

Simulación II

Dpto. Ingeniería Industrial, Universidad de Chile

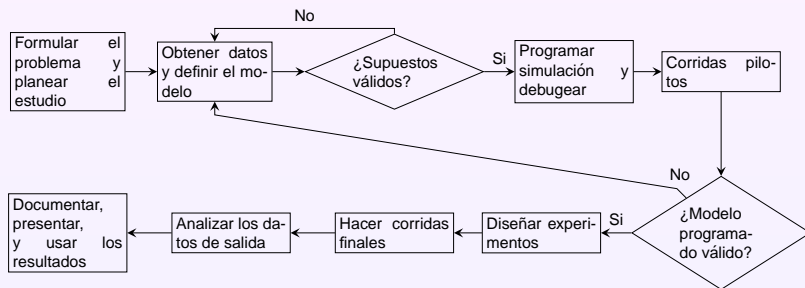
IN47B, Ingeniería de Operaciones

11 de marzo de 2008

Contenidos

1 Construyendo una Simulación

Vista General



Formular el problema y planear estudio

- Problema presentado por una unidad.
 - Problema puede estar mal definido, o en términos cualitativos.
 - Proceso Iterativo es necesario
- Varias reuniones con director proyecto, experto de simulación, experto en el sistema.
 - Definir objetivos del estudio.
 - Definir preguntas específicas a responder.
 - Definir medidas numéricas para comparar diferentes configuraciones.
 - Alcance del modelo.
 - Diferentes configuraciones a probar.
 - Tiempo del estudio y recursos necesarios.
- Seleccionar el software a usar.

Recoger Datos del Sistema

- Recoger información sobre estructura y reglas de operación del sistema.
 - Ningún individuo es suficiente.
 - Personas con conocimiento equivocado del sistema (buscar expertos reales).
 - Procedimientos pueden no estar formalizados.
- Recoger datos para definir parámetros del sistema y distribuciones de entrada.
- Formalizar información anterior en un documento de supuestos.
- Recoger datos sobre el desempeño del sistema real.

Recoger Datos del Sistema

- Escoger nivel de detalle del sistema.
 - Objetivos del proyecto.
 - Métricas a utilizar.
 - Datos disponibles.
 - Problemas de credibilidad.
 - Opinión de los expertos del sistema.
 - Restricciones de tiempo/presupuesto.
- Empezar con un modelo simplificado y refinarlo a medida que sea necesario.
- Interactuar con el director del proyecto y expertos del sistema regularmente.

¿Son los Supuestos Válidos?

- Realizar una revision del documento de supuestos con los expertos del sistema y con el director del proyecto.
 - Asegurar que supuestos son correctos y completos.
 - Ayudar a la interacción entre miembros del equipo.
 - Incrementar el sentido de propiedad del modelo en el equipo.
 - Debería hacerse antes de comenzar a programar.

Construir un programa y verificarlo

- Programar en un lenguaje general (C,C++,Java,C#) o en un software de simulación (Arena, Extend, Flexsim, ProModel).
 - Lenguajes generales tienen la ventaja de que usualmente uno es conocido.
 - Ofrecen control absoluto del programa.
 - Son mucho mas baratos de comprar (Licencias).
 - Pueden resultar en tiempos de ejecucion menores.
 - Lenguajes de simulacion resultan en menores tiempos de programación
 - Proveen de interfaces gráficas (Atractivas para gerencia).
- Debugear el programa.

Corridas Piloto, ¿Es el modelo programado Válido?

- Ejecutar corridas pilotos del programa
- Comparar desempeño real del sistema con sistema simulado.
- Revisar consistencia de resultados con expertos del sistema real y con director proyecto.
- Realizar análisis de sensibilidad, identificando aspectos del sistema que necesitan mayor nivel de detalle o cuidado en el modelo.

Diseño de Experimentos

- Para cada configuración de interés especificar:
 - Largo de cada corrida de simulación.
 - Período transiente de cada corrida (si necesario).
 - Número de simulaciones independientes a realizar, para así definir los correspondientes intervalos de confianza.

Corridas Finales, Análisis de Resultados

- Ejecutar corridas principales.
- Objetivos principales en el análisis son:
 - Determinar desempeño absoluto de cada configuración analizada.
 - Comparar configuraciones alternativas en forma comparativa (análisis de tipo pareto).

Documentar, presentar, y usar resultados

- Documentar supuestos utilizados.
- Documentar código del programa.
- Documentar creiterios de intervalos de confianza, etc.
- Presentar Resultados:
 - Uso de animaciones para presentar modelo a audiencia amplia.
 - Discutir proceso de validación de sistema.
 - Resultados se usarán en la medida de que sean validos y creibles.

Supuestos Generales

- Asumimos que existe un generador de números aleatorios uniforme.
- Nótese que estos no son continuos, tienen entre 32 o 56 bits de resolución.
- Diferencias menores a $10^{9,6}$ o $10^{16,8}$ no pueden observarse.
- Asumimos que aleatoriedad del generador es buena.
- Asumimos que el generador es *eficiente*.

Transformación Inversa

- Dado $U \sim U(0, 1)$, $X \sim F$ probabilidad acumulada, retornar $X = F^{-1}(U)$.
- Ejemplo: $X \sim \exp(\beta)$ entonces
$$F(x) = \begin{cases} 1 - e^{-x/\beta} & x \geq 0 \\ 0 & \text{si no} \end{cases}$$
- $F^{-1}(u) = -\beta \log(1 - u)$.
- Requiere generar un solo número aleatorio.
- Extendible a F no continuas, con saltos numerables.
- Más generalmente $X = \min\{x : F(x) \geq U\}$.
- Si F^{-1} no se conoce, método numérico es necesario.
- Fácil producir distribuciones truncadas.

Compocición

- Aplicable a compocición convexa de variables aleatorias.
- $F(x) = \sum_{i \in \mathbb{N}} p_i F_i(x)$, $p_i \geq 0$, $\sum_{i \in \mathbb{N}} p_i = 1$.
- Generar número j tal que $P(j) = p_j$.
- Retornar X con distribución F_j .
- Ejemplo: $X \sim \text{Trap}(a)$, donde
$$f(x) = \begin{cases} a + 2(1-a)x & 0 \leq x \leq 1 \\ 0 & \text{si no} \end{cases}.$$
- $f(x) = aI_{[0,1]}(x) + (1-a)2xI_{[0,1]}(x)$.
- Requiere sólo un números aleatoreo y generar sub-variable aleatorea.

Convolución

- Aplicable a variables aleatorias que son sumas de otras variables aleatorias independientes.
- $X = \sum_{i=1}^n Y_i.$
- Generar Y_i con la distribución apropiada.
- Retornar $X = \sum_{i=1}^n Y_i.$
- Ejemplo: $X \sim m\text{-Erlang}(\beta)$
- $X = \sum_{i=1}^m \exp(\beta/m).$

Aceptar/Rechazar

- Tenemos X con densidad $f(x)$ a soporte acotado S .
- Definimos $c = \max\{f(x) : x \in S\}$.
- Generar x uniformemente en S .
- generar Y uniformemente en $[0, c]$
- Retornar x si $Y \leq f(x)$, si no, generar nuevamente x, Y .
- Útil cuando otros métodos son difícil de implementar.
- Dependiendo de f puede requerir muchas generaciones de números aleatorios.