Informe

Proyecto de Simulación

Agentes

Carlos Jorge Rodríguez Cuello C-411

[c.cuello@estudiantes.matcom.uh.cu](mailto:c.cuello@estudiantes.matcom.uh.cu)

Enrique Pla Rey C-411

[e.pla@estudiantes.matcom.uh.cu](mailto:e.pla@estudiantes.matcom.uh.cu)

1. **Principales ideas seguidas para la solución del problema**

Puesto que el entorno en que el agente robot se desempeñaba era bastante aleatorio y cambiaba cada cierto tiempo (teniendo en cuenta que los niños se mueven cada turno y modifican el ambiente agregando suciedad y cambiando los obstáculos de posición), no se podía optar por diseñar una estrategia extremadamente proactiva, por lo que se decidió entonces a diseñar un agente que tuviera rasgos tanto proactivos como reactivos.

Dado que el objetivo de la simulación es que el agente robot mantenga la suciedad dentro de un margen especifico, implementamos una estrategia greedy la cual se basa en que el agente robot toma sus decisiones de acuerdo a la distancia en que se encuentran los niños y lo sucio. Esto es debido a que el tiempo es un factor que juega en contra del robot y que cada un tiempo específico el escenario se reestablece.

1. **Modelos de Agentes considerados**

Un agente enteramente reactivo no tendría la capacidad de analizar cuál es la mejor decisión a tomar a largo plazo y a su vez un agente proactivo no sería capaz de adaptarse a un ambiente tan cambiante, por lo cual, como mencionamos anteriormente, diseñar un agente puramente de un estilo u otro no son factibles para la resolución de este problema.

Puesto que como parte del ejercicio había que implementar dos estrategias para el agente robot aquí se presenta una solución breve de los que se presentan en este trabajo, los cuales son agentes deductivos:

Primer agente:

La primera estrategia consiste en un agente que se enfoca más en recoger los niños:

Acción 1-> Si la casilla actual esta sucia, la limpia.

Acción 2-> Si carga un niño da el paso correspondiente al camino más cercano a una posición donde se encuentre una cuna.

Acción 3-> Si existe un camino del robot a una casilla de un niño da el paso correspondiente al camino más cercano a una de estas casillas.

Acción 4-> Si existe un camino del robot a una casilla sucia da el paso correspondiente al camino más cercano a una de estas casillas.

Acción 5-> Gano.

Segundo agente:

La segunda estrategia es similar a la anterior, pero se enfoca más en la suciedad. Las acciones son las mismas, pero se cambia el orden de preferencia de la Acción 3 por la Acción 4 haciendo énfasis en las búsquedas de la suciedad en vez de buscar al niño más cercano.

**3-Ideas seguidas para la implementación**

El primer problema era como simular el ambiente del problema en , en un lenguaje imperativo los tableros usualmente se representan por una lista de lista o una lista de posiciones, etc. Pero usando las herramientas que del lenguaje no fue necesario en su lugar se declararon predicados de la forma:

: El cual tiene como significado si la posición del tablero cumplía la propiedad , donde es para si la posición pertenece al tablero, para el caso que dicha posición este sucia, sí en dicha posición está el robot, etc.

Para generar el ambiente se utilizó el predicado:

: Este predicado genera el tablero, donde es el número de filas del tablero, numero de columnas del tablero, por ciento de casillas sucias, por cientos de casillas limpias, por cientos de casillas donde hay un niño.

Otros predicados importantes del trabajo son:

: Donde es el tipo de acción que va realizar que sería (limpiar o moverse), lista de punto hacia donde se moverá (en el caso que sea moverse), cantidad de turnos que han transcurrido desde que se reinició por última vez la simulación, el tiempo que dura toda la simulación hasta reiniciarla.

: Este predicado cambia el ambiente depende a las acciones que el agente quiere ejecutar. Donde es la acción que recibe, lista de punto hacia donde se moverá (en el caso que sea moverse), es un valor que indica si la simulación termina o no.

**4-Experimentos**

Leyenda:

N: Cantidad de filas.

M: Cantidad de columnas.

PS: Por ciento de suciedad.

PO: Por ciento de obstáculos.

T: Tiempo entre cambios de agentes.

E: Cantidad de éxitos del agente.

D: Cantidad de despidos del agente.

PSR: Por ciento de suciedad resultante.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Agente | N | M | PS | PO | PN | T | E | D | PSR |
| agent1 | 5 | 5 | 0 | 0 | 5 | 10 | 30 | 0 | 0 |
| agent2 | 5 | 5 | 0 | 0 | 5 | 10 | 30 | 0 | 0 |
| agent1 | 5 | 5 | 10 | 0 | 5 | 10 | 30 | 0 | 0 |
| agent2 | 5 | 5 | 10 | 0 | 5 | 10 | 30 | 0 | 0 |
| agent1 | 5 | 5 | 10 | 0 | 8 | 10 | 0 | 30 | 0.61 |
| agent2 | 5 | 5 | 10 | 0 | 8 | 10 | 0 | 30 | 0.66 |
| agent1 | 5 | 5 | 10 | 5 | 8 | 10 | 0 | 30 | 0.609 |
| agent2 | 5 | 5 | 10 | 5 | 8 | 10 | 2 | 28 | 0.58 |
| agent1 | 6 | 6 | 0 | 0 | 5 | 10 | 30 | 0 | 0 |
| agent2 | 6 | 6 | 0 | 0 | 5 | 10 | 30 | 0 | 0 |
| agent1 | 6 | 6 | 0 | 5 | 8 | 10 | 17 | 13 | 0.72 |
| agent2 | 6 | 6 | 0 | 5 | 8 | 10 | 1 | 29 | 0.609 |
| agent1 | 6 | 6 | 10 | 5 | 5 | 10 | 21 | 0 | 0.027 |
| agent2 | 6 | 6 | 10 | 5 | 5 | 10 | 15 | 0 | 0.107 |
| agent1 | 6 | 6 | 5 | 5 | 7 | 10 | 0 | 0 | 0.106 |
| agent2 | 6 | 6 | 5 | 5 | 7 | 10 | 0 | 30 | 0.62 |
| agent1 | 7 | 7 | 0 | 0 | 5 | 10 | 17 | 1 | 0.11 |
| agent2 | 7 | 7 | 0 | 0 | 5 | 10 | 5 | 5 | 0.264 |
| agent1 | 7 | 7 | 0 | 0 | 7 | 10 | 0 | 30 | 0.651 |
| agent2 | 7 | 7 | 0 | 0 | 7 | 10 | 0 | 30 | 0.646 |

**5-Consideraciones a partir de las simulaciones obtenidas:**

* A mayor cantidad de niños la probabilidad de que se despida el robot es muy alta en ambos agentes.
* La cantidad de obstáculos no influye mucho en los despidos de los agentes, pero si en su movimiento afectando la cantidad de triunfos.
* Un por ciento de suciedad determinado puede causar el despido rápido de los agentes.
* En general el agente1 que es el esta enfocado en los niños tiene mejores resultados que el otro.