

# Tercer Proyecto de Simulación

**Tema: Agentes**

**Nombre y apellidos: Lázaro Jesús Suárez Núñez**

**Grupo: C – 412**

## Orden del Problema Asignado

Se define un ambiente como discreto y con forma rectangular de  $N$  filas por  $M$  columnas. El ambiente es de información completa, pues los agentes conocen toda la información sobre los mismos. Además, el ambiente puede variar aleatoriamente cada cierto tiempo.

Existen además agentes que realizan acciones por turnos. Estas acciones son una por cada turno y son las únicas que modifican el ambiente (además de cuando este cambia aleatoriamente).

Los distintos elementos que existen en ambiente solo ocupan una casilla, y algunos incluso pueden estar en la misma casilla. Estos mencionados elementos son:

- Obstáculos: Aparecen solo en el estado inicial. Pueden ser movidos por los niños, pero no por el robot. Ningún otro elemento puede estar en su misma casilla.
- Suciedad: Aparecen en el estado inicial y se pueden ser generados por los niños en casillas que están vacías. Solo el robot puede pasar sobre esta.
- Corral: Aparecen en el estado inicial y existe una cantidad igual a la cantidad de niños. Se encuentran todas en casillas adyacentes. Solo cabe un niño en cada corral. Pueden coexistir un corral, el robot y un niño en la misma casilla si el robot está cargando al niño o si lo acaba de dejar en el corral.
- Niño: Aparecen en el estado inicial. Ellos en el turno del ambiente se mueven aleatoriamente si es posible (si la casilla a donde van esta vacía o hay un obstáculo, además puede que no ocurra movimiento) y a una de las casillas adyacentes. Si esa casilla está ocupada por un obstáculo este es empujado por el niño, si en la dirección hay más de un obstáculo, entonces se desplazan todos. Si el obstáculo está en una posición donde no puede ser empujado y el niño lo intenta, entonces el obstáculo no se mueve y el niño ocupa la misma posición. Los niños son los responsables de que aparezca suciedad. Si en una cuadrícula de 3 por 3 hay un solo niño, entonces, luego de que él se mueva aleatoriamente, una de las casillas de la cuadrícula anterior que esté vacía puede haber sido ensuciada. Si hay dos

niños se pueden ensuciar hasta 3. Si hay tres niños o más pueden resultar sucias hasta 6. Los niños cuando están en una casilla del corral, ni se mueven ni ensucian. Si un niño es capturado por un Robot tampoco se mueve ni ensucia.

- Robot: Aparece al inicio. Se encarga de limpiar y de controlar a los niños. El robot se mueve a una de las casillas adyacentes, las que decida. Solo se mueve una casilla sino carga un niño. Si carga un niño puede moverse hasta dos casillas consecutivas. También puede realizar las acciones de limpiar y cargar niños. Si se mueve a una casilla con suciedad, en el próximo turno puede decidir limpiar o moverse. Si se mueve a una casilla donde está un niño, inmediatamente lo carga. En ese momento, coexisten en la casilla el robot y el niño. Si se mueve a una casilla del corral que está vacía, y carga un niño, puede decidir si lo deja esta casilla o se sigue moviendo. El robot puede dejar niño que carga en cualquier casilla. En ese momento cesa el movimiento del robot en el turno, y coexisten hasta el próximo turno, en la misma casilla.

El objetivo del robot es mantener el ambiente limpio mientras recoge a los niños y los deja en los corrales. Se considera que el ambiente está limpio si menos del 60% de las casillas vacías no están sucias. Los estados finales de la simulación son los siguientes:

- Si en algún momento el ambiente no está limpio el robot es despedido.
- Si el robot deja a los niños en los corrales y deja el ambiente un 100% limpio gana.
- Si pasan  $100 * t$  ( $t$  es la cantidad de turnos entre los cambios del ambiente) se para la simulación y el robot se considera despedido.

El problema consiste en implementar este ambiente y dos modelos distintos del robot, y en dados 10 tipos de ambiente (tamaño, porcentaje inicial de suciedad y de obstáculos, cantidad de niños y cantidad de turnos entre los cambios aleatorios del ambiente), por cada uno generar y simular 30 ambientes aleatorios para observar la diferencia entre los 2 modelos de los agentes, ya sea para ver la cantidad de veces que ganaron o perdieron y el porcentaje de suciedad que dejan en los tableros.

## Principales Ideas para la Solución

La solución propuesta consiste en una colección de clases para representar el ambiente y sus elementos. Se implementó una clase para representar la información del ambiente y una clase para representar los objetos de obstáculos, corrales, suciedad y niños. En el caso de los robots se crearon dos clases, una para cada tipo de modelo de robot que en lo único que difieren es en la forma en que ellos interactúan con el ambiente.

## Modelos de Agentes Considerados

Los dos modelos de agentes considerados varían en determinados aspectos que los hacen ser más proactivos o reactivos. Se construyó un modelo de robot A que corresponde al más proactivo, y uno B que corresponde al más reactivo.

Antes de entrar en detalles cabe recalcar cual sería un comportamiento proactivo y uno reactivo de los agentes en este ambiente. El comportamiento proactivo sería la capacidad del agente de basar sus acciones en sus propias decisiones que toma en cada momento, por ejemplo, aquí sería que el robot decida qué es lo que va a hacer, si llevar un niño a un corral o limpiar alguna suciedad, y en consecuencia realizar esta tarea y luego pasar a la siguiente. El comportamiento reactivo aquí sería que, independientemente de la acción que el robot este haciendo, un determinado cambio del ambiente haga que este cambie la acción tarea actual que está realizando, por ejemplo, si pasan los turnos determinados y el ambiente varía, que el robot determine si es adecuado seguir con la tarea que estaba llevando a cabo o elegir una nueva que sea más rápida o lo que es lo mismo en este caso, más cercana.

El proceso de elegir que tarea se debe hacer en cada momento no es tan evidente, pues una primera idea, la más obvia tal vez que se nos pudiera ocurrir sería recoger primero a todos los niños y después limpiar, ya que los niños son los que generan la suciedad; pero este enfoque puede no ser el mejor puesto que la condición del despido de los robots nos obliga a mantener también al robot limpiando, o sea, hay que llevar un equilibrio entre las veces que se recogen niños y las veces que se recogen suciedad.

El robot con modelo de tipo A mantiene un comportamiento proactivo en la forma en que realiza sus tareas. La forma en que selecciona una tarea para realizar tiene un enfoque greedy, o sea, busca en sus alrededores y selecciona al

niño o suciedad más cercano para llevar a cabo la tarea correspondiente. Una vez que se enfoca en hacer una tarea no se detiene hasta que termina, con algunas excepciones. Por ejemplo: si recoge un niño, pero por la forma actual del ambiente no tiene como llegar a un corral vacío, se pone a limpiar suciedad.

El robot con modelo de tipo B mantiene un comportamiento más reactivo que el del modelo A. Este igualmente tiene un enfoque greedy en la forma que selecciona la tarea que hacer, excepto, cuando el determina que el nivel de suciedad en el ambiente es mayor que el 55% se pone a limpiar suciedad. También se diferencia en que cuando está realizando una tarea de ir a buscar un niño para llevarlo al corral, si pasa cerca de una suciedad (que generalmente por la forma greedy de seleccionar la tarea se debe generar después de que tomo la tarea actual de buscar al niño) la limpia y después continua. El robot actúa de igual forma si tiene cargado al niño y pasa cerca de una suciedad.

Ambos modelos de robots reaccionan cuando el ambiente cambia, esto es necesario porque puede que cuando cambie el ambiente ya no pueda terminar la tarea que se había planteado (ir a poner un niño en el corral, pero ahora no se puede acceder a ningún corral).

## **Ideas Seguidas para la Implementación**

La simulación consiste en turno a turno pedirle a la instancia del ambiente que mueva a los niños y el robot, y verificar si se llegó a alguno de los estados finales. La instancia del ambiente contiene la información del mismo, con una matriz de enteros que según su valor nos dice si en una casilla se encuentra un obstáculo, una suciedad, un corral, un niño o el robot. También contiene la instancia de cada uno de sus elementos (pues puede ser utilizada más tarde para no tener que recorrer la matriz), y los valores correspondientes a la simulación que se realiza sobre esta, o sea, el turno actual, cuanto falta para que varíe, etc.

La instancia de cada elemento contiene la información de la fila y la columna donde se encuentran, además de otros valores, como, por ejemplo, una instancia de corral contiene un valor que nos dice si ya existe un niño puesto en este. El robot contiene la información también del niño que está cargando, si es q lo tiene, y de la tarea actual que está realizando. Los niños contienen también la información referente a si están ya en una cuna o en los brazos del robot (si lo están no pueden caminar).

En cada turno un niño puede moverse o no, esto fue implementado con una variable aleatoria que nos devuelve un número del 1 al 10, que sería la cantidad

de turnos que el niño esperaría para moverse. La generación de suciedad se lleva a cabo cuando el niño se mueve, se genera una variable aleatoria uniforme en (0, 1) y si es menor que 0.4 entonces se crea la suciedad. Por supuesto se sigue con las instrucciones de la definición del problema y se ponen más suciedades si hay más niños en casillas adyacentes, igual con la misma probabilidad cada suciedad, pero, por ejemplo, la segunda suciedad tendrá probabilidad de  $0.4 * 0.4$  de generarse, y así para 3 suciedades igual sería  $0.4 * 0.4 * 0.4$ . Cuando el niño se mueve selecciona una dirección aleatoria a moverse y se mueve si puede, respetando las restricciones del problema y moviendo obstáculos si es necesario.

El comportamiento del robot como ya se mencionó se basa en una tarea actual que él está realizando, si está realizando una tarea de limpieza se mueve a la suciedad más cercana y cuando esta sobre esta la limpia y pasa a la siguiente tarea, y cuando el robot está realizando una tarea de recogida de niño, primero se dirige al niño más cercano, cuando llega a él lo recoge, y luego se dirige al corral más cercano y cuando llega pone al niño en el corral y sigue para la nueva tarea. Este es el mecanismo en general de comportamiento de los robots, pero en el proactivo se añade el suceso que pueden ocurrir cuando se están realizando tareas de recogida y llevado del niño que se pase cerca de una suciedad.

Para la selección de la tarea a hacer se utiliza una búsqueda a lo ancho (BFS) para encontrar la casilla más cercana que contenga a un niño o una suciedad. Para el proceso de moverse a este niño o suciedad más cercana, así como para el corral más cercano o casilla vacía más cercana se implementó otro BFS pero que llevaba un historial del camino a tomar para ir a cada casilla, entonces se selecciona el primer movimiento de este historial en la casilla a donde se va a ir y se camina respectivamente.

## Consideraciones Obtenidas a Partir de las Simulaciones

Se crearon 10 juegos de datos distintos para probar los agentes. Por cada juego de dato se realizaron 30 simulaciones distintas por cada modelo de robot y se guardaron los resultados correspondientes a las veces que ganaron y perdieron los robots, así como el porcentaje de la suciedad que dejaban en el ambiente. Los resultados se encuentran en output.txt que se encuentra en la misma carpeta del proyecto.

Como se puede observar, en general el modelo de agente proactivo tiene un mayor número de veces ganadas que el reactivo en la mayoría de los ambientes

de los juegos de datos, y también el porcentaje de suciedad dejada por el reactivo es mayor.

Con estos resultados se puede deducir que el modelo proactivo es más efectivo, pues al parecer en este modelo de ambiente es mejor no preocuparse tanto por la recogida de suciedad, pues a lo mejor la forma en que está diseñada la movilidad del robot o la generación de suciedad de los niños implica que cambiar de tarea porque otra está más próxima puede ser más costoso en tiempo que finalizarla desde un principio.

Por tanto, aunque la intuición nos diga que un modelo de agente puede ser más inteligente que el otro, depende mucho de las características del ambiente el hecho de que en la práctica este sea más útil o efectivo que uno que se limita a hacer sus tareas sin dejar que los cambios del ambiente lo perturben. Por supuesto, también depende del nivel de detalle con que esté implementado el comportamiento del agente, así como su capacidad de aprendizaje.