

GRADO EN INGENIERÍA EN TECNOLOGÍA DE
TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

***ROBÓTICA COLABORATIVA MEDIANTE
ALGORITMOS DISTRIBUIDOS DE
DIFUSIÓN ESTOCÁSTICA***

Alumno: Jubeto, López, Xabier

Director: Del Ser, Lorente, Javier

Curso: 2018-2019

Fecha: Jueves, 27 de Junio de 2019

PÁGINA EN BLANCO

RESUMEN

El objetivo de este proyecto es desarrollar un sistema de exploración de grandes extensiones mediante una inteligencia de enjambre distribuida, basado en algoritmos de búsqueda de difusión estocástica. El enjambre está compuesto por varios agentes, tantos terrestres como voladores. Los cuales tendrán que comunicarse y cooperar entre sí para realizar tareas de exploración. Estas tareas consisten en delimitar zonas esparcidas por un mapa del cual los agentes no disponen de ninguna información previa. Adicionalmente, estas zonas requieren diferentes tipos de agentes y solo los correspondientes agentes podrán delimitarla.

Estas especificaciones tratan de simular robots explorando grandes extensiones de terreno, las cuales contienen diversos obstáculos o peligros para el ser humano. Por ese motivo, los tipos de agentes representan robots con diferentes herramientas: cámaras, micrófonos, termómetros, barómetros, espectroscopio, etc. A la vez que simulan las características de esas zonas, las cuales podrían ser: fugas de gas, obstáculos físicos, peligros radioactivos, etc.

ABSTRACT

The aim of this project is to develop a system to explore huge land extensions with a distributed swarm intelligence. To accomplish this task Stochastic Diffusion Search algorithms will be used. The swarm is created by various agents, which can be land agents or flying agents. These agents will need to communicate and cooperate for accomplish exploration tasks. These tasks consist in demarcate areas scattered across a map. Additionally, these areas required different types of agents and only those types of agents can demarcate them.

Robots exploring huge land extensions have been trying to simulate with these specifications. Moreover, the map contains several obstacles or dangers for the human being. In addition, the types of agents represent various robots with different tools: cameras, microphones, barometers, spectrometers and etcetera. At the same time, they also simulate the zones, for example: gas leaks, physique obstacles or radioactive dangers.

LABURPENA

Proiektu honen helburu nagusia eremu handien esplorazio sistema bat garatzea da. Banatutako talde inteligentzia baten bidez eta difusio estokatiko bilaketa algoritmoetan oinarrituta. Taldea lurreko eta hegaleko hainbat agente ditu. Hauek komunikatu eta elkarlanean aritu beharko dira esplorazio lanak egiteko. Zeregin hauek mapa batetan sakabanatuta dauden eremuak mugatzea da. Horrez gain, eremu horiek agente mota desberdinak eskatzen dituzte eta dagokion agenteak soilik mugatu ahal izango dituzte.

Zehaztapen horiek eremu handiak esploratzen dituzten robotak simulatzen saiatzen dira, gizakiak oztopo edo arrisku ugari biltzen dituenak. Hori dela eta, robotak tresna desberdinak dituzte: kamerak, termometroak, barometroak, espektroskopia, etab. Aldi berean, eremuen ezaugarriak simulatzen dituzte, adibidez, gasen ihesak, oztopo fisikoak, arrisku erradioaktiboak, etab.

ÍNDICE DE CONTENIDO

REFERENCIAS DE CONTENIDO	2
Ilustraciones y Figuras	2
Tablas	2
Acrónimos	3
Definiciones y aclaraciones	3
1. INTRODUCCIÓN	4
2. CONTEXTO	5
3. ALCANCE Y OBJETIVOS	7
4. BENEFICIOS DEL PROYECTO	8
4.1. Beneficios técnicos	8
4.2. Beneficios económicos	8
4.3. Beneficios sociales	9
5. ESTADO DEL ARTE	10
6. DESCRIPCIÓN DE LA METODOLOGÍA	12
6.1. Planteamiento del problema	12
6.2. Descripción de requerimientos. Especificaciones	12
I. <i>Mapa del terreno</i>	12
II. Zonas a investigar	12
III. Agentes del algoritmo	13
IV. Comunicación entre agentes	14
6.3. Descripción de la solución propuesta. Diseño	14
6.4. Cálculos y algoritmos realizados	19
6.5. Descripción de los resultados	23
I. Comparación de los algoritmos	23
II. Visión individual de los agentes	25
III. Representación del movimiento del enjambre	27
7. PLANIFICACIÓN	29
7.1. Diagrama de Gantt y ciclo de vida	29
I. Estudio inicial y formación	29
IV. Desarrollo especializaciones sobre SDS	29
V. Desarrollo propuesta final	29

VI.	Gestión y documentación del proyecto	30
7.2.	Metodología de desarrollo	30
7.3.	Desarrollo software	31
8.	PRESUPUESTOS Y COSTES	33
8.1.	Proyecto de ingeniería	33
8.2.	Proyecto de implementación	34
9.	CONCLUSIONES	36
10.	BIBLIOGRAFIA	37

REFERENCIAS DE CONTENIDO

Ilustraciones y Figuras

Ilustración 1: Diagrama del algoritmo "Stochastic Diffusion Search"	10
Ilustración 2: Relación de la cuadrícula y el mapa real	15
Ilustración 3: Proceso de detección de elementos del mapa	16
Ilustración 4: Proceso de análisis de compatibilidad	17
Ilustración 5: Proceso de selección de nuevas hipótesis	17
Ilustración 6: Algoritmo del movimiento de un agente	19
Ilustración 7: Algoritmo de comunicaciones del enjambre sin UAV	20
Ilustración 8: Algoritmo de comunicaciones del enjambre con UAV.....	21
Ilustración 9: Algoritmo de comportamiento de los agentes voladores.....	21
Ilustración 10: Algoritmo del movimiento de un agente con criterio aleatorio	22
Ilustración 11: Inteligencia de enjambre con criterio aleatorio.....	23
Ilustración 12: Mapas usados para medir los resultados.....	24
Ilustración 13: Comparación de resultados obtenidos	24
Ilustración 14: Mapa explorado por la inteligencia aleatoria	25
Ilustración 15: Mapa explorado por la inteligencia aleatoria SDS sin UAV.....	26
Ilustración 16: Mapa explorado por la inteligencia SDS con UAVS.....	26
Ilustración 17: Etapas de movimiento.....	27
Ilustración 20: Diagrama de planificación GANT.....	30

Tablas

Tabla 1. Costes: Horas Internas.....	33
Tabla 2. Costes: Amortizaciones	33
Tabla 3. Costes: Gastos	33
Tabla 4. Costes: Total	34
Tabla 5. Presupuesto: Horas Internas	34
Tabla 6. Presupuesto: Amortizaciones.....	35
Tabla 7. Presupuesto: Gastos.....	35
Tabla 8. Presupuesto: Total.....	35

Acrónimos

SDS: Stochastic Diffusion Search

UAV: Unmanned Aerial Vehicle

Definiciones y aclaraciones

Agente: Un robot con las capacidades de comunicarse y cooperar con otros. En este proyecto se definen de dos tipos: terrestres y UAVs. Los primeros son robots que se desplazan por el suelo y los segundos son robots que se desplazan por el aire

Enjambre: Conjunto de agentes que se organizan y cooperan para realizar funciones que no podrían realizar por sí solos.

Mapa: Extensión de terreno sobre la que se ejecuta el programa desarrollado.

Zona: Elemento a delimitar o acotar dentro del mapa

Tarea: Acción de delimitar o acotar una zona

Hipótesis: Suposición que realiza cada agente sobre el lugar donde se podría encontrar una zona inexplorada, basándose en la información que conoce.

Activo: Estado en el que se puede encontrar un agente. Significa que está realizando una tarea.

Pasivo: Estado en el que se puede encontrar un agente. Significa que está explorando el mapa en busca de una tarea.

1. INTRODUCCIÓN

Este proyecto presenta el desarrollo de una inteligencia de enjambre distribuida capaz de realizar labores de exploración de forma autónoma. La idea de este proyecto nace como un intento para mejorar las técnicas actuales para el reconocimiento de todo tipo entornos, de forma que no sea necesaria la intervención de seres humanos. Esta decisión parte del hecho de que la mayoría de estos trabajos de exploración suponen un peligro para la salud de las personas que la realizan. Es decir, con la solución realizada por este proyecto se evita el tener que mandar personas para mapear lugares donde hayan ocurrido desastres naturales, grandes desprendimientos, fugas radiactivas u otros sucesos peligrosos.

La solución planteada se enfoca en localizar y delimitar unas zonas esparcidas por una gran extensión de terreno. Estas zonas simulan lo que serían lugares de interés dentro del propio terreno, y será el objetivo del enjambre encontrarlas y delimitarlas todas. Este algoritmo está planteado de una forma distribuida, por lo tanto la información con la que cuentan los agentes que conforman este enjambre está limitado a la información que dichos agentes han logrado conseguir por sus propios medios.

En este informe se presenta el proceso de desarrollo de la anteriormente mencionada inteligencia de enjambre, así como una explicación detallada de cada una de sus funcionalidades. Primero, se expone el contexto que rodea al proyecto, momento en el cual se explica las aplicaciones de los algoritmos de inteligencia de enjambre tanto en problemas de robótica como en problemas matemáticos. A continuación se especifican el alcance y objetivos del proyecto. Este proyecto cuenta con un objetivo principal y varios secundarios. Después, se indican los beneficios técnicos, sociales y económicos que acompañan a la utilización de esta inteligencia desarrollada. Posteriormente, se realiza un estudio del arte del algoritmo *“Búsqueda por difusión estocástica”* (Stochastic diffusion search), explicando su funcionamiento y exponiendo las áreas en las que actualmente se están aplicando este algoritmo.

Una vez realizada esta introducción al proyecto, se pasa a exponer la metodología realizada. En esta metodología se explica el trabajo realizado durante todo el proceso de desarrollo. Primero, se presenta un resumen de la problemática que se trata de solucionar. Después, se listan las especificaciones y requerimientos planteados para el proyecto. A continuación, se explica la propuesta de diseño realizada, explicando en detalle cada una de las funcionalidades y mecánicas desarrolladas. Posteriormente, se exponen los algoritmos que conforman la inteligencia desarrollada. Para ayudar a la explicación se han incluido varios diagramas de estos algoritmos. Por último, se analizan los resultados obtenidos por la inteligencia propuesta. Se tratarán estos resultados desde un punto de vista comparativo, tratando de distinguir entre todos los modelos presentados.

Para terminar con el informe, se incluye la planificación seguida durante todo el proceso de desarrollo. Esta planificación se hace dividida en tareas, las cuales se explican individualmente de forma detallada. Estas tareas se representan después en un diagrama GANT. Después, se incluyen los costes para la realización de este proyecto, así como el presupuesto necesario para poder proyectarlo.

2. CONTEXTO

La robótica es un área de la ingeniería que está desarrollándose a una velocidad vertiginosa y que se ha vuelto tremendamente popular. Esto se debe a las recientes mejoras en la capacidad de cómputo, la creación de baterías más eficientes y la automatización de tareas y trabajos. Gracias a estos avances tecnológicos, cada vez es más común ver robots en todo tipo de entornos. Un claro ejemplo son los drones aéreos que tan populares se han hecho últimamente o las aspiradoras autónomas que aspiran la suciedad de muchas casas. No cabe duda de que cada vez son más numerosos y realizan más funciones. Por eso mismo surge la necesidad de proveerlos cada vez de una inteligencia más eficiente.

Este proyecto se centra en concreto en los robots que realizan tareas de exploración. El cual es un campo de la robótica bastante extenso, pues engloba tanto a los anteriormente mencionados robots aspiradoras como a los más complejos robots de exploración espacial. Lo que conlleva a que exista un sinfín de métodos de exploración, cada uno con sus peculiaridades, sus ventajas y sus desventajas. Las últimas tendencias tienden al uso de un gran número de robots que cooperan entre ellos y que son de distintos tipos: aéreos, terrestres, anfibios, etc. Para realizar esta inteligencia es necesario robustos algoritmos para la comunicación o coordinación entre robots y también para definir el comportamiento individual de cada uno. Esta subrama de la robótica va a ser el centro de estudio de este proyecto: la inteligencia de enjambre.

El término inteligencia de enjambre (swarm intelligence en inglés) fue creado por Gerardo Beni y Jing Wang [1] y engloba a todos los algoritmos de inteligencia robótica colectiva descentralizada. Estos algoritmos consisten en, primero, definir varios agentes y, segundo, definir las reglas y métodos con los que los agentes se comunicarán o cooperarán para poder realizar tareas o solucionar problemas. Estos problemas pueden ser demasiado complejos como para poder ser resuelto por un único agente o requieran múltiples agentes para poder maximizar su tiempo de resolución. Además, se dice que son algoritmos descentralizados o distribuidos porque los agentes no disponen de ninguna información que no hayan recogido ellos mismos o que no hayan recibido de otro agente. Es decir, en todo momento cuentan únicamente con información parcial y no absoluta. Por esto mismo, depende del entorno o el contexto, los algoritmos tienden a utilizar mecanismos de filtrado y obsolescencia de la información.

Aunque existen algoritmos no basados en comportamientos de la naturaleza, como la optimización de enjambre de partículas o el de búsqueda de difusión estocástica, muchos algoritmos han surgido tras su observación y estudio, tanto de animales como microbios o bacterias. Es por ello que la inteligencia de enjambre recibe el nombre de “enjambre”, porque tratan de simular el comportamiento de un grupo de determinados animales. Un ejemplo típico es el algoritmo de optimización de colonia de hormigas o el del panel de abejas, los cuales consisten en organizar un enjambre con agentes de distintos roles para encontrar “fuentes de comida” de la forma óptima.

La inteligencia de enjambre no sirve únicamente para crear robots que simulen comportamientos de la naturaleza. Se puede hacer una extrapolación de todos estos algoritmos para que también sirvan como sustitutos a algoritmos tradicionales cuando no se puede implementar soluciones basadas en herramientas matemáticas como el cálculo o las matrices por ser demasiado lentas o demasiado imprecisas. Esto es precisamente lo que en ciencia computacional se define como algoritmos metaheurísticos [2] [3] [9]. Es decir, las técnicas de enjambre son metaheurísticos en vez de algoritmos rígidos.

Estos tipos de algoritmos necesitan más procesamiento que los algoritmos clásicos basados en cálculo y no está garantizado que encuentren la solución óptima a todos los problemas. Por eso mismo hay que recurrir a algoritmos especializados para cada tipo de problema. Esto hace que sea importante saber categorizar cada caso de uso y determinar qué tipo de algoritmo se puede emplear. Cabe remarcar que no hay necesidad de usar métodos de enjambre cuando funcionan los métodos tradicionales, pero que los métodos tradicionales no siempre funcionan para resolver los problemas metaheurísticos.

3. ALCANCE Y OBJETIVOS

En este apartado se define y explica el objetivo principal que se persigue con este proyecto y el alcance del mismo. Después se presentarán varios objetivos secundarios que tratan de mejorar u optimizar el algoritmo desarrollado. Estos objetivos secundarios, a pesar de parecer que no son primordiales mejoran en gran medida varios aspectos del algoritmo propuesto, por lo tanto, se les consideran de gran utilidad para el diseño final.

El principal objetivo del proyecto es desarrollar una inteligencia de enjambre, formado por numerosos agentes, capaz de explorar de forma autónoma una gran extensión de terreno. La exploración se realizará con el propósito de delimitar y acotar determinadas zonas o áreas diseminadas a lo largo de la extensión de terreno. Este escenario representa lo que podría ser una ciudad en ruinas, una cueva inexplorada o un planeta como Marte y las áreas serían zonas inaccesibles o peligrosas para el ser humano como fugas radioactivas, lagos contaminados o grietas profundas.

Una vez definido el objetivo principal, se deben cumplir objetivos secundarios que, mediante diferentes mecanismos, optimicen y mejoren el proceso principal. Una de estas mejoras es maximizar el área explorada. De tal forma que, mediante la coordinación y la comunicación entre agentes, se encuentren las zonas a delimitar lo más rápido posible. Esta optimización se puede conseguir si cuando un agente decide moverse, se mueva acorde a criterios que traten de reducir la incertidumbre que se tiene acerca del mapa.

Otro objetivo secundario sirve para poder dividir los agentes por el mapa de forma óptima. Es decir, si se conoce una zona sin delimitar es necesario que los agentes analicen si son útiles para cumplir esa tarea y en caso de serlo encaminarse hacia esa zona. Para ello es necesario que los agentes sean capaces de cuantiar su nivel de utilidad relativa a cada zona que conozca, de manera que se pueda decidir a donde dirigirse.

Por último, otro factor en el que se podría ganar eficiencia es en la comunicación entre diferentes agentes. Esta comunicación es la principal diferencia que separa el movimiento aleatorio del movimiento inteligente del enjambre. Por lo tanto, hay que diferenciar que tipos de datos se van a transmitir, cuando se van a transmitir y como. El optimizar el intercambio de datos supone una optimización prácticamente inmediata del funcionamiento del algoritmo principal.

4. BENEFICIOS DEL PROYECTO

4.1. Beneficios técnicos

El algoritmo desarrollado proporciona una sustancial mejora del algoritmo de búsqueda por difusión estocástica aplicado al ámbito de la exploración y caracterización de diferentes entornos. Esto es debido a que se ha implementado diferentes optimizaciones y alteraciones que convierten a la solución propuesta en una mejora más específica y eficaz. Una de estas optimizaciones es un sistema robusto y eficiente de selección de hipótesis. Estas hipótesis son usados por diferentes agentes en diferentes contextos para determinar cuál es el movimiento óptimo que deben realizar.

Cabe destacar que todos los agentes son totalmente autónomos y que no necesitan en ningún momento la intervención humana. Los agentes son lo suficientemente inteligentes para ser capaces de determinar a donde dirigirse, con quien comunicarse y saber si pueden delimitar un área. Esto, sumado a su capacidad para diferenciar que agentes se necesitan para cada zona, hace que el algoritmo sea adaptable a multitud de tipos de terreno.

4.2. Beneficios económicos

El alcance del proyecto se centra en el desarrollo software del algoritmo de inteligencia de enjambre, la cual puede ser vendida o comercializada de diferentes maneras. Por un lado se puede crear un framework para solucionar problemas metaheurísticos que necesitan algoritmos especiales para resolverlos. Esta podría ser una comercialización más orientada al desarrollo matemático y la investigación. Por otro lado, siguiendo con un enfoque más cercano al estado actual del proyecto, se podría desarrollar un sistema que proporcione servicios o herramientas de simulación. Estas simulaciones podrían ser desde estudios de prevención de amenazas en una reserva forestal hasta implementaciones específicas en videojuegos.

En un futuro, si se llevase el proyecto un paso más adelante y si se optase por seguir desarrollando el proyecto, se podría implementar y crear el hardware de los robots para poder proceder a la comercialización o la venta de estos. Cuando se haya alcanzado este punto, se debería hacer un estudio del mercado para poder determinar el precio de venta correspondiente al producto.

En este punto, el principal beneficio económico para el cliente es la reducción de costes, tanto directos como indirectos, de las labores de exploración. Es decir, gracias a la solución que se propone no sería necesario contratar a un gran número de personal ni de suministrarles ninguna formación ni equipo para realizar trabajos de exploración de entornos extremos. Esto es debido a que el enjambre está diseñado para trabajar de una manera totalmente autónoma, sin ninguna intervención humana y en cualquier entorno independientemente de las características de este.

4.3. Beneficios sociales

Este proyecto desarrollar una inteligencia en enjambre que realice funciones de exploración en entornos peligrosos, de tal forma que se la intervención humana no sea necesaria. El beneficio social es que se evita enviar seres humanos a lugares potencialmente peligrosos, donde pondrían en riesgo su salud. Ejemplos de estos lugares pueden ser: lugares donde hayan ocurrido fugas radiactivas, accidentes nucleares como Chernóbil, montañas con alto peligro de desprendimientos, lagos contaminados, etc.

También proporciona una optimización en el tiempo necesario para realizar la exploración. Esto es debido a que, al tratarse de una inteligencia de enjambre de robots independientes, los agentes se pueden especializar en función del entorno que vayan a explorar. De tal forma que sean lo más resistentes y flexibles posible en todo momento. Logrando que el enjambre pueda llegar a lugares a los que un humano no podría ir y realizando las tareas que un humano no podría realizar.

El algoritmo no es solo útil para determinar el estado de un entorno hostil, tanto provocado por un desastre natural como a consecuencia de la intervención humana, sino que también es de utilidad en el ámbito técnico y científico. Debido a que puede ser usado para multitud de tareas de investigación. Es decir, este algoritmo, dentro del ámbito científico, es de utilidad para tareas grandes como crear robots para explorar otros planetas o para tareas pequeñas como desarrollar múltiples micro-robots inteligentes.

5. ESTADO DEL ARTE

El algoritmo desarrollado en este proyecto es una derivación del algoritmo de inteligencia de enjambre “*Búsqueda por difusión estocástica*” (Stochastic diffusion search en inglés) creado por Mohammad Majid al-Rifaie y John Mark Bishop en el año 1989 [4]. El cual se considera un algoritmo metaheurístico que resuelve de una forma eficiente problemas de búsqueda y optimización que no tienen un procedimiento específico para obtener una solución satisfactoria.

El algoritmo SDS (Stochastic Diffusion Search) puede ser fácilmente aplicado a problemas de optimización donde se pueda descomponer el objetivo final en diferentes componentes que se puedan evaluar por separado. De esta forma, cada agente del enjambre puede tener una hipótesis sobre cada componente y, mediante cooperación y comunicación, llegar a la hipótesis final más óptima.

Para evaluar estas hipótesis, el algoritmo SDS se divide en varias fases. Una primera de inicialización de agentes, seguido por un bucle de análisis y difusión hasta que todos los agentes lleguen a la convergencia de cuál es la hipótesis final óptima, la cual se presentará como el resultado del algoritmo. (Ver ilustración 1).

En la primera fase, el enjambre define a sus agentes, divide el objetivo final en varios segmentos o hipótesis, y los divide entre los diferentes agentes. En la segunda fase, cada agente se encarga de evaluar su segmento y trata de comprobar su hipótesis parcial. En la tercera fase, los agentes se comunican para comparar los resultados obtenidos, descartando las soluciones no válidas. Este proceso de análisis y difusión se repite reiteradamente con diferentes hipótesis hasta que se haya llegado a la convergencia de todos los agentes. Momento en el cual, el algoritmo presenta la hipótesis conjunta de todos sus agentes, y se crea la solución final al problema planteado.

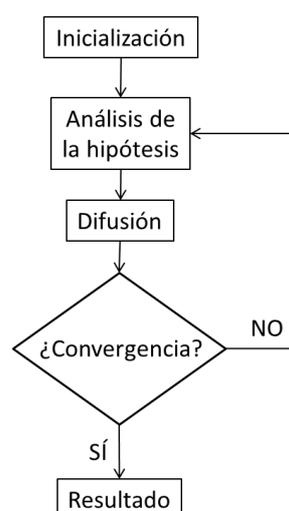


Ilustración 1: Diagrama del algoritmo "Stochastic Diffusion Search"

Una forma sencilla de entender este algoritmo es mediante el ejemplo de “el juego de los mineros” [5]. El cual dice lo siguiente: “Un grupo de mineros saben que hay oro Escondido en una agrupación de colinas pero no saben su distribución. Tratando de maximizar la riqueza colectiva deciden cooperar y ayudarse para encontrar las mejores concentraciones de oro. Para ello los mineros utilizarán una búsqueda estocástica por difusión simple. Al inicio del proceso de minado, los mineros eligen una colina aleatoria (hipótesis). Cada día los mineros cavan una región de esa colina y al final del día analizan lo exitosos que han sido en función del oro recogido (fase de análisis). Cada noche los mineros se reúnen y cada minero que no es exitoso selecciona otro minero al azar para comunicarse con él. Si el segundo minero ha sido exitoso, el primero se dirige a la colina del segundo. Si el segundo minero no ha sido exitoso, el primer minero (fase de difusión). Iterando este proceso se consigue explorar todo el espacio y se hace que los minero se concentren donde más oro haya.”

Las utilidades reales de este algoritmo son casi infinitas ya que sus principios son muy versátiles y fáciles de adaptar a cualquier tipo de escenario o contexto. Por lo tanto es fácil de ver aplicado a proyectos en ámbitos muy distintos como la medicina, la biología, la descodificación o el reconocimiento de patrones. A continuación se exponen tres publicaciones como ejemplo de las posibles aplicaciones que se le pueden dar a este algoritmo. En el primero [6] se indaga en la utilidad de utilizar del algoritmo para fines médicos, más concretamente para la detección de las células cancerígenas provocadas por la metástasis en radiografías del cuerpo humano. El segundo [7] propone utilizar el algoritmo para solucionar problemas derivados del ensamblaje de células de ADN. Por último, el tercero [8] plantea el algoritmo para ser utilizado en labores de reconocimiento de patrones, con el objetivo de poder traducir un texto escrito a mano a un archivo informático.

6. DESCRIPCIÓN DE LA METODOLOGÍA

6.1. Planteamiento del problema

Como ya se ha mencionado en apartados anteriores, el problema a solucionar es una tarea de exploración. Esta tarea de exploración debe ser totalmente autónoma, es decir, sin intervención humana. La extensión a explorar será un mapa cuadrado con diversas zonas circulares esparcidas a lo largo del mapa. El objetivo de este proyecto es desarrollar una inteligencia de enjambre de múltiples agentes que sean capaces de coordinarse entre ellos para delimitar estas zonas.

La forma con la que se comprobará la eficacia de la solución final será mediante una comparativa de eficiencia, en la que se comparará la solución propuesta con el movimiento de un enjambre dirigidos por decisiones aleatorias. La diferencia entre los resultados obtenidos deben ser significativamente grandes para considerarlos relevantes.

6.2. Descripción de requerimientos. Especificaciones

Al inicio del desarrollo de la solución se han planteado varios requerimientos o especificaciones para determinar cómo será el escenario y qué condiciones rigen el comportamiento del enjambre. Estas especificaciones son las condiciones mínimas que se deben cumplir obligatoriamente.

I. Mapa del terreno

Se entiende como mapa al terreno sobre el que operará el enjambre robótico, la cual simulará a una extensión de terreno real. Este mapa será cuadrado, de un tamaño $N \times N$, donde N es un parámetro de entrada que puede ser modificado de ejecución a ejecución.

II. Zonas a investigar

Las zonas a investigar son los lugares en los que los robots deben hacer sus tareas de investigación. Estas tareas de investigación consisten en encontrar unos puntos repartidos por el borde de las zonas, de tal forma que los robots acaben delimitándolas. Por eso mismo, y para acercarlo lo máximo posible a la realidad, los robots no disponen de ninguna información inicial de las zonas y los puntos. Es decir, no conocen ni el número, ni la localización de estos datos.

Adicionalmente, las zonas se deben diferenciar en diferentes tipos y admitir únicamente a aquellos robots que cumplan los requerimientos para identificar ese tipo de zonas. Este requerimiento se establece con la intención de simular las naturalezas de las zonas, las cuales en la vida real podrían ser totalmente diferentes, como por ejemplo: fugas radiactivas, lagos contaminados u obstáculos físicos.

Por otro lado, para permitir mayor control del algoritmo, tanto el número de zonas como el número de puntos son parámetros que se introducen al inicio de la ejecución y pueden variar entre ejecuciones. En relación con esto, el algoritmo tendrá que tener en cuenta que las zonas no se pueden superponer y que por lo tanto la distribución de las zonas en la inicialización del mapa debe hacerse de forma inteligente. Adicionalmente, para poder realizar unas mediciones precisas se debe poder activar o desactivar el factor de aleatoriedad que distribuye las zonas

por el mapa. De tal forma que se pueda elegir si se quiere utilizar el mismo mapa entre ejecuciones o si cada ejecución va a utilizar un mapa nuevo completamente aleatorio.

III. Agentes del algoritmo

Los agentes de este algoritmo van a ser unos robots con una inteligencia de enjambre. Esta inteligencia es una versión modificada del conocido algoritmo “*Búsqueda por difusión estocástica*” (Stochastic diffusion search). Las modificaciones se hacen con el propósito de poder realizar los objetivos de este proyecto de una forma mucho más eficaz. Por lo tanto, heredarán de este algoritmo varias características que lo definen e incorporarán otras nuevas.

La principal característica que hereda el agente es la capacidad de poder cambiar de estado entre activo y pasivo. Es decir, cuando el robot este en estado pasivo se encontrará explorando el mapa, realizando hipótesis de donde se podrían encontrar las zonas y tratando de comunicarse cuando se cumplan las condiciones descritas en el apartado IV de esta sección. En cambio, cuando el estado del robot sea activo, este habrá encontrado una zona que puede delimitar y estará realizando dicha tarea.

Otra característica heredada es la descentralización del enjambre. Lo cual quiere decir que los agentes disponen única y exclusivamente de la información que o bien han recogido ellos de primera mano o han recibido vía comunicación con otro agente. Esta característica implica que, mientras pueden conocer el número total de agentes en el enjambre no pueden saber la localización específica de todos ellos. Si a esta característica le sumamos el hecho de que el agente no dispone de ninguna información inicial de las tareas que ha de realizar, obtenemos un agente plenamente independiente que necesita la ayuda de otros agentes para realizar su misión de la forma más óptima posible.

Para poder interactuar con el entorno los agentes deben incorporar mecanismos de observación y detección eficaces pero que, al mismo tiempo, sean realistas. Por eso mismo, los robots no tienen la capacidad de observar todo el mapa, si no que constan de un cono de visión con un alcance máximo. Los agentes solo serán capaces de determinar lo que está dentro de su cono de visión, diferenciándolo entre zona o no. Cabe destacar que aunque vean una zona en su cono de visión, no se considera que sepa la localización exacta de la zona hasta que se haya introducido en ella.

Por último, como se ha mencionado en el apartado II, las zonas se dividen en diferentes tipos, por lo tanto, los robots se han de dividir también en tipos. De esta forma los robots serán capaces de saber que zonas pueden delimitar y cuáles no. Volviendo a hacer el símil con la realidad, con esta especificación se trata de determinar que herramientas tiene incorporado el robot. De tal forma que, por ejemplo, para delimitar una fuga radiactiva se necesite a un robot con un contador Geiger.

IV. Comunicación entre agentes

La comunicación entre agentes es primordial para poder desarrollar una inteligencia de enjambre eficiente. Es la característica principal, pero no la única, que separa un movimiento guiado por un algoritmo inteligente de un movimiento sin inteligencia realizado mediante decisiones aleatorias.

Para realizar las comunicaciones se va a seguir la metodología original planteada en las bases del algoritmo *“Búsqueda por difusión estocástica”* (Stochastic diffusion search), el cual determina que el agente debe comunicarse con un único agente y tan solo cuando haya llegado a la fase de difusión. Adicionalmente, se introducirá otro limitante a la comunicación: la cobertura. Esta será la distancia máxima a la que se debe encontrar otro agente para poder realizar la comunicación. Este valor, nuevamente, será un parámetro de entrada que puede variar de ejecución a ejecución. Si un robot ha llegado a su fase de difusión y no tiene ningún otro robot dentro de su alcance de cobertura tendrá que realizar una nueva hipótesis mirando únicamente sus datos. Con todo esto se consiguen intercambios de datos realistas donde la distancia entre los interlocutores es importante para realizar una comunicación satisfactoria.

6.3. Descripción de la solución propuesta. Diseño

Con el fin de cumplir las especificaciones descritas anteriormente se ha desarrollado una inteligencia con varias funcionalidades específicas. Estas funcionalidades también se desarrollan con el propósito de mejorar la ejecución del enjambre. A continuación se irán haciendo una descripción de dichas soluciones.

Para empezar se ha dividido el desarrollo software en dos mitades. Por un lado, se desarrolla la inteligencia de un solo robot. Es decir lo que un solo agente puede hacer por su cuenta dentro del algoritmo. Estas capacidades son, entre otros, moverse por el mapa, registrar los puntos que delimitan las zonas y cambiar entre activo y pasivo para realizar las distintas tareas. Por otro lado, se ha desarrollado la inteligencia colectiva. De esta forma los agentes no tienen acceso a la información de los otros agentes. En esta lógica se define cómo los agentes del enjambre interactúan, cómo se comunican y cómo cooperan entre ellos. La combinación de estas dos lógicas conforma la inteligencia de enjambre propuesta.

Así como la lógica está dividida, la creación del propio mapa sobre el que corre el programa también está dividida del algoritmo principal. De esta forma se logra que el algoritmo no contenga ninguna información inicial del terreno, de hecho, el objetivo final del algoritmo es descubrir por sí mismo la información generada por el generador de mapas. Durante la inicialización del programa se llama al creador de mapas para que genere un mapa nuevo totalmente aleatorio, a no ser que se especifique lo contrario, en cuyo caso el creador siempre generará el mismo mapa. Esta última característica es especialmente útil para realizar mediciones y análisis. En la creación del mapa se distribuyen un número de zonas iguales a las especificadas al inicio del programa. En principio tanto el radio como el centro de estas zonas es aleatorio, siempre y cuando no se superpongan o se indique que no se desea una generación aleatoria.

En cuanto al mapa generado, este es una extensión lineal de dos dimensiones. Este mapa se encuentra en el primer cuadrante de un plano cartesiano, lo que significa que el valor de 'x' o de 'y' en ese mapa no puede valer negativo. Además, tampoco pueden superar el valor especificado como tamaño máximo. Al tratarse de un mapa lineal, los puntos que lo conforman son infinitos y cualquiera de esos puntos pueden ser las posiciones de los agentes o de las zonas. Por lo tanto, con el objetivo de poder registrar la información del mapa, cada agente cuenta con una cuadrícula que corresponde con el mapa a descubrir. Gracias a uso de esta cuadrícula, también denominada mapa de exploración, los agentes son capaces de simplificar los datos que perciben. Este proceso es especialmente importante ya que se cuenta con una capacidad de cómputo limitada, lo que hace casi imposible tratar con todos los datos disponibles en la extensión del mapa. Teniendo todo esto en cuenta, se ha definido una cuadrícula de exploración diez veces más pequeña que la extensión del mapa. Este tamaño de cuadrícula permite una ejecución fluida a la vez que evita cualquier pérdida de información relevante.

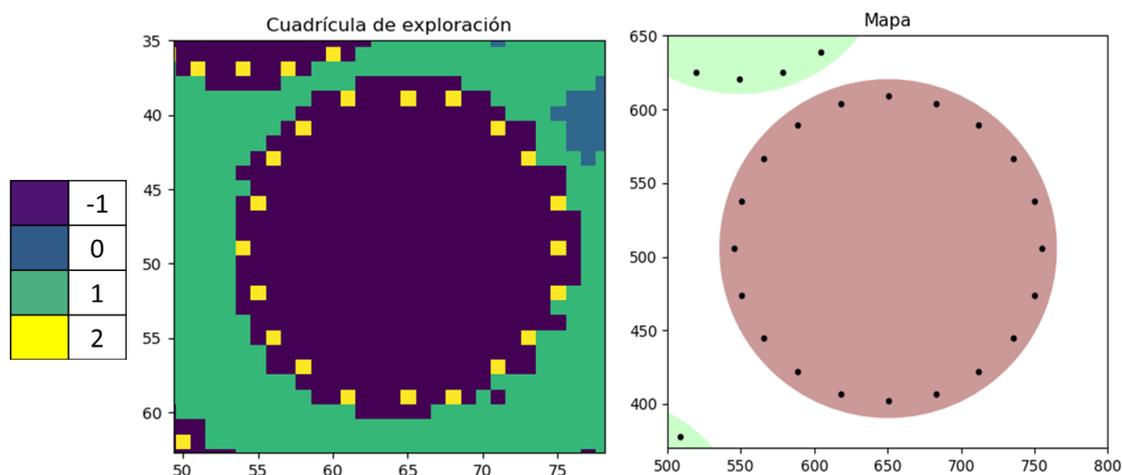


Ilustración 2: Relación de la cuadrícula y el mapa real

En la ilustración 2 se puede observar cómo se traduce una zona del mapa real a la cuadrícula de exploración anteriormente descrita. La cuadrícula contiene valores numéricos que hacen referencia al contenido real del mapa. Dicha numeración es la siguiente: el menos uno representa que ese cuadrado corresponde a una zona, el cero indica que ese cuadrado aún no ha sido explorado, el uno se traduce como aire (no zona) y el dos como los puntos que hay que encontrar con el fin de delimitar la zona y acabar con la tarea. Cuando se arranca la ejecución del programa la cuadrícula de todos los agentes está inicializada a cero. Finalmente, se realiza una representación gráfica de esos valores para poder apreciar el trabajo realizado por el enjambre.

Adicionalmente, la cuadrícula de exploración también es útil a la hora de crear el cono de visión del agente. El cual, al estar colocado en un plano de dos dimensiones está representado como un triángulo isósceles. El agente siempre está en un vértice de este triángulo y la mediana que une ese punto con el centro del lado opuesto corresponde con la dirección y

sentido el agente. De esta forma, el cono siempre está orientado dependiendo del movimiento que realiza el agente. Este cono de visión establece que puede detectar un agente y que no puede detectar, de tal forma que lo que se encuentra en el interior del cono es detectable y lo que se encuentra en el exterior indetectable.

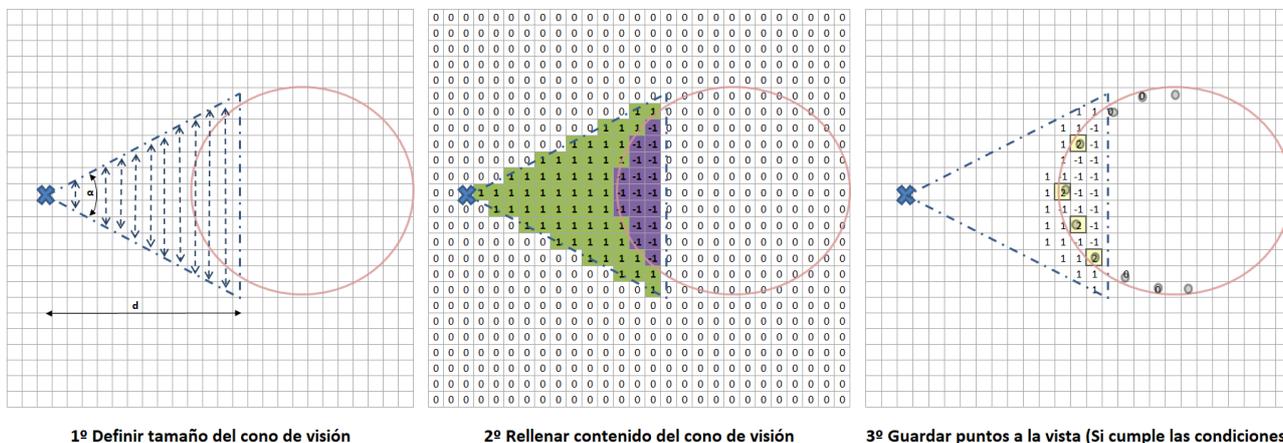


Ilustración 3: Proceso de detección de elementos del mapa

En la ilustración 3 se representa el proceso que se realiza para determinar la visión del agente. Primero, en función del ángulo de visión "alfa" y la distancia de visión "d", se crea el triángulo isósceles y se definen una sucesión de líneas paralelas al lado opuesto del vértice en el que se encuentra el agente. Segundo, se determina esas líneas a qué cuadrados del mapa real corresponden, en función de lo que se encuentre en ese lugar del mapa se guarda un valor u otro en la cuadrícula (-1 si zona o 1 si aire). Tercero, el agente analiza si está en estado activo (porque el tipo del agente y de la zona son compatibles) y en caso de estarlo, sobrescribe en su cuadrícula de exploración el cuadrado correspondiente a la ubicación de un punto.

Sin embargo, antes de poder delimitar una zona y poder encontrar los puntos que la conforman es necesario determinar si el tipo de robot y el tipo de la zona son compatibles. El número de tipos, tanto de agentes como de zonas, es el mismo y cada zona o agente puede ser de más de un tipo. Cuando un agente entra en una zona, independientemente de si ha llegado a su hipótesis, sigue un proceso en el que determina si debe pasar a estado activo con el fin de poder delimitar la zona a la que acaba de introducirse. Una vez delimitada, vuelve a ponerse en estado pasivo y sigue explorando el mapa.

En la ilustración 4 se explica el proceso seguido para determinar si un agente se puede conmutar a estado activo. Primero determina si ya ha sido acotada y si dispone de toda la información de esa zona, ya que no tiene sentido realizar las tareas de exploración en una zona y explorada. En caso negativo analiza si al menos uno de sus tipos corresponde con uno de los tipos de la zona, si coinciden significa que los dos son compatibles y por lo tanto el agente pasa a ser activo y se pone a delimitar la extensión de la zona.

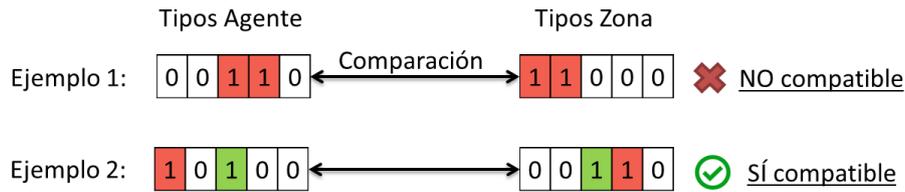


Ilustración 4: Proceso de análisis de compatibilidad

Delimitar todas las zonas esparcidas por el mapa es el objetivo final del algoritmo. La forma de hacerlo es que los agentes se dividan para encontrar todos los puntos que están en los extremos de las zonas. Con el propósito de minimizar el tiempo necesario para realizar estas tareas, los robots establecidos como activos dentro de una zona irán recorriendo el borde de dicha zona a la altura a la que se encuentran los susodichos puntos a detectar.

Con el propósito de evitar la necesidad de que todos los robots descubran todos los puntos, los agentes se intercambian los mapas, o cuadrículas de exploración, en la fase de difusión. Es decir, cuando un agente haya llegado a la fase de difusión trata de buscar otro agente, si detecta otro agente a rango, comparten los datos relativos al mapa de exploración y al lugar donde se encuentran las zonas. De esta forma se logra que agentes individuales se coordinen y cooperen para realizar las tareas que tienen pendientes.

Todos estos procesos se realizan mientras el agente se dirige a una hipótesis o una vez llegado a esta. Una hipótesis es una suposición que realiza cada agente sobre el lugar donde se podría encontrar una zona inexplorada. Por lo tanto, cada vez que llega al lugar donde ha hipotetizado que podría encontrarse una zona, debe, salvo en casos especiales, realizar una nueva hipótesis. El proceso que sigue una vez llegado a su hipótesis se muestra en la ilustración 5 y es el siguiente: Primero, intenta realizar una comunicación con otro agente. Luego, mira si conoce alguna zona con la que tenga un tipo en común. De ser así se dirige hacia esta para poder delimitarla. En caso de haber más de una se dirige a la que más tipos tenga en común. Si, en cambio, no conoce ninguna zona inexplorada busca en su cuadrícula de exploración un punto marcado como no explorado y lo establece como su nueva hipótesis.

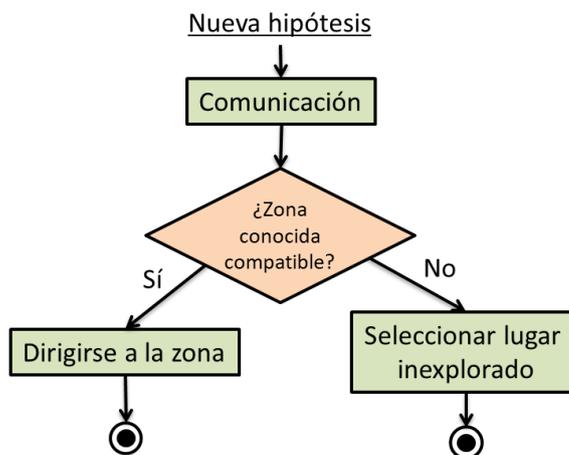


Ilustración 5: Proceso de selección de nuevas hipótesis

Una vez explicadas todas las funcionalidades desarrolladas para satisfacer las especificaciones planteadas, cabe señalar que se han desarrollado dos algoritmos diferentes que implementa todo el diseño anterior. Los cuales, aunque uno tenga mejor rendimiento que otro, son igualmente válidos al cumplir todos los requisitos especificados. La diferencia entre estos dos algoritmos es la utilización de agentes ‘terrestres’, denominados ‘robots’, o el uso de agentes ‘terrestres’ y ‘voladores’, denominados ‘UAV’. Adicionalmente se ha creado un algoritmo que toma decisiones aleatorias para realizar comparaciones sobre la eficacia de los dos algoritmos desarrollados.

El primer algoritmo, denominado SDS sin UAV o simplemente SDS, realiza las labores de exploración utilizando únicamente los anteriormente mencionados robots terrestres. Estos robots terrestres se comunican entre sí para obtener la mayor información posible, aun así, al tratarse de un enjambre descentralizado, puede suceder que un robot se encuentre solo por un periodo largo de tiempo y no haya podido intercambiar su conocimiento con el enjambre. Este hecho puede suceder bastante a menudo en un mapa lo suficientemente grande, lo cual generalmente aumenta el tiempo necesario para encontrar todos los puntos.

El segundo algoritmo, se utiliza un enjambre compuesto de agentes voladores (UAVs) y terrestres (robots). Por un lado, los robots se ocuparán de explorar el mapa y de realizar las tareas de exploración que se encuentren. Por otro lado, los UAVs servirán para realizar la comunicación entre los distintos robots, procurando que todos los robots tengan el máximo de información actualizada en cada momento. Con el objetivo de maximizar la cantidad de información que se transmite, estos agentes solo podrán comunicarse con los UAVs. Además, como ya se ha especificado, esta comunicación se realizará únicamente cuando se haya alcanzado su hipótesis. Con esta solución se evita el caso descrito en el primer algoritmo. Es decir, los UAVs tratarán de dar conectividad a todos los robots, con el propósito de servir para recoger la información de cada robot y distribuirlas entre los demás robots terrestres. De forma que todos los robots dispongan de la información necesaria para realizar la exploración lo más rápidamente y eficientemente posible.

6.4. Cálculos y algoritmos realizados

Como se ha mencionado en el apartado anterior, se han realizado dos inteligencias de enjambre. Ambos están divididos en dos algoritmos, uno que controla lo que un agente hace por su cuenta, de forma independiente al enjambre, y otro que controla el modo de cooperar entre los agentes del enjambre. Este primer algoritmo, el del comportamiento individual del enjambre, es igual para las dos inteligencias y se muestra en la ilustración 6.

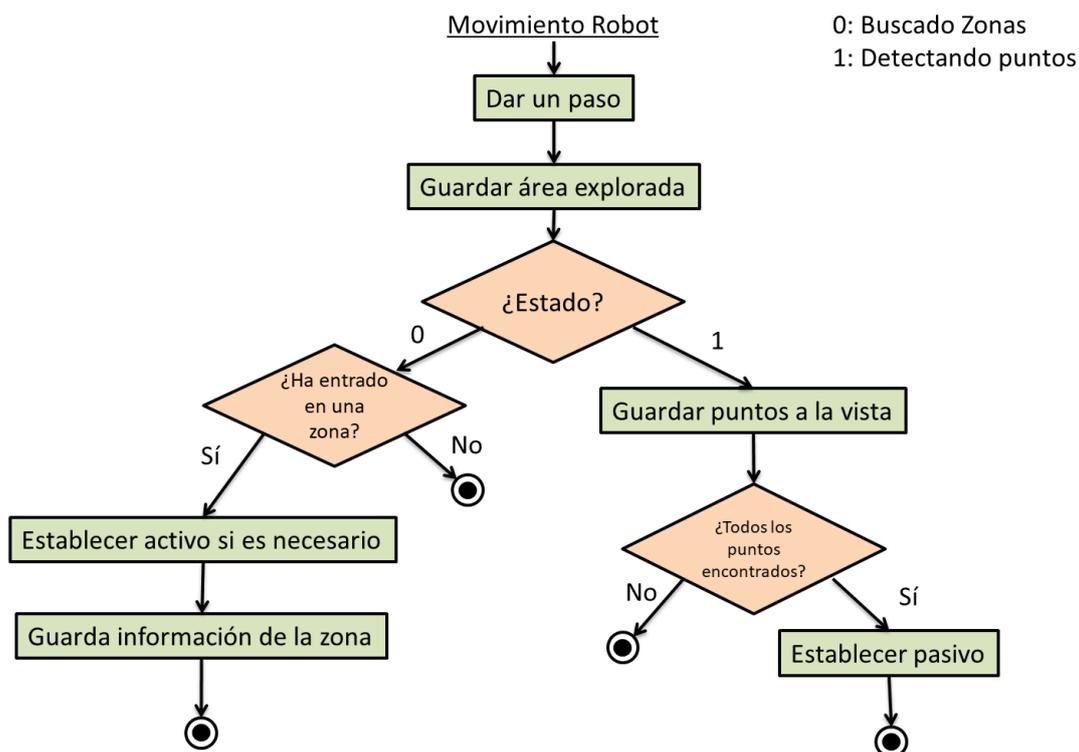


Ilustración 6: Algoritmo del movimiento de un agente

En este primer algoritmo, mostrado en la ilustración 6, lo realiza cada robot en cada instante de tiempo e inicia con el agente avanzando en la dirección en la que se encuentra su hipótesis. Luego, almacena en su cuadrícula de exploración la información que tiene a la vista (dentro de su cono de visión) ignorando los posibles puntos que conformen las zonas. A continuación, realiza funciones diferentes en función del estado en el que se encuentre, es decir, si se encuentra explorando el mapa y buscando zonas o si se encuentra delimitando las zonas a explorar. Por un lado, si se encuentra explorando (pasivo en estado 0), primero determinará si con el nuevo paso que ha entrado en una zona. En caso negativo termina la ejecución del algoritmo, pero en caso positivo y en el caso de que el tipo del agente y el de la zona sean compatibles cambia de estado. Después, guarda la información de la zona y termina el algoritmo. Por otro lado, si se encontraba delimitando la zona (activo en estado 1), primero guarda todos los puntos que conformen esa zona y que se encuentren dentro de su cono de visión. Después, analiza si ha encontrado todos los puntos que delimitan la zona en la que se encuentra, en caso negativo finaliza el algoritmo, en caso positivo, cambia a estado pasivo y termina el algoritmo.

Las dos inteligencias desarrolladas utilizan este algoritmo para el control de los agentes. La diferencia entre ellas reside en los algoritmos que rigen la comunicación y coordinación de los agentes dentro del enjambre. En la ilustración 7 se muestra el algoritmo desarrollado para la comunicación de los agentes dentro de la inteligencia de enjambre sin UAVs, es decir, la que utiliza únicamente agentes de tipo terrestres y ningún agente que se ocupe de centralizar y mejorar las comunicaciones.

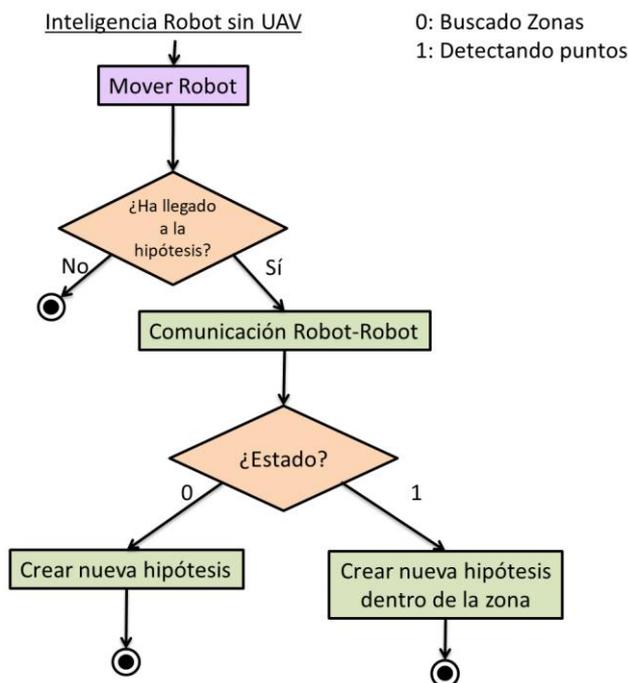


Ilustración 7: Algoritmo de comunicaciones del enjambre sin UAV

En este algoritmo, mostrado en la ilustración 7, también se ejecuta para cada agente en cada instante de tiempo y empieza ejecutando el algoritmo de la ilustración 6. Después determina si el agente ha llegado a la hipótesis que se había planteado. En caso negativo, finaliza la ejecución del algoritmo. En caso positivo, busca otro agente con el que comunicarse para transmitirse la información recogida. A continuación, independientemente de haber tenido éxito o haber fracasado en la comunicación, toma una decisión en función de su estado. Si se encontraba explorando (pasivo en estado 0), crea una nueva hipótesis, la cual le puede dirigir a una zona en la que el agente sea necesario o a un punto inexplorado. En cambio, si se encontraba delimitando la zona (activo en estado 1), crea una nueva hipótesis dentro de la zona que está delimitando, con el propósito de terminar con su tarea de exploración.

La segunda inteligencia de enjambre desarrollada, como se ha mencionado anteriormente, cuenta con el algoritmo explicado en la ilustración 6, una versión modificada del algoritmo de la ilustración 7 y un nuevo algoritmo que rige el comportamiento de los agentes voladores, o UAVs, que se encargan de centralizar la información de todo el enjambre y de optimizar las comunicaciones entre agentes terrestres. En la ilustración 8 se muestra el algoritmo modificado y en la ilustración 9 el algoritmo de los agentes voladores.

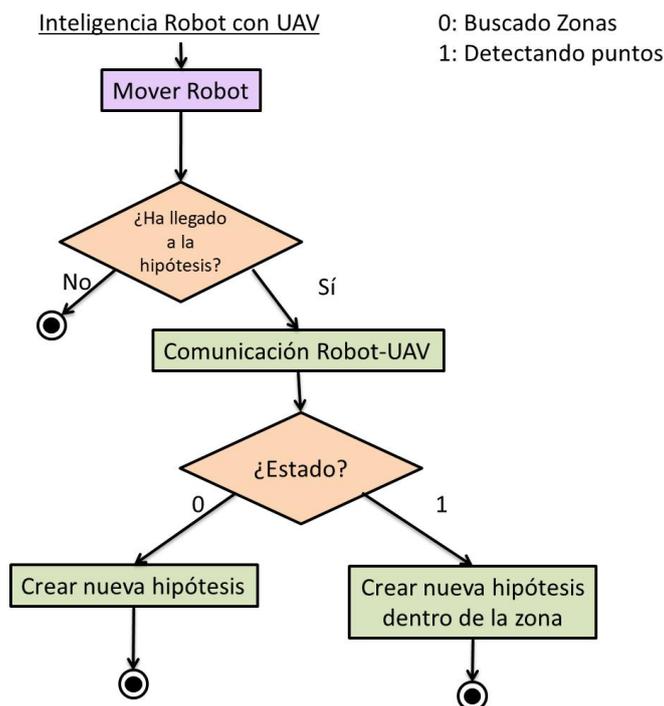


Ilustración 8: Algoritmo de comunicaciones del enjambre con UAV

Este algoritmo, mostrado en la ilustración 8, es una versión modificada del algoritmo de la ilustración 7. La mayor diferencia entre estos dos algoritmos reside en la comunicación entre los agentes. En el de la ilustración 7 la comunicación se hacía directamente desde un agente terrestre a otro agente terrestre cercano. En cambio en el de la ilustración 8 la comunicación se hace únicamente entre robot y UAV. Es decir, en caso de que un robot se encuentre fuera del alcance de un UAV no se comunicará, independientemente del número de agentes terrestres que tenga en las cercanías. Esta decisión está tomada porque los agentes voladores cuentan con toda la información centralizada que han ido recogiendo los agentes terrestres y, por lo tanto, son la mejor fuente de información para dichos agentes.

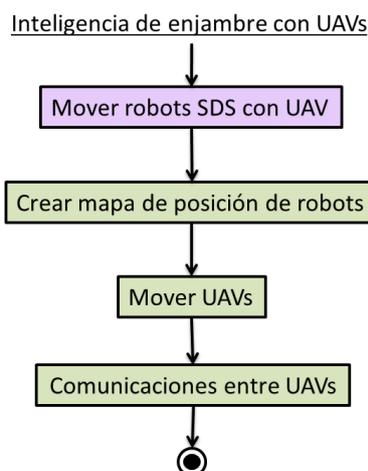


Ilustración 9: Algoritmo de comportamiento de los agentes voladores

En el algoritmo de la ilustración 9 se muestra la inteligencia de enjambre con agentes UAVs. El cual empieza realizando el algoritmo de la ilustración 8. Después crea un mapa con las posiciones de los robots terrestres, cabe destacar que al tratarse de inteligencias descentralizada, la información con la que se cuenta es parcia y que por lo tanto solo se conoce la ubicación de los robots terrestres que se encuentren dentro de las zonas de cobertura de los UAVs. A continuación los robots utilizan el mapa creado para tomar la decisión de en qué lugares se deberían colocar para maximizar la cantidad de robots a la que dan servicio. Finalmente, los agentes UAV se comunican entre sí para tener la información del mapa actualizada.

Adicionalmente a las inteligencias de enjambre basadas en el algoritmo “*Búsqueda por difusión estocástica*” (Stochastic diffusion search), también se ha desarrollado una inteligencia básica basada totalmente en decisiones aleatorias con el propósito de valer como comparativa de la eficiencia de los otros dos algoritmos. Esta inteligencia, al igual que las anteriores, también está dividida en dos algoritmos.

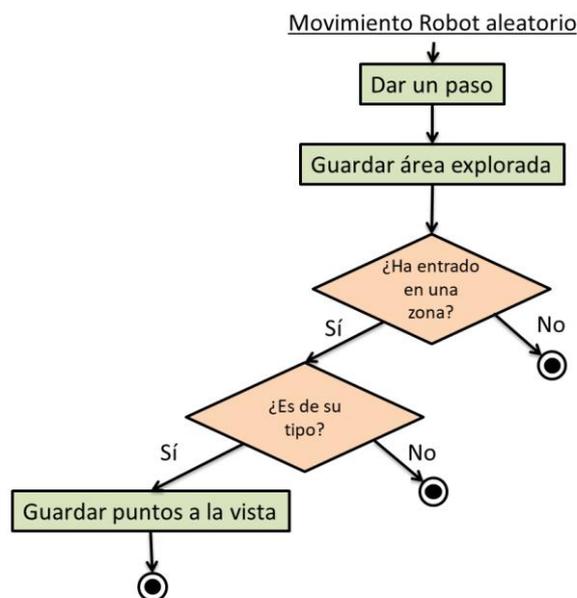


Ilustración 10: Algoritmo del movimiento de un agente con criterio aleatorio

En este algoritmo, mostrado en la ilustración 10, también se ejecuta para cada agente en cada instante de tiempo e inicia desplazándose una distancia en la dirección su hipótesis. Después, guarda en su cuadrícula de exploración la información que tiene a la vista (dentro de su cono de visión) ignorando los posibles puntos que conformen las zonas. A continuación, en el caso de que se haya introducido en una zona con la que comparte tipo registra todos los puntos de esa zona que estén en dentro de su cono de visión. Este algoritmo no introduce las mecánicas de cambio de estado porque es una funcionalidad del algoritmo “*Búsqueda por difusión estocástica*” (Stochastic diffusion search) y por lo tanto no puede ser usadas por esta inteligencia.

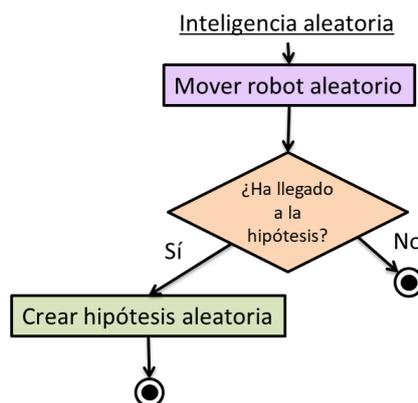


Ilustración 11: Inteligencia de enjambre con criterio aleatorio

Esta inteligencia no tiene las funciones de comunicación de las otras inteligencias, por lo tanto, este segundo algoritmo, mostrado en la ilustración 11, queda muy limitado en comparación a los equivalentes presentados para las demás inteligencias. Este algoritmo se limita a, primero, realizar el algoritmo de la ilustración 10 y, después, una vez llegado a su hipótesis crea otra hipótesis en otro punto totalmente aleatorio.

6.5. Descripción de los resultados

Los resultados principales del proyecto se han evaluado de dos formas distintas. Por un lado se ha realizado un gráfico comparativo de la eficacia de cada inteligencia a la hora de realizar una misión de exploración, comparando la cantidad de puntos que encuentran cada uno en cada instante de tiempo. Por otro lado se ha representado las cuadrículas de exploración sobre la que los agentes anotan la distribución de las zonas y los puntos del mapa, con la intención de determinar cuanta información del total contiene cada agente. Adicionalmente, se ha desarrollado un modo de visualización que permite ver en tiempo real al enjambre desplazándose por el mapa, de tal forma que se pueda analizar el comportamiento y cooperación de los agentes de una forma más directa.

I. Comparación de los algoritmos

Para poder medir los resultados se han elegido tres mapas diferentes y se ha medido la eficacia de los tres algoritmos desarrollados en cada uno de ellos. Estos tres mapas se generan antes de realizar las ejecuciones del programa y son siempre los mismos. La única diferencia entre los tres mapas son el número de zonas que aparecen en ellas (10, 15 y 20 zonas), todo lo demás, el tamaño del mapa, el tipo de las zonas etc. son siempre los mismos. En la ilustración 12 se pueden apreciar dichos mapas. Los colores de las zonas de estos mapas representan el número de tipos que tiene la zona, en azul se muestran las zonas que solo admiten un tipo de robot, en rojo los que admiten dos tipos de robots y en verde los que admiten tres tipos de robots.

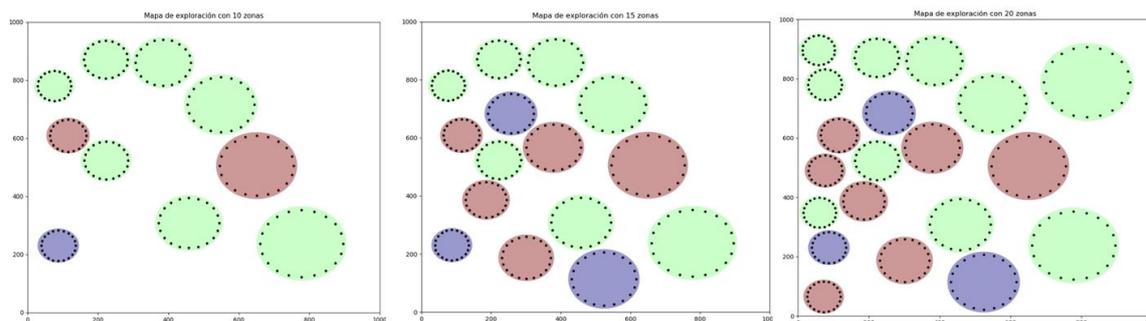


Ilustración 12: Mapas usados para medir los resultados

Después, de generar el mapa se ha ejecutado cada algoritmo veinte veces en cada mapa y se ha registrado cuantos puntos ha encontrado cada algoritmo en cada instante de tiempo. A continuación, se ha calculado el primer, segundo y tercer cuartil de los resultados de cada algoritmo en cada mapa y se ha representado en una gráfica, ver ilustración 13. La línea continua corresponde con el segundo cuartil, equivalente a la mediana, pues indica que el 50% de los datos son menores o iguales que ese valor. El área difuminada es la diferencia que separa al primer cuartil (25% de los datos menor o igual a este valor) del segundo cuartil (75% de los datos menor o igual a este valor). Se le ha aplicado a los datos de cada inteligencia distintos colores para diferenciarlos mejor. El rojo corresponde a los resultados obtenidos por la inteligencia basada en decisiones aleatorias, el verde con la primera inteligencia de enjambre desarrollada, y el azul con la segunda inteligencia desarrollada, la cual incorpora el uso de los UAVs para centralizar la información recogida.

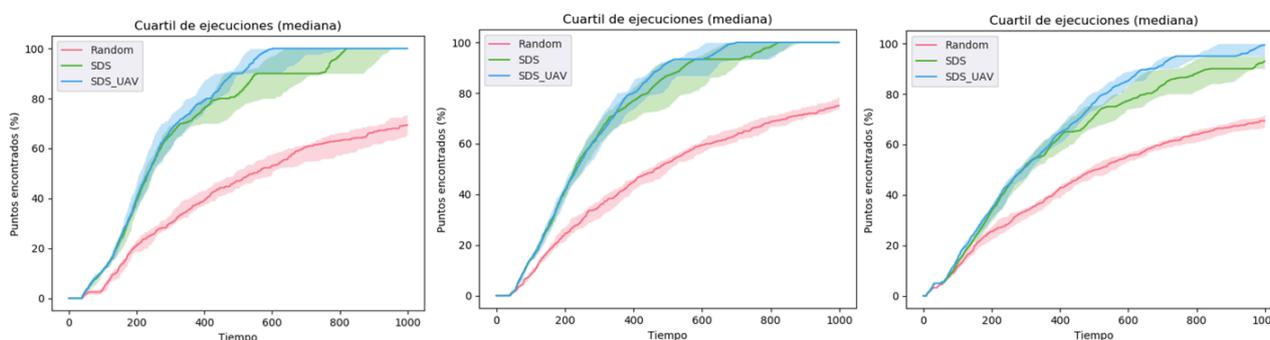


Ilustración 13: Comparación de resultados obtenidos

En la ilustración 13, figuran las gráficas generadas en los tres mapas siguiendo el proceso anteriormente explicado. Se puede observar que as dos inteligencias desarrolladas presentan una significativa mejora en el trabajo realizado comparado por la inteligencia que toma decisiones aleatorias (rojo). Esta inteligencia, va detectando puntos por pura probabilidad mientras sus agentes vagan por el mapa y actúa como referencia para saber si los resultados de las demás inteligencias son buenos o malos.

La inteligencia de enjambre sin UAV (verde) tiene la mayor dispersión entre los resultados obtenidos. Esto es debido a que se puede obtener resultados muy distintos en función de con que agente se comunique un determinado robot, ya que no todos tienen la misma cantidad de

información. Esto a la vez demuestra que la comunicación entre agentes es muy importante a la hora de optimizar la ejecución.

La inteligencia de enjambre con UAV (Azul) evita la dispersión que caracteriza a la función de la inteligencia sin UAV gracias al uso de agentes cuyo objetivo es la centralización de la información y optimización de la comunicación. De este modo, como los agentes terrestres cuentan siempre con el máximo de información descubierta por ellos mismos y pueden tomar decisiones más óptimas.

Teniendo en cuenta todo lo anterior, se puede observar que a pesar de que parezcan que los resultados de las dos inteligencias desarrolladas (azules y verdes) sean parecidos, en realidad los de la inteligencia con UAV son significativamente mejores, al contar, sobretodo, con menos dispersión de resultados entre las veinte ejecuciones. Aun así, se concluye que las dos inteligencias, dentro de sus diferencias, optimizan enormemente las tareas de exploración cumpliendo todos y cada uno de las especificaciones de este proyecto.

II. Visión individual de los agentes

Para la realización de estos resultados se ha ejecutado las tres inteligencias sobre el mapa de las 15 zonas de la ilustración 12 y se ha establecido el tiempo máximo de operación del enjambre en 900 instantes de tiempo. Después, se ha dejado ejecutando el programa durante este periodo de tiempo y se ha representado las cuadrículas de exploración de los agentes que conforman el enjambre. A la izquierda está la cuadrícula de exploración de un solo agente y a la derecha la combinación de todos las cuadrículas de exploración del enjambre.

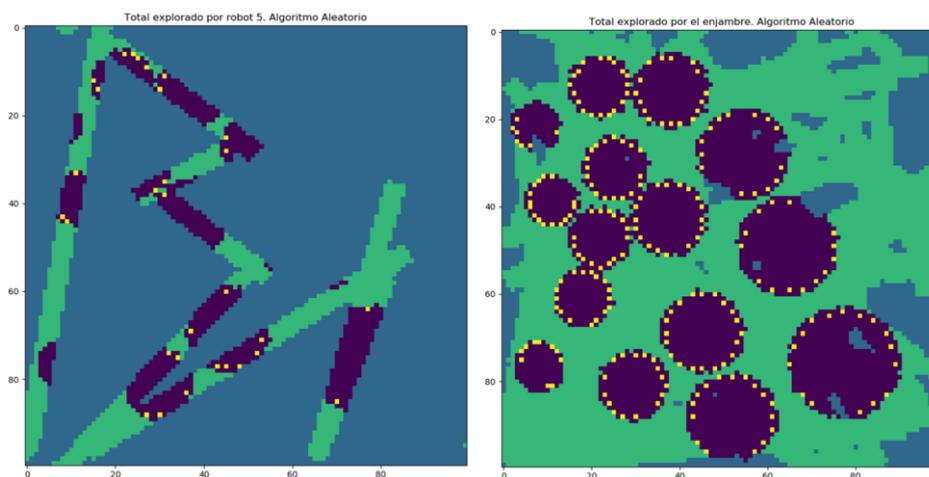


Ilustración 14: Mapa explorado por la inteligencia aleatoria

La ilustración 14 corresponde con la de la inteligencia básica controlada por decisiones aleatorias. A primera vista se puede observar que, mientras que las dos inteligencias de enjambre desarrolladas han cumplido con las tareas de exploración dentro del tiempo establecido, esta la inteligencia no ha logrado cumplir la misión. En el mapa individual del agente solo aparece el recorrido realizado por dicho agente porque esta inteligencia no contempla la comunicación entre agentes. La combinación de todos los agentes nos proporciona una aproximación al estado real del mapa. Aunque pueda parecer que la

combinación total de todos los agentes sea un resultado válido, hay que tener en cuenta que cada agente contiene gran cantidad de información que ningún otro agente dispone, y que por lo tanto, el resultado final queda gravemente comprometido si un solo agente dejara de funcionar.

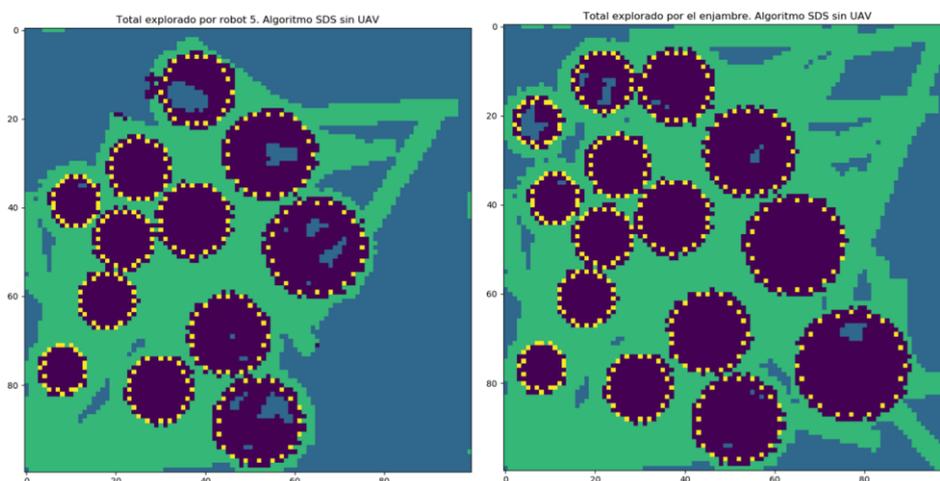


Ilustración 15: Mapa explorado por la inteligencia aleatoria SDS sin UAV

En la ilustración 15 se muestran los resultados de la primera inteligencia de enjambre realizada, la que no cuenta con agentes UAV. En cuanto al mapa del agente, podemos observar una significativa mejora con respecto a la ilustración anterior, pues se distinguen las mayorías de las zonas, las cuales se encuentran totalmente delimitadas. Vemos que la información que cuenta cada agente es mucho mayor y que por lo tanto este enjambre es más robusto ante la pérdida de información por culpa del malfuncionamiento de uno de sus agentes. La combinación de todos los agentes muestra todas las zonas del mapa y todos los puntos que las delimitan.

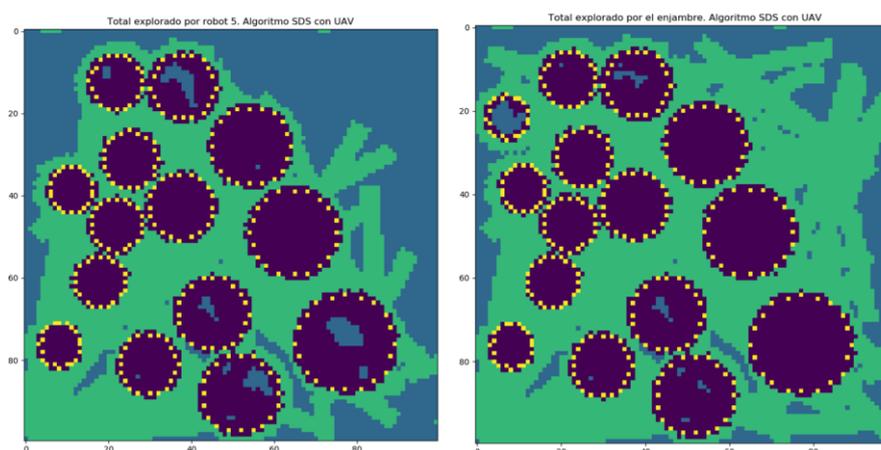


Ilustración 16: Mapa explorado por la inteligencia SDS con UAVS

En la ilustración 16 se representan los resultados de la segunda inteligencia de enjambre desarrollada, la que cuenta con los agentes voladores para centralizar la información. Podemos observar que el agente dispone de prácticamente de la misma información que dispone la totalidad del enjambre. Lo que significa que recuperando tan solo un agente del enjambre ya se dispondría de una aproximación bastante realista de la disposición de las zonas por el mapa. Al mismo tiempo, al tener más región explorada y almacenada, los algoritmos para determinar nuevas hipótesis son mucho más eficaces. En estos resultados se puede observar las ventajas que introduce el uso de agentes especializados única y exclusivamente en las comunicaciones entre agentes dentro del enjambre.

III. Representación del movimiento del enjambre

A parte de los dos métodos de medición anteriores, también se ha desarrollado un modo de visualización que permite seguir el movimiento del enjambre por el mapa en tiempo real. Este modo de visualización es muy útil para analizar el comportamiento del enjambre y ver las interacciones entre todos los agentes del enjambre.

Para este modo de visualización se ha usado el mismo método de representación del mapa a explorar. Es decir, los colores de las zonas del mapa representa el número de tipos que tiene la zona, en azul se muestran las zonas que solo admiten un tipo de robot, en rojo los que admiten dos tipos de robots y en verde los que admiten tres tipos de robots. Por lo tanto, las zonas verdes podrán ser delimitadas por un gran número de agentes mientras que las azules solo podrán ser delimitadas por un número más escaso de agentes.

A continuación, en la ilustración 17, se muestran tres etapas de cómo se desarrolla este modo de visualización. Las cruces azules representan a los agentes terrestres, los cuales realizan las tareas de exploración. Los triángulos magentas representan a los agentes voladores encargados de mantener una comunicación eficiente permaneciendo alrededor de los agentes terrestres. Por último, las flechas marrones muestran el movimiento de los agentes terrestres a través del mapa.

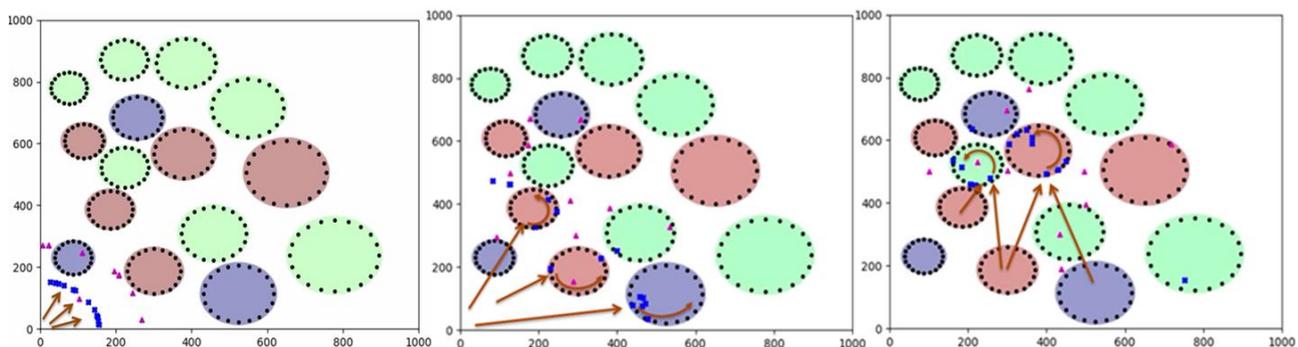


Ilustración 17: Etapas de movimiento

Como se puede apreciar en el mapa de la izquierda, todos los agentes, voladores y terrestres, empiezan la ejecución del programa en la posición (0,0). Después, se dispersan aleatoriamente por el mapa, pues aún no tienen ninguna información en la que basarse para tener un criterio a la hora de establecer hipótesis. En la imagen del medio, los agentes han encontrado una zona con la que son compatibles y proceden a moverse por sus extremos con el fin de

encontrar todos los puntos que la delimitan. Antes de llegar a la tercera etapa, varios agentes han realizado una fase de difusión y comunicación en la que han compartido las cuadrículas de exploración. Por lo tanto, al terminar las tareas de la segunda etapa, se han encaminado directamente hacia otra zona inexplorada como se muestra en la tercera imagen.

7. PLANIFICACIÓN

7.1. Diagrama de Gantt y clico de vida

Con el objetivo de poder realizar el proyecto de una manera ordenada se ha desarrollado un diagrama GANT que ayude a la planificación del proyecto. Estos tipos de diagramas son muy útiles para organizar una planificación clara que ayude a los desarrolladores a cumplir sus objetivos dentro del plazo acordado.

En el caso específico de este proyecto se ha desarrollado un diagrama de 8 meses de duración, desde Octubre de 2018 hasta Junio de 2019. Sin embargo, hay un parón de un mes de duración entre los meses de Diciembre y Enero correspondientes a las fechas de exámenes de la convocatoria ordinaria del primer cuatrimestre.

El proyecto está planteado para poder ser desarrollado en cuatro tareas:

I. Estudio inicial y formación

En esta primera tarea se centra en cimentar unas bases solidadas sobre las que construir el resto del proyecto. Para ello se empieza aprendiendo a utilizar el lenguaje de programación “Python” y obteniendo información sobre el funcionamiento y las últimas novedades del algoritmo “*Búsqueda por difusión estocástica*” (Stochastic diffusion search). Después se pasa a la fase de definición y especificación de los objetivos y alcance del proyecto. Mientras tanto se desarrolla el primer concepto del proyecto, el cual sirve como prototipo. El tiempo estipulado para esta tarea es desde el nacimiento del proyecto el 24 de Octubre de 2018 hasta el 12 de Diciembre de 2018.

IV. Desarrollo especializaciones sobre SDS

Esta segunda tarea de desarrollo de las especializaciones sirve para para implementar las adaptaciones al algoritmo general. De este modo, se logra hacerlo mucho más eficaz resolviendo el problema definido en la primera fase. Las principales adaptaciones que hay que realizar al algoritmo tradicional son la definición del tipo de zonas y el sistema utilizado para crear nuevas hipótesis cada vez que un agente haya completado la fase de difusión. Una vez terminadas las modificaciones hay que validar que el resultado sea óptimo y que consiga mejoras significativas con respecto al movimiento de agentes aleatorio sin ningún criterio. Para ello será necesario desarrollar una primera versión del programa que se encargará de realizar las simulaciones comparativas del enjambre robótico. Esta tarea se realiza en el periodo estipulado, desde el 21 de Enero hasta el 30 de Abril.

V. Desarrollo propuesta final

Una vez realizadas todas las tareas anteriores se habrá obtenido una versión robusta y valida del algoritmo de exploración planteado. Aun así, este algoritmo no es la versión definitiva del proyecto, ya que se planea llevar el algoritmo más allá con el uso de agentes voladores o UAV. La primera fase de esta tarea se dedica a implementar el sistema de estos agentes que funcionan como si fueran centralizadores de información. Después, se pasa a la fase de comprobación en el que se prueba que todas las funcionalidades funcionan como deben funcionar y que se ha cumplido todas las especificaciones y objetivos. Mientras tanto, se realiza la versión final del programa de comparación de algoritmos, el cual ahora compara el

movimiento aleatorio el movimiento SDS sin UAV y el nuevo algoritmo SDS con UAVs. Esta tarea se sitúa la última del proceso de desarrollo y va desde el 1 de Mayo hasta el 17 de Junio.

VI. Gestión y documentación del proyecto

Esta última tarea se realizará en el transcurso de todo el proyecto. Durante todo este periodo de desarrollo se irá registrando el avance diario realizado en el proyecto. También se irá anotando todo el proceso de diseño y guardando cada una de las versiones del proyecto. Con todos estos datos y con el análisis de los resultados obtenidos con el programa de comparación se redactará este informe final. La duración de esta tarea, como se ha mencionado anteriormente, se iguala con la duración final del proyecto, es decir, 8 meses.

A continuación se muestra el GANT propuesto para este proyecto, el cual agrupa las tareas anteriormente citadas:

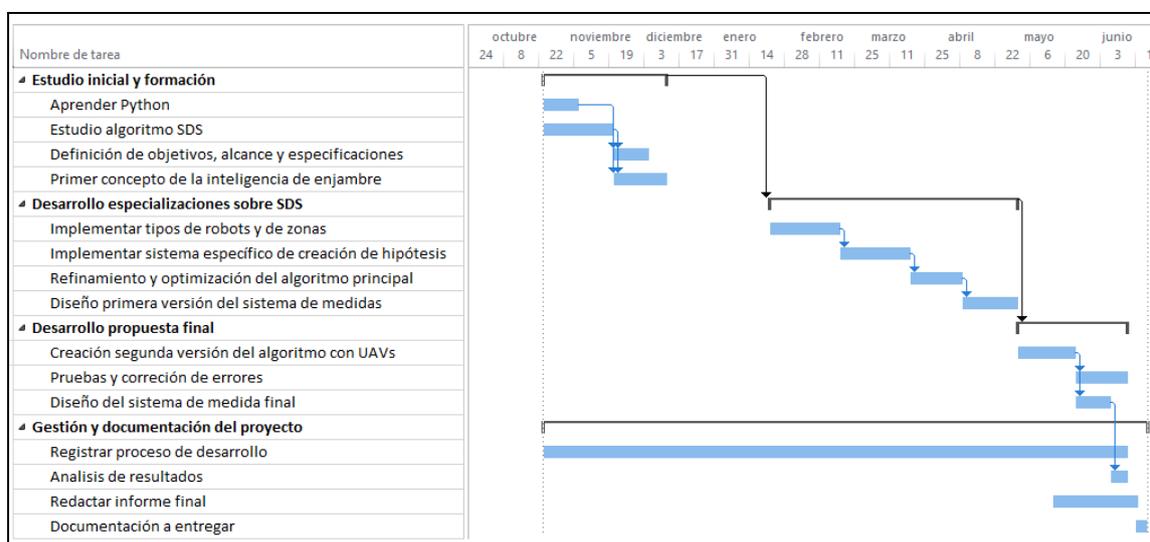


Ilustración 18: Diagrama de planificación GANT

7.2. Metodología de desarrollo

A parte del diagrama de planificación GANT presentados en la sección anterior se han utilizado otras herramientas que ayudan al desarrollo de proyectos, especialmente los centrados en el desarrollo software. Estas herramientas son las metodologías ágiles que tratan de incentivar el ciclo de vida del proyecto y un sistema de control de versiones que permita implementar mejoras de una forma eficaz.

Por un lado, las últimas corrientes del desarrollo software no se puede concebir sin ningún tipo de metodología ágil que haga que el software producido evolucione. Estas metodologías parten del hecho que el software es un producto que nunca está terminado. Cuando se han cumplido todas las especificaciones y el cliente obtiene el software, aparecen nuevos requerimientos o especificaciones que hacen posible el desarrollo de una nueva versión. Cuando esta segunda versión esta nuevamente terminada aparecerán nuevas especificaciones que harán otra vez necesaria otra tercera versión. Este proceso se repite reiteradamente, haciendo del software un elemento vivo con cada vez con más capacidades o funcionalidades y

con la habilidad de solucionar problemas más complejos o de diferente índole. Es por este hecho que se suele decir que las especificaciones siempre cambian.

Al principio la forma más común del desarrollo de software era mediante un proceso secuencial parecido al de una cadena de montaje donde el desarrollo tiene inicio y final. Pero poco a poco se han ido desarrollando nuevas técnicas cada vez más eficaces que han dotado al desarrollo software una realimentación. En otras palabras, se ha cambiado el enfoque que se le daba al desarrollo para considerar al software como algo vivo que necesita evolucionar. Estas técnicas o tendencias son el modelo de desarrollo en V, el modelo incremental o el modelo en espiral, entre otros muchos modelos.

Por otro lado, un sistema de control de versiones permite el desarrollo más organizado del proyecto. Un sistema de control de versiones es un sistema que registra las alteraciones realizadas a un documento del proyecto. De esta forma se hace sencillo navegar entre versiones de código sin perderse. También permiten la recuperación de código antiguo que se había descartado pero que ha resultado ser de utilidad para las versiones nuevas. Adicionalmente, si se utilizan sistemas de control de versiones distribuidos o en línea se puede hacer muy sencilla la cooperación entre miembros de un mismo equipo de desarrollo, al tener todos los miembros acceso al código más actual.

7.3. Desarrollo software

Para realizar este proyecto se ha optado por una metodología ágil basado en un modelo circular, en el que con cada vuelta se van introduciendo nuevas especificaciones. De tal modo que poco a poco se ha obtenido un producto cada vez más complejo e inteligente capaz de realizar su trabajo de manera cada vez más óptima.

Al principio del proyecto se ha diseñado una primera inteligencia de enjambre con funcionalidades muy limitadas. Aun así, la inteligencia de este primer enjambre era superior a las de un enjambre con inteligencia basada totalmente en decisiones aleatorias. Este primer enjambre ya incorporaba las herramientas básicas necesarias para ejecutar el algoritmo “*Búsqueda por difusión estocástica*” (Stochastic diffusion search), aun así no incorporaba ninguna modificación que lo hiciera óptimo para presentarse al problema que se le estaba planteando.

Después, se creó una nueva versión que cumpliera todas las especificaciones propuestas para el problema inicial. Es decir, en esta etapa de desarrollo se hizo necesario modificar las funcionalidades del algoritmo básico para adaptarlas a las del proyecto. Las modificaciones más significativas son, por un lado, la introducción de diferentes tipos de zonas y agentes, ya que la primera versión no consideraba los diferentes tipos, y por otro lado, la implementación de los criterios en los que se basan los agentes para realizar las nuevas hipótesis.

Por último, una vez esta segunda versión estaba operativa, la mejora con la anterior era significativa pero como el proceso podría ser aún más óptimo. Por lo tanto se pasó a la siguiente etapa del proyecto en la que se desarrollaron un método de comunicación con agentes voladores los cuales en vez de dedicarse a explorar el terreno se ocuparían de comunicarse con el máximo de robots. De esta forma hacíamos que todos los agentes del

enjambre tuvieran información mucho más actualizada, lo que mejoraba el tiempo necesario para explorar el mapa y delimitar las zonas a explorar.

8. PRESUPUESTOS Y COSTES

Se ha planteado el presupuesto desde dos perspectivas diferentes. La primera se realiza desde el punto de vista actual del proyecto, en el cual un ingeniero de telecomunicaciones realiza todo el trabajo. La segunda, es desde un punto de vista más corporativo, en el que una empresa decide conformar un grupo de unas pocas personas para realizar el trabajo planteado y necesita un presupuesto.

8.1. Proyecto de ingeniería

A continuación se muestran un desglose de los gastos para la realización del proyecto como un trabajo de fin de grado (TFG). La duración del trabajo ha sido desde el mes de Octubre hasta el mes de Junio, es decir, 8 meses. El tiempo calculado invertido en este proyecto es el equivalente al número de créditos para la realización de un trabajo de fin de grado (TFG), actualmente estimado en 300 horas.

8.1.1 Horas Internas

Trabajador	Nº	Coste Horario	Horas	Coste
Ingeniero de Telecomunicación	1	30,00 €	300	9.000,00 €
Total				9.000,00 €

Tabla 1. Costes: Horas Internas

8.1.2. Amortizaciones

Activo	Coste de Adquisición	Vida útil (meses)	Tiempo de uso en el proyecto (meses)	Amortización
Portátil Lenovo ideapad 500 (i7 8GB 1TB)	900,00 €	48	8	150,00 €
PC 2018 (i6 16GB 3TB)	1.200,00 €	48	8	200,00 €
Total	2.100,00 €			350,00 €

Tabla 2. Costes: Amortizaciones

8.1.3 Gastos

Concepto	Coste
Libro Python	15,00 €
Total	15,00 €

Tabla 3. Costes: Gastos

8.1.4 Coste Total

Concepto	Coste Total
Horas internas	9.000,00 €
Amortizaciones	350,00 €
Gastos	15,00 €
Subcontrataciones	- €
Subtotal	9.365,00 €
Imprevistos (5%)	468,25 €
TOTAL	9.833,25 €

Tabla 4. Costes: Total

8.2. Proyecto de implementación

El siguiente presupuesto es el que se realizaría en una organización para saber cuánto cuesta desarrollar el proyecto. Por lo tanto, al tratarse de un desarrollo más comercial, se plantea la conformación de un equipo de trabajo, la adquisición de equipos más avanzados y diversos elementos que en el escenario anterior no se habrían planteado. Se ha estimado que la duración del proyecto serán dos años, en los cuales los distintos miembros del equipo dedicarán un número diferente de horas. Estos miembros son cuatro: primero, el director del proyecto, el cual se encarga de dirigir y organizar al equipo y cuenta con toda la responsabilidad del proyecto. Segundo, un experto en programación que sepa asesorar al resto del equipo y que ayude a traducir los diseños en implementaciones reales. Tercero, un analista que proponga mejoras y solucione problemas de la arquitectura y diseño desarrollado. Por último, también será necesario un analista de datos que sepa interpretar los resultados obtenidos en cada etapa del proyecto y detecte posibles errores de integración, de interconexión o de validación.

8.2.1 Horas Internas

Trabajador	Nº	Coste Horario	Horas	Coste
Director de Proyecto	1	65,00 €	600	39.000,00 €
Experto en programación	1	60,00 €	1000	60.000,00 €
Analista de Arquitectura	1	55,00 €	500	27.500,00 €
Analista de Datos	1	60,00 €	900	54.000,00 €
Total				180.500,00 €

Tabla 5. Presupuesto: Horas Internas

8.2.2. Amortizaciones

Activo	Coste de Adquisición	Vida útil (meses)	Tiempo de uso en el proyecto (meses)	Amortización
Equipo para clúster y operación (5 ordenadores)	3.000,00 €	48	24	1.500,00 €
Total	3.000,00 €			1.500,00 €

Tabla 6. Presupuesto: Amortizaciones

8.2.3 Gastos

Concepto	Coste
Material de oficina	1.000,00 €
Alquiler oficina	25.000,00 €
Alquiler de servidor (Amazon webServices 100€/mes)	2.400,00 €
Total	28.400,00 €

Tabla 7. Presupuesto: Gastos

8.2.4 Presupuesto Total

Concepto	Coste Total
Horas internas	180.500,00 €
Amortizaciones	1.500,00 €
Gastos	28.400,00 €
Subcontrataciones	- €
Subtotal	210.400,00 €
Imprevistos (5%)	10.520,00 €
TOTAL	220.920,00 €

Tabla 8. Presupuesto: Total

9. CONCLUSIONES

Se considera que se ha realizado una inteligencia de enjambre robusta capaz de solucionar todos los problemas que se le han presentado de una forma eficaz y óptima. También se considera que este proyecto cumple todos y cada uno de los objetivos, tanto principales como secundarios, que se le habían planteado. Debido a que se ha logrado mejorar el algoritmo “*Búsqueda por difusión estocástica*” (Stochastic diffusion search) con las adaptaciones diseñadas para realizar las tareas de exploración delimitando las zonas que se encuentran esparcidas por el mapa.

Además, se han presentado dos soluciones que cumplen ambas con las especificaciones planteadas. Por un lado, la implementación real de la primera inteligencia de enjambre, la que no cuenta con UAV, podría suponer una solución válida cuando por impedimento del terreno no sea posible la utilización de agentes voladores o cuando se limite el desarrollo del hardware de los agentes a producir un único modelo de agente. Por otro lado, la implementación real de la segunda inteligencia desarrollada, supone la mejor opción en la mayoría de los casos. Pues en la que más rápido concluye las tareas de exploración y la que menos variabilidad tiene entre diferentes ejecuciones.

En cuanto al futuro del proyecto, se considera que las dos inteligencias disponen de una gran escalabilidad, lo que hace que sea posible la introducción de posibles mejoras o funcionalidades nuevas. En especial, se plantean tres posibles grandes funcionalidades a las que el proyecto se podría expandir. La primera de ellas es la introducción de zonas con distinta forma y volumen, para que de esta forma poder analizar terrenos más heterogéneos. La segunda es la implementación de mecánicas de colisión, tanto con el entorno como entre agentes. En la tercera, se propone introducir un tercer eje al mapa de tal forma que los espacios a explorar sean de tres dimensiones.

Por último, en el proceso seguido para el desarrollo de este proyecto se ha realizado una labor de investigación que ha servido para aprender nuevas técnicas de programación y nuevos algoritmos para diferentes entornos. También ha servido para aprender a realizar búsquedas de riguroso contenido académico y a aprender el estado de un sector con el que no se había trabajado anteriormente. En general, se considera muy beneficiosa la realización de este proyecto.

10. **BIBLIOGRAFIA**

- [1] Beni, G., Wang, J. *“Swarm Intelligence in Cellular Robotic Systems”*. University of California, Riverside. (1993)
- [2] San Nah Sze and Wei King Tiong. *“A Comparison between Heuristic and Meta-Heuristic Methods for Solving the Multiple Traveling Salesman Problem”*. World Academy of Science, Engineering and Technology.(2007).
- [3] Christian Blum and Andrea Roli *“Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison”*. Université Libre de Bruxelles and Università degli Studi di Bologna (2003)
- [4] Mohammad Majid al-Rifaie and John Mark Bishop. *“Stochastic Diffusion Search Review”*. Goldsmiths College, University of London (1989)
- [5] Mohammad Majid al-Rifaie and John Mark Bishop. *“The mining game: a brief introduction to the stochastic diffusion search metaheuristic”*. Department of Computing, Goldsmiths College, University of London. (2010)
- [6] Mohammad Majid al-Rifaie, Ahmed Aber and Remigijus Rasisys. *“Swarming Robots and Possible Medical Applications”*. International Society for the Electronic Arts (ISEA 2011), Istanbul, Turkey. (2011)
- [7] Fatimah Majid al-Rifaie and Mohammad Majid al-Rifaie. *“Maximising Overlap Score in DNA Sequence Assembly Problem by Stochastic Diffusion Search”*. Part of the Studies in Computational Intelligence book series (SCI, volume 650) (2005)
- [8] J. M. Bishop and P. Torr. *“The Stochastic Search Network”*. Part of the BT Telecommunications Series book series (BTTS, volume 1) (1999)
- [9] Del Ser, Javier, et al. "Bio-inspired computation: where we stand and what's next." *Swarm and Evolutionary Computation* 48 (2019): 220-250