SIMULACIÓN DE SISTEMAS

Práctica 1:

Diferentes Modelos de Simulación

Curso 2020/2021

Capítulo 1

Mi Primer Modelo de Simulación de MonteCarlo

1.1. Planteamiento del Sistema a Estudiar

Un coche está circulando por una calle muy larga (por simplicidad, supongamos que es infinitamente larga), buscando un sitio para aparcar. Las plazas de aparcamiento están numeradas consecutivamente. El coche se encuentra a la altura de la plaza numerada 0, y su destino está a la altura de la plaza número x (por ejemplo, $x = 100$). Por supuesto, *se desea aparcar lo más cerca posible de ese destino*. El conductor alcanza a ver desde su posición actual si las *dos* siguientes plazas están o no libres (o sea, si se encuentra en la posición i, sabe si las plazas i, $i + 1$ e $i + 2$, están libres o no). También se tiene conocimiento general de que cada plaza de aparcamiento está ocupada el 90 % de las veces y libre el 10 % (independientemente unas plazas de las otras). El conductor decide seguir la siguiente estrategia: pasará sin detenerse desde la plaza 0 (su punto de partida) hasta la plaza c (con $c \leq x$), aunque haya plazas libres, y entonces intentará aparcar en la plaza libre que encuentre más cercana a su destino, dentro de su ángulo de visión.

Un conductor "conservador (o pesimista)" elegiría aparcar en el primer sitio que encuentre (o sea elegiría $c = 0$)[1], mientras que un conductor "arriesgado (u optimista)" llegará hasta su destino sin detenerse (elegiría $c = x$)[2].

Se desea tomar una decisión sobre el valor apropiado de c para ni quedarse muy corto ni pasarse, y para ello se quiere estimar cuál es el valor óptimo de c, *desde el punto de vista de minimizar la distancia esperada desde el lugar de aparcamiento hasta el objetivo*. Vamos a utilizar un modelo de MonteCarlo para resolver este problema.

1.2. Tareas a Realizar

El programa `aparcamiento.cpp`, que puede encontrarse en la plataforma de docencia de la asignatura, implementa tal modelo. Básicamente realiza una estimación de la distancia media desde la posición en la que se aparca hasta la posición de destino, para cada posible posición

[1] Con el riesgo de quedarse muy lejos de su destino, pero *más vale malo conocido...*

[2] Con el riesgo de pasarse mucho y terminar aparcando en el "quinto pino", *la avaricia rompe el saco.*

de partida (para cada valor de c, desde 0 hasta x), y determina cuál es la menor distancia y para qué valor de c se alcanza, es decir dónde es mejor empezar a intentar aparcar.

- Utilizad este programa, *varias veces*, para observar los resultados. ¿Qué conclusiones podeis obtener? Resulta interesante realizar representaciones gráficas, por ejemplo mostrando la relación entre los valores de c y los de distancia media a la que se aparca del objetivo.

Naturalmente, el valor óptimo para c depende de varios factores: la distancia entre la posición de inicio y la de destino (el valor x, en nuestro caso 100), la saturación de los aparcamientos (la probabilidad de que un aparcamiento este ocupado, en nuestro caso 0,9) y nuestra agudeza visual (en nuestro caso 2 posiciones más allá de donde nos encontramos). ¿Cómo afectan estos parámetros al comportamiento del sistema, y en consecuencia a nuestra decisión de pasar de largo hasta la posición c?

El mismo programa anteriormente comentado, si se ejecuta suministrándole esos parámetros, también calcula el valor óptimo de c. Realmente el programa sin parámetros equivale a la llamada `aparcamiento 100000 100 2 0.9`. El primer valor se refiere al número de repeticiones que se realizan del proceso para hacer la estimación de la distancia media, dejadlo de momento a ese valor; el segundo parámetro es la posición de destino x; el tercer parámetro es el número de posiciones siguientes que podemos ver desde la actual; el cuarto parámetro es la probabilidad de que un aparcamiento esté ocupado.

- Ahora se trata de ver cómo cambios en estos parámetros afectan al resultado. ¿Qué ocurre si manteneis todos los parámetros menos uno, y ése lo vais variando? Realizad diversas pruebas, tabulad los resultados y sacad conclusiones. El uso de gráficas también resulta aquí interesante. Además de aquellas que relacionan los valores de c con la distancia media, se pueden emplear también gráficas que muestren la relación entre distintas probabilidades de ocupación y el valor óptimo obtenido para c, o entre probabilidades de ocupación y la distancia mínima (en media) a la que se consigue aparcar. Y lo mismo cambiando las distintas probabilidades de ocupación por los distintos niveles de visión.

- Probad también a cambiar más de un parámetro a la vez a ver qué pasa. En este caso se pueden emplear gráficas tridimensionales para representar la variación en la distancia mínima obtenida o el mejor valor de c en función de la probabilidad de ocupación y el nivel de visión. Utilizad también valores extremos de los parámetros, para observar si los resultados son coherentes.

Capítulo 2

Mi primer Modelo de Simulación Discreto

2.1. Planteamiento del Sistema a Estudiar

En una parte remota de cierto país hay 5 radares que se emplean para detectar incursiones aéreas no autorizadas. Un componente particular de los radares está sujeto a frecuentes fallos y, para conservar las 5 unidades de radar operativas, se dispone de un cierto número de repuestos de ese componente. Cuando un componente de algún radar falla, se instala inmediatamente un componente de repuesto, si lo hay. En caso contrario el radar queda fuera de servicio. En todo caso el componente averiado se envia a reparar, y es devuelto (o reemplazado) transcurrido cierto tiempo. Cuando se recibe un componente reparado, si todos los radares están operativos se almacena como repuesto, mientras que si algún radar está fuera de servicio se utiliza inmediatamente para ponerlo en funcionamiento.

El tiempo de vida de cada componente (es decir el tiempo que un radar permanece operativo, desde que se pone en funcionamiento hasta que falla) es de aproximadamente 20 días (según una distribución de probabilidad exponencial con media 20), y cuando un componente se envía a reparar, tarda en estar de vuelta entre 15 y 30 días (según una distribución de probabilidad uniforme entre 15 y 30). El componente es muy caro, por lo que resulta deseable no tener más repuestos que los estrictamente necesarios, desde la perspectiva de mantener la integridad del espacio aéreo.

Vamos a estudiar este problema empleando un modelo de simulación dinámico y discreto. El objetivo es determinar el número mínimo de repuestos que es necesario mantener para "garantizar" un porcentaje de tiempo de desprotección parcial (esto ocurre cuando algún radar esté sin funcionar) inferior al 1 %.

2.2. Tareas a Realizar

El programa `radares.cpp`, disponible también en la plataforma de docencia de la asignatura, implementa un modelo de simulación para este sistema funcionado durante un año. Tiene la capacidad de estimar el número de veces que algún radar falla y no hay repuestos disponibles, así como el porcentaje de tiempo que el espacio aéreo está parcialmente desprotegido. El

programa tiene como parámetros de entrada el número de componentes de repuesto de que se dispone y el número de simulaciones (años) que se van a realizar.

- Investigad experimentalmente, utilizando este programa, cuál es el número de repuestos mínimo que habría que tener. Realizad el estudio varias veces, fijando el parámetro relativo a número de simulaciones a diferentes valores (por ejemplo 1, 5, 10, 50, 100, 500 y 1000). ¿Qué conclusiones se obtienen?

También se puede llamar al programa `radares` pasándole como parámetros el número de radares que hay, el número de repuestos almacenados, los tiempos mínimo y máximo de devolución de componentes reparados, el tiempo medio de vida de cada componente, el tiempo total a simular y el número de simulaciones.

- Investigad experimentalmente la influencia en el rendimiento del sistema de algunos de esos parámetros, por ejemplo la posibilidad de acortar los tiempos de devolución de componentes reparados, o emplear componentes más robustos, con un mayor tiempo de vida.

Capítulo 3

Mi primer Modelo de Simulación Continuo

3.1. Planteamiento del Sistema a Estudiar

Hay una especie de pez pequeño que vive en un lago. Si notamos por x el número de esos peces, entonces su tasa de incremento, en ausencia de factores limitativos, se puede describir aproximadamente mediante la ecuación

$$\frac{dx}{dt} = ax$$

donde a es una constante relacionada con su tasa natural de natalidad. Esta ecuación indica que x se incrementará exponencialmente, sin ningún límite. En cualquier sistema natural habrá una cantidad limitada de recursos, tales como espacio o comida, para mantener a cualquier especie. Si suponemos que el máximo número de individuos de esta especie que podría existir en el lago es igual a la constante b, entonces una ecuación de crecimiento más razonable es:

$$\frac{dx}{dt} = ax\left(1 - \frac{x}{b}\right)$$

donde el término $(1 - x/b)$ refleja la competición entre miembros de la especie por los recursos. Para distintos valores de a y b, la ecuación anterior resulta en patrones de crecimiento más o menos rápido.

Supongamos ahora que en el mismo lago también vive otra especie de pez más grande, cuya comida principal es la especie de pez pequeño. El número de peces grandes, y, tendría una ecuación de crecimiento similar a la anterior, pero el máximo número de individuos de esta especie está limitado por la disponibilidad de comida, que está relacionada directamente con x. Por tanto una ecuación de crecimiento para la especie de pez grande es

$$\frac{dy}{dt} = cy\left(\frac{x}{d} - \frac{y}{e}\right) \tag{3.1}$$

Por supuesto, si los peces pequeños son comidos por los grandes, la tasa de crecimiento de x debería modificarse para reflejar este hecho. Por tanto obtenemos

$$\frac{dx}{dt} = ax\left(1 - \frac{x}{b}\right) - fy \tag{3.2}$$

El sistema de dos ecuaciones anterior describe las interacciones primarias entre las dos especies de peces. Son ecuaciones no lineales y no pueden resolverse analíticamente.

Para ilustrar las características de esas ecuaciones, supongamos que los coeficientes tienen los siguientes valores:

- $a = \frac{2}{30} = 0{,}0667$ días^{-1}
- $b = 10000000$
- $c = \frac{2}{90} = 0{,}0222$ días^{-1}
- $d = 10000000$
- $e = 36000$
- $f = 10$ días^{-1}

Estos valores implican que el número de peces pequeños x crece a una tasa tres veces mayor que el número de peces grandes, que el máximo número de peces de cada tipo que las condiciones del lago permiten es de 10000000 y 36000, y que la "voracidad" de los peces grandes es alta (cada pez grande devora unos 10 peces pequeños cada día). Se tiene interés en estudiar si ese sistema es estable, en ausencia de factores externos (si alcanza un "equilibrio ecológico").

Supongamos ahora que la especie de pez depredador tiene interés comercial, y que, en algún instante de tiempo, por ejemplo el 50 % de esos peces, o un determinado número de ellos, son pescados. Se desea conocer cómo afectará esta intervención al posible equilibrio del sistema, si se podrá recuperar o no, y cuándo.

Por último, se podría plantear el diseño de una política de pesca óptima, en el sentido de decidir cada cuánto tiempo hay que pescar y cuántos peces deben pescarse cada vez (asumiendo que en este proceso no se capturan peces pequeños, se usan redes de tamaño apropiado), de forma que no se agoten los recursos y se maximice el número de capturas.

Vamos a estudiar este sistema empleando un modelo de simulación dinámico y continuo.

3.2. Tareas a Realizar

El programa `Simulacion_lago2especiespeces.C`, disponible en la plataforma de docencia de la asignatura, implementa un modelo de simulación para este sistema. El programa devuelve el número de peces de cada especie en diferentes instantes de tiempo. Al ejecutarlo solicita como parámetros de entrada el tiempo total durante el que se estudiará el sistema (en días) y el número inicial de individuos de cada especie.

- Investigad, empleando un tiempo de simulación de varios años, y diferentes valores para el número de individuos de cada especie, si el sistema es capaz de alcanzar un punto estable de equilibrio. Resulta útil representar gráficamente la evolución del número de peces de cada especie en función del tiempo[1].

[1]Por ejemplo se puede usar gnuplot para esto de forma muy sencilla: ejecutar gnuplot, y luego lanzar el comando `plot ''fichero-de-datos'' using 1:2` (o `1:3`).

- Supuesto que el sistema está en equilibrio, investigad cómo afecta una campaña de pesca que capture determinado número de peces grandes, para diferentes valores posibles.
- Investigad diferentes políticas de pesca (cada cuánto tiempo hay que pescar y cuántos peces), con el objetivo de maximizar el número de capturas. Esto requiere algunas modificaciones en el código del programa, si se quiere hacer de forma automatizada.