Simulación - Simulación basa en Eventos Discretos

Colectivo de Simulación

Facultad de Matemática y Computación
Universidad de La Habana

October 18, 2021

Base General

La simulación de un modelo con elementos probabilísticos requiere generar mecanismos estocásticos del modelo y observar su comportamiento en el tiempo

Idea

Cuando ocurra un evento entonces se genera el tiempo de ocurrencia del próximo evento. Entonces, si hay un único tipo de evento se trata el inmediato que se generó, si existe más de un tipo de evento se trata aquel cuyo tiempo sea el menor.

Es como si todos los eventos se ordenaran en una línea de tiempo y uno la fuera recorriendo y tratando cada evento según su orden.

Elementos de los Modelos basados en Eventos Discretos

- Los eventos
- Las variables

Eventos

Se establecen a partir de la casuística propia de cada problema. Cuando ocurren se actualizan las distintas variables y se determina próximos eventos a partir de este.

Variables

Tiempo: tiempo de simulación transcurrido (t)

Salida: estado del sistema hasta el tiempo t

Contadoras: cantidad de veces que ha ocurrido un evento hasta el tiempo t

Descripción

Los clientes arriban a un servicio acorde a una determinada distribución de probabilidad. El servicio es provisto por un único servidor. Cuando un cliente llega, si el servidor está vació lo atiende sino se pone en cola para ser atendido. El tiempo de atención sigue un determinada distribución. Cuando se termina de atender a un cliente se procede a atender al cliente que lleva más tiempo esperando. El servicio ocurre en un tiempo T y cuando concluye no llegan más cliente pero se atiende a aquellos en espera.

Ejemplo

Una tienda con una única caja

Variables

- Variable de tiempo
  - t tiempo general
  - t_A tiempo de arribo
  - t_D tiempo de salida
- Variables contadoras
  - N_A cantidad de arribos
  - N_D cantidad de partidas
  - A diccionario arribo:tiempo de ese arribo
  - D diccionario partida:tiempo de esa partida
- Variables de estado
  - n número de clientes en el sistema

Iniciar el proceso

Hay que comenzar la simulación, por ello se generan las cantidades iniciales en 0 y también se genera el tiempo del primer evento a tratar en el sistema: arribo del primer cliente

Inicialización

- $t = N_A = N_D = n = 0$
- $A = \{\}$, $D = \{\}$
- Generar t_0 y $t_A = t_0$, $t_D = \infty$

Atender los arribos

Es necesario atender los arribos de clientes. Aquí se atiende un arribo y se genera el tiempo para la próxima llegada. Si no hay clientes en el sistema en el arribo hay que generar también el tiempo de la primera salida.

Eventos de Arribo: $t_A \leq t_D \wedge t_A \leq T$

1. $t = t_A$ (*se mueve la línea de tiempo*)
2. $N_A = N_A + 1$ (*arriba nuevo cliente*)
3. $n = n + 1$ (*un cliente más en el sistema*)
4. Generar t_{At}, $t_A = t + t_{At}$ (tiempo de próximo arribo)
5. Si $n == 1$ generar t_{Dt} y $t_D = t + t_{Dt}$ (*tiempo de próxima partida*)
6. $A[N_A] = t$ (*contabilizar el tiempo de este arribo*)

Atender las salidas

Los clientes llegan al sistema pero también deben salir y, por tanto, hay que generar los tiempos de salidas. Una vez se atiende una salida, se genera el tiempo de la próxima. Se debe tener en cuenta que si hay un solo cliente no se genere un tiempo de salida para un cliente que no existe en ese momento.

Eventos de Salida: $t_D < t_A \wedge t_D \leq T$

1. $t = t_D$ (*se mueve la línea de tiempo*)
2. $N_D = N_D + 1$ (*parte otro cliente*)
3. $n = n - 1$ (*un cliente menos en el sistema*)
4. Si $n == 0$ entonces $t_D = \infty$ sino generar t_{Dt} y $t_D = t + t_{Dt}$ (*tiempo de próxima partida*)
5. $D[N_D] = t$ (*contabilizar el tiempo de esta partida*)

Cierre del sistema

Cuando se llega al tiempo T de cierre no se generan más arribos y se da salida a los clientes dentro del sistema generando los subsecuentes tiempos de salida.

Evento de arribo fuera de tiempo: $min(t_A, t_D) == t_A \wedge t_A > T$

1. $t_A = \infty$ (*arribo fuera de tiempo y no se generan más*)

Eventos de cierre: $min(t_A, t_D) == t_A \wedge t_A > T \wedge n > 0$

1. $t = t_D$ (*se mueve la línea de tiempo*)
2. $N_D = N_D + 1$ (*parte otro cliente*)
3. $n = n - 1$ (*un cliente menos en el sistema*)
4. Si $n > 0$ generar t_{Dt} y $t_D = t + t_{Dt}$ (*próxima partida*)
5. $D[N_D] = t$ (*contabilizar el tiempo de esta partida*)

Descripción

Es similar al anterior, pero el servicio cuenta de dos procesos (atendidos por un servidor cada uno). Cuando un cliente concluye en el primer servidor pasa a tratar de ser atendido en el segundo servidor. Si este está vacío se atiende sino se pone en cola para ser atendido.

Ejemplo

Una tienda con una caja donde se cobra y, posteriormente, otra persona se dedica a verificar que lo que se saca de la tienda es lo que aparece en el vale de venta.

Variables

- Variable de tiempo
  - t tiempo general
  - t_A tiempo de arribo
  - t_{D1} tiempo de salida del servidor 1 (S1)
  - t_{D2} tiempo de salida del servidor 2 (S2)
- Variables contadoras
  - N_A cantidad de arribos
  - N_D cantidad de partidas
  - A_1 diccionario arribo:tiempo de ese arribo a S1
  - A_2 diccionario arribo:tiempo de ese arribo a S2
  - D diccionario partida:tiempo de esa partida
- Variables de estado
  - n_1 número de clientes en S1
  - n_2 número de clientes en S2

Inicialización

- $t = N_A = N_D = n_1 = n_2 = 0$
- $A_1 = \{\}$, $A_2 = \{\}$, $D = \{\}$
- Generar t_0 y $t_A = t_0$, $t_{D1} = t_{D2} = \infty$

Eventos de arribo: $min(t_A, t_{D1}, t_{D2}) == t_A \wedge t_A \leq T$

1. $t = t_A$
2. $N_A = N_A + 1$
3. $n_1 = n_1 + 1$
4. Generar t_{At}, $t_A = t + t_{At}$
5. Si $n == 1$ generar t_{D1t} y $t_{D1} = t + t_{D1t}$
6. $A1[N_A] = t$

Eventos de salida de S1: $min(t_A, t_{D1}, t_{D2}) == t_{D1} \wedge t_{D1} \leq T$

1. $t = t_{D1}$
2. $n_1 = n_1 - 1$
3. $n_2 = n_2 + 1$
4. Si $n_1 == 0$ entonces $t_{D1} = \infty$ sino generar t_{D1t} y $t_{D1} = t + t_{D1t}$
5. Si $n_2 == 1$ entonces generar t_{D2t} y $t_{D2} = t + t_{D2t}$
6.
7. $A2[N_A - n1] = t$

Eventos de salida de S2: $min(t_A, t_{D1}, t_{D2}) == t_{D2} \wedge t_{D2} \leq T$

1. $t = t_{D2}$
2. $N_D = N_D + 1$
3. $n_2 = n_2 + 1$
4. Si $n_2 == 0$ entonces $t_{D2} = \infty$ sino generar t_{D2t} y $t_{D2} = t + t_{D2t}$
5. $D(N_D) = t$

Eventos de Cierre

- Evento de arribo fuera de tiempo
- Evento de cierre de S1
- Evento de cierre de S2

Estos eventos se realizan de manera análoga a como se ejemplificó en el caso para un servidor teniendo en cuenta la nueva dinámica de este caso (nuevos eventos de partida)

¿Es posible una generalización para *n* servidores en serie?

Desccripción

Análogo al ejemplo anterior solo que ahora los servidores están en paralelo. Cuando un cliente llega si hay un servidor vacío se atiende en este, si los dos están llenos se pone en cola a esperar que uno se vacíe.

Ejemplo

Una tienda con dos cajas

Variables

- Variable de tiempo
  - t tiempo general
  - t_A tiempo de arribo
  - t_{D1} tiempo de salida del servidor 1 (S1)
  - t_{D2} tiempo de salida del servidor 2 (S2)
- Variables contadoras
  - N_A cantidad de arribos
  - N_{D1} cantidad de partidas desde S1, D_1 diccionario
  - N_{D2} cantidad de partidas desde S2, D_2 diccionario
- Variables de estado
  - $SS = (n, i_1, i_2, i3, .., i_n)$ n número de clientes en el sistema, i_1 cliente en S1 (0 si no hay), i_2 cliente en S2, $i3, .., i_n$ la cola

Inicialización

- $t = N_A = N_{D1} = N_{D1} = 0$
- $A = \{\}$, $D_1 = \{\}$, $D_2 = \{\}$
- Generar t_0 y $t_A = t_0$, $t_{D1} = t_{D2} = \infty$

Eventos de arribo: $min(t_A, t_{D1}, t_{D2}) == t_A \wedge t_A \leq T$

1. $t = t_A$
2. $N_A = N_A + 1$
3. Generar t_{At}, $t_A = t + t_{At}$
4. $A[N_A] = t$
5. Si $SS == \varnothing$ entonces $SS = (1, N_A, 0)$ y generar t_{D1t}, $t_{D1} = t + t_{D1t}$
6. Si $SS == (1, i_1, 0)$ entonces $SS = (2, j, N_A)$ y generar t_{D2t}, $t_{D2} = t + t_{D2t}$
7. Si $SS == (1, 0, i_2)$ entonces $SS = (2, N_A, j)$ y generar t_{D1t}, $t_{D1} = t + t_{D1t}$
8. Si $n > 1$ entonces $SS = (n + 1, i_1, i_2, ..., N_A)$

Eventos de partida de S1: $min(t_A, t_{D1}, t_{D2}) == t_{D1} \wedge t_{D1} \leq T$

1. $t = t_{D1}$
2. $N_{D1} = N_{D1} + 1$
3. $D_1[i_1] = t$
4. Si $n == 1$ entonces $SS = \varnothing$ y $t_{D1} = \infty$
5. Si $n == 2$ entonces $SS = (1, 0, i_2)$ y $t_{D1} = \infty$
6. Si $n > 2$ entonces $SS = (n-1, i_3, i_2, ...)$ y generar t_{D1t}, $t_{D1} = t + t_{D1t}$

Eventos de partida de S2: $min(t_A, t_{D1}, t_{D2}) == t_{D2} \wedge t_{D2} \leq T$

Análogo al caso de S1

Eventos de Cierre

- Evento de arribo fuera de tiempo
- Evento de cierre de S1
- Evento de cierre de S2

Estos eventos se realizan de manera análoga a como se ejemplificó en el caso para un servidor teniendo en cuenta la nueva dinámica de este caso (nuevos eventos de partida)

¿Es posible una generalización para n servidores en paralelo? ¿Es posible generalizar para n capas en serie con m servidores en paralero? ¿Se puede hacer valores arbitrarios de n y m?

Descripción

Un almacén guarda un tipo de producto y lo vende a un precio fijo (r) por unidad. Los clientes llegan al almacén siguiendo un distribución determinada y demandan una cantidad por cliente que sigue otra distribución particular. Para mantener el inventario del almacén se tiene un umbral máximo (S) y un umbral mínimo (s) de modo que si el nivel de inventario es inferior a s se ordenan $S - x$ unidades para restablecer el inventario (x cantidad en inventario). El costo de ordenar y unidades es $c(y)$ y demora L unidades de tiempo. Hay un costo de h por producto por unidad de tiempo. Se quiere conocer la ganancia en un tiempo T y lo dejado de ganar por solicitudes no satisfechas.

Variables

- Variable de tiempo
  - t tiempo general
  - t_A tiempo de arribo de clientes
  - t_M tiempo de de arribo de reposición
- Variables contadoras
  - C costo por reposición de mercancía
  - H costo por almacenaje
  - R ingresos por solicitudes satisfechas
  - P pérdidas por solicitudes no satisfechas
- Variables de estado
  - x cantidad de mercancía en inventario
  - y cantidad de mercancía a reponer

Inicialización

- $t = C = H = R = P = y = 0$
- x cantida inicial en almacén
- Generar t_0 y $t_A = t_0$, $t_M = \infty$

Eventos de solicitud: $t_A \leq t_M \wedge t_A \leq T$

1. $H = H + (t_A - t) * x * h$ *(actualiza costo de almacenaje)*
2. $t = t_A$ *(actualiza línea de tiempo)*
3. Generar X, $w = min(x, X)$ *(se genera solicitud factible)*
4. $R = R + w * r$ *(actualiza ingresos)*
5. $P = P + (X - w) * r$ *(actualiza posibles pérdidas)*
6. $x = x - w$ *(actualiza inventario)*
7. Si $x < s \wedge y > 0$ entonces $y = S - x$ y $t_M = t + L$ *(se determina si debe haber reposición)*

Eventos de reposición: $t_M < t_A \wedge t_M \leq T$

1. $H = H + (t_M - t) * x * h$ (*actualiza costo de almacenaje*)
2. $t = t_M$ (*actualiza línea de tiempo*)
3. $C = C + C(y)$ (*actualiza gasto por reposición*)
4. $x = x + y$ (*actualiza inventario*)
5. $y = 0,\ t_M = \infty$ (*no hay reposición en curso*)

¿Qué posibles generalizaciones pueden realizarse a este modelo? ¿Cómo pueden pensarse modelos híbridos teniendo en cuenta este y otros modelos?