

Paso de Mensajes

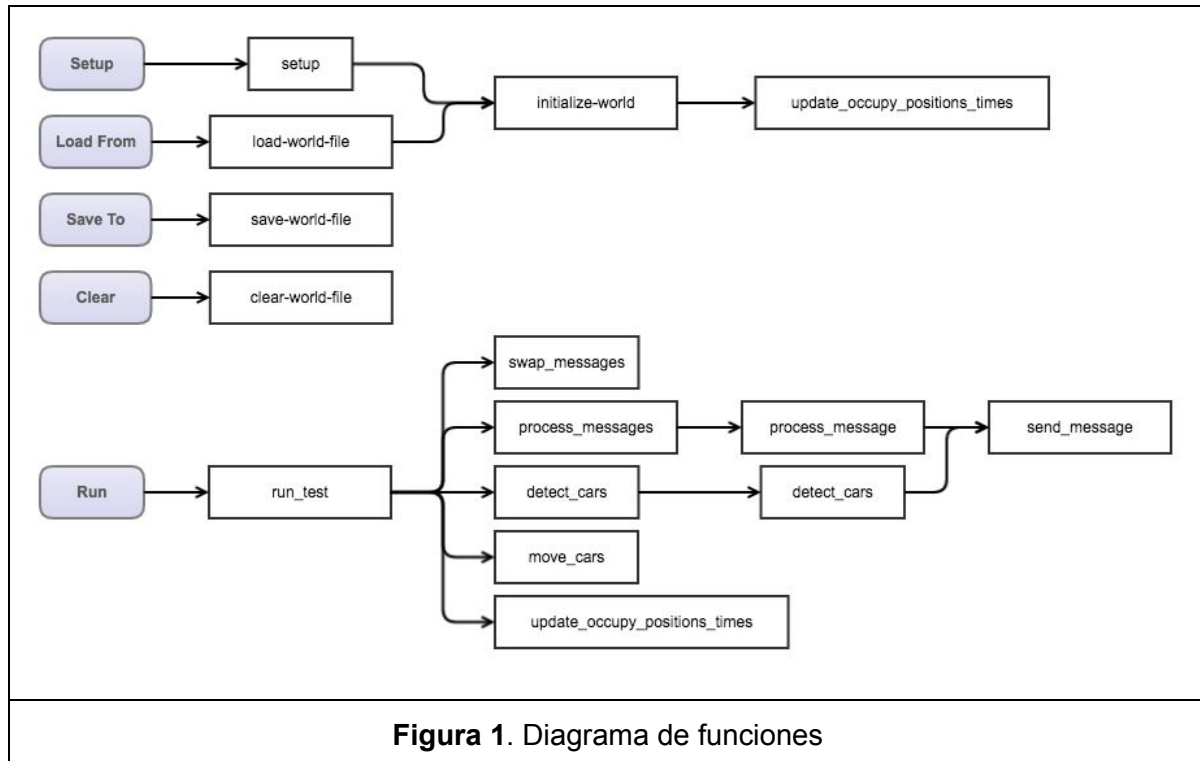
Sesión 5

Informe de Práctica

Juan Marin Vega
Oriol López Sánchez

Implementación

Para esta sesión, hemos implementado el paso de mensajes y la lógica de negocio pedida por el enunciado de la siguiente manera:



Para cada uno de los botones que se presenta en el modelo se han añadido las funciones llamadas, que implementan las siguientes funcionalidades:

- Setup. Inicialización de una configuración de coches, e inicialización de gráficas.
- Carga y guardado de ficheros de configuración de coches mediante los botones “Load From” y “Save To” respectivamente.
- Hemos añadido una función llamada **calc_gradient** que calcula el gradiente entre dos colores, en este caso verde y rojo. Cuanto mas verde sea el color mas alto es el valor ProbOcuparPos. Cuanto más rojo, menor es la probabilidad.

```
to-report calc_gradient[ base-color ]
;; color gradient between red and green
report (list (255 - (255 / 100. * base-color)) (255 / 100. * base-color) 0)
end
```

Codigo1. Implementación del cálculo del gradiente

La simulación funciona en tres etapas:

1. Procesamiento de mensajes entre coches, mediante las funciones **swap_messages** y **process_mesages**.
2. Ejecución de la simulación mediante las funciones **detect_cars** y **move_cars**.

- Actualización de los histogramas mediante la función **update_occupy_positions_times**.

Protocolo de mensajes

Estructura del mensaje

La estructura de los mensajes enviados por los coches es una lista que contiene los valores descritos en la siguiente figura:

Campo	Tipo	Descripción
Emisor del mensaje	Agente de tipo "car"	Como es obvio, indica quién ha enviado el mensaje.
Tipo de mensaje	String	Catalogación del mensaje para poder interpretar en qué fase de la comunicación se encuentra
Message	String	El mensaje en sí que el emisor desea enviar

Figura 2. Tabla de tipos de mensajes

Tipos de mensajes

Para nuestra simulación se ha propuesto un tipo de mensaje llamado **"PASS-THROUGH"** que contiene tres posibles mensajes:

- "ASK"**. Este mensaje es enviado por un coche a aquellos coches que intenta pasar a través de ellos.
- "YES"**. Este mensaje es enviado por aquellos coches que han recibido el mensaje **"ASK"** y pueden ceder el paso.
- "NO"**. Este mensaje es enviado por aquellos coches que han recibido el mensaje **"ASK"** y no permiten el ceda al paso.

Transmisión de mensajes

Para el paso de mensajes, la comunicación siempre comienza con un coche pidiendo el paso a otro en un tick mediante un **"ASK"**. Estas peticiones de mensajes quedan apiladas a la lista **"next-messages"** de los agentes **"car"**.

En el siguiente tick, los agentes intercambian dicha lista con la lista **"current-messages"** para procesar esos mensajes, en cuyo caso acaban enviando las respuestas para el mensaje **"ASK"**.

Por último, mientras tengan mensajes a procesar en ticks sucesivos, no intentarán moverse o detectar otros agentes. De esta manera, permitimos que las evaluaciones de los mensajes se puedan tratar a buen ritmo, sin que los coches se desplacen cuando no es debido.

Pruebas

Para analizar la influencia de asignar un número de energía diferente aleatorio a cada coche, hemos realizado dos pruebas.

1. Iniciamos todos los coches con la misma energía inicial
2. Iniciamos todos los coches con la energía inicial de forma aleatoria

Las dos pruebas siguientes se realizan con un cambio significativo: Mientras un coche tenga mensajes por procesar, no se moverá

- a. Iniciamos todos los coches con la misma energía inicial
- b. Iniciamos todos los coches con la energía inicial de forma aleatoria

Con estas dos pruebas también podemos analizar la influencia del valor ProbOcuparPos en el número de veces que un coche ocupa una posición de otro coche.

Segmentamos los coches en 3 grupos. Los que tienen ProbOcuparPos mayor de 60, ProbOcuparPos mayor o igual que 40 y menor o igual que 60, y el resto.

Resultados

1 - Todos los coches con la misma energía inicial

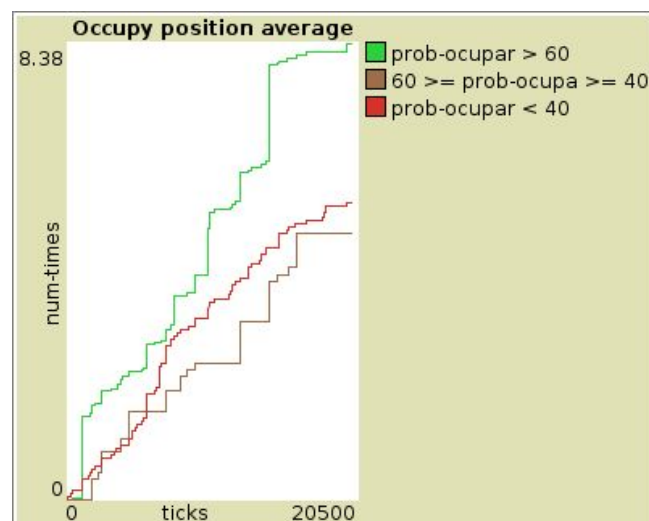


Figura 3. Ocupación media por grupo

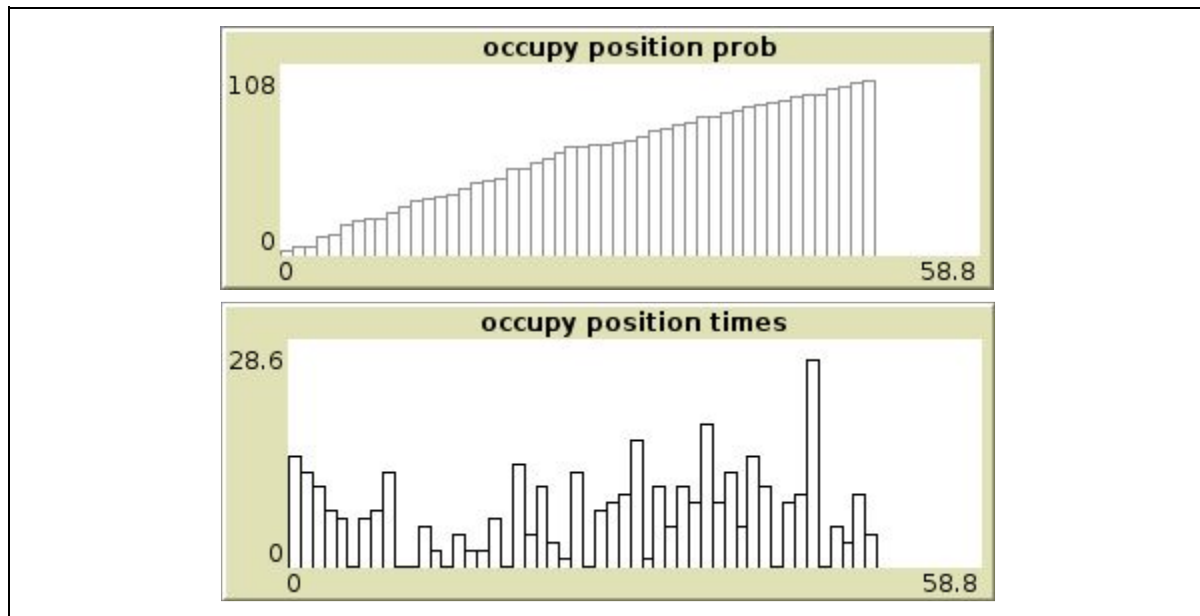


Figura 4. Número de veces que un coche ocupa una plaza una posición, frente a la probabilidad de ocupar una posición

A primera vista no se observa relación aparente, tras 20.000 ticks, entre la probabilidad de ocupar una posición y el número de posiciones ocupadas.

En la Figura 4 observamos dos gráficas, la primera superior representa la probabilidad de ocupar posición de cada coche, ordenada de forma ascendente. La inferior representa el número de posiciones que ha ocupado cada coche, ordenada por la probabilidad de ocupar posición, es decir, las barras inferiores y superiores representan al mismo agente.

Observamos que no hay una relación directa entre la probabilidad de ocupar posiciones y el número de posiciones que ocupa.

2 - Coches con diferente energía inicial

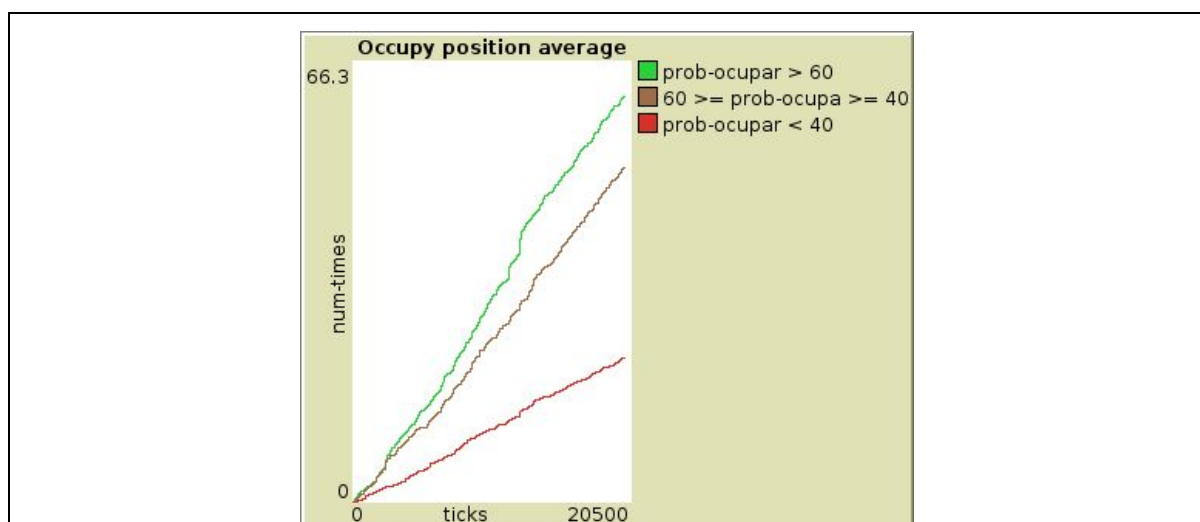
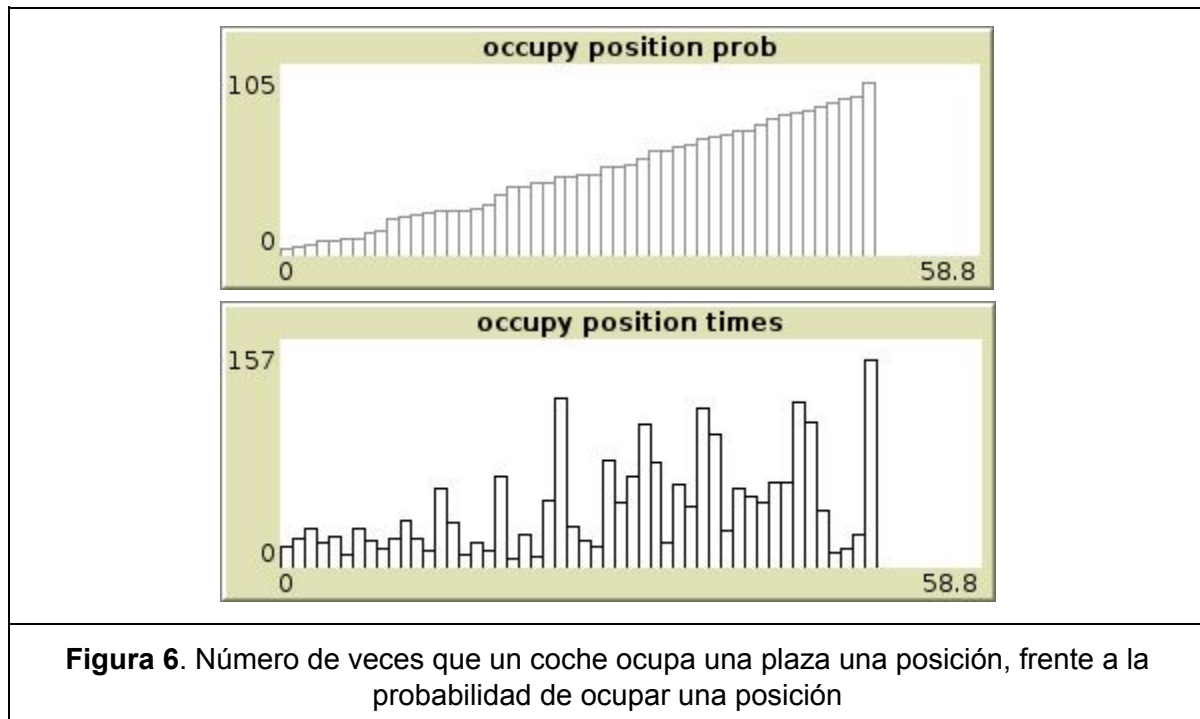


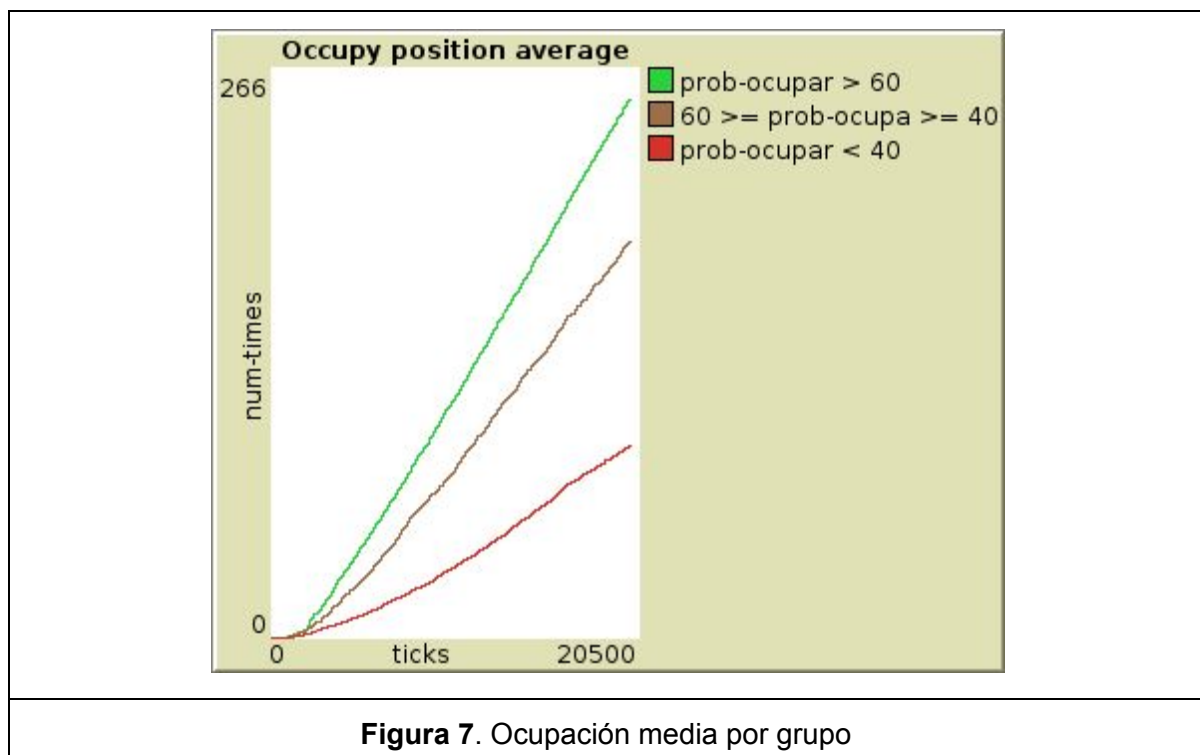
Figura 5. Ocupación media por grupo



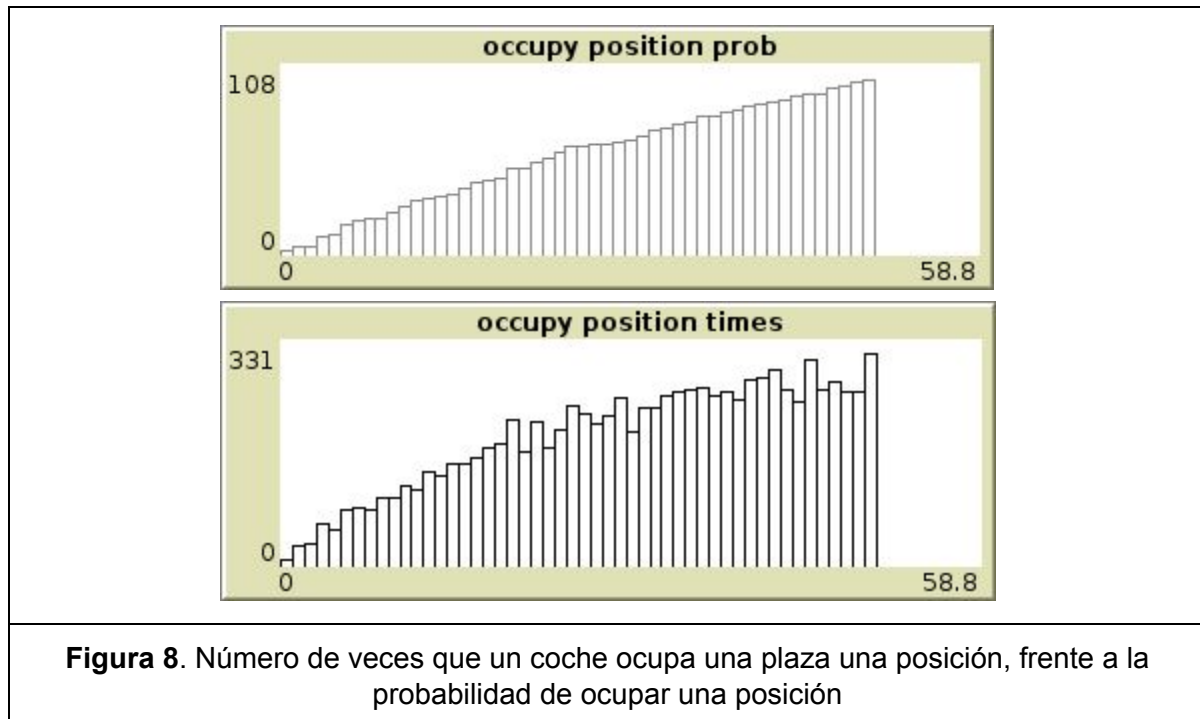
En este caso, la Figura 5 muestra que los grupos con mayores probabilidades ocupan una media de más posiciones. La Figura 6 muestra una leve línea ascendente en la grafica Occupy position times

a - Todos los coches con la misma energía inicial

Estos son los resultados que obtenemos al inicializar todos los coches con la misma energía inicial tras 2000 ticks

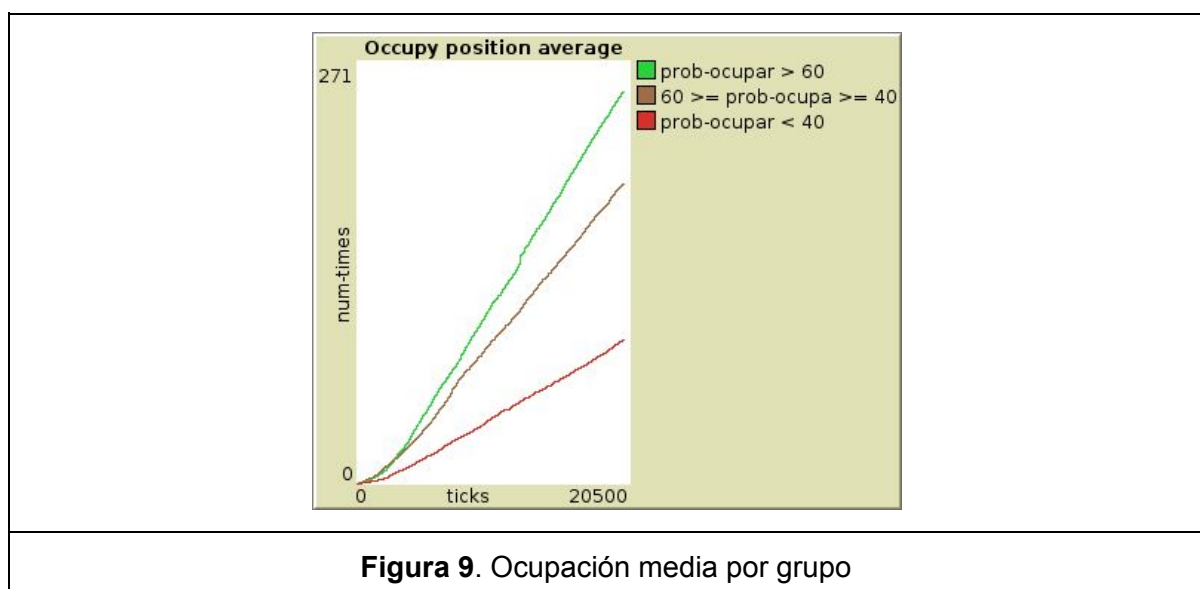


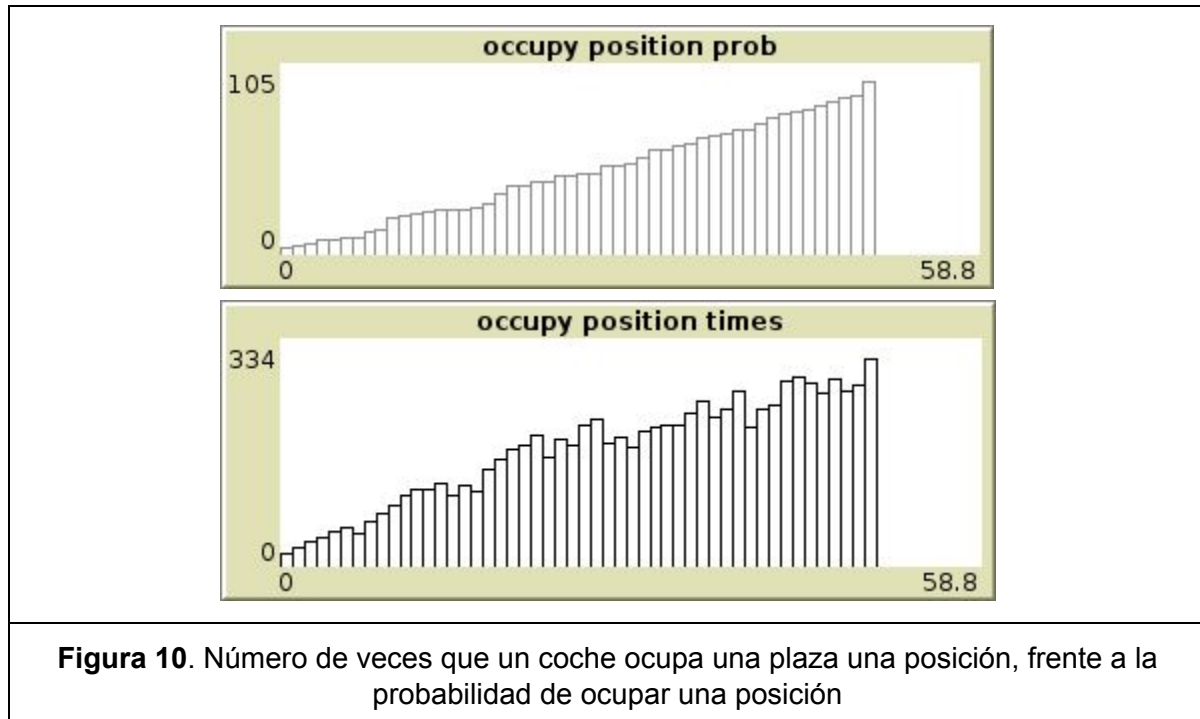
En la Figura 7 observamos que los grupos de coches definidos se comportan de la siguiente manera: Aquellos que tienen más probabilidades de ocupar posición, ocupan, en media, un mayor número de posiciones. Lo que habíamos observado en la Figura 5 pero de una forma mas contundente. Y lo mismo en la Figura 8, aquellos que tienen una probabilidad menor, ocupan menor número de posiciones.



Aquí ya observamos que hay una relación directa entre la probabilidad de ocupar posiciones y el número de posiciones que ocupa.

b - Coches con diferente energía inicial





Observamos a primera vista en la Figura 9 y 10 que esta configuración no presenta grandes diferencias con las figuras 7 y 8.

Análisis y conclusiones

Concluimos, pues, que inicializar los coches con diferente energía presenta diferencias respecto a cuando los iniciamos con diferentes energías iniciales. Esto puede ser debido a que los coches en movimiento tan solo ocupan posiciones de los coches parados, y al inicializar todos los coches con la misma energía, todos se paran a la vez y se mueven a la vez. El número de posiciones que se ocupa es muy reducido, y es en las situaciones en las que los coches se quedan sin energía unos cerca de otros. Cuando un coche está parado, puede detectar coches alrededor (no se especificaba lo contrario en la práctica), a si que mientras los dos coches están en reposo, uno de ellos puede ir solicitando la posición del otro de forma continuada, hasta que este responda afirmativamente.

En las prueba 2, vemos que al parar los coches de forma alterna, se producen más ocupaciones. El número es bajo, pero si aumentamos el tiempo de la prueba se obtienen resultados parecidos a los de las pruebas a y b.

En las pruebas a y b, entendemos que el hecho de que un coche no se mueva (y no reduzca su energía) mientras tenga mensajes por procesar, genera diferencias en los tiempos de movimiento y reposo, con lo cual los coches en movimiento se encuentran mas coches en reposo incrementando el número de posibles ocupaciones.

La influencia del parámetro ProbOcuparPos es mas fuerte en las pruebas 2, a y b, que en la prueba 1, donde no se aprecia prácticamente influencia, debido a que no se producen oportunidades de ocupación.

Los siguientes sliders incluidos facilitan las pruebas rapidas:

- **“NUM-CARS”**. Indica el número de coches con el que se inicia la simulación.
- **“BASE-ENERGY”**. Indica el número máximo de energía con el que inician los coches, si se crea una simulación mediante el botón “Setup”. Si se carga desde un fichero, será la especificada en este.
- **“T-DESCANSO”**. Indica la cantidad de ticks que se quedarán los coches parados repostando.

Así como un input que nos permite detener la prueba con un número de ticks determinado

- **STOP-AT**

Además, se han agregado una serie de gráficas para visualizar distintas métricas de la simulación: