



Inversión e inteligencia artificial: algoritmos enjambre y optimización de carteras

GUILLERMO MELÉNDEZ

RESPONSABLE DEL LABORATORIO DE INTELIGENCIA ARTIFICIAL DE BME

Aplicación de algoritmos enjambre a la exploración y optimización de entornos financieros dinámicos

Artificial Intelligence

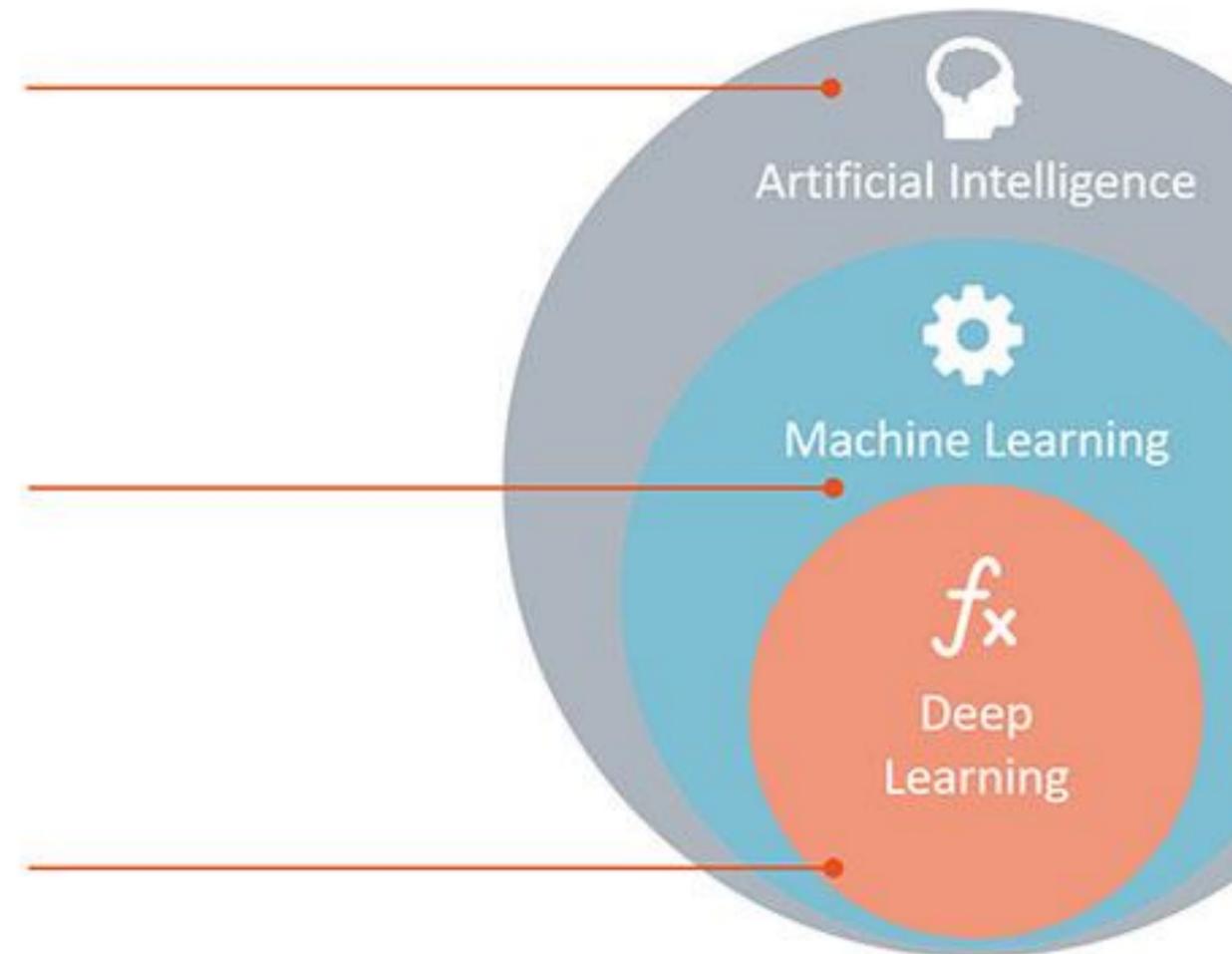
Any technique which enables computers to mimic human behavior.

Machine Learning

Subset of AI techniques which use statistical methods to enable machines to improve with experiences.

Deep Learning

Subset of ML which make the computation of multi-layer neural networks feasible.



Ramas de conocimiento en Inteligencia Artificial



Redes neuronales



Algoritmos genéticos



Algoritmos enjambre



Lógica difusa

➤ Sistemas multi-agente / algoritmos enjambre



Un sistema multiagente (SMA) es un sistema compuesto por múltiples agentes que interactúan entre ellos. Los agentes no tienen porqué ser necesariamente inteligentes, a nivel individual.

Hay dos enfoques para construir sistemas multiagentes:

El enfoque formal o clásico, que consiste en dotar de los agentes de la mayor inteligencia posible, utilizando descripciones formales del problema que resolver.

El enfoque constructivista, que persigue la idea de brindar inteligencia al conjunto de todos los agentes, para que a través de mecanismos de interacción, el sistema mismo genere comportamiento inteligente que no necesariamente estaba definido dentro de los agentes mismos (que pueden ser realmente simples). Este tipo de conducta es habitualmente llamado comportamiento emergente.

➤ Ventajas

Solucionan problemas complejos en un periodo de tiempo aceptable, siendo un sistema muy robusto a mínimos locales. Al ser los agentes, elementos independientes, es muy sencillo paralelizar su comportamiento.

➤ Desventajas

No garantizan una solución óptima.

➤ Tipos de Sistemas multi-agente / algoritmos enjambre

Existen multitud de sistemas multi-agente:



Optimización de la colonia de hormigas: algoritmos de optimización inspirada en las acciones de una colonia de hormigas. Método muy útil en problemas que necesitan encontrar caminos hacia metas. El objetivo es localizar soluciones óptimas moviéndose a través de un espacio de parámetros que representa todas las posibles soluciones. Las hormigas naturales utilizan las feromonas para orientarse entre ellas, señalizando los recursos y explorando su entorno.



Algoritmo de manada

<https://www.youtube.com/watch?v=ID-0D56x30k>



Algoritmo de construcción de las termitas

<https://www.youtube.com/watch?v=xcTh66QwQZE>



En los últimos años han surgido multitud de algoritmos enjambre

https://es.wikipedia.org/wiki/Inteligencia_de_enjambre

- Optimización enjambre de partículas
- Optimización de múltiples enjambres
- Colonia de abejas
- Sistemas inmunológicos artificiales
- Búsqueda de sistema cargado
- Búsqueda gravitacional
- Caída inteligente de gotas de agua
- Optimización magnética

➤ Elementos de los Sistemas multi-agente



¿Qué elementos pensáis que tiene un sistema multi-agente?



¿Qué tendremos que tener en cuenta para programar?

➤ Elementos de los Sistemas multi-agente



Entorno

Es el problema que queremos resolver.

Este es uno de los puntos más importantes del proceso, y por norma general no se le da la importancia necesaria. Cuando se estudian este tipo de algoritmos, se suele dar el entorno hecho para que el alumno se centre en el algoritmo. Pero sin diseñar el entorno, es imposible que se comprenda bien el algoritmo. Volveremos a esto en breve.

Puede evolucionar con el paso del tiempo, ser modificable (o no), tener objetos (o no).



Objetos

Fuentes de alimento, bloques de construcción, ciudades que visitar, obstáculos que salvar... puede ser transportables (o no), temporales o permanentes. Las “BadZones” deben tener las características de alcance (radio) y tiempo restante de vida.



Agentes

Hay que definir el comportamiento de los agentes, las relaciones que tendrán entre sí (jerárquica o igual) y la comunicación entre ellos. Cada agente poseerá un conjunto de operaciones sobre los objetos (transportarlos, usarlos, señalarlos) y sobre los demás agentes (intercambiar objetos, información).



Percepción del mundo

Pueden tener una percepción global del entorno (ver el tablero completo, mapa completo, laberinto desde arriba) o una percepción localizada a su alrededor (como cuando estás dentro de un laberinto, solo puedes ver lo que tienes inmediatamente a tu alrededor). Distintos agentes podrían tener percepciones distintas del entorno.

➤ Elementos de los Sistemas multi-agente



Toma de decisiones

Debemos determinar como reaccionará un individuo en función de sus percepciones y unas reglas a aplicar (sistema experto, lógica difusa, redes neuronales). Con unas pocas reglas simples es posible resolver problemas muy complejos (las redes neuronales se basan en regresiones). Se pueden agregar aspectos estocásticos (aleatoriedad) para permitir que el sistema sea más fluido y flexible.



Cooperación y comunicación

Debemos determinar cómo van a comunicarse los individuos. Podría ser con comunicación directa (dos individuos que se encuentran y utilizan un lenguaje) o comunicación asíncrona (dejando trazas en el entorno como las hormigas).



Capacidad del agente

Pueden tener todas las mismas capacidades o estar organizados en castas (donde cada uno tiene una tarea / capacidad distinta). Es posible que los agentes aprendan con el paso del tiempo o que tengan conocimientos fijos. Si se agrega aprendizaje hay que determinar al algoritmo de aprendizaje subyacente (redes neuronales, algoritmo genético...).

Como véis, hay muchas cosas que determinar antes de ponernos a programar.

➤ Usos de los Sistemas multi-agente



¿Qué usos podría tener un sistema multi-agente?



¿Cuándo deberíamos usarlos?

➤ Usos de los Sistemas multi-agente



¿Qué usos podría tener un sistema multi-agente?

Simulación de multitudes, análisis de reacciones en caso de evacuación, descubrir zonas de posibles aglomeraciones, planificación de tráfico, simulación de tropas, optimización de flotas de vehículos, organización de fábricas (búsqueda de rutas adaptándose a la circulación).

Pero también comprensión de fenómenos complejos, como el crecimiento de poblaciones de bacterias, control de vehículos no tripulados. La Nasa lo está estudiando para el autoensamblaje orbital, cartografía planetaria, detección y eliminación de tumores dentro del cuerpo, a través de nanorobots.

O para la minería. Ya sea de datos, o minería real.

Como véis, tiene multitud de aplicaciones. Pero daros cuenta de que no he dicho finanzas, ni bolsa.



¿Cuándo deberíamos usarlos?

Al igual que con los algoritmos genéticos, si contamos con un sistema determinista que nos ofrezca una solución en un tiempo aceptable, dicho sistema será preferible siempre a una solución multi agente.

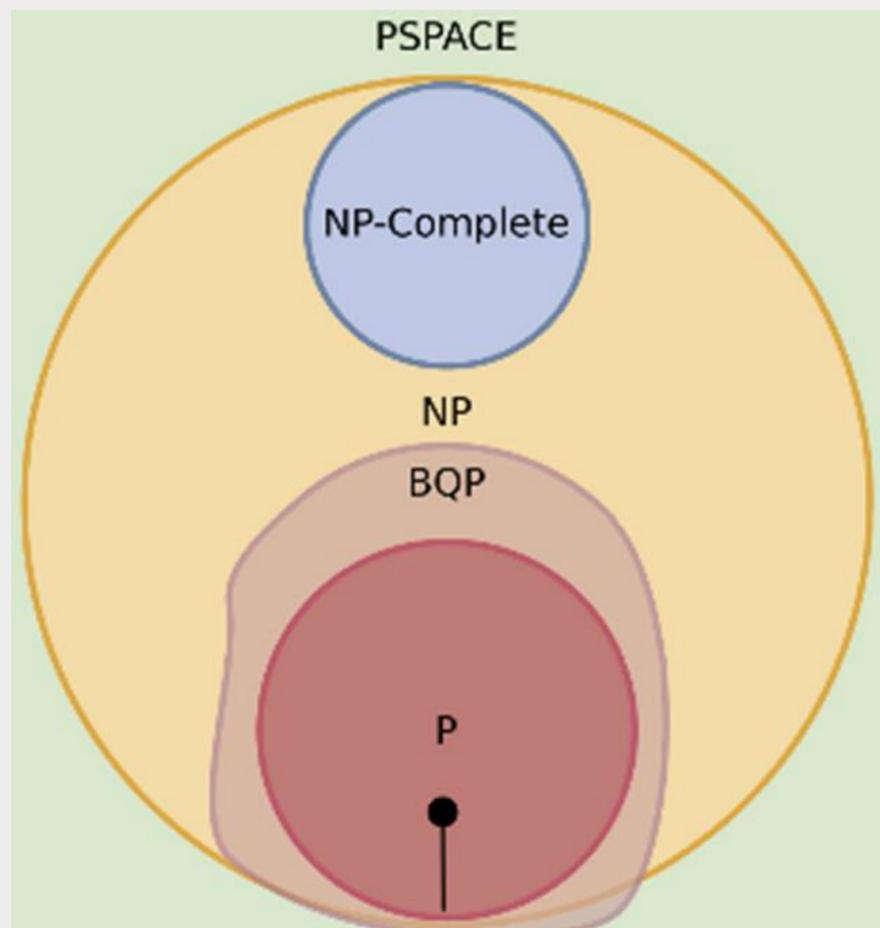
Si por el contrario no contamos con una solución determinista. O esta necesita un tiempo de cálculo no asumible, los algoritmos enjambre podrían ser la mejor solución posible.

¿Cómo optimizamos una cartera de inversión si trabajamos con un universo de miles de fondos de inversión 20.000 – 100.000 fondos?

Uno de los procedimientos estándar consiste en la reducción del universo a través de la clusterización de activos similares: HierarchicalRiskParity. Una solución perfectamente válida.

Los sistemas multi-agente podrían ser una alternativa

➤ ¿Cuándo deberíamos usar sistemas multi-agente?



PSPACE es el conjunto de los problemas de decisión que pueden ser resueltos por una máquina de Turing determinista en espacio de polinomios y tiempo limitados: ajedrez $n \times n$, GO $n \times n$

La clase P son problemas que pueden resolverse en tiempo polinómico con la capacidad de cómputo actual en un ordenador clásico: conectividad de grafos, búsqueda de coincidencias.

BQP (bounded-error quantum polynomial time) son problemas que se pueden resolver de manera cuántica, con una probabilidad de error aceptable: factorización.

Los problemas NP (no deterministas en tiempo polinómico, isomorfismo de grafos) y NP completos (problema del viajante) están fuera de P y de BQP

➤ Algoritmo de la colonia de hormigas (ACO)



¿Cómo funciona?

➤ Funcionamiento del Algoritmo ACO

➤ Las hormigas se comunican mediante feromonas depositadas en el suelo. Las exploradoras se desplazan aleatoriamente por el entorno, cuando encuentran comida, vuelven al hormiguero por el mismo camino dejando un rastro de feromonas (las hormigas recuerdan perfectamente cada paso que han realizado).

Las demás exploradoras que encuentran la feromona tienen tendencia a seguirla (la probabilidad depende de la cantidad de feromona depositada).

Las feromonas se evaporan de manera natural.

Siguiendo este sistema, las hormigas son capaces de determinar el camino más corto entre la colonia y el alimento.

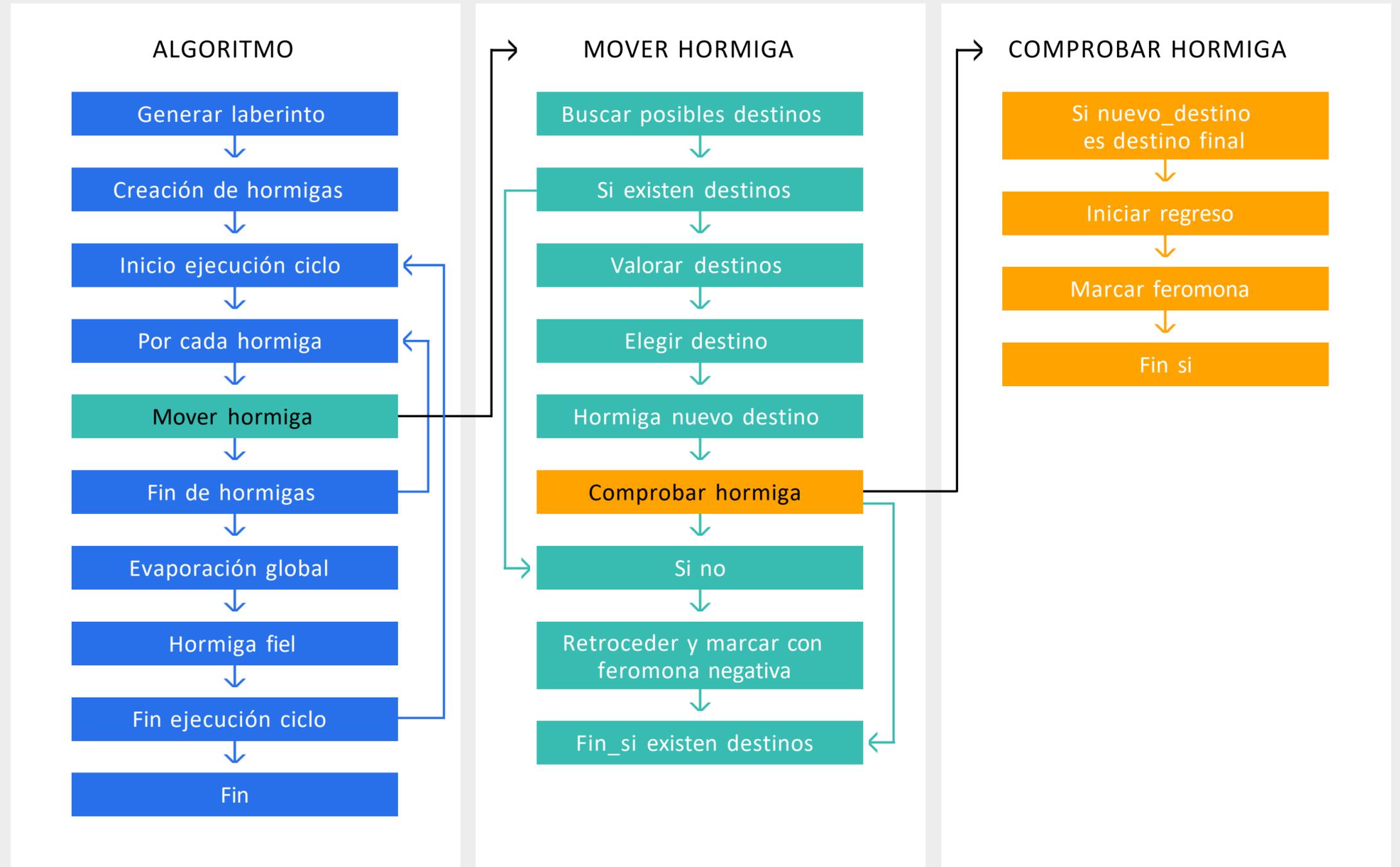
¿Podrías esquematizar este algoritmo?

Más que nada, porque el objetivo de hoy es programarlo juntos.



Funcionamiento del Algoritmo ACO

¿Podrías esquematizar este algoritmo?

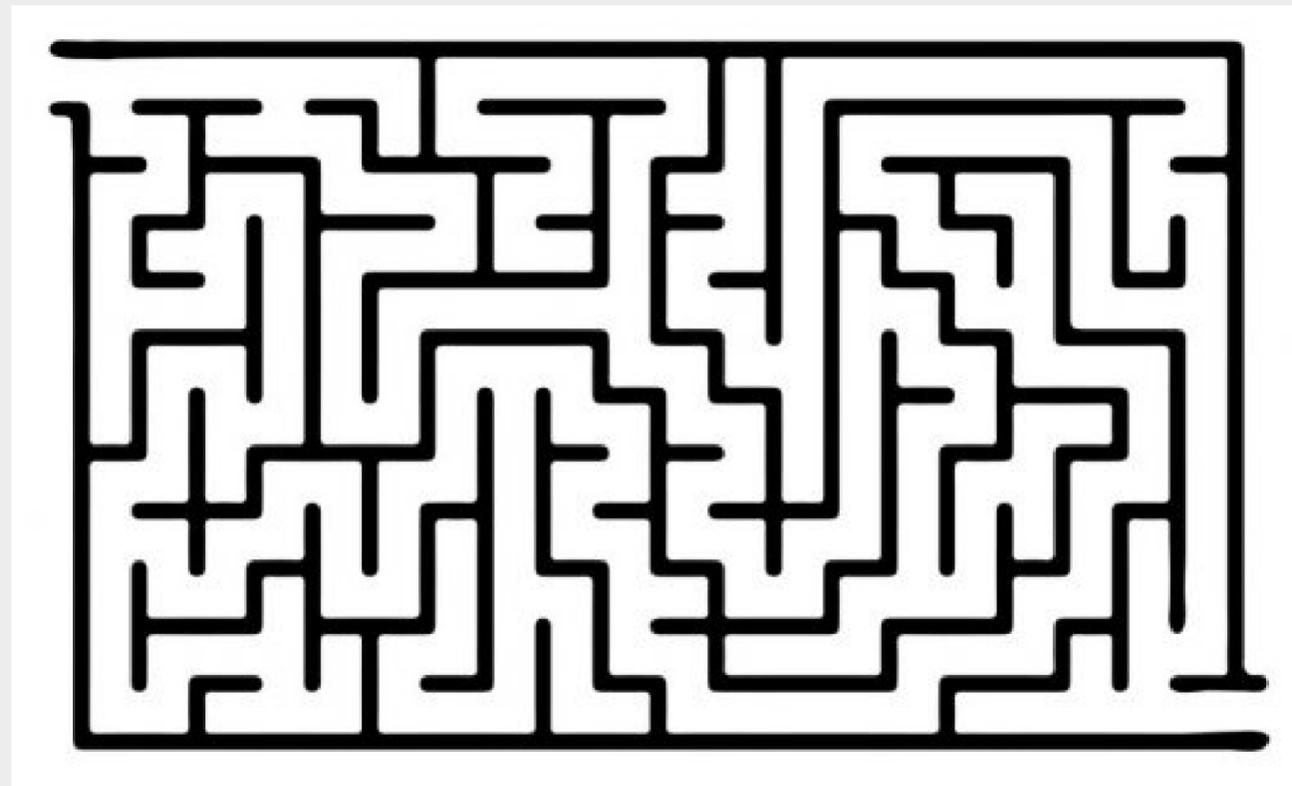


➤ Generación del entorno

- Se busca implementar y comprobar la eficiencia del algoritmo en entornos complejos y, si fuera deseable, cambiantes. Para ello se ha programado un generador de laberintos aleatorios.

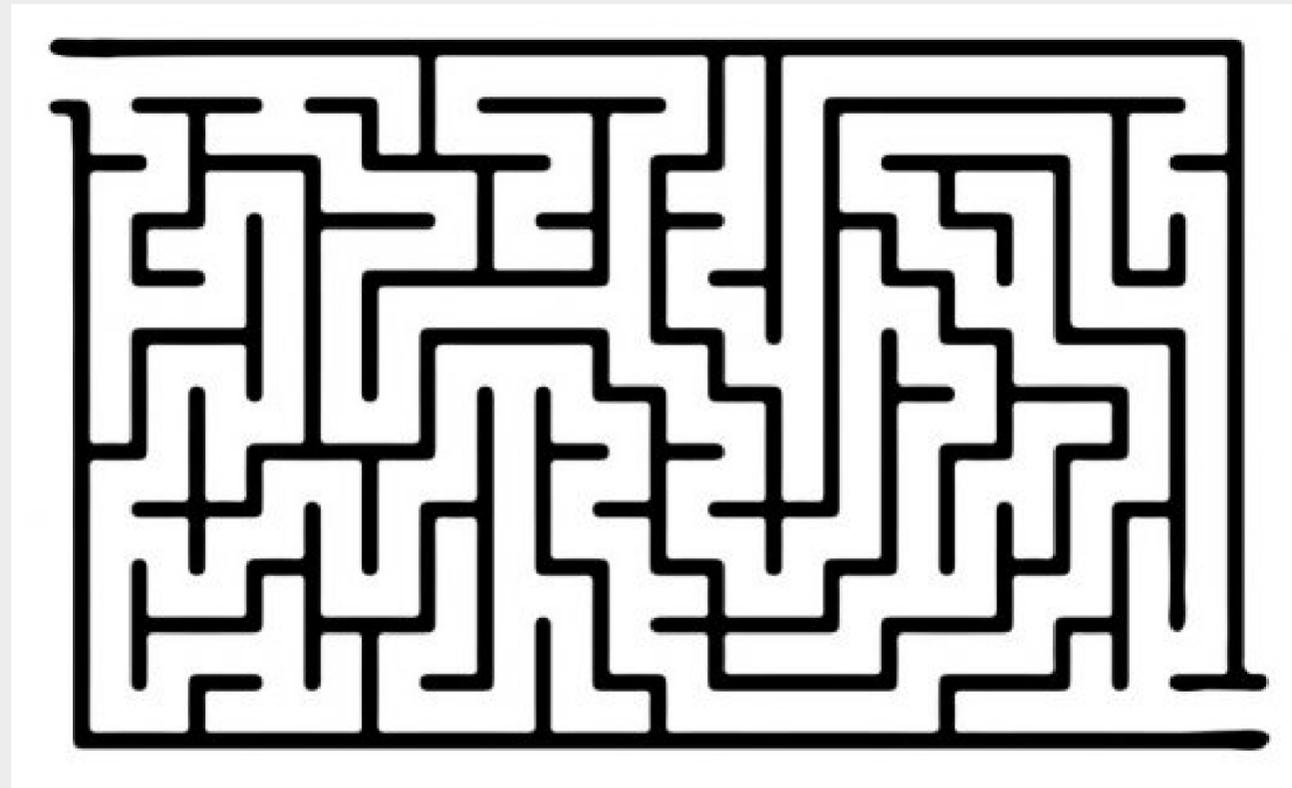
¿Es este un buen laberinto para poner a prueba ACO?

Argumenta tu respuesta



➤ Generación del entorno

➤ ¿Es este un buen laberinto para poner a prueba ACO?



No. No lo es.

¿Cómo voy a poner a prueba un algoritmo de optimización de rutas si solo existe una ruta posible hacia la comida?

➤ Generación del entorno

- Necesitamos un generador de entornos a resolver.

En este caso, un generador de laberintos, que sea capaz de generar entornos con 0, 1 o N soluciones. Para que podamos poner a prueba el algoritmo, y ver qué tal funciona.

La generación de entornos es uno de los puntos más importantes a la hora de extrapolar conocimientos y adquirir la capacidad de resolver problemas con inteligencia artificial.

Ya me lo agradeceréis cuando estudiéis Reinforcement Learning

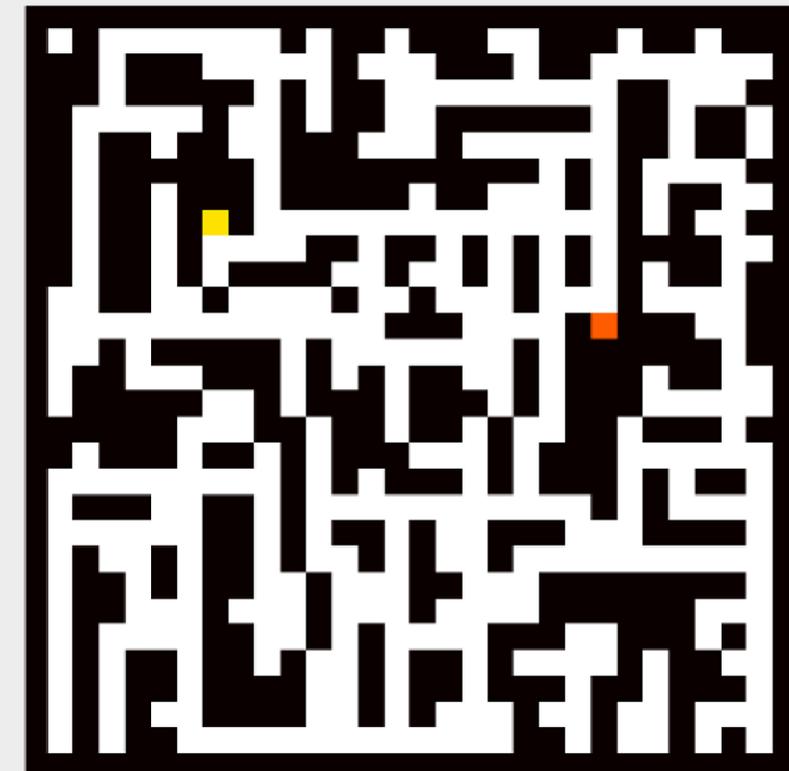
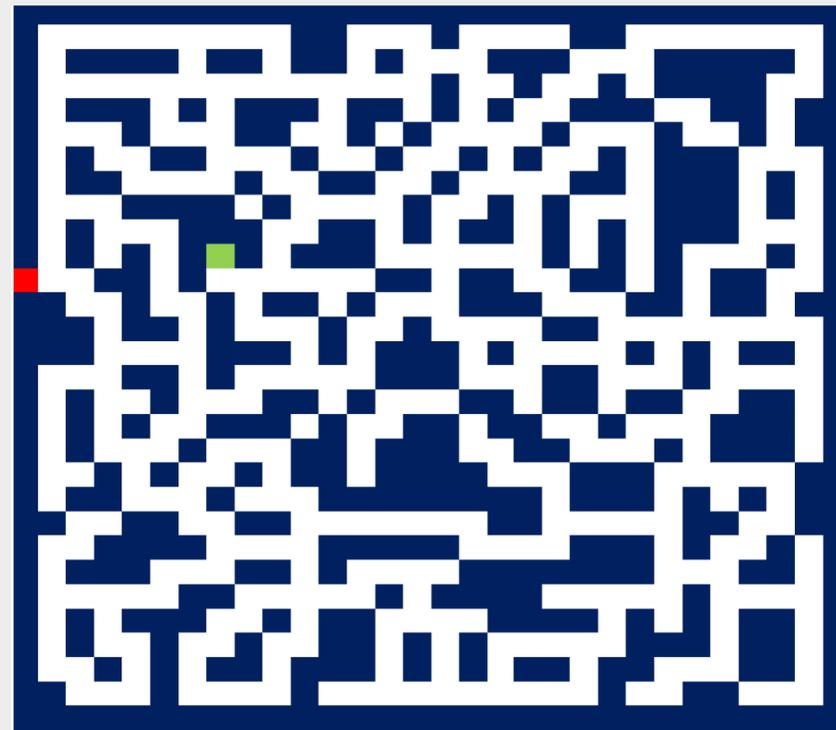
Pero, ¿cómo se genera un laberinto?

➤ Generación del entorno

- Prestar atención a la construcción de tu propio entorno es esencial (en vez de que te den uno hecho).

Por ejemplo:

¿Son estos dos laberintos semejantes?

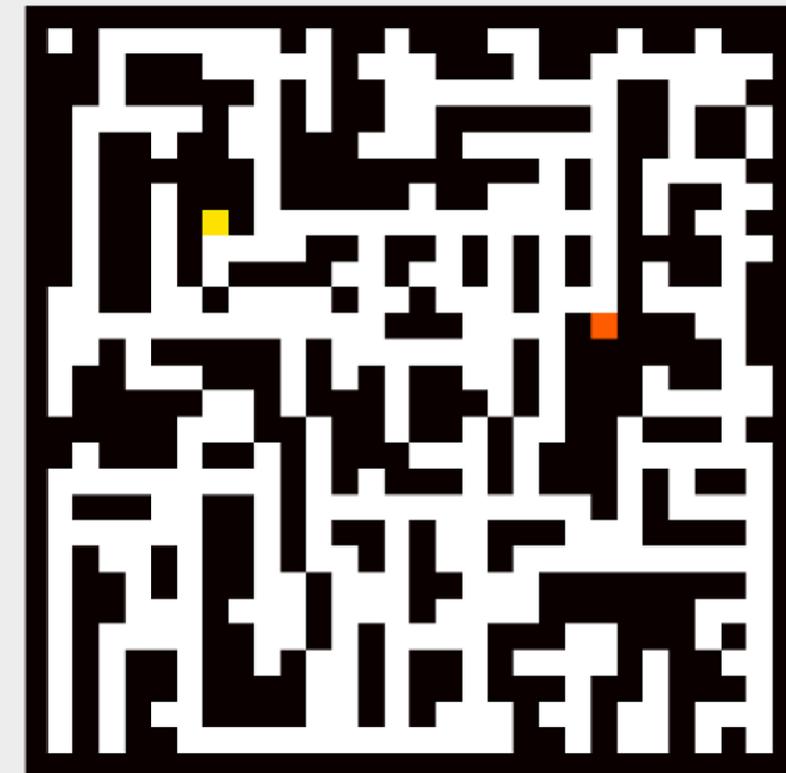


➤ Generación del entorno

➤ ¿Son estos dos laberintos semejantes?

No.

Aunque ambos tienen N soluciones, el de la izquierda tiene un problema de “rotondas” y el de la derecha de “rotondas +”plazas”. Por lo que es más complejo de resolver para las hormigas.



➤ Generación del entorno

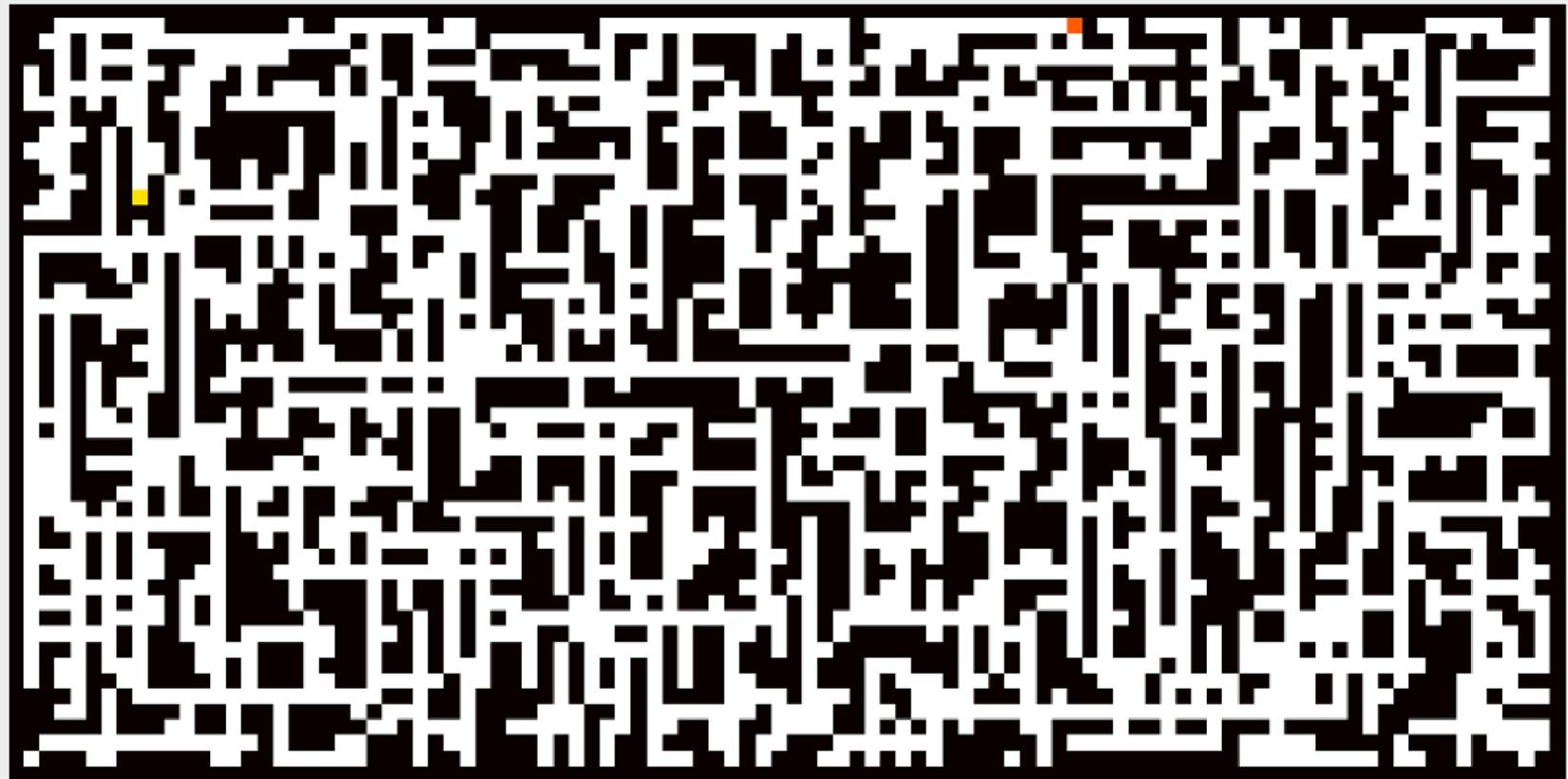
➤ ¿Pero para qué necesito el algoritmo para resolver un laberinto?

Ya puedo hacerlo yo.

➤ Generación del entorno

➤ ¿Pero para qué necesito el algoritmo para resolver un laberinto?

¿Puedes encontrar la ruta óptima en un entorno complejo?



➤ Generación del entorno

➤ ¿Completamente seguro?



➤ Generación del entorno

- Por fortuna, el primer ejercicio consiste en crear un generador de laberintos

Pongámonos a desarrollar nuestro propio entorno a resolver



Abre el ejercicio:
1 Creación de entorno.ipynb

➤ Número de hormigas exploradoras

➤
¿Con cuántas hormigas vamos a trabajar?

➤
¿Cuáles serían las variables que lo determinarían?

➤ Número de hormigas exploradoras

➤ ¿Con cuántas hormigas vamos a trabajar?

➤ ¿Cuáles serían las variables que lo determinarían?

Varios estudios utilizan un número fijo de hormigas, usando 100 como óptimo.

➤ Número de hormigas exploradoras

➤ ¿Con cuántas hormigas vamos a trabajar?

➤ ¿Cuáles serían las variables que lo determinarían?

Desde mi punto de vista, es mejor utilizar el tamaño de laberinto como variable a tener en cuenta.

Por lo que trabajaremos con un número de hormigas proporcional al tamaño del laberinto.

$n_{\text{hormigas}} = \text{longitud_lado}/5.$



➤ Número de hormigas exploradoras

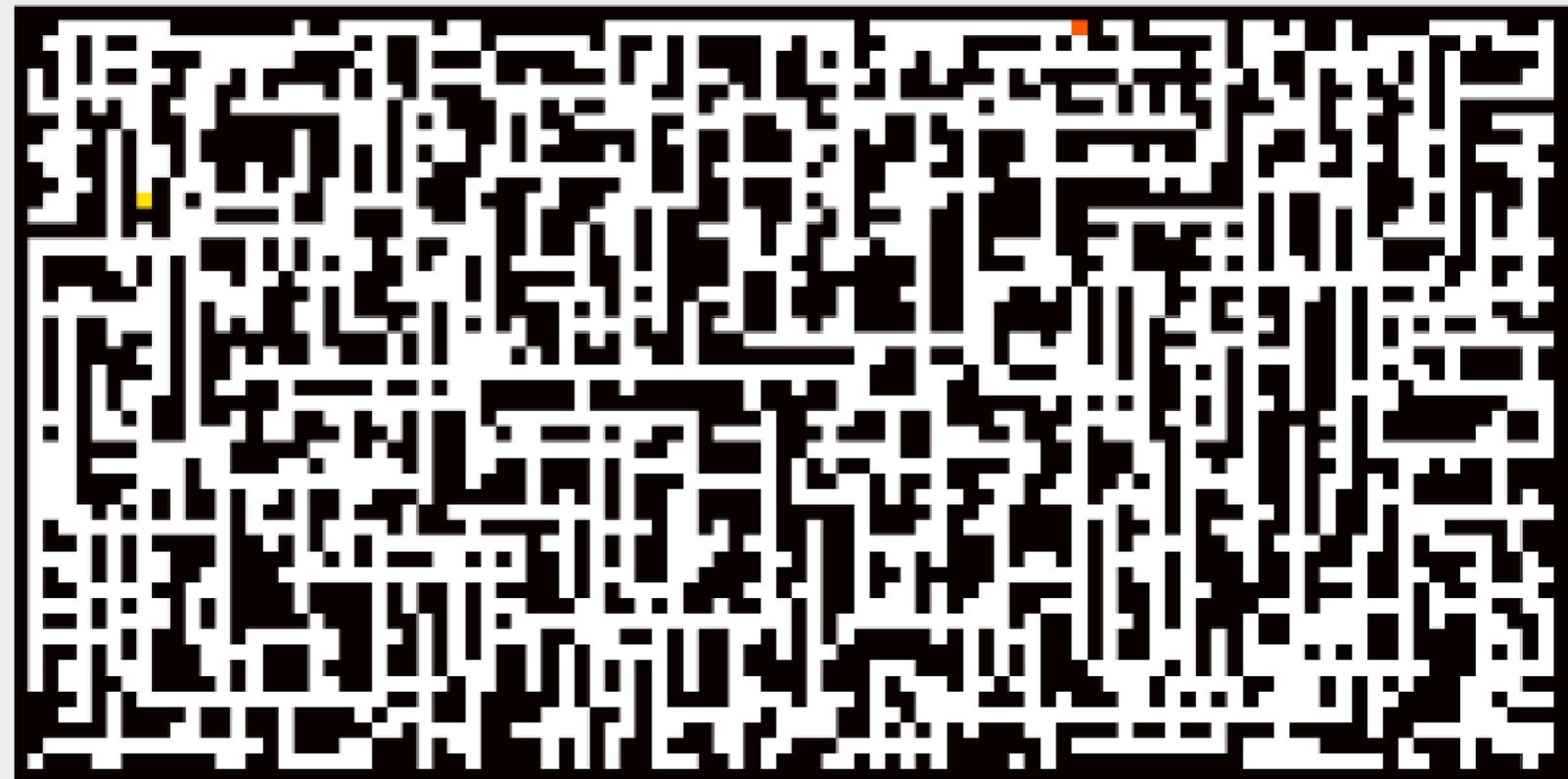
➤ ¿Con cuántas hormigas vamos a trabajar?

➤ ¿Cuáles serían las variables que lo determinarían?

Trabajaremos con un número de hormigas proporcional al tamaño del laberinto.

$n_{\text{hormigas}} = \text{longitud_lado}/5.$

¿Pero qué sucede si el laberinto es rectangular?



➤ Número de hormigas exploradoras

➤ ¿Con cuántas hormigas vamos a trabajar?

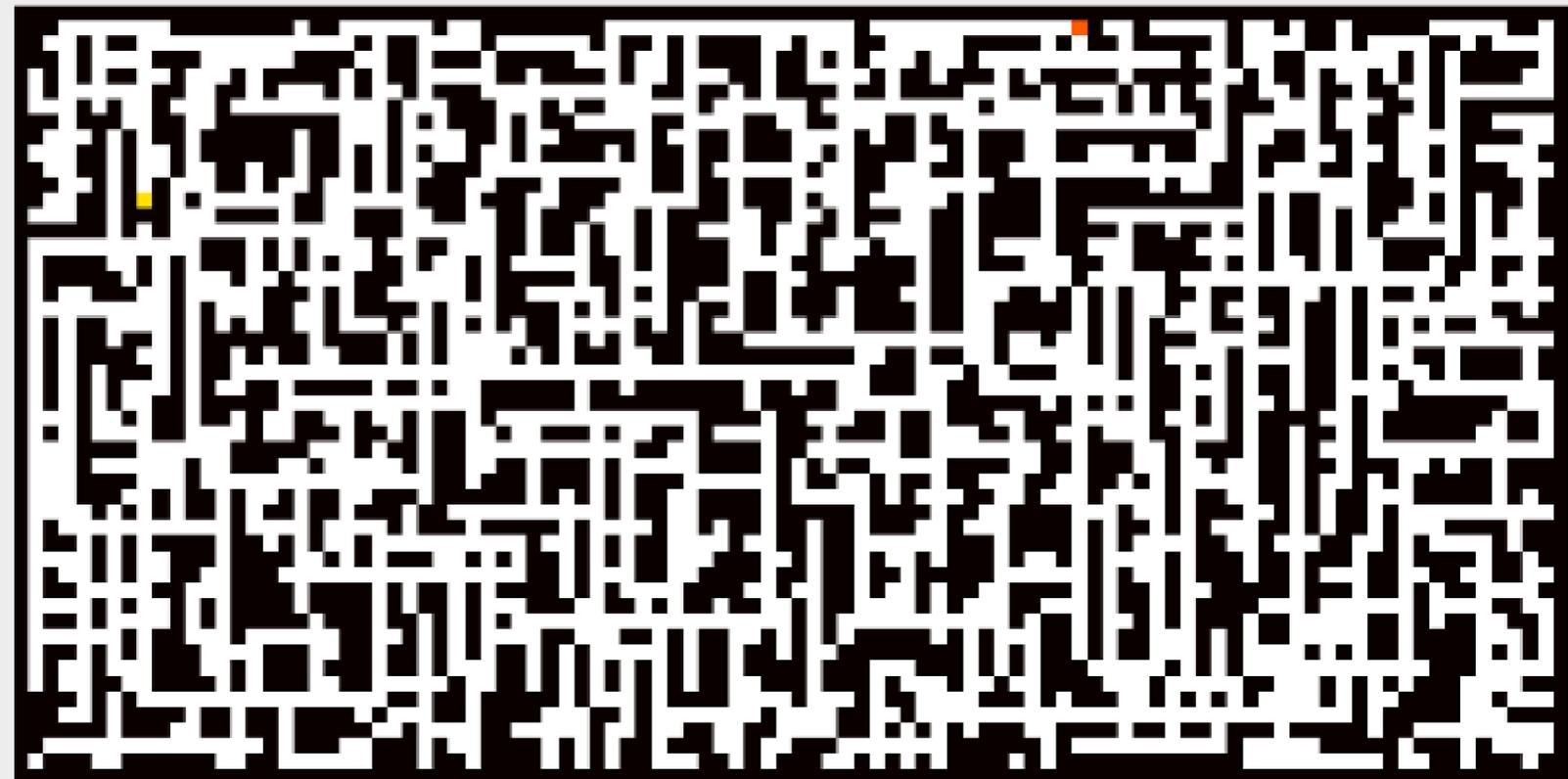
➤ ¿Cuáles serían las variables que lo determinarían?

¿Qué sucede si el laberinto es rectangular?

Podríamos calcular la hipotenusa, por ejemplo.

Trabajaremos con un número de hormigas proporcional al tamaño del laberinto.

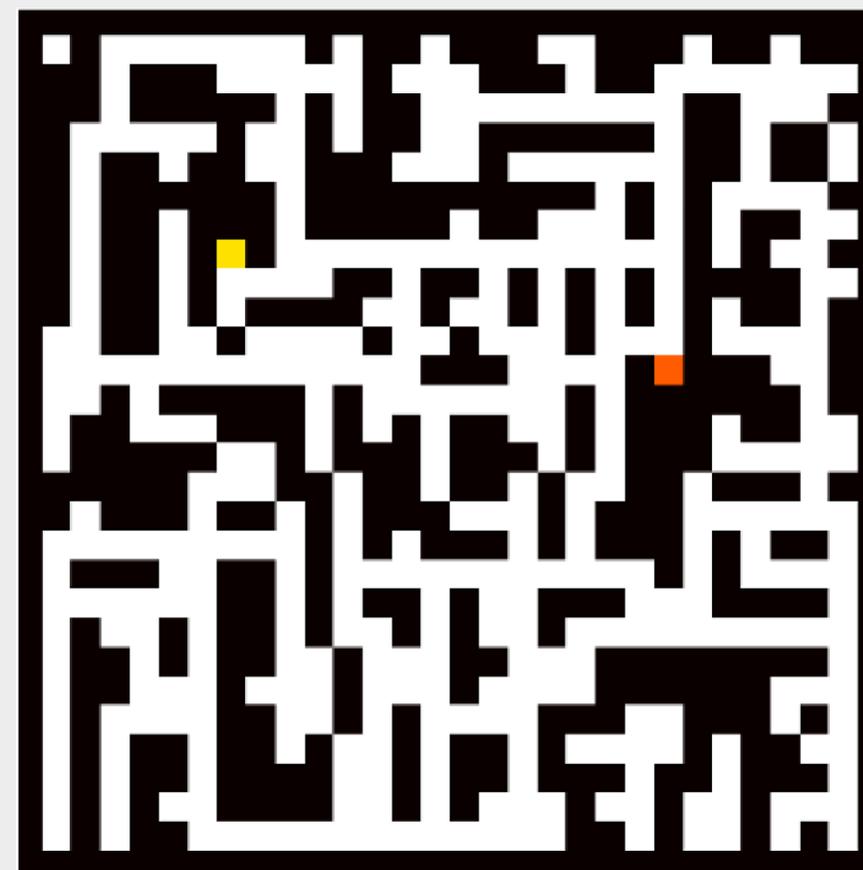
$n_{\text{hormigas}} = \text{hipotenusa}/5.$



➤ Número de hormigas exploradoras

➤ ¿Y en estos dos casos?, ¿trabajaríamos con el mismo número de hormigas?

Argumenta la respuesta.



➤ Número de hormigas exploradoras

➤ ¿Y en estos dos casos? ¿trabajaríamos con el mismo número de hormigas?

Argumenta la respuesta.

Los entornos son muy distintos y, por lo tanto, debemos afrontarlos de distinta manera.

El laberinto azul (50 x 50) tiene rotondas

El laberinto negro (50 x 50) tiene rotondas y plazas

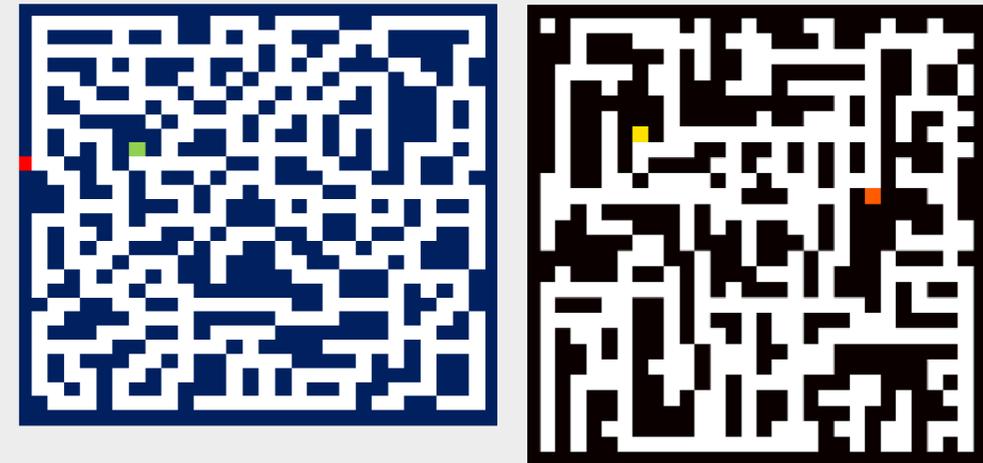
Laberinto azul: $n_hormigas = longitud_lado/5 \rightarrow 10$ hormigas

Laberinto negro: $sum(laberinto.shape)/2 \rightarrow 50$ hormigas

Imaginemos ahora que fueran laberintos de 500 X 500

Laberinto azul: $n_hormigas = longitud_lado/5 \rightarrow 100$ hormigas

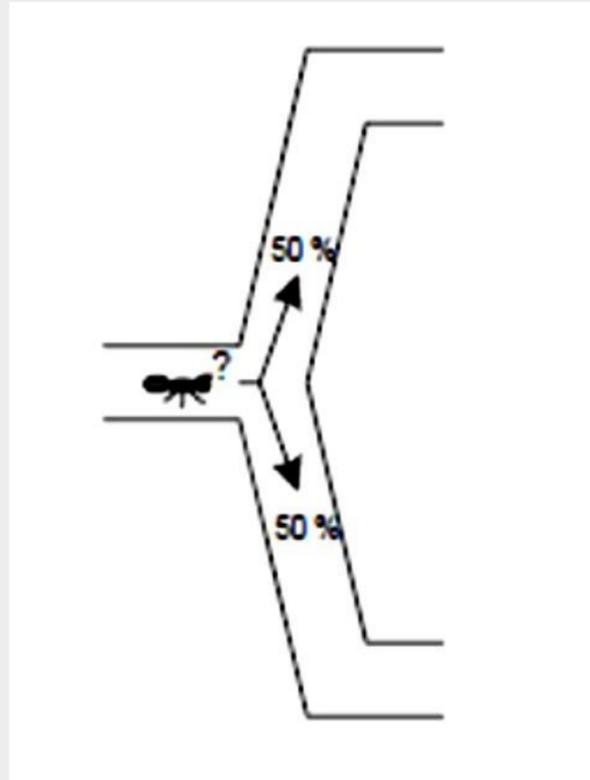
Laberinto negro: $sum(laberinto.shape)/2 \rightarrow 500$ hormigas (100 no serían suficientes)



➤ Movimiento de las hormigas

➤ Elección del camino: ¿Cómo funciona?

La hormiga llega a un cruce y debe tomar una decisión.



Dependiendo de la intensidad de la feromona existente, la hormiga decide el camino a recorrer. A mayor cantidad, mayor probabilidad tiene de elegir esa ruta.

Las hormigas prefieren de manera probabilística los caminos con mayor valor de feromonas. En caso de llegar a un cruce y no detectar ninguna feromona, la hormiga tiene la misma probabilidad de elegir un camino u otro.

➤ Movimiento de las hormigas

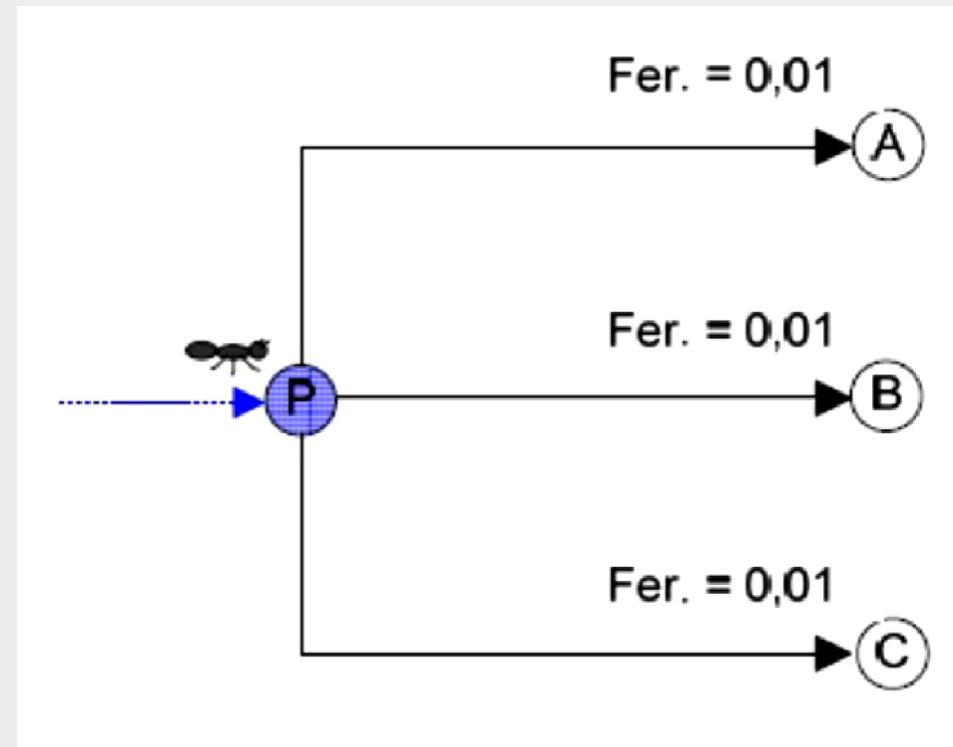
➤ Elección del camino: ¿Cómo funciona?

Suponiendo que la feromona inicial es 1 (no puede ser 0 porque entonces no se exploraría). La elección del camino sería:

Probabilidad camino A = $1/3 = 33,33\%$

Probabilidad camino B = $1/3 = 33,33\%$

Probabilidad camino C = $1/3 = 33,33\%$



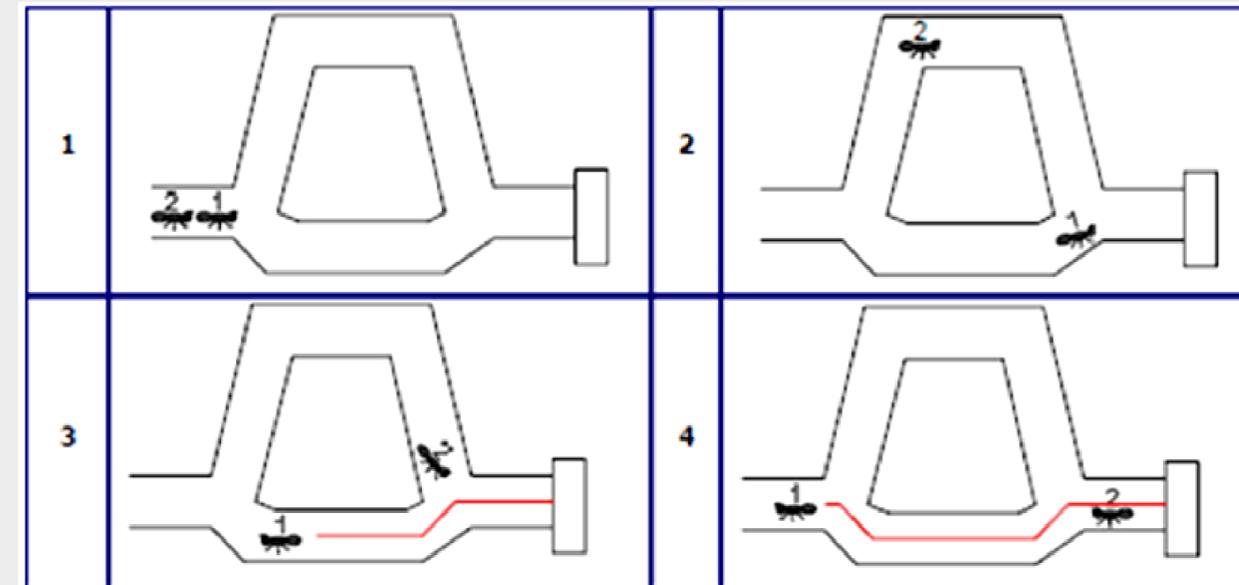
➤ Movimiento de las hormigas

➤ Elección del camino: ¿Cómo funciona?

En la imagen 1 vemos 2 hormigas que se encuentran con una bifurcación

En este caso, cada una de ellas elige un camino distinto (no tendría que ser así, es un proceso estocástico).

En las imágenes 3 y 4 podemos observar que, cada vez que una hormiga encuentra la comida, regresa al hormiguero por el mismo camino que ha recorrido. Depositando mientras lo hace una “feromona de éxito”.



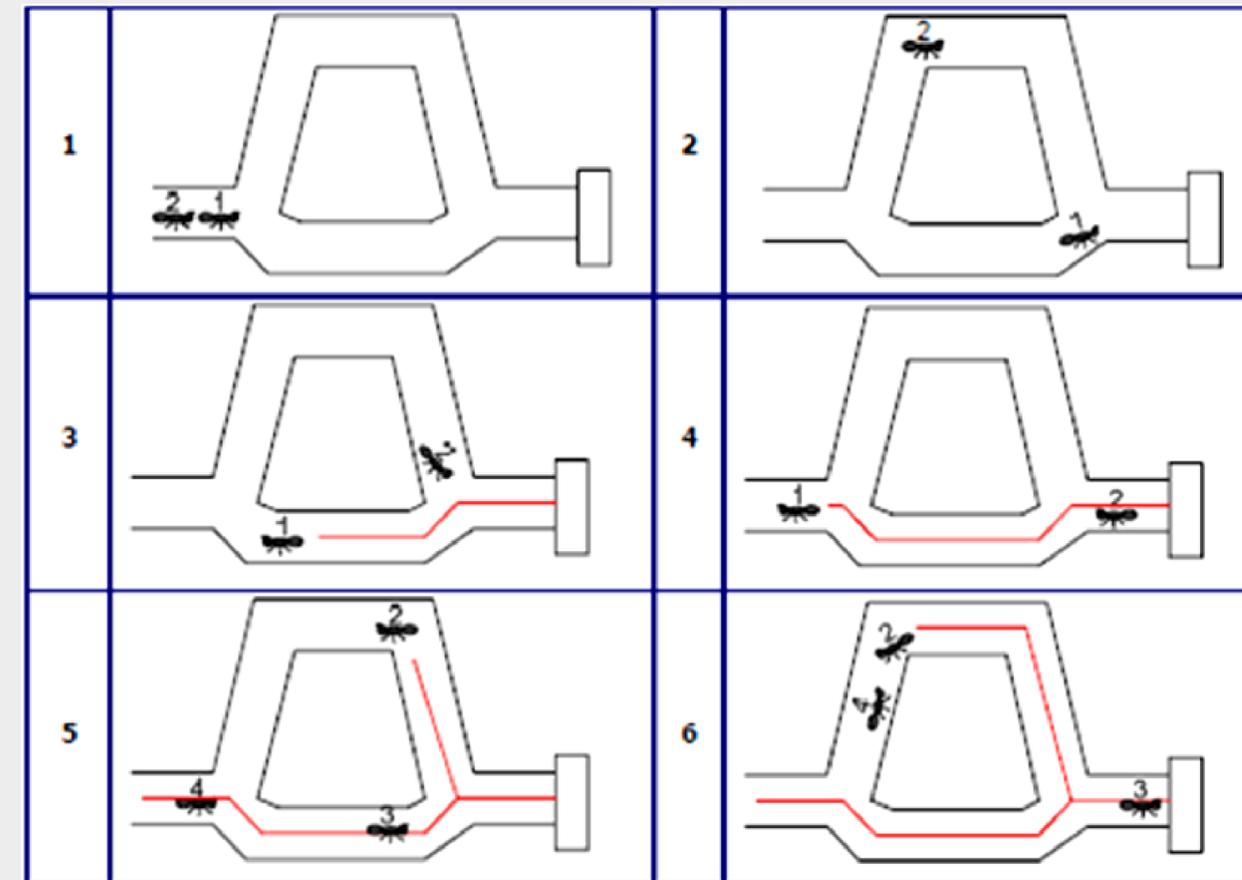
➤ Movimiento de las hormigas

➤ Elección del camino: ¿Cómo funciona?

En la imagen 5, vemos como una tercera hormiga se ha decantado por seguir el camino que está marcado por la feromona de éxito.

Sin embargo, en la imagen 6, podemos ver que la hormiga cuarta decide seguir el camino que aún no tiene feromonas depositadas.

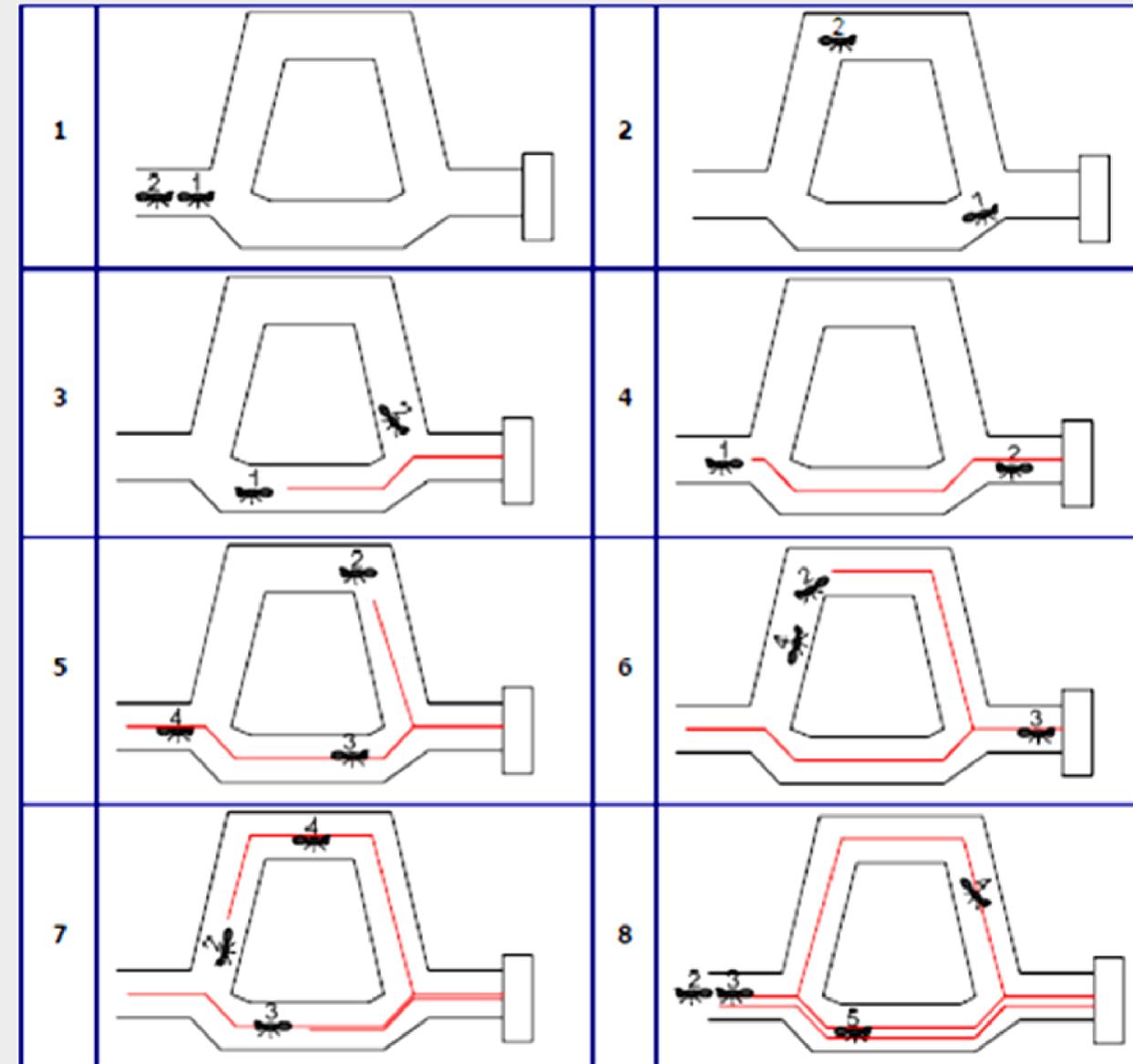
El movimiento de las hormigas es aleatorio. La probabilidad de elección de camino es mayor cuanto más feromonas contenga. Sin embargo un camino puede tener un 95% de probabilidad de ser elegido y la hormiga podría decantarse por el camino del 5% de probabilidad.



➤ Movimiento de las hormigas

➤ Elección del camino: ¿Cómo funciona?

En la imagen 8 vemos que el camino más corto es reforzado con feromonas más frecuentemente que el camino más largo, por lo que cada vez será más probable que las hormigas elijan ese camino.



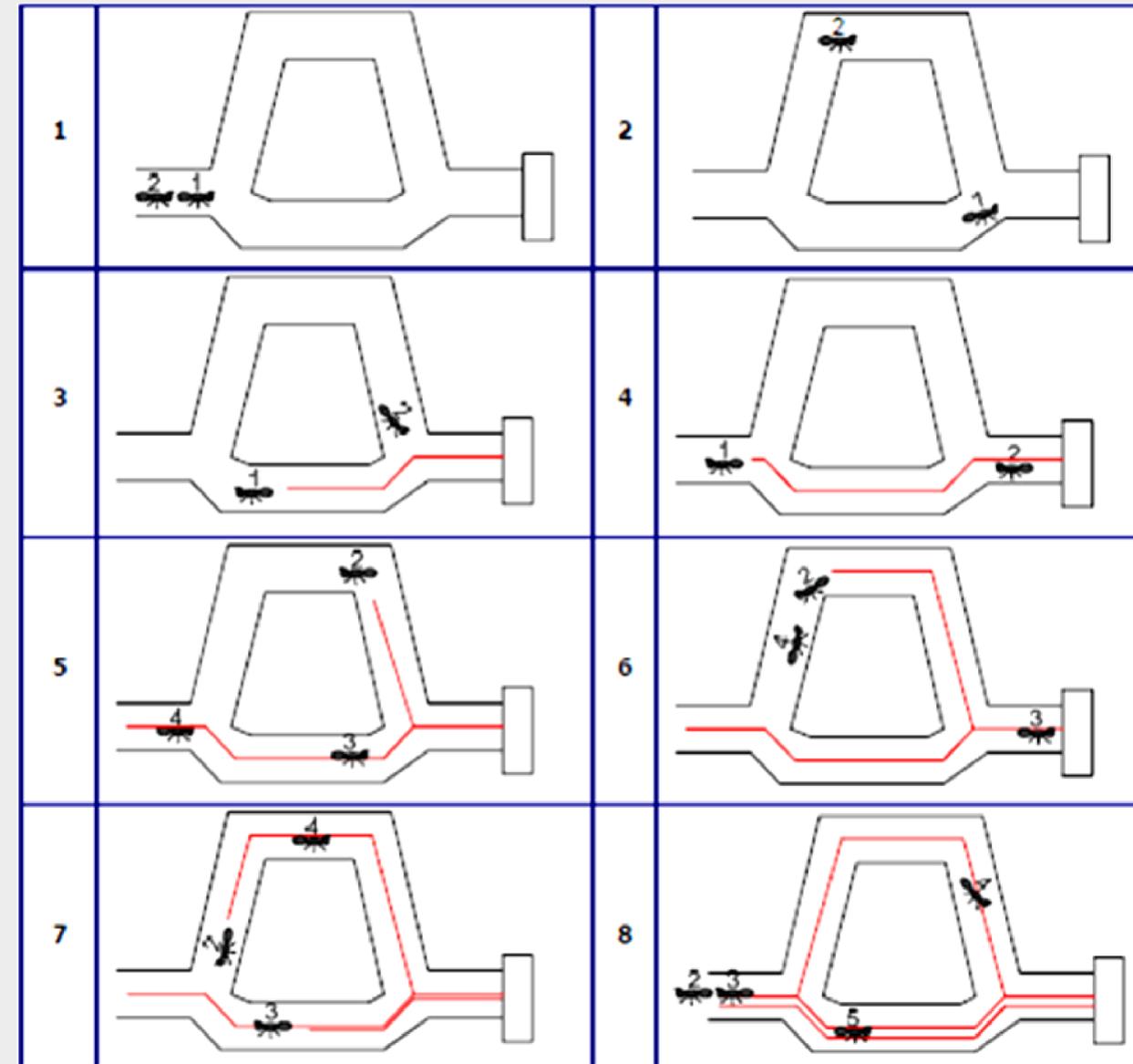
➤ Movimiento de las hormigas

➤ Elección del camino: ¿Cómo funciona?

En entornos reales las feromonas sufren un proceso de evaporación.

Si bien, la teoría indica que en entornos simulados de computación, este proceso de evaporación tiene un coste computacional demasiado elevado y su aporte en la resolución del problema es menor.

Personalmente discrepo. Y lo vamos a comprobar en el ejercicio de hoy.



➤ Movimiento de las hormigas

¿Se ha comprendido cómo se mueven las hormigas?

➤ Movimiento de las hormigas

¿Qué pasos hay que dar
para programarlo?



Intenta detallarlo lo máximo posible.

Intenta pensar en todos los casos que pueden darse.

Escribidlo y me contáis (5 minutos).

➤ Movimiento de las hormigas

➤ ¿Qué pasos hay que dar para programarlo?

Básicamente, la hormiga debe mirar alrededor para determinar cuantas opciones de camino tiene. No debe poder atravesar paredes.

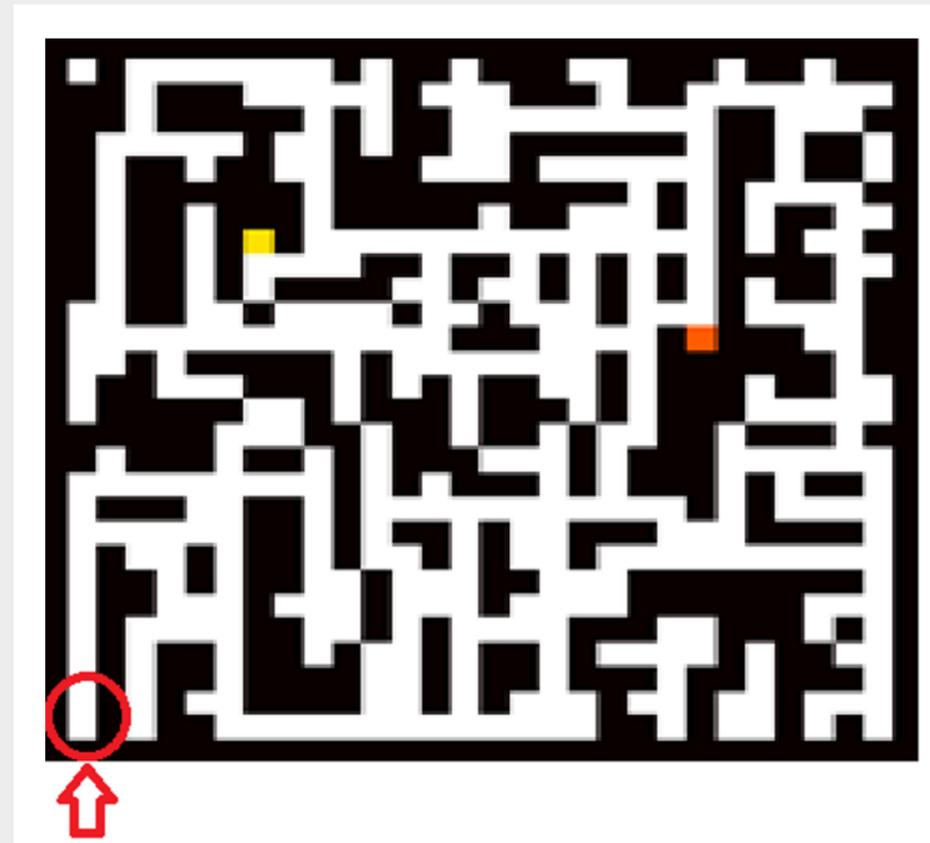
Es muy importante recordar y, anular, la opción del camino por el que ha venido. Si no hacemos esto, la hormiga puede estar dando “un pasito palante, un pasito patras” todo el tiempo. Y no es exactamente el tipo de comportamiento deseado.

Pero claro, este comportamiento programado nos generaría un problema en determinados casos...

➤ Movimiento de las hormigas

➤ ¿Qué pasos hay que dar para programarlo?

Pero claro, este comportamiento programado nos generaría un problema en determinados casos...

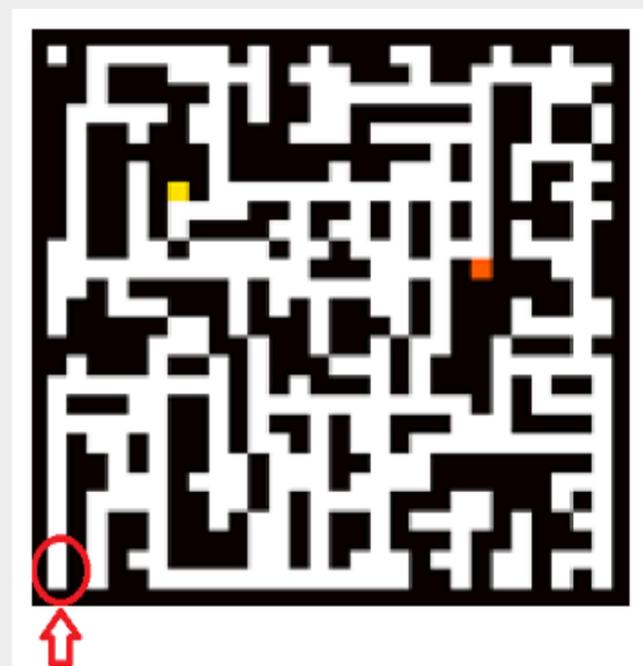


Aquí nos quedaríamos atrapados, sin poder ir hacia ninguna parte.

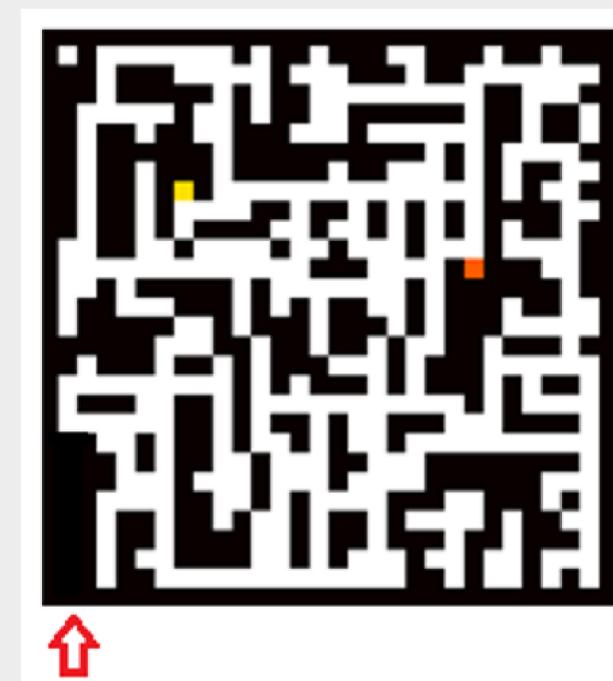
➤ Movimiento de las hormigas

➤ ¿Qué pasos hay que dar para programarlo?

En estos casos, la hormiga debe detectar que se ha quedado “atrapada”, y debe ir retrocediendo, anulando las casillas con una feromona “sin salida”, para que ninguna otra hormiga elija ese camino.



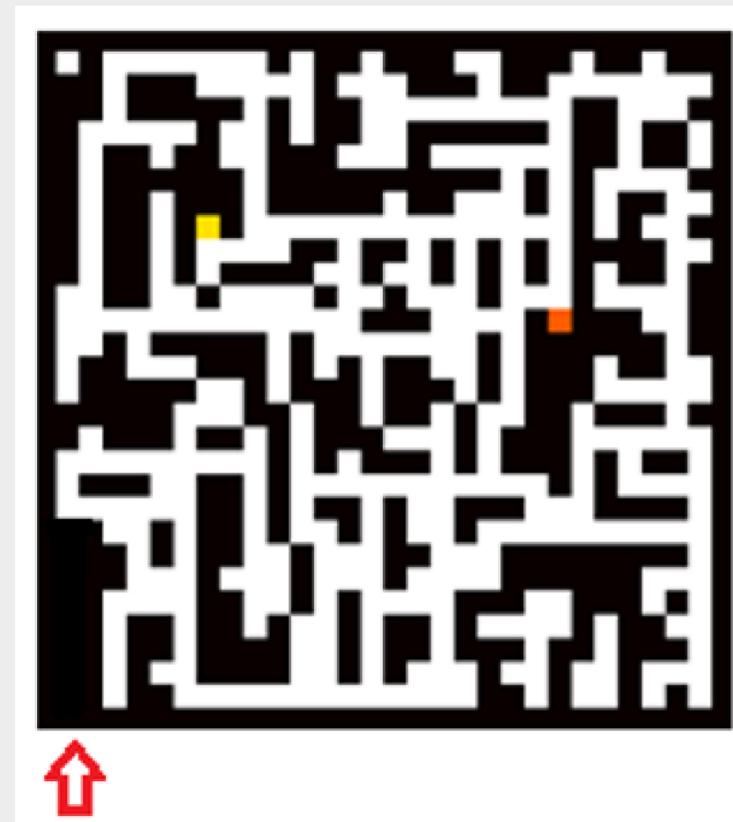
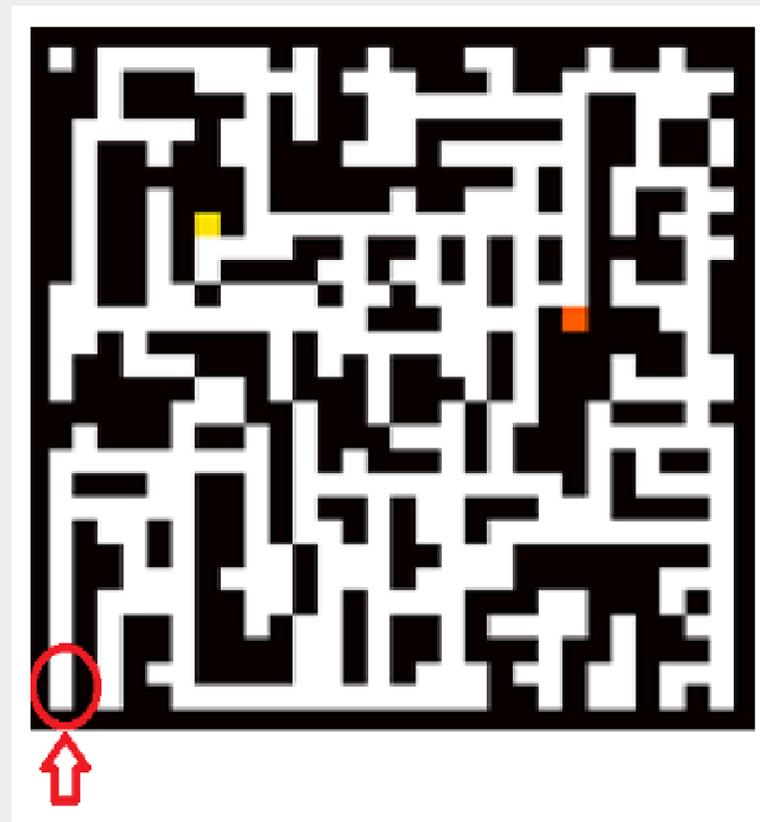
Repetirá este comportamiento hasta que llegue a un cruce.



➤ Movimiento de las hormigas

➤ ¿Qué pasos hay que dar para programarlo?

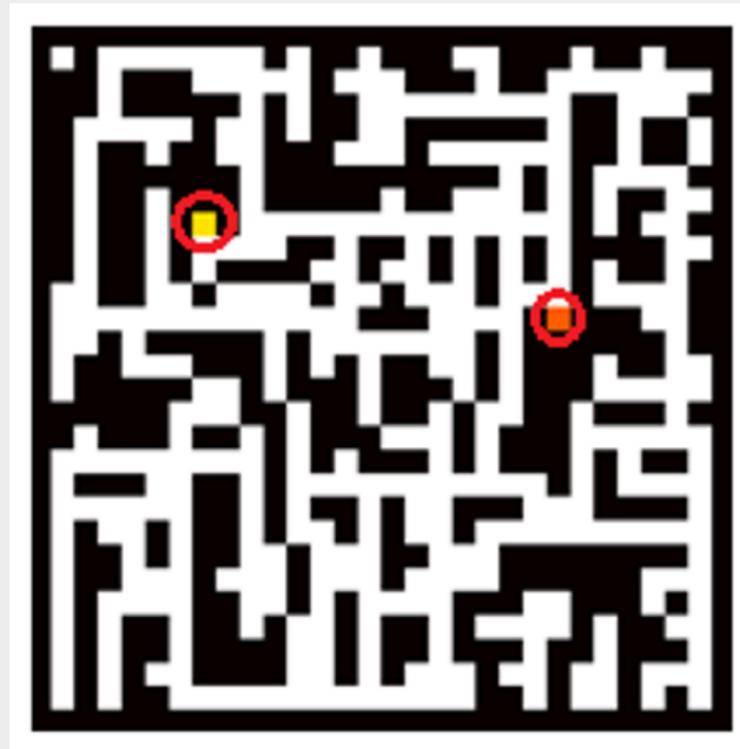
Sin embargo, este procedimiento tiene un peligro.
¿Eres capaz de verlo?



➤ Movimiento de las hormigas

➤ ¿Qué pasos hay que dar para programarlo?

Sin embargo, este procedimiento tiene un peligro.
¿Eres capaz de verlo?



Dependerá de la generación del laberinto, pero hay que tenerlo en cuenta.

El hormiguero, la comida, o ambos, podrían encontrarse en un “callejón sin salida”

Debemos comprobar que no estamos “anulando” con una feromona, ni el hormiguero, ni la comida.

➤ Movimiento de las hormigas

➤ Resumiendo

Si la hormiga no tiene ninguna opción válida de movimiento: retrocede y anula la casilla.

Si la hormiga tiene más de una opción de movimiento: decidimos el movimiento aleatoriamente, en función de las feromonas de la casilla.

Si la hormiga tiene una única opción de movimiento: se mueve.

Sacamos un número aleatorio, y lo comparamos con la distribución actual de feromonas por cada casilla. El intervalo en el que esté el número aleatorio será la dirección que elegirá.

➤ Movimiento de las hormigas

Veámoslo con un ejemplo

➤ Sacamos un número aleatorio, y lo comparamos con la distribución actual de feromonas por cada casilla. El intervalo en el que esté el número aleatorio será la dirección que elegirá.

➤ Tenemos tres posibles caminos:

a= la cantidad de feromona indica que tiene una probabilidad del 20%

b= 50% de probabilidad

c= 30% de probabilidad

➤ Sacamos un número aleatorio ➔ sale 75%

$75 > 20$ por lo que no elije el camino a

$75 > 20 + 50$ por lo que no elije el camino b

$75 < (20 + 50 + 30)$ por lo que elije el camino c

➤ Movimiento de las hormigas

- El segundo ejercicio consiste en crear a las hormigas y dotarlas de movimiento.



Abre el ejercicio:
2 Movimiento de las hormigas.ipynb

➤ Hormiga muerta

➤ En ocasiones, la decisión correcta es matar a la hormiga

¿En qué ocasiones
querríamos hacer algo así?

➤ Hormiga muerta

➤ En ocasiones, la decisión correcta es matar a la hormiga

¿En qué ocasiones
querríamos hacer algo así?

Se ha de establecer un equilibrio correcto
ente exploración y refuerzo de los
caminos.

Especialmente si tenemos en cuenta que
tenemos un número limitado de hormigas.



➤ Hormiga muerta

➤ En ocasiones, la decisión correcta es matar a la hormiga

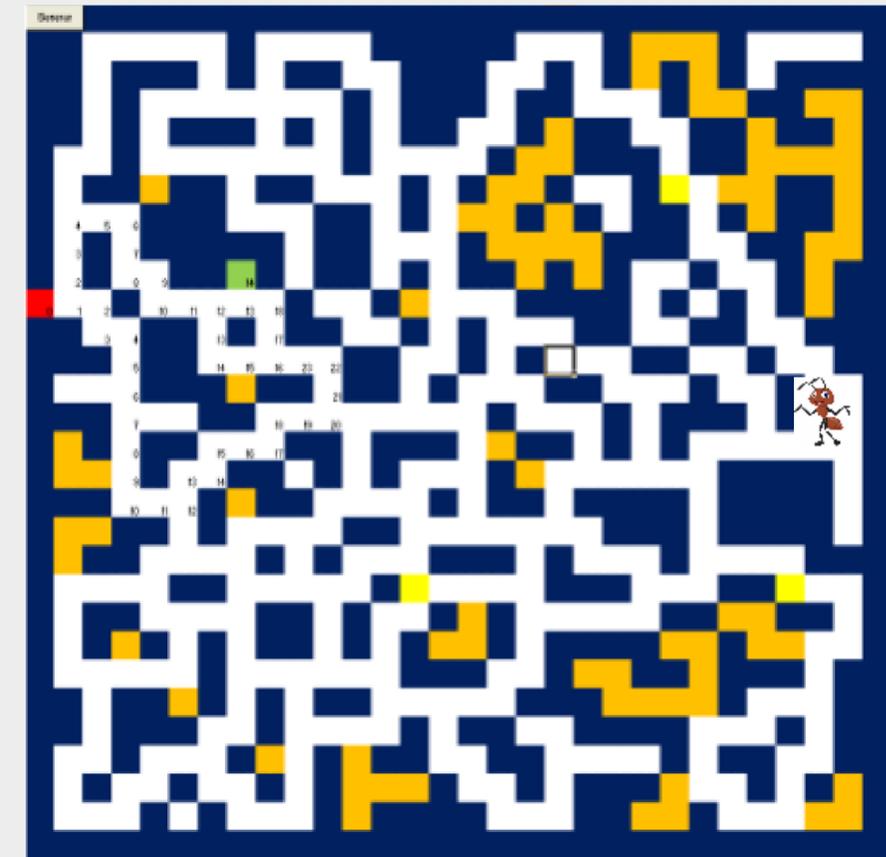
Si el punto verde es el hormiguero, y el rojo la comida, existen dos caminos principales que deberán estar acumulando feromonas.

Como ya hemos visto, el más corto de ellos, acumulará feromonas de éxito con mayor frecuencia, por lo que cada vez se volverá más probable que las hormigas lo escojan.

Sin embargo pueden existir otros caminos, por lo que la exploración es correcta.

El momento más delicado será la primera (primeras veces) que se encuentre la comida. Un camino será reforzado, pero las hormigas estarán dispersas por todo el laberinto.

Debemos establecer un criterio para “matar de hambre” una hormiga, que se haya perdido en el laberinto.



➤ Hormiga muerta

➤ En ocasiones, la decisión correcta es matar a la hormiga

Debemos establecer un criterio para “matar de hambre” una hormiga, que se haya perdido en el laberinto.

¿Pero cómo podríamos establecer este criterio?

Si matamos a la hormiga demasiado pronto, penalizaríamos la exploración. Pudiendo llegar al extremo de que nunca se encuentre la comida. O haciendo que la probabilidad de encontrarla fuera mucho menor que si no existiera la posibilidad de que muera de hambre.

Y si ponemos un criterio demasiado laxo, es probable que un porcentaje demasiado grande del hormiguero esté “perdido” explorando caminos que no conducen al éxito de la colonia.

¿Cómo estableceríais el criterio?

➤ Hormiga muerta

➤ En ocasiones, la decisión correcta es matar a la hormiga

¿Cómo estableceríais el criterio para matar de hambre a una hormiga?

Podría hacerse de diversas maneras

1ª manera

El proceso se inicia cuando se ha encontrado por primera vez la comida. Al haberse encontrado una vez la comida, aunque no se conoce si el camino es bueno o malo, tenemos una primera estimación de distancia con la que trabajar.

El objetivo es ver si la hormiga se ha perdido por el laberinto y, por falta de comida, muere de hambre.

Basándose en la mejor distancia hasta el momento encontrada, se decide eliminar aquellas hormigas que superan esta cantidad de una manera relevante, ya que se consideran perdidas en la red.

Si $\text{pasos_hormiga_N} > \text{pasos_mejor_solución_conocida} * 5$ →
Reiniciar hormiga N en el hormiguero.

Ok. Es una manera de hacerlo

➤ Hormiga muerta

➤ En ocasiones, la decisión correcta es matar a la hormiga

¿Cómo estableceríais el criterio para matar de hambre a una hormiga?

Podría hacerse de diversas maneras

2ª manera

En vez de basarnos en la mejor distancia conocida, dado que no sabemos cómo de buena o mala es, podemos basarnos en el tamaño del laberinto.

Si longitud pasos dados $>$ lado_laberinto * lado_laberinto →
Reiniciar hormiga N en el hormiguero.

Este sistema, al no depender de la solución encontrada hasta el momento, es más lógico.

➔ Comprobación del camino antes de depositar las feromonas

- ➔ La hormiga ha salido del hormiguero
- ➔ Ha recorrido un camino aleatorio
- ➔ Ha encontrado la comida
- ➔ Y está a punto de regresar al hormiguero depositando la feromona de éxito

Pero antes de hacer eso, tenemos
que comprobar algo muy importante

¿Qué puede ser?

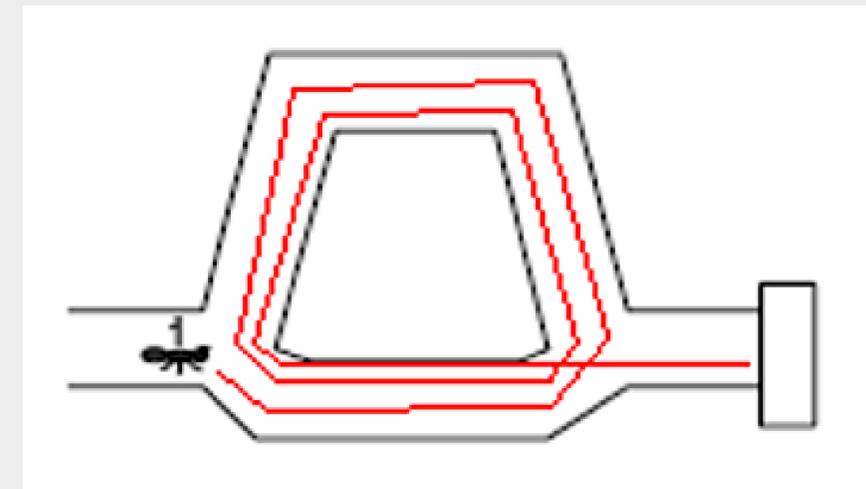
➔ Comprobación del camino antes de depositar las feromonas

➔ Efectivamente. Los bucles.

Antes de depositar la feromona de “éxito”, se ha de revisar y podar el camino que ha recorrido la hormiga desde el hormiguero hasta la comida. Eliminando las repeticiones.

Las hormigas se mueven por el laberinto aleatoriamente, de hecho, pueden volver al hormiguero en su búsqueda de comida o, en un camino circular, puede dar varias vueltas “a la rotonda” antes de salir por el lado adecuado en el que está la comida.

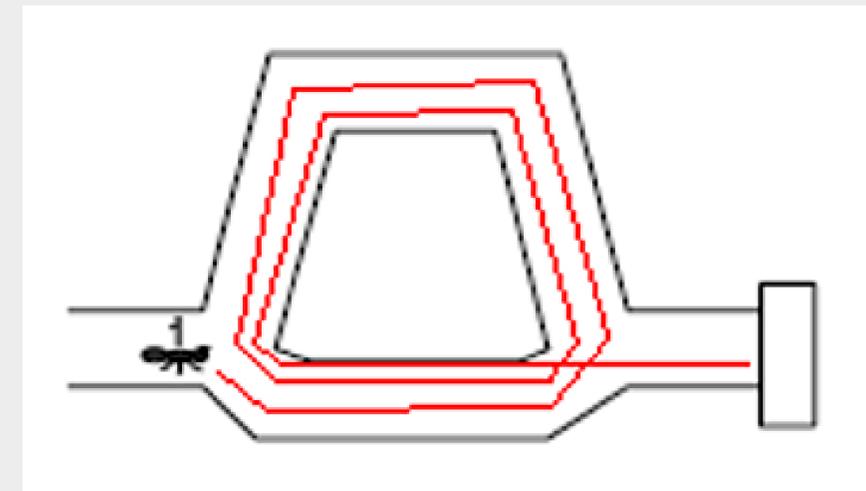
Si complicamos el problema con rotondas + plazas, la probabilidad de que la hormiga haya realizado movimientos redundantes es muy elevada.



➤ Comprobación del camino antes de depositar las feromonas

➤ Hay que podar la solución, el camino encontrado.

Explicadme, con detalle, qué pasaría si no lo hiciéramos



➔ Comprobación del camino antes de depositar las feromonas

➔ Hay que podar la solución, el camino encontrado.

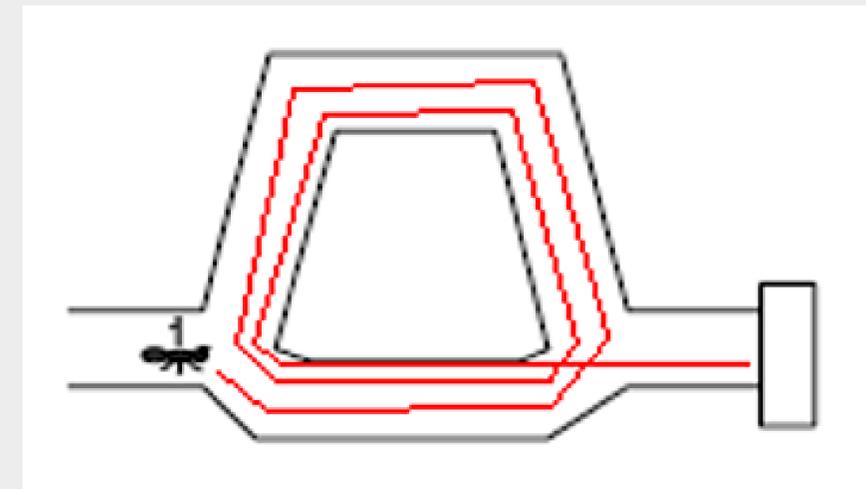
Explicadme, con detalle, qué pasaría si no lo hiciéramos

Si estamos depositando una cantidad de feromona determinada, en cada casilla recorrida por la hormiga, y no hemos realizado la poda, lo que estaremos haciendo será reforzar varias veces unas casillas determinadas.

En el caso de la imagen, reforzaremos, depositaremos mucha más feromona dentro de las casillas que pertenecen a la “rotonda”, que a las casillas que salen de ella.

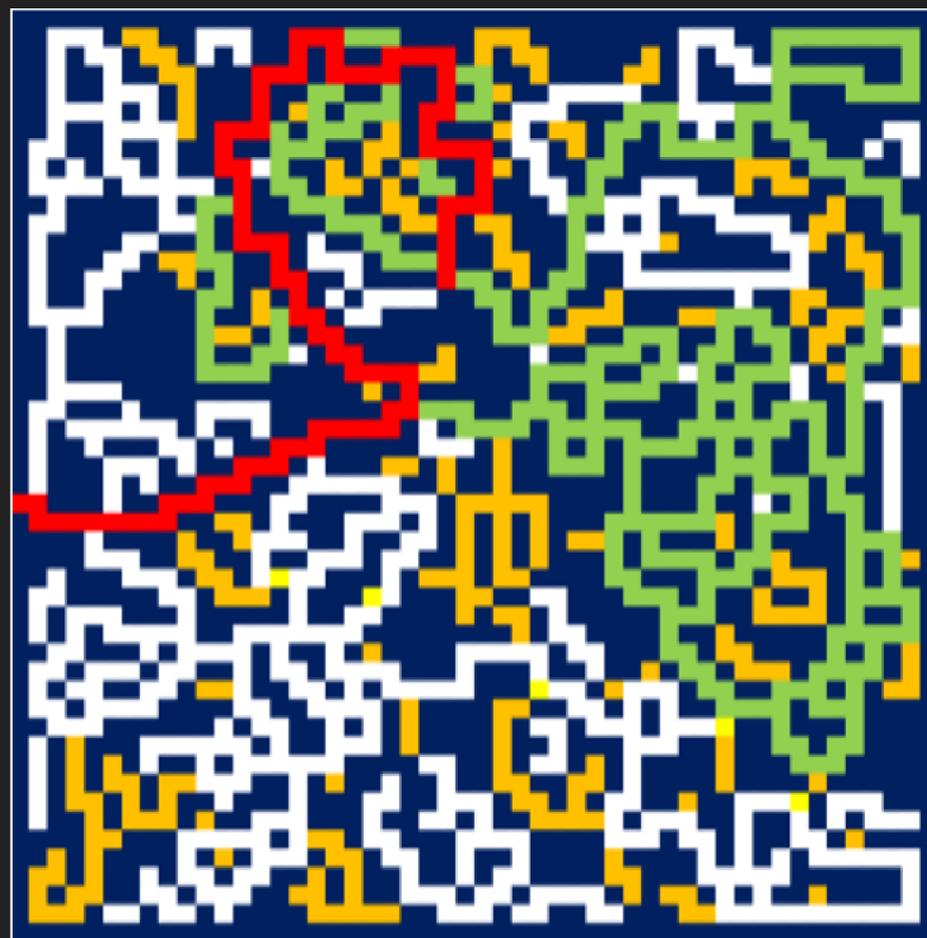
Por lo tanto, la hormiga estará tentada de seguir indefinidamente en la rotonda, siguiendo probabilísticamente el camino con mayor cantidad de feromona. Es decir, permanecerá en la rotonda.

Al menos hasta que muera de hambre.



➤ Comprobación del camino antes de depositar las feromonas

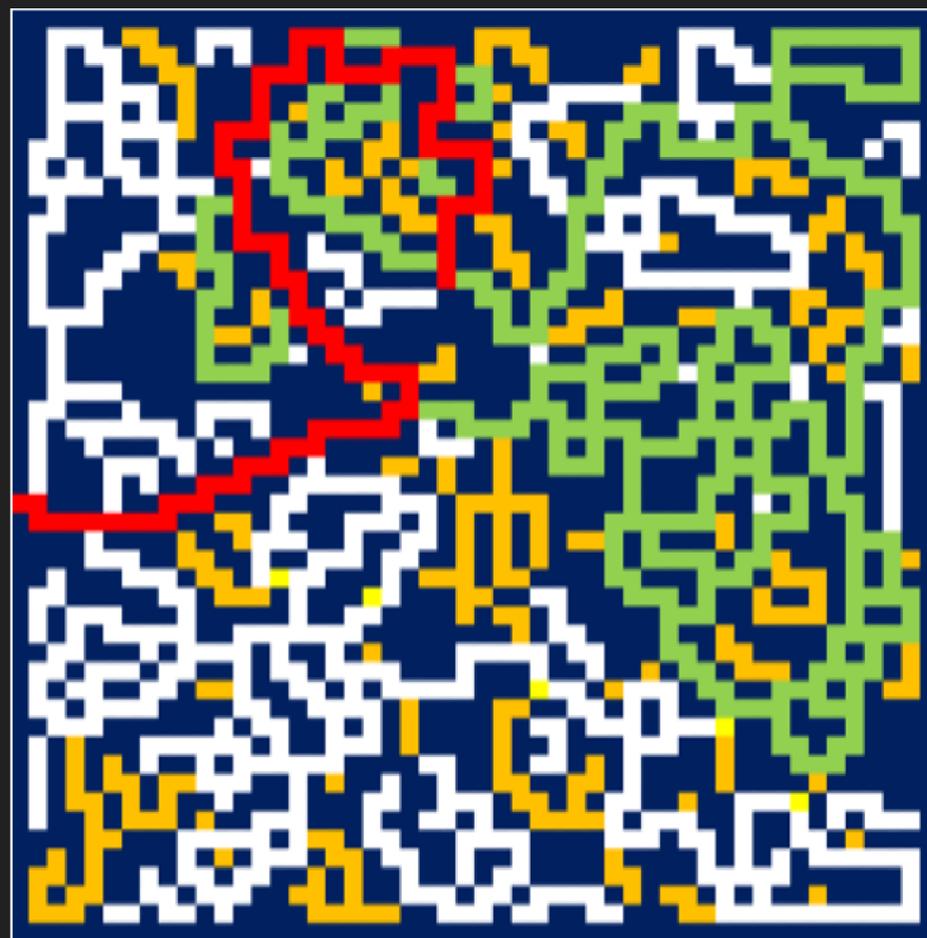
➤ Veamos un caso real



- El camino rojo + verde representa el camino real que ha recorrido la hormiga, desde el hormiguero, hasta que ha encontrado la comida.
- No lo ha encontrado precisamente a la primera ¿verdad?
- El camino rojo representa el camino podado, que ha recorrido la hormiga desde el hormiguero hasta la comida.
- El camino podado será el que será reforzado por la feromona de éxito.

➔ Comprobación del camino antes de depositar las feromonas

➔ Veamos un caso real



➔ Si usamos la matriz de pasos recorridos por la hormiga sin comprobar si existen repeticiones tendríamos, en muchos casos, problemas como el de la rotonda ya explicado. Esto supondría un serio problema dado que, en el caso de depositar feromonas sin realizar esta comprobación y poda, la feromona se depositaría varias veces en la rotonda, haciendo que las siguientes hormigas tiendan a permanecer en su interior en lugar de salir hasta la comida.

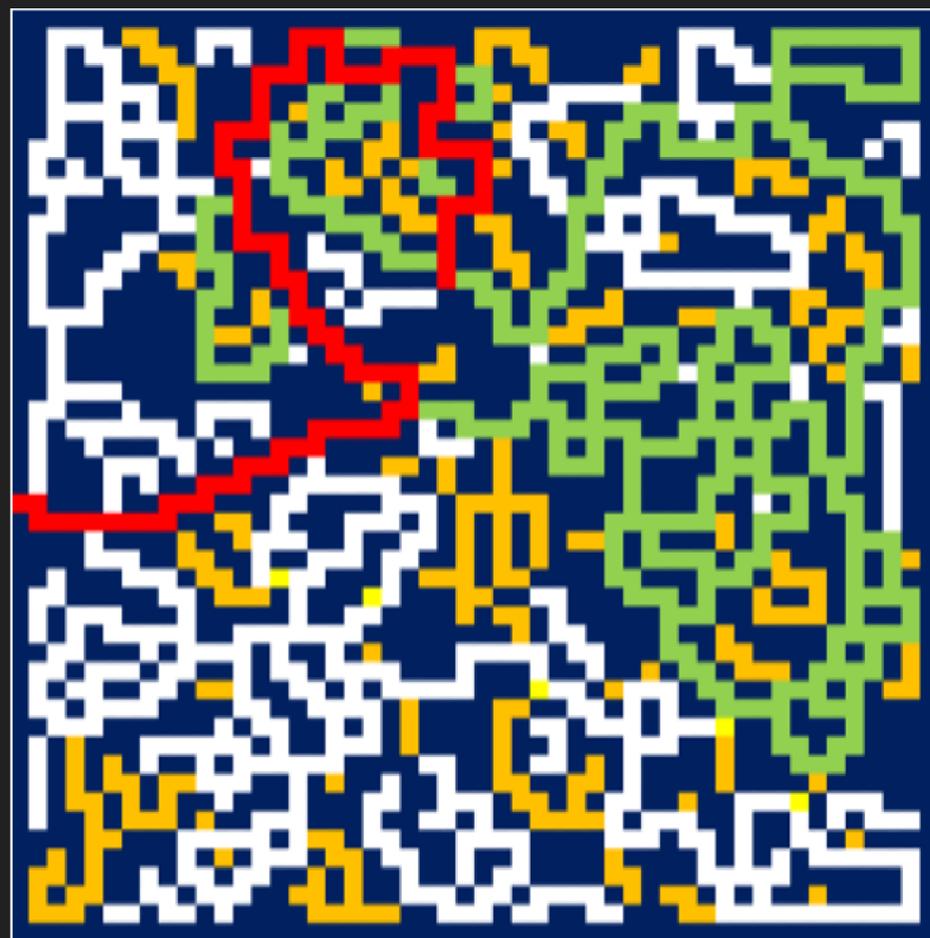
➔ Si este proceso se repitiera varias veces, las hormigas tenderían a permanecer en un bucle sin fin dentro de dicho camino circular.

➔ Ejemplo del proceso de “poda” del camino recorrido por la hormiga:

La hormiga ha recorrido el camino verde + el camino rojo para ir desde el hormiguero hasta la comida. Si hubiésemos depositado la feromona sin realizar una poda de casillas innecesarias, las siguientes hormigas habrían tendido a perderse dentro del laberinto.

➤ Comprobación del camino antes de depositar las feromonas

➤ Veamos un caso real



- Por ello, el camino recorrido por la hormiga se analiza buscando repeticiones (casillas por las que haya pasado dos veces), eliminando el recorrido sobrante.
- El camino rojo representa el recorrido realizado por la hormiga desde el hormiguero hasta la comida, habiendo eliminado las casillas redundantes: el camino verde.

➤ Comprobación del camino antes de depositar las feromonas

- El tercer ejercicio consiste en poner en práctica estos dos conceptos



Abre el ejercicio:

[3 Matar hormiga y podar la solucion.ipynb](#)

➤ Depositar feromonas

➤ Hemos llegado a un punto clave

La hormiga ha encontrado la comida y debe volver a casa para comunicar a sus compañeras que ha encontrado comida.

¿Cuánta feromona tiene que depositar?

¿Qué implicaciones tiene?

Trata de razonar la problemática lo mejor posible

El código en sí es muy simple. Pero el problema no.

¿Qué hacemos y cómo?

➤ Depositar feromonas

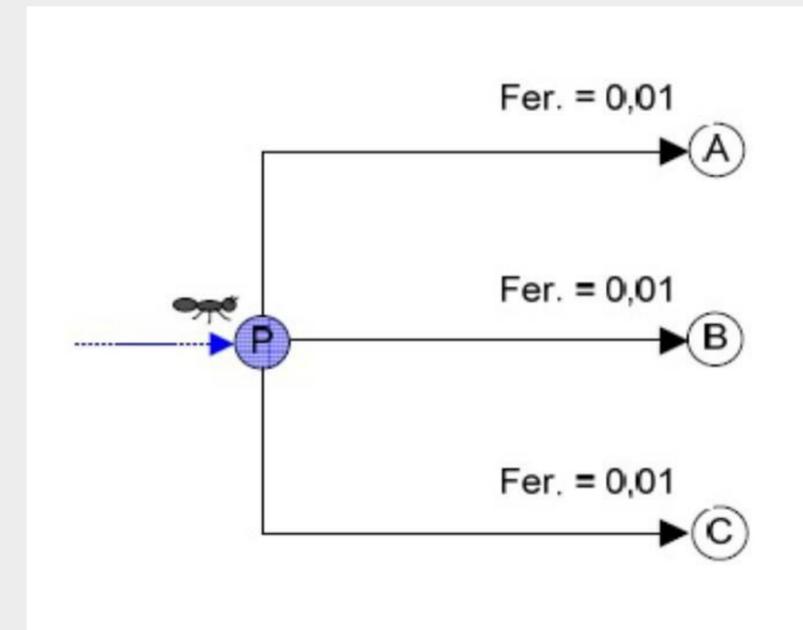
➤ ¿Cuánta feromona tiene que depositar?

➤ ¿Qué implicaciones tiene una u otra cantidad aportada?

La mayoría de la bibliografía se decanta por una aportación de feromona constante. Estableciendo que el sistema será capaz de autoregularse por sí solo si le damos el tiempo suficiente.

Vamos a analizar si es buena idea, y qué implicaciones tendría

Partimos del siguiente estado. Donde la hormiga tiene 3 caminos posibles y, de momento, cada uno tiene únicamente la feromona inicial. En este punto cada camino tiene una probabilidad del 33,33%



➤ Depositar feromonas

- ¿Cuánta feromona tiene que depositar?
- ¿Qué implicaciones tiene una u otra cantidad aportada?

Supongamos que la hormiga vuelve por el camino depositando una cantidad de feromona determinada $K=1$. ¿Qué distribución de probabilidades se encontraría la siguiente hormiga?, ¿qué distribución de probabilidades tendríamos pasadas varias hormigas? Esto es lo que queremos analizar.

Para simplificar los cálculos, vamos a presuponer que la feromona inicial es 1 y que el aporte constante de cada hormiga es también 1

	Feromona inicial	N.º hormigas	Probabilidad
A	1	1	50 %
B	1	0	25 %
C	1	0	25 %
	3	1	

Solo ha pasado una hormiga, reforzando el camino A.

Ahora este tiene un 50% de probabilidades de ser elegido, la próxima vez que una hormiga llegue a ese cruce.

➤ Depositar feromonas

- ¿Cuánta feromona tiene que depositar?
- ¿Qué implicaciones tiene una u otra cantidad aportada?

Pasa una segunda hormiga, elije el camino B, depositando la feromona.

	Feromona inicial	N.º hormigas	Probabilidad
A	1	1	40 %
B	1	1	40 %
C	1	0	20 %
	3	2	

Han pasado 3 hormigas, reforzando los caminos correspondientes.

	Feromona inicial	N.º hormigas	Probabilidad
A	1	2	50 %
B	1	1	33 %
C	1	0	17 %
	3	3	

Han pasado 8 hormigas.
¿Empezáis a ver el problema?

	Feromona inicial	N.º hormigas	Probabilidad
A	1	7	73 %
B	1	1	18 %
C	1	0	9 %
	3	8	

➤ Depositar feromonas

- ¿Cuánta feromona tiene que depositar?
- ¿Qué implicaciones tiene una u otra cantidad aportada?

Existe el peligro de que la primera hormiga condicione al resto.

Si la primera elige el camino A, la segunda tendría un 50% de probabilidad de realizar la misma elección.

Si lo hiciese, la tercera tendría un 60% de repetir el mismo camino.

Si lo hicieran 5, la sexta hormiga tendría un 75% de probabilidades de realizar la misma ruta que las anteriores. Y quizás la ruta A no es la mejor.

	Feromona inicial	N.º hormigas	Probabilidad	N.º hormigas	Probabilidad	N.º hormigas	Probabilidad
A	1	1	50 %	2	60 %	5	75 %
B	1	0	25 %	0	20 %	0	13 %
C	1	0	25 %	0	20 %	0	13 %
	3	1		2		5	

Pero además, hay otro problema.
¿Podéis verlo?

➤ Depositar feromonas

- ¿Cuánta feromona tiene que depositar?
- ¿Qué implicaciones tiene una u otra cantidad aportada?

Si el mismo número de hormigas recorren el mismo número de caminos, este sistema no tendría sentido.

Por ejemplo, 4 hormigas recorren cada una de las tres opciones. El depósito global de feromonas no indica cual es la mejor opción descubierta hasta ahora. Por lo que la siguiente hormiga tomaría una decisión al azar. Habríamos vuelto a la casilla de salida.

	Feromona inicial	N.º hormigas	Probabilidad
A	1	4	33 %
B	1	4	33 %
C	1	4	33 %
	3	12	

Una solución basada en un aporte constante de feromona no es la solución adecuada.

¿Cuál es entonces la solución?

➤ Depositar feromonas

➤ ¿Cuánta feromona tenemos que depositar?

Podríamos hacer que la cantidad a depositar dependiera de la distancia recorrida (hormiguero a comida).

Analicemos si es una buena idea.

➤ Depositar feromonas

➤ ¿Cuánta feromona tenemos que depositar?

Podríamos hacer que la cantidad a depositar dependiera de la distancia recorrida (hormiguero a comida).

Analicemos si es una buena idea.

Supongamos que la hormiga vuelve por el camino recorrido, depositando una cantidad de feromona determinada por $K=1/\text{distancia}$. Siendo la distancia que hay entre el hormiguero y la comida.

¿Qué distribución de probabilidades se encontraría la siguiente hormiga?

Supongamos el siguiente problema:

- El camino A tiene una distancia entre el hormiguero y la comida de 50 pasos
- El B de 45 pasos
- El C de 60 pasos.

La hormiga conoce el número de pasos porque está regresando al hormiguero.

➤ Depositar feromonas

➤ ¿Cuánta feromona tenemos que depositar?

El camino A tiene una distancia entre el hormiguero y la comida de 50 pasos

El B de 45 pasos

El C de 60 pasos.

En este caso el depósito inicial de feromonas debería ser 0,1 en lugar de 1 para que el resultado porcentual tenga sentido.

Podemos ver la distribución de probabilidad después de elegir una hormiga cada ruta y después de elegir 5 hormigas cada ruta.

	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A • 50 pasos	0,1	0,02	33 %	0,1	34 %
B • 45 pasos	0,1	0,022	34 %	0,111	36 %
C • 60 pasos	0,1	0,017	33 %	0,083	31 %
	0,3	0,06		0,29	

Aunque el mismo número de hormigas elijan distintas rutas, vemos que la probabilidad de elección se va decantando por la ruta más corta.

De momento tiene sentido

➤ Depositar feromonas

➤ ¿Cuánta feromona tenemos que depositar?

Por otro lado, si la primera hormiga eligiese el camino A (el segundo mejor), al volver, la distribución de probabilidades se quedaría en un 38% para el segundo mejor camino.

Es posible que la segunda hormiga lo escogiese, pero no tendría por qué hacerlo. No obstante, de elegir el mismo, la tercera hormiga tendría el 41% de probabilidades de repetir lo mismo que las anteriores.

	Inicial	Dep. 1 hormiga	Prob.	Dep. 2 hormigas	Prob.	Dep. 5 hormigas	Prob.
A • 50 pasos	0,1	0,02	38 %	0,04	41 %	0,1	50 %
B • 45 pasos	0,1	0	31 %	0	29 %	0	25 %
C • 60 pasos	0,1	0	31 %	0	29 %	0	25 %
	0,3	0,02		0,04		0,1	

Si lo hiciesen 5 hormigas seguidas, la sexta tendría el 50% de probabilidades de elegir la segunda mejor opción (que es la distribución porcentual que teníamos con el aporte constante y una sola hormiga).

Es decir, el resultado de las hormigas anteriores no condicionan “excesivamente” las decisiones de las hormigas futuras. Por lo que elegir el segundo mejor camino al inicio, no hace que sea la solución definitiva.

Parece que basar la cantidad de feromona en la distancia es la decisión correcta.

➤ Feromona inicial

➤ ¿Condiciona la distancia
el valor de la feromona
inicial?

➤ Feromona inicial

➤ ¿Condiciona la distancia el valor de la feromona inicial?

Analicémoslo

Observemos la evolución de la distribución de probabilidades con: diversas distancias, cuando una hormiga selecciona cada camino y cuando lo hacen 5 hormigas.

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	10	0,1	0,1	33 %	0,5	33 %
B	8	0,1	0,125	37 %	0,625	39 %
C	12	0,1	0,083	30 %	0,417	28 %
		0,3	0,308		1,542	

En este ejemplo tenemos 3 caminos, de entre 8 y 12 pasos entre el hormiguero y la comida.

¿Cómo veis el resultado?

➤ Feromona inicial

➤ ¿Condiciona la distancia el valor de la feromona inicial?

El resultado es exactamente lo que esperamos ver.

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	10	0,1	0,1	33 %	0,5	33 %
B	8	0,1	0,125	37 %	0,625	39 %
C	12	0,1	0,083	30 %	0,417	28 %
		0,3	0,308		1,542	

La primera hormiga no condicionaría en exceso al resto y, en caso de empate, la probabilidad de que la siguiente hormiga elija el camino más corto es mayor.

Funciona perfectamente.

Cambiamos un poco el tamaño.

➤ Feromona inicial

➤ ¿Condiciona la distancia el valor de la feromona inicial?

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	10	0,1	0,1	33 %	0,5	33 %
B	8	0,1	0,125	37 %	0,625	39 %
C	12	0,1	0,083	30 %	0,417	28 %
		0,3	0,308		1,542	

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	50	0,1	0,02	33 %	0,1	34 %
B	45	0,1	0,022	34 %	0,111	36 %
C	60	0,1	0,017	33 %	0,083	31 %
		0,3	0,059		0,294	

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	100	0,1	0,01	33 %	0,05	33 %
B	90	0,1	0,011	34 %	0,056	35 %
C	110	0,1	0,009	33 %	0,045	32 %
		0,3	0,03		0,151	

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	500	0,1	0,002	33 %	0,01	33 %
B	400	0,1	0,003	34 %	0,013	34 %
C	600	0,1	0,002	33 %	0,008	33 %
		0,3	0,006		0,031	

¿Y ahora qué opinión tenemos?

➤ Feromona inicial

➤ ¿Condiciona la distancia el valor de la feromona inicial?

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	10	0,1	0,1	33 %	0,5	33 %
B	8	0,1	0,125	37 %	0,625	39 %
C	12	0,1	0,083	30 %	0,417	28 %
		0,3	0,308		1,542	

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	50	0,1	0,02	33 %	0,1	34 %
B	45	0,1	0,022	34 %	0,111	36 %
C	60	0,1	0,017	33 %	0,083	31 %
		0,3	0,059		0,294	

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	100	0,1	0,01	33 %	0,05	33 %
B	90	0,1	0,011	34 %	0,056	35 %
C	110	0,1	0,009	33 %	0,045	32 %
		0,3	0,03		0,151	

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	500	0,1	0,002	33 %	0,01	33 %
B	400	0,1	0,003	34 %	0,013	34 %
C	600	0,1	0,002	33 %	0,008	33 %
		0,3	0,006		0,031	

Podemos ver que el tamaño de laberinto sí que afecta al valor que debería tener la feromona inicial.

Para laberintos “pequeños” el valor de 0,1 es adecuado.

Para laberintos “medianos y grandes”, después de que 5 hormigas hayan pasado por cada uno de los caminos, la elección de la sexta hormiga se realiza de nuevo al azar.

➤ Feromona inicial

➤ ¿Condiciona la distancia el valor de la feromona inicial?

Ok, ajustemos un poco los valores.

Para laberintos pequeños feromona inicial de 0,1 y para grandes de 0,01

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	10	0,1	0,1	33 %	0,5	33 %
B	8	0,1	0,125	37 %	0,625	39 %
C	12	0,1	0,083	30 %	0,417	28 %
		0,3	0,308		1,542	

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	100	0,1	0,01	33 %	0,05	33 %
B	90	0,1	0,011	35 %	0,056	36 %
C	110	0,1	0,009	32 %	0,045	31 %
		0,3	0,03		0,151	

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	50	0,1	0,02	33 %	0,1	34 %
B	45	0,1	0,022	34 %	0,111	36 %
C	60	0,1	0,017	33 %	0,083	31 %
		0,3	0,059		0,294	

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	500	0,01	0,002	33 %	0,01	33 %
B	400	0,01	0,003	35 %	0,013	37 %
C	600	0,01	0,002	32 %	0,008	30 %
		0,03	0,006		0,031	

¿Funciona?

¿Qué opinión tenemos?

➤ Feromona inicial

➤ ¿Funciona?, ¿Qué opinión tenemos?

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	10	0,1	0,1	33 %	0,5	33 %
B	8	0,1	0,125	37 %	0,625	39 %
C	12	0,1	0,083	30 %	0,417	28 %
		0,3	0,308		1,542	

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	50	0,1	0,02	33 %	0,1	34 %
B	45	0,1	0,022	34 %	0,111	36 %
C	60	0,1	0,017	33 %	0,083	31 %
		0,3	0,059		0,294	

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	100	0,01	0,01	33 %	0,05	33 %
B	90	0,01	0,011	35 %	0,056	36 %
C	110	0,01	0,009	32 %	0,045	31 %
		0,03	0,03		0,151	

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	500	0,01	0,002	33 %	0,01	33 %
B	400	0,01	0,003	35 %	0,013	37 %
C	600	0,01	0,002	32 %	0,008	30 %
		0,03	0,006		0,031	

Para laberintos grandes, un valor de feromona inicial de 0,01 es mucho más adecuado. Sin embargo, si el laberinto fuera “gigante” el valor de 0,01 tampoco nos valdría.

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	10.000	0,01	0	33 %	0,001	33 %
B	8.500	0,01	0	33 %	0,001	34 %
C	11.000	0,01	0	33 %	0	33 %
		0,3	0		0,002	

	Pasos	Inicial	Dep. 1 hormiga	Prob.	Dep. 5 hormigas	Prob.
A	10.000	0,0001	0	33 %	0,001	33 %
B	8.500	0,0001	0	36 %	0,001	37 %
C	11.000	0,0001	0	31 %	0	30 %
		0	0		0,002	

➤ Feromona inicial

➤ Hay que analizar la relación entre tamaño del laberinto y feromona inicial.

Una posible solución sería que la feromona inicial se recalcule la primera vez que una hormiga encuentre la comida y vuelva al hormiguero. Entonces tendremos un dato real de distancia que nos permitirá saber la proporción adecuada de feromona que debe constar para que la distribución de probabilidad tenga sentido. $FI = 1 / \text{pasos_hasta_comida}$.



Otra posible solución sería relacionar la feromona inicial con el tamaño del laberinto. Por ejemplo: $\text{feromona_inicial} = 1 / (\text{sum}(\text{lados del laberinto})/2)$

La relación entre la feromona inicial y la feromona depositada (en función a la distancia) debe estar en perfecto equilibrio o el enjambre no funcionará.

¿Qué solución pensáis que es mejor?



➤ Feromona inicial

➤ ¿Qué solución pensáis que es mejor?

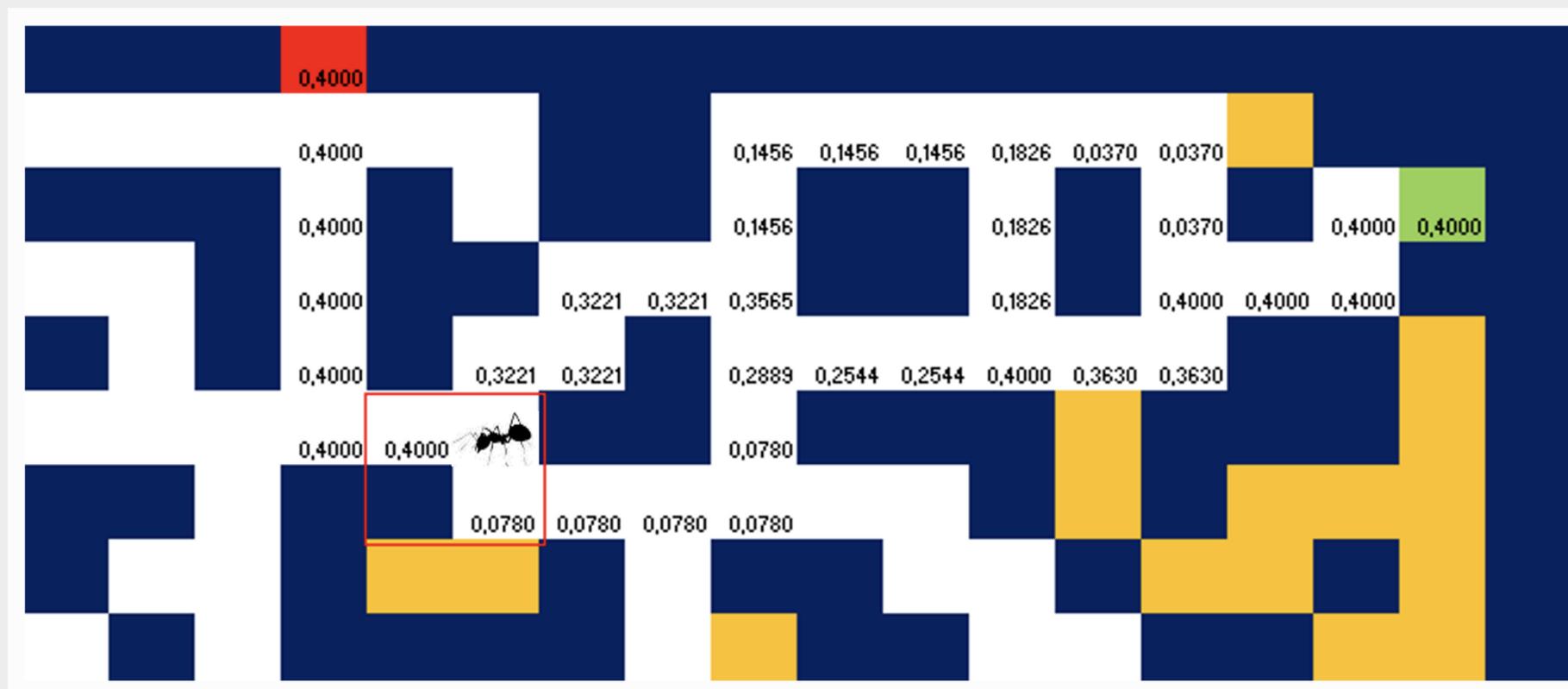
Desde mi punto de vista la segunda:
relacionar la feromona inicial con el tamaño del laberinto.

Pensad que una hormiga puede haber encontrado el camino hasta la comida habiendo dado muchísimas vueltas. Incluso habiendo podado la solución.

Referir la feromona inicial a ese “camino aleatorio”, puede no ser la mejor aproximación a realizar.

➤ Feromona inicial

➤ Ejemplo visual real



El gráfico muestra un ejemplo real de cómo se deposita la feromona según las hormigas realizan un camino u otro hasta la comida.

Si analizamos la toma de decisión de la hormiga que se encuentra en el cuadrado marcado en rojo vemos que tiene $0,4 / (0,4 + 0,078) = 83,68\%$ de probabilidad de ir hacia la izquierda (camino que lleva a la comida), contra $16,31\%$ de probabilidad de irse hacia abajo a explorar un camino que no lleva a la comida (al menos directamente).

➤ Evaporar feromonas

- Los rastros de feromona se podrían reducir en un valor constante. Imitando el proceso de evaporación global que existe en la vida real.

La finalidad de este proceso sería “penalizar” en el tiempo a los caminos que, al inicio de la exploración, fueron seleccionados y que después han demostrado no ser los mejores.

De querer utilizar este sistema, se ha de tener en cuenta que el proceso de evaporación debe realizarse transcurridos

X movimientos o X tiempo. Si lo hiciéramos con cada movimiento se evaporarían rápidamente las feromonas, sin tener tiempo el resto de hormigas de utilizar esta información.

La teoría indica que este proceso se ha mostrado poco eficiente en su aportación de búsqueda del camino óptimo, teniendo además un coste computacional resulta elevado.

[Analicémoslo](#)

➤ Evaporar feromonas

➤ Para ello, vamos a generar dos laberintos de 50 x 50

En uno de ellos vamos a introducir un proceso de evaporación y en el otro no.

➤ Posteriormente, graficaremos las matrices de feromonas sobre los laberintos para analizar si existen diferencias significativas.

De no existir, el proceso de evaporación resultaría redundante y, por lo tanto, ineficiente.

↗ Evaporar feromonas

↗ Con proceso de evaporación: 10% cada vez que se encuentra la comida



↗ Evaporar feromonas

↗ Sin proceso de evaporación



➤ Evaporar feromonas

➤ ¿Conclusiones?



Con evaporación



Sin evaporación

➤ Evaporar feromonas

➤ ¿Conclusiones?

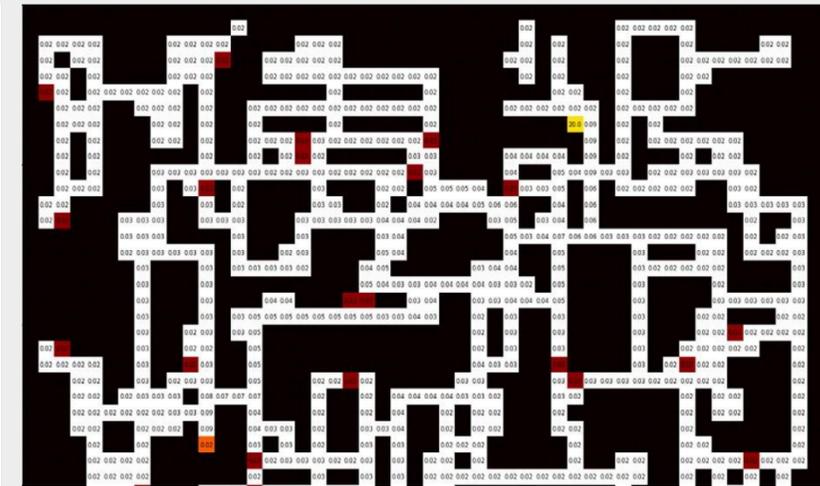
El proceso de evaporación penaliza, progresivamente, aquellos caminos que no son utilizados. Por lo que las hormigas, pasado un tiempo, tienen diversos caminos donde elegir, pero solo aquellos que tienen trazas de feromona.

Igual que en la vida real.

En mi opinión, el proceso de evaporación es muy útil.



Con evaporación



Sin evaporación

➤ Depositar y evaporar feromonas

➤ El cuarto ejercicio consiste en poner en práctica el depósito de feromonas y la evaporación de las mismas.



Abre el ejercicio:
4 Depositar feromonas.ipynb

➤ Hormiga fiel



El objetivo de la hormiga fiel es mostrar la mejor ruta posible, en base al conocimiento global del hormiguero. Es decir, la solución al problema.



No tiene porqué limitarse a mostrar la ruta encontrada más corta. Puede utilizar el conocimiento global de la colmena para combinar caminos, y encontrar así la mejor ruta posible.

➤ Hormiga fiel

- Este proceso trabaja por ciclos. Un ciclo puede ser cada X tiempo o cada X veces que se alcance la comida. Un ciclo podría ser cada 10 hormigas que alcancen la comida. Activarla cada vez que se alcance la comida, o únicamente mostrar la ruta de la hormiga fiel al finalizar el algoritmo.

Al final de cada ciclo, la hormiga viaja hasta la comida para evaluar las soluciones encontradas por las distintas hormigas de la colonia.

Las elecciones de su viaje pueden basarse en:

- 1. Los caminos que tengan mayor aporte de feromona: Eligiendo siempre aquel que tenga más probabilidad de ser elegido por cualquier hormiga.

¿Os parece buena idea?

➤ Hormiga fiel

- Este proceso trabaja por ciclos. Un ciclo puede ser cada X tiempo o cada X veces que se alcance la comida. Un ciclo podría ser cada 10 hormigas que alcancen la comida. Activarla cada vez que se alcance la comida, o únicamente mostrar la ruta de la hormiga fiel al finalizar el algoritmo.

Al final de cada ciclo, la hormiga viaja hasta la comida para evaluar las soluciones encontradas por las distintas hormigas de la colonia.

Las elecciones de su viaje pueden basarse en:

- 1. Los caminos que tengan mayor aporte de feromona: Eligiendo siempre aquel que tenga más probabilidad de ser elegido por cualquier hormiga.

Descartamos esta opción porque la hormiga fiel no siempre recorrería el mejor camino, si no el más probable. Y nunca mejoraría los resultados obtenidos hasta ese instante.

- 2. El camino que tenga menor distancia con respecto a la comida. Esta opción permite a la hormiga fiel elegir, como mínimo, el mejor camino que hasta ahora se ha encontrado y, le permite evaluar en cada cruce la distancia real existente, pudiendo combinar caminos, mejorando el mejor camino encontrado hasta el momento.

Para poder llevar a cabo esta aproximación, tendremos que almacenar, en cada punto del laberinto, distancia mínima descubierta hasta el momento.

➤ Hormiga fiel

- Las feromonas, por tanto, sirven para las hormigas exploradoras. Las permiten decidir el camino en función de los niveles de feromonas existentes.

La distancia mínima hasta la comida sirve para la hormiga fiel. Esta información le permite recorrer la menor ruta descubierta hasta el momento e, incluso, mejorarla. Combinando rutas ya descubiertas.



➤ Finalización del algoritmo

- ¿Cuándo podemos dar por finalizado el algoritmo?
- ¿Qué consecuencias tendría pararlo demasiado pronto o demasiado tarde?

➤ Finalización del algoritmo

- ¿Cuándo podemos dar por finalizado el algoritmo?
- ¿Qué consecuencias tendría pararlo demasiado pronto o demasiado tarde?

El objetivo es encontrar el camino óptimo.

Para ello utilizamos el proceso de depósito de feromonas + evaporación + hormiga fiel

El depósito de feromonas + evaporación suceden cada vez que una hormiga exploradora encuentra la comida.

La combinación de caminos de la hormiga fiel se activará cuando demos la exploración por finalizada.

Lo que debemos determinar es el número de veces que las hormigas deben encontrar la comida para que el proceso de depósito de feromonas y evaporación hayan cumplido su función.

La solución es evidente

➤ Finalización del algoritmo

- ¿Cuándo podemos dar por finalizado el algoritmo?
- ¿Qué consecuencias tendría pararlo demasiado pronto o demasiado tarde?

El objetivo es encontrar el camino óptimo.

Para ello utilizamos el proceso de depósito de feromonas + evaporación + hormiga fiel

El depósito de feromonas + evaporación suceden cada vez que una hormiga exploradora encuentra la comida.

La combinación de caminos de la hormiga fiel se activará cuando demos la exploración por finalizada.

Lo que debemos determinar es el número de veces que las hormigas deben encontrar la comida para que el proceso de depósito de feromonas y evaporación hayan cumplido su función.

La solución es evidente

Debemos vincular el número de veces a encontrar la comida con el tamaño del laberinto

Una posible solución sería: repetir mientras el $n^{\circ}_{\text{veces}} < \text{lado} * \text{lado} / 100$

➤ Finalización del algoritmo

➤ El quinto ejercicio consiste en dotar de habilidades a la hormiga fiel.

 Abre el ejercicio:
5 Hormiga fiel.ipynb

Versión Pro

-  El sexto ejercicio consiste en poner todo el código dentro de clases, para que el algoritmo se ejecute simplemente instanciando la clase laberinto.



Abre el ejercicio:
[6 Ant Colony Optimization.ipynb](#)

➤ Modificando el algoritmo original

➤ ¿Qué mejoras podríamos hacer?

➤ Modificando el algoritmo original

➤ ¿Qué mejoras podríamos hacer?

¿Y si diésemos a las hormigas la capacidad de modificar el entorno?

Con el objetivo de solventar los problemas que tenemos perfectamente identificados

- Plazas
- Rotondas



Solventando el problema de las plazas

¿Cómo podríamos hacer que una hormiga detectara que está en una plaza?

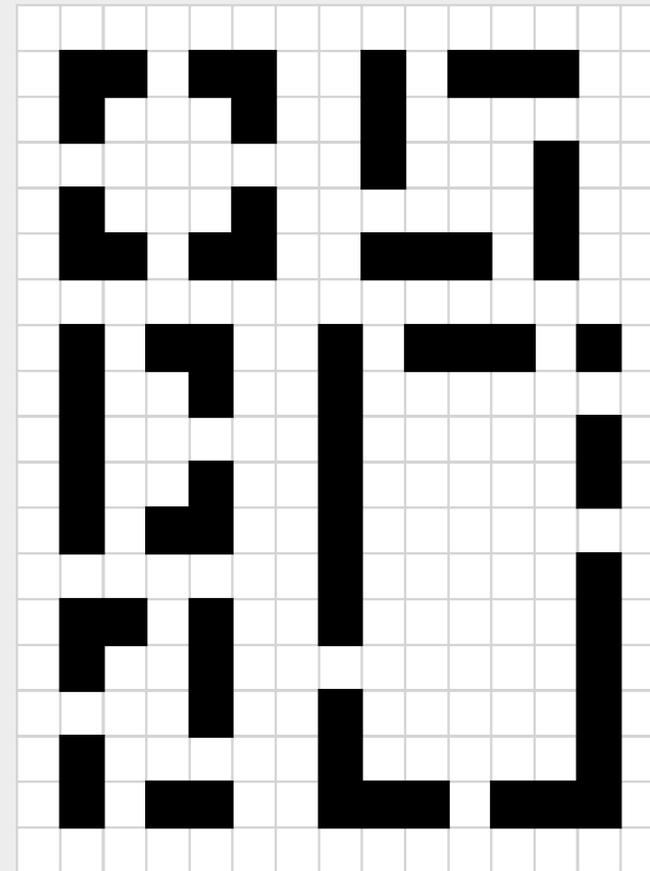
¿Qué habilidad podríamos dar a las hormigas para solventar el problema?

Y lo primero de todo. ¿Qué es una plaza?

➔ Solventando el problema de las plazas

Tenemos multitud de ejemplos de qué es una plaza

Lo que tenemos que conseguir, es extrapolar unas sencillas reglas para que la hormiga pueda detectar que se encuentra en una



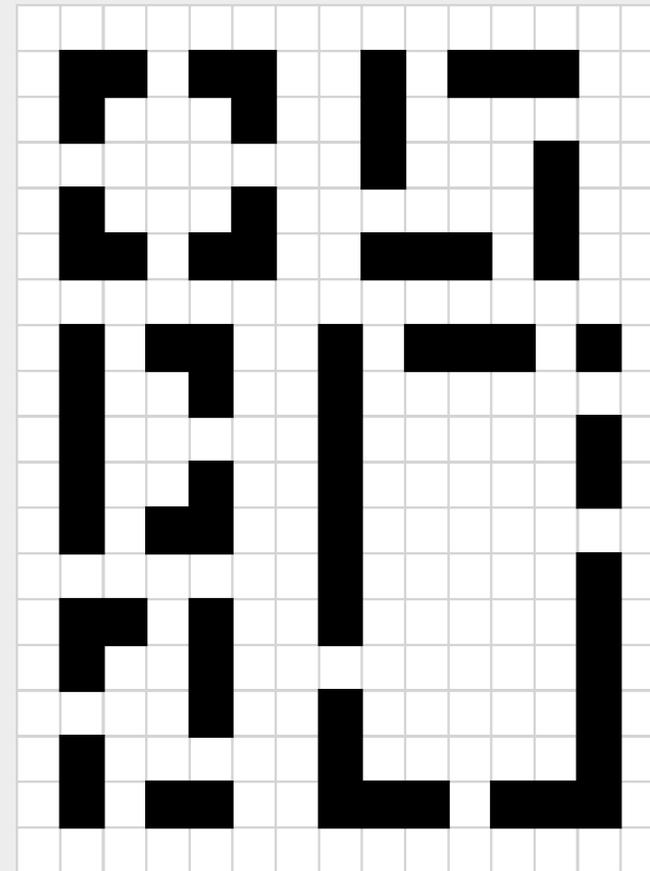
➤ Solventando el problema de las plazas

Una plaza es todo aquello que no sea un pasillo

Es la definición más general que podemos hacer

¿Qué habilidad podemos dar a las hormigas para solventar este problema?

La solución es evidente



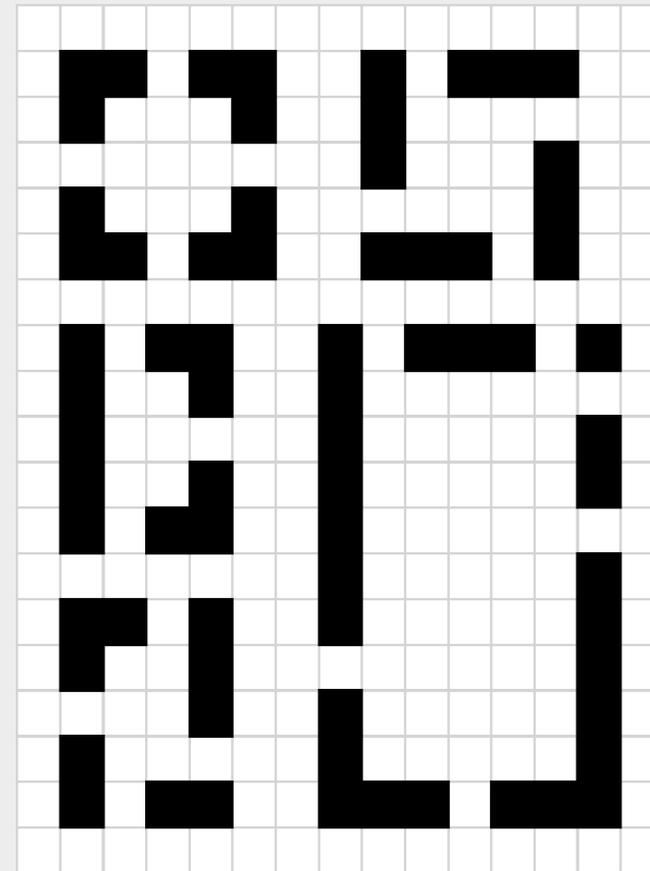
➔ Solventando el problema de las plazas

¿Qué habilidad podemos dar a las hormigas para solventar este problema?

La solución es evidente

Al igual que hicimos con los caminos que no llevaban a ninguna parte, podemos hacer que las hormigas marquen casillas como inútiles: levanten muros

¿Pero cómo? Trata de deducir las reglas exactas



➤ Solventando el problema de las plazas

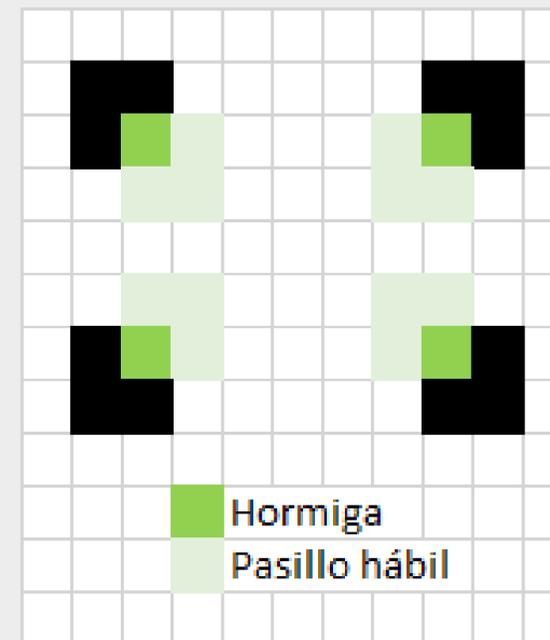
Esquinas

Lo primero que podríamos hacer es ayudar a las hormigas a detectar que se encuentran en una esquina

Confirmar que tendrían un camino útil, en caso de que levantasen un muro

Levantar el muro y retroceder a la casilla anterior

Para ello, podemos usar plantillas



Si la hormiga se encuentra en cualquiera de estas situaciones

Podemos determinar que se encuentra en una esquina

Y que, por lo tanto, puede anular la casilla

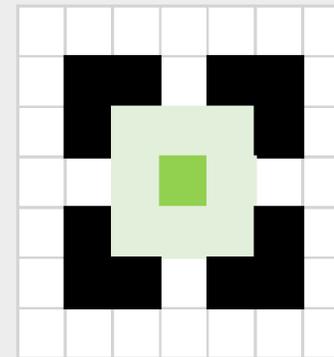
➤ Solventando el problema de las plazas

Centro

Otra plantilla que podríamos usar es la del centro de la plaza

Confirmamos que la hormiga no tiene nada alrededor

Por lo que puede levantar el muro y retroceder a la casilla anterior



Si la hormiga se encuentra en esta situación

Se encuentra en el centro de una plaza

Puede anular la casilla

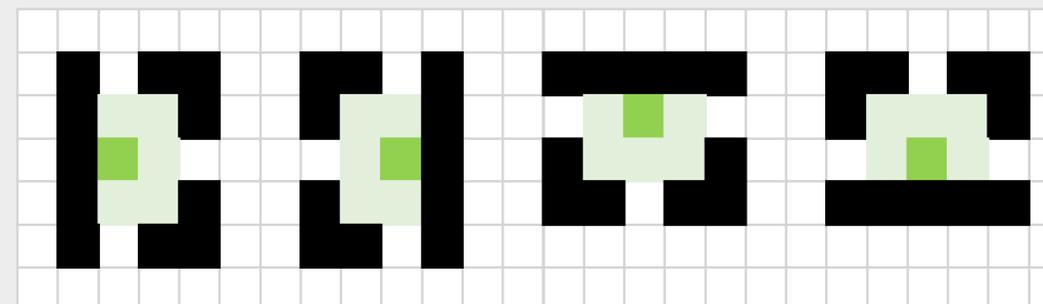
➤ Solventando el problema de las plazas

Pasillos anchos

En estas situaciones, debemos determinar que el muro se encuentra únicamente a uno de los lados

Y confirmar que, en el resto, no tiene nada alrededor

Por lo que puede levantar el muro y retroceder a la casilla anterior



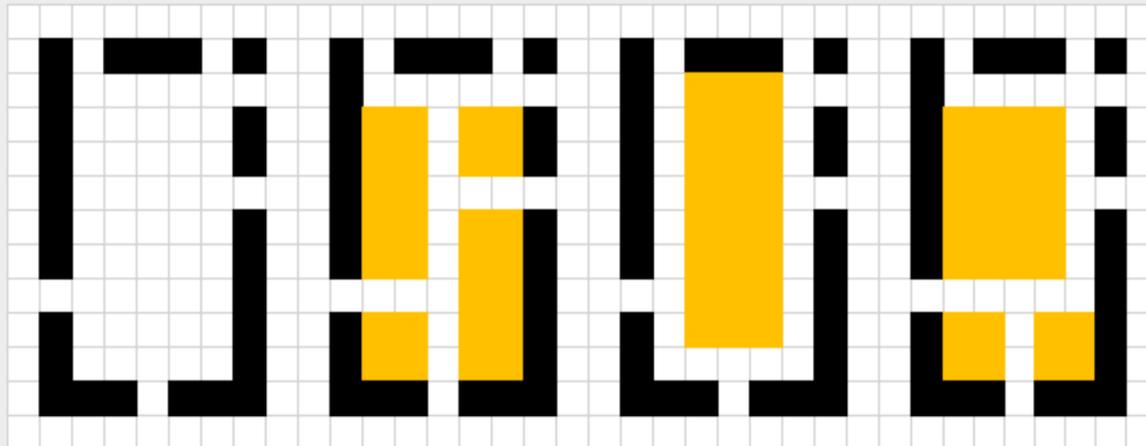
Si la hormiga se encuentra en cualquiera de estas situaciones

Puede anular la casilla

➔ Solventando el problema de las plazas

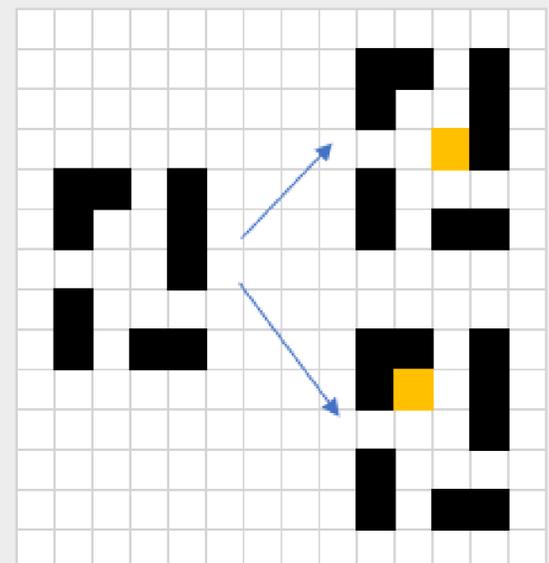
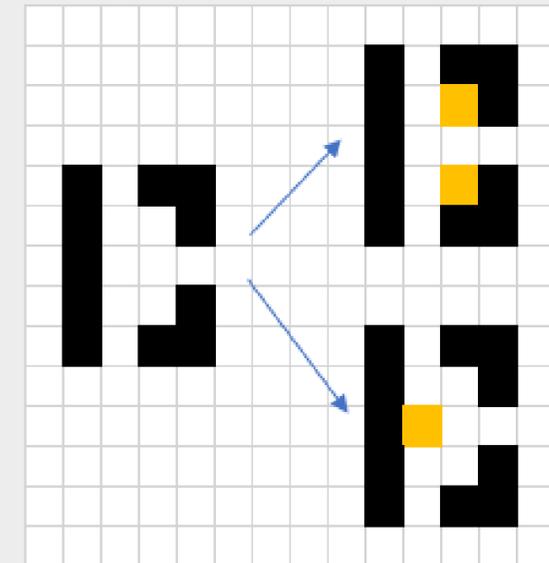
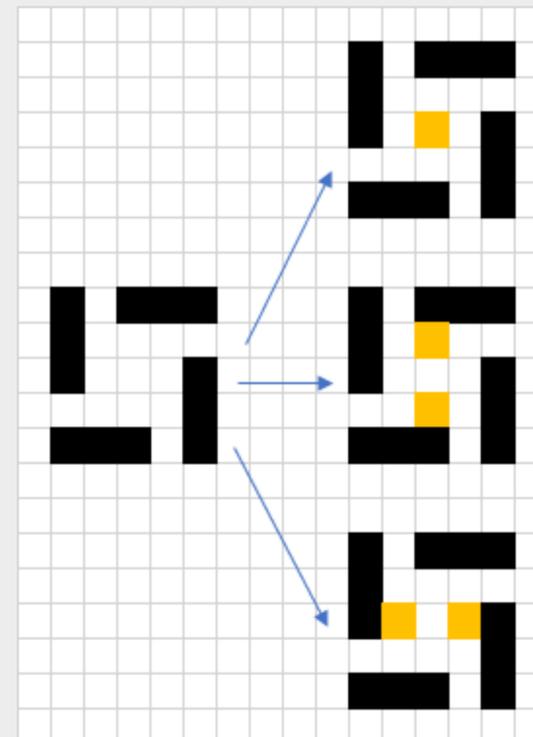
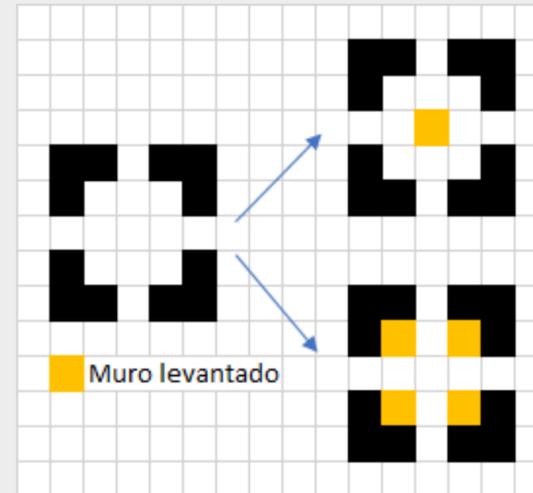
Confirmación de las reglas

Analicemos los distintos ejemplos de plazas que hemos puesto hace poco, y veamos qué podría pasar, en función de la ruta que escoja la hormiga



Partiendo del estado inicial de la izquierda

Podríamos obtener cualquier resultado de la derecha



➤ Solventando el problema de las plazas

Confirmación de las reglas



➤ Solventando el problema de las plazas

Confirmación de las reglas

Laberinto original



Laberinto resuelto



➤ Solventando el problema de las plazas

- El séptimo ejercicio consiste en dotar, a las hormigas, de la capacidad de detectar plazas y levantar muros.



Abre el ejercicio:
[7 plazas.ipynb](#)

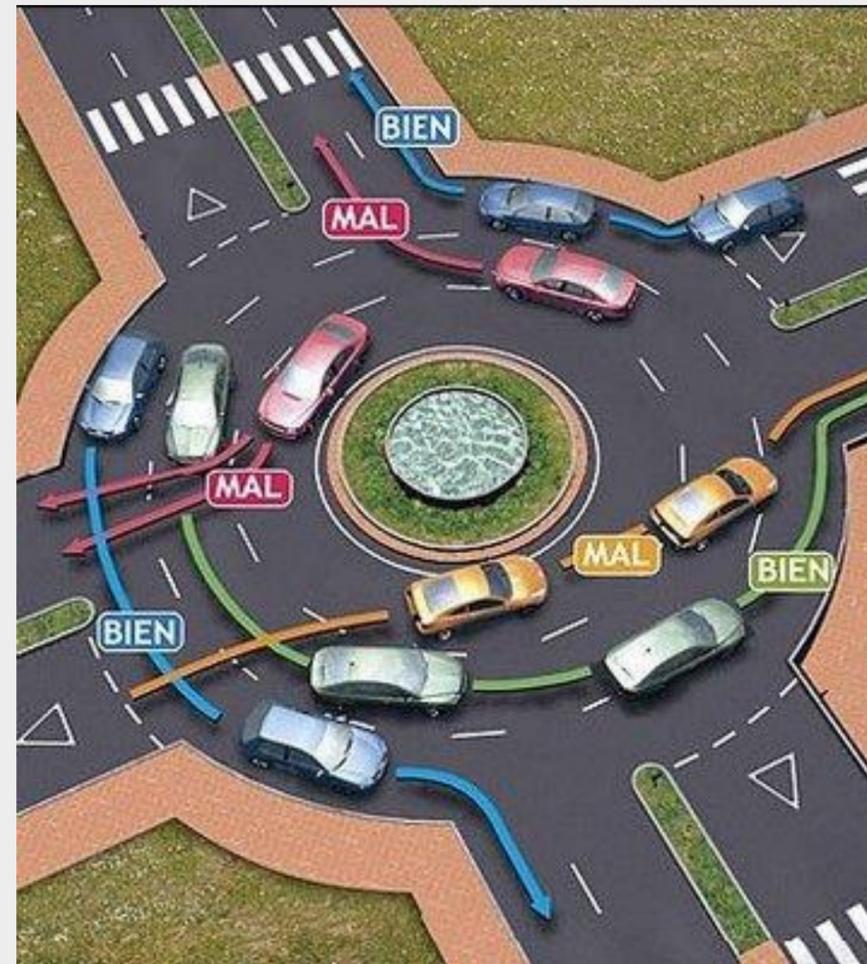
➤ Solventando el problema de las rotondas

¿Cómo podríamos hacer que una hormiga detectara que está en una rotonda?

¿Cómo podríamos anular una rotonda?

¿Qué implicaciones tendría anular una rotonda?

Una rotonda no es, ni mucho menos, tan trivial como una plaza.



Solventando el problema de las rotondas

¿Cómo podríamos hacer que una hormiga detectara que está en una rotonda?

¿Cómo podríamos anular una rotonda?

¿Qué implicaciones tendría anular una rotonda?

Detectar una rotonda es fácil. Tan solo tenemos que comprobar si la hormiga ya ha estado en la casilla a la que acaba de desplazarse.

Si es así, ha completado la rotonda.

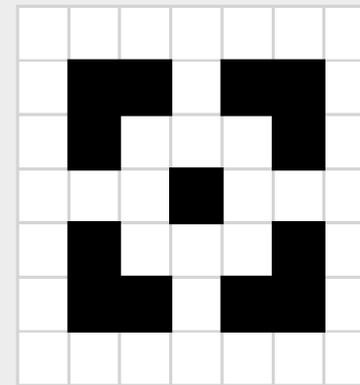
Anular una rotonda también parece fácil: hay que poner un muro.

El cómo, y dónde poner el muro, es lo que tenemos que pensar con cuidado. Porque tiene serias implicaciones.

➔ Solventando el problema de las rotondas

¿Qué tipos de rotondas podríamos encontrarnos?

Los ejemplos son casi infinitos, pero tratemos de poner algunos ejemplos



Esto es una rotonda

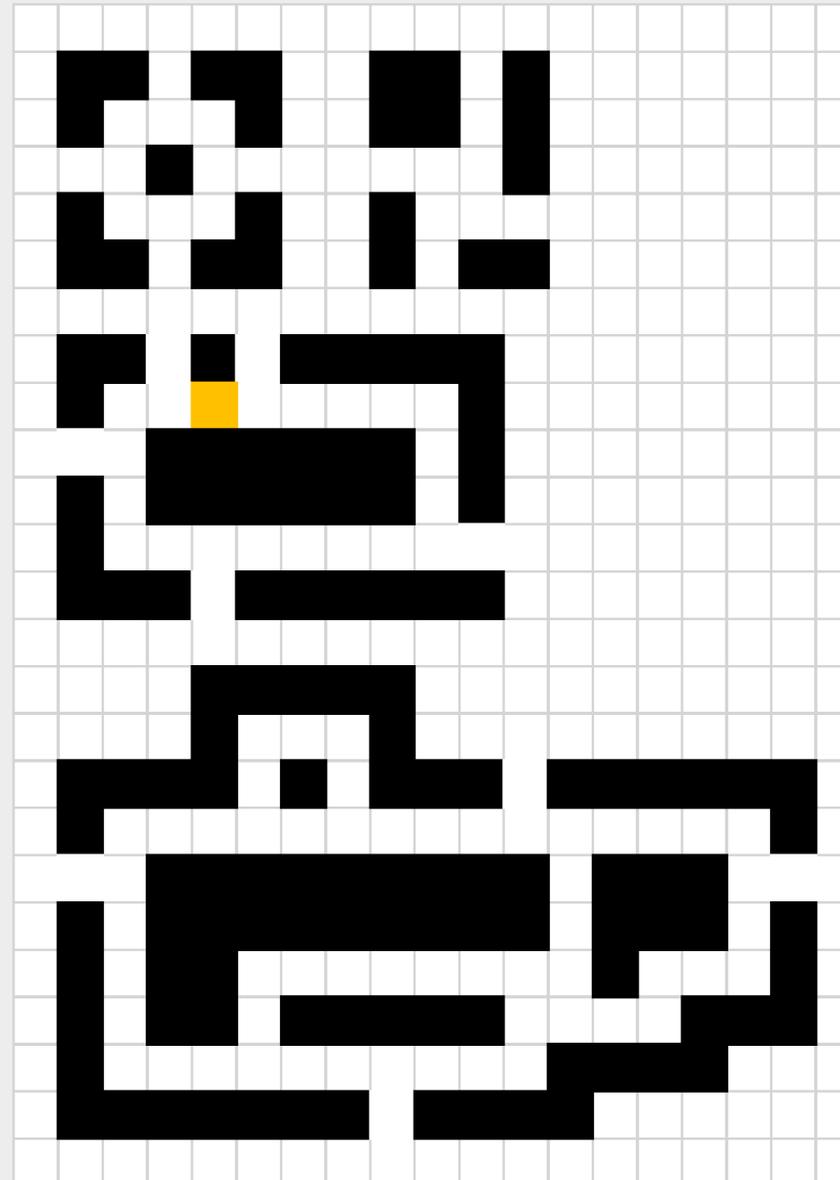
¿Cómo la cortas?, ¿dónde pones el muro?

¿Qué reglas debemos poner para que la hormiga pueda levantar un muro?

➤ Solventando el problema de las rotondas

¿Qué tipos de rotondas podríamos encontrarnos?

Podemos encontrarnos con infinidad de casos



Rotondas con múltiples salidas

Rotondas sin centro, en este caso concreto: irresolubles

Rotondas donde podríamos anular al propio hormiguero, o cortarle una "arteria de salida"

Rotondas de rotondas, donde las rutas para recorrerlas no son únicamente una o dos

¿Cómo cortamos las rotondas?

➤ Solventando el problema de las rotondas

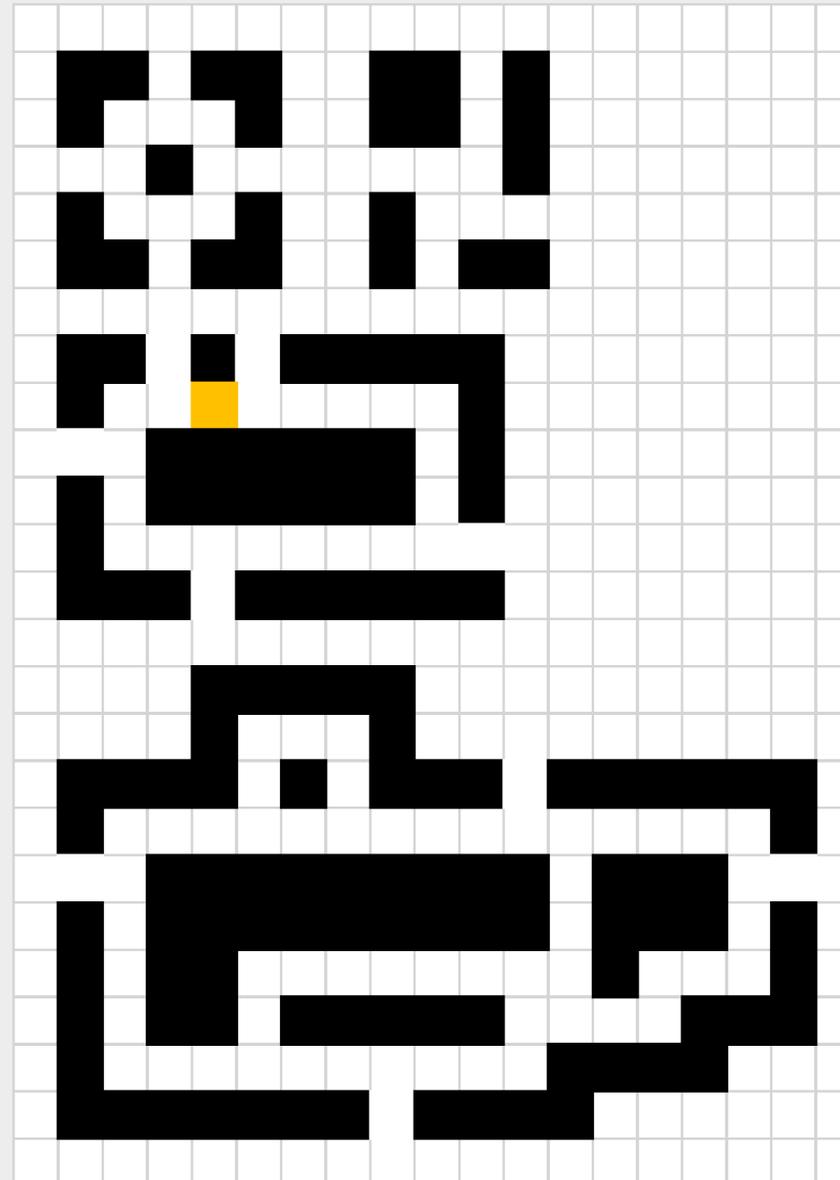
Detectar que hemos recorrido una rotonda es fácil:

¿Hemos vuelto a una casilla en la que ya hemos estado antes?

Hemos recorrido una rotonda

La rotonda puede estar compuesta de 4 casillas o de 400

Queremos cortar la rotonda poniendo un muro.



¿Qué reglas concretas debemos indicar a las hormigas?

➤ Solventando el problema de las rotondas

Hemos completado una rotonda

Podríamos levantar un muro en la casilla anterior a completar la rotonda: cortándola por la “derecha”

O podríamos poner un muro en la segunda casilla de la rotonda: cortándola por la “izquierda”

Después debemos anular el bucle recorrido por la hormiga, para que pueda seguir su camino

Podemos hacer esto, siempre y cuando ¿qué?



➔ Solventando el problema de las rotondas

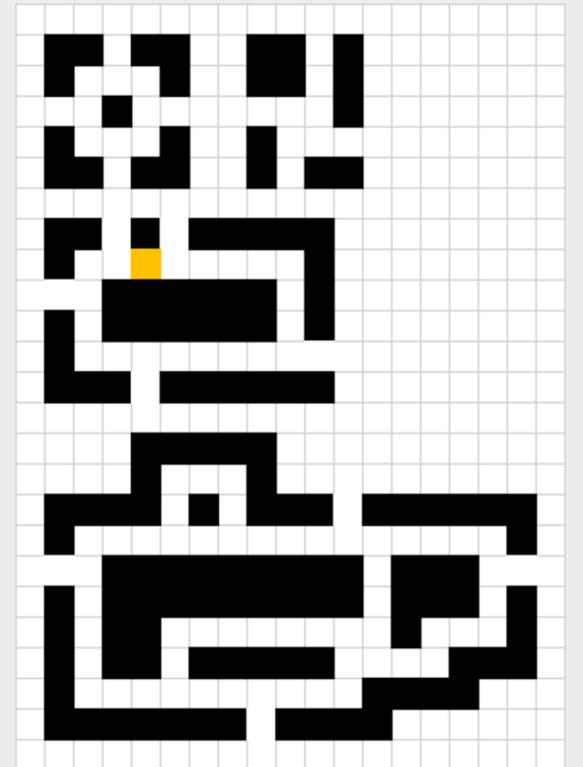
Hemos completado una rotonda

Podemos levantar un muro, siempre y cuando:

- No estemos bloqueando hormiguero
- La casilla donde voy a levantar el muro sea un pasillo (tenga solo dos movimientos posibles). Queremos evitar bloquear salidas válidas.

Parece coherente, y son reglas sencillas de implementar

¿Algo más que debemos tener en cuenta?



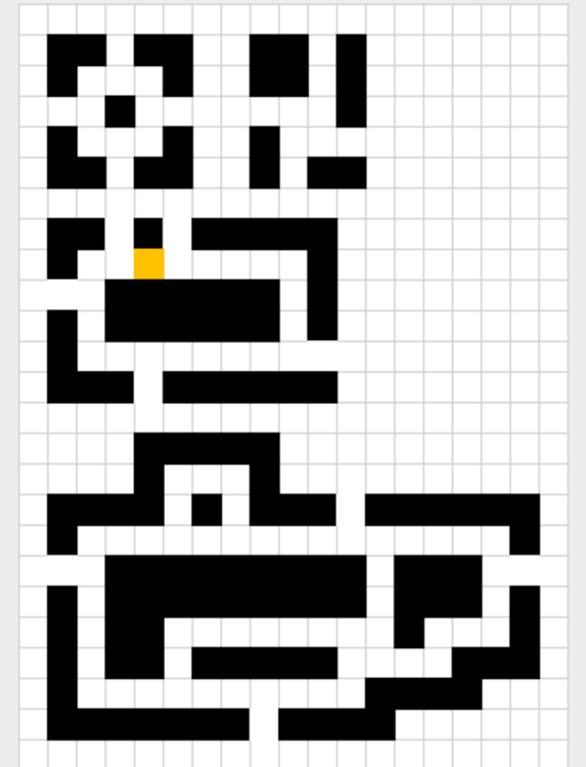
➤ Solventando el problema de las rotondas

Podemos levantar un muro, siempre y cuando:

- No estemos bloqueando hormiguero
- La casilla donde voy a levantar el muro sea un pasillo (tenga solo dos movimientos posibles). Queremos evitar bloquear salidas válidas.

¿Y si otra hormiga ya ha pasado por la celda que vamos a tapiar?

¿Qué consecuencias tendría levantar un muro en esa celda?

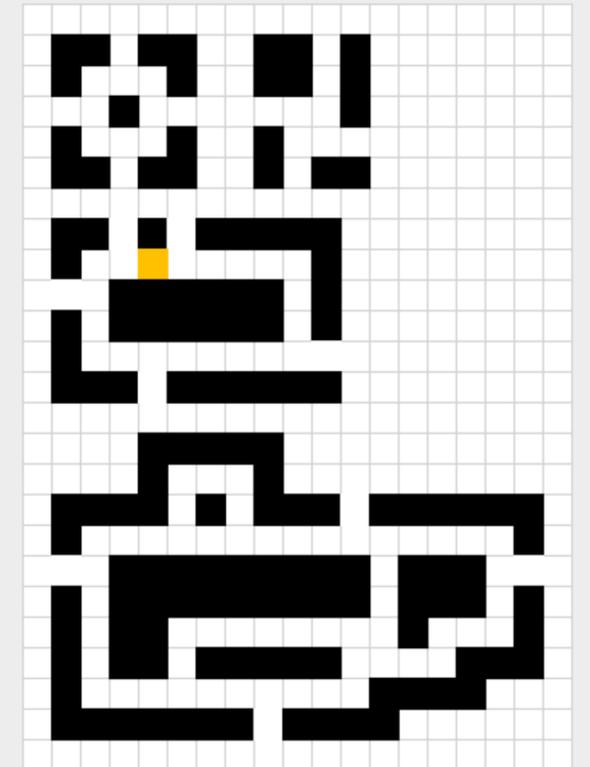


➤ Solventando el problema de las rotondas

¿Y si otra hormiga ya ha pasado por la celda que vamos a tapiar?
¿Qué consecuencias tendría levantar un muro en esa celda?

- Podría ser que la hormiga esté completando la rotonda con un recorrido distinto. En ese caso, al completarla, levantaría un muro en otro sitio. Posiblemente invalidando completamente la ruta, o rutas, para todo el hormiguero.
- Podría ser que llegase a la comida, sin completar ninguna rotonda, pero que al volver no pudiese llegar al hormiguero, ya que se encontraría un muro en su camino.

Ok, hay que tener esto en cuenta. ¿Qué podemos hacer?



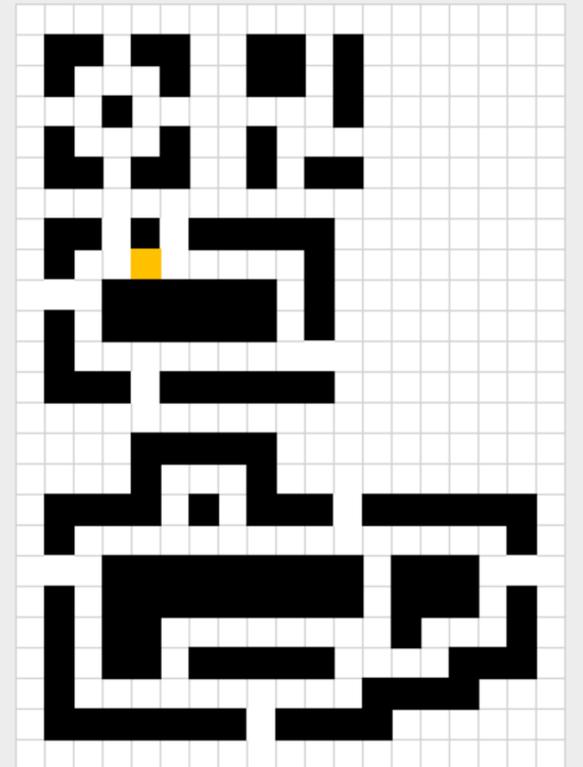
➤ Solventando el problema de las rotondas

Ok, hay que tener esto en cuenta. ¿Qué podemos hacer?

Debemos comprobar, con el movimiento de cada hormiga, si la ruta que ya ha recorrido sigue siendo válida. O si, por el contrario, ahora hay un muro en su camino.

En ese caso, debemos matar a esa hormiga, y que surja otra en el hormiguero.

¿Algo más que debemos tener en cuenta?

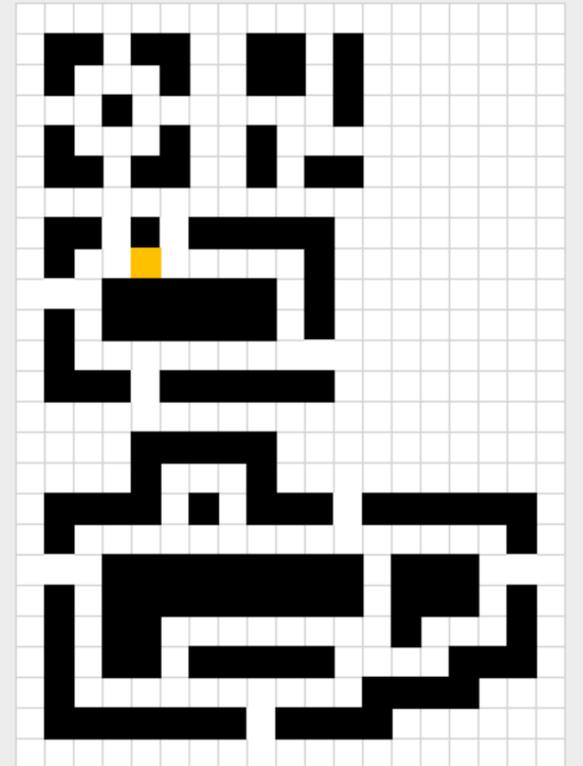


➔ Solventando el problema de las rotondas

¿Algo más que debemos tener en cuenta?

¿Y si la celda que vamos a anular forma ya parte de un camino que lleva a la comida? Un camino que tiene información para la hormiga fiel.

¿Qué hacemos en ese caso?

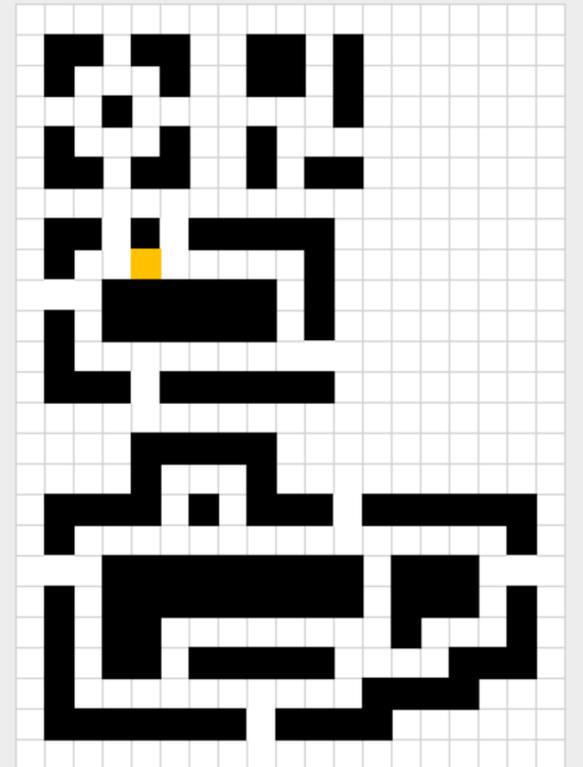


➤ Solventando el problema de las rotondas

Vamos a levantar un muro siempre y cuando:

- No estemos bloqueando el hormiguero
- La casilla donde voy a levantar el muro es un pasillo
- Eliminar a todas las hormigas que hayan pasado por allí
- La casilla a bloquear no contenga información para la hormiga fiel

Si se cumple todo lo anterior, podemos levantar un muro



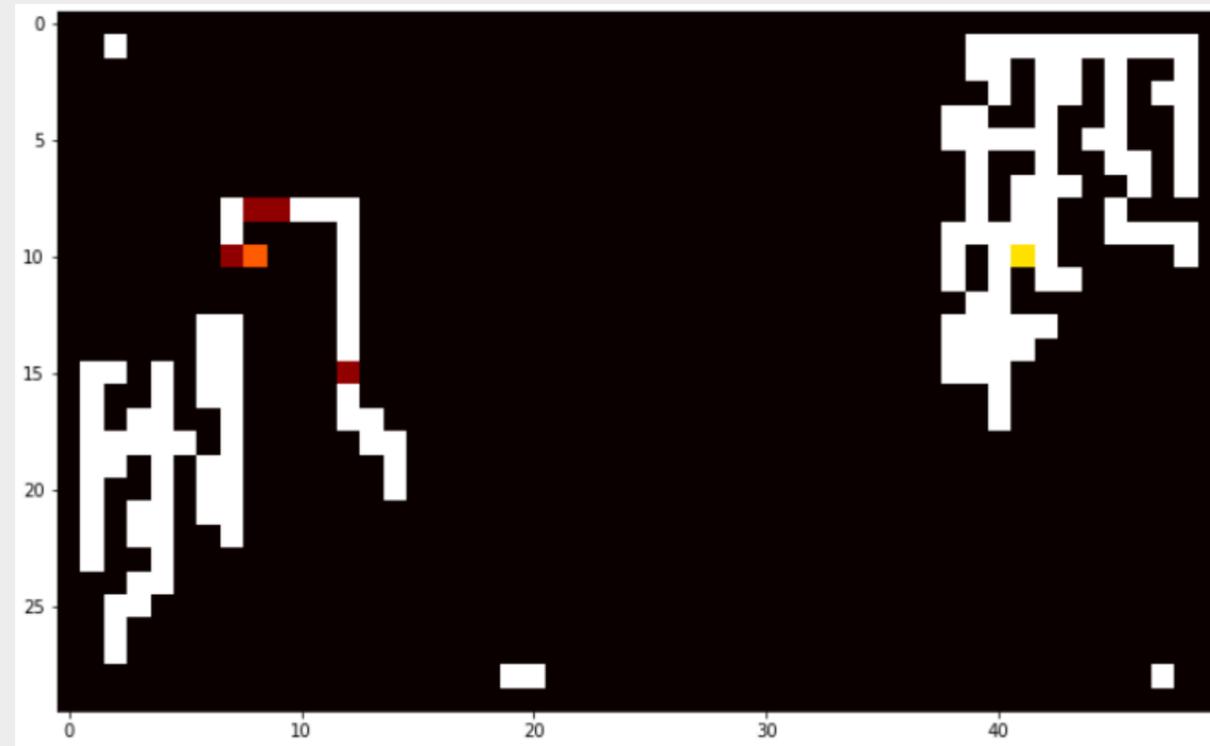
➤ Solventando el problema de las rotondas

Confirmación de las reglas

Ok, hemos hecho todo lo anterior

El resultado es el siguiente:

Hemos ahogado el hormiguero



¿Qué hemos hecho mal???

Vamos a levantar un muro siempre y cuando:

- No estemos bloqueando el hormiguero
- La casilla donde voy a levantar el muro es un pasillo
- Eliminemos a todas las hormigas que hayan pasado por allí
- La casilla a bloquear no contenga información para la hormiga fiel

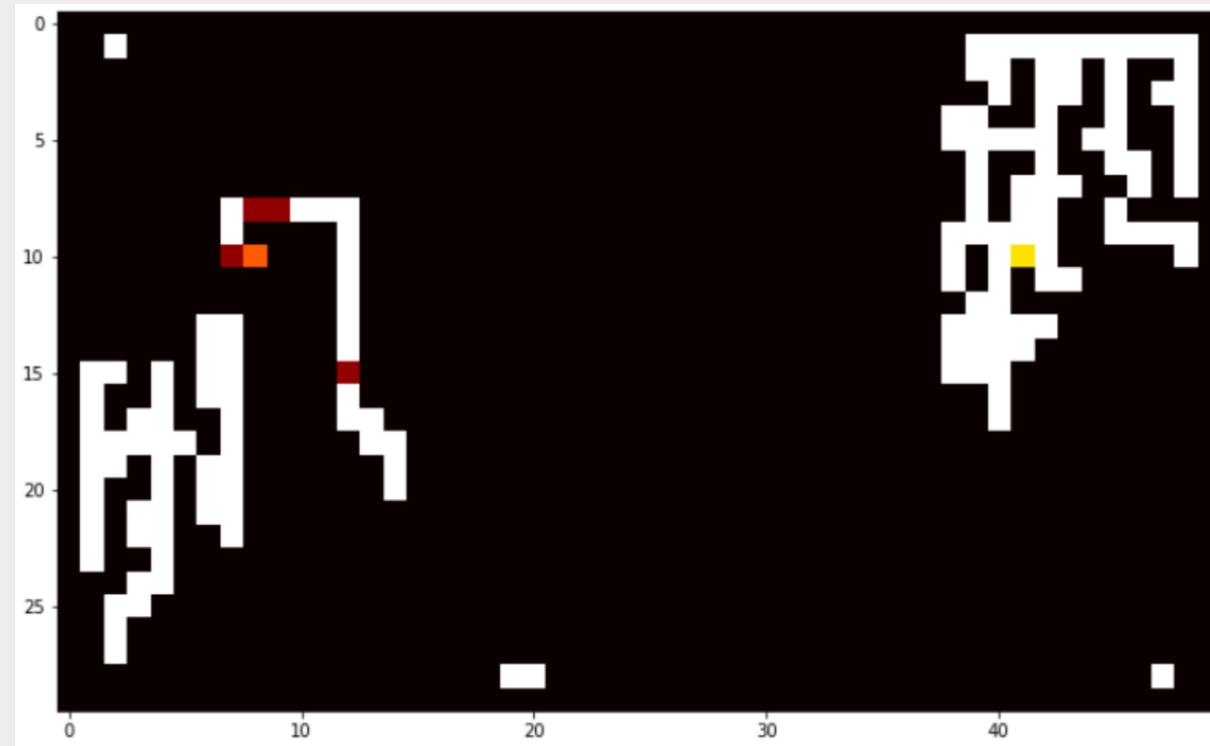
➤ Solventando el problema de las rotondas

Confirmación de las reglas

¿Qué hemos hecho mal?

Si confirmo que la casilla a bloquear tiene solo 2 posibilidades, que es un pasillo, y nos encontramos en una rotonda sin centro, el algoritmo tiende a colapsar sobre sí mismo.

Ahogando al hormiguero



Vamos a levantar un muro siempre y cuando:

- No estemos bloqueando el hormiguero
- **La casilla donde voy a levantar el muro es un pasillo**
- Eliminar a todas las hormigas que hayan pasado por allí
- La casilla a bloquear no contenga información para la hormiga fiel

Quizás deberíamos proponer otro conjunto de reglas
¿Alguna propuesta?

➤ Solventando el problema de las rotondas

Nueva propuesta: vamos a levantar un muro siempre y cuando:

Vamos a extraer la 1ª y última celda de la rotonda para determinar cual debemos bloquear.

Requisitos generales

- No estamos bloqueando el hormiguero
- Eliminemos a todas las hormigas que hayan pasado por allí

Si ambas casillas son válidas, debemos determinar dónde estará mejor el muro:

1. Si ambas casillas tienen información distinta para la hormiga fiel, se tapia la que tenga el valor más alto.
2. Si ambas tienen la misma información para la hormiga fiel, se miran las feromonas. La que tenga menor cantidad, se tapia.
3. Si ninguna tiene feromona (o están empatadas), se tapa la posición que cierra la rotonda.

Si se cumple todo lo anterior, podemos levantar un muro

➤ Solventando el problema de las rotondas

- El ejercicio 8A consiste en dotar a las hormigas de la capacidad de detectar rotondas, y anularlas.

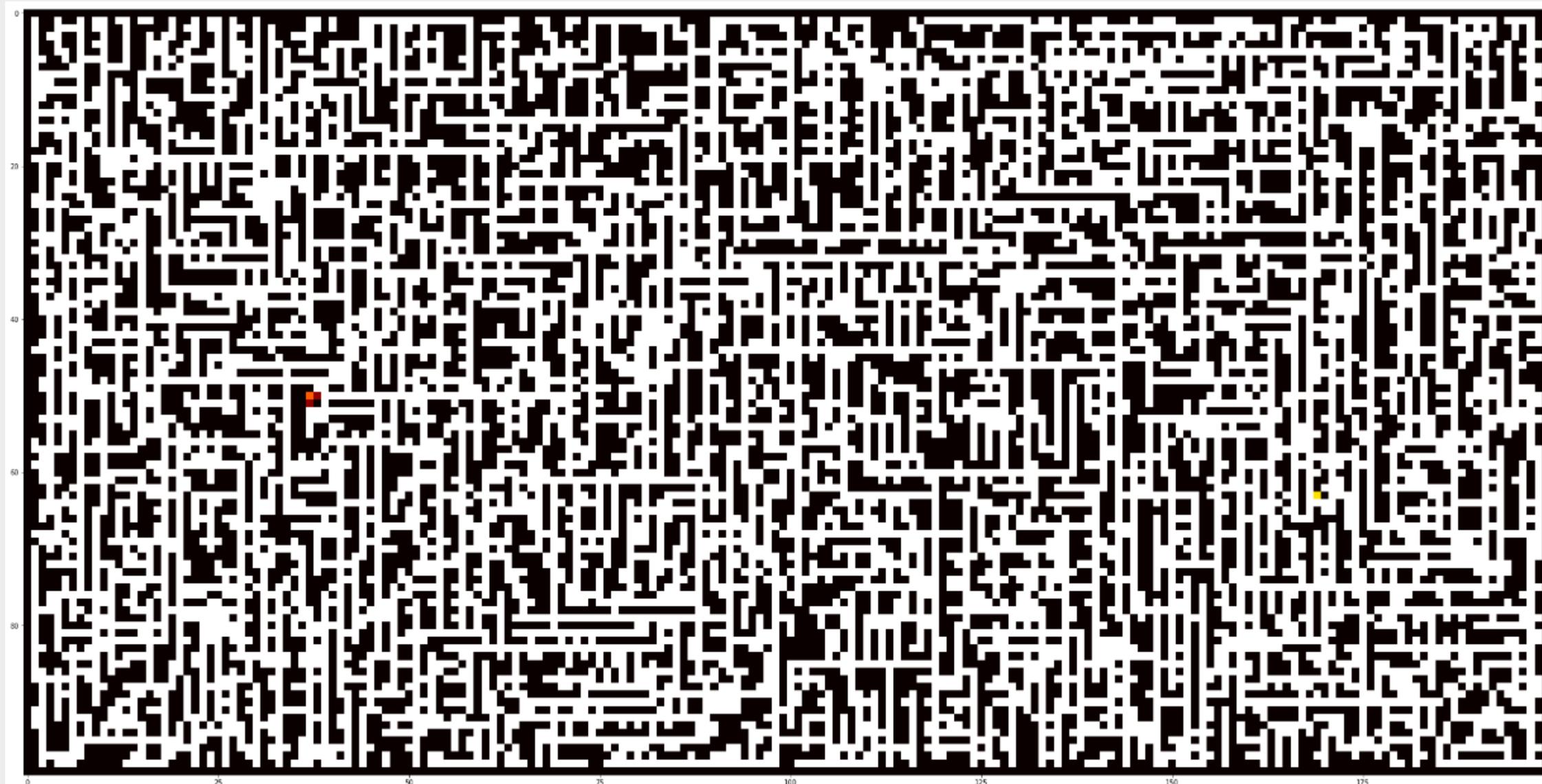
Abre el ejercicio:
8 A rotondas.ipynb



➤ Solventando el problema de las rotondas

Confirmación de las reglas 8A

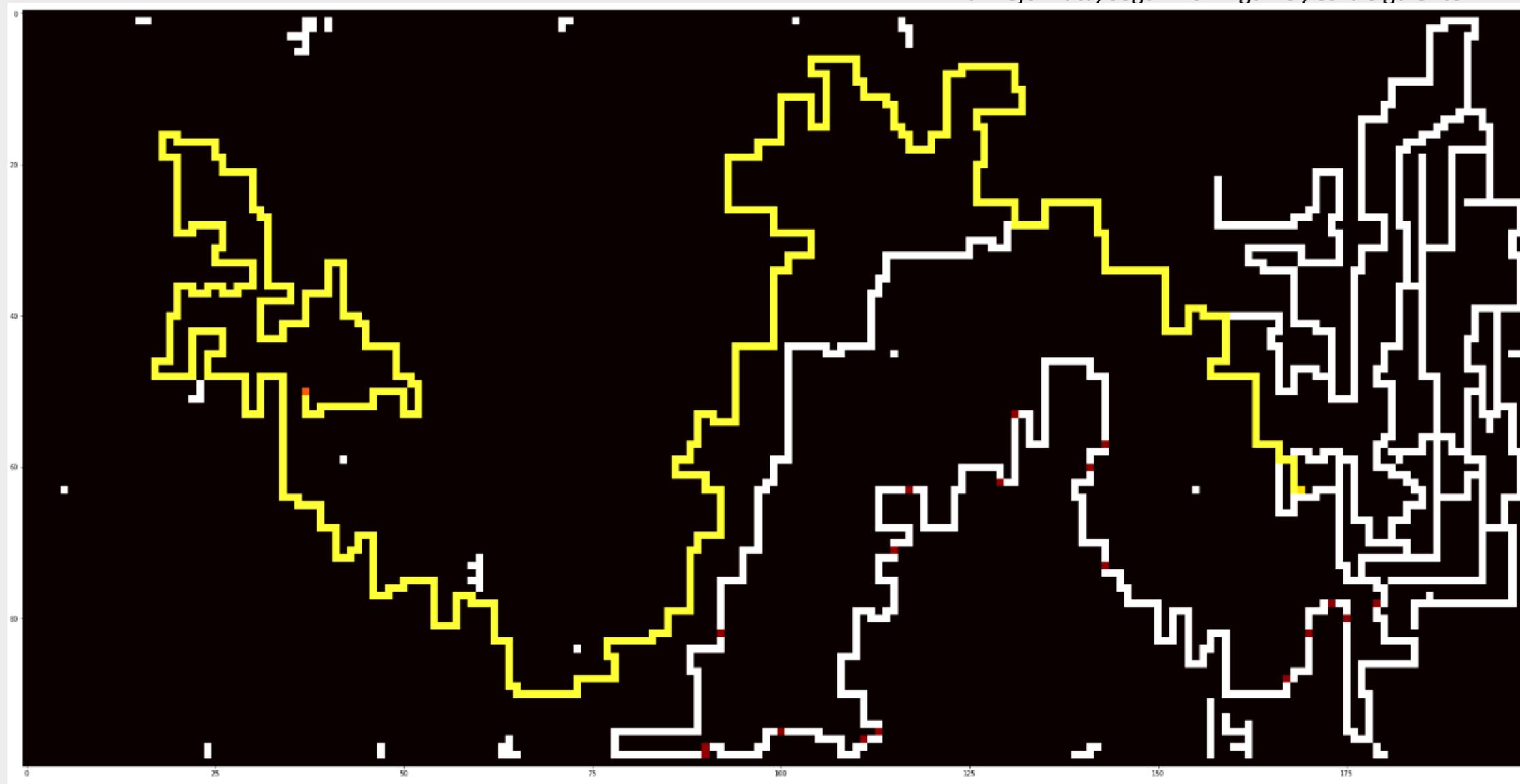
Creamos un laberinto. Uno complejo



➤ Solventando el problema de las rotondas

Confirmación de las reglas 8A

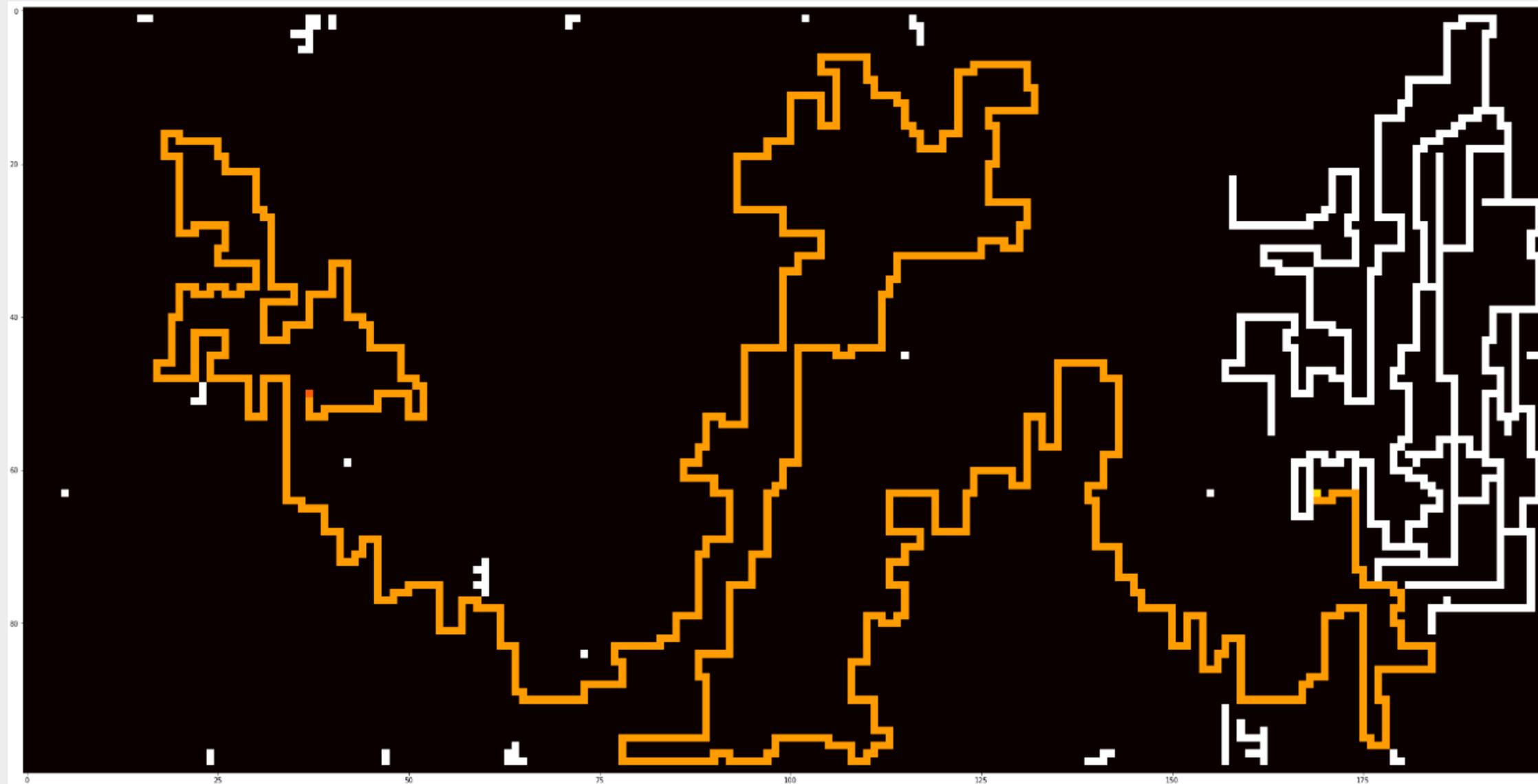
La mejor ruta, según hormiga fiel, es la siguiente



➤ Solventando el problema de las rotondas

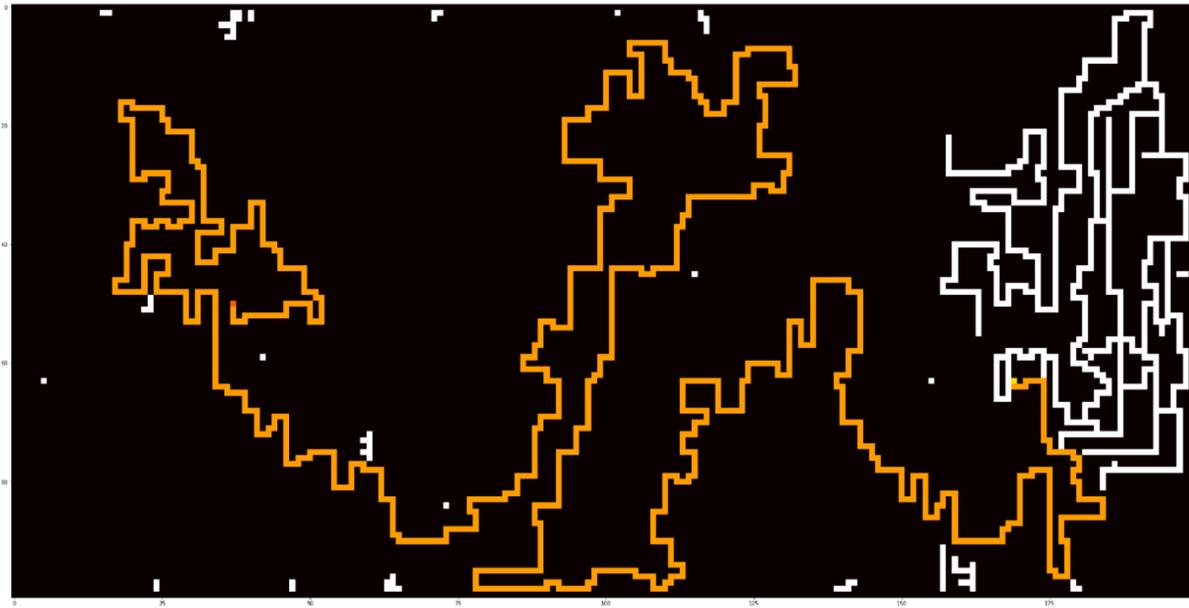
Confirmación de las reglas 8A

La ruta, seguida por las hormigas, al finalizar la exploración, es esta

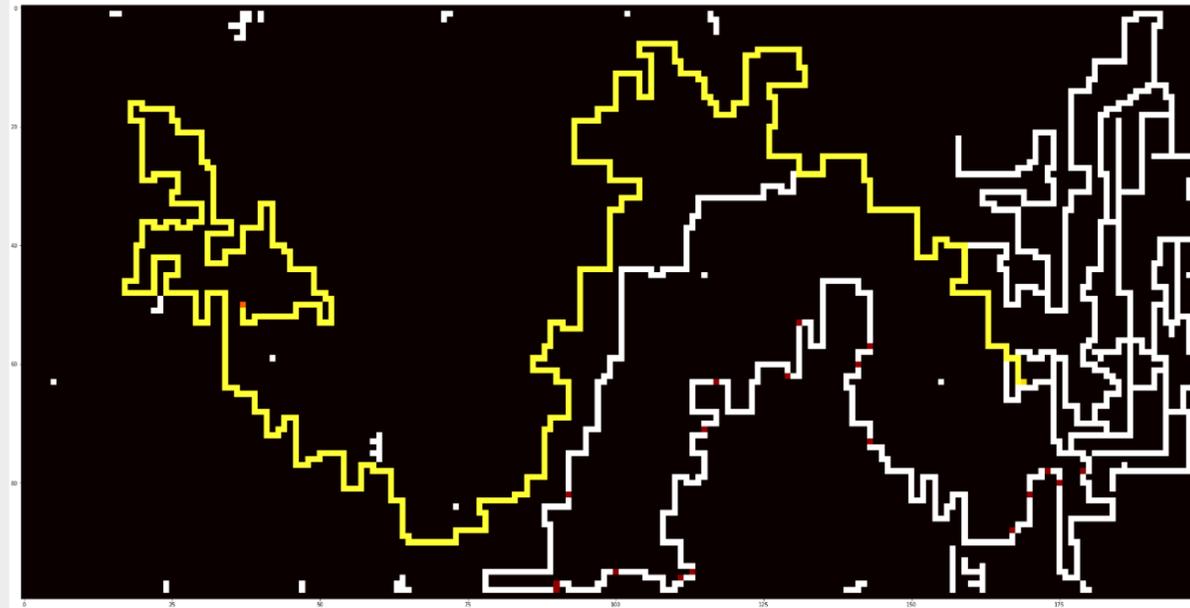


➤ Solventando el problema de las rotondas

Confirmación de las reglas 8A



La ruta, seguida por las hormigas, al finalizar la exploración, es esta

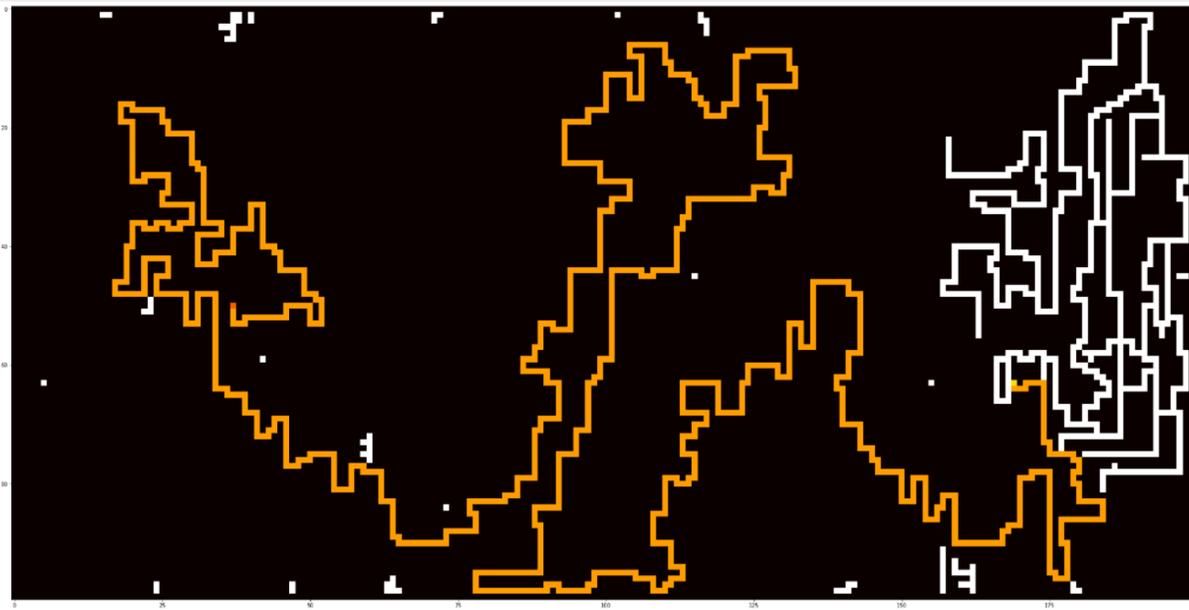


La mejor ruta, según hormiga fiel, es la siguiente

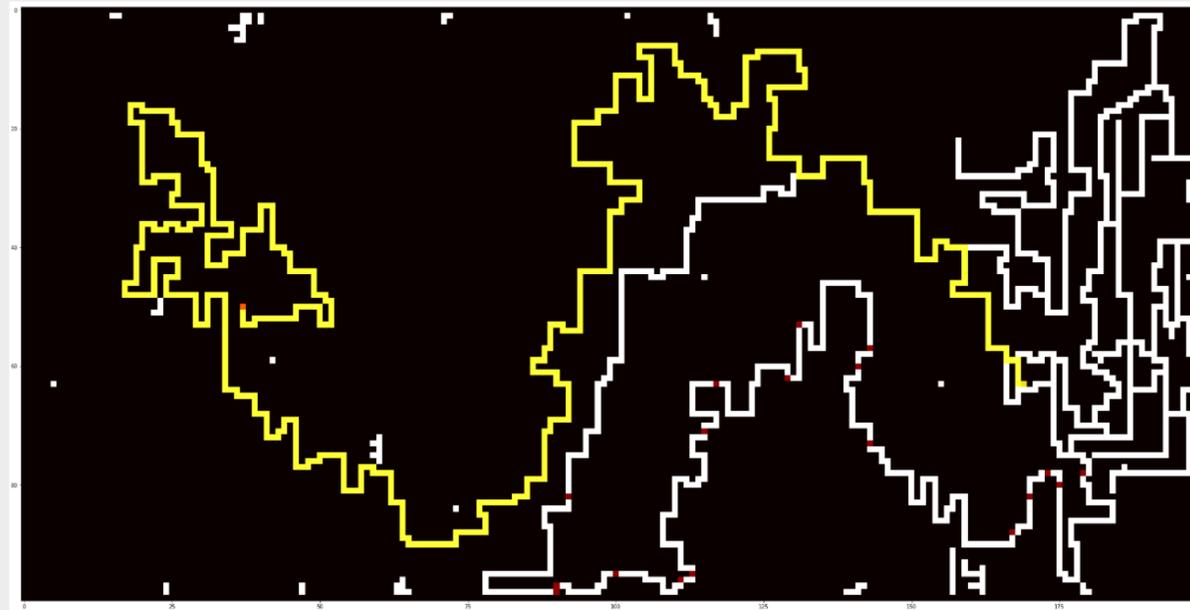
¿Alguien detecta algo raro?

➤ Solventando el problema de las rotondas

Confirmación de las reglas 8A



La ruta, seguida por las hormigas, al finalizar la exploración, es esta



La mejor ruta, según hormiga fiel, es la siguiente

¿Alguien detecta algo raro?

El algoritmo tiende a simplificar el laberinto, cortando, en muchas ocasiones, las mejores rutas YA encontradas

➤ Solventando el problema de las rotondas

Confirmación de las reglas 8A

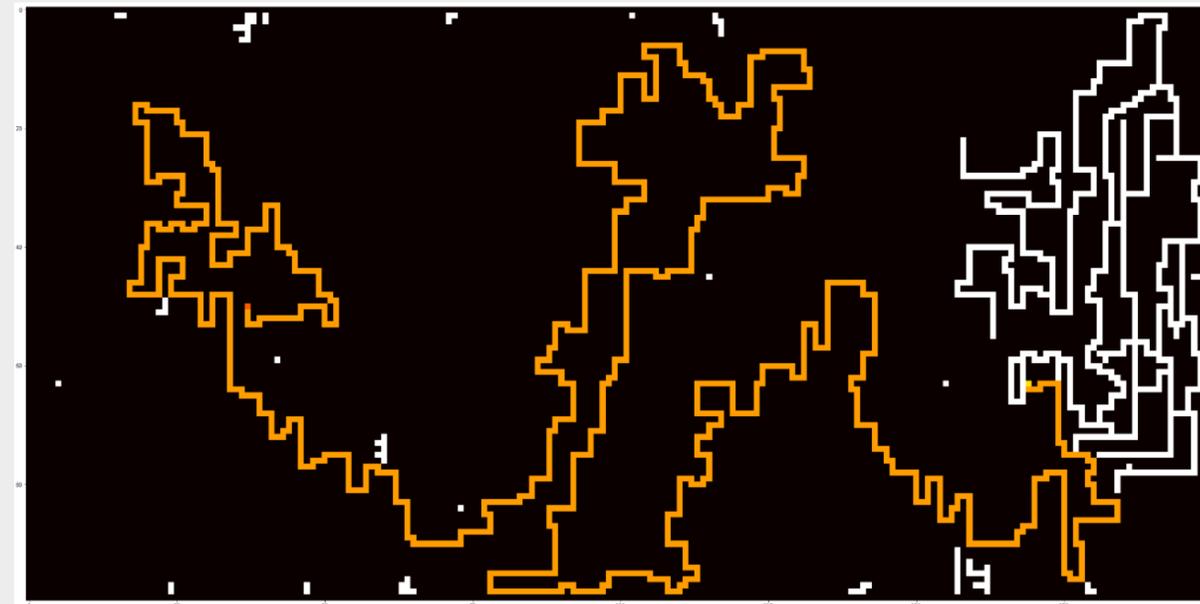
La solución 8A “funciona”.

Las hormigas son capaces de resolver el laberinto y encontrar un camino, directo, hasta la comida.

El problema es que ese camino está lejos de ser el óptimo. El cual hubiéramos encontrado, necesitando mucho más tiempo, si no hubiéramos implementado la mejora de rotondas

Vamos a intentarlo desde otra perspectiva

¿Alguna idea?



➤ Solventando el problema de las rotondas

Requisitos generales

- No estamos bloqueando el hormiguero
- Eliminemos a todas las hormigas que hayan pasado por allí

Nueva propuesta 8B

En vez de poner el muro en el último, o primer movimiento de la rotonda, vamos a analizar todas las casillas de esta

Concretamente, nos vamos a centrar en la información que contienen para la hormiga fiel

Vamos a levantar el muro en la casilla que tenga un valor más alto (la que esté más lejos del hormiguero)

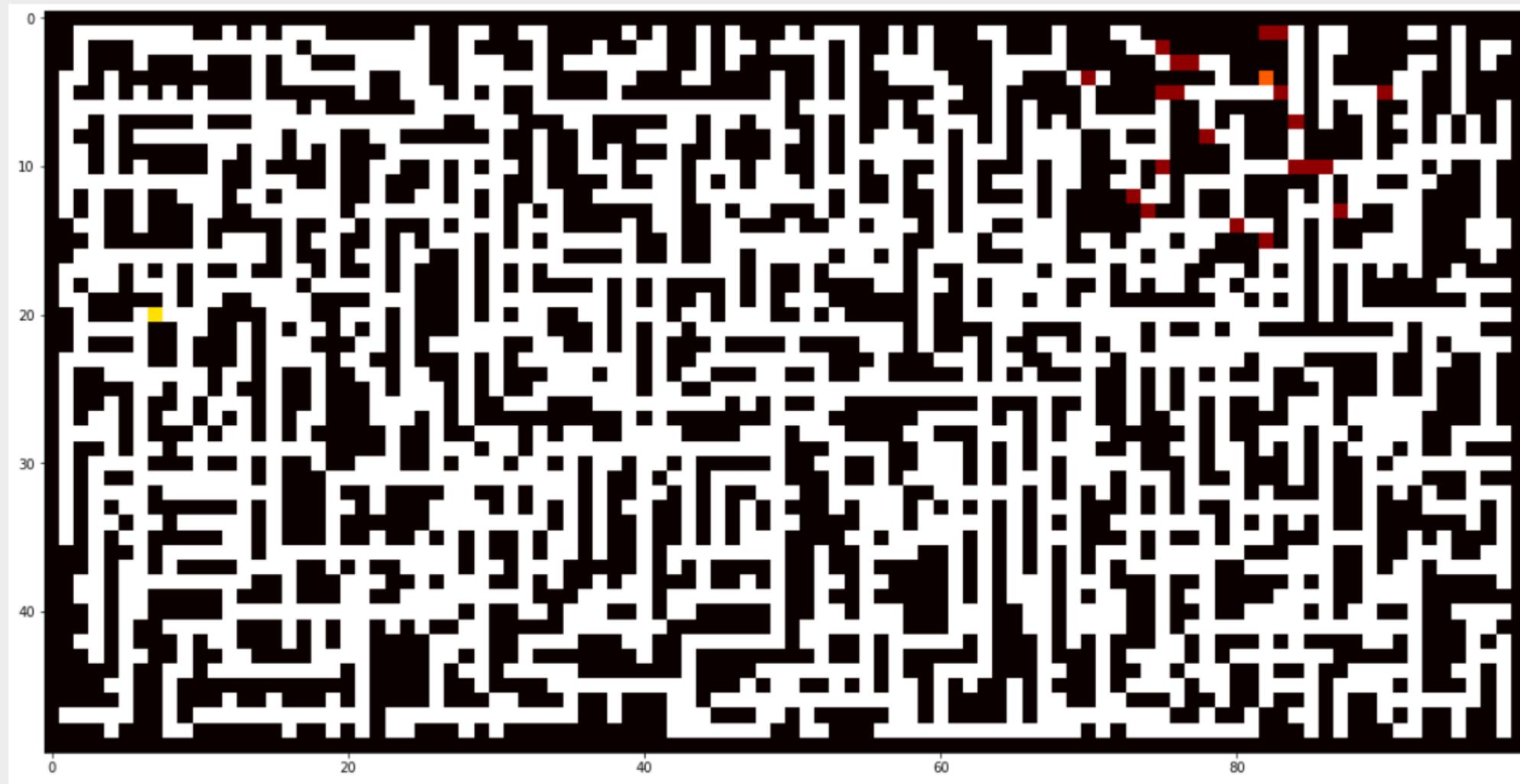
Esta casilla no tiene porqué ser la 1ª o última del recorrido realizado por la hormiga. De hecho, esas dos casillas, al estar una, al lado de la otra, contendrían información muy similar para la hormiga fiel.

En caso de empate, levantaremos el muro en la primera que tenga el valor a eliminar.

➤ Solventando el problema de las rotondas

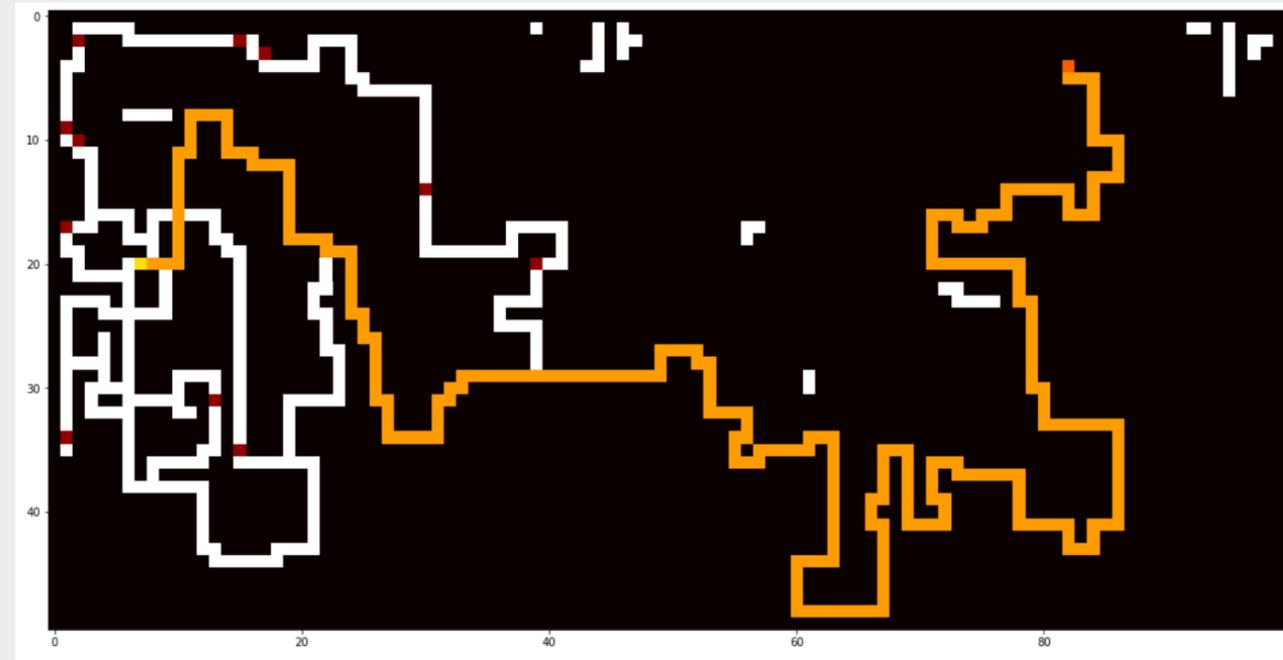
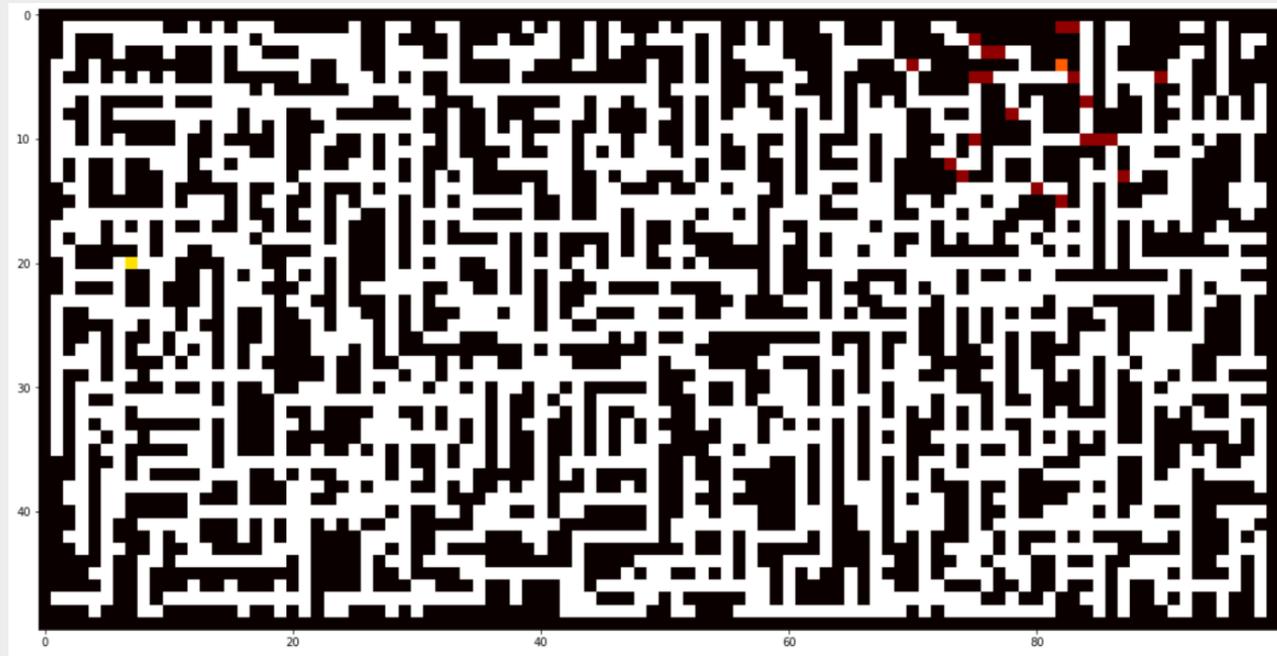
Confirmación de las reglas 8B

Creamos un laberinto. Uno complejo



➔ Solventando el problema de las rotondas

Confirmación de las reglas 8B



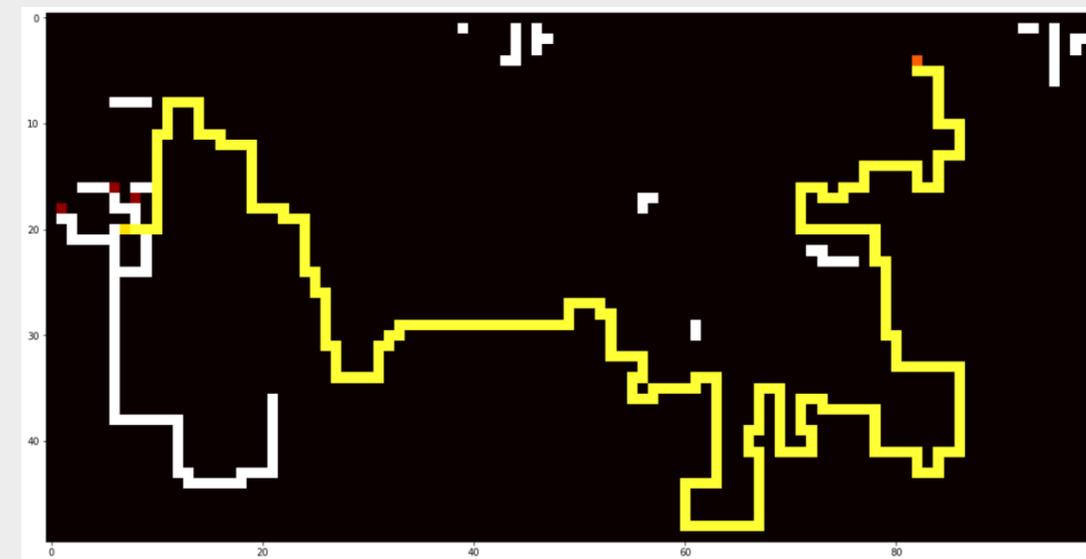
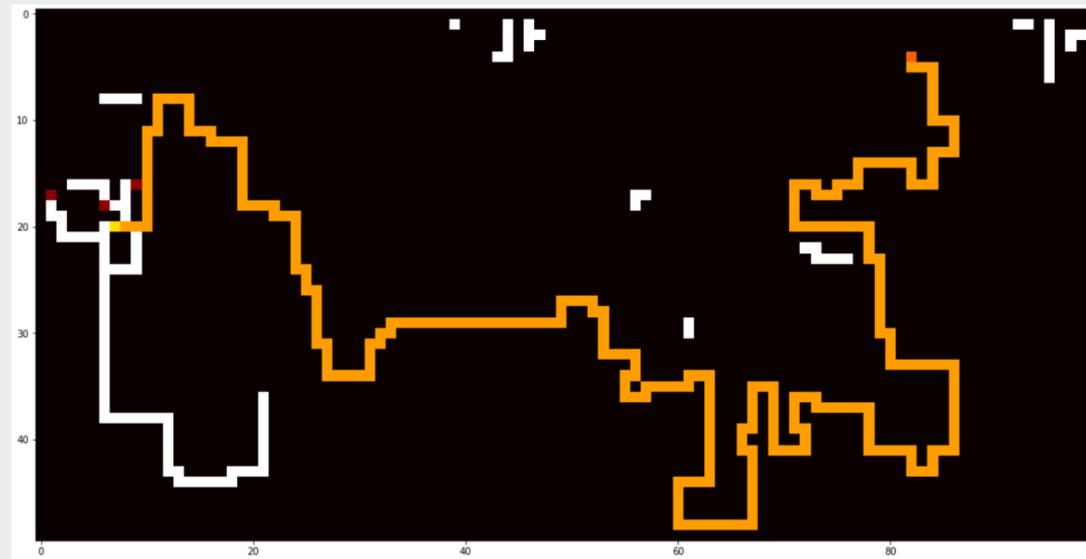
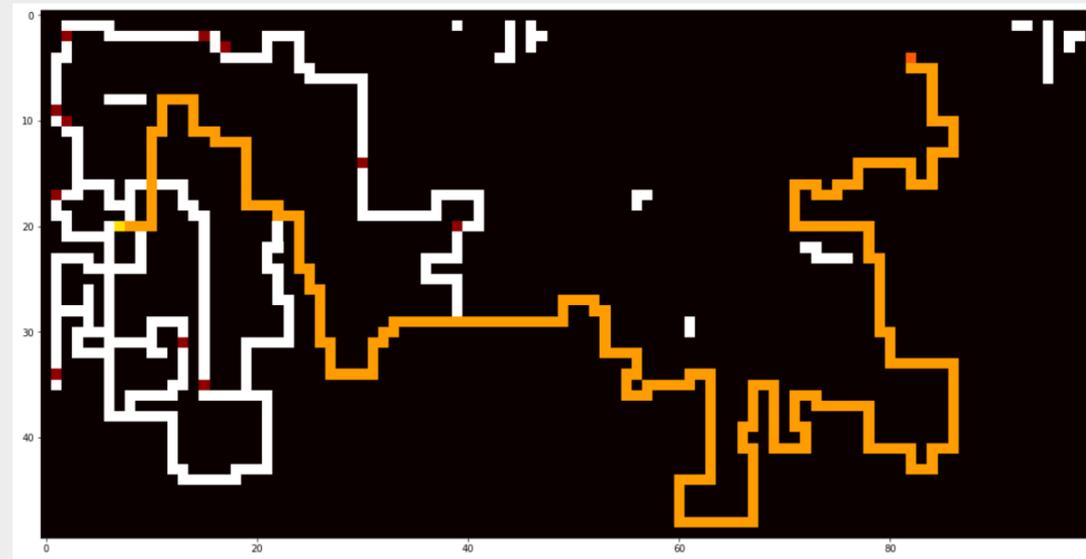
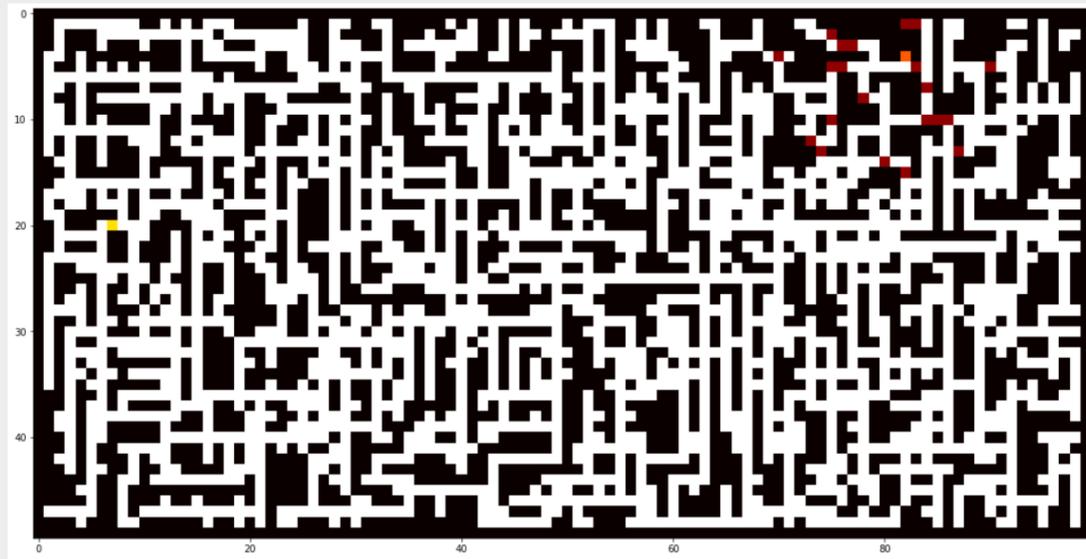
Las hormigas encuentran la comida por 1ª vez

La optimización de la parte derecha no es perfecta, pero se ha realizado en solo unos pocos segundos

Tenemos aún varias rutas que llevan a la comida. Lo que queremos ver es si las hormigas “cortan” la mejor ruta que YA han encontrado

➤ Solventando el problema de las rotondas

Confirmación de las reglas 8B



Solventando el problema de las rotondas

Análisis de propuesta 8B

Hasta que encuentra la comida por primera vez, funciona igual de bien que la solución 8A

¿Detectamos alguna mejora desde que encuentra la comida por primera vez?

¿Bloquea con menor frecuencia los caminos cortos entre el hormiguero y la comida?

Tras muchas pruebas, este sistema funciona mucho mejor

No corta las mejoras rutas encontradas hasta el momento por el hormiguero

Por lo que la ruta de la hormiga fiel y las últimas rutas (naranjas) coinciden.

Parece que hemos solventado el problema

➤ Solventando el problema de las rotondas

- El ejercicio 8B consiste en dotar a las hormigas de la capacidad de detectar rotondas, y anularlas.

Abre el ejercicio:
8 B rotondas.ipynb



➤ Solventando el problema de las rotondas

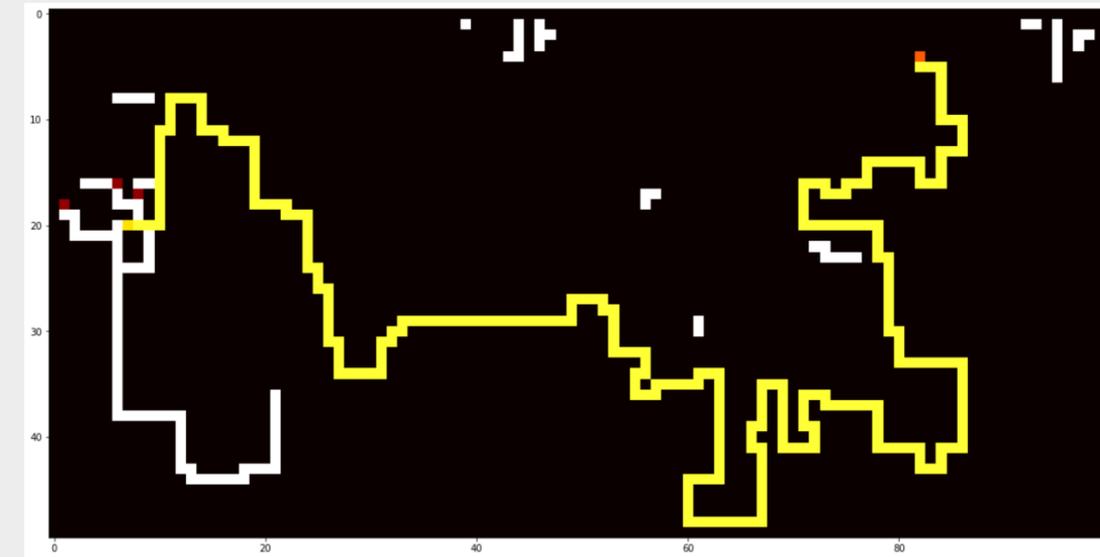
Confirmación de las reglas 8B

La solución 8B funciona.

Las hormigas son capaces de resolver el laberinto y encontrar un camino, directo, hasta la comida.

Dicho camino no será el óptimo absoluto. Pero la relación tiempo / eficiencia es mucho mejor que en la solución 8A, y mucho mejor que la solución original, si necesitamos una respuesta en un tiempo razonable.

Sin embargo, me gustaría intentar algo más



➤ Solventando el problema de las rotondas

Nueva propuesta 8C

El problema del 8A era que cortaba las rotondas sin tener en cuenta la ruta óptima encontrada hasta el momento

Aunque no afectaba a la solución final, que depende de la hormiga fiel, visualmente, no tenía mucho sentido. Dado que empeoraba el camino que utilizaban las hormigas para llegar hasta la comida (al menos, frecuentemente).

¿Y si dotamos de una nueva habilidad a la hormiga fiel?

Cuando levantábamos un muro, en el 8A, anulábamos la feromona de la casilla y la marcamos como un muro

Pero no alterábamos la matriz fiel: en la que se basa en movimiento de la hormiga fiel

Por este motivo, al terminar la optimización, la hormiga fiel mostraba la mejor ruta encontrada

¿Y si dotamos a la hormiga fiel de la capacidad de excavar?, de reabrir caminos bloqueados.

➤ Solventando el problema de las rotondas

Nueva propuesta 8C

Si dotamos a la hormiga fiel de la capacidad de excavar, se nos abren nuevas posibilidades, con sus nuevos problemas

Nos seguimos basando en el 8A. Es decir, levantaremos un muro en la 1ª o última casilla de la rotonda, en función de la información que estas contengan de la matriz fiel. En caso de empate, se consultan las feromonas.

Dotar a la hormiga fiel de la capacidad de excavar es fácil: hay que marcar las casillas de su ruta, de nuevo, como pasillos.

Y hay que invocar a la hormiga fiel con mayor frecuencia. No solo al finalizar el algoritmo. Podríamos llamarla, por ejemplo, cada vez que se encuentre la comida.

¿Algo más que debemos tener en cuenta?

Solventando el problema de las rotondas

Nueva propuesta 8C

¿Algo más que debemos tener en cuenta? Efectivamente, sí:

1. Además de marcar la casilla como pasillo, deberemos devolverle feromonas. Dado que también las eliminamos. Si no hacemos esto, ninguna hormiga usará jamás ese camino.

➤ Solventando el problema de las rotondas

Nueva propuesta 8C

¿Algo más que debemos tener en cuenta? Efectivamente, sí:

1. Además de marcar la casilla como pasillo, deberemos devolverle feromonas. Dado que también las eliminamos. Si no hacemos esto, ninguna hormiga usará jamás ese camino.

El proceso es muy sencillo: debemos calcular el percentil 98 – 99 de las feromonas actuales del laberinto. La gran mayoría de casillas tendrán las feromonas a cero (muros), por lo que debemos calcular un percentil alto.

Posteriormente, debemos asignar ese grado de feromonas, a cada una de las casillas del camino reabierto. Instando a las hormigas a que lo utilicen de nuevo

¿Algo más?

➤ Solventando el problema de las rotondas

Nueva propuesta 8C

¿Algo más que debemos tener en cuenta? Efectivamente, sí:

1. Además de marcar la casilla como pasillo, deberemos devolverle feromonas. Ej: percentil 99 del laberinto.
2. Las hormigas tienden a volver a cerrar, constantemente, los caminos reabiertos.

El problema está en que hemos vuelto a abrir una rotonda. Las hormigas la detectarán a los pocos segundos, y aplicarán el algoritmo 8A. En consecuencia, da igual las veces que reabramos el camino óptimo. Las hormigas lo cerrarán de nuevo una y otra vez.

La solución es evidente

➤ Solventando el problema de las rotondas

Nueva propuesta 8C

¿Algo más que debemos tener en cuenta? Efectivamente, sí:

1. Además de marcar la casilla como pasillo, deberemos devolverle feromonas. Ej: percentil 99 del laberinto.
2. Las hormigas tienden a volver a cerrar, constantemente, los caminos reabiertos.

El problema está en que hemos vuelto a abrir una rotonda. Las hormigas la detectarán a los pocos segundos, y aplicarán el algoritmo 8A. En consecuencia, da igual las veces que reabramos el camino óptimo. Las hormigas lo cerrarán de nuevo una y otra vez.

La solución es evidente: debemos marcar el camino como “no bloqueable”. Al menos, mientras este sea el camino óptimo encontrado por el hormiguero. Si las hormigas encontrasen otro mejor, el camino anterior sería bloqueable de nuevo. Y si no fuera así, sus feromonas caerían gradualmente por el efecto de la evaporación.

Solventando el problema de las rotondas

Nueva propuesta 8C

1. La hormiga fiel de marcará su ruta como pasillo, y como “no bloqueable”
2. Asignaremos el percentil 99 de la matriz de feromonas a dicho camino
3. Las hormigas deberán consultar si la celda es bloqueable, o no, antes de levantar un muro
4. Resto, algoritmo 8A

➤ Solventando el problema de las rotondas

Confirmación de las reglas 8C

Creamos un laberinto. Uno complejo



➤ Solventando el problema de las rotondas

Confirmación de las reglas 8B



Las hormigas encuentran la comida por 1ª vez

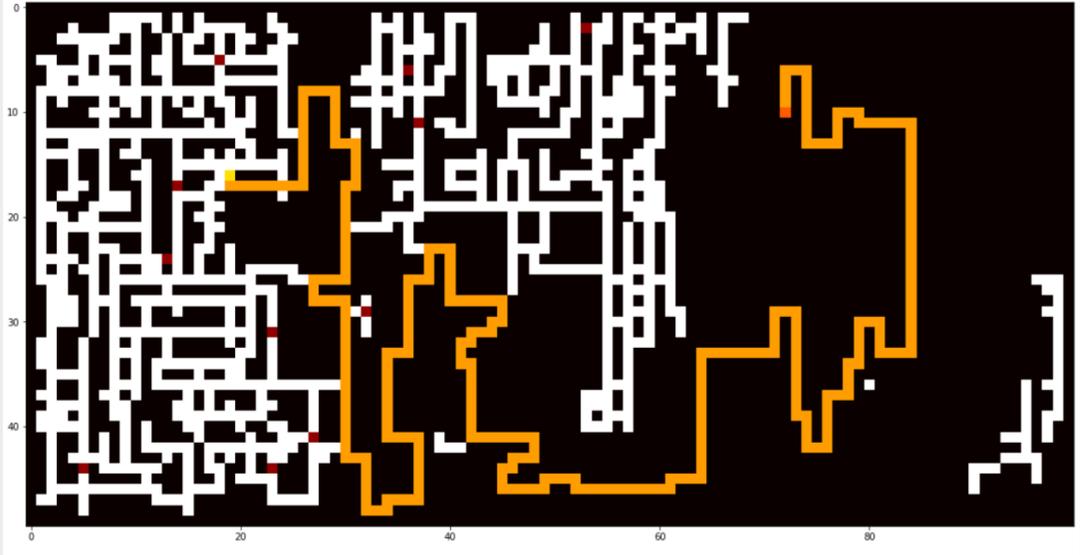
La optimización de la parte derecha no es perfecta, pero se ha realizado en pocos segundos

Tenemos aún muchas rutas que llevan a la comida.

A partir de este instante, la hormiga fiel se encargará de reabrir la ruta óptima encontrada hasta ese instante, marcándola como no bloqueable de nuevo.

➤ Solventando el problema de las rotondas

Confirmación de las reglas 8B



Solventando el problema de las rotondas

Análisis de propuesta 8C

Hasta que encuentra la comida por primera vez, funciona igual de bien que la solución 8A

¿Detectamos alguna mejora desde que encuentra la comida por primera vez?

Tras muchas pruebas, este sistema funciona mucho mejor

No corta la mejor ruta encontradas hasta el momento

Y es capaz de adaptar la ruta, según se encuentran nuevos caminos.

Hemos solventado el problema de nuevo

➤ Solventando el problema de las rotondas

- El ejercicio 8C consiste en dotar a la hormiga fiel de la capacidad de reabrir caminos

Abre el ejercicio:
8 C rotondas.ipynb



Solventando el problema de las rotondas

Ventajas

Ya no es necesario el proceso de poda al encontrar la comida

El tiempo de resolución del laberinto es sensiblemente menor. Si lo comparamos con el tiempo de resolución que tenía antes de “resolver” el problema de las plazas y rotondas

Desventajas

Ya no se garantiza encontrar la mejor ruta posible

➤ Otros algoritmos enjambre

- Algoritmo ABC: colonia de abejas
- Algoritmo de la manada
- Algoritmo de crecimiento bacteriano
- Algoritmo PSO: enjambre de partículas

➤ Algoritmo ABC: colonia de abejas

Las abejas en busca de alimento tratan de localizar la región del espacio con mayor densidad de flores, ya que es allí donde presumiblemente existe más cantidad de polen. Cada abeja vuela de modo errático por el espacio, recordando en todo momento cuál es la región donde ha visto más flores. A su vez, el enjambre sabe colectivamente cuál es la región del espacio, de entre todas las exploradas, donde se han encontrado más flores.

Cada abeja variará individualmente su movimiento con arreglo a estas dos direcciones, volando hacia algún lugar intermedio (vecino). Es posible que la abeja durante ese sobrevuelo encuentre una región con mayor densidad de flores que la conocida hasta entonces (óptimo local). Pasado un tiempo, si se descubre otra región con mayor densidad floral, el enjambre reorientará nuevamente la búsqueda hacia allí, y así sucesivamente.

El algoritmo define una colmena artificial formada por una zona de comunicación (o zona de baile) y dos tipos de abejas

Exploradoras: Marcan zonas con abundancia de flores

Obreras: Deciden ir a una zona marcada por una exploradora para buscar alimento.

Las abejas exploradoras acuden a zonas con abundancia de alimento y bailan en esa área para marcarla. Las abejas obreras deciden a qué fuente de alimento acudir en función de los bailes de las exploradoras (recordando los sitios marcados anteriormente). Cuando el alimento se agota en la zona, la abeja obrera baila para marcarlo como agotado.

➤ Algoritmo ABC: colonia de abejas

Se inicializa la población de abejas
Se inicializan las fuentes de alimento

REPETIR

Las abejas exploradoras se desplazan buscando una fuente de alimento

Evalúa su cantidad de néctar y bailan para trasladar la información a la colmena

Cada abeja obrera analiza la danza, elige una de las fuentes y se dirige hacia allí.

Elige una flor de la zona (vecino), evalúa su cantidad de néctar, y decide si llevarlo a la colmena o pasar a otro vecino.

Se determina si el alimento de la zona está agotado.

Se registran / recuerdan las mejores fuentes de alimento encontradas hasta el momento.

Iteración = iteración + 1

Se cumple criterio de parada

➤ Algoritmo de la manada

Partiendo de algunas reglas sencillas, es posible simular los comportamientos de manadas o de grupos.

Cada individuo tiene tres comportamientos:

- Un individuo muy cercano se separará para evitar invadir o chocar con otro individuo.
- Un individuo próximo modifica la dirección de movimiento, existiendo una tendencia a alinearse en la dirección del vecino.
- Un individuo a una distancia media provoca un acercamiento (como cuando un animal ve a otro de su misma especie).

Es posible agregar un ángulo muerto para hacerlo lo más realista posible.

En función de las distancias configuradas, podemos observar individuos que se desplazan en manadas comportándose como un banco de peces o una bandada de pájaros, cuyas trayectorias parecen aleatorias, evitando chocarse entre sí y evitando a los depredadores.

- Inicializar movimiento líder / líderes de grupo
- Evitar muro o límite del entorno
- Evitar obstáculo
- Evitar otros individuos
- Comportamiento de alineamiento

➤ Algoritmo de crecimiento bacteriano

La evolución del estado de las células está determinada por el estado inicial y no necesita ninguna entrada de datos posterior. El "tablero de juego" es una malla (de dos, o más dimensiones) formada por células que se extiende en todas las direcciones.

En una malla de dos dimensiones, cada célula tiene 8 células vecinas, que son las que están próximas a ella, incluidas las diagonales. Las células tienen dos estados: están "vivas" o "muertas". El estado de las células evoluciona a lo largo de unidades de tiempo discretas (se podría decir que por turnos). El estado de todas las células se tiene en cuenta para calcular el estado de las mismas al turno siguiente. Todas las células se actualizan simultáneamente en cada turno, siguiendo estas reglas:

- Una célula muerta con exactamente 3 células vecinas vivas "nace" (es decir, al turno siguiente estará viva).
- Una célula viva con 2 o 3 células vecinas vivas sigue viva
- En caso contrario muere: ya sea por "soledad" o por "superpoblación".

➤ Algoritmo PSO: enjambre de partículas

PSO permite optimizar un problema a partir de una población de soluciones candidatas, llamadas "partículas". Estas, se han de mover por todo el espacio de búsqueda teniendo en cuenta la posición y la velocidad de cada una de las partículas.

El movimiento de cada partícula se ve influido por:

- Su mejor posición encontrada hasta el momento
- La mejor posición global encontrada por el enjambre

Cada partícula consta de 4 elementos:

- Una posición que representa una determinada combinación de valores de las variables
- El valor de la función objetivo en la posición donde se encuentra la partícula
- Una velocidad que indica cómo y hacia donde se desplaza la partícula
- Un registro de la mejor posición en la que ha estado la partícula hasta el momento

➤ Algoritmo PSO: enjambre de partículas

Crear partícula: definida por posición, velocidad y valor

Crear enjambre: inicializar n nuevas partículas

Mientras no se cumpla el criterio de parada (maximización o minimización de la función)

Evaluar partícula: calcular el valor de la función objetivo en posición actual

Evaluar enjambre: calcular la función objetivo para cada partícula

Mover partícula: actualizar velocidad y posición

$$v_i(t+1) = wv_i(t) + c_1r_1[\hat{x}_i(t) - x_i(t)] + c_2r_2[g(t) - x_i(t)]$$

Mover enjambre: actualizar la posición de cada partícula

Guardar mejor posición conocida de la partícula

Guardar mejor posición conocida del enjambre

Iteración = iteración + 1

Se cumple criterio de parada: devolver mejor posición global

$c_1r_1[x_i(t) - x_i(t)]$ es la componente cognitiva, responsable de que la partícula tienda a moverse hacia la posición donde ha obtenido mejores resultados hasta el momento

$c_2r_2[g(t) - x_i(t)]$ es la componente social, responsable de que la partícula tienda a moverse hacia la mejor posición encontrada por el enjambre hasta el momento. Puede interpretarse como el “conocimiento colectivo”.

➤ Otros algoritmos interesantes

- Algoritmo de enfriamiento simulado: Simulated annealing

➤ Enfriamiento simulado

Para evitar óptimos locales, este algoritmo permite cambiar/saltar a soluciones peores, en función de una probabilidad T (temperatura), que disminuye con el tiempo. La T_0 se inicia con un valor alto y se enfría en cada iteración en función de α hasta alcanzar T_f .

Cada iteración evalúa un nº de vecinos $L(T)$, el cual puede ser fijo para toda la ejecución o depender de la iteración (no analiza todos los vecinos).

Si el mejor vecino es mejor que la solución actual se adopta, en caso contrario, se podría saltar en función de la diferencia entre la solución actual, la del vecino, y de la probabilidad asociada a la temperatura. Finalizada la iteración, se enfría la temperatura.

Prob cambiar = $e^{(- \text{dif entre solución actual y vecina} / \text{temperatura})}$.

Ej $2,71^{(-1/10)}$ da una probabilidad del 90,5% de saltar, $(-10/10)$ un 36,9%, $(-10/1)$ 0,04%.

Inicio Se inicia la temperatura, genera la solución inicial. Repetir Repetir evaluar vecinos (la definición de vecino depende de ti) Si la solución del vecino es mejor que la actual se adopta y se guarda como la mejor. Si la solución es peor, se calcula la probabilidad de saltar. Si aleatorio es $>$ prob saltar, Se salta al mejor vecino $T = \alpha * T$ (se enfría la temperatura). Hasta (la temperatura baje de un umbral mínimo y no haya ningún vecino mejor). Devolver (mejor solución). Fin	Valor inicial de la temperatura T_0 Mecanismos de enfriamiento (alternativas) Descenso constante Descenso geométrico α entre 0,8 y 0,99 Condición de parada: Cuando T baja de un umbral prefijado o un nº fijo de iteraciones
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

➤ Aplicación a finanzas

Como habéis podido comprobar, el algoritmo en sí no es complejo, ni de entender, ni de programar.

- ¿Porqué no se utiliza en finanzas?
 - ¿Cuál es el reto a la hora de aplicarlo a bolsa?
-

➤ Aplicación a finanzas

Como habéis podido comprobar, el algoritmo en sí no es complejo, ni de entender, ni de programar.

➤ ¿Porqué no se utiliza en finanzas?

➤ ¿Cuál es el reto a la hora de aplicarlo a bolsa?

El entorno

La construcción de un entorno financiero coherente, que permita aplicar el algoritmo a las finanzas no es trivial.

Analicemos, con detalle, cómo podríamos aplicar este algoritmo a las finanzas

➤ Aplicación a finanzas: Universo de activos

Para testear la aplicación a finanzas, de los algoritmos enjambre, necesitaremos un universo de activos que explorar
 Para ello, vamos a disponer de 20205 fondos de inversión, desde mayo de 2016, hasta julio de 2021 (1444 filas x 20205)
 Los datos de la tabla son los navs (precios de cierre) de cada uno de los FI para cada día

allfunds_id	206399	27584	12993	8123	11125	8109	11108	27583	10013	157417	...	230294	19023	42535	267220	34808	34806	249846	2913	2914	255530
2016-01-05	23.46	30.93	10.00	26.92	28.92	28.77	30.90	33.22	21.07	10.30	...	100.35	99.97	100.00	94.73	99.95	99.95	127.43	100.23894	100.23600	1390.45
2016-01-06	23.22	30.61	10.00	26.64	28.66	28.47	30.63	32.92	20.92	10.34	...	100.33	99.97	100.00	93.86	99.95	99.95	126.86	100.23894	100.23600	1391.02
2016-01-07	22.42	29.57	10.00	25.74	27.92	27.51	29.84	32.08	20.45	10.34	...	100.25	99.97	100.00	92.57	99.95	99.95	123.59	100.23894	100.23600	1392.28
2016-01-08	22.28	29.37	10.00	25.57	27.80	27.32	29.71	31.94	20.38	10.32	...	100.26	99.97	100.00	92.39	99.95	99.95	119.88	100.23894	100.23600	1393.15
2016-01-11	22.03	29.05	10.00	25.28	27.55	27.02	29.44	31.66	20.20	10.33	...	100.20	99.97	100.00	91.60	99.95	99.95	117.92	100.23894	100.23600	1393.10
...
2021-07-12	29.10	42.81	14.39	34.80	41.50	38.22	45.58	51.05	32.76	11.11	...	92.92	103.91	96.47	107.52	120.90	122.33	228.05	109.47140	120.46439	1524.30
2021-07-13	29.10	42.81	14.39	34.80	41.50	38.22	45.58	51.05	32.76	11.11	...	92.93	103.92	96.47	107.81	120.87	122.30	228.05	109.47140	120.46439	1524.30
2021-07-14	29.10	42.81	14.39	34.80	41.50	38.22	45.58	51.05	32.76	11.10	...	93.01	103.93	96.48	107.97	120.97	122.40	228.05	109.47140	120.46439	1571.50
2021-07-15	29.10	42.81	14.39	34.80	41.50	38.22	45.58	51.05	32.76	11.10	...	92.92	103.94	96.49	107.60	121.21	122.64	228.05	109.47140	120.46439	1573.14
2021-07-16	29.10	42.81	14.39	34.80	41.50	38.22	45.58	51.05	32.76	11.08	...	92.92	103.94	96.49	107.60	121.21	122.64	228.05	109.47140	120.46439	1573.14

Además, dispondremos de las cotizaciones del MSCI en el mismo periodo de tiempo

➤ Aplicación a finanzas: Maestro de valores

Para cada fondo, dispondremos de la siguiente información

- `asset`: Tipo de activo al que pertenece el fondo del 0 al 7 (mide el riesgo)
- `asset_type`: Abreviatura del tipo de activo. Se pueden construir familias con esta característica (`asset_class`, `family`, `subcategory`)
- `class_code`: Clase del fondo. Letra con la que se identifica.
- `clean_share`: No tiene comisión de retrocesión. Es decir, el fondo no paga comisión al distribuidor.
- `currency`: Divisa en la que cotiza el fondo
- `geo_area`: Área geográfica a la que pertenece el fondo (1: 'america', 2: 'europa', 3: 'asia', 4: 'global')
- `geo_zone`: Zona geográfica dentro del área a la que pertenece el fondo (`full_name`, `nested_family`, `nested_cat`)
- `inception_at`: Fecha de creación del fondo
- `income`: Indicador política de distribución de dividendos. Si el fondo es de distribución o acumulación (si reparte dividendos o no)
- `management_fee`: Comisión Gestión (%)
- `manager_id`: Identificación de la gestora
- `manager_name`: Nombre de la gestora
- `name`: Nombre del fondo
- `ongoing_charges`: Total de gastos del fondo según aparece en el KIID del fondo (`management_fee` + custodia y administración)

➤ Aplicación a finanzas: algoritmos enjambre

¿Y ahora qué?

➤ Aplicación a finanzas: algoritmos enjambre

¿Y ahora qué?

¿Dónde están los muros?

¿Dónde los pasillos?

¿Qué es el hormiguero, la comida?

¿Qué hacemos con las feromonas?, ¿cómo la depositamos?, ¿dónde?

➤ Aplicación a finanzas: algoritmos enjambre

¿Y ahora qué?

¿Dónde están los muros?

¿Dónde los pasillos?

¿Qué es el hormiguero, la comida?

¿Qué hacemos con la feromona?, ¿cómo la depositamos?, ¿dónde?

Tenemos muchas preguntas que hacernos

- Respecto al entorno
- Respecto al enjambre
- Respecto a los individuos del enjambre

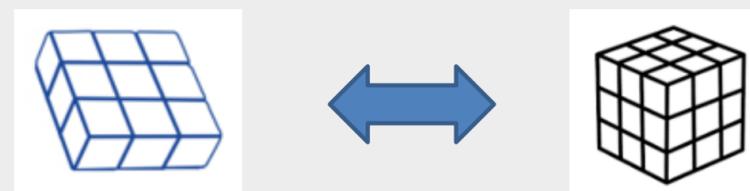
➤ Aplicación a finanzas: Entorno

Tenemos muchas preguntas que hacemos respecto al entorno. De su correcta construcción dependerá que los enjambres puedan explorarlo y obtengamos resultados coherentes. O no.

Tenemos 20.205 fondos de inversión. Ya no tenemos un laberinto, con sus pasillos, paredes, hormiguero y comida.

¿Cómo vais a traducir los fondos es un entorno explorable?.

¿Cómo vais a colocar los fondos?



Podríamos enfocar el problema colocando los fondos en un entorno 3D. Pero tendríais que seguir resolviendo el problema de la su colocación, de cara a optimizar la exploración del entorno.

Generación de entornos y colocación de activos (preguntas a contestar):

- ¿Cuántos entornos vais a generar?
- ¿Cómo colocaréis los activos?
- ¿Qué información pondréis en cada celda?
- ¿El entorno es estático o dinámico?, ¿con qué frecuencia cambia su información?

➤ Aplicación a finanzas: Enjambres

¿Vais a utilizar más de un enjambre?, si es así, ¿de distintas tipologías o de la misma?

¿Cómo vas a adaptar los algoritmos a un entorno financiero?

- **Algoritmo de la colonia de hormigas:** el algoritmo se basaba en encontrar la comida y volver al hormiguero dejando feromonas. Ahora no hay “una comida”, ni un hormiguero al que volver. La feromona no podrá guiar de igual manera al resto de hormigas. Tendrás que adaptar el algoritmo.
- **Algoritmo de la colonia de abejas:** probablemente es el algoritmo más fácil de adaptar al problema. Cuentas con abejas obreras y exploradoras. Sin embargo, deberás definir correctamente la “zona de baile” y cómo las abejas obreras deciden a qué zona acudir para sacar el máximo rendimiento al algoritmo.
- **Algoritmo del enfriamiento del metal:** Este algoritmo no está pensado para este tipo de optimización financiera. Definir bien el sistema de exploración de vecinos, así como la definición de vecino, será crucial.
- **Algoritmo de crecimiento bacteriano:** Interesante para explorar, pero no para optimizar carteras. Deberás adaptar este algoritmo si quieres utilizarlo en un entorno financiero. A menos que no lo quieras para optimizar.
- **Algoritmo del enjambre de partículas:** ¿Qué es una partícula? Un inversor podría entenderse como un enjambre (solución global), o como una partícula. Hay muchos parámetros que optimizar, y mucho trabajo para adaptar este algoritmo. El cómo hayas colocado los activos en el entorno será clave.

➤ Aplicación a finanzas: Individuos del enjambre

Respecto a los individuos, tenemos infinidad de preguntas que hacernos

Asignación: ¿Cómo colocas, inicialmente, a los individuos en el entorno? Si has decidido implementar más de un entorno, ¿cómo asignas a cada individuo a su entorno?, ¿puede moverse entre entornos?

Movimiento: ¿Cómo se moverá el individuo a lo largo del entorno?, ¿qué algoritmo utilizarás? Ahora no tenemos un laberinto que resolver, una comida que encontrar, ni un hormiguero al que volver.

Orientación: Podría ser interesante implementar varios algoritmos enjambre y asignar a cada individuo, al nacer, un algoritmo:

Algoritmo de la colonia de hormigas

Algoritmo de la colonia de abejas

Algoritmo del enfriamiento del metal

Algoritmo del enjambre de partículas

Puesta en producción: en el máster hemos llegado a trabajar con 100 millones de individuos. ¿Con cuántos deberíamos trabajar en nuestro sistema multi-enjambre, con N entornos a explorar? Si el número de individuos fuera muy elevado, ¿cómo lo pondrías en producción?

➤ Aplicación a finanzas: Individuos del enjambre

Número de activos: ¿Con cuántos activos trabajará cada individuo?, ¿será un número constante durante toda su vida?

Criterio de selección: Cada individuo va a moverse por su entorno, encontrando constantemente distintos fondos de inversión. Ya no hay una única “comida” que encontrar. ¿Cuál va a ser el criterio de selección de cada individuo?

Orientación: cada individuo podría tener un criterio de selección / optimización de cartera, que le sería asignado al nacer.

- Maximizar la rentabilidad de capital
- Minimizar las pérdidas que podemos tener
- Maximizar un ratio estadístico (info adicional en 2 diapos)
- Replicar una referencia
- Batir una referencia
- Distribución periódica de rentas
- Obtener rentabilidad positiva en cualquier entorno de mercado

A los criterios de selección, se les puede añadir **criterios adicionales**, con rangos aleatorios por individuo.

- Preferencia por algunos activos exóticos, descorrelacionados, o de una temática en concreto (hasta un % de la cartera)
- Activos populares (elegidos por un número relevante de otros individuos)

➤ Aplicación a finanzas: Individuos del enjambre

Ventana: ¿Con qué ventana de datos vas a trabajar? Cada individuo podría tener una ventana distinta con la que realizar los cálculos para optimizar el criterio de selección.

Criterio de asignación: Una vez has seleccionado el activo, o los activos de la cartera, ¿cómo vas a realizar la asignación de recursos (capital disponible)?

Orientación: optimiza (maximiza o minimiza, depende del caso) el criterio de selección de cada individuo.

Criterio de rebalanceo: Desde el instante inicial (en tiempo y posición), el individuo explorará el entorno, generando una cartera óptima. ¿Cuál va a ser el criterio de rebalanceo de cada individuo?

Información adicional: ratios interesantes

- **Alpha de Jensen:** medida de calidad sobre la gestión realizada. Indica el exceso de rentabilidad obtenida para un nivel de riesgo determinado. Cuanto mayor, mejor.
- **Máximo drawdown:** Mayor caída que ha tenido el fondo en el plazo de estudio. Se expresa en porcentaje.
- **Tiempo de recuperación:** Mide el tiempo que ha tardado el fondo en recuperar el valor respecto al máximo drawdown.
- **Tracking error:** Mide lo que se desvía un fondo respecto del índice de referencia. Es importante para ver si el fondo es de gestión activa o pasiva. Cuanto más cercano a 0, más pasivo.
- **Omega:** Mide la rentabilidad por cada unidad de riesgo (volatilidad) que se asume. Un ratio cercano a 1, o superior, es muy bueno.
- **Sharpe:** proporciona el exceso de rentabilidad sobre el activo sin riesgo por unidad de riesgo asumida (riesgo diversificable + no diversificable). Cuanto mayor sea el ratio, mejor.

➤ Información adicional: ratios interesantes

- **Calmar:** Variación del ratio de Sharpe. Utiliza el máximo drawdown en vez de la volatilidad como medida de riesgo. Mide la rentabilidad obtenida por cada unidad de riesgo máximo asumido. Cuando mayor, mejor.
- **Sortino:** Variación del ratio de Sharpe. Usa como medida de riesgo únicamente las desviaciones negativas en vez de la volatilidad. Mide la rentabilidad conseguida, por cada unidad de riesgo asumido, en las caídas.
- **Information ratio:** Mide si las desviaciones respecto al índice que conforman el tracking error aporta o destruyen valor. Cuanto mayor, mejor.
- **Beta:** Mide como es la intensidad de los movimientos del fondo respecto al índice. Por encima de 1 los amplifica. Por debajo de 1, los mitiga.





Inversión e inteligencia artificial: algoritmos enjambre y optimización de carteras

GUILLERMO MELÉNDEZ

RESPONSABLE DEL LABORATORIO DE INTELIGENCIA ARTIFICIAL DE BME