

Proyecto de Simulación

AGENTES

30 de noviembre de 2020



Facultad de Matemática y Computación
Universidad de la Habana

Autor: Daniel Alberto García Pérez **C412**
d.garcia@estudiantes.matcom.uh.cu

1. Ideas seguidas para la solución del problema

Después del estudio de la bibliografía acerca de *agentes*, se modeló el ambiente definido para nuestro problema, así como la generación de los elementos que intervienen en el mismo (objetos y agentes), de igual forma la variación periódica. Los objetos y los agentes fueron modelados siguiendo determinadas pautas explicadas mas adelante. De igual forma se expondrán los detalles de la implementación de los modelos.

2. Modelos

En el caso de los *bebés*, se modeló pensando en que su movimiento se realizará de forma aleatoria.

En el caso de los *Robots de la Casa*, se usaron dos modelos, ambos reactivos, pero con alto grado de pro-actividad diferenciándose únicamente en la prioridad de sus categorías.

Cabe destacar que ambos tienen una arquitectura de **Brooks**, los cuales se presentan a continuación. Las categorías son las siguientes:

1. Si la posición actual está sucia, se limpia.
2. Si el robot está libre y puede alcanzar algún bebé, se mueve hacia el bebé más cercano.
3. Si el robot puede alcanzar alguna basura, se mueve hacia la basura más cercana.
4. Si el robot carga un niño y puede alcanzar algún corral, se mueve hacia el corral más cercano.
5. Se queda en su lugar.

Prioridad de cada modelo:

- **Kangaroo:** $1 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow 5$
- **Cleaner:** $1 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 5$

Como se aprecia, la diferencia radica en el orden entre la 2da y 3ra categoría. El efecto que se logra con esto es que el modelo **Kangaroo** es más dado a buscar niños y llevarlos al corral, en cambio **Cleaner** prioriza la limpieza. Cada uno tiene como su segundo objetivo el principal objetivo del otro.

Con este escenario tan interesante entre manos veremos que nos revelan las simulaciones, pero la intuición indica que **Kangaroo** debería obtener mejores resultados ya que trata de cumplir con el objetivo de la limpieza de forma indirecta, eliminando la fuente de la suciedad, **Cleaner** lo hace de forma directa, pero con tantos niños sin ser colocados en su corral, debería generarse más basura que la que él es capaz de limpiar.

3. Detalles de la implementación

La implementación fue realizada en **Python**. Se implementaron todos los elementos presentes en el problema, tanto el ambiente, los objetos como los modelos de agentes que participan.

Se tiene dos ficheros, el primero `environment.py`. Aquí es donde está implementado el ambiente de este problema: `KinderGarden`, así como sus características específicas. Sus parámetros son: las dimensiones del rectángulo, el porcentaje inicial tanto de suciedad como de obstáculos, una lista de objetos que representan a los bebés, una lista de objetos que representan a los robots y finalmente el tiempo cada cuanto ocurre la variación. Dichos objetos tanto los bebés y robots, deben estar dotados de una función `action` que retorna algunas de las acciones válidas en este ambiente, definidas en este mismo archivo.

En cuanto a los detalles de los cambios de ambiente:

- los bebés al moverse generan basura con una variable aleatoria que distribuye $\min(\lfloor X \rfloor, k)$, donde $X \sim \exp(1)$ y k es la cantidad máxima de basura que se puede generar según la cantidad de bebés en su cuadrícula de 3×3 ,
- la variación cambia de posición todos los objetos y agentes, de forma aleatoria, incluyendo el corral, pero este último mantendrá la propiedad de seguir conectado después de la variación.

El segundo fichero, `agent.py`, es donde se encuentran las definiciones de agentes abstractos: `Agent`, `Robot`, así como la implementación concreta

de las modelaciones de los mismos usadas en este proyecto: **Baby**, **Person**, **StaticRobot**, **Kangaroo** y **Cleaner**.

Agent, es al clase base para los agentes, y esta dotada, de una posición (x, y) en el tablero, de la función **action**, aún sin implementar y la función **set** para actualizar el estado de su ubicación en el tablero ambiente.

Robot, que hereda de **Agent**, es la base para cualquier modelo de *Robot de la Casa*, dotado de funciones para la actualización de su estado interno tales como: **hold** y **drop**. La función auxiliar **get_direction_to** sirve para dado un tipo de celda obtener cuál es el movimiento que más acerca al robot a este objetivo en el caso de ser alcanzable, la misma no es más que un simple algoritmo de **BFS**.

Baby, que hereda de **Agent**, es la implementación de nuestro modelo de *bebé*, el cual en **action** retorna uno de los movimientos válidos en el ambiente que se percibe de forma aleatoria, solo en el caso de que el bebé se pueda mover. A su vez esta clase sirve de base para cualquier implementación futura de modelos de bebés más inteligentes, que por ejemplo huyan de los robots o busquen a otros bebés para crear focos de bebés.

Person y **StaticRobot**, son implementacion concretas de un **Robot**, pero solo con fines de prueba del buen funcionamiento de las características del ambiente. El primero toma la acción de la entrada estándar y el segundo simplemente siempre retorna **Action.Stay**. Muy útiles a la hora del *debugging*.

Kangaroo y **Cleaner**, son implementacion concretas de un **Robot**. Exactamente la implemenatción de los modelos que se explicaron en la sección anterior.

4. Simulaciones

A continuación los resultados de un conjunto de simulaciones partiendo de 12 juegos de parámetros distintos para conformar los ambientes que se simularan unas 30 veces y se recogerán los resultados por cada modelo, atendiendo a la cantidad de veces que el robot ganó, fue despedido y el promedio de porciento de suciedad en el ambiente.

**4.1. Tablero de 10×10 , 5 % de basura, 0 % de obstáculos,
4 niños y $t=5$**

Kangaroo:

- Victorias: 30/30
- Despedido: 0/30
- Suciedad: 13.21 %

Cleaner:

- Victorias: 29/30
- Despedido: 1/30
- Suciedad: 20.68 %

**4.2. Tablero de 10×10 , 5 % de basura, 0 % de obstáculos,
4 niños y $t=10$**

Kangaroo:

- Victorias: 30/30
- Despedido: 0/30
- Suciedad: 14.83 %

Cleaner:

- Victorias: 29/30
- Despedido: 1/30
- Suciedad: 21.16 %

4.3. Tablero de 10×10 , 5 % de basura, 0 % de obstáculos, 7 niños y $t=5$

Kangaroo:

- Victorias: 24/30
- Despedido: 6/30
- Suciedad: 27.94 %

Cleaner:

- Victorias: 6/30
- Despedido: 24/30
- Suciedad: 36.81 %

4.4. Tablero de 10×10 , 5 % de basura, 0 % de obstáculos, 7 niños y $t=10$

Kangaroo:

- Victorias: 20/30
- Despedido: 10/30
- Suciedad: 29.69 %

Cleaner:

- Victorias: 7/30
- Despedido: 23/30
- Suciedad: 37.51 %

4.5. Tablero de 10×10 , 5 % de basura, 25 % de obstáculos, 4 niños y $t=5$

Kangaroo:

- Victorias: 30/30
- Despedido: 0/30
- Suciedad: 17.04 %

Cleaner:

- Victorias: 25/30
- Despedido: 1/30
- Suciedad: 25.30 %

4.6. Tablero de 10×10 , 5 % de basura, 25 % de obstáculos, 4 niños y $t=10$

Kangaroo:

- Victorias: 30/30
- Despedido: 0/30
- Suciedad: 16.24 %

Cleaner:

- Victorias: 26/30
- Despedido: 4/30
- Suciedad: 26.80 %

4.7. Tablero de 10×10 , 5 % de basura, 25 % de obstáculos, 7 niños y $t=5$

Kangaroo:

- Victorias: 11/30
- Despedido: 19/30
- Suciedad: 35.53 %

Cleaner:

- Victorias: 1/30
- Despedido: 29/30
- Suciedad: 38.79 %

4.8. Tablero de 10×10 , 5 % de basura, 25 % de obstáculos, 7 niños y $t=10$

Kangaroo:

- Victorias: 8/30
- Despedido: 22/30
- Suciedad: 35.36 %

Cleaner:

- Victorias: 3/30
- Despedido: 27/30
- Suciedad: 40.12 %

4.9. Tablero de 10×10 , 5 % de basura, 50 % de obstáculos, 4 niños y $t=5$

Kangaroo:

- Victorias: 27/30
- Despedido: 2/30
- Suciedad: 21.77 %

Cleaner:

- Victorias: 25/30
- Despedido: 5/30
- Suciedad: 27.99 %

4.10. Tablero de 10×10 , 5 % de basura, 50 % de obstáculos, 4 niños y $t=10$

Kangaroo:

- Victorias: 16/30
- Despedido: 14/30
- Suciedad: 31.78 %

Cleaner:

- Victorias: 12/30
- Despedido: 18/30
- Suciedad: 32.88 %

4.11. Tablero de 10×10 , 5 % de basura, 50 % de obstáculos, 7 niños y $t=5$

Kangaroo:

- Victorias: 2/30
- Despedido: 28/30
- Suciedad: 40.71 %

Cleaner:

- Victorias: 0/30
- Despedido: 30/30
- Suciedad: 42.33 %

4.12. Tablero de 10×10 , 5 % de basura, 50 % de obstáculos, 7 niños y $t=10$

Kangaroo:

- Victorias: 1/30
- Despedido: 29/30
- Suciedad: 41.22 %

Cleaner:

- Victorias: 1/30
- Despedido: 29/30
- Suciedad: 41.05 %

5. Conclusiones

Como se puede ver, **Kangaroo** es el modelo que arroja mejores resultados, resultando estar nunca por debajo de los resultados del **Cleaner**, cuyo mejor escenario es pocos niños y muchos obstáculos. En general **Kangaroo** es un modelo efectivo, que todavía se pudiera mejorar, agregándole ciertas características del **Cleaner** que mejoren los resultados, quizás un modelo híbrido sea una buena solución, pero con más prioridad a la parte del **Kangaroo**, el cual tiene sus peores resultados cuando hay muchos obstáculos.

Otro detalle que resalta a la vista, es que con un t mayor a uno con el cual los resultados son buenos, tienden a empeorar, y esto es debido quizás al hecho de que ambos modelos siempre colocan a los bebés en los corrales más cerca, que tienden a estar en el borde del área de corrales, y llega un momento que el interior se vuelve inaccesible, en este caso la variación es favorable en ese sentido para los robots.

El código fuente de la implementación se puede encontrar [aquí](#).

Índice

1. Ideas seguidas para la solución del problema	2
2. Modelos	2
3. Detalles de la implementación	3
4. Simulaciones	4
4.1. Tablero de 10×10 , 5 % de basura, 0 % de obstáculos, 4 niños y $t=5$	5
4.2. Tablero de 10×10 , 5 % de basura, 0 % de obstáculos, 4 niños y $t=10$	5
4.3. Tablero de 10×10 , 5 % de basura, 0 % de obstáculos, 7 niños y $t=5$	6
4.4. Tablero de 10×10 , 5 % de basura, 0 % de obstáculos, 7 niños y $t=10$	6
4.5. Tablero de 10×10 , 5 % de basura, 25 % de obstáculos, 4 niños y $t=5$	7
4.6. Tablero de 10×10 , 5 % de basura, 25 % de obstáculos, 4 niños y $t=10$	7
4.7. Tablero de 10×10 , 5 % de basura, 25 % de obstáculos, 7 niños y $t=5$	8
4.8. Tablero de 10×10 , 5 % de basura, 25 % de obstáculos, 7 niños y $t=10$	8
4.9. Tablero de 10×10 , 5 % de basura, 50 % de obstáculos, 4 niños y $t=5$	9
4.10. Tablero de 10×10 , 5 % de basura, 50 % de obstáculos, 4 niños y $t=10$	9
4.11. Tablero de 10×10 , 5 % de basura, 50 % de obstáculos, 7 niños y $t=5$	10
4.12. Tablero de 10×10 , 5 % de basura, 50 % de obstáculos, 7 niños y $t=10$	10
5. Conclusiones	11