Jia Long Ji Qiu
Victor Teixidó López
Jiabo Wang
You Wu

Entregable 4

Situaciones

Situación 1

En la primera situación imaginamos un escenario en el que varios agentes a cooperar tienen el rol de explorador asignado. Para poder maximizar la eficiencia de todos los movimientos, el conjunto de exploradores se irán comunicando para, siempre que sea posible, evitar desplazarse por zonas cercanas entre ellos dentro del entorno.

Asegurando dicha comunicación y comportamiento, conseguimos evitar en gran medida que distintos agentes estén explorando la misma zona o porciones del mapa ya visitados por otros agentes, permitiendo de esta manera hacer un seguimiento eficiente sobre los recursos restantes del mapa. En otras palabras, optimizamos la recolección de recursos ofreciendo a los recolectores información rápida y actualizada, mediante una distribución equilibrada de los exploradores sobre el mapa.

Situación 2

En este segundo escenario, nos ponemos en la situación en la que un agente recolector, una vez necesita depositar los minerales, se comunica con un agente almacenador. En este caso, el agente almacenador recibirá una petición por parte del agente recolector para que el primero no se aleje, de tal manera que no se produzcan situaciones absurdas como que un agente recolector persiga sin éxito a un almacenador.

A través de este comportamiento, minimizamos al máximo el número de casillas que los recolectores tendrán que desplazarse para depositar los minerales conseguidos. A la larga, este comportamiento permitirá una mayor recolección de minerales, lo cual es el objetivo de los agentes recolectores. Por otro lado, también conseguimos que el almacenador tenga una gran cantidad de minerales almacenados, lo cual es beneficioso para él ya que es su objetivo principal a seguir.

Jia Long Ji Qiu
Victor Teixidó López
Jiabo Wang
You Wu

Mecanismos

Situación 1

Para la primera situación, el mecanismo ideal sería poder dividir el mapa a explorar en secciones según el número de agentes exploradores, para luego repartirlos entre estos mediante un algoritmo de consenso como Paxos. Sin embargo, en el escenario de Dedale los agentes no disponen de dicha información, sino que deben descubrir por sí mismos la topografía en la que se encuentran y los agentes que se encuentran en el entorno. Por este motivo, una buena alternativa sería la definición de una institución basada en normas.

A la hora de explorar un mapa mediante algoritmos de recorrido de grafos, en un contexto donde varios exploradores están distribuidos sobre el mismo grafo, un problema común es el cruce de caminos a seguir por cada explorador, dando lugar a situaciones en las que un explorador va detrás de otro y acaban recorriendo la misma zona. Aplicando un *Deep First Search* (DFS), por ejemplo, cuando los agentes están recorriendo exclusivamente nodos abiertos o sin visitar (porque aún no se ha llegado a una situación en la que todos los nodos cercanos son cerrados o ya visitados), realmente no nos importa cómo se muevan: todos los movimientos están siendo útiles y, si hay agentes cercanos, pueden ir compartiendo la información del mapa y por ende la exploración será más rápida. El problema surge cuando todos los nodos adyacentes al agente ya han sido visitados y debe volver a un nodo abierto que se había guardado previamente: a la hora de buscar un nuevo punto en el que aplicar nuevamente el algoritmo de DFS, es probable que varios agentes decidan ir a la misma zona sin explorar y acaben recorriendo la misma sección, resultando en movimientos inútiles y un entorpecimiento por parte de algún agente.

Para evitar esto, la institución basada en normas a definir sería: siempre que varios agentes se encuentren y sus caminos a seguir se crucen entre sí, deberá mantener el camino aquel agente cuyo nodo abierto destino (donde se quiera iniciar nuevamente un DFS) se encuentre más cerca, mientras que el resto de agentes deberán buscar otros nodos abiertos que no produzcan una encrucijada con el agente que mantiene su camino. De esta manera, a pesar de no garantizar una distribución totalmente equitativa (que no es posible si no se conoce completamente el mapa), se intenta minimizar la presencia de agentes en una misma sección del mapa alejándolos entre sí siempre que se encuentren y quieran ir hacia la misma zona.

Finalmente, en el caso de que ya se haya explorado todo el mapa pero sigan quedando recursos pendientes por recoger, esta vez sí que se podría considerar hacer una repartición del mapa y llegar a un consenso entre los exploradores para ser situados en una cierta sección, de manera que cada agente tendría que visitar menos nodos y la comunicación entre agentes se podría hacer entre secciones contiguas, permitiendo así tener información actualizada y de rápida transmisión sobre todos los recursos remanentes del mapa. Para ello, el mapa se dividiría en N secciones de más o menos el mismo tamaño (definido por los nodos de cada sección), donde N es el número de agentes que se han encontrado a lo

largo del proceso de exploración, y cada sección se asignaría a uno de los N agentes. Dicha asignación sería el consenso a llevar a cabo entre los agentes exploradores.

Situación 2

Para esta segunda situación, hemos decidido que el mecanismo más adecuado es el de *commitments*.

En un principio, el agente recolector va recogiendo minerales en el mapa, y una vez tenga al máximo su capacidad de almacenamiento, hace una búsqueda de un almacenador cercano en el que poder depositar los minerales. Cuando encuentra uno, se efectúa un *commitment* entre ambos agentes, por un lado, el recolector se compromete a depositar los minerales en dicho almacenador, por otro lado, el agente almacenador se compromete a no moverse de su posición hasta que los minerales hayan sido depositados. Una vez depositados los recursos, ambos agentes vuelven a ser libres y sin compromisos y pueden realizar acciones de la forma habitual.

Un problema de este mecanismo es que puede que el recolector al tener su capacidad de almacenamiento lleno, no tenga ningún agente almacenador en su rango de visibilidad, en estos casos, una posible solución es que no tenga que esperar a tener la capacidad al máximo para ir a buscar al almacenador y hacer el commitment. Es decir, a partir de haber superado, por ejemplo, la mitad de la capacidad, mientras se siguen buscando minerales, paralelamente se vayan buscando agentes almacenadores. Otro aspecto a tener en cuenta es que al tener varios *commitments* simultáneos, el almacenador puede estar quieto por un largo periodo de tiempo, aún así, este no es un aspecto malo ya que significa que, más tarde o más temprano, dicho agente recibirá minerales en su depósito.

Una posible mejora de este mecanismo, es que cuando el almacenador reciba solamente un commitment por parte de un recolector, entonces en vez de quedarse quieto, también vaya a buscar al recolector y así minimizar el trayecto que tenga que recorrer el recolector y aumentar la eficiencia. Esto no es posible cuando se tiene más de un commitment por parte de diversos recolectores, ya que al ir a por uno en concreto se puede alejar de otros. Esto solo podría ser efectivo cuando uno tenga una cantidad de recursos mucho mayor que el resto, pero como solo se hace commitments cuando tengan lleno más de la mitad de la capacidad de almacenamiento, no puede haber una diferencia muy grande entre los recolectores.