

# Simulación - Agentes

Denis Gómez Cruz

C-412

## Requerimientos y ejecución:

Se debe contar con los paquetes `pandas`, `numpy` y `matplotlib` para la ejecución del proyecto, para esto solo es necesario correr con `python3` el fichero `main.py`. Los resultados obtenidos mediante la ejecución se encontraran en el fichero `fig.png`.

## Problema

Se quieren usar varios modelos de agentes sobre un entorno (rectángulo de  $N \times M$  en este caso) donde hay bebés que pueden moverse y generan suciedad. Los agentes tienen como tarea mantener limpio el entorno y llevar todos los bebés hacia el corral (un conjunto de posiciones especiales en el entorno). Para más detalles del problema a resolver véase la orientación del proyecto en el archivo `orden.pdf`.

## Ideas para la solución y implementación del problema

Se creo una clase `environment` (véase `environment.py`) que es la que representa el entorno. Esta es la encargada de mantener el estado de cada casilla (sucio, limpio, obstáculo, etc...) y los agentes, también contiene algunos métodos que son de utilidad para los agentes. El método `simulate` inicia la ejecución de la simulación teniendo en cuenta los posibles estados finales y retorna una serie de valores que son el resultado de la experimentación los cuales se verán mas adelante.

En el archivo `agents.py` se encuentran varias clases que son implementaciones de distintos agentes y todos comparten el entorno en el que se encuentran, la posición donde están dentro de este y un método `move` que es el encargado de ejecutar la acción que el agente decida cuando sea necesario.

- **child** representa a los niños y simula el comportamiento de estos, posible movimiento y ensuciar de acorde al comportamiento descrito que estos deben tener.
- **robot** representa a el robot de casa, contiene todos los métodos necesarios para realizar las acciones de estos como limpiar, recoger niños, moverse, etc..., también contiene métodos primitivos fundamentales para obtener información sobre el ambiente y poder implementar diferentes estrategias.

- **robot\_\_proactive** especialización del agente **robot** el cual usa una estrategia pro-activa.
- **robot\_\_reactive** especialización del agente **robot** el cual usa una estrategia reactiva.

## Agentes

El agente pro-activo tiene un comportamiento *goal-directed*, tras varias experimentaciones con distintas estrategias la que mejores resultados obtuvo, establece como meta llevar los niños al corral primero, limpiando siempre que sea posible y después encargarse de moverse hacia las casillas que aun permanezcan sucias.

EL agente reactivo es similar al agente pro-activo, pero tiene en mayor medida el entorno, pudiendo cambiar su comportamiento bajo determinadas situaciones según estime conveniente, por lo cual este tiene una mejor respuesta ante los cambios en este.

## Simulación

t	Niños	Obstaculos	Suciedad	Proactivo F	Proactivo D	Proactivo P	Reactivo F	Reactivo D	Reactivo P
5	1	25	25	30	0	13.91604	30	0	13.18349
6	2	23	23	30	0	17.45286	30	0	17.09736
7	3	21	21	29	1	23.95183	26	4	25.03996
8	3	19	19	28	2	22.11033	28	2	21.21667
9	4	17	17	23	7	31.04158	18	12	33.029
10	4	15	15	21	9	30.53423	21	9	31.18815
11	5	13	13	12	18	38.24829	8	22	37.87858
12	5	11	11	11	19	36.3905	13	17	36.06759
13	6	9	9	3	27	38.18341	5	25	37.51847
14	7	7	7	0	30	38.85182	0	30	38.79426

En la tabla anterior se hicieron varias simulaciones 30 por cada ambiente, las letras (F, D, P) representan si el robot finalizo (ubico todos los niños en el corral y todas las casillas están limpias), el robot fue despedido (se alcanzó el 60% de casillas sucias) y el porcentaje de casillas de casillas sucias como promedio durante toda la ejecución respectivamente.

Como se puede apreciar el robot pro-activo se comporta mejor con pocos niños dado que su estrategia es buena cuando hay pocos niños, sin embargo, el robot reactivo mantiene un menor porcentaje de casillas sucias en la mayoría de las simulaciones, incluso se comporta mejor cuando la cantidad de niños es grande, lo cual tiene sentido ya que el porcentaje de suciedad puede ser muy grande y una simple estrategia como la del robot-activo no es buena en estos casos.