

# Proyecto de Agentes. Simulación.

Antonio Jesús Otaño Barrera  
Grupo C411

A.OTANO@ESTUDIANTES.MATCOM.UH.CU

## 1. Principales ideas asumidas a la hora de resolver el problema

Se realizaron una serie de suposiciones antes de pasar a la solución del problema, sin contradecir la esencia de la orden asignada, con el objetivo de desambiguar algunos puntos en la orden y lograr una correcta concepción del problema. A continuación se listan las principales ideas asumidas:

1. Se interpreta como variación aleatoria la generación de un nuevo ambiente con las mismas dimensiones, la misma cantidad de niños, obstáculos y casillas sucias iniciales.
2. En el ambiente interviene un único agente a la vez.
3. El robot, los niños y los obstáculos pueden moverse en dirección horizontal, vertical o diagonal.
4. Cuando decimos que un objeto se encuentra en una cuadrícula de 3 X 3 asumimos que está en el centro de la misma.
5. Un robot puede pasar solo o cargando un niño por cualquier otra casilla del tablero excepto por una que contenga un obstáculo (Puede estar cargando un niño y a la vez estar en un corral que tiene otro niño, lo que no lo puede soltar)

## 2. Modelos de agentes considerados

Para modelar el comportamiento de los robots se utilizaron máquinas de estados. Un estado determina una acción que se encuentra haciendo el robot y una transición está determinada por una condición y una acción que toma el robot en consecuencia. Los posibles estados en los que se puede encontrar el robot son:

I: Estado inicial

SK: Buscando un niño

SD: Buscando una casilla sucia

MK: Cargando un niño

FS: El robot concluyó exitosamente su tarea (Estado final)

FF: El robot falló en su tarea (Estado final)

El primer modelo constituye un híbrido entre agente reactivo y proactivo. El robot se traza una meta, digamos, recoger un niño (no importa cual), e inmediatamente se pone a trabajar para conseguirlo. Una vez escogida la meta a perseguir el robot no se desviará de su propósito hasta alcanzarla. Una vez cumpla con la meta trazada el robot determinará la siguiente meta a alcanzar. Hasta aquí se puede decir que el agente es de tipo proactivo. Como característica reactiva del agente se puede mencionar que el robot determina en cada turno la mejor manera de alcanzar su meta. Por ejemplo, si está buscando un niño, en cada turno determina el camino hacia el niño más cercano, que puede variar en el siguiente turno. Mientras el porcentaje de casillas sucias no exceda un valor prefijado, el robot se concentrará en recoger y depositar los niños en los corrales. Una vez se compruebe que dicho porcentaje está por encima del valor prefijado entonces el robot se dedicará a limpiar casillas sucias para mantener ese porcentaje relativamente bajo. Las transiciones del primer modelo se representan a continuación:

- I (a) Si el porcentaje de casillas sucias es mayor que 40, entonces ir a FF.
- (b) Si todos los niños están en los corrales y todas las casillas están limpias, entonces ir a FS.
- (c) Si el porcentaje de casillas limpias es menor que 25 y hay niños fuera del corral, entonces buscar el niño más cercano. Si está en una casilla adyacente moverse hacia ella, cargarlo e ir a MK. Sino caminar una casilla en dirección al niño e ir a SK.
- (d) Sino, buscar una casilla sucia, avanzar una casilla en dirección hacia ella e ir a SD

SK (a) Si el porcentaje de casillas sucias es mayor que 40, entonces ir a FF.

- (b) Si el niño más cercano se encuentra en una casilla adyacente, moverse hacia ella, cargarlo e ir hacia MK.
- (c) Sino, avanzar una casilla en dirección al niño más cercano y permanecer en SK
- SD (a) Si el porcentaje de casillas sucias es mayor que 40, entonces ir a FF.
- (b) Si la casilla donde está el robot está sucia, limpiarla. Si el porcentaje de casillas sucias está por debajo de 15, entonces ir a I. Sino mantenerse en SD.
- (c) Sino, buscar la casilla sucia más cercana, moverse una casilla en dirección a ella y permanecer en SD.
- MK (a) Si el porcentaje de casillas sucias es mayor que 40, entonces ir a FF.
- (b) Si el corral más cercano está a una o dos casillas de distancia, moverse hacia él, dejar el niño y pasar a I.
- (c) Sino, avanzar una o dos casillas en dirección al corral más cercano y mantenerse en MK.

El segundo modelo tiene un enfoque un poco más proactivo. El robot se encargará de recoger a todos los niños y llevarlos a sus corrales y una vez haya cumplido con esta tarea, limpiará todas las casillas que quedaron sucias. Las transiciones que salen de los estados SK y MK se mantienen iguales que en el anterior modelo.

- I (a) Si el porcentaje de casillas sucias es mayor que 40, entonces ir a FF.
- (b) Si todos los niños están en los corrales y todas las casillas están limpias, entonces ir a FS.
- (c) Si hay niños fuera del corral, entonces buscar el niño más cercano. Si está en una casilla adyacente moverse hacia ella, cargarlo e ir a MK. Sino caminar una casilla en dirección al niño e ir a SK.
- (d) Sino, buscar una casilla sucia, avanzar una casilla en dirección hacia ella e ir a SD
- SD (a) Si el porcentaje de casillas sucias es mayor que 40, entonces ir a FF.
- (b) Si no hay más casillas sucias entonces ir a FS.
- (c) Si la casilla donde está el robot está sucia, limpiarla y mantenerse en SD.
- (d) Sino, buscar la casilla sucia más cercana, moverse una casilla en dirección a ella y permanecer en SD.

### 3. Principales ideas seguidas para la implementación

La implementación se realizó en Python. El código fuente se encuentra dentro de la carpeta src del repositorio de Github. La misma consta de 3 ficheros:

1. *robot\_strategies.py* Es donde se encuentran implementados los modelos de los agentes y la definición de la clase Robot.
2. *simulation.py* Script utilizado para realizar comparaciones entre los modelos a través de simulaciones en varios ambientes.
3. *environment.py* Donde se encuentra definida la clase Environment que representa el ambiente y sus variaciones.

Para la implementación de los modelos de los agentes utilizamos elementos de Programación Orientada a Objetos. Cada agente tiene como clase padre una clase llamada Robot que tiene un conjunto de funcionalidades que toda entidad de su tipo debe tener, como son moverse y determinar el camino más cercano hacia un objetivo. Por otro lado toda clase hija de la clase Robot debe tener implementado un método llamado *execute* que es el encargado de controlar las decisiones y acciones del robot en un turno. En esta función es donde está implementada la máquina de estados que determina el comportamiento del robot. El robot mantiene un estado interno que le permite conocer en cada llamado a la función *execute*, dadas ciertas condiciones, qué acción realizar y a qué estado cambiar.

La implementación del ambiente y sus cambios se realiza en una clase con el nombre Environment, que tiene como principal atributo el tablero, que no es más que una matriz que guarda el contenido de cada casilla de la habitación donde se realiza la simulación. Para determinar si se genera suciedad cuando se mueve un niño se le asigna una probabilidad de que ocurra a este evento, obteniéndose resultados aceptables con el valor 0.33. Las variaciones aleatorias del ambiente se simulan comenzando una nueva simulación con un ambiente con las mismas propiedades que el anterior pero manteniendo el tiempo transcurrido.

#### 4. Consideraciones obtenidas a partir de la ejecución de las simulaciones del problema

Se construyeron de manera aleatoria 10 ambientes factibles y se realizaron 30 simulaciones por ambiente con cada modelo. Los resultados por cada ambiente, se muestran en detalle en las tablas 1 y 2. Los resultados generales demuestran que el segundo modelo es mejor que el primero.

Ambiente	Exitos	Fallos	Promedio de casillas sucias (%)
1	30	0	10
2	9	21	26
3	0	30	31
4	3	27	28
5	30	0	13
6	0	30	32
7	26	4	18
8	3	27	28
9	0	30	32
10	0	30	32
Total	101	199	25

Cuadro 1: Resultados de las simulaciones para el primer modelo

Ambiente	Exitos	Fallos	Promedio de casillas sucias (%)
1	30	0	10
2	25	5	19
3	0	30	33
4	13	17	22
5	30	0	13
6	13	17	23
7	30	0	15
8	22	8	22
9	0	30	30
10	0	30	31
Total	163	137	22

Cuadro 2: Resultados de las simulaciones para el segundo modelo