



Simulación de Sistemas

Trabajo Práctico Especial

Estrategia comercial de una software factory

Grupo 3

Civile, Juan Pablo	50453
Crespo, Alvaro	50758
Marseillan, Agustín	50134

17 de Junio de 2013

1. Introducción

El presente Trabajo Práctico busca representar la estrategia comercial de una consultora de software (*software factory*). En cuanto a la selección de proyectos, se busca la mejor combinación entre aprovechar el trabajo de los programadores y mantener cierta reserva de capacidad para no tener que rechazar proyectos *atractivos*, es decir, de mayor rentabilidad.

La simulación se basa en datos “pseudo-reales”, dada la dificultad que implica la recolección de datos y la escasez de ellos.

2. Modelo Conceptual

2.1. Modelado del problema

2.1.1. Variables de control

Se definen como variables de control del problema, las siguientes:

- Cantidad de programadores que se “reservan” (para proyectos *atractivos*).
- Estrategia de decisión de aceptación de proyectos.

2.2. Variables aleatorias

Se encontraron las siguientes variables aleatorias:

- Cantidad de proyectos que llegan en un determinado período.
- Tipo del proyecto (pequeño, mediano o grande).
- Tamaño del proyecto (medido en horas-hombre).
- Precio por hora del proyecto.
- Fecha de entrega del proyecto.

Los proyectos son modelados como una tupla de horas-hombre, precio por hora y fecha de entrega. La función de decisión sobre proyectos es la encargada de decidir si un proyecto es elegible o no.

2.3. Plan de cuadros

TODO

2.4. Funciones objetivo

Las funciones objetivo que se consideran son:

- Costo de oportunidad, ingreso que hubieran generado los proyectos rechazados.
- Ingreso generado por los proyectos aceptados.

- Porcentaje de recursos utilizados.

Para representar los resultados y la evolución de estas magnitudes, se tiene una vista que muestra, paso por paso, estos valores y, al final de la simulación, un gráfico con los intervalos de valores.

Cabe destacar que para esto, previamente se debe fijar, para las estrategias de decisión que los tengan, los parámetros de entrada, buscando que sean los que arrojen los mejores resultados.

2.4.1. Restricciones

Para el modelado del problema, se ignoraron varias variables reales del problema, que hubieran hecho impracticable la representación y resolución del mismo. El simulador se ve restringido a un uso académico, dado que no tiene en cuenta muchas variables del mundo real.

Las variables que se ignoran son:

- La capacitación del personal y su curva de aprendizaje: se supone que el programador conoce el proyecto y no tarda en empezar a programar.
- Nuevos requerimientos en el proyecto: Desde un primer momento, se conoce el tiempo de desarrollo que involucrará un proyecto, no se pueden agregar funcionalidades a la mitad del desarrollo.
- Competencia, mercado, precios: estas variables están sujetas a la llegada de proyectos, pero no se toman como variables separadas sino que se engloban en la llegada de proyectos.
- La situación financiera de la empresa: se considera que la empresa tiene fondos como para pagar sus costos de funcionamiento durante la duración de la simulación.
- El traspaso de desarrolladores: se supone que la cantidad de desarrolladores permanece constante a lo largo de la simulación.

2.5. Diagrama de Bloques

TODO

2.6. Diagrama de Flujo

Muestras artificiales (MA) Variables

1. Cantidad de Proyectos por mes(CA)
VA Poisson (λ)
 2. Tipo de Proyecto (TP)
VA Uniforme
 3. Tamaño de Proyecto [hs]
VA Triangular(a, b, c)
 4. Costo Hora-Hombre [\$]
VA Triangular(a, b, c)
 5. Plazo de Entrega del Proyecto [hs]
VA Triangular(a, b, c)
- T: Tiempo del reloj de la simulación
 - Tmax: Cantidad de tiempo a simular
 - P: Proyecto actual
 - R: Recursos (programadores)
 - M: Dinero
 - CO: Costo de Oportunidad
 - M(P): Ganancia del proyecto P
 - R(P): Función de asignación de recursos usados por el proyecto P
 - T(P): Tiempo que abarca el proyecto P
 - D: Función o Estrategia de decisión, toma un proyecto y devuelve si se lo acepta o no
 - O(P₁, ..., P_n, D): Ordena los proyectos P₁ a P_n según el criterio de la estrategia D

2.7. Estrategias de decisión

Tanto en el diagrama de bloques como en el diagrama de flujo se observa una función o estrategia de decisión para aceptar o no un proyecto. A continuación se detallan las estrategias que se estudiarán. Debe ser tomado en cuenta que independientemente de la estrategia, si no se tienen suficientes recursos como para completar el proyecto antes de su fecha de entrega el valor de $D(P)$, es decir la función que acepta o no el proyecto, es siempre *NO*.

1. Se ordenan los proyectos arribados según el cociente Ganancia / Tamaño, y se aceptan los proyectos que tengan los mayores valores de tal cociente, o que obtengan valores mayores a un cierto umbral. Variables: μ : valor mínimo de aceptación para el ratio Ganancia / Tamaño de un proyecto
2. Se ordena los proyectos encolados según su tamaño, y se aceptan los de menor tamaño o aquellos que sean menor a un cierto tamaño. Variables: τ : valor máximo de tamaño de un proyecto para su aceptación

2.8. Estrategia de asignación de recursos

Para la asignación de recursos, al comienzo de cada período, se calcula la cantidad de horas/hombre que se van a asignar al proyecto de la siguiente manera: se divide el total de horas/hombre restantes del proyecto, llamémosle W, y se lo divide por la cantidad de períodos que faltan hasta la fecha entrega. En caso de queden recursos sin asignar, se asignarán al proyecto más cercano a su fecha de entrega. De esta forma, se busca balancear la asignación de recursos, buscando cumplir con las fechas de entrega, y en caso de ser posible, adelantar los proyectos cercanos a su fin, con el objetivo de poder aceptar otros nuevos.

3. Modelo de Datos

3.1. Variables Aleatoria

- A) Cantidad de proyectos por mes
- B) Tipo de proyecto
- C) Tamaño de proyecto [h]
- D) Precio hora-hombre del proyecto [\$]
- E) Plazo de entrega del proyecto [h]

3.2. Estrategia de obtención

TODO

Cuadro de valores para las distribuciones según tipo de proyecto

Tipo	Porcentaje
Grande	15 %
Mediano	75 %
Pequeño	10 %

	Tipo	Mín.	Máx.	Moda
Horas	Grande	4500	8000	5300
	Mediano	2000	4500	3200
	Pequeño	500	2000	3200
Precio	Grande	180	310	230
	Mediano	200	240	220
	Pequeño	200	300	290
Fecha de Entrega	Grande	Horas/6	Horas/4	Horas/5
	Mediano	Horas/4	Horas/2	Horas/3
	Pequeño	Horas/3	Horas/1	Horas/2

4. Modelo Operacional

4.1. Arquitectura

Para el desarrollo del simulador se utilizará el lenguaje *Python*. Se hará uso de la librería *numpy* para la generación de números aleatorios, *matplotlib* para la generación de gráficos y *wxPython* para la GUI.

4.2. Etapas y Plan de Desarrollo

Etapa 1 - Modelo de Organización

Como primer paso del desarrollo, se hará el modelo de organización y proyectos. Esto es, codificar las reglas de decisión y asignación de recursos. Desde luego que para

poder desarrollar estas reglas, se debe codificar los tipos de datos proyecto y organización.

Etapas 2 - Motor de Simulación

El principal objetivo de esta etapa es poder simular el análisis mensual de la organización. Con la llegada de proyectos, la decisión de toma de proyectos y la asignación de los recursos para el mes de trabajo.

Etapas 3 - Modelo de Datos

El objetivo de esta etapa es ajustar de forma correcta el valor de todas las variables y parámetros especificados en el modelo de datos. Se deberían implementar los generadores de cada variable y deberían ser verificados.

Etapas 4 - Interfaz visual

El objetivo de esta etapa es que pueda visualizarse la corrida de la simulación, mostrando las diferentes etapas, procesos y/o resultados intermedios. También se deberían poder visualizar las métricas en especial aquellas que son funciones objetivo.

4.3. Plan de experimentación

Para probar el simulador se diseñarán diferentes estrategias de aceptación y asignación de recursos, y se variará la cantidad de desarrolladores. Se probarán incluso escenarios de irreales, como la estrategia trivial de aceptar todo proyecto en según su orden de llegada, sólo con el motivo de probar el simulador y verificar que no surgen problemas.

4.4. Plan de pruebas

4.4.1. Verificación

- **Verificación de generadores aleatorios:** En el simulador se colocaron los histogramas de los valores que se generan durante la simulación. El objetivo de esto es verificar que en la simulación los histogramas se correspondan con las distribuciones de cada variable.
- **Verificación de estrategias de aceptación:** Una de las cosas importantes a verificar es que el simulador elija los caminos acorde a la estrategia seleccionada. Para ello se diseñaron dos pequeños conjuntos de testeo, compuestos por proyectos, generados correctamente según las distribuciones que corresponden, para los cuales, después de un corto tiempo de simulación se conocen los resultados.

5. Conclusiones

A. Anexo