

Análisis de navegación de agentes reactivos en entornos discretos

Joaquín Santiago Toranzo Calderón^{1,2}, Ramiro Germán Rodríguez Colmeiro¹ y Sergio Alberino¹

1 Universidad Tecnológica Nacional, Facultad Regional Buenos Aires, Grupo de Inteligencia Artificial y Robótica (GIAR), Av. Medrano 951, (C1179AAQ), Ciudad Autónoma de Buenos Aires, Argentina

2 Consejo Nacional de Investigaciones Científicas y Técnicas, Instituto de Investigaciones Filosóficas, Sociedad Argentina de Análisis Filosófico (SADAF), Bulnes 642, (CP1176), Ciudad Autónoma de Buenos Aires, Argentina

jtoranzocalderon@frba.utn.edu.ar

Recibido el 13 de diciembre de 2023, aprobado el 14 de febrero de 2024

Resumen

Evaluar la navegación de agentes reactivos en entornos discretos rectangulares tiene una complejidad heterogénea, y el éxito de esta tarea depende tanto del diseño del agente como de la topología del entorno. Se explora el problema modelando agentes con máquinas de estados finitos con entradas y acciones disponibles mínimas. La navegación se parametriza respecto del tamaño del entorno y de la distancia entre el comienzo y la meta a alcanzar. Se caracterizó un generador aleatorio de entornos que fueron usados para testear agentes típicos. Como resultado, los modelos más exitosos siguen un comportamiento esperable, el de rodear obstáculos indefinidamente.

PALABRAS CLAVE: SIMULACIONES – MÁQUINAS DE ESTADOS FINITOS – ENTORNOS DISCRETOS

Abstract

The evaluation of the navigation of reflex agents in rectangular discrete environments has a heterogeneous complexity, and the task success depends both on the agent design and the environment topology. The problem is explored by modeling agents as finite state machines with minimal available inputs and actions. The navigation is parameterized with respect to the environment size and the distance to the goal that has to be reached. A random environment generator was characterized, for testing typical agents. As a result, the most successful models follow expected behaviour, that is, they go around obstacles indefinitely.

KEYWORDS: SIMULATIONS – FINITE STATE MACHINES – DISCRETE ENVIRONMENTS

Introducción

Cuando una plataforma robótica de poca complejidad (como las presentadas por Álvarez *et al.*, (2013) y Alberino (2015, 2019)) se enfrenta al problema de navegar por un entorno (e.g. para alcanzar un lugar específico) existen diferentes estrategias. Éstas van desde agentes reactivos simples (Russell y Norvig, 2009, pp. 48-52) hasta agentes planificadores (Russell y Norvig, 2009, pp. 234-479), o incluso grandes modelos de aprendizaje por refuerzo (LaValle, 2019). Aunque usualmente los modelos de órdenes superiores ofrecen mejores resultados, los agentes reactivos son aquellos con menos restricciones de implementación y por lo tanto suelen seleccionarse para plataformas simples (*i.e.* con pocos recursos energéticos o computacionales). Sin embargo, no hay una cota clara respecto de la complejidad máxima que estos agentes reactivos pueden manejar para cierta tasa de éxitos.

Los agentes reactivos son un tipo de agente que selecciona acciones sobre la base de sus percepciones (los aspectos observables del entorno). La selección de estas acciones usualmente se representa por reglas condicionales que constituyen una estrategia abstracta. Un agente que se guía sólo por sus percepciones actuales es denominado agente reactivo simple, y uno que tiene en cuenta su historial de percepciones, total o parcialmente, es denominado agente reactivo basado en modelos (Russell y Norvig, 2009). A diferencia de los primeros, los agentes basados en modelos incorporan estados internos que hacen que las acciones a ejecutar dependan de su historial de percepciones, representando así algunos aspectos del entorno relevantes para su estrategia. Esto los hace más complejos que un agente reactivo simple, dado que necesitan registrar su estado interno. De todos modos, esto puede alcanzarse con un costo mínimo en su complejidad y resulta en un agente más ventajoso.

El éxito de un agente no sólo depende de su complejidad, sino también del tipo de entorno a navegar (Russell y Norvig, 2009, pp. 34-63). Existen muchos tipos de entornos, aunque pueden dividirse groseramente en entornos discretos o continuos. Esta distinción es esencial dado que restringe el tipo de reglas que el agente debe tener. Si el tiempo o el espacio fueran tratados de manera continua, y el agente fuera a seleccionar la acción de girar hacia su izquierda sobre la base de sus percepciones y su estado interno, seleccionaría esta acción hasta que la información que perciba cambie, especialmente si girar hacia la izquierda también fuera una acción continua. Si el tiempo y el espacio fueran tratados de manera discreta, sería suficiente seleccionar, por ejemplo, una acción discreta como girar 90° hacia la izquierda. En ciertos contextos continuos, algunas técnicas de teoría de control pueden implementarse para ejecutar acciones consideradas espacial y temporalmente discretas (LaValle, 2009; Correll, 2014).

Incluso aunque no todo entorno deba ser modelado como un entorno discreto, muchos problemas de navegación pueden reducirse a una representación discreta (Correll, 2014, pp. 69-80). De los entornos discretos, en este trabajo sólo analizaremos los entornos rectangulares bidimensionales, que se estructuran como una grilla con celdas cuadradas, todas del mismo tamaño y, por el resto de este trabajo, consideradas del tamaño de una unidad. Son posibles muchas otras estructuras para un problema bidimensional, pero ésta es una estructura común que puede generalizarse fácilmente.

En lo que sigue, se define que una celda puede considerarse o bien libre o bien bloqueada, de manera que un agente sólo puede moverse por celdas libres. Las celdas en las que un agente comienza y termina su recorrido deben ser libres, pero además deben estar conectadas por algún camino, *i.e.* debe haber una o más secuencias de celdas libres adyacentes. En particular, la topología del entorno debe ser tal que un entorno pueda tener dos o más caminos entre el comienzo y la meta que no sean redundantes; si éste no fuera el caso, existen estrategias que siempre tendrán éxito y el problema sería trivialmente resoluble (Kilibarda *et al.*, 2003). El caso extremo donde un camino no redundante exista

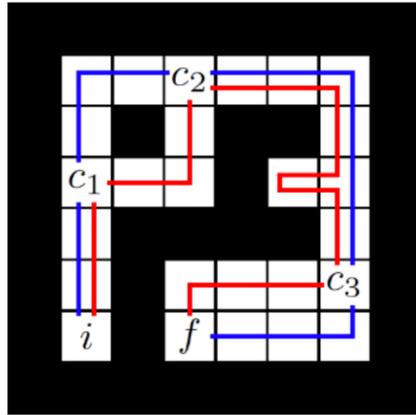


Fig. 1. Ejemplo de entorno discreto de tamaño 6 (no se cuentan las celdas fronteras). Se destacan las celdas de inicio (i) y de final (f). Las celdas negras son celdas bloqueadas y las blancas son celdas libres. Además de la frontera, tiene una única isla (con una sola celda bloqueada). Los caminos entre las celdas c_1 y c_2 , y entre c_3 y f no son redundantes. El camino rojo entre c_2 y c_3 sí es redundante, mientras que el azul no.

es el caso en el que existen celdas bloqueadas que no estén conectadas, *i.e.* en contacto, entre sí (como se ejemplifica en Figura 1). Las celdas más alejadas del centro del entorno deben ser siempre celdas bloqueadas; se las denomina celdas fronteras y conforman la frontera del entorno. Toda celda conectada a una celda frontera es también una celda frontera. Todo otro grupo de celdas bloqueadas conectadas entre sí se denomina isla.

Como se anticipó, en este trabajo se analiza el éxito o fracaso de ciertos agentes basados en modelos para resolver el problema de navegar por un entorno discreto rectangular para ir desde una celda determinada a otra. Se considera únicamente un conjunto de entradas mínimo, como sensores que chequean si la celda adyacente en cierta dirección está libre o bloqueada. Las acciones consideradas también pertenecen a un conjunto reducido, como moverse hacia adelante una celda, girar a la izquierda 90° o girar a la derecha 90° . No se consideran acciones que pudieran afectar al entorno de ningún modo. Formalmente, otras acciones podrían considerarse, como hacer un giro de 180° o moverse hacia adelante dos celdas; nótese que estas últimas pueden definirse a partir de las primeras, aunque para ello se requerirían estados internos adicionales.

Los agentes reactivos basados en modelos aquí presentes se describen del mismo modo que una máquina de estados finita, *i.e.* definiendo una función de salida y una función de transición entre estados internos (Isasi *et al*, 1997, c. 3). La función de salida selecciona una acción a realizar sobre la base de sus entradas y estado interno actuales, y la de transición selecciona el estado interno correspondiente sobre la misma base. El alfabeto de entrada de las máquinas de estados finitos depende de la manera en la que sus entradas o sensores codifican su información. Todas las máquinas de estado consideradas en este trabajo tienen cuatro sensores que codifican si la celda adyacente en cierta dirección está libre, con un '0', o bloqueada, con un '1', de manera que la secuencia de entradas es una secuencia <sensor frontal, sensor derecho, sensor trasero, sensor izquierdo>. De este modo, si un agente estuviera en una celda tal que sólo tiene una celda libre adyacente detrás de sí, la codificación de sus entradas consistiría en la secuencia <1101>, y si estuviera rodeada de celdas libres consistiría en la secuencia <0000>. Algo similar ocurre con el alfabeto de salida, que codifica las acciones posibles. Las máquinas consideradas aquí codifican las acciones para moverse una celda hacia adelante, girar a la derecha 90° , girar a la izquierda 90° y mantenerse quietas, codificadas por los símbolos \uparrow , \curvearrowright , \curvearrowleft y \times .

En primer lugar se proponen dos máquinas de estados finitos similares: MEF1 y MEF2. MEF1 comienza moviéndose de manera tal que posicione una celda bloqueada a su derecha, perteneciente a la frontera o a una isla. Luego avanza manteniendo las celdas de

la frontera o isla a su derecha, avanzando si se pudiera, girando a la izquierda si no. Si por adelantarse pierde ese grupo de celdas, realiza un giro a su derecha, avanza un casillero y retoma su comportamiento anterior. MEF2 es similar, pero las celdas a las que sigue las posiciona a su izquierda. Su definición formal se encuentra en la Tabla 1, de manera que las funciones de transición y salida se representan por lo que devuelven al recibir como argumentos las secuencias de entrada correspondiente a cada fila y el estado interno correspondiente a cada columna¹.

En segundo lugar, se proponen otras dos máquinas también similares entre sí: MEF3 y MEF4, definidas en la Tabla 2. Ambas se mueven hacia adelante tanto como puedan, o giran para un costado en caso de que sea imposible (cuando tanto a la derecha como a la izquierda haya un camino libre, una máquina va a hacia la derecha y la otra hacia la izquierda), excepto en el caso en el que se encuentren rodeadas por delante y por los costados de celdas bloqueadas. En ese caso, ambas máquinas cambian de estado interno a uno en el que se realiza un giro hacia atrás (dos hacia la izquierda o dos hacia la derecha) y luego se avanza similarmente al estado anterior, hasta que se llegue a una celda que además de estar libre hacia adelante, esté libre hacia algún costado; en ese caso, se vuelve al estado interno inicial tras girar hacia el costado correspondiente (de haber dos caminos posibles, cada máquina prioriza uno distinto).

Tabla 1. Funciones de transición y salida de MEF1 y MEF2

Entradas	Estados de MEF 1			Estados de MEF 2		
	A	B	C	A	B	C
0000	A ↑	C ↷	B ↑	A ↑	C ↷	B ↑
0001	A ↷	C ↷	B ↑	B ↑	B ↑	-
0010	A ↷	-	B ↑	A ↷	-	B ↑
0011	A ↷	-	B ↑	B ↑	B ↑	-
0100	B ↑	B ↑	-	A ↷	C ↷	B ↑
0101	B ↑	B ↑	-	B ↑	B ↑	-
0110	B ↑	B ↑	-	A ↷	-	B ↑
0111	B ↑	B ↑	-	B ↑	B ↑	-
1000	A ↷	C ↷	-	A ↷	C ↷	-
1001	A ↷	C ↷	-	A ↷	B ↷	-
1010	A ↷	-	-	A ↷	-	-
1011	A ↷	-	-	A ↷	B ↷	-
1100	A ↷	B ↷	-	A ↷	C ↷	-
1101	A ↷	B ↷	-	A ↷	B ↷	-
1110	A ↷	B ↷	-	A ↷	-	-
1111	A ×	-	-	A ×	-	-

Tabla 2. Funciones de transición y salida de MEF3 y MEF4

Entradas	Estados de MEF3		Estados de MEF 4	
	A	B	A	B
0000	A ↑	A ↷	A ↑	A ↷
0001	A ↑	A ↷	A ↑	A ↷
0010	A ↑	-	A ↑	-
0011	A ↑	-	A ↑	-
0100	A ↑	A ↷	A ↑	A ↷
0101	A ↑	B ↑	A ↑	B ↑
0110	A ↑	-	A ↑	-
0111	A ↑	B ↑	A ↑	B ↑
1000	A ↷	A ↷	A ↷	A ↷
1001	A ↷	A ↷	A ↷	A ↷
1010	A ↷	-	A ↷	-
1011	A ↷	-	A ↷	-
1100	A ↷	A ↷	A ↷	A ↷
1101	B ↷	B ↷	B ↷	B ↷
1110	A ↷	B ↷	A ↷	B ↷
1111	A ×	-	A ×	-

¹ Ciertas combinaciones de entradas y estados internos no son posibles, dada la definición de los entornos y el efecto de cada acción, por lo que las funciones no se definen para ellas, lo que facilita la lectura de las tablas.

Tabla 3. Funciones de transición y salida de MEF5

Entradas	Estados de MEF 5						
	A	B	C	D	E	F	G
0000	B ↑	B ↑	-	B ↑	B ↘	B ↑	-
0001	B ↑	B ↑	D ↑	B ↑	F ↘	B ↑	B ↑
0010	A ↑	B ↑	-	D ↑	-	F ↑	-
0011	A ↑	B ↑	-	D ↑	E ↑	F ↑	G ↑
0100	B ↑	B ↑	D ↑	B ↑	D ↘	B ↑	B ↑
0101	A ↑	B ↑	-	D ↑	E ↑	F ↑	G ↑
0110	A ↑	B ↑	-	D ↑	E ↑	F ↑	G ↑
0111	A ↑	-	-	-	E ↑	F ↑	G ↑
1000	C ↘	C ↘	-	C ↘	C ↘	C ↘	C ↘
1001	A ↘	B ↘	-	D ↘	E ↘	F ↘	G ↘
1010	A ↘	-	-	-	-	-	-
1011	A ↘	-	-	-	-	-	-
1100	A ↘	B ↘	-	D ↘	E ↘	F ↘	G ↘
1101	A ↘	B ↘	-	D ↘	E ↘	F ↘	G ↘
1110	A ↘	-	-	-	E ↘	F ↘	G ↘
1111	A ×	-	-	-	-	-	-

La última máquina de estados finitos a analizar, MEF5, tiene un comportamiento más complejo. Como las dos anteriores, avanza mientras puede, con los giros que le permitan seguir avanzando, hasta que llegue a una celda con más de un camino posible. Si hay más de un camino libre, tomará uno de ellos hasta que llegue a un camino sin salida (rodeado por celdas bloqueadas) u otra celda con varios caminos posibles. En el primer caso se volverá sobre sí (con un giro de 180°) hasta regresar a la celda en la que se tomó la decisión; de regreso en esta celda aprovechará sus estados internos para evitar tomar el mismo camino más de una vez y explorar los caminos que no haya recorrido. En el segundo caso, en el que se encuentran dos o más bifurcaciones consecutivas, el estado interno de MEF5 se corresponderá con la situación de haber encontrado una bifurcación por primera vez. La definición formal de MEF5 se representa en la Tabla 3.

Las máquinas son evaluadas de acuerdo a sus probabilidades de éxito en un entorno dado. Es claro que una máquina de estados finitos tiene poca información sobre su entorno, y que es principalmente local. Esto puede causar que una máquina falle en alcanzar su meta por distintas razones. En el caso de MEF1 y MEF2, saldrían exitosas sólo si la meta es una celda adyacente a la misma isla (o a la frontera) que la celda inicial del agente: estas máquinas no pueden abandonar una isla (dado que su comportamiento consiste en rodearlas) y nunca encontrarían un camino hasta la meta (repetirían cierta secuencia de movimientos una y otra vez). Puede razonarse de forma análoga acerca de las otras máquinas propuestas, pero su diversidad y complejidad impide un análisis general.

Sin embargo, el tamaño del entorno y la longitud del camino mínimo requerido para alcanzar la meta parecen relevantes. Estos factores se vinculan con la distribución de ciertas estructuras en el espacio de entornos posibles, lo que condiciona el éxito de una máquina dada. El tamaño del entorno se define como la cantidad de filas o columnas (la misma si son entornos cuadrados). La longitud del camino mínimo se define por las ubicaciones de las celdas de inicio y finalización, y dado que las celdas de un entorno pueden considerarse como los nodos de un grafo, la longitud de este camino puede relacionarse con el diámetro del grafo (Kilibarda et al., 2003). Por lo tanto, la probabilidad de éxito se tendrá en cuenta con respecto al tamaño de los entornos, y la mínima proporción del diámetro promedio del entorno (respecto a su tamaño) que podría necesitar un agente para alcanzar la meta.

Metodología

Las máquinas de estados finitos son puestas a prueba sobre una colección de entornos generados aleatoriamente. Estos entornos varían en tamaño y en la longitud del camino mínimo entre la celda de inicio y la meta. Todo entorno tiene la misma cantidad de filas que columnas, que varían entre 10 y 60 en saltos de 10. La longitud del camino mínimo varía entre el 10 % y el 90 % (en saltos de 10%) del diámetro promedio para entornos de cada tamaño. Para cada combinación de tamaño y longitud de camino mínimo, se generaron 100 entornos para evaluar la tasa de éxito de cada máquina.

Generación de entornos aleatorios

La generación aleatoria de entornos se basa en el algoritmo *Cellular Automaton* de John Stilley, presentado en su paquete de *Python mazelib* (Stilley, 2023). Este algoritmo toma una grilla de T filas y columnas, y luego genera una nueva grilla de $2 \times T + 1$ filas y columnas. Esto se debe a que la nueva grilla representa un entorno de tamaño T , cuyas celdas son todas libres, pero que entre celda y celda puede o no haber un camino directo, lo que se logra agregando celdas libres o bloqueadas, respectivamente. Para ello es necesario insertar filas y columnas alrededor de las originales.

Al comienzo de este algoritmo, las celdas nuevas son celdas libres (excepto las que conforman la frontera, que son celdas bloqueadas). Luego, el algoritmo atraviesa dos etapas. En la primera, selecciona algunas de las nuevas celdas adyacentes a la frontera y las convierte en celdas bloqueadas. En la segunda etapa, procede de forma similar a un autómata celular, pero seleccionando una celda por ciclo. Comienza con una celda aleatoria de las nuevas celdas y, si está libre y tiene exactamente tres celdas vecinas bloqueadas, entonces el algoritmo la vuelve una celda bloqueada, pero si la celda está bloqueada y no tiene entre una y tres celdas vecinas, se libera. Finalmente, el algoritmo selecciona una de las celdas vecinas y repite el proceso.

Este algoritmo cumple con los desiderata indicados anteriormente para entornos discretos. En particular, genera entornos que usualmente tienen islas (se vuelven más probables cuando el tamaño del entorno se incrementa). No obstante, modifica el tamaño inicial del entorno, por lo que se requerirá tener en cuenta redimensionar las mediciones realizadas respecto del diámetro.

Determinación de las propiedades topológicas de los entornos

El diámetro promedio de los entornos generados se obtiene experimentalmente. Se generó un conjunto de 32 entornos de tamaño 10 a 60 en saltos de 10. Para cada entorno se calculó la excentricidad de cada celda, *i.e.* el máximo de las distancias mínimas para cada otra celda. El máximo de las excentricidades de todas las celdas de un entorno consiste en su diámetro. Esto arroja un diámetro experimental para cada uno de los tamaños de entornos señalados. Los diámetros experimentales se emplean para obtener una aproximación (lineal) de la relación entre el diámetro promedio de un entorno y su tamaño. Esta medida debe dividirse en dos para compensar el cambio en las dimensiones del entorno al usar el generador mencionado.

Además del diámetro, la excentricidad de cada celda también se usa para registrar la excentricidad promedio de las celdas previas a la expansión que realiza el generador aleatorio. Dado que la generación de entornos simétricos es muy improbable, para cada tamaño se generaron nuevos entornos como resultado de rotar los 32 iniciales (90°, 180° y 270°); se repitió el proceso con sus versiones espejadas. Estas transformaciones resultaron en una muestra de 256 entornos usados para computar la excentricidad promedio de cada celda. Estos resultados también deben dividirse por dos debido a la expansión de celdas que fuerza el generador. Usando esta información, se puede obtener un modelo

para conocer la probabilidad de que haya un par de celdas a cierta distancia mínima.

Generación de entornos con recorrido mínimo fijo

Con esa información es posible generar conjuntos de entornos con un tamaño fijo y una distancia mínima entre el inicio y la meta previamente especificada.

Después de que un entorno de cierto tamaño es generado, una celda aleatoria (de las representadas antes de la expansión que hace el generador) es seleccionada para servir como celda inicial.

Sin embargo, no toda celda es adecuada para ser la celda inicial, dado que podría estar “muy conectada” con el resto del entorno, *i.e.* la excentricidad de la celda podría ser menor que la distancia requerida, y entonces no habría ninguna otra celda adecuada para ser la celda final. En lugar de elegir celdas y evaluar una por una si hay otras celdas a la distancia especificada, la selección de las celdas depende de la excentricidad promedio que se obtuvieron en los modelos anteriores. Estos son usados para saber qué celdas tienen menos probabilidades de tener otras celdas a la distancia requerida; concretamente se impide la selección de celdas que tengan una excentricidad promedio menor a la distancia requerida.

A continuación de la selección de la celda inicial, se analiza el entorno para identificar celdas a la distancia requerida. Si hay una o más celdas a esta distancia, se selecciona una de ellas al azar para ser la meta; en caso contrario, la celda seleccionada como inicial es descartada, se selecciona una nueva celda inicial y se repite el proceso.

Determinación de la tasa de éxitos

Una vez que el conjunto de entornos de evaluación es generado, un simulador selecciona un entorno, luego selecciona, instancia y corre una máquina de las definidas, *i.e.* itera sus funciones de salida y transición hasta reconocer cierta condición. La simulación de una máquina termina siempre que alcance la meta (éxito) o repita una combinación de estado interno, ubicación y orientación en el entorno (fallo); esto último se debe a que el agente habrá entrado en un ciclo que repetirá una y otra vez, sin pasar por la meta. Esto se hace para cada máquina de estados finitos definida, y para cada entorno del conjunto.

La tasa de éxitos para cada agente se obtiene como la suma de las simulaciones exitosas; se discrimina tanto por el tamaño del entorno como por la proporción del diámetro promedio de entornos de ese tamaño que fuera seleccionada como distancia entre la celda inicial y la meta.

Resultados

Determinación de las propiedades topológicas de los entornos

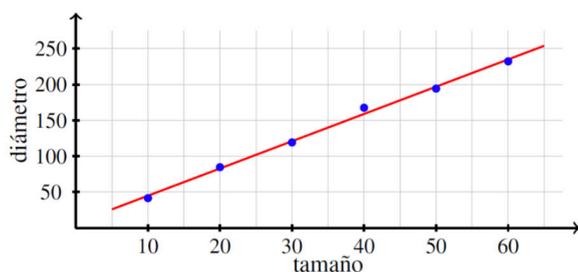


Fig. 2. Diámetro promedio según tamaño de los entornos. El promedio para cada tamaño se representa con un punto azul, y la aproximación lineal entre ambas características se representa con una línea roja.

El diámetro promedio relacionado con el tamaño de los entornos se muestra en la Figura 2. También se muestra una aproximación lineal para esta relación, indicando su tendencia, en (1); el error cuadrático medio de la aproximación es de 17,8 celdas.

$$(1) \text{ diámetro promedio} = 6,9 + 3,8 \times \text{tamaño}$$

La excentricidad promedio de cada celda para entornos con tamaños de 10 a 60, en saltos de 10, se muestra en las Figuras 3 a 5. Se agregaron algunos delimitadores para resaltar las zonas donde se encuentran ciertos valores de excentricidad. Puede verse que las longitudes de los caminos que se corresponden con mayores proporciones del diámetro promedio están más próximas a la frontera.

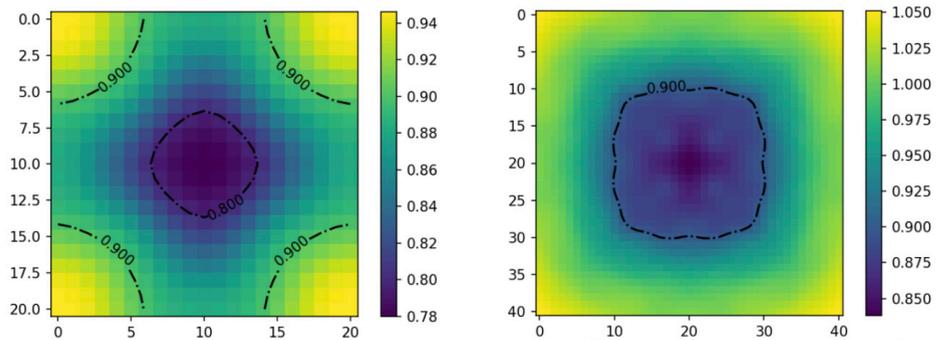


Fig. 3. Excentricidad promedio de entornos de tamaño 10 (izquierda) y de tamaño 20 (derecha). A ambas gráficas se les aplicó un filtro gaussiano.

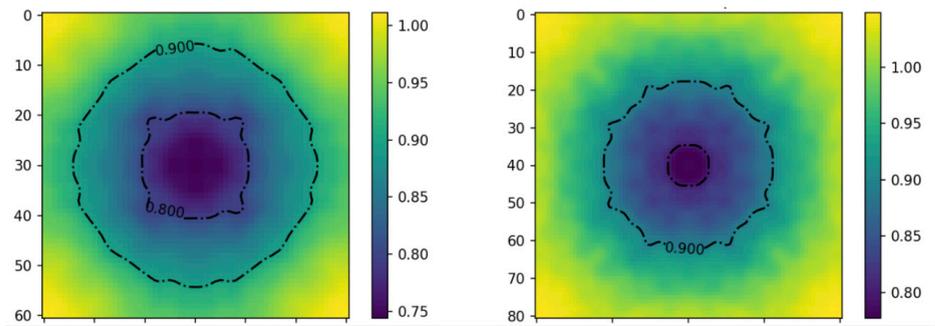


Fig. 4. Excentricidad promedio de entornos de tamaño 30 (izquierda) y de tamaño 40 (derecha). A ambas gráficas se les aplicó un filtro gaussiano.

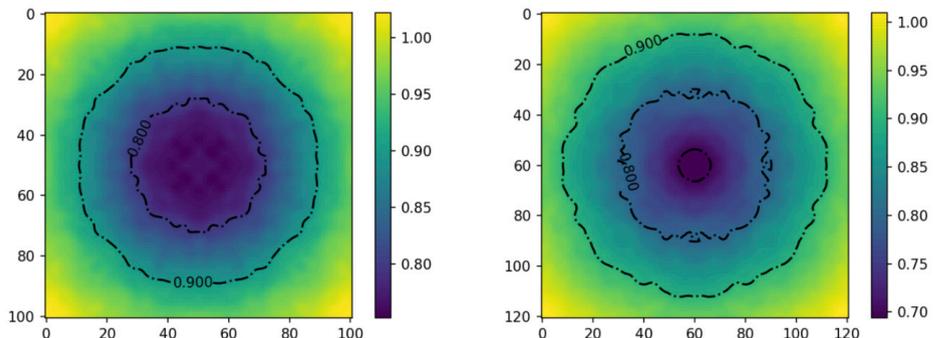


Fig. 5. Excentricidad promedio de entornos de tamaño 50 (izquierda) y de tamaño 60 (derecha). A ambas gráficas se les aplicó un filtro gaussiano.

Determinación de la tasa de éxitos

La longitud del camino elegido para cada punto de evaluación se muestra en la Tabla 4 como una función dependiente del tamaño y proporción del diámetro promedio (para el tamaño correspondiente).

Las tasas de éxito de cada máquina de estados finitos se muestran en las Tablas 5 a 9. Cada celda representa la tasa de éxito correspondiente a entornos de cierto tamaño (fila) y con una distancia mínima entre la celda inicial y la meta igual a cierta proporción del diámetro promedio correspondiente a entornos de ese tamaño (columna), como se indica en la Tabla 4. Las celdas de cada tabla están gradualmente coloreadas desde rojo hasta blanco, para valores bajos y altos, respectivamente.

Tabla 4. Puntos de evaluación para cada combinación de tamaño y proporción del diámetro promedio correspondiente

Tamaño	Proporción del diámetro								
	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
10	4	9	13	18	22	27	31	36	40
20	8	17	25	33	41	50	58	66	75
30	12	24	36	48	60	73	85	97	109
40	16	32	48	64	79	95	111	127	143
50	20	39	59	79	98	118	138	158	177
60	23	47	70	94	117	141	164	188	211

Tabla 5. Tasa de éxitos para MEF1, según tamaño de los entornos evaluados y distancia a recorrer

Tamaño	Proporción del diámetro								
	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
10	98	96	96	98	98	100	98	100	100
20	86	84	83	87	87	86	92	97	99
30	65	72	81	68	79	77	86	91	96
40	59	52	57	53	70	68	72	84	87
50	63	52	41	47	57	64	73	81	87
60	52	41	29	33	51	59	69	79	88

Tabla 6. Tasa de éxitos para MEF2, según tamaño de los entornos evaluados y distancia a recorrer

Tamaño	Proporción del diámetro								
	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
10	97	95	94	97	97	100	98	100	100
20	87	87	85	86	86	87	91	97	99
30	68	73	83	68	81	77	83	91	98
40	60	50	56	53	63	65	73	87	86
50	60	53	41	45	57	66	73	80	86
60	51	39	31	32	47	59	71	78	91

Tabla 7. Tasa de éxitos para MEF3, según tamaño de los entornos evaluados y distancia a recorrer

Tamaño	Proporción del diámetro								
	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
10	60	45	21	12	7	6	5	1	2
20	33	11	5	3	0	0	0	0	0
30	14	4	1	0	0	0	0	0	0
40	17	1	0	0	0	0	0	0	0
50	11	0	0	0	0	0	0	0	0
60	3	0	0	0	0	0	0	0	0

Tabla 8. Tasa de éxitos para MEF4, según tamaño de los entornos evaluados y distancia a recorrer

Tamaño	Proporción del diámetro								
	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
10	61	39	23	18	7	7	3	1	3
20	34	10	5	4	0	0	0	0	0
30	18	6	0	0	0	0	0	0	0
40	12	2	0	0	0	0	0	0	0
50	4	0	0	0	0	0	0	0	0
60	7	0	0	0	0	0	0	0	0

Tabla 9. Tasa de éxitos para MEF5, según tamaño de los entornos evaluados y distancia a recorrer

Tamaño	Proporción del diámetro								
	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
10	75	60	39	39	24	28	25	20	21
20	51	23	12	8	2	2	2	0	3
30	23	13	3	3	0	1	0	0	0
40	33	3	2	0	0	1	0	0	0
50	18	3	0	1	0	0	1	0	0
60	6	2	1	0	0	0	0	0	0

Discusión

El análisis del diámetro promedio como dependiente del tamaño de los entornos presenta un comportamiento lineal para la muestra generada. Considerando que el diámetro representa la mayor de las distancias posibles entre dos celdas cualesquiera, es interesante que el diámetro crezca linealmente con el tamaño del entorno (indistintamente, cantidad de filas o columnas), dado que la cantidad de celdas crece cuadráticamente. Lamentablemente, por cuestiones de costo computacional, no pudo extenderse el rango de la muestra para incluir entornos discretos de mayor tamaño que permitieran determinar si para entornos más grandes conviene trabajar con una aproximación de mayor grado.

Con respecto a las excentricidades medias obtenidas, para aquellos entornos de mayor tamaño (ver Figuras 4 y 5) se observan distribuciones similares, con la menor excentricidad en el centro, expandiéndose hacia las fronteras con un patrón circular. Sin embargo, para los entornos de menor tamaño (ver Figura 3), aunque también se observa que la menor excentricidad se encuentra en el centro, su distribución es más heterogénea, demandando análisis adicionales para explicarla. Esto motiva la generación de entornos de menor dimensión, para determinar si las distribuciones de la excentricidad promedio se vinculan a una familia de superficies que expresan un comportamiento variable dependiente de algún patrón que permita caracterizar al generador aleatorio de entornos de manera más abstracta.

El análisis de las máquinas de estados finitos definidas resultó en varios puntos de interés. En primer lugar, se puede notar que MEF1 y MEF2 tienen tasas de éxito altas para entornos de menor tamaño, que decrecen para entornos de tamaño mayor. También puede observarse que tienen una tasa de éxito alta cuando la proporción del diámetro promedio a recorrer es o muy baja o muy alta, mientras que las peores tasas de éxitos se encuentran alrededor de distancias que se corresponden con proporciones del 30 % del diámetro. Esto motiva un muestreo de mayor resolución de manera que pueda caracterizarse el desempeño general de estas máquinas.

En segundo lugar, las tasas de éxito de MEF3 y MEF4 son menores, y decrecen rápidamente cuando la proporción del diámetro a recorrer aumenta. Algo similar ocurre con MEF5, pero en menor medida. Si bien son comportamientos que parecen ir acompañados de simplemente un aumento en la longitud absoluta (en oposición a una proporción del diámetro promedio para un tamaño específico) de los caminos a recorrer, es de esperar que sea el resultado de que ciertos patrones en la topología de los entornos son más probables cuando los caminos son más extensos.

Finalmente, tal como se esperaba, máquinas de estados finitos simétricas obtuvieron tasas de éxito similares. Esto ocurre con MEF1 y MEF2 que varían esencialmente en cuál de las dos secuencias de celdas bloqueadas seguirán, si la de su derecha o la de su izquierda, y con MEF3 y MEF4 que varían en qué celdas libres explorarán cuando hay más de una opción disponible. Este resultado refuerza el supuesto de que la técnica de generación de entornos muestrea de manera satisfactoria, al menos en términos de simetría.

Conclusión

A la luz de estos resultados, algunas máquinas de estados finitos parecen mostrar mayores tasas de éxito sobre el espacio de evaluación, tal es el caso de MEF1 y MEF2, en oposición a MEF3, MEF4 y MEF5. Esto es así a pesar del conjunto mínimo de acciones y entradas de percepción disponibles en las definiciones de máquinas de estados finitos (un límite común en las máquinas físicas). El éxito de MEF1 y MEF2 parece ser alto siempre que la longitud mínima del camino entre las celdas inicial y final sea menor o mayor que un 30 % del diámetro promedio de un entorno de cierto tamaño. Esta tasa está fuertemente relacionada con las características topológicas de los entornos, y otras topologías darían como resultado tasas diferentes que deben explorarse.

Agradecimientos

Los autores le agradecen a Alejandro Gastón Álvarez por sus valiosos comentarios y revisiones, y al Grupo de Inteligencia Artificial y Robótica (FRBA-UTN) por el espacio para discutir estas ideas.

Referencias

- ALBERINO, S.; SAMPIERI, F.; FOLINO, P.; GÓMEZ, J. C. y VERRASTRO, C. A., (2015), “Programación de robots utilizando hoja de cálculo”, en Actas de las VIII Jornadas Argentinas de Robótica, Buenos Aires, Argentina.
- ALBERINO, S.; TORANZO CALDERÓN, J. S.; GÓMEZ, J. C. y VERRASTRO, C. A., (2019), “Método de programación para robótica didáctica”, en Actas de las X Jornadas Argentinas de Robótica, Neuquén, Argentina.
- ÁLVAREZ, A. G.; MARUFO DA SILVA, M.; SAMPIERI, F.; PIRLO, D.; GÓMEZ GABIS, J. M.; DI SABATO, P.; MARCOVECCHIO, N.; CISMONTI, L.; FOLINO, P.; ALBERINO, S.; VERRASTRO, C. y GÓMEZ, J. C., (2013), “Minipi: Plataforma inteligente de tamaño reducido”, en Libro de Actas XV-RPIC (Reunión de Trabajo en Procesamiento de la Información y Control), San Carlos de Bariloche, Río Negro, Argentina, pp. 519–524.
- CORRELL, N., (2014), Introduction to Autonomous Robots, CreateSpace Independent Publishing Platform.
- ISASI, P.; MARTÍNEZ, P. y BORRAJO, D., (1997), Lenguajes, gramáticas y autómatas. Un enfoque práctico, Addison-Wesley, 1997.
- KILIBARDA, G.; KUDRYAVTSEV, V. B. y UŠČUMLIĆ, Š., (2003), “Independent systems of automata in labyrinths”, Discrete Mathematics and Applications, 13 (3), pp. 221–255.
- LAVALLE, S. M., (2006), Planning Algorithms, Cambridge, U.K.: Cambridge University Press.
- RUSSELL, S. J. y NORVIG, P., (2009), Artificial Intelligence: a modern approach, 3^a ed., Prentice Hall.
- STILLEY, J., (2023), “mazelib” (v. 0.9.14), en Python Package Index (PyPI), Python Software Foundation. Disponible en <https://pypi.org/project/mazelib/> y consultado en diciembre de 2023.