Informe del Proyecto de Agentes de Simulación

Liset Alfaro lisetalfarogonzalez@gmail.com

Facultad de Matemática y Computación (MATCOM),
Universidad de la Habana (UH), Cuba.

1. Robot de la casa

Este trabajo simula el accionar de un agente (robot niñera) en un ambiente(casa) donde se tienen niños, corrales y objetos. Los niños son capaces de generar basura cuando se mueven en el ambiente, además de una cantidad de basura inicial que existía en el ambiente antes de la llegada del agente. El objetivo de este es mantener la casa limpia, un caso ideal sería cuándo el robot logra tener un comportamiento excelente: casa totalmente limpia y todos los niños en un corral. Si en un momento determinado la casa está sucia: más del 60 % de las casillas sucias, el robot es despedido.
Se desea dada una configuración de ambiente determinada generar el mismo y simular (durante un tiempo t = 100) el comportamiento del robot y guardar las estadísticas: cantidad de despidos, cantidad de veces que fue excelente,cantidad de veces que al menos la casa quedó limpia y el promedio de % de suciedad al finalizar al menos 30 simulaciones

2. Objetos:

- Agente: El Robot de Casa 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 pude 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 Robot y 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 al 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, Robot y niño.
- Niño los niños ocupan solo una casilla. Ellos en el turno del ambiente se mueven, si es posible (si la casilla no está ocupada: no tiene suciedad, no está el corral, no hay un Robot de Casa), y aleatoriamente (puede que no ocurra movimiento), a una de las casilla 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 de Casa tampoco se mueve ni ensucia.

- Corral: el corral ocupa casillas adyacentes en número igual al del total de niños presentes en el ambiente. El corral no puede moverse. En una casilla del corral solo puede coexistir un niño. En una casilla del corral, que esté vacía, puede entrar un robot. En una misma casilla del corral pueden coexistir un niño y un robot solo si el robot lo carga, o si acaba de dejar al niño.
- Obstáculos: estos ocupan una única casilla en el ambiente. Ellos pueden ser movidos, empujándolos, por los niños, una única casilla. El Robot de Casa sin embargo no puede moverlo. No pueden ser movidos ninguna de las casillas ocupadas por cualquier otro elemento del ambiente.
- Suciedad: La suciedad es por cada casilla del ambiente. Solo puede aparecer en casillas que previamente estuvieron vacías. Esta, o aparece en el estado inicial o es creada por los niños.

Principales ideas seguidas para la solución del problema

Para solucionar el problema se tuvo en cuenta las características del ambiente(variable) y por tanto la implementación de la solución varió en dependencia del tipo de agente que se quería simular. En ambos modelos se hizo uso del algoritmo BFS con la adición de una lista con los nodos que se deseaban alcanzar, de este modo, con un mismo algoritmo, pueden operar todos los tipos de robots, cambiando solamente los objetivos a alcanzar.
En general se tiene una clase Robot que implementa este BFS y un conjunto de funciones útiles para el correcto movimiento del mismo; y ademas tiene una función move() vacía. Cada tipo de agente es una nueva clase que hereda de Robot y para su funcionamiento solo debe implementar la función move() de acuerdo a su o sus objetivos.
Esta implementación tiene la ventaja de ser muy práctica a la hora de agregar nuevos tipos de agente y probar otros métodos

Modelos de Agentes considerados (por cada agente se deben presentar dos)

Se consideraron dos tipos de agentes:

- **Proactivo**: Este agente no es puramente Proactivo, ya que debido a que su ambiente cambia constantemente, en cada turno toma un objetivo que aunque podría no variar, debe considerarse que puede hacerlo. Este Robot de casa se enfoca, en primer lugar en encontrar a un niño(el más cercano) y luego de que lo alcanzó lo lleva a un corral(no precisamente al mas cercano, debido a que se usó para esto un algoritmo que clasifica por accesibilidad a los corrales y se le lleva primero a los menos accesibles), luego de dejarlo ahí se dispone a buscar a otro niño y llevarlo a un corral y así sucesivamente hasta que todos los niños estén en un corral, solo entonces se dispone a limpiar la basura
- **Reactivo**: Este agente, a diferencia del anterior, toma una decisión en dependencia de la acción que más pronto pueda realizar. Si carga un niño sus objetivos son limpiar la basura o llegar a un corral, y si no pues limpiar basura o encontrar a un niño.

Ideas seguidas para la implementación

En el al correr en consola el archivo simulation.py se pueden observar el conjunto de experimentos realizados para cada uno de los agentes con las mismas configuraciones de ambiente. Una configuración de ambiente está dada por la tupla:
$((heigth, withd, childrenCount, garbagePercent, objectPercent), variationTime)$
Con cada configuración se realizaron 30 simulaciones por cada tipo de agente.
Si se desea agregar alguna otra configuración se puede modificar la lista *enviroments* de la función principal del archivo *simulation.py*

Consideraciones obtenidas a partir de la ejecución de las simulaciones del problema (determinar a partir de la experimentación cuáles fueron los modelos de agentes que mejor funcionaron)

Aunque en la mayoría de los ambientes ambos se comportaron muy parecidos en cuanto al % de basura final(aunque en algunas ocasiones se separan mucho, la mayoría de las veces se aproximaron bastante), se puede decir que el modelo de agentes que mejor funcionó fue el agente Proactivo, pues logró no ser despedido una cantidad bastante menor de ocasiones.
Se pudo observar que lo más dificulta el trabajo de ambos agentes es la cantidad de niños por la cantidad de basura que generan, así que mientras menos niños tengan mejor se comportarán.
Hay una configuración donde el $variation_time = -1$ esta se hizo con el objetivo de comprobar si el hecho del que el ambiente variara menos le iba a ser más eficiente al robot y la respuesta fue que casi no hay diferencia.

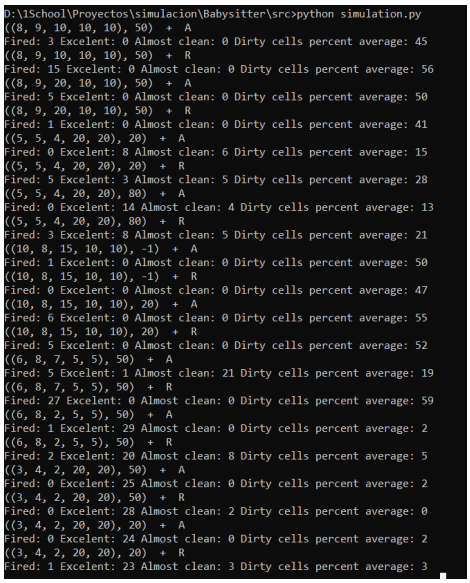

Figura 1. Sección de la salida de la simulación .