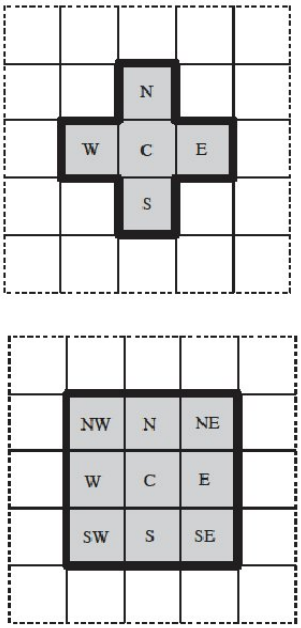


Figura IV.8 Ejemplo comparativo de vecindarios de von Neumann (arriba) y de Moore (abajo) de una celda de radio (Quartieri et al. 2010).

todas. Además, para forzar más este comportamiento, es importante que la recompensa aumente a medida que se dejan menos celdas sin descubrir (Ec. IV.3), ya que buscarán siempre la mayor recompensa posible. Es decir, la recompensa actual de descubrir una celda nueva (R') depende un valor mínimo (R) modulado en base al máximo entre el número de filas (F) y columnas (C) del mapa de celdas dividido entre el número de celdas que quedan por visitar (N), siguiendo una estrategia *Hill-Climbing* (Kimura et al. 1995). Es decir, encontrada una solución, intenta encontrar una mejor ante una nueva perturbación. Además de hacer que cada UAV tenga predilección por descubrir celdas nuevas, hay que hacer que rechace volver a pasar por donde ya lo ha hecho. Por ello, se requiere otra recompensa para las celdas que ya han sido visitadas. De este

modo, el UAV tiene una recompensa en caso de que sea mejor sobrevolar una celda ya visitada para llegar a una no visitada que rodearla (por ejemplo, cuando quedan celdas espurias sin visitar), pero ha de ser menor que la recompensa de descubrir nuevas celdas para evitar que pase mucho tiempo sobrevolando zonas ya conocidas. Finalmente, para evitar que los UAV vuelen a celdas que no pueden visitar, se les da la recompensa más baja.

$$\text{R'} = \text{R} \times (1 + \frac{max(\text{F, C})}{\text{N}}) \tag{IV.3}$$

Como se vio en la Ecuación de Bellman, un estado solo depende del anterior. Esta limitación implica que los UAV no tengan el recuerdo de acciones previas a la anterior si no las han aprendido previamente. Este hecho es limitante ya que recuerdos pasados que no parecían importantes pueden ser de gran ayuda en el futuro. Como se describe en el capítulo destinado a analizar el Estado de la Cuestión, muchos autores vieron que es interesante guardar estas experiencias para cuando puedan ser necesarias. A esta técnica se la conoce como *Memory Replay* y su tamaño es importante (Liu y Zou 2018). Se trata de una técnica en la que el agente aprende de un conjunto de observaciones vividas que son almacenadas denominado memoria. Las observaciones contienen información variada, como las acciones realizadas y su recompensa. Mejora la eficiencia del muestreo al reutilizar repetidamente las experiencias y ayuda a estabilizar el entrenamiento del modelo (Foerster et al. 2017). Es importante que la memoria contenga tantas observaciones recientes como sea posible ya que almacenar experiencias muy viejas puede implicar recordar situaciones que se pueden repetir con muy baja probabilidad, pero tiene un tamaño máximo para optimizar los recursos computacionales. Por esta razón, la memoria sigue un esquema en el que las experiencias más viejas se eliminan para dejar entrar a las más nuevas.

Cada UAV del enjambre puede tener su propia memoria. En dicha memoria almacena las observaciones con las acciones que realiza el propio UAV. En ningún momento se almacenan las acciones de otros UAV. De este modo se puede evitar añadir ruido de otros UAV a la información. El hecho de que una acción no sea correcta para un UAV no implica que sea incorrecta para los demás, ya que pueden estar en posiciones diferentes en el mapa.

Por otra parte, si la memoria es colectiva e igual para todos los UAV del enjambre se adquiere conocimiento más rápido. Por contrapartida, se mezclan experiencias de diferentes aeronaves en diferentes situaciones, dando luugar a un aumendo del ruido en la memoria.

Para el desarrollo de esta Tesis Doctoral se consideran los dos casos: una memoria local para cada UAV, lo que implica tener un modelo (RNA) por UAV; y una memoria

global única para todos los UAV, lo que implica tener un único modelo (RNA) para todos los UAV. De esta manera, se pueden ver los efectos de la memoria en la calidad de los resultados finales.

Para evaluar la calidad de las rutas obtenida es necesario establecer criterios de optimización. Estos criterios permitirán determinar de forma cuantitativa y de manera objetiva la bondad de las rutas de vuelo. Así, se podrá estudiar qué experimentos son mejores y realizar análisis estadísticos sobre estos resultados. Así, para esta Tesis Doctoral se han elegido dos criterios: el tiempo de vuelo y la longitud de las rutas de vuelo.

El tiempo de vuelo es una de las principales limitaciones de los UAV citadas anteriormente. Sus pequeñas baterías implican que se puede almacenar poca energía, lo que limita el tiempo de vuelo. Así, las rutas que se obtienen en menor tiempo pueden ser indicativo de ser más cortas y eficientes. De este modo, si se tarda menos tiempo en simular una ruta que otra implica que es más rápida o tiene menos bucles y, por lo tanto, el consumo energético es menor. Esto se debe a que es muy difícil predecir el consumo energético ya que está sujeto a otras variables externas e impredecibles, como el viento (Thibbotuwawa et al. 2019).

El tiempo de vuelo puede no ser suficiente ya que muchos UAV permanecen estáticos en el aire si éstos pierden la comunicación. Estas paradas implican un incremento en el tiempo de vuelo y no hay manera de predecir si van a ocurrir, por lo que no se pueden filtrar en el cálculo del tiempo. Sin embargo, no se ha realizado un movimiento, por lo que se ha consumido energía y tiempo pero los UAV no han completado la tarea o se han acercado más a su completitud. Por ello, se complementa este criterio con el propio cálculo del número de movimitientos necesarios para realizar cada una de las rutas de vuelo. Así, una ruta que precise menos movimientos puede determinar que sea más corta y rápida para recorrer para un UAV. Ya que los mapas empleados están divididos en celdas, el cálculo de los movimientos se puede hacer en base al número de celdas. Con ello, se obtiene una métrica generalizable a diferentes mapas sin necesidad de conocer el tamaño de las celdas.

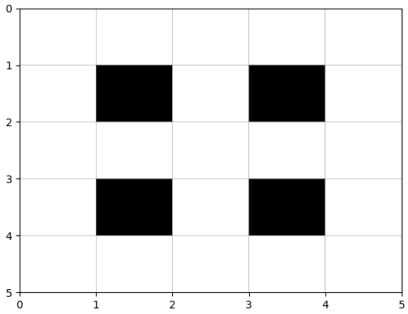
4.2 Diseño Experimental

Para llevar a cabo la parte experimental del proyecto se tiene en cuenta el incremento de la complejidad que suponen los diversos factores subyacentes que pueden ser el tamaño del mapa, el número de obstáculos y el número de UAV del enjambre. Con el fin de probar las capacidades de adaptación del algoritmo desarrollado, es necesario que sea probado en diferente número de UAV. Además, debe considerarse que los entornos de vuelo representen diferentes condiciones y tamaños de manera que pueda verse si el

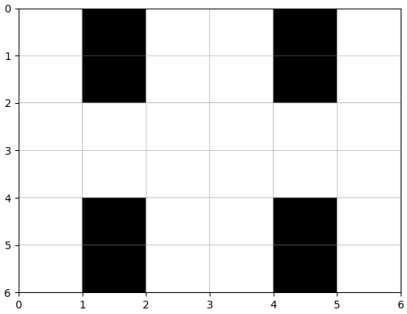
algoritmo es sensible a cambios en los entornos de vuelo.

Con respecto al número de UAV, existe un caso particular. No solo es necesario probar las capacidades del algoritmo ante diferentes tamaños de enjambres, sino que es importante conocer su comportamiento con un único UAV. Este caso es importante porque reflejaría la situación más extrema que permita la operación dentro del enjambre, que sería cuando todos los UAV del enjambre no puede volar menos uno de ellos. Además, serviría para demostrar las bondades del uso de enjambres en lugar del uso individual de los UAV. Es decir, permitiría comprobar si se obtienen mejores resultados con un UAV o con un enjambre de éstos.

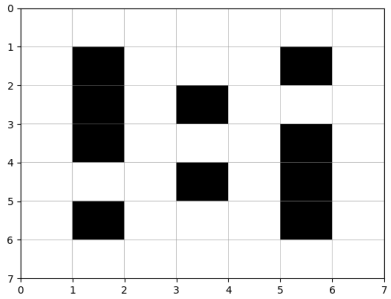
Los mapas o entornos de vuelo también requieren establecer una estrategia para la experimentación. Si se va aumentando la complejidad de los mapas se puede probar la adaptabilidad del algoritmo ante diferentes situaciones. Por ello, es necesario establecer un punto de partida más sencillo donde ese emplean mapas de celdas sin obstáculos pero de diferentes tamaños con el fin de conocer si es capaz de encontrar soluciones en mapas ideales. Luego, se prueba en un conjunto de mapas de diferentes tamaños con obstáculos con diferentes morfologías (Figura IV.9) con el fin de probar las capacidades del sistema en situaciones más complejas y donde se fuercen diferentes situaciones de vuelo. Los entorno de vuelo elegidos tiene menos celdas en comparación con otros estudias. La decisión ha sido influida por el coste de cubrir mapas extensos, ya que, si cada celda requiere una parada para la captura de datos, conlleva un elevado consumo de energía. Dividir el mapa en menos celdas reduce las paradas y conserva energía, aunque cada celda cubre un área mayor. Los datos capturados de regiones de mayor tamaño ofrecen más información contextual y son más adecuados para el procesamiento, a pesar de tener menor detalle o de requerir sensores más avanzados para obtener detalles.



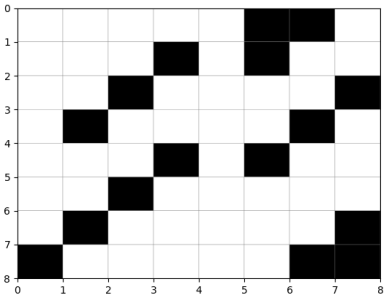
(a) 5×5



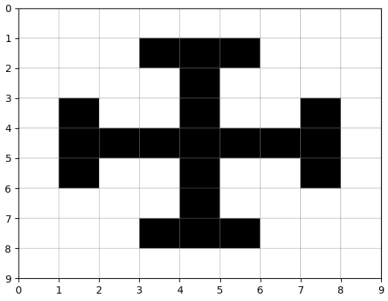
(b) 6×6



(c) 7×7



(d) 8×8



(e) 9×9

Figura IV.9 Mapas utilizados en los entornos de vuelo. Los obstáculos se muestran en negro. En blanco están las celdas que se pueden sobrevolar. Los UAV deben visitar el mayor número posible de celdas blancas (Puente-Castro et al. 2023).

Capítulo V

Resultados

Este capítulo muestra y discute los resultados obtenidos en esta Tesis Doctoral. Estos últimos describen los experimentos realizados, sus resultados y las limitaciones encontradas

5.1 Análisis del Dominio

Para realizar el diseño de los experimentos se realizó un análisis bibliográfico del dominio de las técnicas de IA aplicadas a la los enjambres de UAV para la planificación de rutas, que es detallado es Puente-Castro et al. (2022*a*) (página 69 de esta Tesis Doctoral). El fin de este análisis fue poder dirigir la experimentación y, así, reducir la presencia de errores.

Al ser previo a los experimentos, el análisis se realizó de los cinco años previos a éstos y del año en los que los experimentos dieron comienzo. Así, el periodo de 6 años abarca del año 2016 al año 2021.

Para llevar a cabo dicho análisis se recurrió a la búsqueda de los términos descritos en la Tabla 1 del artículo (página 76 de esta Tesis Doctoral). En ella, se detalla una serie de términos pertenecientes a categorías muy concretas. Mediante la combinación de los términos de estas categorías se consigue crear una serie de consultas. Dichas categorías abarcan diferentes aspectos a tener en cuenta en problemas de planificación de rutas como el tipo de vehículo, el entorno de operación o el algoritmo utilizado. Con las mencionadas consultas, se realizan búsquedas en diferentes fuentes de bibliografía académica y, así, se consigue encontrar la información perteneciente a 39 trabajos académicos.

Dados los 39 trabajos encontrados en el período elegido se ha realizado el análisis bibliométrico en base a un abanico de factores (páginas 79 a 81 de esta Tesis Doctoral). Así, se han podido hacer diferentes clasificaciones de dichos sistemas la literatura de las técnicas de IA aplicadas a la los enjambres de UAV para obtener diferentes indicadores.

Teniendo en cuenta la Figura 2 de ese documento (página 80), se puede ver una tendencia creciente en el número de publicaciones por año. Esta tendencia es indicativa del potencial de estos sistemas debido a su alta aplicabilidad a diferentes problemas. Cabe tener en cuenta que no es así en el año 2021, pero es debido a que solo se han tenido en cuenta las publicaciones hasta el momento de la realización del análisis bibliométrico.

Teniendo en cuenta el tipo de técnica empleada (Figura 3 de la página 80), RL es la técnica más empleada. Incluso, si se observa el número de publicaciones por técnica dependiendo si es del ámbito militar o civil (Figura 4 de la página 80), sigue siendo la más representada. Después de estas técnicas, son las de EC las más utilizadas, pero solo con representación en el ámbito civil. Teniendo en cuenta que las técnicas de SI son una rama de la EC, se podría decir que EC son las técnicas más empleadas en el ámbito civil. Seguramente, esto se deba a su potencial en la investigación académica a pesar de la poca aplicabilidad industrial debido al tiempo necesario para entrenar estos algoritmos.

A pesar del uso militar original de los UAV (Figura 6 de la página 81), más recientemente ha habido un cambio de tendencia en la situación. Cada vez son más las aplicaciones para uso civil, superando a las aplicaciones de uso militar (Figura V.1). De este cambio de tendencia también puede suponerse que el ámbito militar es más reservado a publicar sus descubrimientos por motivos de seguridad, por lo que es probable que se siga innovando en este sector pero no se publique o, incluso, que se publiquen como si fuesen de ámbito civil.

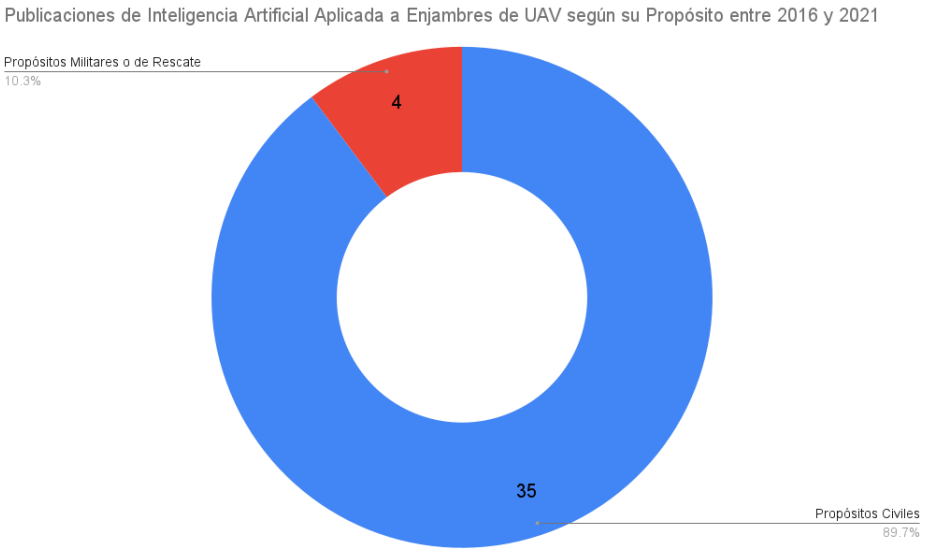


Figura V.1 Gráfico circular de resumen del número de publicaciones de los años 2016 al 2021 donde se aplican las técnicas de IA a la los enjambres de UAV según el propósito para el que se diseñan (Puente-Castro et al. 2022*a*).

Todo los sistemas presentes en el estado de la cuestión precisan de entornos para su evaluación (Figura 7 de la página 81). Idealmente, los sistemas deberían ser probados en entornos reales debido a que, así, se enfrentan a todas las posibles variables del mundo real. Todo esto es muy costoso y, además, la legislación de muchos países puede no permitir vuelos experimentales. Por ello, cada vez más publicaciones emplean entornos simulados. Estos entornos pueden ser totalmente artificiales o pueden ser simulaciones de entornos del mundo real (Figura V.2). Estos entornos simulados permiten reducir el coste y evitar ciertos aspectos reales pero puede no representar todos los aspectos físicos de manera fidedigna a los que está sujeto el vuelo de cada UAV del enjambre. Por ejemplo, cuando dos UAV vuelan uno cerca del otro surgen turbulencias que comprometen su estabilidad.

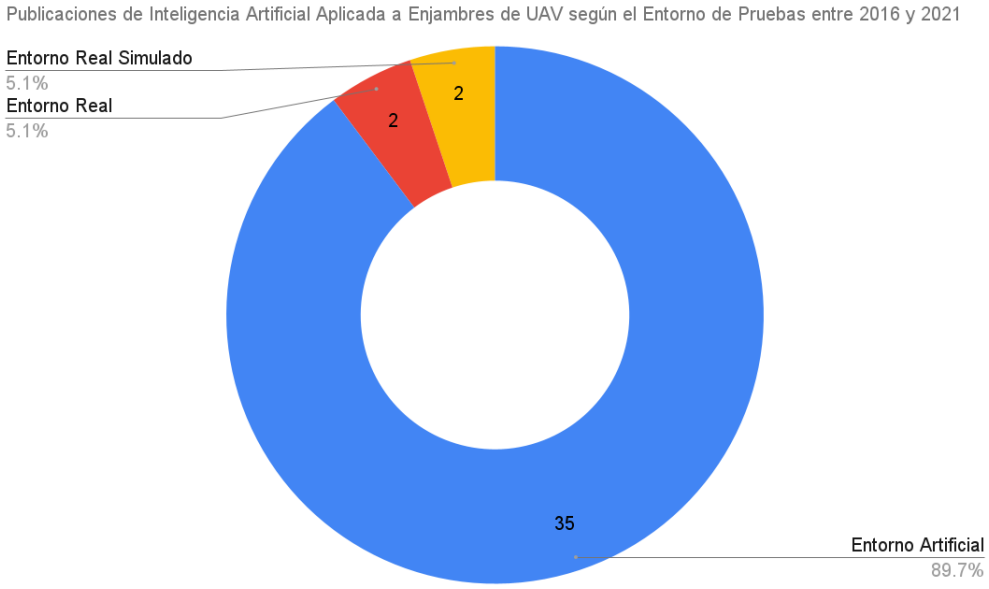


Figura V.2 Gráfico circular de resumen del número de publicaciones de los años 2016 al 2021 donde se aplican las técnicas de IA a la los enjambres de UAV según el tipo de entorno en el que se prueban las técnicas (Puente-Castro et al. 2022*a*).

En resumen, el número de publicaciones sobre enjambres de UAV, aunque limitado, parece estar en crecimiento, lo que refleja su reciente interés en la comunidad académica. Esta tendencia se vincula con la tardía adopción de UAV en aplicaciones civiles, debido a su mayor accesibilidad y menor costo. Esto ha llevado a una mayor experimentación y descubrimiento de posibles aplicaciones en campos como la agricultura, la gestión de recursos naturales y la logística. Esta evolución está impulsando el desarrollo de tecnologías y herramientas más asequibles, lo que promete un aumento en la investigación en los enjambres de UAV en los próximos años, ofreciendo oportunidades para abordar diversos desafíos y aplicaciones.

Tras el análisis realizado en la primera publicación, se pudieron tomar en consi-

deración diferentes aspectos que guiaron la toma de decisiones con respecto a la experimentación. Dadas las tendencias observadas. se han podido identificar debilidades y fortalezas en el dominio. Así, se decidió tomar una estrategia incremental donde se procuraba resolver cada vez un problema más complejo que el anterior.

5.2 Resultados Experimentales

Durante el proceso de experimentación se ha tenido ha seguido siempre un objetivo común: encontrar RR.NN.AA. que sean capaces de completar la operación de vuelo manteniendo la mayor eficiencia posible. Por ello, se han realizado inicialmente experimentos en mapas sin obstáculos y, a partir de las conclusiones obtenidas en ellos, se procedió con la experimentación en mapas con obstáculos.

Todos los experimentos realizados han tenido lugar en entornos simulados propios.

5.2.1 Mapas Sin Obstáculos

Los primeros experimentos realizados fueron ideados en los mapas más sencillos posibles. Es decir, mapas sin obstáculos. Estos experimentos permiten conocer cómo afecta el tamaño del mapa y el tamaño del enjambre a la solución del problema. Para ello, se realizaron un total de 45 experimentos que son los realizados en Puente-Castro et al. (2022*b*) (página 87 de esta Tesis Doctoral) donde se establece como línea base el caso donde se usa un único UAV en cada uno de los mapas. Para los demás casos con múltiples aeronaves, se prueban las capacidades del algoritmo ante diferentes configuraciones de RNA y de tamaño de mapa (Figura 8 de la página 95).

Los resultados de los experimentos son los resumidos en la Tabla 4 de la página 98. En ellos, se detalla el tiempo mínimo necesario para encontrar una solución y el número de soluciones que ha encontrado cada sistema. Como se puede apreciar, el número de soluciones encontradas suele caer a medida que el mapa aumenta su tamaño. Sin embargo, el tiempo necesario no crece de manera proporcional al número de celdas de los mapas (Figura 12 de la página 100). En general, a medida que aumenta el tamaño del mapa, el tiempo mínimo para encontrar una solución solo se ve incrementado en unos pocos minutos para un mismo número de UAV.

Con respecto al número de UAV en el enjambre, a medida que aumenta el número de aeronaves se reduce el porcentaje de acciones correctamente realizadas (Figura 13 de la página 101). Esta situación se debe a que en las primeras fases del aprendizaje los UAV cometen muchos errores que, al ser un enjambre, incrementa mucho el número de errores.

Como limitación observada, establecer el tiempo como criterio de optimización lo hace muy dependiente de los recursos computacionales del sistema. Es decir, equipos

informáticos con mayores capacidades van a ser capaces de encontrar más rutas que cumplan con las restricciones temporales impuestas que equipos menos avanzados. Por ello, si el algoritmo es probado en equipos de capacidades muy limitadas puede ser que no llegue a encontrar soluciones.

También cabe destacar la limitación señalada en Puente-Castro et al. (2022*b*) donde se indica que tiene mucha dependencia de la inicialización (Figura 10 de la página 99). Esto se debe a que si inicialmente cada UAV comete muchos errores, éstos quedan almacenados en la memoria y acaban memorizando estas situaciones. En estos casos, se propagan los errores y puede condicionar la obtención de rutas de vuelo satisfactorias.

En general, el modelo elegido ha demostrado su capacidad para encontrar resultados satisfactorios en cualquiera de las situaciones sin obstáculos y dados los límites temporales establecidos. En cualquier caso de los planteados en los experimentos, se ha podido alcanzar, por lo menos, un conjunto de rutas como solución. Todo esto demuestra que el algoritmo propuesto es capaz de operar en mapas sin obstáculos.

5.2.2 Mapas Con Obstáculos

Al ampliar la experimentación a un conjunto de mapas con obstáculos (Figura IV.9), como se ve en Puente-Castro et al. (2023), se procuró conocer el comportamiento del algoritmo ante diferentes condiciones en mapas de diferentes tamaños (página 105 de esta Tesis Doctoral). Para evitar la dependencia de los recursos computacionales se prescinde de la medida del tiempo y se sustituye por la medida del número de movimientos o acciones que toman los UAV para realizar la tarea. Así, se tiene una métrica únicamente dependiente del algoritmo y no de las capacidades computaciones, a pesar de perder la restricción temporal de vuelo.

En los resultados presentados en el artículo se refuerzan las bondades del uso de enjambres de UAV (Tabla 3 de la página 115). A medida que aumenta el número de UAV en el enjambre se reduce el número de movimientos necesarios para cubrir el terreno. Esta situación es común a las dos configuraciones de la RNA con respecto a los agentes (Figura 3 de la página 112): una red global para todos y una red local para cada uno. Cabe destacar que la poca varianza que hay en la mayoría de casos demuestran que el sistema realiza predicciones robustas y poco arbitrarias.

Realizando un test de significancia comparando los resultados obtenidos con la RNA del artículo Puente-Castro et al. (2022*b*) y la RNA actual, solo se aprecian diferencias significativas en el mapa de 5×5 celdas para 2 UAV (Figura 5 de la página 115). A pesar de ello, el modelo propuesto suele tener mejores resultados.

La principal limitación de la aproximación propuesta es la atomicidad de los movimientos. Al tratarse de mapas con obstáculos fijos habría rutas que requiriesen menos

acciones si se tratase de movimientos en diagonal. Por contrapartida, los movimientos en diagonal implican mayor control del UAV por tratarse de la combinación de movimientos atómicos en vertical y horizontal.

El algoritmo ha demostrado sus capacidades en obstáculos fijos que no varían sus posiciones en ningún momento. Por otra parte, no ha sido probado en entornos con obstáculos que se puedan mover o que puedan surgir espontáneamente. La aparición de estos obstáculos o su desplazamiento implica cómputos adicionales, pues la ruta de vuelo debe ser modificada. El caso de los obstáculos que surgen espontáneamente debe ser considerado con cuidado, pues un obstáculo que aparezca en un momento dado y desaparezca poco después, como puede ser un pájaro, puede no afectar a las rutas o puede asumirse cierto margen de error en ellas.

Debido al procedimiento incremental de experimentación, se ha probado que el algoritmo propuesto es capaz de operar ante diferentes situaciones a medida que se aumenta la complejidad. Así, queda probada la hipótesis inicial expuesta en el Capítulo II con los objetivos de esta Tesis Doctoral.

Chapter VI

Conclusions

This Ph.D. Thesis proposes that Q-Learning can be used for controlling unmanned aerial vehicles (UAV) swarms in maps, both with and without obstacles. To undertake this research, it started with an exhaustive bibliographical analysis of the field of study. This analysis allowed for the proper orientation of the experimental design by identifying current trends and areas where significant improvement was needed.

The results of the conducted bibliographic analysis demonstrate that UAV swarm artificial intelligence approaches are thriving and continuously evolving. An objective indicator of this phenomenon is the increasing utilization of AI methods in UAV swarms for solving Path Planning problems over time. There is a notable rise in the publication rate of papers in this domain, and given the substantial number of publications in the first quarter, it is likely that we will observe a continued high rate of publications even in 2021. Furthermore, quantitative research reveals that, irrespective of the specific application domain, Reinforcement Learning (RL) and Evolutionary Computationg (EC) remain the most widely adopted methods. These techniques are predominantly assessed in simulated flight environments, primarily due to the numerous external factors that could impact UAV performance, making real-world operations challenging for many of these systems.

Nonetheless, certain challenges persist, with the most notable being the absence of standardization in the results. This stems from the fact that each research paper concentrates on a distinct variable, exploring various facets of path planning. This lack of standardized criteria for determining the superiority of one technique over another could potentially limit the development of new systems in this field. However, it also underscores the relatively nascent nature of the subject matter.

From the experimental process designed after the bibliographic analysis, it has been possible to obtain results of different nature. These results encompass various aspects, ranging from quantitative data showcasing the performance metrics of the proposed

methods to qualitative insights that shed light on the nuanced intricacies of their application.

The results obtained in the experiments were highly satisfactory, supporting the initial hypothesis that the combination of Q-Learning and Artificial Neural Networks can provide an effective method for controlling UAVs across a wide range of diverse environments. It is noteworthy that this Ph.D. Thesis distinguishes itself by harnessing the deliberate omission of any supplementary map data beyond the inherent morphology and obstacle layout, thus supporting the initial hypothesis. Consequently, the application of Artificial Intelligence techniques for comprehensive terrain data acquisition is now assured. This distinctiveness not only sets it apart but also holds immense promise for the broader research community. It presents an opportunity for fellow authors to explore and adapt the concept of simplified maps as a foundational building block for their own endeavors in swarm control and related fields. By demonstrating the efficacy of such an approach, this Thesis opens doors to a more streamlined and efficient methodology in controlling UAV swarms.

In essence, this Thesis serves as a starting point, offering a novel perspective on the intersection of Artificial Neural Networks and UAV swarm control. Its innovative approach, centered around the utilization of simplified map data, has the potential to reshape the landscape of research and application in this domain. As such, it not only contributes to the existing body of knowledge but also invites others to embark on a journey of exploration and innovation to pursue more efficient and effective swarm control systems.

Under the light of these findings, it is evident that the system under consideration is not only capable but also highly proficient in tackling Path Planning problems where the primary objective is to maximize the traversable surface area. This makes it a good tool for addressing a wide range of complex scenarios that demand efficient and comprehensive Path Planning solutions.

The demonstrated proficiency of this system in optimizing travel paths across expansive surfaces under different conditions without extra information represents a significant breakthrough. It opens up new avenues for addressing real-world challenges that involve extensive terrain coverage, such as autonomous exploration, search and rescue missions, and agricultural field mapping, reaffirming its potential to drive innovation and efficiency across various domains.

The system for performing the experiments offers adaptability across various devices by optimizing flight paths and reducing actions as the UAV swarm expands. This shift towards autonomous UAV swarms provides cost savings, time efficiency, and improved fault tolerance compared to single UAVs or manual management.

Since it is not necessary to extract additional features of the spatial relationship of the obstacles within the rest of the environment, it can be understood that the sequence of movements and the position of the UAVs in the swarm are more critical. Additionally, unlike other published work in this field, there is no need to provide additional information on the map to direct the paths. Therefore, the system can calculate the optimal paths using only the information from the cell map, the current position of the UAVs on the map, and the evolution of the flight paths along the map. Using such limited information eliminates the necessity of prior terrain knowledge. If additional information is to be added to guide the UAV paths, a thorough study is necessary. Consequently, many users may opt not to use the system due to this added difficulty.

On the other hand, if it is necessary to guide the paths, the system can become biased because user errors may prevent the discovery of better paths. The disadvantage of not employing targets, as other authors in the state of the art have done, is that the algorithm relies on scanning. As it heavily depends on scanning, it is vital to tune the algoritm parameters by an experimental process in order to minimize issues with path computations.

The experiments have certain limitations that should be acknowledged. Firstly, the UAV movements are treated atomically, which may not be ideal for tasks requiring smoother paths and efficient data capture. This granularity might restrict the system's ability to handle tasks demanding more continuous and refined movement. Additionally, the system does not take into account varying UAV heights, which could potentially impact path calculations and the accuracy of rewards based on data quality. However, it is worth noting that UAVs typically maintain altitudes that accommodate disturbances, and adjusting the height to navigate obstacles, such as birds, would typically involve only minor changes. Despite these limitations, it is important to emphasize that the system consistently achieves satisfactory results across different flight heights, demonstrating its robustness and adaptability in real-world scenarios.

Capítulo VII

Trabajos Futuros

Esta Tesis Doctoral proporciona una base para futuras investigaciones sobre enjambres de UAV y la planificación de rutas de vuelo, en particular en lo que se refiere a experimentos con mapas divididos en celdas.

El análisis bibliográfico realizado presenta diferentes maneras de clasificar los trabajos encontrados en el estado del arte. Mantener la revisión bibliográfica actualizadas puede ser de gran ayuda. Gracias a estas taxonomías, se pueden ver las tendencias en el dominio en términos de innovación y desarrollo. Con estas tendencias, otros autores pueden tener referencias a la hora de diseñar sus experimentos e incluso, crear productos comerciales.

El proceso experimental descrito en esta Tesis Doctoral puede ser fundamental en la investigación científica, ya que no solo proporciona resultados concretos, sino que también sienta las bases para futuras experimentaciones. Al exponer y detallar las métricas utilizadas para evaluar las rutas experimentales, junto con sus limitaciones y ventajas, se crea un valioso recurso para la comunidad científica. Estas métricas se pueden convertir en un sistema de referencia esencial que permite a los investigadores evaluar sus propios proyectos de manera objetiva y rigurosa. Además, esta información facilita la realización de comparativas y análisis estadísticos entre diferentes estudios, lo que enriquece el entendimiento del tema y contribuye a la validación de los resultados obtenidos.

Todo el proceso experimental también puede generar beneficios significativos para diversos sectores de negocio debido a que pone en conocimiento las capacidades y las limitaciones encontrados durante su desarrollo experimental. La investigación y desarrollo de nuevas tecnologías, productos y servicios a menudo se basa en los resultados de experimentos científicos. Estos hallazgos pueden dar lugar a innovaciones que mejoran la eficiencia, la calidad y la competitividad de las empresas. A pesar de ello, diferentes sectores implican diferentes necesidades para el cálculo de rutas de vuelo. Por ello, la experimentación debe ampliarse a diferentes tipos de necesidad y no solo cubrir

la totalidad del mapa. Por ejemplo, en casos de patrulla, las rutas deben garantizar que los UAV se muevan de manera cíclica y, además, sus sensores no tengan puntos ciegos.

Otra línea de trabajo futuro podría basarse en el uso de EC. En una primera aproximación, este tipo de algoritmos podría ser utilizado para determinar el cálculo de las rutas de vuelo, prescindiendo del uso de RL y de una RNA, y dejando que el proceso evolutivo realice todo este trabajo. Otra opción es emplear estos algoritmos para encontrar la mejor parametrización de una RNA, de manera que sea altamente eficiente y efectiva en la resolución de tareas de control de enjambre de UAV. Esto podría lograrse al optimizar los hiperparámetros y la arquitectura de la propia RNA utilizando técnicas de EC en conjunto con datos de experiencias acumuladas en la memoria empleada por el algoritmo *Q-Learning*. Este enfoque permitiría investigaciones donde también se podrían emplear entornos más complejos, como mapas en 3D, permitiendo a los UAV ejecutar diferentes movimientos propios de las aeronaves en diferentes ejes, como cabeceo y alabeo. Las mejoras podrían consistir en implementar acciones como detenerse para mitigar los riesgos de colisión en rutas que se cruzan en vez de considerar solo la evasión.

Otra de las posibles mejoras consiste en la consecución de un conjunto más amplio y diversificado de movimientos disponibles para los UAV. Esta expansión de posibilidades se traduce en un mayor espectro de acciones que los UAV pueden emprender para alcanzar sus objetivos. Estas acciones pueden incluso ser concebidas como combinaciones en diferentes grados de las acciones atómicas previamente mencionadas.Incluso, la inclusión de comandos aparentemente simples, como “parada”, en la gama de acciones disponibles puede enriquecer significativamente la versatilidad del sistema.

La incorporación de un mayor número de acciones, incluidas algunas acciones compuestas o combinadas, introduce un nuevo nivel de complejidad en el control de rutas de los UAV. Esto puede dificultar el proceso de obtención de las rutas, ya que la matriz de movimientos se expande. Sin embargo, esta mayor complejidad puede traducirse en una precisión mejorada en los movimientos ejecutados por los UAV. Cada acción, ya sea simple o compuesta, representa una herramienta adicional para los UAV en la consecución de sus objetivos.

Aumentar la precisión de los movimientos puede conllevar una mayor complejidad del sistema. Por ejemplo, podría estudiarse la integración de varias RR.NN.AA., cada una para llevar a cabo distintas funciones. La combinación de varias RR.NN.AA. ofrece la posibilidad de incorporar capacidades de vuelo adicionales, como ajustes de altitud o inclinación. El empleo de varias RNA para coordinar movimientos compuestos, como el ascenso y los giros simultáneos, puede mejorar la precisión y agilizar los resultados.

También cabe destacar la presencia de obstáculos dinámicos que surjan de mane-

ra inesperada. Estos obstáculos pueden forzar el cambio repentino de la trayectoria. Dichos cambios repentinos implican mejoras en el control, puesto que no todas las aeronaves permiten reacciones rápidas y bruscas.

De manera similar, es necesario contemplar casos en los que UAV del enjambre se vean comprometidos de manera inesperada. En tales circunstancias, la planificación y la ejecución de rutas se vuelven críticas, ya que los drones restantes deben adaptarse de manera ágil y eficiente para asumir las responsabilidades de los miembros afectados.

La gestión de contingencias y la replanificación de rutas se convierten en factores clave en estos escenarios. Los algoritmos de control deben estar diseñados para detectar de manera proactiva las interrupciones en la formación del enjambre y coordinar respuestas rápidas y efectivas. Esto puede incluir la redistribución de tareas y la asignación dinámica de objetivos, garantizando que los drones restantes puedan absorber la carga de trabajo de los miembros afectados sin comprometer la eficacia global de la misión.

Además de las direcciones de investigación mencionadas anteriormente, es importante destacar la oportunidad de expandir aún más la experimentación y evaluación del algoritmo propuesto. Se puede explorar el rendimiento del algoritmo en una variedad de vehículos autónomos, como UUV, USV o UGV, ya sea en enjambres exclusivamente compuestos por un tipo de vehículo o en enjambres mixtos. Esta extensión ampliaría la aplicabilidad del algoritmo y su adaptabilidad a diferentes contextos operativos y desafíos de movimiento, como la navegación terrestre o marítima, y abriría nuevas posibilidades para aplicaciones colaborativas y multidisciplinarias. Un ejemplo de estas aplicaciones puede ser el uso de de UAV para asistir a la navegación de USV para realizar estudios marinos o, incluso, en tareas militares.

Bibliografía

Aggarwal, S. y Kumar, N. (2020), 'Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges', *Computer Communications* **149**, 270–299.

Ahmed, M., Seraj, R. y Islam, S. M. S. (2020), 'The k-means algorithm: A comprehensive survey and performance evaluation', *Electronics* **9**(8), 1295.

Akhloufi, M. A., Arola, S. y Bonnet, A. (2019), 'Drones chasing drones: Reinforcement learning and deep search area proposal', *Drones* **3**(3), 58.

Albani, D., IJsselmuiden, J., Haken, R. y Trianni, V. (2017), Monitoring and mapping with robot swarms for agricultural applications, *in* '2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)', IEEE, pp. 1–6.

Alrubaye, G. I. H. y Miyauchi, H. (2023), 'Global drone regulations and research survey with the examination of its application', *Technical Journal of Advanced Mobility* **4**(5), 62–73.

Aslan, M. F., Durdu, A., Sabanci, K., Ropelewska, E. y Gültekin, S. S. (2022), 'A comprehensive survey of the recent studies with uav for precision agriculture in open fields and greenhouses', *Applied Sciences* **12**(3), 1047.

Baldazo, D., Parras, J. y Zazo, S. (2019), Decentralized multi-agent deep reinforcement learning in swarms of drones for flood monitoring, *in* '2019 27th European Signal Processing Conference (EUSIPCO)', IEEE, pp. 1–5.

Beard, R. W., Ferrin, J. y Humpherys, J. (2014), 'Fixed wing uav path following in wind with input constraints', *IEEE Transactions on Control Systems Technology* **22**(6), 2103–2117.

Bellman, R. (1966), 'Dynamic programming', *Science* **153**(3731), 34–37.

Beni, G. (2004), From swarm intelligence to swarm robotics, *in* 'International workshop on swarm robotics', Springer, pp. 1–9.

Beni, G. y Wang, J. (1993), Swarm intelligence in cellular robotic systems, *in* 'Robots and biological systems: towards a new bionics?', Springer, pp. 703–712.

Bhattacharya, P. y Gavrilova, M. L. (2008), 'Roadmap-based path planning-using the

voronoi diagram for a clearance-based shortest path', *IEEE Robotics and Automation Magazine* **15**(2), 58–66.

Bishop, C. M. y Nasrabadi, N. M. (2006), *Pattern recognition and machine learning*, Vol. 4, Springer.

Borenstein, J., Everett, H. R., Feng, L. y Wehe, D. (1997), 'Mobile robot positioning: Sensors and techniques', *Journal of robotic systems* **14**(4), 231–249.

Buckley, J. D. y Buckley, J. J. (1999), *Air power in the age of total war*, Indiana University Press.

Cekmez, U., Ozsiginan, M. y Sahingoz, O. K. (2016), Multi-uav path planning with parallel genetic algorithms on cuda architecture, *in* 'Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion', pp. 1079–1086.

Cekmez, U., Ozsiginan, M. y Sahingoz, O. K. (2018), Multi-uav path planning with multi colony ant optimization, *in* 'Intelligent Systems Design and Applications: 17th International Conference on Intelligent Systems Design and Applications (ISDA 2017) held in Delhi, India, December 14-16, 2017', Springer, pp. 407–417.

Chen, Y.-J., Chang, D.-K. y Zhang, C. (2020), 'Autonomous tracking using a swarm of uavs: A constrained multi-agent reinforcement learning approach', *IEEE Transactions on Vehicular Technology* **69**(11), 13702–13717.

Cimino, M. G., Lazzeri, A. y Vaglini, G. (2016), Using differential evolution to improve pheromone-based coordination of swarms of drones for collaborative target detection., *in* 'ICPRAM', pp. 605–610.

Clifton, J. y Laber, E. (2020), 'Q-learning: Theory and applications', *Annual Review of Statistics and Its Application* **7**, 279–301.

Contreras-Cruz, M. A., Ayala-Ramirez, V. y Hernandez-Belmonte, U. H. (2015), 'Mobile robot path planning using artificial bee colony and evolutionary programming', *Applied Soft Computing* **30**, 319–328.

Debnath, S. K., Omar, R., Bagchi, S., Sabudin, E. N., Shee Kandar, M. H. A., Foysol, K. y Chakraborty, T. K. (2021), Different cell decomposition path planning methods for unmanned air vehicles-a review, *in* 'Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019: NUSYS'19', Springer, pp. 99–111.

Delavarpour, N., Koparan, C., Nowatzki, J., Bajwa, S. y Sun, X. (2021), 'A technical study on uav characteristics for precision agriculture applications and associated practical challenges', *Remote Sensing* **13**(6), 1204.

Deng, L., Mao, Z., Li, X., Hu, Z., Duan, F. y Yan, Y. (2018), 'Uav-based multispectral

remote sensing for precision agriculture: A comparison between different cameras', *ISPRS journal of photogrammetry and remote sensing* **146**, 124–136.

Ding, F., Zhang, Z., Fu, M., Wang, Y. y Wang, C. (2018), Energy-efficient path planning and control approach of usv based on particle swarm optimization, *in* 'OCEANS 2018 MTS/IEEE Charleston', IEEE, pp. 1–6.

Dorigo, M., Birattari, M., Garnier, S., Hamann, H., de Oca, M. M., Solnon, C. y Stützle, T. (2007), 'Swarm intelligence', *Scholarpedia* **2**(9), 1462.

Dubins, L. E. (1957), 'On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents', *American Journal of mathematics* **79**(3), 497–516.

Fan, J., Wang, Z., Xie, Y. y Yang, Z. (2020), A theoretical analysis of deep q-learning, *in* 'Learning for Dynamics and Control', PMLR, pp. 486–489.

Fernández-Blanco, E. (2010), 'Modelado de un sistema celular artificial para generación de formas y procesado de información'.

Foerster, J., Nardelli, N., Farquhar, G., Afouras, T., Torr, P. H., Kohli, P. y Whiteson, S. (2017), Stabilising experience replay for deep multi-agent reinforcement learning, *in* 'International conference on machine learning', PMLR, pp. 1146–1155.

Forrest, S. (1996), 'Genetic algorithms', *ACM computing surveys (CSUR)* **28**(1), 77–80.

Gage, D. W. (1995), Ugv history 101: A brief history of unmanned ground vehicle (ugv) development efforts, Technical report, NAVAL COMMAND CONTROL AND OCEAN SURVEILLANCE CENTER RDT AND E DIV SAN DIEGO CA.

Gago, J., Douthe, C., Coopman, R. E., Gallego, P. P., Ribas-Carbo, M., Flexas, J., Escalona, J. y Medrano, H. (2015), 'Uavs challenge to assess water stress for sustainable agriculture', *Agricultural water management* **153**, 9–19.

Garland, M., Le Grand, S., Nickolls, J., Anderson, J., Hardwick, J., Morton, S., Phillips, E., Zhang, Y. y Volkov, V. (2008), 'Parallel computing experiences with cuda', *IEEE micro* **28**(4), 13–27.

Giesbrecht, J. (2004), Global path planning for unmanned ground vehicles, Technical report, Defence Research and Development Suffield (ALBERTA).

Gläscher, J., Daw, N., Dayan, P. y O'Doherty, J. P. (2010), 'States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning', *Neuron* **66**(4), 585–595.

Gold, C. M., Remmele, P. R. y Roos, T. (1996), 'Voronoi methods in gis', *Advanced School on the Algorithmic Foundations of Geographic Information Systems* pp. 21–35.

Gonzalez-Aguilera, D. y Rodriguez-Gonzalvez, P. (2017), 'Drones—an open access journal'.

Hafez, A. T., Givigi, S. N., Yousefi, S. y Iskandarani, M. (2017), 'Multi-uav tactic switching via model predictive control and fuzzy q-learning', *Journal of Engineering Science and Military Technologies* **1**(2), 44–57.

Hassanalian, M., Khaki, H. y Khosravi, M. (2015), 'A new method for design of fixed wing micro air vehicle', *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* **229**(5), 837–850.

Hausknecht, M., Stone, P. y Mc, O.-p. (2016), On-policy vs. off-policy updates for deep reinforcement learning, *in* 'Deep reinforcement learning: frontiers and challenges, IJCAI 2016 Workshop', AAAI Press New York, NY, USA.

Heppner, F., Behaviour, A. et al. (2009), 'Organized flight in birds', *Animal Behaviour* **78**, 777–789.

Holland, J. H. (1992), *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, MIT press.

Huang, T., Wang, Y., Cao, X. y Xu, D. (2020), Multi-uav mission planning method, *in* '2020 3rd International Conference on Unmanned Systems (ICUS)', IEEE, pp. 325–330.

Hung, S.-M. y Givigi, S. N. (2016), 'A q-learning approach to flocking with uavs in a stochastic environment', *IEEE transactions on cybernetics* **47**(1), 186–197.

Hwang, Y. K., Ahuja, N. et al. (1992), 'A potential field approach to path planning.', *IEEE transactions on robotics and automation* **8**(1), 23–32.

Kaluđer, H., Brezak, M. y Petrović, I. (2011), A visibility graph based method for path planning in dynamic environments, *in* '2011 Proceedings of the 34th International Convention MIPRO', IEEE, pp. 717–721.

Kimura, H., Yamamura, M. y Kobayashi, S. (1995), Reinforcement learning by stochastic hill climbing on discounted reward, *in* 'Machine Learning Proceedings 1995', Elsevier, pp. 295–303.

Koubâa, A., Allouch, A., Alajlan, M., Javed, Y., Belghith, A. y Khalgui, M. (2019), 'Micro air vehicle link (mavlink) in a nutshell: A survey', *IEEE Access* **7**, 87658–87680.

Krogh, A. (2008), 'What are artificial neural networks?', *Nature biotechnology* **26**(2), 195–197.

Kusyk, J., Uyar, M. U., Ma, K., Samoylov, E., Valdez, R., Plishka, J., Hoque, S. E., Bertoli, G. y Boksiner, J. (2021), 'Artificial intelligence and game theory controlled autonomous uav swarms', *Evolutionary Intelligence* **14**, 1775–1792.

LeCun, Y., Bengio, Y. y Hinton, G. (2015), 'Deep learning', *nature* **521**(7553), 436–444.

Lin, C. A., Shah, K., Mauntel, L. C. C. y Shah, S. A. (2018), 'Drone delivery of medications: Review of the landscape and legal considerations', *The Bulletin of the American Society of Hospital Pharmacists* **75**(3), 153–158.

Liu, R. y Zou, J. (2018), The effects of memory replay in reinforcement learning, *in* '2018 56th annual allerton conference on communication, control, and computing (Allerton)', IEEE, pp. 478–485.

Liu, Y., Liu, H., Tian, Y. y Sun, C. (2020), 'Reinforcement learning based two-level control framework of uav swarm for cooperative persistent surveillance in an unknown urban area', *Aerospace Science and Technology* **98**, 105671.

Liu, Y. y Passino, K. M. (2000), 'Swarm intelligence: Literature overview', *Department of electrical engineering, the Ohio State University* .

Liu, Z., Zhang, Y., Yu, X. y Yuan, C. (2016), 'Unmanned surface vehicles: An overview of developments and challenges', *Annual Reviews in Control* **41**, 71–93.

Luo, W., Tang, Q., Fu, C. y Eberhard, P. (2018), Deep-sarsa based multi-uav path planning and obstacle avoidance in a dynamic environment, *in* 'Advances in Swarm Intelligence: 9th International Conference, ICSI 2018, Shanghai, China, June 17-22, 2018, Proceedings, Part II 9', Springer, pp. 102–111.

Lytridis, C., Kaburlasos, V. G., Pachidis, T., Manios, M., Vrochidou, E., Kalampokas, T. y Chatzistamatis, S. (2021), 'An overview of cooperative robotics in agriculture', *Agronomy* **11**(9), 1818.

Maja, M. M. y Ayano, S. F. (2021), 'The impact of population growth on natural resources and farmers' capacity to adapt to climate change in low-income countries', *Earth Systems and Environment* **5**, 271–283.

Majd, A., Ashraf, A., Troubitsyna, E. y Daneshtalab, M. (2018), Integrating learning, optimization, and prediction for efficient navigation of swarms of drones, *in* '2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)', IEEE, pp. 101–108.

Marin-Plaza, P., Hussein, A., Martin, D. y Escalera, A. d. l. (2018), 'Global and local path planning study in a ros-based research platform for autonomous vehicles', *Journal of Advanced Transportation* **2018**, 1–10.

McCulloch, W. S. y Pitts, W. (1943), 'A logical calculus of the ideas immanent in nervous activity', *The bulletin of mathematical biophysics* **5**, 115–133.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Gra-

ves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G. et al. (2015), 'Human-level control through deep reinforcement learning', *nature* **518**(7540), 529–533.

Monaco, M., Cimino, M. G., Vaglini, G., Fusai, F. y Nico, G. (2021), Managing the oceans cleanup via sea current analysis and bio-inspired coordination of usv swarms, *in* '2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS', IEEE, pp. 8344–8347.

Moshkovitz, M., Dasgupta, S., Rashtchian, C. y Frost, N. (2020), Explainable k-means and k-medians clustering, *in* 'International conference on machine learning', PMLR, pp. 7055–7065.

Nex, F. y Remondino, F. (2014), 'Uav for 3d mapping applications: a review', *Applied geomatics* **6**, 1–15.

Nguyen, H. T., Nguyen, T. D., Garratt, M., Kasmarik, K., Anavatti, S., Barlow, M. y Abbass, H. A. (2019), A deep hierarchical reinforcement learner for aerial shepherding of ground swarms, *in* 'Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part I', Springer, pp. 658–669.

Nguyen-Huu, P.-N., Titus, J., Tilbury, D. y Ulsoy, G. (2009), 'Reliability and failure in unmanned ground vehicle (ugv)', *Digit. Equip. Corp., Maynard, MA, USA, Tech. Rep. GRR-TR* **1**.

Oliveira, L. F., Moreira, A. P. y Silva, M. F. (2021), 'Advances in agriculture robotics: A state-of-the-art review and challenges ahead', *Robotics* **10**(2), 52.

Olson, J. M., Bidstrup, C. C., Anderson, B. K., Parkinson, A. R. y McLain, T. W. (2020), Optimal multi-agent coverage and flight time with genetic path planning, *in* '2020 International Conference on Unmanned Aircraft Systems (ICUAS)', IEEE, pp. 228–237.

Omran, M. G., Engelbrecht, A. P. y Salman, A. (2007), 'An overview of clustering methods', *Intelligent Data Analysis* **11**(6), 583–605.

O'Sullivan, D. y Turner, A. (2001), 'Visibility graphs and landscape visibility analysis', *International journal of geographical information science* **15**(3), 221–237.

Owen, G. (2013), *Game theory*, Emerald Group Publishing.

Pan, Y., Yang, Y. y Li, W. (2021), 'A deep learning trained by genetic algorithm to improve the efficiency of path planning for data collection with multi-uav', *Ieee Access* **9**, 7994–8005.

Perez-Carabaza, S., Besada-Portas, E., Lopez-Orozco, J. A. y de la Cruz, J. M. (2018), 'Ant colony optimization for multi-uav minimum time search in uncertain domains', *Applied Soft Computing* **62**, 789–806.

Puente-Castro, A., Rivero, D., Pazos, A. y Fernandez-Blanco, E. (2022*a*), 'A review of artificial intelligence applied to path planning in uav swarms', *Neural Computing and Applications* pp. 1–18.

Puente-Castro, A., Rivero, D., Pazos, A. y Fernandez-Blanco, E. (2022*b*), 'Uav swarm path planning with reinforcement learning for field prospecting', *Applied Intelligence* **52**(12), 14101–14118.

Puente-Castro, A., Rivero, D., Pedrosa, E., Pereira, A., Lau, N. y Fernandez-Blanco, E. (2023), 'Q-learning based system for path planning with unmanned aerial vehicles swarms in obstacle environments', *Expert Systems with Applications* p. 121240.

Quartieri, J., Mastorakis, N. E., Iannone, G., Guarnaccia, C. et al. (2010), A cellular automata model for fire spreading prediction, *in* 'Latest Trends on Urban Planning and Transportation', Vol. 1, WORLD SCIENTIFIC AND ENGINEERING ACAD AND SOC, pp. 173–178.

Ramirez-Atencia, C., Bello-Orgaz, G., R-Moreno, M. D. y Camacho, D. (2017), 'Solving complex multi-uav mission planning problems using multi-objective genetic algorithms', *Soft Computing* **21**, 4883–4900.

Ramirez-Atencia, C., R-Moreno, M. D. y Camacho, D. (2017), 'Handling swarm of uavs based on evolutionary multi-objective optimization', *Progress in Artificial Intelligence* **6**, 263–274.

Rosenblatt, F. (1958), 'The perceptron: a probabilistic model for information storage and organization in the brain.', *Psychological review* **65**(6), 386.

Roudneshin, M., Sizkouhi, A. M. M. y Aghdam, A. G. (2019), Effective learning algorithms for search and rescue missions in unknown environments., *in* 'WiSEE', pp. 76–80.

Rummery, G. A. y Niranjan, M. (1994), *On-line Q-learning using connectionist systems*, Vol. 37, University of Cambridge, Department of Engineering Cambridge, UK.

Russell, S. y Norvig, P. (2021), 'Artificial intelligence: a modern approach, 4th us ed'.

Sands, T. (2020), 'Development of deterministic artificial intelligence for unmanned underwater vehicles (uuv)', *Journal of Marine Science and Engineering* **8**(8), 578.

Sathyan, A., Ernest, N. D. y Cohen, K. (2016), 'An efficient genetic fuzzy approach to uav swarm routing', *Unmanned Systems* **4**(02), 117–127.

Sethu, M., Kotla, B., Russell, D., Madadi, M., Titu, N. A., Coble, J. B., Boring, R. L., Blache, K., Agarwal, V., Yadav, V. et al. (2022), 'Application of artificial intelligence in detection and mitigation of human factor errors in nuclear power plants: A review', *Nuclear Technology* pp. 1–19.

Shah, S., Dey, D., Lovett, C. y Kapoor, A. (2018), Airsim: High-fidelity visual and physical simulation for autonomous vehicles, *in* 'Field and Service Robotics: Results of the 11th International Conference', Springer, pp. 621–635.

Sharkey, A. J. y Sharkey, N. (2006), The application of swarm intelligence to collective robots, *in* 'Advances in applied artificial intelligence', IGI Global, pp. 157–185.

Sharma, A., Shoval, S., Sharma, A. y Pandey, J. K. (2022), 'Path planning for multiple targets interception by the swarm of uavs based on swarm intelligence algorithms: A review', *IETE Technical Review* **39**(3), 675–697.

Sharma, P., Diwakar, M. y Lal, N. (2013), 'Edge detection using moore neighborhood', *International Journal of Computer Applications* **61**(3).

Speck, C. y Bucci, D. J. (2018), Distributed uav swarm formation control via object-focused, multi-objective sarsa, *in* '2018 Annual American Control Conference (ACC)', IEEE, pp. 6596–6601.

Storn, R. y Price, K. (1997), 'Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces', *Journal of global optimization* **11**(4), 341.

Su, X.-h., Zhao, M., Zhao, L.-l. y Zhang, Y.-h. (2016), A novel multi stage cooperative path re-planning method for multi uav, *in* 'PRICAI 2016: Trends in Artificial Intelligence: 14th Pacific Rim International Conference on Artificial Intelligence, Phuket, Thailand, August 22-26, 2016, Proceedings 14', Springer, pp. 482–495.

Sun, H., Qi, J., Wu, C. y Wang, M. (2020), Path planning for dense drone formation based on modified artificial potential fields, *in* '2020 39th Chinese Control Conference (CCC)', IEEE, pp. 4658–4664.

Susanto, T., Setiawan, M. B., Jayadi, A., Rossi, F., Hamdhi, A. y Sembiring, J. P. (2021), Application of unmanned aircraft pid control system for roll, pitch and yaw stability on fixed wings, *in* '2021 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE)', IEEE, pp. 186–190.

Sutton, R. S. y Barto, A. G. (2018), *Reinforcement learning: An introduction*, MIT press.

Sutton, R. S., Precup, D. y Singh, S. (1998), Intra-option learning about temporally abstract actions., *in* 'ICML', Vol. 98, pp. 556–564.

Tahir, A., Böling, J., Haghbayan, M.-H., Toivonen, H. T. y Plosila, J. (2019), 'Swarms of unmanned aerial vehicles—a survey', *Journal of Industrial Information Integration* **16**, 100106.

Thibbotuwawa, A., Nielsen, P., Zbigniew, B. y Bocewicz, G. (2019), Energy consumption in unmanned aerial vehicles: A review of energy consumption models and their

relation to the uav routing, *in* 'Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology–ISAT 2018: Part II', Springer, pp. 173–184.

Toffoli, T. y Margolus, N. (1987), *Cellular automata machines: a new environment for modeling*, MIT press.

Tripicchio, P., Satler, M., Dabisias, G., Ruffaldi, E. y Avizzano, C. A. (2015), Towards smart farming and sustainable agriculture with drones, *in* '2015 international conference on intelligent environments', IEEE, pp. 140–143.

Tsouros, D. C., Bibi, S. y Sarigiannidis, P. G. (2019), 'A review on uav-based applications for precision agriculture', *Information* **10**(11), 349.

Van Hasselt, H. y Wiering, M. A. (2007), Reinforcement learning in continuous action spaces, *in* '2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning', IEEE, pp. 272–279.

Walter, B., Sannier, A., Reiners, D. y Oliver, J. (2006), 'Uav swarm control: Calculating digital pheromone fields with the gpu', *The Journal of Defense Modeling and Simulation* **3**(3), 167–176.

Wang, G., Yang, Y. y Wang, S. (2020), 'Ocean thermal energy application technologies for unmanned underwater vehicles: A comprehensive review', *Applied Energy* **278**, 115752.

Wang, L., Liu, Z., Liu, A. y Tao, F. (2021), 'Artificial intelligence in product lifecycle management', *The International Journal of Advanced Manufacturing Technology* **114**, 771–796.

Wang, T., Qin, R., Chen, Y., Snoussi, H. y Choi, C. (2019), 'A reinforcement learning approach for uav target searching and tracking', *Multimedia Tools and Applications* **78**, 4347–4364.

Watkins, C. J. y Dayan, P. (1992), 'Q-learning', *Machine learning* **8**, 279–292.

Wu, J., Song, C., Ma, J., Wu, J. y Han, G. (2021), 'Reinforcement learning and particle swarm optimization supporting real-time rescue assignments for multiple autonomous underwater vehicles', *IEEE Transactions on Intelligent Transportation Systems* **23**(7), 6807–6820.

Xia, G., Sun, X. y Xia, X. (2021), 'Multiple task assignment and path planning of a multiple unmanned surface vehicles system based on improved self-organizing mapping and improved genetic algorithm', *Journal of Marine Science and Engineering* **9**(6), 556.

Xie, J., Zhou, R., Liu, Y., Luo, J., Xie, S., Peng, Y. y Pu, H. (2021), 'Reinforcement-

learning-based asynchronous formation control scheme for multiple unmanned surface vehicles', *Applied Sciences* **11**(2), 546.

Yang, Q., Jang, S.-J. y Yoo, S.-J. (2020), 'Q-learning-based fuzzy logic for multi-objective routing algorithm in flying ad hoc networks', *Wireless Personal Communications* **113**, 115–138.

Yang, X.-S. (2015), *Recent advances in swarm intelligence and evolutionary computation*, Springer.

Ye, F., Chen, J., Tian, Y. y Jiang, T. (2020), 'Cooperative multiple task assignment of heterogeneous uavs using a modified genetic algorithm with multi-type-gene chromosome encoding strategy', *Journal of intelligent and robotic systems* **100**, 615–627.

Yildiz, O., Yilmaz, A. E. y Gokalp, B. (2009), 'State-of-the-art system solutions for unmanned underwater vehicles', *sensors* **1**(2).

Yogeswaran, M. y Ponnambalam, S. (2012), 'Reinforcement learning: Exploration–exploitation dilemma in multi-agent foraging task', *Opsearch* **49**, 223–236.

Zadeh, L. A. (2023), Fuzzy logic, *in* 'Granular, Fuzzy, and Soft Computing', Springer, pp. 19–49.

Zhang, H.-y., Lin, W.-m. y Chen, A.-x. (2018), 'Path planning for the mobile robot: A review', *Symmetry* **10**(10), 450.

Zhao, W., Qiu, W., Zhou, T., Shao, X. y Wang, X. (2019), Sarsa-based trajectory planning of multi-uavs in dense mesh router networks, *in* '2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)', IEEE, pp. 1–5.

Zhen, Z., Xing, D. y Gao, C. (2018), 'Cooperative search-attack mission planning for multi-uav based on intelligent self-organized algorithm', *Aerospace Science and Technology* **76**, 402–411.

Zheng, C., Li, L., Xu, F., Sun, F. y Ding, M. (2005), 'Evolutionary route planner for unmanned air vehicles', *IEEE Transactions on robotics* **21**(4), 609–620.

Zhou, W., Liu, Z., Li, J., Xu, X. y Shen, L. (2021), 'Multi-target tracking for unmanned aerial vehicle swarms using deep reinforcement learning', *Neurocomputing* **466**, 285–297.

Zhou, Z., Luo, D., Shao, J., Xu, Y. y You, Y. (2020), 'Immune genetic algorithm based multi-uav cooperative target search with event-triggered mechanism', *Physical Communication* **41**, 101103.

Zhuang, Y., Sun, Y. y Wang, W. (2012), 'Mobile robot hybrid path planning in an

obstacle-cluttered environment based on steering control and improved distance propagating', *Int. J. Innov. Comput. Inf. Control* **8**, 4095–4109.

Segunda Parte

Lista de Publicaciones

1. Alejandro Puente-Castro, Daniel Rivero, Alejandro Pazos, Enrique Fernandez-Blanco. **A review of artificial intelligence applied to path planning in UAV swarms**. Neural Computing and Applications 34, 153–170 (2022). https://doi.org/10.1007/s00521-021-06569-4

2. Alejandro Puente-Castro, Daniel Rivero, Alejandro Pazos, Enrique Fernandez-Blanco. **UAV swarm path planning with reinforcement learning for field prospecting**. Applied Intelligence 52, 14101–14118 (2022). https://doi.org/10.1007/s10489-022-03254-4

3. Alejandro Puente-Castro, Daniel Rivero, Eurico Pedrosa, Artur Pereira, Nuno Lau, Enrique Fernandez-Blanco. **Q-Learning based system for path planning with unmanned aerial vehicles swarms in obstacle environments**. Expert Systems with Applications, 121240 (2023). https://doi.org/10.1016/j.eswa.2023.121240

Lista de Congresos y Conferencias

1. Alejandro Puente-Castro, Daniel Rivero, Alejandro Pazos, Enrique Fernandez-Blanco. **Using Reinforcement Learning in the Path Planning of Swarms of UAVs for the Photographic Capture of Terrains**. Engineering Proceedings. 2021; 7(1):32.
   doi:10.3390/engproc2021007032

2. Alejandro Puente-Castro, Daniel Rivero, Alejandro Pazos, Enrique Fernandez-Blanco. **Artificial Intelligence techniques for autonomous drone swarms**, in Proceedings of the MOL2NET'21, Conference on Molecular, Biomed., Comput. and Network Science and Engineering, 7th ed., 25 January–30 December 2021, MDPI: Basel, Switzerland.
   doi:10.3390/mol2net-07-11845

3. Jornadas para la difusión del conocimiento sobre la amenaza que representan para la seguridad física y la protección de la Fuerza los drones comerciales y la eficacia de los medios de mitigación disponibles, organizadas por la Fuerza de Protección de la Armada en la Base Naval de Rota el día 18 de abril de 2023.

ANEXOS

Neural Computing and Applications (2022) 34:153–170
https://doi.org/10.1007/s00521-021-06569-4

REVIEW

A review of artificial intelligence applied to path planning in UAV swarms

Alejandro Puente-Castro[1] · Daniel Rivero[1] · Alejandro Pazos[1,2] · Enrique Fernandez-Blanco[1]

Received: 16 June 2021 / Accepted: 21 September 2021 / Published online: 14 October 2021

Abstract
Path Planning problems with Unmanned Aerial Vehicles (UAVs) are among the most studied knowledge areas in the related literature. However, few of them have been applied to groups of UAVs. The use of swarms allows to speed up the flight time and, thus, reducing the operational costs. When combined with Artificial Intelligence (AI) algorithms, a single system or operator can control all aircraft while optimal paths for each one can be computed. In order to introduce the current situation of these AI-based systems, a review of the most novel and relevant articles was carried out. This review was performed in two steps: first, a summary of the found articles; second, a quantitative analysis of the publications found based on different factors, such as the temporal evolution or the number of articles found based on different criteria. Therefore, this review provides not only a summary of the most recent work but it gives an overview of the trend in the use of AI algorithms in UAV swarms for Path Planning problems. The AI techniques of the articles found can be separated into four main groups based on their technique: reinforcement Learning techniques, Evolutive Computing techniques, Swarm Intelligence techniques, and, Graph Neural Networks. The final results show an increase in publications in recent years and that there is a change in the predominance of the most widely used techniques.

Keywords Unmanned aerial vehicle · UAV · Swarm intelligence · Path planning

1 Introduction

Swarms of Unmanned Aerial Vehicles or UAVs are a revolution in both industrial and recreational fields. They make it possible to perform industrial tasks faster and more economical while maintaining safety. Mostly because to their compact size, low cost, and overall ease of management and operation [39]. In this way, UAVs are very useful tools when it comes to carrying out tasks in places that are difficult to access. The battery life can be considered in as a major disadvantage due to their limited operational time. Thus, tasks that require flying over large areas are a problem.

In addition to the individual advantages and challenges, operating in heterogeneous groups or swarms provides other advantages. Among them, the most important is the time reduction of some operations by performing the same task simultaneously and the capacity to perform tasks that require flying over large areas [15].

Several sectors could benefit from these advantages. One example is the agricultural sector, where these swarms are used in tasks such as field or crop monitoring [2]. Other papers propose applications in military or rescue cases [60]. Within the field of emergencies and rescues, they can also be used in monitoring natural disasters such as floods [7].

Nevertheless, not all applications are purely industrial, other examples are recreational. For example, there are numerous works that coordinate multiple UAVs for image capture and composition, like the work of Moeller et al. [69]. Another recreational activity in which UAV swarms are being used is their use as an alternative to fireworks [22]. This last activity is being highly considered in other countries because of the lack of legislation on autonomous

✉ Alejandro Puente-Castro
a.puentec@udc.es

[1] Faculty of Computer Science, CITIC, University of A Coruna, 15072 A Coruna, Spain

[2] Biomedical Research Institute of A Coruna (INIBIC), University Hospital Complex of A Coruna (CHUAC), 15006 A Coruna, Spain

flights over civil populations. Therefore, many countries do not have legislation for this case.

All operations, whether industrial or non-industrial, depend on the flight path. It is important to know the most optimal path possible. In this way, the flight runs quickly and efficiently. This path calculation is known as the Path Planning problem [30]. These problems seek, in addition to path calculation, the autonomous control of the UAVs. Therefore, less operator intervention is required, and they maintain, during the whole operation, the efficiency of the flight regardless of obstacles or other problems that may arise. In other words, reducing personal and aerial distance results in significant cost savings. The most recent works are focused on autonomous swarm coordination. How this coordination is performed could reduce the operations to find the optimal paths.

For better autonomous control of the swarms while maintaining path optimization, the authors are mainly making use of Artificial Intelligence (AI) techniques in their works [83]. Thus, they obtain systems capable of efficiently abstracting knowledge and, from this knowledge, calculating paths and controlling UAV trajectories simultaneously and automatically. The importance of these techniques and their application to Path Planning problems in UAV swarms is discussed in more detail in the following sections.

The main aim of this article is to review the articles on autonomous UAV swarms based on AI. The reason for the choice of AI is that these algorithms make it possible to reduce the number of navigation sensors required by each aircraft. AI can infer information from patterns in the data very efficiently, thus reducing the amount of information to be captured [11]. The fewer sensors, the lower the battery consumption. This allows the surplus energy to be used for longer flight times or to add devices that allow the task to be carried out, such as different multi-spectral cameras. Despite of the existence of works that address this Path Planning problem with a single UAV, this article focuses on the swarms because they add a multitude of challenges in which UAVs can perform tasks that individually they could not or would do with difficulty. In contrast to Artificial Intelligence works applied to a single UAV, the use of Artificial Intelligence applied to UAV swarms has emerged recently. In spite of this, the number of works with swarms is multiplying every year due to their increasingly successful applications. It is therefore a good time to analyze the current state of this field and identify the main trends that will develop in the coming years on the basis of the work already developed.

For this article, 39 articles on AI techniques applied to UAV swarms in Path Planning or Mission Planning problems have been reviewed. This review was carried out in two steps: first, a summary of the found articles was made, and they were grouped by techniques; second, a quantitative analysis was made of the evolution over time of the publications found based on the techniques used, the flight environment and the field of use. In the first step, we found the different groups of techniques used and which models or methods are the most common for each group. Finally, the last step is the quantitative study of the publications found. The end result of this stage helps determine the current state of trends in this knowledge area, as well as the application of the techniques examined and the publishers of the articles discovered.

For the selection of the papers, a search was carried out in the main online search engines. For this purpose, an initial set of search terms was defined. In addition, the references in the papers found were reviewed in order to find even more articles. Once the papers were selected, the most relevant and novel ones were selected. A more detailed description can be found in Sect. 3.1.

For a better understanding of the review, a summary of the papers found has been made. This summary is complemented by a quantitative analysis of the papers found according to the method used, the type of flight environment, and the field of use. An analysis of the results of each paper could not be performed due to the lack of standardization.

The main contributions of this paper are as follows:

- A detailed explanation of the many approaches discovered using AI techniques to tackle the Path Planning problem for UAV swarms.
- The articles were then classified broadly based on the AI approaches used: Reinforcement Learning, Evolutionary Computing, Swarm Intelligence and Graph Neural Networks.
- Identification and discussion of the upward trends based on the number of publications over the years.

Apart from this introductory section, the outline of this paper is structured as follows: in Sect. 2, the aspects inherent to the development of Artificial Intelligence algorithms for the control of UAV swarms are explained; in Sect. 3, there is a summary and classification of the found articles; in Sect. 4, the results obtained from the found articles are discussed; in Sect. 5, the conclusions obtained after reviewing the found articles are listed; finally, in Sect. 6, the possible works and research from which the problem to be addressed can be derived are listed.

2 Fundaments

To have a better understanding of all the technical aspects faced by each AI project in autonomous UAV swarms, and to make the reading more comfortable, the following technical aspects are explained: first, what is a UAV; second, what are UAV Swarms; third, the Path Planning problem; fourth, artificial flight environments; and, finally, Path Planning with Swarm Intelligence using Artificial Intelligence.

2.1 Unmanned aerial vehicle

An Unmanned Aerial Vehicle (UAV), commonly known as a drone, is a semi-autonomous aircraft that can be controlled and operated remotely, without an aircrew onboard, by using electronic intelligence and control subsystem [4]. In recent years, UAVs' popularity has increased, and they are widely used in different professional, and recreational applications. UAVs represent one of the most challenging and high-potential tech available nowadays. Initially limited to military uses, they are now expanding into different commercial and industrial sectors [1]. This is due, in particular, to the improvements in technology and power capacities of these vehicles [5, 38].

Their structure, configuration and equipment vary depending on the task to be performed [38]. Having different configurations and equipment implies an improvement in terms of electricity consumption, operation time, and safety risks in the operation. This results in a reduction in costs due to the improvement in the efficiency of the operation. Apart from their original military use [13], other new utilities of the UAVs such as photography, air rescue, or agriculture stand out. However, with the growing popularity and use of drones for consumer applications, the number of incidents involving drones is increasing dramatically [64]. This increase in air accidents is due to the increase in this type of air traffic and the lack of knowledge of the use by some users. Mainly because many UAVs can be acquired without licenses or aptitude tests.

2.2 UAV swarms

Most risky or laborious tasks often require several UAVs. This is due, in particular, to a large amount of time required for operation and the limited autonomy of these small vehicles. When at work, available vehicles assume the function of those that fail. Thus, the task is developed in parallel and the necessary time is shortened compared to when each drone is used one by one.

This strategy is based on the group behavior of natural biological models such as birds or ants [80]. Individuals of these species are able to coordinate and interact with each other when carrying out a task for a common goal, such as flying to warm places or transporting food to their colonies, as well as with their environment. This leads to different groups of swarms being considered to be flocks or herds, depending on the organism.

In Computer Science, Swarm Intelligence (SI) or Swarm Behavior is known as the complex collective, self-organized, coordinated, flexible, and robust behavior of a group of individuals that follows common simple rules [12]. Back in the 1970s, some works about the application of swarm intelligence to small and non-air vehicles can be found, while not until 1990s the UAVs appeared together with the first studies on these devices [67]. It may be due to improvements in the performance of these vehicles and their communications, which speeds up experimentation. The main objective of this experimentation is the ability to achieve algorithms that facilitate navigation and self-organization of a group of UAVs in order to achieve an objective without human interaction.

Robotic swarms are proving their ability to perform certain tasks with respect to cases with a single robot [99]. Especially with UAVs, where each vehicle is part of the assigned task in conjunction with other vehicles perfectly coordinated automatically [14]. In this way, the group is more fault-tolerant, and a shorter execution time is required [84]. This means a significant reduction in costs and operation time.

There were methods previously used to solve path planning problems, especially in individual systems. Algorithms, such as dynamic programming [8] or geometric algorithms like A* search [56], have usually formulated the problem as a heuristic-based numerical cost minimization problem, regardless of computational cost or path correction. During the dynamic programming process, a local cost is assigned to each subdivision of the grid that forms the operation map [6]. It is assumed that the cost of flying over a subzone is independent of the path taken by the UAV to reach the target. Therefore, the cost considered is different from the actual cost [58, 59]. On the other hand, A* algorithm [102], which is a variant of the shortest path algorithm, has difficulties in solving problems with multiple constraints. This type of algorithm is largely based on the cost map, which must be calculated and stored at all times, and the production of the cost map is a time-consuming and error-prone task. All these methods suffer from relatively high execution time. AI methods were proposed as candidates to overcome these problems. These methods use inaccurate and incomplete knowledge and can produce control actions in an adaptive way. This is similar to the inference of knowledge performed by biological systems like humans. In the last decade, an increasing number of studies in the literature have focused on AI methods to

solve path planning problems, with one or multiple vehicles.

2.3 Path planning problem

Path planning is the process of using accumulated sensor data and initial information to allow an autonomous robot to find the best path to reach a goal position. It is a very common problem within the problems with any type of mobile robot, not only UAVs. They are also known as Mission Planning problems. This term is very common within the military. Thus, the term Path Planning is mostly reserved for the civilian field.

It is composed of two main steps: first, compiling all the available information into an effective and appropriate configuration space; and second, using a search algorithm to find the best path in that space [30].

With respect to the first step, there are different types of representation of the flight environment information:

- Cell-maps: this is the most used technique. The map is divided into a set of representative areas known as cells. In those cells, several authors describe the characteristics of the world for each of the cells (elevation, permissibly to fly, etc.).
- Roadmaps: this type of map attempts to describe the world in terms of how to get from a place of origin to a place of destination, taking into account the cost of moving between them. They are much more difficult and time-consuming to create than the previous maps. As an advantage, they are faster to process once created.
- Potential Fields: each UAV is represented as an object under the influence of a field of potentials created by goals and obstacles in the world. These potentials influence the UAV as if it were a physical quantity. This method has most often been used for local obstacle avoidance in mobile robots, but can also contribute to efficient path planning.

At present, it involves a high cost to fly over a real area of the world. In addition, there is a lack of legislation for experimental flights in many countries. Therefore, most authors use artificial flight environments.

The second step is the most critical, as it is responsible for the path calculation and control of the UAVs. Due to the complexity of the problem and the need to automate the process, the most commonly used methods are those based on Artificial Intelligence techniques.

These techniques must be able to overcome as much as possible all the challenges present in this type of problem. These include:

- Path length: the shorter the path, the more optimal it is. If a path is shorter than another and connects the same points, it means that it has fewer loops and fewer curves, making it more energy-efficient.
- Obstacle avoidance: the system must be capable of permiting UAVs to avoid any obstacle that appears during flights. Whether dynamic or fixed.
- Restricted areas: the system must be able to control that UAVs do not fly over restricted areas. Thus, the user is not exposed to legal risk situations
- Fault tolerance: especially in swarms, the system must be able to reorganize the paths of the UAVs in case one fails. Thus, the other UAVs can complete the task.
- Completeness: it is necessary that the system can satisfy a completeness criterion according to the assigned task. If the objective is to map as much terrain as possible, it is of interest that the system searches for the solution that covers the most area of that terrain taking into account the constraints of the UAVs. On the other hand, in tasks such as logistics, it is of interest that the UAVs cover the distance from the warehouse to the recipient in its entirety.
- UAV configuration: the system must be able to adapt the path to the limits of each UAV. That is, depending on factors such as the number of engines, their layout, or their autonomy, the path must be adapted so that the UAV can fly it.
- Other external factors: another challenge is to be able to take into account external factors that influence the trajectory of UAVs. Factors such as wind, birds, rain, or solar storms are obstacles in the paths.

2.4 Artificial flight environments

The development of Swarm Intelligence (SI) with UAV for the Path Planning problem studies involves experimentation with vehicles and robots in real conditions. This is not always possible, due to economic requirements, the need for controlled spaces, or the legislation in force in each country.

Fortunately, more and better artificial flight environments and simulation libraries are being developed. They mimic the limitations and underlying physical forces of UAVs in different environments. In addition, these environments mimic different conditions that a real UAV may face in a real environment.

Among the most used and modern simulators there is Microsoft's AirSim [89]. This simulator is aimed at developing algorithms for autonomous vehicles. To do this, AirSim allows the capture of data from many scenarios in order to train different agents. It can handle multiple drones in real 3D environments. Users can create environments

with countless variables that can be modified, such as the intensity of the wind or its direction. AirSim also has the possibility of handling land vehicles for the development of algorithms for autonomous land vehicles.

One of the most used UAV control libraries is DroneKit [23]. It is an Open Source library made in Python [110], which allows its combination with AI or UAV camera management libraries. It is known for its simple handling and the ability to receive and send large amounts of information to the vehicle. There are other options for commercial UAV like PyParrot [65], made for controlling Parrot UAV. It was developed to teach children STEM concepts, such as programming, in Parrot mini-drones.

2.5 Path planning with swarm intelligence using artificial intelligence

Path planning depends on various factors such as telecommunications [15] or Artificial Intelligence (AI) algorithms [118]. This study is oriented to AI techniques for path planning, so it relies on the second option.

The concept of SI was initially introduced by Gerardo Beni et al. applied to cellular robotic systems [10]. Beni's swarm agents act like AI agents, where they autonomously learn and take action based on an environment [83]. In this way, the agent is able to abstract high-level knowledge without being explicitly programmed. The knowledge is often difficult to represent in its entirety due to its complexity or its wide range of cases. Because of this similarity between SI robots and AI agents, most experts considered SI as a sub-technique of AI.

The main idea is that desired swarm behaviors are not explicitly coded with hierarchical command or control structure but are instead an emergent consequence of the interaction of individual agents with each other and their environment [9]. This kind of algorithm or distributed problem-solving device is inspired by the collective behavior of biological social groups like insect colonies and other animal societies [91]. The agents at the group use simple local rules to govern their actions and via the interactions of the entire group the swarm achieves its objectives [57].

All the agents in a swarm abstract knowledge from information obtained. The great advantage of SI is that agents can be heterogeneous. Therefore, the knowledge abstracted by each agent can be obtained differently. Unlike in a homogeneous swarm, all agents can be trained in different ways to abstract information. For this reason, SI makes use of other techniques of AI to reach its objective.

There are different approaches, all of them based on different AI techniques. In the State of the Art, path planning studies with one or multiple vehicles that use Reinforcement Learning (RL) and Evolutionary Computing (EC) are the most common. For example, Hüttenrauch et al. use Deep RL for controlling groups of agents [45]. On the other hand, Zhao et al. combine EC with other techniques to develop a new SI method [120].

Rules followed by the agents that make up the swarms, and the strategy taken before them, are those that characterize each of the existing types of SI algorithms. There are two main approaches among all the existing ones: the first one makes use of RL-based algorithms; and, the second one of EC-based. We introduce the necessary concepts for ease of understanding of the most common kinds of algorithms.

2.5.1 Reinforcement learning

The first named approach, Reinforcement Learning (RL), is a set of algorithms where the agent must learn behaviors by trial-and-error interactions with a dynamic environment [47, 97, 111]. The goal is to optimize the behavior of the agent with respect to a reward signal that is provided by the environment. The actions of the agent can also affect the environment, complicating the search for the optimal behavior [103].

All RL algorithms follow a common structure. The only difference is the learning strategies. There are several types of these strategies. They all follow different policies that allow them to deal with different problems. The most common types of RL strategies used in SI are explained below.

Q-Learning [108] is one of the most used strategies amongst RL-based algorithms at SI. It follows a model-free strategy [31], so it updates its knowledge following a policy purely by trial-and-error. It is based on off-policy learning, it permits the agents to use their experience for learning the values of all the policies in parallel, even when they can follow only one policy at a time [98]. Q-Learning's classical learning optimal function for computing Q-table values ($Q(s, a)$) is based on Bellman's Equation (Eq. 1).

$$Q(s,a) \leftarrow r + \gamma \times \arg\max_{a'}(Q(s',a')) \tag{1}$$

There are several examples where Q-Learning is used in swarm robotics. For instance, Hung et al. developed an algorithm for controlling flocks of small fixed-wing UAV and tested it in a non-stationary stochastic environment [44]. On the other hand, Rui et al. use Q-Learning to tune the corresponding parameters of a fuzzy multi-UAV formation controller [81].

Using exclusively Bellman's Equation may present abstraction issues in some scenarios. In certain cases, the Q-table values are calculated based on predictions from

Deep Learning models [54]. These models learn from the actions taken, their rewards, and the surrounding environment. This results in a model that is able to abstract more concepts from available data and calculate all future Q values more accurately. This type of algorithm is known as Deep Q-Learning and is part of the well-known Deep RL [68].

Deep Learning [54] is a branch of Machine Learning [66] known for being able to make high-level abstractions automatically. In other words, there is no need for experts to extract characteristics from the data in order for the model to learn them.

The most used and common models used in Deep Learning are known as Artificial Neural Networks (ANN). These networks are large structures formed by connected layers of nodes. Each node is known as an artificial neuron [78]. In Fig. 1 there is an example of the most neuron in an ANN. Neurons perform as summing and nonlinear mapping junctions. The main purpose of ANN is to be able to reproduce some flexibility and power of the human brain by the artificial means [77, 126]. Neural networks applied to Deep Q-Learning are known as Deep Q-Networks.

Deep Q-Learning is the most widely used variant of Q-Learning in UAV swarms. It is also the technique used in the most recent works such as Yijing et al. [116] and Baldazo et al. [7].

Similarly to Q learning, State-Action-Reward-State-Action (SARSA) [82] is a close approach. The key difference is that SARSA is an on-policy learning algorithm [98]. Therefore, that SARSA learns Q-table values based on the action performed by the current policy instead of the greedy policy. This implies that SARSA has constraints over the next action. This is the reason why Q-Learning is utilized less frequently.

Like Q-Learning, there is the Deep SARSA approach [122]. This version with Deep Learning models also shows more flexibility and power of abstraction than its classic version.

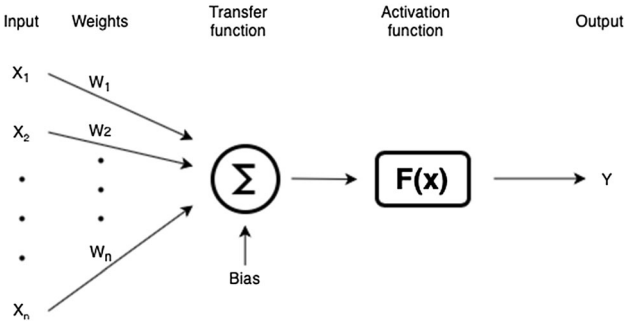


Fig. 1 General schema of an artificial neuron. First, the inputs are multiplied each by its corresponding weight and added with the bias. Then, this result is used as input to the activation function. The output of the neuron is the result of this function

The use of SARSA in UAV and robotics is recent. Most of the publications found show an increasing trend in its use as the years go by. There are SI studies with multiple UAVs like Luo et al. [63] and Speck et al. [93].

2.5.2 Evolutionary computation

The second approach, Evolutionary Computation (EC), is about algorithms based on Charles Darwin's theory of natural evolution. These algorithms try to achieve the best heuristic based on populations and their inheritance from one generation to the next one [20, 92].

The whole process of natural selection begins with the selection of the fittest elements or individuals from an initial population. Combinations of some of them produce offspring [29]. These descendants inherit some characteristics from their parents and will be added to the next generation in addiction with some random probability of mutations and of inheriting some characteristics [94]. If parents have the best fitness, their offspring will be better than their parents and will have a better chance of surviving. The process keeps on repeating over generations until convergence is reached or there are no generations remaining. Finally, a generation with the fittest individuals will be found. This generation will be able to solve the given problem in the most optimal possible way [33, 52].

There are two main domains among all domains in EC: first, Genetic Algorithms (GA) [34]; and, second, Genetic Programming (GP) [50]. The main difference is that Genetic Algorithms use real values-based exploration and GP is an extension of GA tree-based exploration [42, 51].

There is a great variety of work with EC in UAV swarms. This is because it was one of the first approaches tested when applying SI to UAV. It is possible to find work such as Duano et al. [24], Lamon et al. [53] and Gaudiano et al.[28].

2.5.3 Other methods

The methods explained above are the most widely used. This does not imply that other methods are not being exploited with satisfactory results. These include some approaches that are explained below.

Another of the most commonly used methods is pure Swarm Intelligence (SI) based methods. As stated above, these AI methods try to mimic the complex collective, self-organized, coordinated, flexible, and robust behavior of a group of homogeneous or heterogeneous individuals [12].

There are many variations of these methods. The most common are distributed optimization based ones. These techniques are widely used in minimization problems because of their potential. Therefore, they are used to minimize path lengths [113]. Among the most commonly

used in Path Planning are Ant Colony Optimization [21] and Particle Swarm Optimization [48].

To coordinate these swarms it is necessary to employ communication mechanisms between group members. More biologically puristic approaches employed mechanisms that mimicked communication by means of odor or pheromones.

The most recent publications are based mostly on pheromones. These techniques are known as virtual pheromones based methods [73]. In this way, the agents employ mechanisms that imitate the pheromones used by insects such as ants.

The use of pheromones to coordinate and interact with the environment is known as stigmergy [100]. The concept was first introduced by Grassé when he observed interactions in two species of termites: *Bellicositermes Natalensis* and *Cubitermes* [36].

In this variant of SI, the work of Parunak et al. [72] is very relevant, although new approaches are emerging.

Classic Deep Learning [54] effectively captures hidden patterns of Euclidean data, like images or text recognition, but there is an increasing number of applications where data are represented in the form of graphs [112], like molecules or proteomics. Graph Neural Networks extend existing neural network methods for processing the data represented in graph domain[87].

Unlike the data used in classical Deep Learning models, graphs do not have a defined structure. A node on a graph may have no connections or many connections, which may not be directed. Graphs in a data set can have a varying number of nodes and edges arranged in different ways. Based on their different distributions, graphs can be acyclic, cyclic, set, or unset. In general, it makes the data handling process more computationally expensive.

In this discipline, it is possible to find some of the most recent work. Among them are those of Li et al. [55] and Tolstaya et al. [101].

3 Artificial intelligence applied to path planning in UAV swarms

Being a dynamic and relatively new knowledge field, it is difficult to identify which works are related to and to identify the future challenges that this newborn area should tackle in the upcoming years. In this section, an analysis is proposed based on the 39 works found from 2016 to 2021. The last 5 years have been chosen because they are a close time period that can sufficiently indicate the current trend of the field. In this analysis, articles are grouped by the type of AI technique used for making reading easier.

3.1 Methodology

The well-known online tools Google Scholar [35], Scopus [88], Web of Science [109], IEEE Xplore [46] and arXiv [3] were used to obtain the articles related to the topic. In them, searches were performed for the terms listed in Table 1.

The references of the works found are also reviewed. Thus, more relevant ones can be found that have not appeared in the search tools. Then, the whole set of articles found is selected by the year. In this way, the most recent and relevant ones are found.

3.2 Content review

This sections presents a summary of the main points of the articles found and selected. For ease of reading, they have been grouped by the technique used.

3.2.1 Reinforcement learning

Starting with the discipline of Q-Learning, Hung et al. manage fixed-wing UAV flock in which there are a leader and a set of followers [44]. Thus, it gets groups of autonomous UAV that move in a synchronized way, similar to a flock of birds. Using a leading aircraft improves the computation time since it is important to improve the leading path and the others would be derived from it. Nevertheless, in case of failure of the leading UAV, it would be more expensive to recalculate all the paths. If there was no dependence between paths, only the UAVs closest to the fallen aircraft would be affected. The use of fixed-wing UAVs greatly limits their application due to their more complex control and lower stationary flight capability. Khalil et al. succeed in making a multi-agent system by improving the classical Q-Learning, which they call economic Q-Learning [49]. In their system, they copied the decision techniques used in Economic Theory. In the described technique, it is considered that what is most chosen is what is most useful and frequent in the future.

There are variations of the algorithm, like the work of Hafez et al. where Fuzzy Systems with Q-Learning are combined for control of UAV swarms for military use [37]. Their method shows robustness to failure. In this way, it can be recovered in the event of a UAV falling. In this article, the UAVs have to maintain a formation, which conditions the computation of the paths. Moreover, they were tested in closed indoor environments, so they do not contemplate changes in the wind or dynamic flying obstacles such as birds. Also, combined with Fuzzy Logic, Su et al. make use of fuzzy matrices as a reward function to recalculate paths of groups of drones [96]. In this case, the

Table 1 Table with search terms grouped by category

Category	Terms
Vehicles	Unmanned aerial vehicle, UAV, aircraft, drone, remotely piloted aircraft, RPA
Group of vehicles	Multiple, multi, swarm, group, flock, formation, collaborative
Technique	Artificial intelligence, swarm intelligence, reinforcement learning, evolutionary computation, Q-learning, SARSA, artificial neural network, ANN, genetic programming, genetic algorithm, particle swarm optimization, PSO, ant colony optimization, ACO
Problem	Path planning, mission planing, mission control, autonomous flight, autonomous control, navigation
Field of application	Civilian, agriculture, emergency, forestry, military, surveillance, photography, filming

use of clustering techniques for the initial distribution of the land makes it dependent on the initialization parameters. Therefore, a good study of the parameters should be made so that they can be used in a variety of real environments. Continuing with fuzzy computing, in 2020, Yang et al. [114] also propose to combine it with Q-Learning. A very important point in their project is that it is one of the few that take into consideration the battery level. Also in 2020, Chen et al. [18] propose a multi-agent Q-Learning system based on constrained actions. Thus, they facilitate autonomous in-flight decision-making by taking into account the uncertainty of the location of each landmark. Their system was tested with a different number of UAVs. They showed that as the number of UAVs increases, the task failure rate increases.

The best known variation of Q-Learning is DQN, because of its power of generalization and its proposed professional applications. One of the most important approaches is the one proposed by Roudneshin et al. in which they perform ANN to control swarms composed of UAVs and heterogeneous robots [79]. This work of military nature does not present a work purely in UAVs but adds terrestrial robots. However, this is a problem of swarm path planning with greater difficulty than using only UAVs. This increase in the difficulty of the problem is due to the different limitations presented by air and land vehicles. Thus, a land vehicle can encounter non-geographic obstacles and has more limited movements. As a practical utility, they expose the capabilities for use in search and rescue missions. Also in emergency or rescue conditions, Baldazo et al. propose a DQN model to coordinate multiple UAV for flood monitoring and minimize damage costs [7]. This paper has a very good choice of the type of UAV when it comes to flood monitoring. Fixed-wing UAVs are the most efficient solution for long-distance travel because of their higher speed. As they have less stationary flight capability than other configurations, fixed-wing UAVs require smooth flight paths, without sudden changes. If applied in real scenarios, the paths calculated should have mechanisms to smooth out the curves due to possible abrupt changes in case of obstacles. Later, in 2020, Zhao et al. [119] proposed a variation of Deep Q-Learning known as Wire Fitting Neural Network Q (WFNNQ) learning. Combining this technique with hill-climbing algorithms successfully creates smoothed flight paths in simulated environments. Despite the computational cost, his system avoids having to perform a final phase of path smoothing. Venturini et al. also created a system capable of controlling multiple UAVs using DQN techniques. In 2020 proposed a system capable of operating on square cell maps [104]. In 2021 the maps were simulations of real maps [105]. While it is a project that demonstrates capabilities to operate on different maps, it is necessary to establish targets to direct the paths. Therefore, it is very dependant on the initialization.

Goh et al. designed a DQN model with Convolutional Recurrent Neural Network [32]. In this way, they could control multiple UAVs to pursue the target. The most remarkable thing about their project is the freedom of movement of the UAVs. For added visibility, their system has been tested in the AirSim simulator.

On the other hand, with the SARSA algorithm less recent approximations were found. However, they show satisfactory results in different cases. Luo et al. tested their Deep-SARSA algorithm in dynamic environments, where the obstacles can change [63]. The paper offers an efficient method in dynamic environments. This demonstrates its ability in changing environments, which reinforces its usefulness in the real world. However, the model requires a pretraining phase, which may limit its deployment in novel environments due to the time required for pretraining. Speck et al. combine object-focused learning with the SARSA algorithm in order to improve the algorithm itself [93]. This paper presents a very efficient decentralized approach in terms of generalization. The capacity for generalization may be limited when dealing with fixed-

wing UAVs for the same reasons as the papers cited above. Thus, the configuration of the UAV limits the range of application of the system to cases where it is optimal to use fixed-wing UAVs. The paper written by Zhao et al. shows a new method for the coordination of UAV swarms in mesh networks [121]. These networks are very important in disaster areas to maintain communications. In this way, their approach contemplates the limitations of communications in these cases. Despite contemplating such limitations, mesh networks cannot always be deployed if the environment is rugged or very difficult to access. Therefore, consideration should be given to limiting the number of paths needed to make it as viable as possible.

3.2.2 Evolutionary computation

In the field of EC, another huge volume of papers is available. For example, Sathyan et al. combine GA with Fuzzy logic to improve accuracy during path planning [86]. The paper approaches the problem from a very interesting point of view, as they interpret the paths as polygons. Thus, they quickly solve the problem of each UAV returning to the starting point at the end of the operation as if it were part of the path itself. The main drawback of this article is that they do not take into account fuel consumption or possible collisions. Thus, paths can have great lengths or abrupt changes of direction that the range cannot support. In addition, paths can be so close that UAVs can collide.

Ramirez et al. use, in some of their works, variations of the Multi-objective Genetic Algorithm (MOGA) for mission planning with multiple UAVs [75, 76]. In their work, they carry out an exhaustive evaluation of their system and show the evolution of the results as the complexity increases. In both works, there is a lack of detail in the description of the data sets they use, so it is vague whether these changes in complexity are correctly interpreted. Cekmez et al. find control points in the terrain by using K-means clustering [16]. Then, a parallel genetic algorithm solves the multi-UAV path planning problem of each subset of control points. The advantage of this genetic algorithm is its implementation on CUDA, which allows for faster experimentation. The use of K-means clustering can be limiting for area partitioning. Many clustering algorithms, such as K-means or K-medians, are known to be strongly dependent on the initialization parameters. Therefore, this partitioning should be tested in a huge amount of different environments until satisfactory parameters are achieved.

There are also approaches such as the one made by San et al. [85], in which genetic algorithms of chromosomes with multidimensional genes are used. In this paper, the shortest possible path is computed, as it is for parcel UAVs. For this purpose, two fitness functions are considered, one for the weight of the load and another for the path, which achieves great results. UAVs tend to consume more battery power if they need to be constantly stabilized, so they should consider the oscillation of the load during the flight to minimize the battery. Otherwise, a heavy object with many oscillations increases battery consumption because the aircraft needs to be constantly correcting its trajectory. Liu et al. employ Genetic Algorithms to adjust ANN for flight path generation [61]. Relying only on the ANN for path computation makes it dependent on more parameters than weights. Therefore, other parameters such as learning rates or adjusting the architecture of the ANN should be adjusted. Duan et al. also improve a genetic algorithm, in this case with a local search algorithm. To do so, they combine a memetic algorithm with the VND search algorithm [25]. In their work, an initial individual is generated based on the heuristics of the nearest neighbors and the other initial individuals are configured as random. Using the closest neighbors can greatly limit the generation of individuals. Especially if there are many equally close neighbors. In that case, a criterion should only be established to determine whether the individual is a member of a group.

Cimino et al. employ Differential Evolution for UAV swarms to detect targets collaboratively [19]. The major difference between Differential Evolution compared to other Evolutionary Computing algorithms, such as GA, is that it depends more on the mutation operator [95] than on the crossover operator. Thus, a descendant can be the exclusive mutation of a parent. Having less dependence on one type of operator than the other makes it more difficult to find new individuals in the population. Therefore, it can be more expensive to find the optimal path. In the work of Zhou et al. multiple UAVs are made to fly over a portion of terrain in the presence of dynamic targets. For this, they make use of the Immune Genetic Algorithm (IGA) [125]. The drawback of their method is the need for path smoothing.

Olson et al. also designed GA for multi-UAV systems [70]. In their case, they seek to create 3D maps using multiple UAVs. To do this they simplify the flight map to a 2D map. Once created, their system searches for paths that maximize coverage and reduce flight time. The use of flight time in GA is also used by other authors, such as Huang et al. [43]. In their paper, they take into account the time taken by each UAV to find a target. A great point to note in their work is that it is one of the few that take into account the attributes of the UAVs. Other authors take into account the flight time of the entire swarm depending on the task to be performed. As in the case of Ye et al. where they seek to minimize the overall flight time of the swarm [115]. Thus, it may be more efficient in global terms to minimize the time of one UAV even if the time of another is not

minimized. Time is calculated using Dubin's car model. This model usually refers to the shortest curve connecting two points in the two-dimensional Euclidean plane. It may not be the most optimal way to obtain UAV paths and times because it only considers curves.

In 2021, Pan et al. combined GAs with Deep Learning to compute optimal paths for multiple UAVs to capture data from multiple nodes [71]. Through this combination, they improve the results concerning using purely GAs in case of having numerous nodes.

3.2.3 Swarm intelligence based methods

Among the proposals of SI for this type of problem is a great variety of algorithms. Cekmez et al. make use of Ant Colony Optimization (ACO) for planning optimal UAV paths while avoiding complex obstacles such as radars [17]. In their paper, they implement a version of the algorithm for GPUs allowing them to perform more iterations of that algorithm at the same time. This allows getting closer to the optimal solution. They consider constant flight speed, so the curves to be made for each UAV may not be the most efficient. Perez-Carabaza et al. also use this technique to plan flight paths so that multiple UAVs can find targets in unknown environments in the minimum possible time [74]. The use of its heuristics is very accurate, because of the speed of computation. In addition, correctly defined heuristics can reduce the computational cost. As the authors state, paths should be smoothed or it would be limited to a certain number of UAV types. Another approach to this technique is its use in cooperative search-attack mission planning for multiple UAVs [123]. These types of problems are very similar to path planning. In these cases, it is usually a matter of finding a target and getting closer to attack. In particular, they tend to face more changes in paths because the targets frequently change. In this work, they also consider constant flight speeds. If they are high speeds, plotted curves may not be feasible. Following ACO, Zhen et al. proposed a distributed version of ACO in 2020. A respectable aspect of their paper is that their system is one of the few that considers flight range constraints among all the constraints considered [124].

Vijayakumari et al. make use of another well-known SI technique known as Particle Swarm Optimization (PSO) for optimal control of multiple UAVs in a decentralized way [106]. In their work, they manage to simplify the computation of the problem by means of discretization. For collision avoidance, they rely on distances. Although this is a dynamic variable, in certain types of non-stationary flight UAVs, such as fixed-wing UAVs, it does not guarantee collision avoidance. In these cases, a metric that predicts the state of the UAV and the obstacle in future instants is of interest and thus makes a decision. Otherwise, the UAV would continue to move forward while the decision is being computed. Li et al. also use this technique for UAV swarm control and demonstrated the effectiveness of the results in several terrains at Shaanxi province in China [62]. It is one of the few found studies applied to agricultural UAVs that have been tested in real field simulations. It is a great indicator of the project's viability and potential. The paths shown in the images have abrupt changes in direction, so the path should be smoothed. Otherwise, some UAVs would not be able to take the bends. More recently, there is work such as that of Hoang et al. in which they employ a variation of PSO known as Angle-Encoded PSO for the planning of flight paths in UAV swarms [41]. The main advantage of the proposed model is that it considers flying height. Most works consider 2D flights where UAVs do not need to vary their height. The 2D flight does not guarantee that the path traced in the presence of obstacles is optimal. In many cases, a sudden change of direction can be avoided by varying the height. The paper uses waypoints to assist the path. This makes the model very dependent on the initialization of the waypoints. Also, with an improved version of PSO, Shao et al. proposed the coordination of multiple UAVs by comprehensively improved PSO [90]. In this type of PSO, parameter tuning is done adaptively. Thus, the parameters are better tuned than in classical versions of PSO. In March 2021, He et al. proposed their improvement of PSO for cooperative UAV systems on 3D maps [40]. Despite good results, the paths need to be smoothed and UAV formations should be fixed.

In 2020, Wang et al. proposed a Path Planning system for multiple UAVs using the Pidgeon-Inspired Optimization algorithm [107]. The main point that makes it distinguishable from the SI papers described above is that, unlike the others in general use, it employs a specific SI algorithm for path planning with aerial robots [26]. Another algorithm different from those mentioned above is the Bean Robot Optimization Algorithm used in UAV swarms for target searching by Zhang et al. [117]. The algorithm takes into account the free-moving space of individual UAVs and adds a free-space search mechanism to improve target search efficiency.

3.2.4 Graph neural networks

Finally, in the newest technique, Graph Neural Networks, a single article was found. In it, Li et al. [55] make use of these neural networks for path computation in robotic systems. Thus, they achieve more capacity for generalization in the face of new cases than other more widely used techniques. Since we are dealing with two ANNs, previous training is necessary in different and very varied cases.

Table 2 Summary of works where artificial intelligence methods are applied to path planning in UAV swarms

Publication	Technique	Year	Flight environment
Hung et al. [44]	RL	2016	Artificial environment
Khalil et al. [49]	RL	2021	Artificial environment
Hafez et al. [37]	RL	201	Real environment
Su et al. [96]	RL	2016	Artificial environment
Yang et al. [114]	RL	2020	Artificial environment
Chen et al. [18]	RL	2020	Artificial environment
Roudneshin et al. [79]	RL	2019	Artificial environment
Baldazo et al. [7]	RL	2019	Artificial environment
Luo et al. [63]	RL	2018	Artificial environment
Speck et al. [93]	RL	2018	Artificial environment
Zhao et al. [119]	RL	2020	Artificial environment
Venturini et al. [104]	RL	2020	Artificial environment
Venturini et al. [105]	RL	2021	Artificial simulation over a real environment
Goh et al. [32]	RL	2021	Artificial environment
Zhao et al. [121]	RL	2019	Artificial environment
Sathyan et al. [86]	EC	2016	Artificial environment
Ramirez et al. [75]	EC	2017	Artificial environment
Ramirez et al. [76]	EC	2017	Artificial environment.
Cekmez et al. [16]	EC	2016	Artificial environment
San et al. [85]	EC	2016	Artificial environment
Liu et al. [61]	EC	2019	Artificial environment
Duan et al. [25]	EC	2018	Artificial simulation over a real environment
Cimino et al. [19]	EC	2016	Artificial environment
Zhuo et al. [125]	EC	2020	Artificial environment
Olson et al. [70]	EC	2020	Artificial environment
Huang et al. [43]	EC	2020	Artificial environment
Ye et al. [115]	EC	2020	Artificial environment
Pan et al. [71]	EC	2021	Artificial environment
Cekmez et al. [17]	SI	2018	Artificial environment
Perez-Carabaza et al. [74]	SI	2018	Artificial environment
Zhen et al. [123]	SI	2018	Artificial environment
Zhen et al. [124]	SI	2020	Artificial environment
Vijayakumari et al. [106]	SI	2019	Artificial environment
Li et al. [62]	SI	2016	Artificial simulation over a real environment
Hoang et al. [41]	SI	2019	Real environment
Saho et al. [90]	SI	2020	Artificial environment
Wang et al. [107]	SI	2020	Artificial environment
Zhang et al.[117]	SI	2020	Artificial environment
Li et al. [55]	GNN	2019	Artificial environment

Otherwise, ANNs may be overfitted in several flight areas and swarm structures.

A summary of the publications cited above is shown in Table 2.

3.3 Bibliometric analysis

For the bibliographic analysis, the number of publications, the number of publishers, and their evolution over the last 6 years will be taken into account. These three factors, along with their relationships, provide quite a bit of information on how UAV swarm AI applications are doing for path planning and mission control problems.

The first chosen graph (Fig. 2), the evolution of the number of publications in the last 6 years, shows the interest in the subject and the evolution of the field based on the State of the Art. The Fig. 2 shows a decline in the publications found over the years until 2018. This

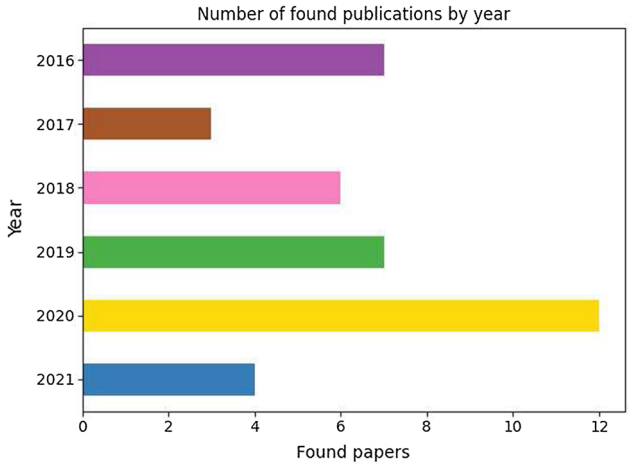


Fig. 2 Relevant and novel publications found per year. The bar of the year 2021 is the number of papers of the first quarter

coincides with the regularization of legislation in many countries, facilitating development in the field. For example, in Europe, EASA regulated the situation in 2018 by establishing the basis for all member countries [27]. Having a solid and current legislation applied to UAVs favors development and innovation in these aircraft. Being able to conduct experiments in a safe and controlled manner by having guidelines increases confidence in research and reduces fear of legal consequences due to uncertainty. In 2020, a large number of articles have been found, thus reinforcing the growing trend in the number of UAV projects. In 2021 quite a few publications were found considering that they are only those belonging to the first quarter.

As mentioned above, RL and EC are among the most widely used techniques. Figure 3 shows how RL outperforms EC, but they are still the most widely used techniques. The reduction in the cost of computational resources means that more and more authors are opting for these more expensive but more efficient methods compared to SI techniques. Taking into account the scope of use of the systems proposed in the papers, civil publications use different cited techniques in a variety of ways (Fig. 4). In spite of this, RL and EC continue to be the most widely used techniques. In general terms, they are always the most used regardless of the purpose.

In 2019, the most used technique was RL (Fig. 5). Its evolution contrasts sharply with 2016 when it was in the minority. Unlike EC, its popularity in this type of problem has been increasing. This change in trend may be due to the normally lower computational cost of the RL and its greater ease of development. On the other hand, an equal number of EC, RL, and SI papers were found in 2020. The elevated number of publications shows indications of the high impact of UAV swarms. Thus, each year seems to be increasing.

Figure 6 shows that most of the studies found are for civil purposes. Years ago, most articles were for military purposes or rescue operations. The change in purpose reinforces the fact that these aircraft found a niche in civil functions and operations. Studies applied to non-civil purposes remain constant and scarce over the years. On the other hand, the number of studies on civil purposes found is much higher. The decrease in 2018 may be caused by regulatory changes in many countries. These changes often bring uncertainty and loopholes that are corrected later. These corrections may explain the increase, again, of publications in 2018.

Figure 7 shows that most studies use artificial environments. This may be due to the difficulty in reserving airspace for experimentation. In many countries, these requests are expensive and take a long time to confirm. Publications for non-civil purposes are those that make the most use of real flight environments. Normally, military authorities in countries usually have airspace reserved for

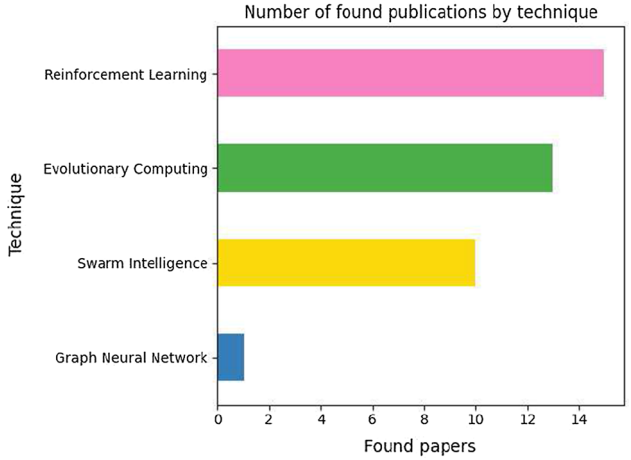


Fig. 3 Relevant and novel publications found per technique

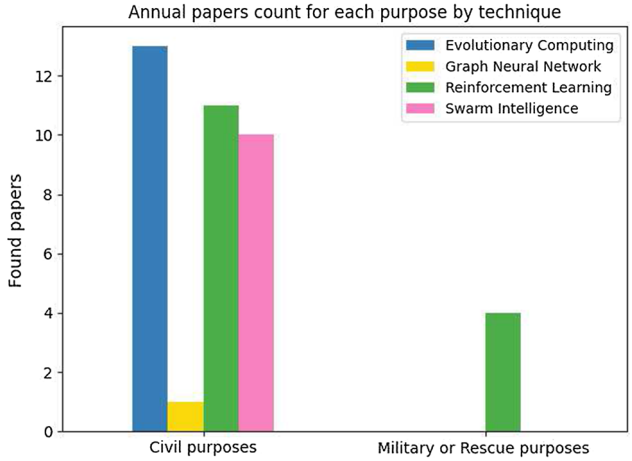


Fig. 4 Publications per technique for each different purpose

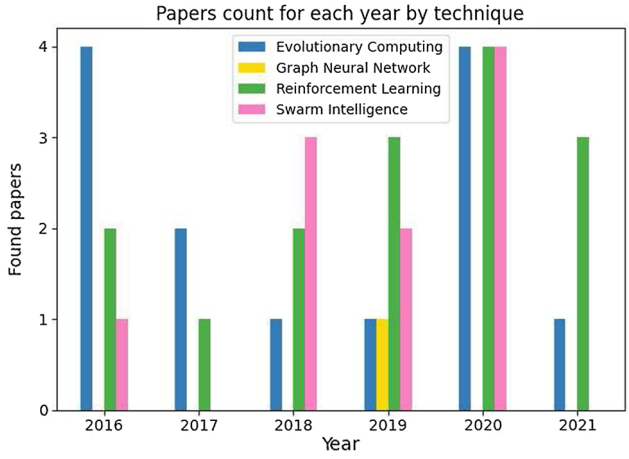


Fig. 5 Evolution of publications per technique for each year

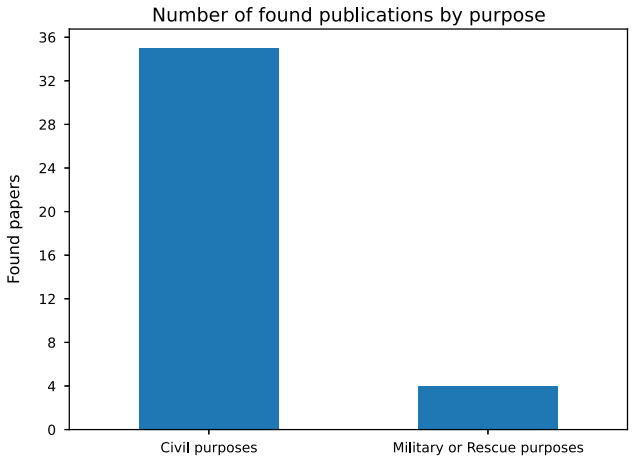


Fig. 6 Publications per purpose

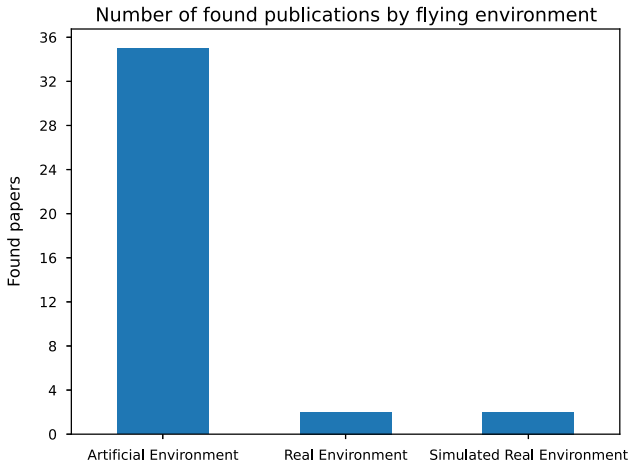


Fig. 7 Publications per flight environment

their flights. In addition, they are more likely to reserve airspace when necessary. Quite a few publications use simulations of real locations. In these cases, they map real environments and then simulate them virtually. In this way, there is no need to reserve the flight area, but the mapping is often expensive.

4 Discussion

The development of systems for Path Planning with UAVs is a common problem, but it is in the early stages. Despite this, there are more and more applications and studies of their use at the professional and domestic levels. One of its most novel applications of systems for Path Planning is in UAV swarms. Thus, costs and operation time can be reduced by having several aircraft operating at the same time in a coordinated manner. To assist in the coordination of the swarms, more and more authors are making use of AI techniques, which is the focus of this review.

One of the factors triggering this boom is the decrease in their market price and the regulation of the laws concerning their use. Consequently, more and more people can access them and have their airspace reserved. This facilitates their use for developing activities and tests with them. In Fig. 2 this increase in the last few years is shown, as 2018 is one of the years with the most changes in the law. Despite this, more papers are being published every year. In 2020, this growth will be even more accelerated. In the first quarter of the year 2021, there is a significant number of papers, which may indicate that in 2021 there will numerous papers. It could even surpass the year 2020.

With respect to the techniques reviewed, RL and EC are the main ones in the number of publications (Fig. 3). Many of these articles may present the use of these techniques because of tradition and because they are more developed. Other techniques such as SI usually present ad-hoc methods or a great diversity of different methods. However, the most commonly used are distributed optimization based ones because of their ability to minimize the length of the paths. GNN is the technique with the fewest publications, being the only one in pre-published status. As it is a very new technique, few studies are sufficiently advanced to be published no matter the discipline in which they are applied.

Over the last 6 years, it seems there has been a change in the trend. The techniques of EC present fewer publications, while those of RL are on the rise (Fig. 5). This may be due to different factors such as the generally lower computational cost for RL techniques or the fact that many RL articles have yet to be published in the field.

Most of the publications found are for civil purposes. This may be because they are becoming more accessible to the public. As a result, these aircraft can be used in a wider range of sectors and tasks.

Artificial environments are the most used among civil-purpose publications. It may be caused by the difficulty in reserving airspace for testing. On the other hand, publications for non-civil purposes have more facilities for this, so they are usually tested in real environments.

In general terms, few publications have been found on the subject of the study. This is due to the novelty of the problem. In other words, the advantages of the use of UAV swarms are still beginning to be perceived.

The results achieved in the reviewed papers cannot be compared. In the few cases where it is possible to compare them, it is almost impossible to obtain a meaningful interpretation of the comparison. This issue involves multiple factors such as the variables to be taken into account or the type of Path Planning problem to be solved.

The most important factor is the lack of common evaluation methods to communicate the results and demonstrate the goodness of the methods. This seems to be a fairly common factor in new research areas. In this situation, the authors of new contributions are not sufficiently informed or do not have access to sufficient previous work. This leads to a lack of information which, in turn, causes authors to opt for different approaches to communicating results. Some of them are the time consumed, the length of the paths or the number of solutions found by the system. Nevertheless, with the summary and classification of the papers found, together with the proposed figures, an attempt is made to provide as objective a review as possible of the most recent and novel projects.

As a final summary, the lack of standardization of the results together with the growing number of studies reinforces the idea that this is an increasingly important field of research. The most commonly used methods are RL and EC. This convergence may be limiting in the development of new systems, as there is less innovation in other different and possibly more promising methods. There are more and more applications in the civilian field, mainly characterized by the use of artificial flight environments. The use in non-real environments can be limiting since in real environments, there are usually more obstacles and external factors than many authors consider.

5 Conclusion

AI techniques applied to Path Planning problems with UAV swarms are booming and continuously developing. The increasing use of AI techniques in UAV swarms for Path Planning problems over the years may be an objective indicator of it. More and more papers are being published. Even in 2021, there may be many publications due to the already high number published in the first quarter. Moreover, in the quantitative analysis, it can be seen that RL and EC are the most used methods regardless of the domain. To test these methods, mostly artificial flight environments are used. Therefore, many of these methods may have difficulties operating in real environments due to the large number of external elements that may affect the UAV.

As these are novel systems that use AI for the control of UAV swarms, there are still shortcomings. Especially the lack of standardization of the results. As each paper focuses on a different aspect of Path Planning, each one focuses on a different variable. This can be limiting in the development of new systems due to the lack of criteria to evaluate which approach is better. On the other hand, it is indicative of this being a novel topic. In addition, it may also be indicative that Path Planning problems should be divided into subproblems, each focusing on its variable of interest. Thus, there would be branches that would try to find the solutions with the shortest flight time, another where the solutions involve the routes with the fewest number of turns, etc.

In conclusion, the low but growing number of publications shows that this is a recent problem. The late emergence of UAV swarms coincides with the late incorporation of UAVs in non-military fields. Being more accessible and cheaper allows the public to experiment with them, finding more possible fields of application.

6 Future work

Based on the graphs shown it can be understood that the use of AI techniques for UAV swarms in path planning problems is growing. This growth will be greater as countries adapt their laws to swarms of autonomous vehicles. Other sectors such as self-driving cars will also contribute to this increase with studies that can also be taken to the world of UAVs.

As there is an increase in UAV swarm works and studies more sectors will be able to benefit from them. In addition, other new fields within the sectors are appearing. For example, the 3D animation sector as a substitute for fireworks has emerged in the recreational sector.

With only one article on the GNN technique and in a pre-publish status, a new research path is opened in the domain. The existence of a single paper demonstrating the possibility of the use of GNN in UAV swarms encourages many researchers to take it as a starting point for their research.

The change of tendency experienced in the papers found of RL and EC indicates that the majority of possible works will be of RL. This is not a definitive statement, since it may be more about fashion than about improving results. Therefore, many future works may end up combining both techniques, just as it is used in swarms of other robotic

systems. On the other hand, in 2020 there have been a large number of SI articles, so in 2021 there may also be a large number of them.

Finally, improvements in swarming other types of vehicles and improvements in UAV navigation to require fewer sensors may work together. In this way, information collected on the paths of other vehicles, such as autonomous aircraft, can benefit the computation of UAV paths. And vice versa, information collected from UAV paths can complement the computation of paths for other vehicles such as avoiding congestion in self-driving cars.

Funding This work is supported by Instituto de Salud Carlos III, grant number PI17/01826 (Collaborative Project in Genomic Data Integration (CICLOGEN) funded by the Instituto de Salud Carlos III from the Spanish National plan for Scientific and Technical Research and Innovation 2013–2016 and the European Regional Development Funds (FEDER)—"A way to build Europe.". This project was also supported by the General Directorate of Culture, Education and University Management of Xunta de Galicia ED431D 2017/16 and "Drug Discovery Galician Network" Ref. ED431G/01 and the "Galician Network for Colorectal Cancer Research" (Ref. ED431D 2017/23). This work was also funded by the grant for the consolidation and structuring of competitive research units (ED431C 2018/49) from the General Directorate of Culture, Education and University Management of Xunta de Galicia, and the CYTED network (PCI2018_093284) funded by the Spanish Ministry of Ministry of Innovation and Science. This project was also supported by the General Directorate of Culture, Education and University Management of Xunta de Galicia "PRACTICUM DIRECT" Ref. IN845D-2020/03.

Availability of data and material Not applicable

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Code availability Not applicable

References

1. Akhloufi MA, Arola S, Bonnet A (2019) Drones chasing drones: reinforcement learning and deep search area proposal. Drones 3(3):58
2. Albani D, IJsselmuiden J, Haken R, Trianni V (2017) Monitoring and mapping with robot swarms for agricultural applications. In: 2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS). IEEE, pp 1–6
3. arxiv. https://arxiv.org/. Accessed 24 Mar 2021
4. Austin R (2011) Unmanned aircraft systems: UAVS design, development and deployment, vol 54. Wiley, London
5. Bachmann RJ, Boria FJ, Vaidyanathan R, Ifju PG, Quinn RD (2009) A biologically inspired micro-vehicle capable of aerial and terrestrial locomotion. Mech Mach Theory 44(3):513–526
6. Bakker B, Zivkovic Z, Krose B (2005) Hierarchical dynamic programming for robot path planning. In: 2005 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 2756–2761
7. Baldazo D, Parras J, Zazo S (2019) Decentralized multi-agent deep reinforcement learning in swarms of drones for flood monitoring. In: 2019 27th European signal processing conference (EUSIPCO). IEEE, pp 1–5
8. Bauso D, Giarré L, Pesenti R (2004) Multiple uav cooperative path planning via neuro-dynamic programming. In: 2004 43rd IEEE conference on decision and control (CDC) (IEEE Cat. No. 04CH37601), vol 1. IEEE, pp 1087–1092
9. Beni G (2004) From swarm intelligence to swarm robotics. In: International workshop on swarm robotics. Springer, pp 1–9
10. Beni G, Wang J (1993) Swarm intelligence in cellular robotic systems. In: Robots and biological systems: towards a new bionics?. Springer, pp 703–712
11. Bishop CM (2006) Pattern recognition. Mach. Learn. 128(9)
12. Bonabeau E, Meyer C (2001) Swarm intelligence: a whole new way to think about business. Harv Bus Rev 79(5):106–115
13. Buckley J (2006) Air power in the age of total war. Routledge, London
14. Bürkle A, Segor F, Kollmann M (2011) Towards autonomous micro uav swarms. J Intell Robot Syst 61(1–4):339–353
15. Campion M, Ranganathan P, Faruque S (2018) A review and future directions of uav swarm communication architectures. In: 2018 IEEE international conference on electro/information technology (EIT). IEEE, pp 0903–0908
16. Cekmez U, Ozsiginan M, Sahingoz OK (2016) Multi-uav path planning with parallel genetic algorithms on cuda architecture. In: Proceedings of the 2016 on genetic and evolutionary computation conference companion. ACM, pp 1079–1086
17. Cekmez U, Ozsiginan M, Sahingoz OK (2017) Multi-uav path planning with multi colony ant optimization. In: International conference on intelligent systems design and applications. Springer, pp 407–417
18. Chen YJ, Chang DK, Zhang C (2020) Autonomous tracking using a swarm of uavs: a constrained multi-agent reinforcement learning approach. IEEE Trans Veh Technol 69(11):13702–13717
19. Cimino MG, Lazzeri A, Vaglini G (2016) Using differential evolution to improve pheromone-based coordination of swarms of drones for collaborative target detection. In: ICPRAM, pp 605–610
20. Davis L (1991) Handbook of genetic algorithms
21. Dorigo M, Bonabeau E, Theraulaz G (2000) Ant algorithms and stigmergy. Futur Gener Comput Syst 16(8):851–871
22. droneblog: LED equipped drones that can "draw" three-dimensional figures in midair|Droneblog. https://www.droneblog.com/2014/09/26/led-equipped-drones-that-can-draw-three-dimensional-figures-in-midair/ (2014). Accessed 24 Mar 2021
23. DroneKit: DroneKit. https://dronekit.io (2021). Accessed 24 Mar 2021
24. Duan H, Luo Q, Shi Y, Ma G (2013) Hybrid particle swarm optimization and genetic algorithm for multi-uav formation reconfiguration. IEEE Comput Intell Mag 8(3):16–27
25. Duan F, Li X, Zhao Y (2018) Express uav swarm path planning with vnd enhanced memetic algorithm. In: Proceedings of the 2018 international conference on computing and data engineering. ACM, pp 93–97
26. Duan H, Qiao P (2014) Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. Int J Intell Comput Cybern 7(1):24–37
27. EASA: Regulations | EASA. https://www.easa.europa.eu/regulations#regulations-uas—unmanned-aircraft-systems (2021). Accessed 24 Mar 2021
28. Gaudiano P, Bonabeau E, Shargel B (2005) Evolving behaviors for a swarm of unmanned air vehicles. In: Proceedings 2005

IEEE Swarm intelligence symposium, 2005. SIS 2005. IEEE, pp 317–324
29. Gestal Pose M (2010) Soft computing methods for practical environment solutions: techniques and studies: techniques and Studies. IGI Global, New York
30. Giesbrecht J (2004) Global path planning for unmanned ground vehicles. Technical report. Defence Research and Development Suffield (Alberta)
31. Gläscher J, Daw N, Dayan P, O'Doherty JP (2010) States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning. Neuron 66(4):585–595
32. Goh KC, Ng RB, Wong YK, Ho NJ, Chua MC (2021) Aerial filming with synchronized drones using reinforcement learning. Multimed Tools Appl 80:1–26
33. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning, Addison Wesley, reading, ma. Summary the applications of GA-genetic algorithm for dealing with some optimal calculations in economics
34. Goldberg DE (2006) Genetic algorithms. Pearson Education India, New York
35. Google scholar. https://scholar.google.com/. Accessed 24 Mar 2021
36. Grassé PP (1959) La reconstruction du nid et les coordinations interindividuelles chezbellicositermes natalensis etcubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. Insectes sociaux 6(1):41–80
37. Hafez AT, Givigi SN, Yousefi S, Iskandarani M (2017) Multi-uav tactic switching via model predictive control and fuzzy q-learning. J Eng Sci Mil Technol 1(2):44–57
38. Hassanalian M, Khaki H, Khosravi M (2015) A new method for design of fixed wing micro air vehicle. Proc Inst Mech Eng Part G J Aerosp Eng 229(5):837–850
39. Hayat S, Yanmaz E, Muzaffar R (2016) Survey on unmanned aerial vehicle networks for civil applications: a communications viewpoint. IEEE Commun Surv Tutor 18(4):2624–2661
40. He W, Qi X, Liu L (2021) A novel hybrid particle swarm optimization for multi-uav cooperate path planning. Appl Intell 2021:1–15
41. Hoang VT, Phung MD, Dinh TH, Zhu Q, Ha Q (2019). Reconfigurable multi-uav formation using angle-encoded pso. In: 2019 IEEE 15th international conference on automation science and engineering (CASE). IEEE, pp 1670–1675
42. Howard LM, D'Angelo DJ (1995) The ga-p: a genetic algorithm and genetic programming hybrid. IEEE Expert 10(3):11–15
43. Huang T, Wang Y, Cao X, Xu D (2020). Multi-uav mission planning method. In: 2020 3rd international conference on unmanned systems (ICUS). IEEE, pp 325–330
44. Hung SM, Givigi SN (2016) A q-learning approach to flocking with uavs in a stochastic environment. IEEE Trans Cybern 47(1):186–197
45. Hüttenrauch M, Adrian S, Neumann G et al (2019) Deep reinforcement learning for swarm systems. J Mach Learn Res 20(54):1–31
46. Ieee xplore. https://ieeexplore.ieee.org/Xplore/home.jsp. Accessed 24 Mar 2021
47. Kaelbling LP, Littman ML, Moore AW (1996) Reinforcement learning: a survey. J Artif Intell Res 4:237–285
48. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks, vol 4. IEEE, pp 1942–1948
49. Khalil AA, Byrne AJ, Rahman MA, Manshaei MH (2021) Efficient uav trajectory-planning using economic reinforcement learning. arXiv:2103.02676
50. Koza JR, Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection, vol 1. MIT press, New York
51. Koza JR, Poli R (2005) Genetic programming. Springer, Boston, pp 127–164. https://doi.org/10.1007/0-387-28356-0_5
52. Koziel S, Michalewicz Z (1998) A decoder-based evolutionary algorithm for constrained parameter optimization problems. In: International conference on parallel problem solving from nature. Springer, pp 231–240
53. Lamont GB, Slear JN, Melendez K (2007) Uav swarm mission planning and routing using multi-objective evolutionary algorithms. In: 2007 IEEE symposium on computational intelligence in multi-criteria decision-making, IEEE, pp 10–20
54. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
55. Li Q, Gama F, Ribeiro A, Prorok A (2019) Graph neural networks for decentralized multi-robot path planning. arXiv:1912.06095
56. Li J, Sun XX (2008) A route planning's method for unmanned aerial vehicles based on improved a-star algorithm. Acta Armamentarii 7:788–792
57. Liu Y, Passino KM (2000) Swarm intelligence: literature overview. Department of Electrical Engineering, The Ohio State University, Ohio
58. Liu W, Zheng Z, Cai K (2013) Adaptive path planning for unmanned aerial vehicles based on bi-level programming and variable planning time interval. Chin J Aeronaut 26(3):646–660
59. Liu W, Zheng Z, Cai KY (2013) Bi-level programming based real-time path planning for unmanned aerial vehicles. Knowl Based Syst 44:34–47
60. Liu J, Wang W, Wang T, Shu Z, Li X (2018) A motif-based rescue mission planning method for uav swarms usingan improved picea. IEEE Access 6:40778–40791
61. Liu C, Xie W, Zhang P, Guo Q, Ding D (2019) Multi-uavs cooperative coverage reconnaissance with neural network and genetic algorithm. In: Proceedings of the 2019 3rd high performance computing and cluster technologies conference. ACM, pp 81–86
62. Li X, Zhao Y, Zhang J, Dong Y (2016) A hybrid pso algorithm based flight path optimization for multiple agricultural uavs. In: 2016 IEEE 28th international conference on tools with artificial intelligence (ICTAI). IEEE, pp 691–697
63. Luo W, Tang Q, Fu C, Eberhard P (2018) Deep-sarsa based multi-uav path planning and obstacle avoidance in a dynamic environment. In: International conference on sensing and imaging. Springer, pp 102–111
64. Majd A, Ashraf A, Troubitsyna E, Daneshtalab M (2018). Integrating learning, optimization, and prediction for efficient navigation of swarms of drones. In: 2018 26th Euromicro international conference on parallel, distributed and network-based processing (PDP). IEEE, pp 101–108
65. McGovern A (2021) PyParrot. https://github.com/amymcgovern/pyparrot. Accessed 24 Mar 2021
66. Michie D, Spiegelhalter DJ, Taylor C et al (1994) Machine learning, neural and statistical classification. Citeseer 13
67. Miller PM (2006) Mini, micro, and swarming unmanned aerial vehicles: a baseline study. Inn: Library of congress Washington DC, Federal Research Div
68. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G et al (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529
69. Moeller M, Pohl D, Gurdan T (2019) Unmanned aerial vehicle swarm photography. US Patent App. 15/811,726
70. Olson JM, Bidstrup CC, Anderson BK, Parkinson AR, McLain TW (2020). Optimal multi-agent coverage and flight time with

genetic path planning. In: 2020 International conference on unmanned aircraft systems (ICUAS). IEEE, pp 228–237
71. Pan Y, Yang Y, Li W (2021) A deep learning trained by genetic algorithm to improve the efficiency of path planning for data collection with multi-uav. IEEE Access 9:7994–8005
72. Parunak HV, Purcell M, O'Connell R (2002) Digital pheromones for autonomous coordination of swarming uav's. In: 1st UAV conference, p 3446
73. Payton D, Daily M, Estowski R, Howard M, Lee C (2001) Pheromone robotics. Auton Robot 11(3):319–324
74. Perez-Carabaza S, Besada-Portas E, Lopez-Orozco JA, Jesus M (2018) Ant colony optimization for multi-uav minimum time search in uncertain domains. Appl Soft Comput 62:789–806
75. Ramirez-Atencia C, Bello-Orgaz G, R-Moreno MD, Camacho D (2017) Solving complex multi-uav mission planning problems using multi-objective genetic algorithms. Soft Comput 21(17):4883–4900
76. Ramirez-Atencia C, R-Moreno MD, Camacho D (2017) Handling swarm of uavs based on evolutionary multi-objective optimization. Progr Artif Intell 6(3):263–274
77. Rosenblatt F (1961) Principles of neurodynamics. Perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Lab Inc, Buffalo
78. Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 65(6):386
79. Roudneshin M, Sizkouhi AMM, Aghdam AG (2019) Effective learning algorithms for search and rescue missions in unknown environments. In: 2019 IEEE international conference on wireless for space and extreme environments (WiSEE). IEEE, pp 76–80
80. Roy S, Biswas S, Chaudhuri SS (2014) Nature-inspired swarm intelligence and its applications. Int J Modern Educ Comput Sci 6(12):55
81. Rui P (2010) Multi-uav formation maneuvering control based on q-learning fuzzy controller. In: 2nd international conference on advanced computer control, vol 4. IEEE, pp 252–257
82. Rummery GA, Niranjan M (1994) On-line Q-learning using connectionist systems, vol 37. Department of Engineering Cambridge, University of Cambridge, London
83. Russell SJ, Norvig P (2016) Artificial intelligence: a modern approach. Pearson Education Limited, Malaysia
84. Sahin E, Winfield AF (2008) Special issue on swarm robotics. Swarm Intell 2(2–4):69–72
85. San KT, Lee EY, Chang YS (2016). The delivery assignment solution for swarms of uavs dealing with multi-dimensional chromosome representation of genetic algorithm. In: 2016 IEEE 7th annual ubiquitous computing, electronics and mobile communication conference (UEMCON). IEEE, pp 1–7
86. Sathyan A, Ernest ND, Cohen K (2016) An efficient genetic fuzzy approach to uav swarm routing. Unmanned Syst 4(02):117–127
87. Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G (2008) The graph neural network model. IEEE Trans Neural Netw 20(1):61–80
88. Scopus. https://www.scopus.com/. Accessed 24 Mar 2021
89. Shah S, Dey D, Lovett C, Kapoor A (2018) Airsim: high-fidelity visual and physical simulation for autonomous vehicles. In: Field and service robotics. Springer, pp 621–635
90. Shao S, Peng Y, He C, Du Y (2020) Efficient path planning for uav formation via comprehensively improved particle swarm optimization. ISA Trans 97:415–430
91. Sharkey AJ, Sharkey N (2006) The application of swarm intelligence to collective robots. In: Advances in applied artificial intelligence. IGI Global, pp 157–185
92. Sivanandam S, Deepa S (2008) Genetic algorithms. Introduction to genetic algorithms. Springer, pp 15–37
93. Speck C, Bucci DJ (2018). Distributed uav swarm formation control via object-focused, multi-objective sarsa. In: 2018 Annual American control conference (ACC). IEEE, pp 6596–6601
94. Srinivas M, Patnaik LM (1994) Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Trans Syst Man Cybern 24(4):656–667
95. Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359
96. Su Xh, Zhao M, Zhao Ll, Zhang Yh (2016) A novel multi stage cooperative path re-planning method for multi uav. In: Pacific rim international conference on artificial intelligence. Springer, pp 482–495
97. Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. MIT press, New York
98. Sutton RS, Precup D, Singh SP (1998) Intra-option learning about temporally abstract actions. ICML 98:556–564
99. Tan Y, Zheng Z (2013) Research advance in swarm robotics. Defence Technol 9(1):18–39
100. Theraulaz G, Bonabeau E (1999) A brief history of stigmergy. Artif Life 5(2):97–116
101. Tolstaya E, Gama F, Paulos J, Pappas G, Kumar V, Ribeiro A (2020) Learning decentralized controllers for robot swarms with graph neural networks. In: Conference on robot learning. PMLR, pp 671–682
102. Tseng FH, Liang TT, Lee CH, Der Chou L, Chao HC (2014) A star search algorithm for civil uav path planning with 3g communication. In: 2014 Tenth international conference on intelligent information hiding and multimedia signal processing. IEEE, pp 942–945
103. Van Hasselt H, Wiering MA (2007) Reinforcement learning in continuous action spaces. In: 2007 IEEE international symposium on approximate dynamic programming and reinforcement learning. IEEE, pp 272–279
104. Venturini F, Mason F, Pase F, Chiariotti F, Testolin A, Zanella A, Zorzi M (2020)Distributed reinforcement learning for flexible uav swarm control with transfer learning capabilities. In: Proceedings of the 6th ACM workshop on micro aerial vehicle networks, systems, and applications, pp 1–6
105. Venturini F, Mason F, Pase F, Chiariotti F, Testolin A, Zanella A, Zorzi M (2021) Distributed reinforcement learning for flexible and efficient uav swarm control. arXiv:2103.04666
106. Vijayakumari DM, Kim S, Suk J, Mo H (2019) Receding-horizon trajectory planning for multiple uavs using particle swarm optimization. In: AIAA Scitech 2019 forum, p 1165
107. Wang BH, Wang DB, Ali ZA (2020) A cauchy mutant pigeon-inspired optimization-based multi-unmanned aerial vehicle path planning method. Meas Control 53(1–2):83–92
108. Watkins CJ, Dayan P (1992) Q-learning. Mach Learn 8(3–4):279–292
109. Web of science. https://www.webofknowledge.com/. Accessed 24 Mar 2021
110. Welcome to python.org. https://www.python.org/. Accessed 24 Mar 2021
111. Wiering M, Van Otterlo M (2012) Reinforcement learning. Adapt Learn Optim 12:3
112. Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS (2019) A comprehensive survey on graph neural networks. arXiv:1901.00596
113. Yang T, Yi X, Wu J, Yuan Y, Wu D, Meng Z, Hong Y, Wang H, Lin Z, Johansson KH (2019) A survey of distributed optimization. Annu Rev Control 47:278–305

114. Yang Q, Jang SJ, Yoo SJ (2020) Q-learning-based fuzzy logic for multi-objective routing algorithm in flying ad hoc networks. Wirel Person Commun 113:1–24
115. Ye F, Chen J, Tian Y, Jiang T (2020) Cooperative multiple task assignment of heterogeneous uavs using a modified genetic algorithm with multi-type-gene chromosome encoding strategy. J Intell Robot Syst 100:615–627
116. Yijing Z, Zheng Z, Xiaoyi Z, Yang L (2017). Q learning algorithm based uav path learning and obstacle avoidence approach. In: 2017 36th Chinese control conference (CCC). IEEE, pp 3397–3402
117. Zhang X, Ali M (2020) A bean optimization-based cooperation method for target searching by swarm uavs in unknown environments. IEEE Access 8:43850–43862
118. Zhao Y, Zheng Z, Liu Y (2018) Survey on computational-intelligence-based uav path planning. Knowl Based Syst 158:54–64
119. Zhao W, Fang Z, Yang Z (2020) Four-dimensional trajectory generation for uavs based on multi-agent q learning. J Navig 73(4):874–891
120. Zhao H, Pei Z, Jiang J, Guan R, Wang C, Shi X (2010) A hybrid swarm intelligent method based on genetic algorithm and artificial bee colony. In: International conference in swarm intelligence. Springer, pp 558–565
121. Zhao W, Qiu W, Zhou T, Shao X, Wang X (2019). Sarsa-based trajectory planning of multi-uavs in dense mesh router networks. In: 2019 international conference on wireless and mobile computing, networking and communications (WiMob). IEEE, pp 1–5
122. Zhao D, Wang H, Shao K, Zhu Y (2016). Deep reinforcement learning with experience replay based on sarsa. In: 2016 IEEE symposium series on computational intelligence (SSCI). IEEE, pp 1–6
123. Zhen Z, Xing D, Gao C (2018) Cooperative search-attack mission planning for multi-uav based on intelligent self-organized algorithm. Aerosp Sci Technol 76:402–411
124. Zhen Z, Chen Y, Wen L, Han B (2020) An intelligent cooperative mission planning scheme of uav swarm in uncertain dynamic environment. Aerosp Sci Technol 100:105826
125. Zhou Z, Luo D, Shao J, Xu Y, You Y (2020) Immune genetic algorithm based multi-uav cooperative target search with event-triggered mechanism. Phys Commun 41:101103
126. Zurada JM (1992) Introduction to artificial neural systems, vol 8. West publishing company St. Paul, Berlin

Applied Intelligence (2022) 52:14101–14118
https://doi.org/10.1007/s10489-022-03254-4

UAV swarm path planning with reinforcement learning for field prospecting

Alejandro Puente-Castro[1] · **Daniel Rivero**[1] · **Alejandro Pazos**[1,2] · **Enrique Fernandez-Blanco**[1]

Accepted: 15 January 2022 / Published online: 3 March 2022

Abstract
There has been steady growth in the adoption of Unmanned Aerial Vehicle (UAV) swarms by operators due to their time and cost benefits. However, this kind of system faces an important problem, which is the calculation of many optimal paths for each UAV. Solving this problem would allow control of many UAVs without human intervention while saving battery between recharges and performing several tasks simultaneously. The main aim is to develop a Reinforcement Learning based system capable of calculating the optimal flight path for a UAV swarm. This method stands out for its ability to learn through trial and error, allowing the model to adjust itself. The aim of these paths is to achieve full coverage of an overflight area for tasks such as field prospection, regardless of map size and the number of UAVs in the swarm. It is not necessary to establish targets or to have any previous knowledge other than the given map. Experiments have been conducted to determine whether it is optimal to establish a single control for all UAVs in the swarm or a control for each UAV. The results show that it is better to use one control for all UAVs because of the shorter flight time. In addition, the flight time is greatly affected by the size of the map. The results give starting points for future research, such as finding the optimal map size for each situation.

Keywords UAV swarm · Path planning · Reinforcement learning · Q-Learning · Artificial neural network · Agriculture

1 Introduction

New applications of Unmanned Aerial Vehicle (UAV or drones) swarms are developed nearly every day for different problems, such as crop monitoring [1, 2], forestry activities [3], space exploration [4, 5], or military and rescue missions [6]. The main reason for that popularity lies in the advantages offered by UAVs, such as low cost, great maneuverability, safety, and convenient size for certain kinds of maneuvers [7]. However, they also have disadvantages, the main one being battery consumption, which limits flight time. When UAVs are used in a group or swarm, their flight time limitations are reduced. In other words, several UAVs flying simultaneously allows many tasks to be carried out in less time because flight paths are shorter (Fig. 1). The flight paths of each UAV are shorter when multiple UAVs fly at the same time (Fig. 1d) than if one UAV has to fly over the complete flight environment (Fig. 1c). This minimizes the probability that the UAVs' battery capacities will be insufficient to allow them to fly over the terrain. As a result of the lower energy usage, there is a lower risk of a UAV crashing in the middle of an activity, resulting in less damage.

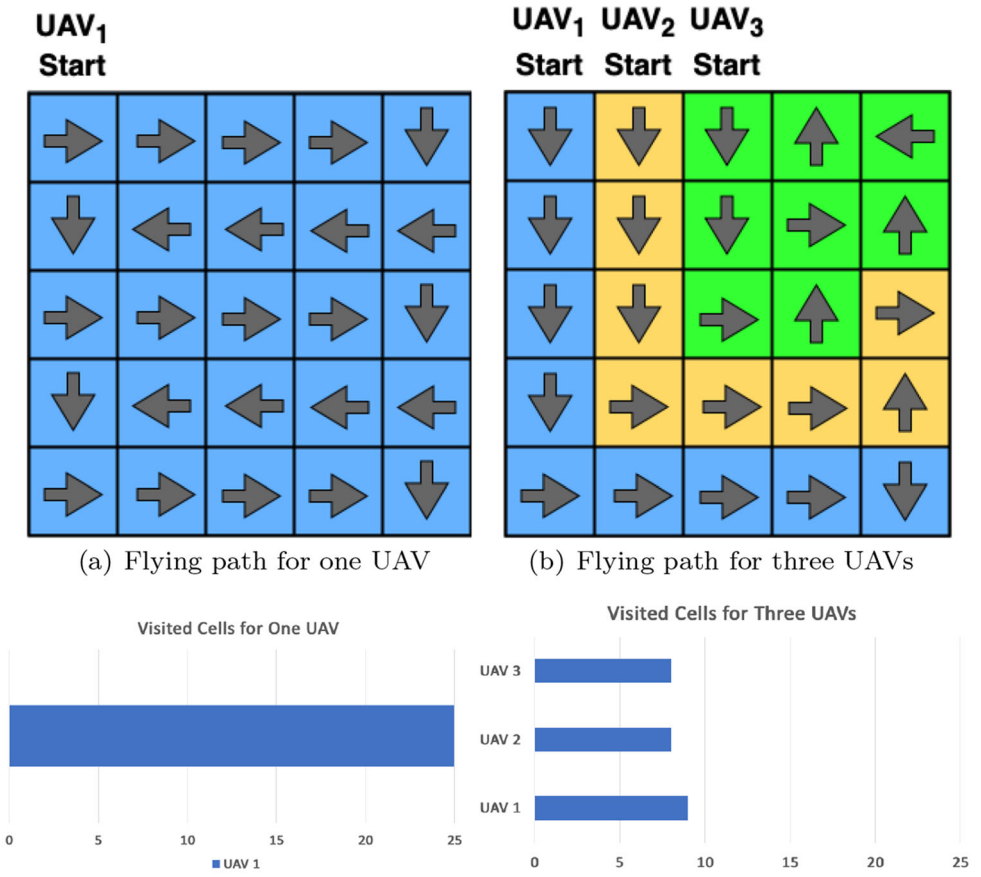
The use of UAV swarms can also provide fault tolerance. If only one UAV is used and it crashes, the activity must be stopped. However, if there are several UAVs, the surviving UAVs could assume all or part of the duties of the fallen UAV. This ensures that the work is completed to the best of our ability. Suspending a process when a job is urgent, such as in an emergency or a rescue operation, is difficult since time is of the essence. As a solution, even if one of the UAVs fails, the rescue can continue when using swarms of UAVs.

When the first flight tests were conducted, as many operators as UAVs were required, significantly increasing the operational costs. More recently, advances have been registered in the creation of algorithms [8] and telecommunications [9] necessary for the control of the entire swarm with only one user capable of executing the systems. These advances grant better and faster communications between

✉ Alejandro Puente-Castro
a.puentec@udc.es

[1] Faculty of Computer Science, CITIC, University of A Coruna, A Coruna, 15007, Spain

[2] Biomedical Research Institute of A Coruna (INIBIC), University Hospital Complex of A Coruna (CHUAC), A Coruna, 15006, Spain



(a) Flying path for one UAV (b) Flying path for three UAVs

(c) Number of cells visited for one UAV (d) Number of cells visited for three UAV

Fig. 1 When a swarm of three UAVs is used instead of a single UAV, the number of cells visited drastically changes. When one UAV is used alone, it visits a disproportionately large number of cells compared with the number of cells it would visit if used in conjunction with other UAVs. If the map size is too extensive, the UAV may not be able to visit as many cells

UAVs and grant the fast calculation of collision avoidance paths, so that less human intervention is required if there is any risk. Thus, the operation is less expensive because it requires fewer personnel.

To deal with the complexity of this kind of development, in Swarm Intelligence, different algorithms are proposed that are capable of simultaneously coordinating numerous agents. This coordination is based on a group of individuals that follows common simple rules in a self-organized and robust way [10].

Today, some of these path planning algorithms have military applications. The few civilian applications are usually to follow or reach targets, such as mapping paths through cities [11]. There are few systems oriented to agricultural and forestry use, specially dedicated to the optimization of the field prospecting tasks. Table 1 lists ten publications that demonstrate how different systems solve the Path Planning problem in various scenarios.

The aim of this paper is to use Q-Learning techniques to build a system for solving the Path Planning problem in 2D grid-based maps with different numbers of UAVs. The main contributions of this paper are: 1) a novel system capable of calculating the optimal flight path for UAVs in a swarm for field coverage in prospecting tasks; 2) a system capable of calculating the flight path of any number of UAVs and with any map size; 3) one of the few systems capable of calculating paths without the need to set targets or provide information other than the actual state of the map; and 4) a study on the difference in the results of using a global ANN for all UAVs and using one ANN per UAV.

This paper has the following structure: in Section 2, there is a brief summary of the current state of the art; in Section 3, an explanation is given of the technical aspects necessary for the development of the proposed algorithm; in Section 4, there is a summary of the results obtained from the experimentation process; in Section 5, the results obtained are discussed; in Section 6, the conclusions reached after reviewing the results obtained are listed; finally, Section 7, lists the possible works and studies into which the problem to be addressed can derive.

2 Background

In the state of the art, there are several approaches, two of which are particularly noteworthy [11]: the first one makes

Table 1 The two primary types of techniques utilized for Path Planning problems with UAVs are summarized in the table below

Technique	Publication	Observations
Reinforcement Learning [12]	Baldazo et al. [13]	Use in emergencies or disasters.
	Yang et al. [14]	Battery energy is considered for path planning.
	Roudneshin et al. [15]	Combine UAVs and ground robots.
	Luo et al. [16]	Requires pretraining phase.
	Speck et al. [17]	Tested on fixed-wing UAVs.
Evolutionary Computation [18]	Duan et al. [19]	Duan et al. [19]
	Zhuo et al. [20]	Path smoothing is required.
	Olson et al. [21]	Focused in maximizing coverage while minimizing flight time.
	Huang et al. [22]	Flight time is considered.
	Perez-Carabaza et al. [23]	Paths should be smoothed.

Each example includes an observation to demonstrate the wide range of approaches to the problem

use of Reinforcement Learning (RL) [12]; while the second one focuses on Evolutionary Computing (EC) [18].

RL algorithms for path planning are the most abundant in the state of the art. For example, Xie et al. use the Q-Learning strategy for three-dimensional path planning [24]. The notion of Heuristic Q-Learning was introduced. This allows a more precise adjustment of the reward depending on the current state and possible actions, leading to faster convergence to the optimal result. Deep Q-Learning is used by Roudneshin et al. to control swarms of UAVs and heterogeneous robots [15]. Rather than using only UAVs, this research incorporates a mix of terrestrial robots into the swarms. However, this is a more challenging problem of swarm path planning than using only UAVs. Due to the differences in restrictions faced by air and ground vehicles, the problem has become more complex to solve. As a result, a land vehicle is more constrained in its mobility and might meet non-geographic impediments.

Others, such as Luo et al., employ the RL algorithm known as SARSA [25], where they tested their Deep-SARSA algorithm in dynamic environments with changing obstacles [16]. They demonstrate how their system behaves in different contexts, demonstrating their utility in the real world. The model requires a pretraining phase, which may limit its application in unfamiliar situations due to the time commitment. When generalizing, Speck et al. integrate object-focused learning with this method in a highly efficient decentralized way [17]. As it was designed for fixed-wing UAVs, this capacity for generalization may be limited, and because these aircraft lack stationary flying capabilities, the configuration of these UAVs restricts the system's use to situations where fixed-wing UAVs are the best option.

On the other hand, there are EC-based methods. For example, Duan et al. combine a genetic algorithm with the VND search algorithm [19]. An initial individual is generated based on the heuristics of its nearest neighbors and the rest of the initial individuals are configured randomly. The use of the closest neighbors limits the generation of individuals. Especially in the case of many equally close neighbors. In that situation, it is necessary to establish a criterion to determine whether the individual is a member of a group. Recently, Liu et al. employed Genetic Algorithms to adjust ANN for flight path generation [26]. Relying only on the ANN for path computation makes their system dependent on more parameters than weights. Therefore, other parameters, such as learning rates or adjusting the architecture of the ANN should be adjusted.

There are other methods applied to path planning with UAV swarms. For example, Vijayakumari et al. make use of Particle Swarm Optimization for optimal control of multiple UAVs in a decentralized way [27]. They manage to simplify the computation of the problem by means of discretization. They rely on distances for collision avoidance. Although this is a dynamic variable, in certain types of non-stationary flight UAVs, such as fixed-wing UAVs, it does not guarantee collision avoidance. In these cases, a metric that predicts the state of the UAV and the obstacle at future moments in time is of interest and thus makes a decision. Otherwise, the UAV would continue to move forward while the decision is being computed. Li et al. make use of Graph Neural Networks for path computation in robotic systems. Thus, they achieve more capacity for generalization in the face of new cases than other more widely used techniques [28]. Since they are dealing with two ANNs, previous training is necessary in different and very varied cases. Otherwise, ANNs could be overfitted in several flight areas and swarm structures.

As has been shown in the associated literature, systems often require extra map information, such as targets or distance maps. In addition, they use maps with a fixed number of cells. The aim of this work is to propose a system

without the need for extra map information and that works with any map size.

3 Materials and methods

3.1 Problem formulation

Path Planning issues with multiple vehicles are subject to several factors in order to ensure standards of control, cooperation and safe operation while maintaining efficiency and effectiveness. Therefore, it is necessary to be able to solve the problems related to these variables as problems inherent to the main objective.

These problems are (Fig. 2): first, to establish the fligh t environment; second, to define the UAV movements; third, to establish the most appropriate technique for path calculation; fourth, to establish the optimal parameters to solve the problem; and, finally, to define mechanisms to confirm the validity of the proposed model and the satisfaction criteria of the results obtained.

3.2 Flight environments

Despite the existence of well-known tools for flight simulation with UAV swarms, such as AirSim [29], no large datasets are known to be used by most authors. Previously published works, described in Section 2, used fixed squared maps of dimensions between 10×10 and 20×20. The approach presented in this paper takes a wider point of view allowing the use of arbitrary polygons as maps, e.g. the one presented in Fig. 3a. To do this, the following steps have been followed:

1. The minimum bounding rectangle (MBR) of the map has to be calculated such as in Fig. 3b. The map polygon is surrounded by a rectangle of the smallest possible size based on the combined spatial extent of one or more selected map features [30]. In this case, based on its vertices.
2. The resulting MBR is divided into cell, as shown in Fig. 3c. Cells in the resulting grid have to be labelled as visitable and non-visitable.

3.3 Proposed method

3.3.1 Reinforcement learning

Reinforcement Learning (RL) [12] was chosen as the technique for calculating the optimal path to cover the maps by the UAVs. With this technique, the agents learn the desired behavior based on a trial-and-error scheme of tests executed in an interactive and dynamic environment [12, 31, 32]. The goal is to optimize the behavior of the agent in respect to a reward signal that is provided by the environment. The actions of the agent can also affect the environment, complicating the search for the optimal behavior [33].

All RL algorithms follow a common structure, the only difference is the learning strategy. There are several types of these strategies which allow the models to deal with different problems. For this paper, it has been decided to use a variant known as Q-Learning [34]. The main motivation is that, unlike other variants, it does not require a model of the environment.

3.3.2 Q-Learning

Classic Q-Learning algorithms [34] are a kind of off-policy RL algorithms, so the agents can use their experience to learn the values of all the policies in parallel, even when they can follow only one policy at a time [35]. It follows a model-free strategy [36], where the agent acquires knowledge by following a policy only by trial-and-error. In this way, Q-Learning convergence towards the optimal solution is greedy, allowing the optimal solution to be reached without being dependent on the decision-making policy. In other words, it makes decisions based purely on the environment surrounding the agent and its interactions with it. In this way, it is guaranteed that the system can work with different types of environments without having to search for the optimal policy that works in all of them. The "Q" in Q-learning stands for quality, which tries to represent how useful a given action is in gaining some future reward.

The most well-known advantage of Q-Learning over other RL techniques is that it can compare the expected utility of various actions without requiring an environment model. The ease with which it generalizes environments

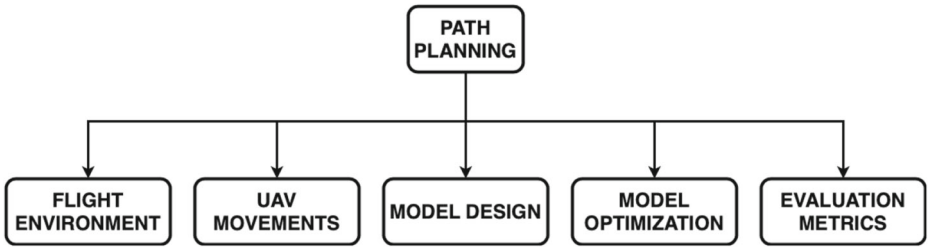


Fig. 2 Diagram with the formulation of Path Planning problems. It summarizes all the inherent and necessary problems to guarantee the validity of the final system

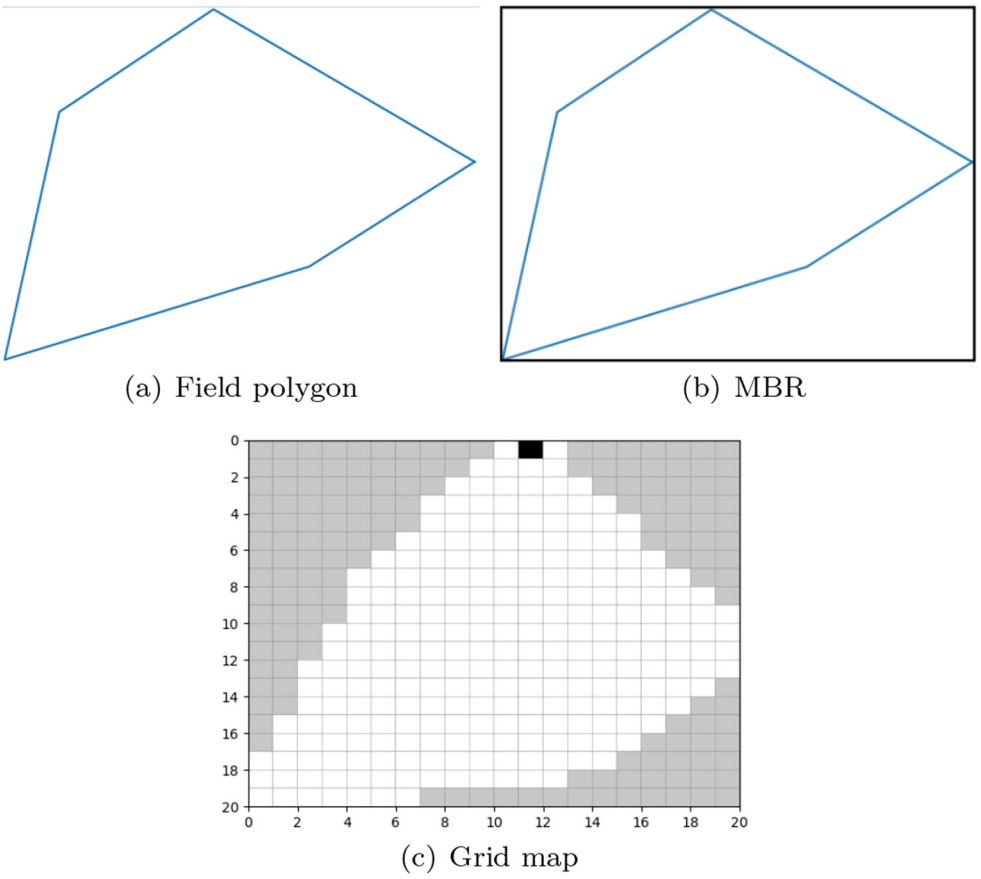


Fig. 3 (a) Example of the polygon representing the area of the indicated field. (b) The same polygon (blue) surrounded by its minimum bounding rectangle (black). The MBR must be as close as possible to the polygon. (c) The MBR divided into cells. The black cell is the starting point of the UAVs. The gray cells are those that cannot be flown over and the white cells are those that can be flown over

without having to model them is the main reason for it having been chosen for this research. In addition, the main difference between these algorithms and other RL algorithms is that they determine the best action based on the values in a table. The table is known as Q-table and the values as Q-values. These values determine how rewarding it would be to perform each action given the current state of the environment. From these values, the action with the highest value for each state is chosen. Typically, models are trained by combining their previous predictions with Bellman's (1). The equation has different elements: $Q(s, a)$ is the function that calculates the Q-value for the current state (s), of the set of states S, and for the giving action (a), of the set of actions A, r is the reward of the action taken in that state and it is computed by the reward function $R(s, a)$, γ is the discount factor and $\arg\max_{a'}(Q(s', a'))$ is the maximum computed Q-value of the pair (s', a') represented as $Q(s', a')$. The pair (s', a') is a potential next state-action pair. (s' is the next state and it is given by the transition function $T(s, a)$ which returns the state resulting from performance of the selected action. The a', is each one of the available actions. Through an initial exploration process, the chosen value for γ is 0.91.

$$Q(s, a) \leftarrow r + \gamma \times \arg\max_{a'}(Q(s', a')) \tag{1}$$

Alternatively, in recent years, a modification has arisen called Deep Q-learning. This method differs from the classic Q-Learning [34] in that it seeks to improve the calculation of the Q table through Machine Learning [37] or Deep Learning models [38]. The model is able to abstract enough knowledge to infer the values of the Q table. In this way, it is possible to overcome Bellman's Equation bias issues in some scenarios [39].

The aim is to improve classical Q-Learning by using small ANNs. In this study, authors chose to use fully connected ANNs with two layers. Using only two-layer learning and decision-making usually takes less time compared with convolutional deep ANNs [40] that other authors propose in their papers. Therefore, the following steps are followed in each Q-Learning experiment:

1. Build the ANN model(s) based on the chosen configuration.
2. Employ the model(s) to determine Q-table values in order to choose the best action for each UAV in the swarm.
3. Train the model(s) according to the consequences of taking each of the selected actions.
4. Select the cases where the flight time required to explore the entire map is lower.

Through empirical experimentation, a network formed by two dense layers [41] has been chosen: the first one with 167 neurons and linear activation function [42] and the second one with 4 neurons and softmax activation function. The chosen optimizer for the ANN was RMSprop [43]. Maps are the only input of the network (Fig. 4). Thus, ANN does not need more information than that included in the maps.

From this point, the system could be used in two different approaches with no clear advantage for either of them. First, a single ANN is developed and used as the control for each of the UAVs. Therefore, all UAVs are going to have the same architecture and weights and their behavior will depend on the current state of the UAV. On the other hand, each UAV can have a different ANN, therefore its response would not only be the result of the state but also of the weights and architecture codified in it.

In all Q-Learning problems, a part of the actions is made randomly with a probability epsilon ($\epsilon = 0.47$), and with probability $1-\epsilon$ the action with the highest Q-value for that state is taken. The sequence of actions taken by an agent for a given ϵ until it reaches an end condition (task completed, end of time...) is known as an episode. In each episode, the task is restarted from the beginning. As episodes occur during testing, the ϵ value is reduced multiplying it by a reduction factor equal to 0.93. In this way, the choice of actions falls more on the calculated Q-values and less by random selection. To avoid overfitting, ϵ is prevented from reaching a value very close to 0 by setting the minimum value at 0.05. Both values were chosen through a previous exploratory study.

3.3.3 Rewards

In order to prioritize the UAV to move to unvisited areas, the reward must be the highest of all, as shown in Table 2. In addition, it is important that it increases as fewer cells are left undiscovered (2). This is known as Hill-Climbing [44]. Another reward is required for cells that have already been visited. Thus, the UAV has a reward in case it is better to fly over an already visited cell to reach an unvisited one than to go around it (for example, when there are spurious cells left unvisited). To prevent UAVs from flying into cells that they cannot visit, they are given the lowest reward. The choice of the selected reward values was made through an initial exploratory process.

Table 2 Assigned rewards to each kind of cell each UAV visits

	Reward
New cell base reward	358.74
Visited cell reward	−31.14
Non-visitable cell	−225.17

The initial values chosen for the rewards by means of a previous random exploration where the best combinations of rewards have been selected

$$\text{new cell reward} = \text{new cell base reward} \times \left(1 + \frac{max(\text{rows}, \text{columns})}{\text{visited cells}}\right) \tag{2}$$

3.3.4 Flying Actions

The possible movements or actions (a) from the set of actions A that UAVs can take were codified. Thus, all possible movements are encoded to a discrete list of values.

Despite the natural complexity of UAV flight, the possible movements have been simplified into straight movements, thereby making it easier to interpret flight paths in a map divided into cells. Otherwise, a UAV could draw a curve passing over the corner of a cell without actually passing through the entire cell. This would create the dilemma of

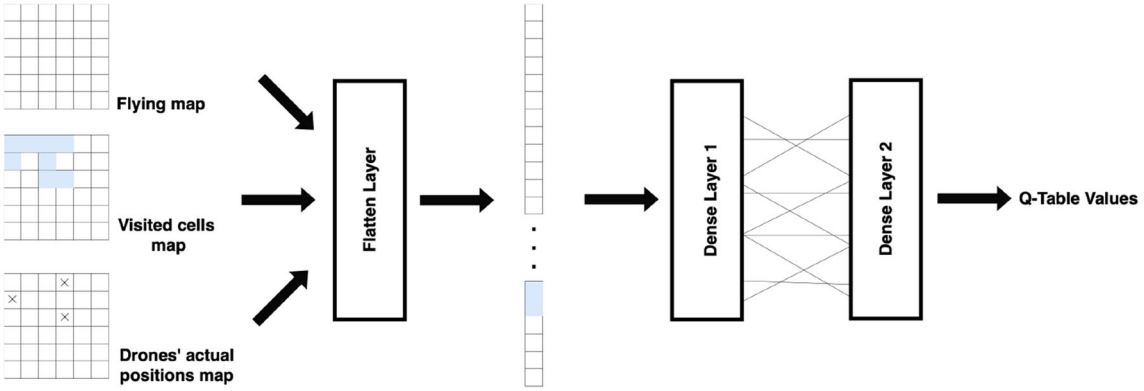


Fig. 4 Diagram showing how data is processed in the ANN in order to get Q-values of the Q-table for a given state of the environment. The maps are combined into a multidimensional matrix and then flattened into vectors. These vectors are used to abstract the knowledge for computing Q-values

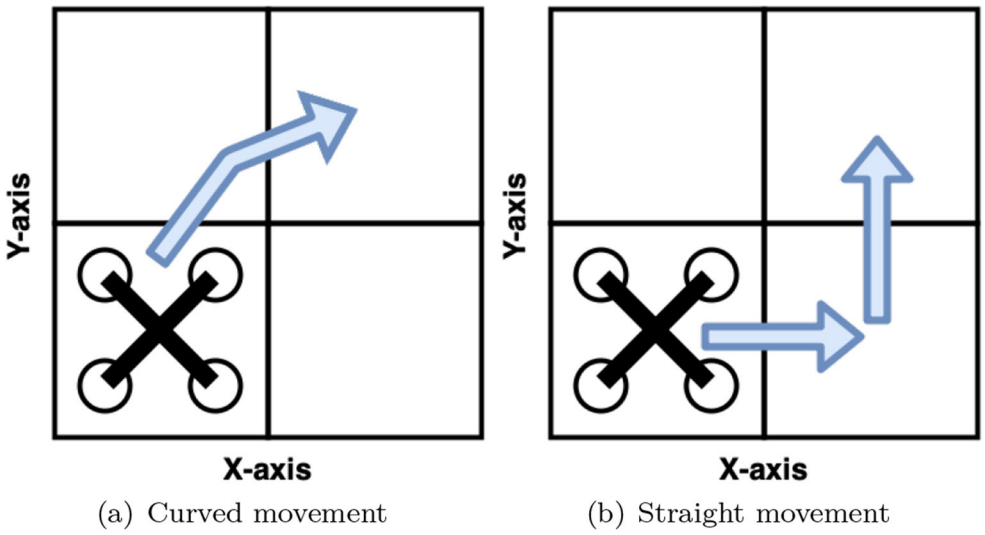


Fig. 5 Comparison of curved movements with straight movements Curved motions produce more easily interpreted flight paths. Moreover, they are limited to the atomicity of the map cells

whether to mark that cell as visited or not (Fig. 5). Not tracing curves ensures that the graphic data obtained with UAVs always have the same angle and are easier to combine.

3.3.5 Memory replay

In most of the State of the Art, the experience obtained by agents from the environment is reinforced with the Memory Replay technique. Memory Replay is a technique where the model is trained with a set of stored observations called memory. The observations contain a variety of information, such as the actions taken and their reward. It improves sample efficiency by repeatedly reusing experiences and helps to stabilize the training of the model [45]. It is important that the memory contains as many recent observations as possible, but it has a maximum size in order to optimize computational resources. For this reason, the memory follows a First-In-First-Out scheme for eliminating old observations.

Each UAV in the group has its own memory. In its memory, it stores observations with the actions that the UAV itself takes. At no time the actions of other UAVs are stored. This avoids adding noise to the information. The fact that an action is not correct for one UAV does not imply that it is incorrect for the others since they can be in different positions on the map.

The size of the memory can greatly influence the final results [46]. For this study, a memory size of 60 actions with their respective rewards was chosen after an exploration process. It is important to have a large value with respect to the number of map cells because in the first iterations of the process UAVs make many errors. Thus, learning from most of the errors and training the model multiple times with them will help to avoid them and, thus, achieve a more efficient solution.

3.3.6 Workflow

The scheme shown in Fig. 6 summarizes the main workflow of the proposed method. Starting from the reading of the initial data, that is the vertices of the area to be covered and

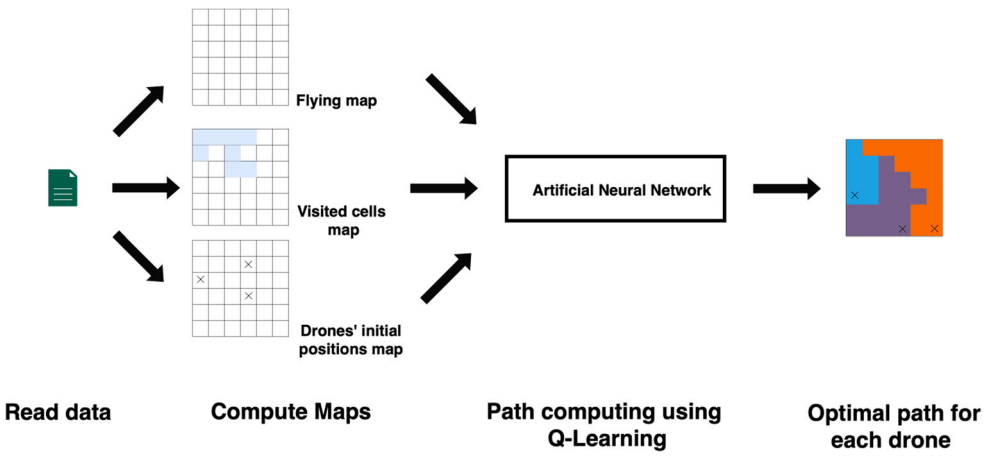


Fig. 6 The workflow diagram of the study. The map data and the position of the UAVs are read from a file. The maps that the system will use are constructed from the data read. Using the Q-Learning technique, the best possible path is calculated for each UAV so that the task is completed

the initial positions of the drones, the maps are reconstructed. After that, by using those maps, the ANN is trained with the Q-Learning technique [34]. For part of the experiments a global ANN is used, whereas in another part one ANN is used per UAV. This is going to determine the best action for each UAV.

3.4 Battery estimation

As in the works discussed in Section 2, the authors have not found a standardized method to predict battery power consumption during flight. This is because the consumption depends on the UAV configuration and flight conditions. Normally, most commercial UAVs send to their mobile apps the amount of energy they have left over at regular intervals. On the other hand, there is an increasing number of websites that help to calculate how much battery time is left. The lack of standardization is due to the influence of many variables. The incident wind, the number of direction changes, speed, and many other variables greatly affect the flight time.

As a solution to the problem of battery consumption, the swarm is forced to find a solution in the time corresponding to the minimum remaining battery time among the UAVs of the swarm. In this way, it is expected that in a limited time the UAVs will try to get as close as possible to the desired solution. In this study, it has been assumed that the UAVs all have a maximum energy load that allows them to fly for 30 minutes because no realistic calculation of the remaining battery life has been found.

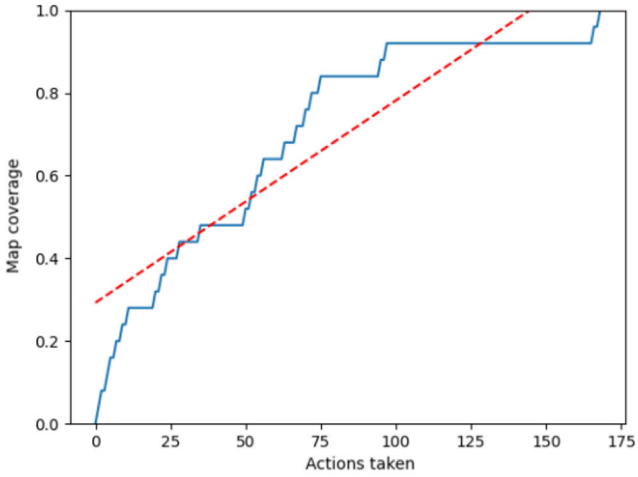
3.5 Performance measures

As for performance measures, the most common ones will be taken into account: the time needed to find the solution, the percentage of correct actions out of all actions taken, and the evolution of the map coverage.

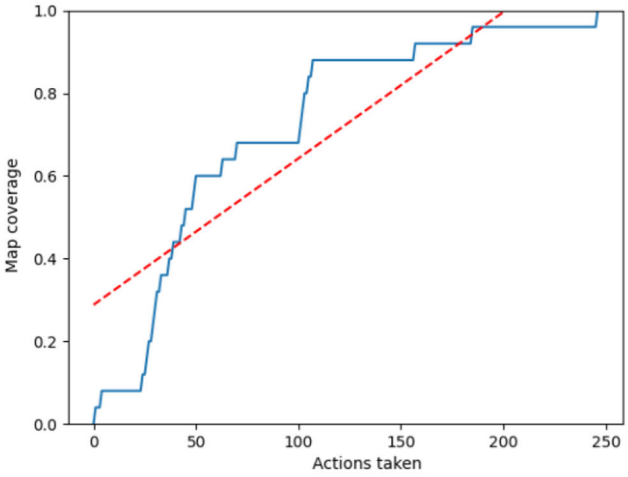
It is necessary to find a system that solves the problem as quickly as possible. Thus, it will require less operator time and battery consumption when used in real fields. Low battery consumption indicates that the paths are as short as possible. In addition, due to the charging time of the UAV batteries, low battery consumption might allow the user to do more work without having to stop charging the batteries. This makes it the measure of greatest interest and the most commonly chosen one. For this purpose, we will look for the episodes with the shortest execution time (ET), which is computed as the difference between the actual time when the episode finished or TE_1 and the actual time when the episode started or TE_0 (3). It is important to compute this coverage for each action taken in each individual episode in order to obtain the curve that relates the change in the number of cells discovered by the agents versus the number of total actions that they carry out. The greater the growth of the curve means that fewer movements are needed to reach the solution. This implies that the paths they take have fewer cycles and are therefore more efficient. In Fig. 7 there is an example plot of the curve for one ANN per UAV using two UAVs. By comparison with using the same ANN for all UAVs (a global ANN), its growth is higher, and it reaches total coverage much faster.

$$ET = TE_1 - TE_0 \tag{3}$$

Even though time is used as a measure of performance, it is also needed for calculating the length of the paths UAVs



(a) One ANN per UAV



(b) A global ANN

Fig. 7 Example of curves of the evolution of map coverage (y-axis) as a function of the number of actions taken by two UAVs (x-axis). The dashed line represents the coverage growth trend. Comparing the case of one ANN per UAV with a global ANN it can be seen that the case with a global ANN takes more actions to fly over the whole map

take for each episode. Depending on the configuration of the UAV and the way the UAV flies, the battery consumption can vary greatly. Therefore, it is interesting for the path to be as short as possible. It can be understood as the largest possible number of valid actions to be taken. Valid actions have been defined as those where a new cell is discovered. That is, without loops or passing through cells that cannot be visited. Therefore, it is of interest to know the fraction (PA) of valid actions (VA) out of all the actions taken by the UAVs (TA) (Eq. 4). The closer to 1, the better.

$$PA = \frac{VA}{TA} \tag{4}$$

Knowing how the total map coverage evolves makes it possible to distinguish which methods are better. It is calculated as the fraction of cells that have been visited divided by the total number of cells. Sometimes, the operational resources available (number of UAVs, battery levels, etc.) might not be sufficient to overfly the selected terrain in its entirety. Even so, in such cases, it is important to cover as large an area as possible. That is, a system in which it is able to get closer and closer to 100% map coverage is ideal. The closer to 1, the better.

4 Results

A set of combinations of map sizes and number of UAVs has been defined for conducting the experiments and subsequent analysis of the results. For the analysis of the results obtained, factors such as the evolution of the time required to explore the map and the percentage of actions performed by each UAV have been taken into account.

4.1 Experiment design

To test the capabilities of the system proposed in this paper, 25 experiments have been designed. In each of them, the configuration of the ANNs, the number of UAVs, and the size of the map are different.

The experiments were carried out in a square cell map as in those cited in Section 2.

The aim of this experiment is to identify the best controller for the UAVs. There are two approaches at this point: one ANN per UAV and one ANN for all UAVs (Fig. 8). Both approaches were compared using the same maps and the same UAVS. The results are listed in Table 3. As can be seen in the table, the experiments with one ANN per UAV have been omitted when there is only a single UAV. Using one ANN for only one UAV would be the same as using a global ANN for only one UAV. Therefore, it has been simplified to execute only once with a global ANN for one UAV and it is referred to as baseline (Fig. 9). Thus, it is taken as the starting point of the experimentation taking it as the simplest case, which is to control a single UAV.

Since it is important for the system to operate with any number of UAVs, each selected map type was tested with an increasing number of UAVs. To be more precise, separate experiments have been performed with 1, 2, and 3 UAVs. Thus, it is proved that the system can adapt to a different number of UAVs.

As in the papers mentioned in Section 2, all the maps chosen are grid maps. The experiments were performed with 5×5, 6×6, 7×7, 8×8, and 9×9 cell grid maps. Having different map sizes provides insight into the capabilities of the system in the face of unfixed map sizes.

In addition, the number of cells in the chosen flight environment is smaller than other cited papers. The cost of flying over large maps is a major constraint. By making one

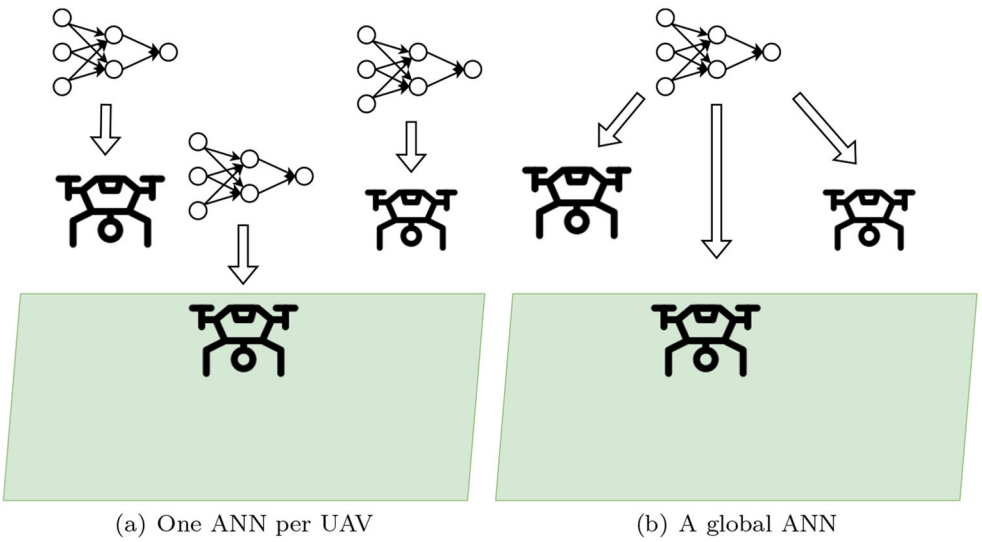


Fig. 8 Illustrative diagram showing the relationship between the ANN and the UAV in the two proposed approaches: an ANN per UAV and a global ANN. The training process is the same, only the relationship between the model or models and the UAVs changes. In the case of an ANN per UAV, the same ANN architecture is maintained, only the weights change

Table 3 Table with the 45 experiments performed

Approach	Map size	Number of UAVs
Baseline	5×5	1 UAV
	6×6	1 UAV
	7×7	1 UAV
	8×8	1 UAV
	9×9	1 UAV
One ANN per UAV	5×5	2 UAVs
		3 UAVs
		4 UAVs
		5 UAVs
	6×6	2 UAVs
		3 UAVs
		4 UAVs
		5 UAVs
	7×7	2 UAVs
		3 UAVs
		4 UAVs
		5 UAVs
	8×8	2 UAVs
		3 UAVs
		4 UAVs
		5 UAVs
	9×9	2 UAVs
		3 UAVs
		4 UAVs
		5 UAVs
Global ANN	5×5	2 UAVs
		3 UAVs
		4 UAVs
		5 UAVs
	6×6	2 UAVs
		3 UAVs
		4 UAVs
		5 UAVs
	7×7	2 UAVs
		3 UAVs
		4 UAVs
		5 UAVs
	8×8	2 UAVs
		3 UAVs
		4 UAVs
		5 UAVs
	9×9	2 UAVs
		3 UAVs
		4 UAVs
		5 UAVs

Each one of them with different configuration. The experiments for an ANN per UAV for a single UAV have been omitted because it is the same as using a global network for a single UAV

stop per cell to photograph the surface of the map each cell contains means that in very large maps the UAVs have to make numerous stops, considerably affecting their battery. Dividing the map into fewer cells reduces the number of stops and starts made by each UAV decreasing their energy consumption.

Another factor to consider is the area of land that each cell represents. The larger, the better, the more information each image contains and the more favorable it is for further processing. These cells must contain an adequate surface area size for each type of activity performed. For example, in tasks such as water stress [47], in which one flies at a height of 12 meters, the area size of the map contained in each cell is enormous.

In many countries the distance from the position to which it flies is limited by the height at which a UAV can fly. For example, in many European countries it is 500 meters or additional measures would have to be taken that not all operators can overcome [48]. Therefore, as it is known that the terrain cannot be too large for the system to be used in a generic way, it is not necessary to use maps with numerous cells. For example, 400-cell maps, such as those used in some papers described in Section 2.

4.2 Experimental results

The results indicate that a global ANN is usually the best choice because usually it has the fastest solutions, specially in larger maps (Table 4). Although it is slower on several cases like 5×5 cell maps for 3 or more UAVs, it is only by a few seconds. When using larger maps, the model needs more time to find a solution regardless of the number of UAVs. This is due to the size of the exploration tree. That is, the larger the map, the more possible path combinations the ANNs have to evaluate. Usually, when using 3 UAVs these exploration times are slower than when using 2 UAVs. It is not a problem since it is often a few seconds and, the more UAVs, the easier it is to reassign the task of a fallen UAV to the others.

In all the experiments conducted the map has been completely covered at least once. The total number of episodes in an experiment where UAVs fly over all cells in the map is usually higher when a global ANN is used. This is indicative of the robustness of the ANN configuration for the problem. Having a greater number of solutions demonstrates that the system is learning and is able to find solutions as the randomness component ϵ is reduced.

Regarding the evolution of the flight paths (Fig. 10), it was found to be highly dependent on the initialization and random component of all Q-Learning problems. In other words, if many wrong decisions are made from the outset, in the long-run this negatively affects future decisions. In the example of Fig. 10, the UAVs keep flying over the cells

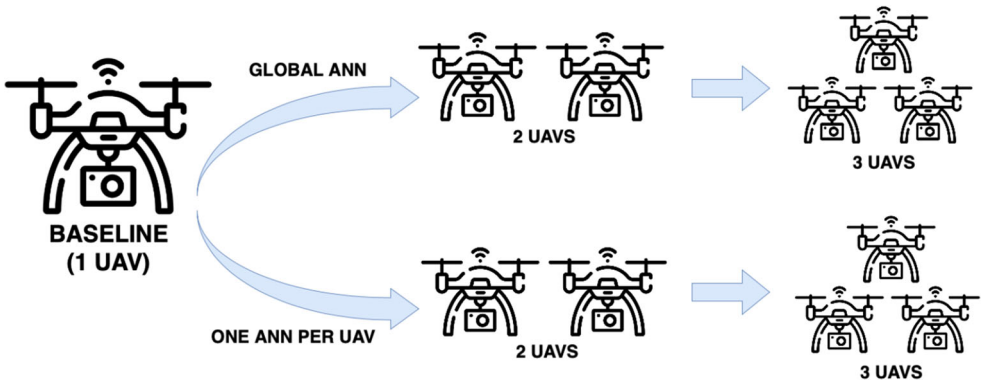


Fig. 9 Diagram of how the experiments are conducted for each map. Initially, the system is tested with a given map and a single UAV. If a solution is obtained, the number of UAVs is increased as the system is able to solve the problem for that map. For the experiments, it is differentiated to use a single ANN for all UAVs and a global ANN

at the left of the map. In this case, the UAVs started the flight by taking many actions that resulted in passing through the cells on the left side. An excess of these actions caused the UAVs to barely fly over other cells. In addition, the innermost cells are the least visited, and there are even some cells that have never been visited. This situation leads to understand that the proposed model identifies better the navigable cells of the edges because they have fewer neighbors. It is also worth noting that the cell at the bottom right has been visited when one of its neighbors has not. This reinforces the idea that the proposed model had better performance on cells on the left edge, which causes that when a cell is on the right edge it did not know how to behave.

Having solutions with different map sizes and different numbers of UAVs confirms that the system is generic enough to work under different conditions. To the best of our knowledge, this is the only paper that can do that. Other papers only work with predefined map sizes [49–51].

If we take the times for each configuration of number of UAVs and ANNs we see that they follow a Gaussian distribution according to the Shapiro-Wilk test [52] with a significance level (α) of 5% (Table 5).

Since it was shown that all distributions were Gaussian, we chose the ANOVA statistical significance test [53]. Thus, we can know if the results present significant differences that allow us to decide which option is more appropriate. According to the results listed in Table 6, the solutions of the ANOVA test are significantly different for a 5% significance level, so it can be understood that there are substantial changes when applying some solutions or others. Considering the result of applying the same test to all distributions except Baseline we have also observed that they are significantly different for the same significance level. This reinforces the idea that using a global ANN for all UAVs is significantly better than using a per UAV ANN. Unlike the Baseline case, employing more than one UAV guarantees a level of fault tolerance. Therefore, we can conclude that using a single ANN for the whole swarm is the best configuration and that using more than one UAV reduces the risk during operation.

4.3 Required time evolution

The speed to cover the entire map is also reflected in the time measure. Many episodes have a run time of 30 min. These coincide with the cases in which the entire map is not covered. However, when this is not the case, the time required decreases as the training advances. It can be seen that it is highly dependent on the size of the map and the number of UAVs (Fig. 11). In many cases, once the overall minimum is reached, the results worsen significantly. It is caused by the noise introduced by the random component of the chosen method. Because of this, the sequence of steps that is optimal is that of the episode of the sequence that finds the solution in the shortest time.

The evolution of the time taken by a fixed number of UAVs to find the solution on different maps is highly dependent on the size of the map. In Fig. 12, an example plot with 2 UAVs is shown in which it can be seen that the time curve shows more growth than the curve of the number of cells in a map.

4.4 Taken actions evolution

The fraction of the actions where a new cell was discovered, also known as valid actions, among those taken shows different behavior. Therefore, this factor should be taken into account, as it also determines whether the system makes too many errors or takes too many cycles. The more UAVs, the more actions are performed, decreasing the overall percentage of valid actions among all actions taken (Fig. 13). In the initial episodes, the UAVs take a lot of wrong actions. The accumulation of this number of failures impairs the percentage of valid actions taken over the total number of actions. In theory, this is not a problem, since the desired solution is reached faster in the next episode.

Table 4 Summary table with the observed results of the experiments

Map size	Number of UAVs	One ANN per UAV	Global ANN
5×5	1	00:00:40 (28 found solutions out of 30 episodes)	00:00:40 (28 found solutions out of 30 episodes)
	2	00:02:09 (26 found solutions out of 30 episodes)	00:01:32 (29 found solutions out of 30 episodes)
	3	00:01:10 (27 found solutions out of 30 episodes)	00:01:29 (25 found solutions out of 30 episodes)
	4	00:01:07 (23 found solutions out of 30 episodes)	00:01:59 (13 found solutions out of 30 episodes)
	5	00:01:16 (24 found solutions out of 30 episodes)	00:02:21 (16 found solutions out of 30 episodes)
6×6	1	00:02:38 (28 found solutions out of 30 episodes)	00:02:38 (28 found solutions out of 30 episodes)
	2	00:04:23 (7 found solutions out of 30 episodes)	00:02:12 (22 found solutions out of 30 episodes)
	3	00:04:27 (12 found solutions out of 30 episodes)	00:04:39 (14 found solutions out of 30 episodes)
	4	00:02:55 (16 found solutions out of 30 episodes)	00:03:16 (6 found solutions out of 30 episodes)
	5	00:04:32 (13 found solutions out of 30 episodes)	00:03:55 (12 found solutions out of 30 episodes)
7×7	1	00:03:14 (24 found solutions out of 30 episodes)	00:03:14 (24 found solutions out of 30 episodes)
	2	00:06:17 (9 found solutions out of 30 episodes)	00:06:16 (14 found solutions out of 30 episodes)
	3	00:07:01 (12 found solutions out of 30 episodes)	00:06:51 (9 found solutions out of 30 episodes)
	4	00:07:59 (4 found solutions out of 30 episodes)	00:11:47 (4 found solutions out of 30 episodes)
	5	00:04:35 (7 found solutions out of 30 episodes)	00:10:09 (7 found solutions out of 30 episodes)
8×8	1	00:07:37 (9 found solutions out of 30 episodes)	00:07:37 (9 found solutions out of 30 episodes)
	2	00:19:31 (2 found solutions out of 30 episodes)	00:14:14 (5 found solutions out of 30 episodes)
	3	00:16:58 (2 found solutions out of 30 episodes)	00:13:52 (5 found solutions out of 30 episodes)
	4	00:10:16 (3 found solutions out of 30 episodes)	00:08:17 (6 found solutions out of 30 episodes)
	5	00:13:50 (2 found solutions out of 30 episodes)	00:21:18 (2 found solutions out of 30 episodes)
9×9	1	00:12:17 (8 found solutions out of 30 episodes)	00:12:17 (8 found solutions out of 30 episodes)
	2	00:24:45 (1 found solutions out of 30 episodes)	00:16:15 (2 found solutions out of 30 episodes)
	3	00:20:53 (1 found solutions out of 30 episodes)	00:20:27 (1 found solutions out of 30 episodes)
	4	00:17:21 (2 found solutions out of 30 episodes)	00:22:09 (1 found solutions out of 30 episodes)
	5	00:27:56 (1 found solutions out of 30 episodes)	00:18:27 (1 found solutions out of 30 episodes)

Results are displayed with the minimum time in each configuration needed for finding a solution. As in the papers discussed in Section 2, the results shown here are those obtained from a single execution due to the computational and time costs of averaging the results of multiple runs

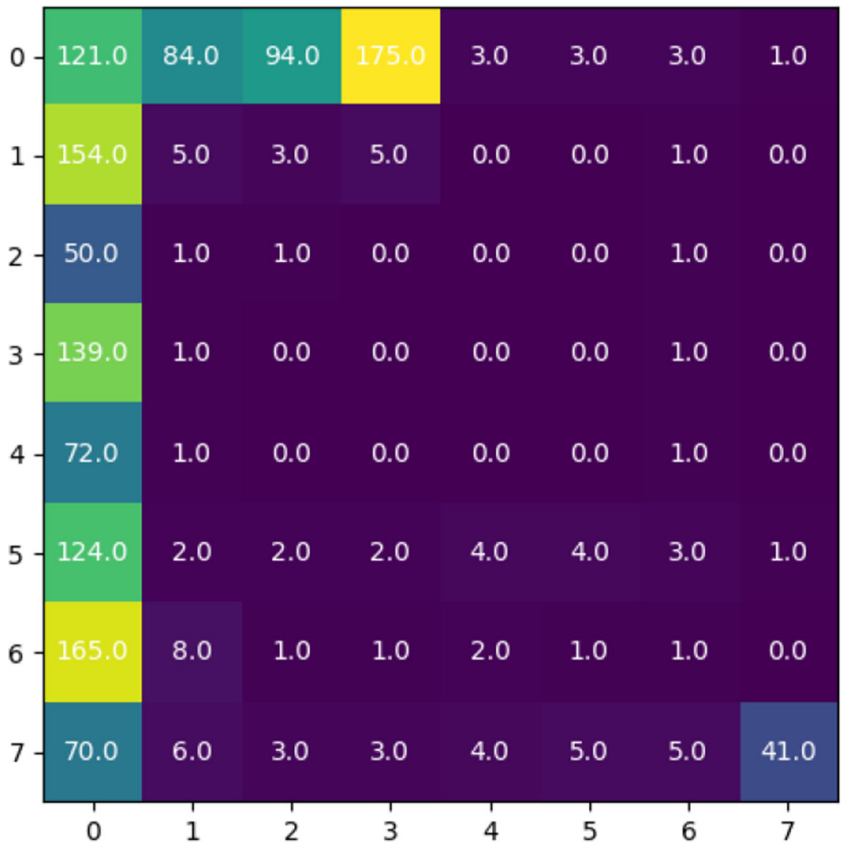


Fig. 10 Example of a heatmap reflecting the number of times a UAV passes through each cell. In the first actions UAVs take from the beginning, the model took the wrong sequence of initial movements. This caused the UAVs to have a preference for crossing the left edge, thus consuming most of the time. In this way, many cells inside are left unvisited

The improvement of having more errors at the beginning comes from the fact that the map exploration tree is covered faster due to the simultaneous flight of the UAVs. As the exploration tree is covered faster, more information is extracted. Therefore, usually more UAVs in a swarm means that the task is performed faster in future episodes (Table 4). If it is slower, it may be an indicator that more episodes are needed to obtain a better solution. The computational cost is very high considering that it is only a few seconds or minutes slower.

Table 5 Summary table with the p-values resulting from the Saphiro-Wilk test [52]

Distribution	p-value
Baseline	0.5013
Global ANN for 2 UAVs	0.2596
Global ANN for 3 UAVs	0.6719
Global ANN for 4 UAVs	0.4792
Global ANN for 5 UAVs	0.4287
2 UAVs with an ANN per UAV	0.2312
3 UAVs with an ANN per UAV	0.4887
4 UAVs with an ANN per UAV	0.7522
5 UAVs with an ANN per UAV	0.1788

For a significance level of 5%, all distributions are Gaussian

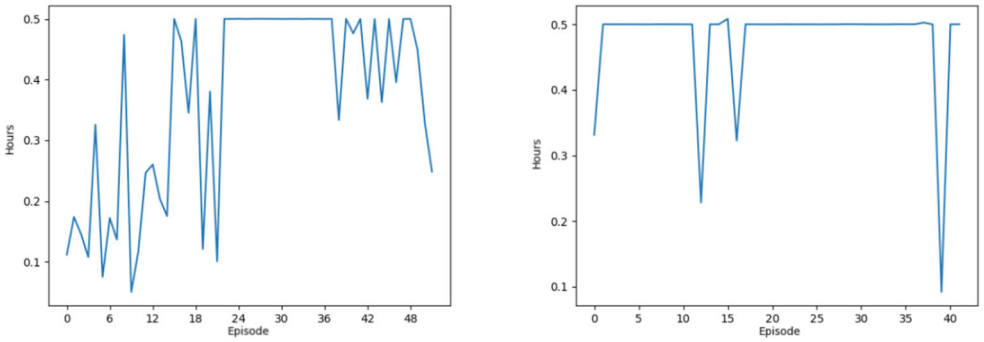
5 Discussion

This paper, like the other publications in Section 2, suggests the usage of ANNs based on dense layers. This type of layer has also shown the ability to coordinate groups of UAVs. Some authors, such as Liu et al., have already tested the capabilities of fully connected ANNs in Path Planning problems with UAVs. Moreover, being trained in each case with the memory of each UAV it seems to be able to assign correct actions to the UAVs without extracting spatial information from the map like Convolutional Neural Networks (CNN). As a result, it appears that using networks that need automatic feature extraction, such as CNN, as employed by other authors, is no longer necessary. Because they have already extracted spatial features from the maps, these networks have a slower training time. It may mean that

Table 6 Comparative table of the p-values obtained by performing an ANOVA test [53]

	p-value
All distributions	0.9698
All distributions except baseline	0.9968

Even without the Baseline distribution (1 UAV) the distributions are still significantly different at a 5% significance level



(a) 6×6 cell map (b) 8×8 cell map

Fig. 11 Example of two plots with the evolution of the hours (y-axis) consumed as the episodes elapse (x-axis) for different maps. In both cases, it can be seen that the solution with the shortest time can be followed by episodes with worse results. The optimal results are those episodes with the shortest duration. The shortest time implies that it is the one with the least number of incorrect actions and the least number of loops

the most important thing may not be the spatial relationship of the map, but the sequence of movements of each UAV without the noise of the actions taken by the other UAVs.

As in other papers in the state of the art, the system has been tested on squared cell maps. Unlike the other papers, it has been tested on different map sizes, not on fixed-size maps [54, 55]. The maps used do not present additional information, like those used in the other papers. That is, it is not necessary to add more information, such as targets or distance maps, so it is not necessary to make previous studies of the map.

Since using a single global ANN for all UAVs usually requires less time than using one ANN per UAV, this indicates that the appropriate configuration is to use a global ANN. This means that paths calculated using a global network have fewer errors and loops, indicating that the paths are as direct as possible. The more direct they are, the shorter they are, therefore they are more optimal. In other multi-agent problems global ANNs were the best option, like in the paper by Mnih et al. with their A3C algorithm [56]. Despite this, some authors, including Wang et al., have proved the effectiveness of systems that use an ANN per UAV [57].

The overall percentage of correct actions taken decreases with the increase in UAVs. This is due to the fact that in the first episodes too many wrong actions are taken because the agents do not have much knowledge. Despite this, the solution is sometimes reached in less time due to the simultaneity of their flight and, the more UAVs, the more fault tolerance is ensured in case a UAV not being able to continue its flight.

The time taken to explore the entire map is strongly affected by the number of cells on the map. The rise in the time taken exceeds the growth of the number of cells of the maps used in the experiments. This is mainly due to the size

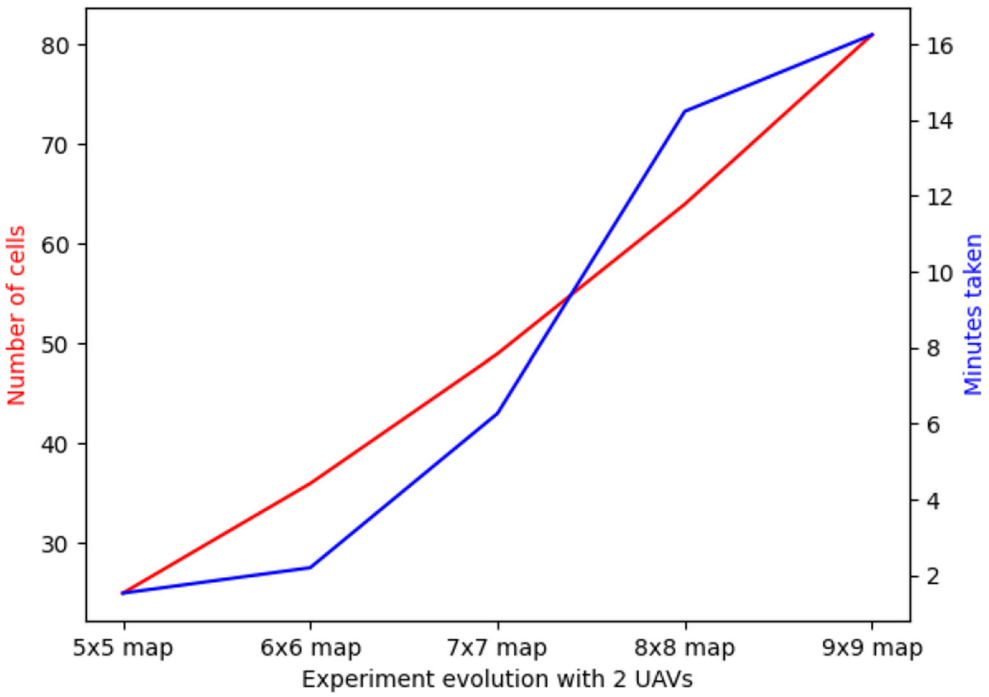
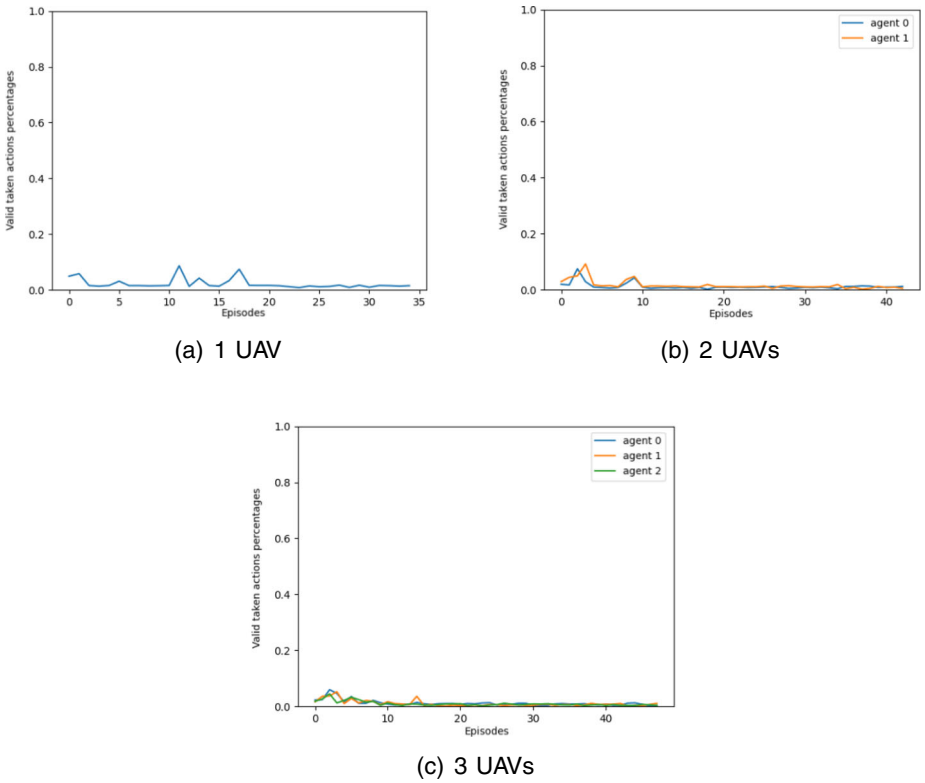


Fig. 12 Comparison of the growth of the curve of the time taken for each map with respect to the growth of the curve of the number of cells contained in each map. The growth is greater in the time curve. Specifically, starting from the map of 8×8 cells



(a) 1 UAV (b) 2 UAVs

(c) 3 UAVs

Fig. 13 Percentage of non-error actions (y-axis) taken by each UAV over the course of the episodes (x-axis). Each line of a different color symbolizes a UAV. The more UAVs the lower the percentage. This percentage is affected by the sum of the errors of all UAVs together. When taking the first actions UAVs make many mistakes. This accumulation of errors hurts the percentage of valid actions taken by the UAVs as a whole

of the exploration space that ANN faces in order to find the best paths. The greater the number of cells, the larger the space. In addition, we have to add the sequences of the UAV paths, whether they are one or more. That is, a path has to be a correct sequence of adjacent and fully navigable cells, which adds complexity to the system by having to maintain this consistency as there are more and more cells. This is why the higher the number cells, the more time the ANN requires for obtaining correct pathways.

6 Conclusions

This paper puts forward a system capable of calculating the paths with the shortest flight time for UAV swarms using Q-Learning [35] techniques. To enhance the capabilities of these techniques, decision-making is done with the help of fully connected ANNs. Employing a single global ANN for all UAVs presents more solutions in less time. Finding models that find solutions quickly makes the system more portable to different systems. In this way, users will find it more convenient to use since money does not have to be spent on expensive systems. Typically, the cost savings can be invested in improving UAV features such as battery life by users.

The system is capable of achieving satisfactory results with squared cell maps of different sizes. The evolution of the time required to find a solution increases faster than the increase in the number of cells in each experiment regardless of the number of UAVs in the swarm. Therefore, it is necessary to adapt the size of the map to the activity to be carried out in order to get the best results possible. Tasks that imply high altitude do not need as many cells because their sensors capture a large part of the terrain in each cell. Reducing the number of cells allows the system to make better and faster decisions due to the smaller exploration space. Other state of the art systems divide the map into a fixed size, which is typically a very large number of cells in the case of a very large map. Because there are more alternative optimal paths to explore for the map, the time it takes to find a solution and the computational cost of processing the maps are higher, and in many cases unnecessary.

It is not necessary to provide additional information on the map to direct the paths. Therefore, the system can calculate the optimal paths using only the information of the cell map, the current position of the UAVs on the map, and the evolution of the flight paths along the map. Using so little information avoids having to know the terrain in advance. If information is to be added to guide the UAV paths, it is necessary to make such a study. Therefore, many users may end up discarding the use of the system due to this additional difficulty. On the other hand, if it is necessary to guide the paths, the system can be biased because user errors can be made that prevent better paths from being found. The disadvantage of not employing targets, as other authors in

the state of the art have done, is that the algorithm is scan-dependent. Because it is so reliant, it is vital to establish algorithm parameters that are as accurate as possible in order to minimize problems with path computations.

The ideal swarm size can also be determined by looking at the change in the time it takes to fly through a map for each swarm size. For example, if a swarm takes less time to do a task than a larger swarm would, it is understood that investing in more UAVs is unnecessary because the task can be solved in less time and at a lower cost with fewer UAVs. In other publications, authors test their systems with a fixed number of UAVs and do not compare this to testing with just one UAV. Furthermore, if the terrain is relatively small, having a large number of UAVs may be excessive and counterproductive. The more UAVs used, the more paths must be computed. If fewer UAVs are used, however, outcomes can be achieved in less time and with less computational resources. Furthermore, using fewer UAVs minimizes the risk of collision.

Due to the atomicity of the movements that each UAV can make, it is not necessary to make a smooth path. Unlike other publications in the state of the art [58], it is not necessary to spend computational time smoothing the path. In addition to this advantage, the movements are simpler if there is no path smoothing, so it is not necessary to compute parameters such as the UAV's yaw angle or tilt angle. On the other hand, paths without smoothing have sharper turns that increase the UAV's battery consumption, so there is no guarantee that consumption is minimized as much as possible.

Finally, the limitations of this system include the fact that the flight height is not considered in the calculation of the paths. In theory, this is not a problem if the working height is high enough to avoid obstacles. On the other hand, if two UAVs flying at the same altitude are in the same cell, the wind thrust or turbulence that one may generate to the other is not considered. These winds or turbulences do not usually distort the paths to a great extent, but they can increase the battery consumption because the UAV needs more power to be able to overcome these environmental disturbances.

7 Future work

This paper is a starting point, laying out some bases for the creation of other systems capable of working on different maps. In this way, generic systems with commercial potential can be obtained.

In the future, efforts will be made to improve these results and a study on the optimal initial distribution of the UAVs on the map should be carried out. Also, models will be trained on cell maps optimally divided according to the resolution of the UAV cameras.

Future developments will include experiments with 3D maps in which more movements, such as pitch and roll, will be possible.

Experiments will be made with maps with obstacles in order to help agents learn how to reduce the risks during the flight. Obstacles can be fixed obstacles (trees, poles, etc.) or dynamic obstacles (birds, other UAVs, etc.).

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. Funding for open access charge: Universidade da Coruña/CISUG. This work is supported by Instituto de Salud Carlos III, grant number PI17/01826 (Collaborative Project in Genomic Data Integration (CICLOGEN) funded by the Instituto de Salud Carlos III from the Spanish National plan for Scientific and Technical Research and Innovation 2013–2016 and the European Regional Development Funds (FEDER)—"A way to build Europe.". This project was also supported by the General Directorate of Culture, Education and University Management of Xunta de Galicia ED431D 2017/16 and "Drug Discovery Galician Network" Ref. ED431G/01 and the "Galician Network for Colorectal Cancer Research" (Ref. ED431D 2017/23). This work was also funded by the grant for the consolidation and structuring of competitive research units (ED431C 2018/49) from the General Directorate of Culture, Education and University Management of Xunta de Galicia, and the CYTED network (PCI2018_093284) funded by the Spanish Ministry of Ministry of Innovation and Science. This project was also supported by the General Directorate of Culture, Education and University Management of Xunta de Galicia "PRACTICUM DIRECT" Ref. IN845D-2020/03.

Availability of Data and Material Not applicable

Code Availability Source code and a Docker container are available at: https://github.com/TheMVS/uav_swarm_reinforcement_learning https://hub.docker.com/r/themvs/uav_swarm_reinforcement_learning

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Albani D, IJsselmuiden J, Haken R, Trianni V (2017) Monitoring and mapping with robot swarms for agricultural applications. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, pp 1–6
2. Huuskonen J, Oksanen T (2018) Soil sampling with drones and augmented reality in precision agriculture. Comput Electron Agric 154:25–35

3. Corte APD, Souza DV, Rex FE, Sanquetta CR, Mohan M, Silva CA, Zambrano AMA, Prata G, Alves de Almeida DR, Trautenmüller JW, Klauberg C, de Moraes A, Sanquetta MN, Wilkinson B, Broadbent EN (2020) Forest inventory with high-density uav-lidar: Machine learning approaches for predicting individual tree attributes. Comput Electron Agric 179:105815
4. Bocchino R, Canham T, Watney G, Reder L, Levison J (2018) F prime: an open-source framework for small-scale flight software systems
5. Rabinovitch J, Lorenz R, Slimko E, Wang K-SC (2021) Scaling sediment mobilization beneath rotorcraft for titan and mars. Aeolian Res 48:100653
6. Liu J, Wang W, Wang T, Shu Z, Li X (2018) A motif-based rescue mission planning method for uav swarms usingan improved picea. IEEE Access 6:40778–40791
7. Yeaman ML, Yeaman M (1998) Virtual air power: a case for complementing adf air operations with uninhabited aerial vehicles. Air Power Studies Centre
8. Zhao Y, Zheng Z, Liu Y (2018) Survey on computational-intelligence-based uav path planning. Knowl-Based Syst 158:54–64
9. Campion M, Ranganathan P, Faruque S (2018) A review and future directions of uav swarm communication architectures. In: 2018 IEEE international conference on electro/information technology (EIT). IEEE, pp 0903–0908
10. Bonabeau E, Meyer C (2001) Swarm intelligence: A whole new way to think about business. Harvard Bus Rev 79(5):106–115
11. Puente-Castro A, Rivero D, Pazos A, Fernandez-Blanco E (2021) A review of artificial intelligence applied to path planning in uav swarms. Neural Comput Appl:1–18
12. Wiering M, Van Otterlo M (2012) Reinforcement learning. Adapt Learn Optim 12:3
13. Baldazo D, Parras J, Zazo S (2019) Decentralized multi-agent deep reinforcement learning in swarms of drones for flood monitoring. In: 2019 27th European signal processing conference (EUSIPCO). IEEE, pp 1–5
14. Yang Q, Jang S-J, Yoo S-J (2020) Q-learning-based fuzzy logic for multi-objective routing algorithm in flying ad hoc networks. Wirel Pers Commun:1–24
15. Roudneshin M, Sizkouhi AMM, Aghdam AG (2019) Effective learning algorithms for search and rescue missions in unknown environments. In: 2019 IEEE international conference on wireless for space and extreme environments (WiSEE). IEEE, pp 76–80
16. Luo W, Tang Q, Fu C, Eberhard P (2018) Deep-sarsa based multi-uav path planning and obstacle avoidance in a dynamic environment. In: International conference on sensing and imaging. Springer, pp 102–111
17. Speck C, Bucci DJ (2018) Distributed uav swarm formation control via object-focused, multi-objective sarsa. In: 2018 annual american control conference (ACC). IEEE, pp 6596–6601
18. Davis L (1991) Handbook of genetic algorithms
19. Duan F, Li X, Zhao Y (2018) Express uav swarm path planning with vnd enhanced memetic algorithm. In: Proceedings of the 2018 international conference on computing and data engineering. ACM, pp 93–97
20. Zhou Z, Luo D, Shao J, Xu Y, You Y (2020) Immune genetic algorithm based multi-uav cooperative target search with event-triggered mechanism. Phys Commun 41:101103
21. Olson JM, Bidstrup CC, Anderson BK, Parkinson AR, McLain TW (2020) Optimal multi-agent coverage and flight time with genetic path planning. In: 2020 international conference on unmanned aircraft systems (ICUAS). IEEE, pp 228–237
22. Huang T, Wang Y, Cao X, Xu D (2020) Multi-uav mission planning method. In: 2020 3rd international conference on unmanned systems (ICUS). IEEE, pp 325–330
23. Perez-Carabaza S, Besada-Portas E, Lopez-Orozco JA, Jesus M (2018) Ant colony optimization for multi-uav minimum time search in uncertain domains. Appl Soft Comput 62:789–806
24. Xie R, Meng Z, Zhou Y, Ma Y, Wu Z (2019) Heuristic q-learning based on experience replay for three-dimensional path planning of the unmanned aerial vehicle. Sci Prog:0036850419879024
25. Rummery GA, Niranjan M (1994) On-line q-learning using connectionist systems, vol 37. University of Cambridge, Department of Engineering Cambridge, England
26. Liu C, Xie W, Zhang P, Guo Q, Ding D (2019) Multi-uavs cooperative coverage reconnaissance with neural network and genetic algorithm. In: Proceedings of the 2019 3rd high performance computing and cluster technologies conference. ACM, pp 81–86
27. Vijayakumari DM, Kim S, Suk J, Mo H (2019) Receding-horizon trajectory planning for multiple uavs using particle swarm optimization. In: AIAA Scitech 2019 Forum, p 1165
28. Li Q, Gama F, Ribeiro A, Prorok A (2020) Graph neural networks for decentralized multi-robot path planning. In: 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 11785–11792
29. Shah S, Dey D, Lovett C, Kapoor A (2018) Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: Field and service robotics. Springer, pp 621–635
30. Chaudhuri D, Samal A (2007) A simple method for fitting of bounding rectangle to closed regions. Pattern Recogn 40(7):1981–1989
31. Kaelbling LP, Littman ML, Moore AW (1996) Reinforcement learning: A survey. J Artif Intell Res 4:237–285
32. Sutton RS, Barto AG (2018) Reinforcement learning: An introduction. MIT press
33. Van Hasselt H, Wiering MA (2007) Reinforcement learning in continuous action spaces. In: 2007 IEEE international symposium on approximate dynamic programming and reinforcement learning. IEEE, pp 272–279
34. Watkins ChristopherJCH, Dayan P (1992) Q-learning. Mach Learn 8(3-4):279–292
35. Sutton RS, Precup D, Singh SP (1998) Intra-option learning about temporally abstract actions. In: ICML, vol 98, pp 556–564
36. Gläscher J, Daw N, Dayan P, O'Doherty JP (2010) States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning. Neuron 66(4):585–595
37. Michie D, Spiegelhalter DJ, Taylor CC et al (1994) Machine learning. Neural Stat Classif 13
38. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
39. Fan J, Wang Z, Xie Y, Yang Z (2020) A theoretical analysis of deep q-learning. In: Learning for Dynamics and Control. PMLR, pp 486–489
40. Albawi S, Mohammed TA, Al-Zawi S (2017) Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET). IEEE, pp 1–6
41. Huang G, Liu Z, Van DML, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708
42. Marreiros AC, Daunizeau J, Kiebel SJ, Friston KJ (2008) Population dynamics: variance and the sigmoid activation function. Neuroimage 42(1):147–157
43. Hinton G, Srivastava N, Swersky K (2012) Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. Cited on 14(8)
44. Kimura H, Yamamura M, Kobayashi S (1995) Reinforcement learning by stochastic hill climbing on discounted reward. In: Machine Learning Proceedings 1995. Elsevier, pp 295–303

45. Foerster J, Nardelli N, Farquhar G, Afouras T, Torr PhilipHS, Kohli P, Whiteson S (2017) Stabilising experience replay for deep multi-agent reinforcement learning. In: International conference on machine learning. PMLR, pp 1146–1155
46. Liu R, Zou J (2018) The effects of memory replay in reinforcement learning. In: 2018 56th annual allerton conference on communication, control, and computing (Allerton). IEEE, pp 478–485
47. Gago J, Douthe C, Coopman RE, Gallego PP, Ribas-Carbo M, Flexas J, Escalona J, Medrano H (2015) Uavs challenge to assess water stress for sustainable agriculture. Agric Water Manag 153:9–19
48. (2016). (EASA) EASA. Gtf. https://www.easa.europa.eu/sites/default/files/dfu/GTF-Report_Issue2.pdf. [Online; accessed 19-July-2008]
49. Zhang C, Zhen Z Z, Wang D, Li M (2010) Uav path planning method based on ant colony optimization. In: 2010 Chinese Control and Decision Conference, pp 3790–3792
50. Wang Z, Li G, Ren J (2021) Dynamic path planning for unmanned surface vehicle in complex offshore areas based on hybrid algorithm. Comput Commun 166:49–56
51. Li W, Yang B, Song G, Jiang X (2021) Dynamic value iteration networks for the planning of rapidly changing uav swarms. Front Inf Technol Electron Eng:1–10
52. Razali NM, Wah YB et al (2011) Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. J Stat Model Anal 2(1):21–33
53. Hecke TV (2012) Power study of anova versus kruskal-wallis test. J Stat Manag Syst 15(2-3):241–247
54. Albani D, Manoni T, Arik A, Nardi D, Trianni V (2019) Field coverage for weed mapping: toward experiments with a uav swarm. In: International conference on bio-inspired information and communication. Springer, pp 132–146
55. Venturini F, Mason F, Pase F, Chiariotti F, Testolin A, Zanella A, Zorzi M (2020) Distributed reinforcement learning for flexible uav swarm control with transfer learning capabilities. In: Proceedings of the 6th ACM workshop on micro aerial vehicle networks, systems, and applications, pp 1–6
56. Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K (2016) Asynchronous methods for deep reinforcement learning. In: International conference on machine learning. PMLR, pp 1928–1937
57. Wang L, Wang K, Pan C, Xu W, Aslam N, Hanzo L (2020) Multi-agent deep reinforcement learning-based trajectory planning for multi-uav assisted mobile edge computing. IEEE Trans Cogn Commun Netw 7(1):73–84
58. He W, Qi X, Liu L (2021) A novel hybrid particle swarm optimization for multi-uav cooperate path planning. Appl Intell:1–15

Expert Systems With Applications 235 (2024) 121240

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Q-Learning based system for Path Planning with Unmanned Aerial Vehicles swarms in obstacle environments

Alejandro Puente-Castro [a,*], Daniel Rivero [a], Eurico Pedrosa [b], Artur Pereira [b], Nuno Lau [b], Enrique Fernandez-Blanco [a]

[a] Faculty of Computer Science, CITIC, University of A Coruna, A Coruna, 15007, Spain
[b] IEETA, DESI, LASI, University of Aveiro, Portugal

ARTICLE INFO

Keywords:
UAV
Swarm
Obstacle
Path Planning
Reinforcement learning
Artificial Neural Network

ABSTRACT

Path Planning methods for the autonomous control of Unmanned Aerial Vehicle (UAV) swarms are on the rise due to the numerous advantages they bring. There are increasingly more scenarios where autonomous control of multiple UAVs is required. Most of these scenarios involve a large number of obstacles, such as power lines or trees. Despite these challenges, there are also several advantages; if all UAVs can operate autonomously, personnel expenses can be reduced. Additionally, if their flight paths are optimized, energy consumption is reduced, leaving more battery time for other operations. In this paper, a Reinforcement Learning-based system is proposed to solve this problem in environments with obstacles by utilizing Q-Learning. This method allows a model, in this case, an Artificial Neural Network, to self-adjust by learning from its mistakes and successes. Regardless of the map's size or the number of UAVs in the swarm, the goal of these paths is to ensure complete coverage of an area with fixed obstacles for tasks like field prospecting. Setting goals or having any prior information apart from the provided map is not required. During the experimentation phase, five maps of varying sizes were used, each with different obstacles and a varying number of UAVs. To evaluate the quality of the results, the number of actions taken by each UAV to complete the task in each experiment was considered. The results indicate that the system achieves solutions with fewer movements as the number of UAVs increases. An increasing number of UAVs on a map lead to solutions in fewer moves. The results have been compared, and a statistical significance analysis has been conducted on the proposed model's outcomes, demonstrating its capabilities. Thus, it is shown that a two-layer Artificial Neural Network used to implement a Q-Learning algorithm is sufficient to operate on maps with obstacles.

1. Introduction

New uses for swarms of unmanned aerial vehicles (UAVs) are being developed to solve different industrial and emergency problems (Albani, IJsselmuiden, Haken, & Trianni, 2017; Bocchino, Canham, Watney, Reder, & Levison, 2018; Corte et al., 2020; Huuskonen & Oksanen, 2018; Liu, Wang, Wang, Shu, & Li, 2018; Rabinovitch, Lorenz, Slimko, & Wang, 2021). The advantages provided by UAVs, such as their low cost, excellent mobility, safety, and convenient size for some maneuvers, are the main reasons for their growing popularity (Yeaman & Yeaman, 1998). All these advantages are offered by the wide variety of UAVs that exist to fulfill every need. This variety allows for the integration of different types of sensors with varying capabilities. The development of sensors for UAVs is on the rise, particularly in remote sensing (Noor, Abdullah, & Hashim, 2018). The flexibility in the characteristics of UAVs, such as their architecture or the sensors they accommodate, makes them popular tools for diverse needs (Austin, 2011).

However, UAVs have drawbacks, with the most significant one being power consumption, which reduces operational time. Due to their small size, it is challenging to acquire compact power sources with substantial capacities while also keeping the weight low. When the weight is minimal, flight operations benefit from extended flight time availability.

The limitations on flight time imposed by batteries can be mitigated when groups or swarms are employed. In essence, as flight paths

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: a.puentec@udc.es (A. Puente-Castro), daniel.rivero@udc.es (D. Rivero), efp@ua.pt (E. Pedrosa), artur@ua.pt (A. Pereira), nunolau@ua.pt (N. Lau), enrique.fernandez@udc.es (E. Fernandez-Blanco).

https://doi.org/10.1016/j.eswa.2023.121240
Received 27 April 2023; Received in revised form 16 August 2023; Accepted 16 August 2023
Available online 23 August 2023

become shorter with the simultaneous operation of multiple UAVs, numerous tasks can be completed more swiftly. This approach reduces the likelihood of UAVs' batteries being insufficient to sustain their flight over the terrain. The occurrence of UAVs stopping midway through an operation due to diminished energy availability is lessened, thereby reducing the risk of mid-air disruptions.

Similar to any form of robotic swarm, UAV swarms can be utilized in the real world for various activities, just as they would in their individual applications. The primary advantage of the swarm robotics technique lies in its robustness, which manifests in numerous ways. Firstly, a swarm can self-organize or dynamically reorganize how individual robots are deployed, as it comprises many relatively simple agents that are not predetermined for specific roles or duties. Additionally, and for the same reasons, the swarm technique is highly resilient to individual agent failures. There is no singular point of common-mode failure or vulnerability within the swarm due to the entirely decentralized control. In contrast to the substantial technical investment required for achieving fault tolerance in traditional robotic systems, one might argue that the elevated level of robustness observed in UAV swarms is inherent to the swarm robotics methodology (Sahin & Winfield, 2008).

The number of UAV operators required for the initial flight tests with swarms was equivalent to the number of UAVs, significantly increasing the operational costs when employed in groups. Recent advancements have been made in the development of algorithms (Zhao, Zheng, & Liu, 2018) and communications (Campion, Ranganathan, & Faruque, 2018) that allow for the control of the entire swarm by just one person operating the systems. These advancements facilitate more efficient and rapid communication among UAVs, along with improved calculations for collision avoidance paths. This reduces the requirement for human intervention in hazardous situations. Consequently, the latest approaches are geared towards achieving autonomous control of the entire swarm. Flight paths need to be computed at minimal cost while maximizing efficiency. This is known as the Path Planning Problem (Aggarwal & Kumar, 2020), in which the aim is to plan the sequence of movements of robots such as UAVs. Given the often low altitude of their operations, UAVs must navigate around obstacles within the flight area. Consequently, flight path calculations must account for these obstacles and the anticipated positions of all swarm UAVs, in order to prevent collisions among fleet members. The objective is to devise paths that are optimized while circumventing obstacles and other UAVs.

To deal with the complexity of this kind of development, different algorithms are offered in Swarm Intelligence (SI) (Kennedy, 2006). These algorithms aim to coordinate a substantial number of agents concurrently. This coordination relies on a collective of individual actors operating in a self-organized and cohesive manner, while adhering to fundamental, common rules (Bonabeau & Meyer, 2001). In essence, each UAV within the swarm acts as an individual actor. Each actor possesses its information, and its behavior is influenced by its information, the system's rules, and the information shared by other actors. This coordinated behavior is aimed at achieving an objective in the most effective manner (Stentz, 1997).

Certain Path Planning algorithms find utility in military applications. Meanwhile, the scope of civilian applications is relatively restricted, primarily encompassing pursuits or goal-oriented tasks, such as mapping routes through urban areas (Puente-Castro, Rivero, Pazos, & Fernandez-Blanco, 2021). Despite the multitude of potential applications, there exists a scarcity of technologies explicitly designed for agricultural and forestry purposes, particularly those aimed at enhancing the efficiency of field prospecting tasks.

The objective of this research is to create a system that addresses the Path Planning problem within 2D grid-based maps featuring static obstacles and varying quantities of UAVs, accomplished through the utilization of Q-Learning techniques bolstered by Artificial Neural Networks (ANN). Consequently, the principal contributions of this study can be outlined as follows:

1. An innovative Q-Learning-based system that can determine the best possible flying path for a UAV swarm to cover as much area as possible during prospecting activities.
2. A system that can estimate the flight path of any number of UAVs on any sized map with varied sets of obstacles with different shapes and without additional map information such as targets or potential fields.
3. A system capable of calculating paths without the need for a subsequent smoothing stage.
4. A statistical analysis of the results of using a single ANN for each UAV against a global ANN for all UAVs under the same conditions.
5. A path optimization criterion for Q-Learning not dependent on aircraft architecture and capabilities.

The structure of this paper is as follows: An overview of the state of the art is provided in Section 2; a description of the inherent aspects for solving Path Planning problems is developed in Section 3; a description of the technical aspects required for the development of the proposed method is presented in Section 4; a summary of the results of the experimental process is provided in Section 5; the conclusions drawn after evaluating the results, and the possible works and studies to derive the problem to be addressed are provided in Section 6.

2. Background

2.1. Path Planning problems

Path Planning problems involve determining geometric paths for vehicles or robots to follow a set of milestones to reach a designed goal (Gasparetto, Boscariol, Lanzutti, & Vidoni, 2015). Different authors have focused on the development of systems to solve these problems for several years (Patle, Pandey, Parhi, Jagadeesh, et al., 2019). All these authors have employed an extensive array of techniques, spanning from conventional methodologies to Artificial Intelligence approaches (Karur, Sharma, Dharmatti, & Siegel, 2021).

Kong, Nie, and Xu (2022) have put forth a Genetic Algorithm (GA) for controlling swarms within 3D environments. This algorithm underwent testing in a simulator, with results indicating its capability to evade convergence to local maxima. However, it is noteworthy that this approach may entail a relatively higher computational expense in comparison to alternative methods. Liu (2022) have proposed another GA for 3D environments with terrain obstacles. Their method is proficient in deriving smoothed paths without necessitating a subsequent smoothing phase. Additionally, GA can serve as a complementary tool to other algorithms. In their research, they present a system wherein the fitness function is predicated on the UAVs' distance to the final target. This metric, though simplistic, may lead to sluggish approaches if UAVs adopt a spiral trajectory, consequently incurring substantial battery consumption during gradual approaches. In general, the flight environment strongly influences the behavior of the algorithm. That is, a 3D map implies controlling the height of the aircraft while a 2D grid-map implies knowing the state of each cell. To conclude with these techniques, there is a branch within Evolutionary Computation (EC) known as Swarm Intelligence (SI) (Kennedy, 2006) that seeks to mimic the collective behavior of natural systems. For example, Xu, Li, Zhou, Mao, and Huang (2022) have introduced the utilization of GA to optimize a system grounded in the Wolf Pack Algorithm, a purely SI technique, for the coordination of multiple UAVs. Their research illustrates the effectiveness and efficiency of their approach in contrast to alternative methods. However, it is worth noting that they have not presented specific examples of the testing environments for these systems.

Among the category of pure SI techniques, a significant degree of diversity exists. While not as extensively recognized as the methods

mentioned earlier, SI techniques have showcased their prowess in optimizing a wide array of problems of diverse natures. This is attributed to their bio-inspired collective behaviors (Minh, Sang-To, Theraulaz, Wahab, & Cuong-Le, 2023; Sang-To, Le-Minh, Mirjalili, Wahab, & Cuong-Le, 2022; Sang-To, Le-Minh, Wahab, & Thanh, 2023). For example, Yang, Zhang, Zhang, and Xiangmin (2019) have harnessed Particle Swarm Optimization (PSO) in conjunction with a voting mechanism to manage multi-UAV control. They have devised an intricately structured voting system tailored to the conventional PSO method, further refining its spatial aspects. Moreover, their innovative approach incorporates time considerations, effectively generating collision-free routes for multiple UAVs within a comparable timeframe. Similarly, Pamosoaji, Piao, and Hong (2019) have employed this algorithm for UAV control. They have carefully factored in the constraints associated with slower aircraft to minimize their flight duration. In their published work, they have demonstrated the algorithm's proficiency in deriving flight paths. However, it is worth noting that they have not provided a quantifiable assessment of the effectiveness or satisfaction of these generated paths. Jain, Yadav, Prakash, Shukla, and Tiwari (2019) put forward the utilization of the Multiverse Optimizer algorithm (MVO) to govern the behavior of multiple UAVs and juxtapose it with the application of a single UAV. This showcases the system's ability to generalize across scenarios. While the system boasts considerable capabilities, it is noteworthy that significant environmental factors impacting aerial operations are not factored into the approach. In a broader sense, one of the predominant limitations of SI techniques is their inclination to converge towards local optima (Yang, 2014). In addition, describing the collective behavior of natural systems is very difficult; it may not be realistic.

2.2. Reinforcement Learning and Q-Learning in Path Planning

Another of the most commonly used set of techniques is Reinforcement Learning (RL) techniques (Puente-Castro, Rivero, et al., 2021). An example of these Artificial Intelligence (AI) techniques can be seen in the research of Qiu, Xu, Wang, Yang, and Liao (2022) where they have implemented an Actor–Critic Reinforcement Learning algorithm to achieve concurrent control of multiple UAVs. Notably, each UAV exclusively possesses local information concerning the environment. This implies that each UAV solely retains its data and does not communicate any information with the other members of the group. Consequently, certain environmental details might be overlooked, or alternatively, some information could potentially be redundantly captured. Additionally, Wei, Huang, et al. (2022) have employed the Actor–Critic RL technique for collaborative data collection across expansive regions. They have introduced a method for estimating energy consumption solely based on time, although a limitation arises from not accounting for the specific type of movement. Consequently, flights involving frequent changes in direction might consume more energy compared to linear flights. To circumvent the challenge of sparse rewards, a common issue in such scenarios, they have implemented an incentive mechanism. Incentives are also applied by Salimi and Pasquier (2021) for the control of a type of UAV group called flocks instead of swarms. In their paper, the authors mention utilizing environments featuring up to 50 obstacles, but unfortunately, they have not provided visual examples. Furthermore, it appears that they depend on the progression of rewards to gauge the system's functionality. However, this approach does not ensure optimal objective completion, as the system might become trapped in a local optimum. A more comprehensive assessment of the system's overall performance, including global-level results, would be necessary to gain deeper insights into its effectiveness.

Continuing with RL techniques, Chen, Dong, Shang, Wu, and Wang (2022) This approach offers a significant advantage, as simulated environments can incorporate intricate details and facilitate the eventual transition to real-world applications. However, it is important to acknowledge a limitation outlined in the paper. Specifically, their focus on cooperative environments. Such environments presuppose that all UAVs will collectively pursue a singular target simultaneously. While this simplifies certain aspects, it does constrain movement flexibility and neglects potential UAV failures or deviations from the uniform path. The use of simulated real environments is also considered by Tu and Juang (2023). In their paper, they utilize the widely adopted AirSim simulator to evaluate the performance of their RL-based system. However, they highlight a limitation in their approach: their system exclusively relies on ultrasonic sensors for obstacle avoidance. Consequently, their UAVs might not effectively detect obstacles constructed from sound-absorbing materials.

A very popular algorithm within RL is Q-Learning (Watkins & Dayan, 1992) and many authors have applied it. This algorithm searches greedily for the best action in each state based on a value given to each available action. By selecting actions with the highest assigned values, it assembles the most optimal sequence of moves. Therefore, it is imperative to accurately determine how to calculate the value attributed to each action. Souto, Alfaia, Cardoso, Araújo, and Francês (2023) have created a system based on Q-Learning, wherein they incorporate external variables unrelated to the UAVs to minimize energy consumption while computing UAV paths. They validate their system's efficacy through simulations conducted within realistic urban environments. The high level of realism exhibited by the simulated environment renders the system readily adaptable to real-world scenarios. Also, de Carvalho et al. (2022) focus on reducing energy consumption by applying Q-Learning techniques. One limitation of their approach is the absence of a defined metric for calculating this consumption. Instead, they have implemented a reward prioritization mechanism based on the type of turn executed by the UAV. It is worth noting that they only account for four specific types of turns based on their angles. Consequently, turns not encompassed within their prioritization framework are not taken into consideration.

2.3. Artificial Neural Networks in Path Planning

Still within AI, a widely used model is the Artificial Neural Network (ANN) (Rosenblatt, 1958). These models are based on Artificial Neurons and have demonstrated their ability to generalize knowledge (McCulloch & Pitts, 1943). Typically, these models are employed to enhance the computation of various essential factors required for path planning, as they can encapsulate a greater depth of knowledge compared to pre-defined formulas. An example of this is the paper by Shiri, Park, and Bennis (2020) where they use ANN to approximate the Hamilton–Jacobi–Bellman equation. Accordingly, the developed ANNs or algorithms reduce biases and can overcome the limitations imposed by the equation. Furthermore, this approach has facilitated the incorporation of wind dynamics into the system, enhancing its reliability and applicability in real-world environments. Another approach is that of Sanna, Godio, and Guglieri (2021), where they use ANN to obtain the best actions to be performed by UAVs. In this manner, they illustrate the system's capacity to acquire additional knowledge by contrasting it with both a non-parametric model and a conventional search model. A similar approach is that of Liu, Zheng, Qin, Zhang, and Yao (2022). Interestingly, they also juxtapose their approach with a classical search algorithm. In contrast to the earlier mentioned paper, the primary objective here is not to cover an entire map. Instead, the focus centers on computing a path between a source point and a destination point. It would indeed be intriguing to comprehend how their system performs in the absence of any indicators, such as those source and destination points. This versatility of not needing points to control the path is also regarded by de Castro et al. (2023). Furthermore, they put forth an alternative approach involving ANN, where it is trained to approximate a conventional search algorithm. This training process involves utilizing the output of the aforementioned algorithm. This innovative strategy allows them to combine the efficiency of a classical search algorithm with the real-time adaptive capabilities inherent in an ANN.

2.4. Artificial Neural Networks applied to Q-Learning in Path Planning

Although the techniques and models described above have demonstrated their capabilities in Path Planning problems, their strength is when used in combination. This is known as Deep Reinforcement Learning (DRL) and consists of training ANNs using RL techniques for Path Planning problems, which is a great advantage because it allows faster abstraction of knowledge in complex environments (Li, 2023).

The Deep Q-Learning or Deep Q-Network (DQN) (Mnih et al., 2015) is one of the most important DRL techniques (Clifton & Laber, 2020). In this case, ANNs are used to enhance the capabilities offered by Q-Learning for Path Planning. For example, Puente-Castro, Rivero, Pazos, and Fernandez-Blanco (2022) propose a dense two-layer ANN applying Q-Learning. The model they introduce is exclusively tested on obstacle-free maps, potentially presenting significant constraints when applied to maps with obstacles. The demonstrated operation is solely time-based, which means its performance would be contingent upon the hardware capabilities of the requisite equipment. Consequently, equipment with superior capabilities would yield improved times, but this would correspondingly entail higher costs. Dhuheir, Baccour, Erbad, Al-Obaidi, and Hamdi (2022) also propose an ANN for their system where they segment a map for each UAV to collect information taking into account latency constraints. While it is important to control such latencies, the test computer is a Rapsberry Pi which a very specific and limited model that, at the date of publication of the article, already has more recent and powerful versions. This is a major limitation because they employ convolutional ANNs, which are known for their high computational cost (Li, Liu, Yang, Peng, & Zhou, 2021). In the paper of Khalil and Rahman (2022) they try to go one step further by making a Federated Learning scheme with an Aggregator (Rieke, Hancox, Li, Milletari, Roth, Albarqouni, Bakas, Galtier, Landman, Maier-Hein, & et al., 2020) applied to a global ANN to converge earlier than those cases where the ANN is trained in the traditional way. Thus, the network acquires more variety of data in less time. The Aggregator module brings together the experience of the ANNs of each UAV that is retrieve individually by each UAV. Therefore, they can train UAVs that escape from hostile systems in the military domain. Within the topic of UAVs in hostile environments, Zhang, Zong, Zhang, Dou, and Tian (2022) propose a similar system but not based on Federated Learning. An added advantage over the previous work is that they test their system in a simulation depicting a complex urban environment, so that the capability of their system can be better seen. There are also publications that make use of DQN but with static targets. An example is the paper by Zhou, Liu, Li, Xu, and Shen (2021), where they address the planning of UAVs swarms with targets. This facilitates the path calculation, but increases the probability that the paths become too dependent on the targets. Similar is the case of Kong, Wang, Gao, and Yu (2023) where they have to establish an allowance threshold error in order to overcome these limitations. Therefore, they have used an ANN that is not only able to calculate the Q-values, but also the distribution of the movements taken by UAVs. Another example of ANN applied to Q-Learning is the work of Raja, Anbalagan, Narayanan, Jayaram, and Ganapathisubramaniyan (2019). Despite not presenting the findings, their paper claims that their technology is scalable to 100 UAVs. In addition to this, a generic system with significant commercial potential can be created by having a system that is scalable to any number of UAVs.

2.5. Summary and contributions

In summary, several papers in the state-of-the-art present a subsequent path-smoothing stage, such as Liu (2022), Susanto et al. (2021) or Correl (2016). By establishing this later step, sharp turns in the paths are modified to make them softer and more gradual. In this way, paths with softer curves are obtained, which lengthens the battery time. However, this process entails increased computational demands, and if the original path contained errors, they could potentially be propagated. Notably, the proposed system omits the path smoothing stage to minimize computation time and preserve the integrity of subsequent data collection, ensuring comprehensive coverage of the terrain during flight.

The authors in the state-of-the-art test their systems on different maps with obstacles but mostly with the same number of UAVs for that given map, like in the work of Kong et al. (2022). In contrast to their suggested model, the proposed system undergoes testing with varying numbers of UAVs to assess its adaptability across different group sizes. Furthermore, maps featuring obstacles of diverse shapes are introduced to ascertain that the system does not merely memorize the obstacle topologies.

Several models in the state-of-the-art require guide points, which can take the form of targets, potential fields, or other indicators. The utilization of these points necessitates the extraction or addition of information to the maps. The dynamic alteration of maps by incorporating new information carries the risk of introducing errors, which could consequently impact the generated paths. Therefore, it is necessary for the map representations chosen to be as complete as possible without the need to add information to them. In the proposed model, maps are tested where only the location of obstacles is indicated and no other information is added.

The main optimization criterion in every work is energy saving. Despite being a common purpose, there are different ways to determine consumption. They are all based on a criterion where an estimation of how much each vehicle can consume according to each type of movement is carried out but these are not equally expensive in different types of UAVs. This highlights the need for a versatile and comprehensive metric, such as the count of movements executed by each UAV. Consequently, a path is considered superior if it entails fewer movements. Moreover, while precise quantitative energy expenditure may remain elusive, it can be inferred that fewer movements inherently translate to reduced energy consumption.

3. Problem formulation

The main aim of this research is to develop a system capable of solving the Path Planning Problem for UAV swarms in maps with obstacles. In scenarios involving multiple vehicles like UAVs, successful Path Planning necessitates the consideration of various variables to ensure optimal efficiency, effectiveness, control, collaboration, and safety. Therefore, it becomes imperative to tackle the challenges posed by these variables, as they form integral components of the overarching goal.

This way of looking at a Path Planning problem as the union of different inherent problems is common in the literature (He, Qi, & Liu, 2021; Puente-Castro et al., 2022). Accordingly, the experimentation process is more precise and organized. The formulation of the Path Planning problem presented is divided into the following areas:

Flight Environments Set
UAV movements
Proposed Model Design
Model Optimization
Model Evaluation Metric

The main point to keep in mind is that solving some aspects of Path Planning problems involves employing simplifications of environment, movement, and other variables simplifications (Giesbrecht, 2004). In the real world, UAVs fly in complex continuous environments. These environments consist of an infinite number of points, and determining the optimal flight path involves exploring infinite combinations. To manage this complexity, the utilization of cell-based maps is a prevalent approach within the field. By dividing the map into finite cells, the exploration process involves fewer combinations, simplifying the task.

However, there is the limitation that the representation of the terrain is greatly simplified, so details that may be crucial for the paths to be calculated may be lost. Even setting up maps divided into cells can complicate the calculation of optimization criteria such as path length because it oversimplifies the representation of the real environment.

It is also necessary to take into account the movements of UAVs. Currently, these aircraft have great flight stability, but their movements are complex and result from the combination of other simpler movements (Susanto et al., 2021). Indeed, to address the challenges related to field coverage and to expedite path calculations, UAV movements are often simplified by being treated as atomic actions without accounting for curves or changes in altitude. Consequently, it is easier to determine whether or not a movement implies that a UAV flies over a cell. On the other hand, the major limitation is to involve abrupt changes in the path, so it may be the case that smooth curves are a better path.

A final limitation to take into account is the tendency of the proposed method in this paper to converge to local maxima by its nature (Jaakkola, Singh, & Jordan, 1994). Therefore, a situation may arise where a good solution is found but a better one is not found. Taking this into account, it is necessary to apply alternatives to reduce this risk.

4. Proposed method

4.1. Reinforcement learning

The solution for the Path Planning Problem for UAVs has been developed by applying Reinforcement Learning (RL) (Sutton & Barto, 2018). Similar to other computational techniques, this method eliminates the necessity of explicitly defining the desired behavior within the system. Instead, a specific component of the algorithm, referred to as an agent, acquires the intended behavior through a process of trial and error. This learning process unfolds within an interactive and dynamic environment, where the agent conducts various tests to adapt and internalize the optimal behavior (Kaelbling, Littman, & Moore, 1996; Wiering & Van Otterlo, 2012). The agent must exploit what it already knows in order to profit from rewards, but it must also explore in order to choose its future actions more wisely. The problem is that pursuing either exploration or exploitation solely would result in failure. The agent must test several different options and gradually favor the ones that seem to work the best. For each action on a stochastic task to gain a valid estimate of the expected reward, several trials must be made. In essence, achieving the ideal behavior requires a dynamic interplay between learning from past experiences and venturing into uncharted territories. All while considering the potential repercussions of the agent's actions on its surroundings.

The explicit consideration of the entire issue of a goal-directed agent interacting with an unpredictable environment is another important aspect of RL. Contrary to many techniques, RL does not take into account sub-problems without considering how they might relate to a bigger one. In other words, it addresses the problem "as a whole".

The learning method differs only slightly in most RL algorithms (Sutton & Barto, 2018). These strategies come in a variety of forms that let the systems handle a wide range of problems. It has been decided to employ a technique known as Q-Learning for this more appropriate to use this study (Watkins & Dayan, 1992). The biggest factor is that, in contrast to other variants, it does not require a model of the environment (model-free approach).

4.1.1. Q-Learning

The agents have to find and follow strategies that allow them to solve problems. These strategies are known as policies. The agents can use their experience to learn the values of all the policies in parallel even when they can only follow one policy at a time thanks to the traditional Q-Learning algorithms (Watkins & Dayan, 1992). The agent learns to follow a policy only through trial and error in this model-free approach (Gläscher, Daw, Dayan, & O'Doherty, 2010). The remarkable convergence property of Q-Learning, known as "greedy convergence", leads to the attainment of an optimal solution regardless of the decision-making policy. This characteristic classifies Q-Learning as an off-policy algorithm. In other words, it only bases its decisions on the agent's interactions with the environment around it. This design ensures the system's adaptability across diverse environments, eliminating the requirement to identify the best policy for each specific scenario. The "Q" in Q-learning stands for "quality" and endeavors to quantify the usefulness of a given action in procuring future rewards.

The most well-known benefit of Q-Learning over other RL techniques is that it allows for the comparison of predicted utility across different actions without the need for an environment model. That means the key factor that led to its selection for this study is how easily it learns and infers situations without requiring their modeling. These algorithms stand out from other RL approaches due to their fundamental distinction: they make decisions based on values stored within a table. These values are known as Q-values, and the table is referred to as a Q-table. The Q-values essentially represent the anticipated reward of an action within the specific context of the environment. From these values, the action with the highest value for each state is chosen. Typically, Bellman's equation (Eq. (1)) is combined with the system's prior predictions to train it. The equation has different elements: $Q(s,a)$ is the function that calculates the Q-value for the current state (s), of the set of states S, and for the given action (a), of the set of actions A, r is the reward of the action taken in that state and it is computed by the reward function $R(s,a)$, γ is the discount factor and $\arg\max_{a'}(Q(s',a'))$ is the maximum computed Q-value of the pair (s',a') represented as $Q(s',a')$. The pair (s',a') is a potential next state–action pair. (s' is the next state and it is given by the transition function $T(s,a)$ which returns the state resulting from the execution of the selected action. The a' is each one of the available actions.

$$Q(s,a) \leftarrow r + \gamma \times \max_{a'}(Q(s',a')) \tag{1}$$

With probability ϵ, a portion of the actions in a Q-Learning problem are made at random, and with probability $1-\epsilon$, the action with the greatest Q-value for that state is adopted. An episode is the series of actions that an agent performs for a certain ϵ until it achieves an end condition (task completion, end of time, etc.) (Shang & Li, 2022). The operation is started over at the beginning of each episode. During testing, episodes reduce the value of ϵ by a factor of reduction. In this approach, the decision of what to do is influenced more by the computed Q-values and less by chance. By considering the minimal value of ϵ, it is kept from becoming too close to zero and to avoid overfitting (Zhang, Vinyals, Munos, & Bengio, 2018).

The key to convergence in Q-Learning is that it is a variant of a Markov Decision Process (MDP). This process is artificially controlled and is known as the action-replay process (ARP) (Watkins & Dayan, 1992). It should be noted that this description assumes a representation of a look-up table, indicating that Q-Learning might not converge correctly for other representations. The requirement that an unlimited number of episodes for each beginning state and action must be included is the most significant implicit condition in the convergence.

Recently, a variant known as Deep Q-learning (DQN) (Mnih et al., 2015) has emerged as an alternative. This approach varies from traditional Q-Learning in that it aims to enhance the calculation of the Q table using Machine Learning (Michie, Spiegelhalter, Taylor, et al., 1994) or Deep Learning (LeCun, Bengio, & Hinton, 2015). The model may deduce the values of the Q table by abstracting sufficient knowledge. In some cases, Bellman's Equation bias concerns can be resolved in this way (Fan, Wang, Xie, & Yang, 2020).

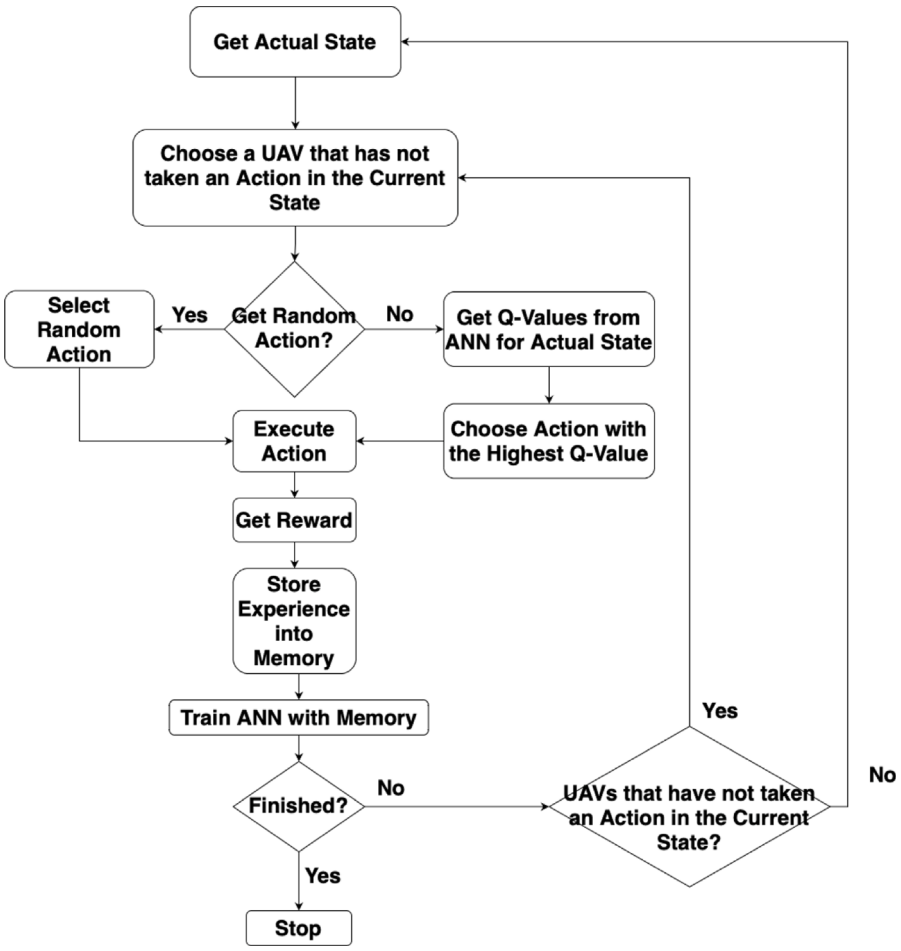


Fig. 1. Flowchart showing the steps that are followed within each episode of the proposed model. It shows how the ANN interacts with Q-Learning. That is, the ANN learns from the experience gained from performing actions on one or more UAVs.

4.1.2. Artificial Neural Network

One of the key points of this paper is the use of ANNs to enhance Q-learning by approximating the Bellman's equation (Krogh, 2008). The authors of this work choose a two-layer fully connected ANN. Unlike convolutional deep ANNs (Albawi, Mohammed, & Al-Zawi, 2017) that other authors have suggested in their studies, it is not assumed that the neighborhood of a cell provides additional information. Hence, it does justify the need to use dense layers (Huang, Liu, Van Der Maaten, & Weinberger, 2017). The only input is the combination of the original environment map, the map with the position of each UAV, and the map with the visited cells and the output are the Q-values for all possible movements. As a result, each Q-Learning experiment follows:

1. Build the ANN model(s) using the selected configuration.
2. Determine the Q-table values and the optimum course of action for each UAV in the swarm using ANN model(s).
3. Train the model(s) based on each chosen action's outcomes.
4. Select the cases where the number of movements required to explore the entire map is lower.

The information available to each agent or UAV in the swarm is very important. If the information is only local (the one perceived by the UAV itself), it implies the loss of information from other UAVs, which can be very useful. Local information may lead to the loss of valuable insights from other UAVs, while global information necessitates efficient communication mechanisms to maintain accurate knowledge updates. Therefore, errors in path planning are reduced. According to previous studies on the state of the art, the system might be employed in two different ways without clear benefits for any of them. The first step is to create a single ANN that will be used to control all of the UAVs moving, determining the movement of each one at each time and verifying the reward received (Fig. 1). Consequently, if a global ANN approach is chosen, all UAVs will share the same design and weights, with their behavior determined solely by their present state. Conversely, adopting a local ANN strategy grants each UAV a distinct ANN, resulting in responses influenced by individual design, weights, and state. That is, the main objective of the experiments is to determine which ANN configuration is better as a controller with respect to the UAVs: one ANN for all UAVs (global ANN), or one ANN for each UAV (local ANN). In both cases, the input data is the same, the information obtained from all UAVs.

4.1.3. Rewards

The Reward Function serves as a guiding mechanism within RL problems, providing agents with a framework of rewards and penalties to discern favorable and unfavorable actions. Agents seek to maximize overall gains, i.e. the summation of all rewards in the episode, even at the expense of current actions.

The largest reward must be given in order for the UAV to move to previously unexplored locations. It is also crucial that it grows as fewer cells remain undiscovered (Eq. (2)). In other words, it follows a Hill-Climbing scheme (Kimura, Yamamura, & Kobayashi, 1995). For previously visited cells, another reward is needed. The UAV has a reward in the event that flying over a cell that has previously been visited in order to reach an unvisited one is preferable to flying around it (for example, when there are spurious cells left unvisited). They are given the lowest incentive to prevent UAVs from flying into cells that

they are unable to visit. In these situations, the incentive is the lowest and the goal is to maximize rewards. Consequently, UAVs learn that it is best to avoid these situations and opt for the ones that offer higher rewards, which will allow them to maximize the total reward outcome.

$$\text{new cell reward} = \text{new cell base reward} \times (1 + \frac{max(\text{rows}, \text{columns})}{\text{non visited cells}}) \quad (2)$$

4.1.4. Memory Replay

The Memory Replay technique is a prevalent method employed in much of the current research to enhance agents' learning from their interactions with the environment. In this approach, the model undergoes training using a stored set of past observations. These observations encompass a range of information, encompassing the actions taken by the agent as well as the corresponding rewards received. This technique leverages past experiences to enrich the learning process, aiding agents in better understanding and adapting to their environment. Regularly reusing experiences increases sample efficiency and helps in stabilizing the model's training process (Foerster et al., 2017). The memory is designed to retain a substantial number of recent observations, although its capacity is constrained to make optimal use of computational resources. To manage this limitation, the memory employs a First-In-First-Out (FIFO) approach, discarding older observations to accommodate new ones. The memory is capped at a maximum capacity of 60 elements, ensuring a balance between retaining valuable recent experiences and efficiently managing available resources.

In some works in the state-of-the-art, each UAV in the group has a separate memory when using the Memory Replay approach, such as in the paper of Omoniwa, Galkin, and Dusparic (2022). It records observations together with the operations the UAV itself has taken in its memory. The actions of other UAVs are never recorded. This keeps the information from becoming cluttered. Given that multiple UAVs may be located at different locations on the map, the fact that an action is erroneous for one UAV does not always mean that it is improper for others. Moreover, by combining the observations of all UAVs, one UAV may discover actions or combinations of actions that can serve other UAVs later on. The end results might be significantly impacted by the memory's size and structure (Liu & Zou, 2018).

4.1.5. Optimization metric

To estimate the goodness of the proposed method, it has been decided to count the number of actions (also known as movements) performed by all UAVs in the system (Eq. (4)). The number of actions performed by a single UAV is the same as the length of its flight path (Eq. (3)). For a flight path, having too many actions implies higher energy consumption and errors. For instance, it is worse than another path with fewer actions and that flies over the same cells.

Some authors in the state-of-the-art opt for smoothing the paths to make them simpler and better according to an optimization criterion (Correl, 2016). A grid-map will produce paths with several abrupt turns, but a sampling-based technique will produce paths that are randomly zigzagged. Implementing an additional algorithm to smooth the path and reduce some of the sharp turns can notably improve outcomes. However, it is worth noting that path smoothing may introduce inaccuracies in data retrieval since not all cells covered during flight may be completely surveyed, potentially affecting precision. This trade-off between path smoothness and data accuracy underscores the complexity of optimizing UAV trajectories.

$$\text{drone}_i \text{ taken actions} = \text{length}(\text{drone}_i \text{ path}) \quad (3)$$

$$\text{Total actions} = \sum_{i=1}^{n} \text{drone}_i \text{ taken actions} \quad (4)$$

As the desire is to lower the energy consumption for each operation in order to shorten the load time between processes, UAVs are considered to stop once the task is completed and are not considered to automatically return to the starting point. Therefore, the energy consumption of flying back is reduced.

4.1.6. Completeness criterion

As in any Path Planning problem, it is necessary to know if the results are correct. In other words, if they meet a completeness criterion (Giesbrecht, 2004). With this criterion, it is possible to quantify whether each solution obtained is better than the others.

This is a project that seeks to maximize the coverage of a field. That is why the best way to determine completeness is to measure how long it takes the UAVs in a swarm to cover an entire map. This methodology aligns with that of other researchers in the field, who similarly evaluate various parameters with the overarching aim of ascertaining whether all regions of a given map have been successfully surveyed. By employing this comprehensive approach, the project aims to effectively measure the effectiveness of the UAV swarm in achieving optimal coverage across the designated area (Albani, Manoni, Arik, Nardi, & Trianni, 2019; Albani, Nardi, & Trianni, 2017; Qu et al., 2022).

Having a completeness measure that works at the same time as an optimization criterion will allow the proposed method to obtain the best possible path. That is, being able to determine the number of moves the UAVs make to complete the task enables to quantify how good a solution is. Moreover, if a solution fails to cover a map because it converges too early, it will be discarded.

4.2. Experimentation system

As the proposed model for the experiments, a system based on Q-Learning techniques that relies on ANN for better results has been chosen. The best ANN architecture and the best parameters for all precise aspects have been sought through a random hyperparameter search (Bergstra & Bengio, 2012). Thus, the best possible combination of parameters to train the ANNs are obtained in order to have the best possible results.

An ANN made up of two dense layers (Heaton, 2008; Huang et al., 2017), one with 1013 neurons and a ReLU activation function (Agarap, 2018), and the other with 4 neurons and a softmax output function (Gao & Pavel, 2017), has been chosen through empirical experimentation (Fig. 2). The Stochastic Gradient Descend (SGD) (Sutskever, Martens, Dahl, & Hinton, 2013) optimizer was selected as the ANN's optimizer. Two hidden layers have been chosen because architectures from one to three hidden layers have been proven to be universal solutions equivalent to a Turing Machine (Wei, Chen, & Ma, 2022). These kinds of networks can approximate any mapping regardless of the required accuracy, which means it might not be necessary to use a path smoothing stage (Heaton, 2008). The network's input includes the initial environment map, the map with visited cells, and the map indicating UAV positions. This means the ANN uses existing environment data without needing extra information. The ANN's outputs are Q-values for actions in a state, adjusted using the softmax function. There are four distinct Q-values for each movement direction: North, East, South, and West.

To meet all the requirements of the Q-Learning issues explained in Section 4.1.1, an epsilon value (ϵ) of 0.49 has been selected through a preliminary testing process as the probability of making actions at random. The factor of reduction for ϵ equals 0.93, in order not to decrease the value too much and the model continues to learn from the exploration. The minimal value for ϵ is 0.05. The value chosen for the discount factor (γ) is 0.83. All values are selected after previous exploratory research.

The reward values for the agents are in Table 1. The approach employs a dual reinforcement scheme, combining positive and negative reinforcement. New cell discovery is rewarded while revisiting a cell is penalized. This encourages agents to explore new areas rather than revisiting known ones. In addition, passing through an already visited cell is penalized less than passing through a forbidden area. The main reason behind this behavior is that it may be necessary to have paths that cross each other and that is not a mistake. If the rewards were equal, there is a risk that agents would retrace their steps as it is a

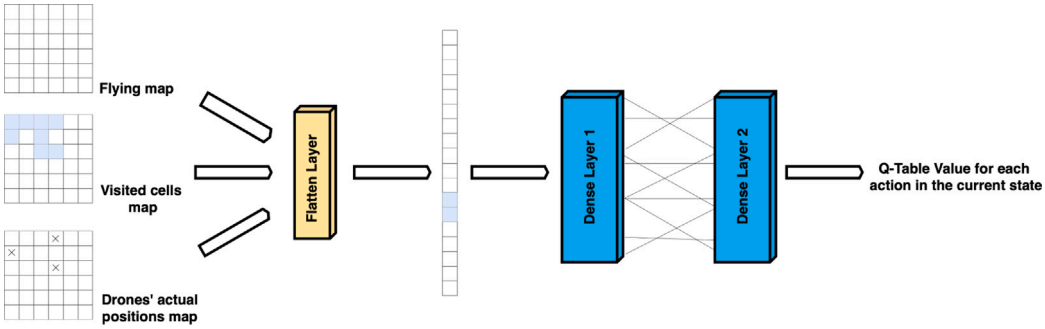


Fig. 2. Diagram with the proposed ANN model. The three inputs are combined into one and the model calculates the Q-values corresponding to each action for the current state like the one proposed in Puente-Castro, Cebrián, Sierra, and Fernandez-Blanco (2021).

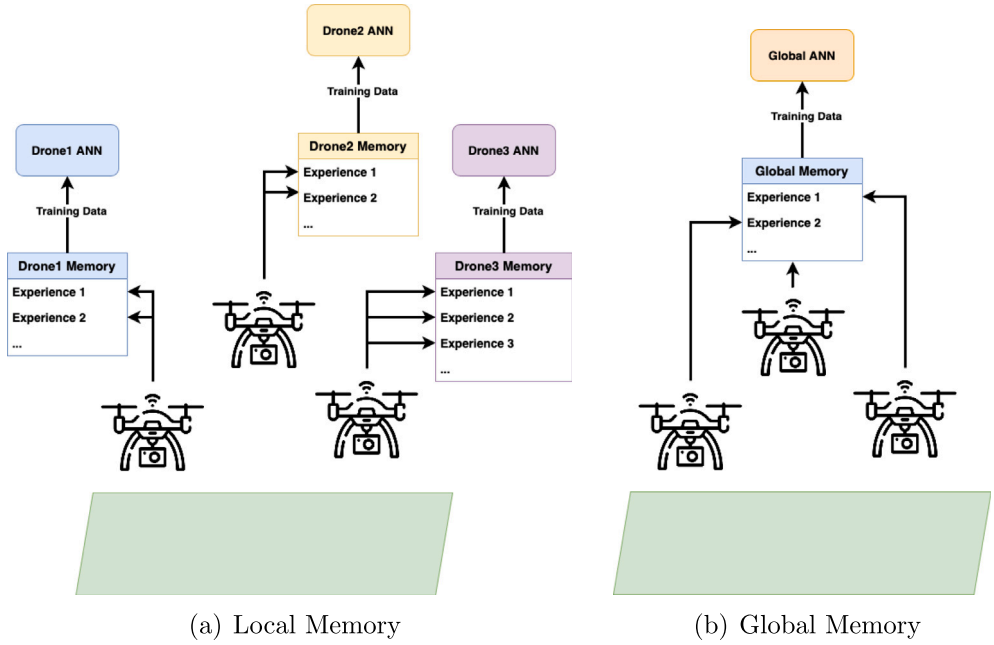


Fig. 3. Diagram illustrating the differences in the UAV experience memory system: Fig. 3(a) shows how UAVs write their experiences (one for each step they take) in their own memories, which will be used to train their own ANNs, so each memory only has experiences from one UAV. Fig. 3(b) illustrates how the UAVs record their experiences in order in a single memory, which will then be used to train a neural network, mixing the experiences of all UAVs together.

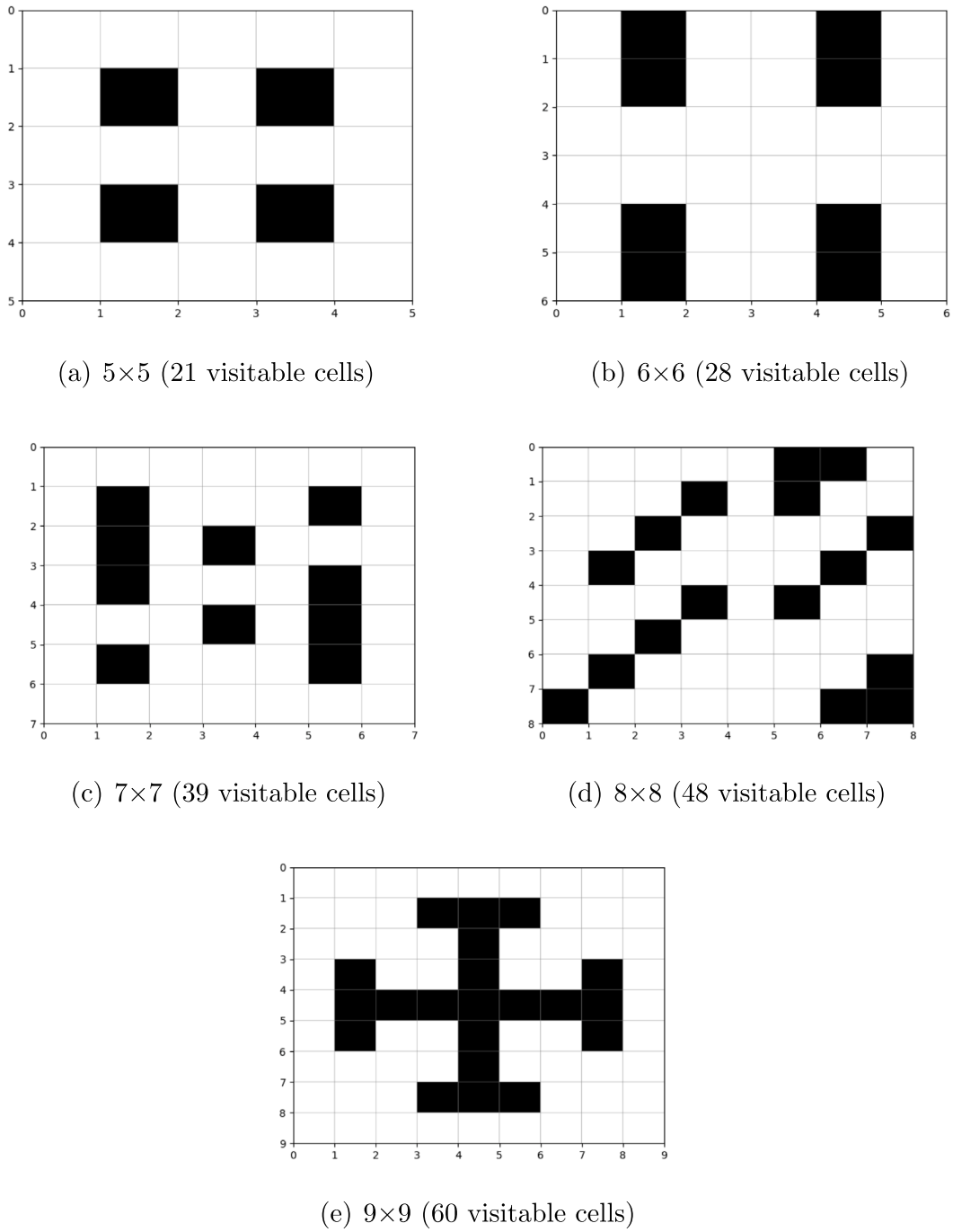
Table 1
Assigned rewards to the various cell types that each UAV visits. The initial rewards values were determined from a prior random exploration (Bergstra & Bengio, 2012) in which the most advantageous reward combinations were chosen.

	Reward
New cell base reward	29.40
Visited cell reward	−31.66
Non-visitable cell	−45.44

reward maximization problem. Therefore, this situation is penalized in case it is not avoidable for the cases in which it is essential to cross paths.

A memory size of 60 actions with their corresponding rewards was selected for this investigation. This choice is driven by the common occurrence of UAV mistakes during the initial phases of the process. A larger memory size is essential to store numerous experiences, considering the total map cells, enabling effective learning from errors. This approach facilitates the avoidance of repeated mistakes and contributes to refining the solution by retraining the model based on the majority of errors. Despite the memory's limitation to 60 elements, it remains vital to assess its behavior in relation to the ANN and its impact on the overall learning process (Fig. 3). Therefore, if it is an ANN per UAV, each ANN will have its memory with the unique experience of a single UAV (Fig. 3(a)). Contrarily, when dealing with a single ANN for all UAVs, it has been decided to use a single collective memory (Fig. 3(b)). Thus, the network learns the cases faced by all UAVs, and, in addition, the data are arbitrarily arranged, similar to having a random buffer in classical Memory Replay (Liu & Zou, 2018). By having the elements arranged randomly, the model is prevented from memorizing movement patterns and learning to generalize flight behavior. In this case, the elements are random but there are elements that represent the experience of each UAV, not just the shuffled experiences of a single UAV.

In addition to the above, the situation in which the system does not find solutions for the given circumstances has been taken into account. Therefore, as a limiting condition for shutdown, the maximum flight time has been set at 30 min. This decision is informed by the typical flight autonomy of commercial UAVs, which often operate for approximately 30 min. Thus, this duration is deemed the upper limit for



(a) 5×5 (21 visitable cells)

(b) 6×6 (28 visitable cells)

(c) 7×7 (39 visitable cells)

(d) 8×8 (48 visitable cells)

(e) 9×9 (60 visitable cells)

Fig. 4. Maps used in the flight environments. Obstacles are shown in black. In white are the cells that can be flown over. UAVs must visit as many white cells as possible.

the UAV swarm's airborne capability, ensuring that the system operates within practical constraints.

To conduct the tests and analyze the results, a set of combinations of map sizes with obstacles and UAV count have been defined. The number of actions carried out by each UAV was taken into consideration when analyzing the results.

4.3. Experiment design

Twenty-five experiments have been created to evaluate the system's capabilities, as presented in this paper. The number of UAVs, the number of ANNs, and the size of the map vary between each one of them.

Since these are ANNs with random initialization, different seeds are tested to have a higher generalization power (Zhang, Ballas, & Pineau, 2018). In addition, for better statistical measurement, the experiments are repeated 5 times with different seeds to have their mean and standard deviation.

Most of the studies in Section 2 employ maps with fixed dimensions ($5 \times 5, 10 \times 10$, or 20×20 cells), but some also use continuous maps without cell division. Continuous maps segmented into uniform cells were chosen over other options for this study. The rationale behind this decision is the paper's focus on complete map coverage for data collection. By employing equally-sized cells, each with identical visitation costs, the objective is to systematically divide the total area into manageable sections. This approach facilitates efficient data collection

by ensuring uniform coverage and organized exploration of the entire map.

The selected flight maps have fewer cells compared to the mentioned previous works. This choice is motivated by the consideration of the cost associated with flying over expansive maps. Given that each cell necessitates a stop for surface capture, larger maps would require multiple stops, leading to substantial battery drainage for the UAVs. By employing fewer cells in map division, the frequency of stops and starts for each UAV is reduced, resulting in decreased energy consumption. This strategy aims to optimize energy efficiency during data collection operations. For this purpose, 5 maps have been defined, ranging in size from 5×5 cells to 9×9 cells (Fig. 4). In the design of the obstacles, the paths were configured in such a way that many changes in direction were compelled and even backward travel was required. This is because they force the UAV to take non-linear paths, which are the ones that have higher energy requirements. In order to establish a common starting point for all maps, the upper right corner has been set. By doing so, it is intended to replicate the fact that operators always begin their operations from a corner.

In the case of the 5×5 cell map (Fig. 4(a)), it is intended to simulate the case of tree crops such as olive trees, which are regularly arranged. To complicate that task, the 6×6 cell map has been designed (Fig. 4(b)) to display situations that involve turning the UAVs around. In this way, the UAVs are forced to move backward.

Both the 5×5 and 6×6 cell maps are horizontally and vertically symmetrical. To test how the UAVs behave outside these conditions, the 7×7 cell map has been designed (Fig. 4(c)). Furthermore, following this premise, the 8×8 cells map has been designed (Fig. 4(d)), which also tests the behavior of the system if the obstacles are arranged diagonally.

The last map to be tested is the 9×9 cells map (Fig. 4(e)). In the previous maps, UAVs could pass through the gaps between the obstacles. This map tests the behavior of the system if a single large obstacle has to be circled. In addition, corners have been added to make it more difficult for UAVs to retrace their steps at some points.

Finally, despite varying the obstacles, the number of cells in the maps is also varied to test that the system works for any size. Therefore, the system is tested to prove that it is effective in different situations.

Each chosen map type was evaluated with an increasing number of UAVs because it is crucial for the system to function with any quantity of UAVs. Separate tests with 1, 2, and 3 UAVs have been carried out. Thus, it is demonstrated that the system can adapt to a variety of UAV numbers. It is also worth highlighting the equivalence of employing a global or local approach when a single UAV is used. So, those executions have been referred to as baseline. As a result, it is assumed that the experiment will begin by controlling a single UAV, which is the simplest situation.

In this study, the chosen flight environment has fewer cells compared to the mentioned studies. This decision was influenced by the cost of covering extensive maps, as each cell requires a stop for image capture, leading to high energy consumption. Dividing the map into fewer cells reduces stops and conserves energy, although each cell covers a larger area. Larger captured images offer more contextual information and are better suited for processing, despite having lower detail. Adjusting the cell count to the map size is crucial to prevent loss of information, where a single large cell might miss small obstacles and be treated as an obstacle itself.

The decision to use atomic movements (North, South, East, and West) for the UAVs was made to streamline processing. UAVs can execute these well-defined actions without the need for additional turns. This approach also minimizes energy demands, especially in scenarios with numerous curves that tend to increase energy consumption.

The range of possible movements or actions (a) that UAVs can take was encoded using integer values from 0 to 3, representing the directions: North, East, South, and West. This discrete coding simplifies the representation of movements in the technique and assigns distinct values to each direction.

All these variables are summarized in Table 2.

Table 2
Summary table with the values chosen for experimentation. All the values have been obtained through a preliminary testing process.

Variable	Value
Neurons First Dense Layer	1013 neurons
Activation Function First Dense Layer	ReLU
Neurons Second Dense Layer	4 neurons
Activation Function Second Dense Layer	Linear
ANN Output Function	Softmax
Epsilon (ϵ)	0.49
Epsilon decay	0.93
Minimum Epsilon (ϵ)	0.05
Discount Factor (γ)	0.83
Memory Size	60 actions
Maximum Flight Time	30 min
Maximum Number of Episodes	30 episodes
Possible actions	North, East, South, West

5. Results

Table 3 shows the results obtained from the experimentation. For each map size, the mean and standard deviation of actions taken for each ANN configuration when faced with different numbers of UAVs are compared. To better show the capabilities of the proposed model (known as Proposed in Table 3) it is compared with the model proposed by Puente-Castro et al. (2022), known as Control, which already demonstrated its capabilities on obstacle-free maps. It can be seen that the means of the results of the proposed model are lower than those of the model with which they are contrasted. This can be interpreted as an indication that the paths take fewer actions to complete the operation. Therefore, they are better and more efficient.

With regard to the number of UAVs, it is evident that as the number of UAVs increases, the required number of actions decreases. This supports the notion that coordinated movements among cooperative UAV groups enhance operational speed and efficiency. However, this reduction is not strictly proportional to the number of UAVs, as it is influenced by factors such as map size and obstacles, which vary across different scenarios. For example, the difference in actions required for the 8×8 map is greater in all cases than for the 7×7 map despite being a map with only 15 more cells.

In the 5×5 cell map (Fig. 4(a)) both models present a similar behavior, but the proposed model finds the solution with fewer actions. The local ANNs exhibit higher speed and lower variance compared to global ANNs. The lower variance implies more consistent and optimized paths, showcasing the model's robust behavior. This same pattern is true in the 6×6 cell map (Fig. 4(b)). Moreover, in this second map, the obstacles are not islands to go around but form corners that force the UAVs to retrace their steps. Having to retrace their steps is what causes such a large increase in the average movement despite having only 11 more cells, of which 4 are new obstacles.

There is a trend change in the 7×7 cell map (Fig. 4(c)), not only because there are more obstacles and the map is larger, but also because the obstacles do not present horizontal or vertical symmetry. In both models, the scenarios involving 2 UAVs exhibit a sudden increase in movement, suggesting potential disruption of paths due to interaction between the UAVs. Interestingly, the proposed global model yielded the lowest mean movement compared to the local model. However, these global paths display higher variance, indicating reduced robustness compared to the local ANN solution.

For the 8×8 cell map (Fig. 4(d)) there is no longer such an abrupt growth in the means of the actions of the paths. In this specific scenario, the proposed model does not achieve the lowest mean movement for 2 UAVs. However, it stands out for having the lowest variance, indicating greater accuracy in its computations.

Finally, in the 9×9 cell map (Fig. 4(e)) both models behave similarly. It should be noted that the results are similar to those of the

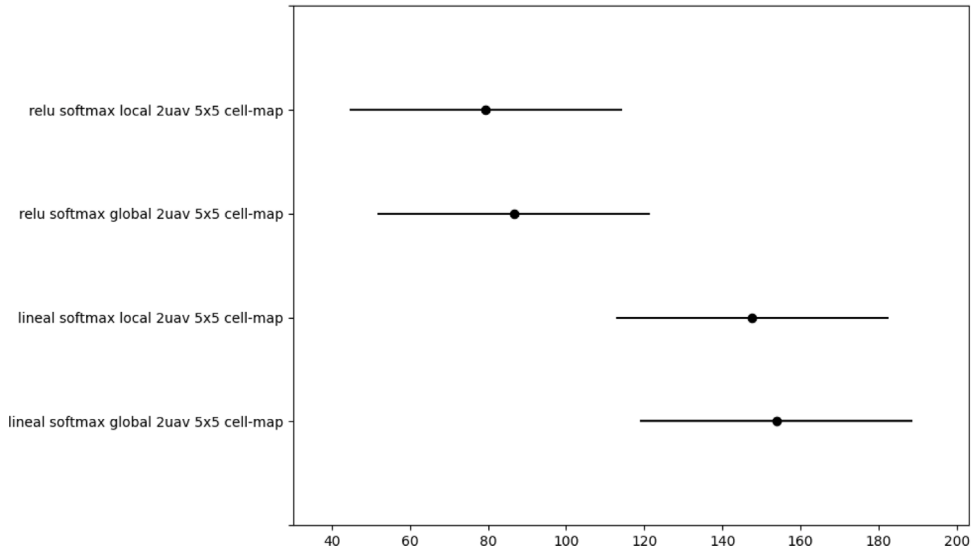


Fig. 5. Plot of the universal confidence interval resulting from Tukey's test. The results for the distributions with statistically significant results are displayed. In the y axis, distributions are listed. In the x axis, the average actions taken for the flight paths of each distribution are displayed.

Table 3
Table with the mean and standard deviation of total actions taken by the swarm of UAVs for each map and for each ANN configuration. Generally, the more UAVs in the swarm, the fewer actions the swarm takes to fly over the entire map.

Map size	Number of UAVs	ANN configuration			
		Local control	Global control	Local proposed	Global proposed
5 × 5	Baseline	283.20 ± 97.79		189.00 ± 91.06	
	2 UAVs	147.60 ± 38.68	153.80 ± 56.42	79.40 ± 7.82	86.60 ± 34.62
	3 UAVs	76.60 ± 53.26	100.60 ± 51.76	56.20 ± 21.32	61.60 ± 47.26
6 × 6	Baseline	503.60 ± 195.34		212.60 ± 49.42	
	2 UAVs	145.40 ± 24.93	232.60 ± 189.49	123.00 ± 12.98	230.00 ± 156.62
	3 UAVs	122.00 ± 55.29	139.60 ± 51.71	127.20 ± 69.83	135.60 ± 51.08
7 × 7	Baseline	523.60 ± 127.93		491.20 ± 15.61	
	2 UAVs	384.80 ± 112.3	537.40 ± 425.41	348.80 ± 151.46	278.40 ± 143.34
	3 UAVs	199.40 ± 66.09	292.20 ± 181.60	166.60 ± 56.00	151.20 ± 72.70
8 × 8	Baseline	1011.00 ± 258.16		1367.80 ± 543.17	
	2 UAVs	700.00 ± 221.08	611.80 ± 484.85	757.60 ± 127.29	654.80 ± 285.69
	3 UAVs	681.80 ± 192.50	582.00 ± 268.86	533.60 ± 369.07	675.8 ± 387.96
9 × 9	Baseline	1332.00 ± 804.16		2264.60 ± 1148.34	
	2 UAVs	980.40 ± 522.45	1107.60 ± 157.47	1232.00 ± 573.74	1087.20 ± 549.05
	3 UAVs	645.00 ± 203.67	690.60 ± 308.33	564.40 ± 210.77	761.00 ± 297.29

8 × 8 map despite being larger, so it can be understood that the layout of the obstacles is more influential to the size of the map.

Statistical tests are performed at a significance level of $\alpha = 0.1$. First, a Shapiro–Wilk (Razali, Wah, et al., 2011) test of normality was performed to find out which statistical significance test can be applied. Not all distributions appear not to follow a normal disposition (Table 4). This phenomenon is more noticeable in scenarios involving multiple UAVs as opposed to a single UAV. The reason behind this could be the interference caused by one UAV's path on the trajectories of others, whether it is due to prior passage through a cell or simultaneous occupancy of the same cell. Essentially, the movement of one UAV has an impact on the paths of both itself and the other UAVs, creating a complex interplay of interactions.

Since not all the distributions obtained do not follow a normal distribution, a Kruskal–Wallis significance test (McKight & Najab, 2010) was used to determine whether they follow significantly different distributions. For this test, a significance level ($alpha$) equal to that used for the normality tests was used.

According to the Kruskal–Wallis test, there are distributions that are significantly different. It is necessary to determine which are significantly different from each other, so a series of Tukey's tests (Tukey, 1949) was performed to find out which are significantly different from each other. The same level of significance was also used for the tests.

Table 4
Table with the p-values of the non-normal distributions resulting from performing the Shapiro–Wilk test (Razali et al., 2011).

Model	Configuration	Number of UAVs	Map size	p-value
Control	Global	2 UAVs	6 × 6	0.095
			5 × 5	0.019
		3 UAVs	7 × 7	0.026
			8 × 8	0.017
	Local	3 UAVs	6 × 6	0.079
			7 × 7	0.075
Proposed	Global	2 UAVs	7 × 7	0.066
			8 × 8	0.094
		3 UAVs	5 × 5	0.011
	Local	2 UAVs	5 × 5	0.049
			8 × 8	0.053
		3 UAVs	6 × 6	0.057

The cases in which there has been statistical significance are those resulting from experimenting with 2 UAVs and maps of 5 × 5. As can be seen in Fig. 5, the indicated distributions show differences if the proposed model is compared with the one used for the contrast (Puente-Castro et al., 2022). These show around half of the average number

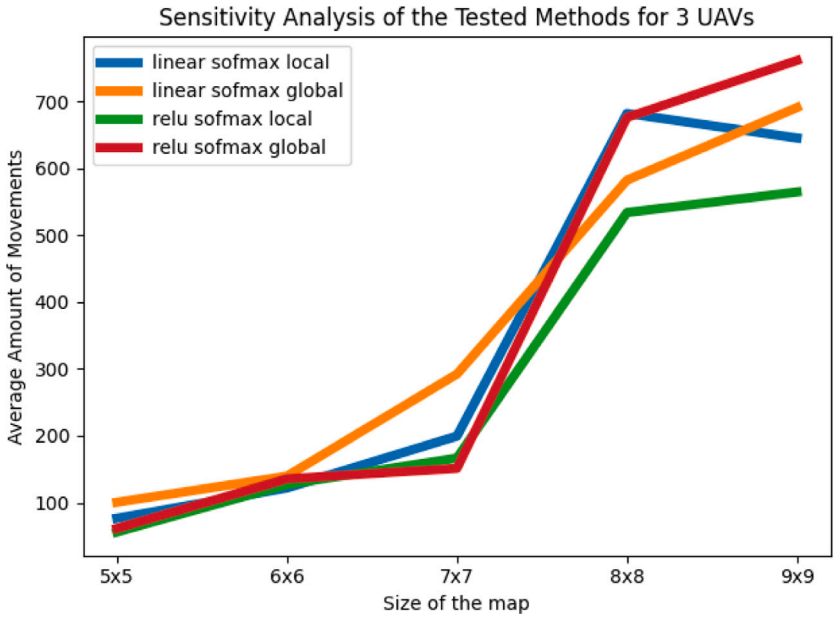


Fig. 6. Sensitivity analysis of the evolution of the models tested under equal conditions. In it, it can be seen that some models have a more stable behavior before smaller maps but that they grow a lot in larger maps. In addition, other models have a less pronounced growth as the size of the maps and the number of obstacles increase.

of actions (x axis) in the proposed model than with the one used to contrast the results. It may be indicative of the models having non-significantly different behavior in all scenarios except for the indicated cases of 2 UAVs on 5×5 cells maps. Hence, opting for the proposed model is advantageous due to its tendency to offer shorter or at least equally sized paths. Even though there is no significant difference, these marginal enhancements can prove valuable in practical scenarios. Shorter paths, no matter how minor the difference, contribute to energy savings during flight, making them beneficial in real-world applications.

Both the proposed model and the one involved in the contrast show no significant differences when comparing their global and local variants for the same model. This suggests that the choice between the two approaches might not yield substantial variations in results. Consequently, opting for local models for all UAVs appears favorable, as it typically involves fewer steps and offers comparable outcomes.

The behavior of the contrasted models can be seen in an alternative way by showing a sensitivity analysis between them, as other authors do in the field of RL in economics (Pröllochs, Feuerriegel, & Neumann, 2016). The tested models exhibit structural similarities, yet their diverse parameters and configurations lead to substantial behavioral differences. These variations, while not easily discernible from tabulated data, become evident through sensitivity analysis. By observing how results evolve under different circumstances, we gain insights into the models' behavior. For this purpose, it was decided to look at the evolution of the average number of steps required for 3 UAVs as the complexity of the maps increased (Fig. 6).

6. Conclusions and future work

This study proposes a new system that employs Q-Learning and ANNs with two dense layers to control UAV swarms in maps with obstacles. By optimizing flight paths and reducing actions as the UAV swarm grows, the system offers adaptability across different devices. This shift towards an autonomous UAV swarm provides cost savings, time efficiency, and improved fault tolerance compared to single UAVs or manual management.

Since it is not necessary to know the spatial relationship of the obstacles with the rest of the environment, it can be understood that the sequence of movements and the position of the UAVs in the swarm is more important. Thus, the actions of a single UAV affect the paths of the others, since it modifies the reward values perceived by the others. Additionally, unlike other published work in this field, it is not necessary to include targets or other metrics to guide the computation of paths.

The system has certain limitations. Firstly, the UAV movements are treated atomically, which might not be ideal for tasks needing smoother paths and efficient data capture. The system also does not consider varying UAV heights, potentially affecting path calculations and the accuracy of rewards based on data quality. However, UAVs generally maintain altitudes that accommodate disturbances and adjusting height for obstacles like birds would involve only minor changes. Despite these limitations, the system achieves satisfactory results across different flight heights.

This work provides a basis for further investigation on UAV swarms for Path Planning, particularly concerning experiments with compact fully-connected ANNs in obstacle-ridden maps. Further investigations could encompass more intricate environments like 3D maps, allowing UAVs to execute diverse motions including pitch and roll. Enhancements might involve implementing actions like stopping to mitigate collision risks in intersecting paths.

Enhancing movement precision can entail increased system complexity. For instance, integrating ANNs for distinct functions could be explored. The combination of multiple ANNs offers the potential to incorporate additional flight capabilities, like altitude adjustments or tilting. Employing multiple ANNs to coordinate composite movements, such as simultaneous ascent and turns, may lead to improved accuracy and quicker outcomes.

The most important improvement is to achieve a system that allows a greater variety of movements. For example, these actions can be combinations in different degrees of the above. The "stop" command could even be used as an action. Having more actions and some combined ones makes it more difficult to count the paths, but it can improve the precision of the movements. In this way, the data capture is optimized and the risk of maneuvers is reduced.

Code availability

Source code and a Docker container are available at:

https://github.com/TheMVS/UAV_SWARMS_RL_FIXED_OBSTACLES_MAPS

https://hub.docker.com/repository/docker/themvs/uav_swarms_rl_fixed_obstacles_maps/

Funding

This project was supported by the FCT - Foundation for Science and Technology, Portugal, in the context of the project [grant number UIDB/00127/
2020], and also POCI 2020, in the context of the Germirrad project [grant number POCI-01-0247-FEDER-072237]. Also, the General Directorate of Culture, Education, and University Management of Xunta de Galicia [grant number ED431D 2017/16]. This work was also funded by the grant for the consolidation and structuring of competitive research units [grant number ED431C 2022/46] from the General Directorate of Culture, Education and University Management of Xunta de Galicia, and the CYTED network, Spain [grant number PCI2018_093284] funded by the Spanish Ministry of Innovation and Science. This project was also supported by the General Directorate of Culture, Education and University Management of Xunta de Galicia "PRACTICUM DIRECT" [grant number IN845D-2020/03].

CRediT authorship contribution statement

Alejandro Puente-Castro: Conceptualization, Methodology, Software, Validation, Formal analysis, Resources, Data curation, Writing – original draft, Visualization, Investigation. **Daniel Rivero:** Writing – review & editing, Supervision. **Eurico Pedrosa:** Writing – review & editing. **Artur Pereira:** Writing – review & editing. **Nuno Lau:** Writing – review & editing, Supervision. **Enrique Fernandez-Blanco:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

Agarap, A. F. (2018). Deep learning using rectified linear units (ReLU). arXiv preprint arXiv:1803.08375.

Aggarwal, S., & Kumar, N. (2020). Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Computer Communications*, *149*, 270–299.

Albani, D., IJsselmuiden, J., Haken, R., & Trianni, V. (2017). Monitoring and mapping with robot swarms for agricultural applications. In *2017 14th IEEE international conference on advanced video and signal based surveillance* (pp. 1–6). IEEE.

Albani, D., Manoni, T., Arik, A., Nardi, D., & Trianni, V. (2019). Field coverage for weed mapping: Toward experiments with a UAV swarm. In *Bio-inspired information and communication technologies: 11th EAI international conference, BICT 2019, Pittsburgh, PA, USA, March 13–14, 2019, Proceedings 11* (pp. 132–146). Springer.

Albani, D., Nardi, D., & Trianni, V. (2017). Field coverage and weed mapping by UAV swarms. In *2017 IEEE/RSJ international conference on intelligent robots and systems* (pp. 4319–4325). Ieee.

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. In *2017 International conference on engineering and technology* (pp. 1–6). Ieee.

Austin, R. (2011). *Unmanned aircraft systems: UAVs design, development and deployment*. John Wiley & Sons.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(2).

Bocchino, R., Canham, T., Watney, G., Reder, L., & Levison, J. (2018). F Prime: An open-source framework for small-scale flight software systems. Preprint.

Bonabeau, E., & Meyer, C. (2001). Swarm intelligence: A whole new way to think about business. *Harvard Bus. Rev.*, *79*(5), 106–115.

Campion, M., Ranganathan, P., & Faruque, S. (2018). A review and future directions of UAV swarm communication architectures. In *2018 IEEE international conference on electro/information technology* (pp. 0903–0908). IEEE.

de Carvalho, K. B., de Oliveira, I. R. L., Villa, D. K., Caldeira, A. G., Sarcinelli-Filho, M., & Brandão, A. S. (2022). Q-learning based path planning method for uavs using priority shifting. In *2022 International conference on unmanned aircraft systems* (pp. 421–426). IEEE.

de Castro, G. G., Pinto, M. F., Biundini, I. Z., Melo, A. G., Marcato, A. L., & Haddad, D. B. (2023). Dynamic path planning based on neural networks for aerial inspection. *Journal of Control, Automation and Electrical Systems*, *34*(1), 85–105.

Chen, Y., Dong, Q., Shang, X., Wu, Z., & Wang, J. (2022). Multi-UAV autonomous path planning in reconnaissance missions considering incomplete information: A reinforcement learning method. *Drones*, *7*(1), 10.

Clifton, J., & Laber, E. (2020). Q-learning: Theory and applications. *Annual Review of Statistics and Its Application*, *7*, 279–301.

Correl, P. (2016). Introduction to autonomous robots. Kinematics, perception, localization and planning. (pp. 85–86).

Corte, A. P. D., Souza, D. V., Rex, F. E., Sanquetta, C. R., Mohan, M., Silva, C. A., et al. (2020). Forest inventory with high-density UAV-lidar: Machine learning approaches for predicting individual tree attributes. *Computers and Electronics in Agriculture*, *179*, Article 105815.

Dhuheir, M., Baccour, E., Erbad, A., Al-Obaidi, S. S., & Hamdi, M. (2022). Deep reinforcement learning for trajectory path planning and distributed inference in resource-constrained UAV swarms. *IEEE Internet of Things Journal*.

Fan, J., Wang, Z., Xie, Y., & Yang, Z. (2020). A theoretical analysis of deep Q-learning. In *Learning for dynamics and control* (pp. 486–489). PMLR.

Foerster, J., Nardelli, N., Farquhar, G., Afouras, T., Torr, P. H., Kohli, P., et al. (2017). Stabilising experience replay for deep multi-agent reinforcement learning. In *International conference on machine learning* (pp. 1146–1155). PMLR.

Gao, B., & Pavel, L. (2017). On the properties of the softmax function with application in game theory and reinforcement learning. arXiv preprint arXiv:1704.00805.

Gasparetto, A., Boscariol, P., Lanzutti, A., & Vidoni, R. (2015). Path planning and trajectory planning algorithms: A general overview. *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*, 3–27.

Giesbrecht, J. (2004). *Global path planning for unmanned ground vehicles*: *Technical report*, Defence Research and Development Suffield (ALBERTA).

Gläscher, J., Daw, N., Dayan, P., & O'Doherty, J. P. (2010). States versus rewards: Dissociable neural prediction error signals underlying model-based and model-free reinforcement learning. *Neuron*, *66*(4), 585–595.

He, W., Qi, X., & Liu, L. (2021). A novel hybrid particle swarm optimization for multi-UAV cooperate path planning. *Applied Intelligence*, *51*(10), 7350–7364.

Heaton, J. (2008). *Introduction to neural networks with Java* (p. 158). Heaton Research, Inc..

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708).

Huuskonen, J., & Oksanen, T. (2018). Soil sampling with drones and augmented reality in precision agriculture. *Computers and Electronics in Agriculture*, *154*, 25–35.

Jaakkola, T., Singh, S., & Jordan, M. (1994). Reinforcement learning algorithm for partially observable Markov decision problems. *Advances in Neural Information Processing Systems*, *7*.

Jain, G., Yadav, G., Prakash, D., Shukla, A., & Tiwari, R. (2019). MVO-based path planning scheme with coordination of UAVs in 3-D environment. *Journal of Computer Science*, *37*, Article 101016.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Karur, K., Sharma, N., Dharmatti, C., & Siegel, J. E. (2021). A survey of path planning algorithms for mobile robots. *Vehicles*, *3*(3), 448–468.

Kennedy, J. (2006). Swarm intelligence. In *Handbook of nature-inspired and innovative computing* (pp. 187–219). Springer.

Khalil, A. A., & Rahman, M. A. (2022). FED-UP: Federated deep reinforcement learning-based UAV path planning against hostile defense system. In *2022 18th international conference on network and service management* (pp. 268–274). IEEE.

Kimura, H., Yamamura, M., & Kobayashi, S. (1995). Reinforcement learning by stochastic hill climbing on discounted reward. In *Machine learning proceedings 1995* (pp. 295–303). Elsevier.

Kong, F., Nie, Y., & Xu, X. (2022). An improved GA-based approach for UAV swarm formation transformation. In *2022 IEEE 6th information technology and mechatronics engineering conference, vol. 6* (pp. 1715–1720). IEEE.

Kong, F., Wang, Q., Gao, S., & Yu, H. (2023). B-APFDQN: A UAV path planning algorithm based on deep Q-network and artificial potential field. *IEEE Access*.

Krogh, A. (2008). What are artificial neural networks? *Nature biotechnology*, *26*(2), 195–197.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Li, S. E. (2023). Deep reinforcement learning. In *Reinforcement learning for sequential decision and optimal control* (pp. 365–402). Springer.

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*.

Liu, J. (2022). An improved genetic algorithm for rapid UAV path planning. *Journal of Physics: Conference Series*, *2216*, Article 012035.
Liu, J., Wang, W., Wang, T., Shu, Z., & Li, X. (2018). A motif-based rescue mission planning method for UAV swarms usingan improved PICEA. *IEEE Access*, *6*, 40778–40791.
Liu, Y., Zheng, Z., Qin, F., Zhang, X., & Yao, H. (2022). A residual convolutional neural network based approach for real-time path planning. *Knowledge-Based Systems*, *242*, Article 108400.
Liu, R., & Zou, J. (2018). The effects of memory replay in reinforcement learning. In *2018 56th Annual Allerton conference on communication, control, and computing* (pp. 478–485). IEEE.
McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*, 115–133.
McKight, P. E., & Najab, J. (2010). Kruskal-Wallis test. In *The corsini encyclopedia of psychology* (p. 1). Wiley Online Library.
Michie, D., Spiegelhalter, D. J., Taylor, C., et al. (1994). Machine learning. *Neural and Statistical Classification*, *13*.
Minh, H. L., Sang-To, T., Theraulaz, G., Wahab, M. A., & Cuong-Le, T. (2023). Termite life cycle optimizer. *Expert Systems with Applications*, *213*, Article 119211.
Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533.
Noor, N. M., Abdullah, A., & Hashim, M. (2018). Remote sensing UAV/drones and its applications for urban areas: A review. In *IOP Conference Series: Earth and Environmental Science*: *vol. 169*, IOP Publishing, Article 012003.
Omoniwa, B., Galkin, B., & Dusparic, I. (2022). Optimizing energy efficiency in UAV-assisted networks using deep reinforcement learning. *IEEE Wireless Communications Letters*, *11*(8), 1590–1594.
Pamosoaji, A. K., Piao, M., & Hong, K.-S. (2019). PSO-based minimum-time motion planning for multiple vehicles under acceleration and velocity limitations. *International Journal of Control, Automation and Systems*, *17*(10), 2610–2623.
Patle, B., Pandey, A., Parhi, D., Jagadeesh, A., et al. (2019). A review: On path planning strategies for navigation of mobile robot. *Defence Technology*, *15*(4), 582–606.
Pröllochs, N., Feuerriegel, S., & Neumann, D. (2016). Detecting negation scopes for financial news sentiment using reinforcement learning. In *2016 49th Hawaii international conference on system sciences* (pp. 1164–1173). IEEE.
Puente-Castro, A., Cebrián, D., Sierra, A., & Fernandez-Blanco, E. (2021). Artificial intelligence techniques for autonomous drone swarms. In *MOL2NET'21, Conference on molecular, biomedical & computational sciences and engineering* (7th ed.). MDPI.
Puente-Castro, A., Rivero, D., Pazos, A., & Fernandez-Blanco, E. (2021). A review of artificial intelligence applied to path planning in UAV swarms. *Neural Computing and Applications*, 1–18.
Puente-Castro, A., Rivero, D., Pazos, A., & Fernandez-Blanco, E. (2022). UAV swarm path planning with reinforcement learning for field prospecting. *Applied Intelligence*, 1–18.
Qiu, X., Xu, L., Wang, P., Yang, Y., & Liao, Z. (2022). A data-driven packet routing algorithm for an un-manned aerial vehicle swarm: A multi-agent reinforcement learning approach. *IEEE Wireless Communications Letters*.
Qu, C., Boubin, J., Gafurov, D., Zhou, J., Aloysius, N., Nguyen, H., et al. (2022). Uav swarms in smart agriculture: Experiences and opportunities. In *2022 IEEE 18th international conference on E-science* (pp. 148–158). IEEE.
Rabinovitch, J., Lorenz, R., Slimko, E., & Wang, K. S. C. (2021). Scaling sediment mobilization beneath rotorcraft for Titan and Mars. *Aeolian Research*, *48*, Article 100653.
Raja, G., Anbalagan, S., Narayanan, V. S., Jayaram, S., & Ganapathisubramaniyan, A. (2019). Inter-UAV collision avoidance using deep-Q-learning in flocking environment. In *2019 IEEE 10th annual ubiquitous computing, electronics & mobile communication conference* (pp. 1089–1095). IEEE.
Razali, N. M., Wah, Y. B., et al. (2011). Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *Journal of Statistical Modeling and Analytics*, *2*(1), 21–33.
Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H., Albarqouni, S., et al. (2020). The future of digital health with federated learning. *NPJ digital medicine*, *3*, 119.
Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386.
Sahin, E., & Winfield, A. F. (2008). Special issue on swarm robotics. *Swarm Intelligence*, *2*(2–4), 69–72.
Salimi, M., & Pasquier, P. (2021). Deep reinforcement learning for flocking control of UAVs in complex environments. In *2021 6th international conference on robotics and automation engineering* (pp. 344–352). IEEE.
Sang-To, T., Le-Minh, H., Mirjalili, S., Wahab, M. A., & Cuong-Le, T. (2022). A new movement strategy of grey wolf optimizer for optimization problems and structural damage identification. *Advances in Engineering Software*, *173*, Article 103276.
Sang-To, T., Le-Minh, H., Wahab, M. A., & Thanh, C.-L. (2023). A new metaheuristic algorithm: Shrimp and Goby association search algorithm and its application for damage identification in large-scale and complex structures. *Advances in Engineering Software*, *176*, Article 103363.
Sanna, G., Godio, S., & Guglieri, G. (2021). Neural network based algorithm for multi-UAV coverage path planning. In *2021 International conference on unmanned aircraft systems* (pp. 1210–1217). IEEE.
Shang, Y., & Li, S. (2022). Hybrid combinatorial remanufacturing strategy for medical equipment in the pandemic. *Computers & Industrial Engineering*, Article 108811.
Shiri, H., Park, J., & Bennis, M. (2020). Remote UAV online path planning via neural network-based opportunistic control. *IEEE Wireless Communications Letters*, *9*(6), 861–865.
Souto, A., Alfaia, R., Cardoso, E., Araújo, J., & Francês, C. (2023). UAV path planning optimization strategy: Considerations of urban morphology, microclimate, and energy efficiency using Q-learning algorithm. *Drones*, *7*(2), 123.
Stentz, A. (1997). Optimal and efficient path planning for partially known environments. In *Intelligent unmanned ground vehicles* (pp. 203–220). Springer.
Susanto, T., Setiawan, M. B., Jayadi, A., Rossi, F., Hamdhi, A., & Sembiring, J. P. (2021). Application of unmanned aircraft PID control system for roll, pitch and yaw stability on fixed wings. In *2021 International conference on computer science, information technology, and electrical engineering* (pp. 186–190). IEEE.
Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *International conference on machine learning* (pp. 1139–1147). PMLR.
Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT Press.
Tu, G. T., & Juang, J. G. (2023). UAV path planning and obstacle avoidance based on reinforcement learning in 3D environments. In *Actuators*: *vol. 12*, (no. 2), (p. 57). MDPI.
Tukey, J. W. (1949). Comparing individual means in the analysis of variance. *Biometrics*, 99–114.
Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3), 279–292.
Wei, C., Chen, Y., & Ma, T. (2022). Statistically meaningful approximation: A case study on approximating turing machines with transformers. *Advances in Neural Information Processing Systems*, *35*, 12071–12083.
Wei, K., Huang, K., Wu, Y., Li, Z., He, H., Zhang, J., et al. (2022). High-performance UAV crowdsensing: A deep reinforcement learning approach. *IEEE Internet of Things Journal*.
Wiering, M., & Van Otterlo, M. (2012). Reinforcement learning. *Adaptation, learning, and optimization*, *12*, 3.
Xu, S., Li, L., Zhou, Z., Mao, Y., & Huang, J. (2022). A task allocation strategy of the UAV swarm based on multi-discrete wolf pack algorithm. *Applied Sciences*, *12*(3), 1331.
Yang, X. S. (2014). Swarm intelligence based algorithms: A critical analysis. *Evolutionary Intelligence*, *7*(1), 17–28.
Yang, L., Zhang, X., Zhang, Y., & Xiangmin, G. (2019). Collision free 4D path planning for multiple UAVs based on spatial refined voting mechanism and PSO approach. *Chinese Journal of Aeronautics*, *32*(6), 1504–1519.
Yeaman, M. L., & Yeaman, M. (1998). *Virtual air power: A case for complementing ADF air operations with uninhabited aerial vehicles*. Air Power Studies Centre.
Zhang, A., Ballas, N., & Pineau, J. (2018). A dissection of overfitting and generalization in continuous reinforcement learning. arXiv preprint arXiv:1806.07937.
Zhang, C., Vinyals, O., Munos, R., & Bengio, S. (2018). A study on overfitting in deep reinforcement learning. arXiv preprint arXiv:1804.06893.
Zhang, R., Zong, Q., Zhang, X., Dou, L., & Tian, B. (2022). Game of drones: Multi-uav pursuit-evasion game with online motion planning by deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*.
Zhao, Y., Zheng, Z., & Liu, Y. (2018). Survey on computational-intelligence-based UAV path planning. *Knowledge-Based Systems*, *158*, 54–64.
Zhou, W., Liu, Z., Li, J., Xu, X., & Shen, L. (2021). Multi-target tracking for unmanned aerial vehicle swarms using deep reinforcement learning. *Neurocomputing*, *466*, 285–297.

Alejandro Puente-Castro BSc. in Computer Science, gained his MSc. in Bioinformatics for Health Sciences and has worked infields, such as early detection of Alzheimer's disease using Deep Learning techniques or self-quantification. Currently, his research is focused on applying Artificial Intelligence techniques to the coordination of heterogeneous groups of Unmanned Aerial Vehicles (UAVs) and Bioinformatics.