

SIM 1.0, BIBLIOTECA DE ALGORITMOS EN R PARA EL APOYO A LA ASIGNATURA DE SIMULACIÓN.

SIM 1.0, BIBLIOTECA DE ALGORITMOS EN R.

AUTORES: Denis Luis Espinosa Pérez¹

Eliecer Sánchez Ramírez²

Julio Jesús Garcia Coste³

DIRECCIÓN PARA CORRESPONDENCIA: dlespinosa@uci.cu

Fecha de recepción:

Fecha de aceptación:

Resumen

En la Universidad de las Ciencias Informáticas se incluye en el 4to año de la carrera la asignatura de Simulación. La misma centra su objetivo en aplicar la simulación en el planteamiento y solución de problemas de dirección para el apoyo a la toma de decisiones. La práctica en la enseñanza de la asignatura se apoya en materiales que realizan los profesores. Sin embargo no se cuenta con un compendio de los algoritmos relevantes y usados en la asignatura según su programa analítico. La presente investigación describe una solución a la problemática planteada a partir de la implementación de una biblioteca de carácter didáctica, utilizando el software R, que contenga la totalidad de los algoritmos utilizados en la asignatura, así como una salida que describa los aspectos de interés de cada uno de ellos. Sim pone a disposición algoritmos para la generación de números aleatorios, de observaciones aleatorias con distribución no uniforme, métodos estadísticos para la validación de aleatoriedad y uniformidad, así como técnicas de reducción de varianza en un conjunto de números aleatorios generados.

Palabras clave: Simulación, enseñanza, algoritmo.

Denis Luis Espinosa Pérez: Ing. en Ciencias Informáticas. Universidad de las Ciencias Informáticas. La Habana. Cuba. dlespinosa@uci.cu .

Eliecer Sánchez Ramírez: Ing. en Ciencias Informáticas. Universidad de las Ciencias Informáticas. La Habana. Cuba. eliecersr@uci.cu .

Julio Jesús Garcia Coste: Ing. En Ciencias Informáticas. Instructor. Universidad de las Ciencias Informáticas. La Habana. Cuba. jjcoste@uci.cu .

SIM 1.0, ALGORITHMS LIBRARY IN R TO SUPPORT THE SUBJECT OF SIMULATION.

In the Informatics Sciences University, in the 4th year of the career the subject of Simulation is included. It focuses in applying the simulation in the approach and the direction problems solution to support the decision takings. Practice in the teaching of the subject is supported through materials made by professors. However, there is not a compendium of the outstanding and used algorithms in the subject according to its analytical program. The current research describes a solution to the problematic mentioned before, starting from the implementation of a library with a didactic approach, which contain all the algorithms used in the subject, as well as a gateway that describes the interesting aspects of each one of them. SIM makes available different algorithms for the random numbers, random observations with non-uniform distribution, statistic methods to validate the randomness and uniformity, as well as different variance reduction techniques generation in a set of generated random numbers.

Keywords: simulation, learning, algorithm

Introducción

La simulación es una técnica numérica para conducir experimentos en una computadora digital. Estos experimentos comprenden ciertos tipos de relaciones matemáticas y lógicas, las cuales son necesarias para describir el comportamiento y la estructura de sistemas complejos del mundo real a través de largos periodos de tiempo. A través de un estudio de simulación, se puede estudiar el efecto de cambios internos y externos del sistema, al hacer alteraciones en el modelo del sistema y observando los efectos de esas alteraciones en el comportamiento del sistema.

Esta técnica ha sido incluida como asignatura en el 4to año de la carrera de ingeniería en ciencias informáticas, impartida en la Universidad de las Ciencias Informáticas, en el curso 2014-2015. La misma plantea sistema de habilidades a vencer:

1. Identificar los elementos que caracterizan a los sistemas discretos: sistemas de servicio y otros.
2. Identificar los elementos que componen un modelo de simulación discreta.
3. Aplicar métodos para generar números y variables aleatorias.

4. Aplicar la simulación en el planteamiento y solución de problemas discretos de dirección o toma de decisiones, aplicando el software existente cuando ello sea posible.
5. Analizar e interpretar económicamente, con ayuda de métodos estadísticos adecuados, la solución obtenida y seleccionar la mejor alternativa o proponer mejoras de solución.
6. Identificar y solucionar por los métodos analíticos adecuados tales problemas, aplicando el software existente cuando ello sea posible (Vazquez, 2014).

Para la correcta asimilación de los objetivos 3,4 y 6, se hace necesaria la interacción con el lenguaje R, siendo este último el recomendado en el programa analítico de la asignatura.

La asignatura está dividida en 2 temas. El tema 1: Introducción a la simulación. Generación de variables aleatorias, abarca una serie de contenidos a fines, entre los cuales se encuentran:

- Generación de números aleatorios.
- Generación de observaciones aleatorias.
- Validación de uniformidad en un conjunto de números aleatorios.
- Validación de independencia en un conjunto de números.

Para ambos contenidos existen métodos que son de vital importancia para el campo que estudian, como son los métodos congruenciales lineales y Tausworthe para la generación de números aleatorios, y los métodos: Transformada inversa de la función, Box y Muller, Teorema del límite central, Composición, Convolución y Aceptación rechazo para la generación de observaciones aleatorias. Como mismo lo son los métodos para validación de uniformidad en un conjunto de números aleatorios: Prueba de Kolmogorov-Smirnov⁴, en lo adelante KS, y la prueba de Chi-cuadrado. En el contenido de validación de independencia en un conjunto de números, son expuestos los métodos de prueba de "gaps" o huecos y prueba de autocorrelación.

El tema 2: Simulación de eventos discretos, se enfoca en la aplicación de los métodos estudiados en el tema 1 para llevar a cabo la simulación manual con fines didácticos y la simulación computarizada usando el lenguaje R con muestras relevantes para llegar a conclusiones, poder extraer conocimiento del proceso y ayudar a la toma de decisiones.

⁴**Kolmogorov-Smirnov:** Método que se basa en la diferencia absoluta máxima D entre los valores de la distribución acumulada de una muestra aleatoria de tamaño n y una distribución teórica determinada para la validación de uniformidad en un conjunto de números aleatorios.

Desarrollo

La biblioteca Sim en su versión 1.0 se implementa con carácter didáctico. Cuenta con 14 algoritmos que abarcan la totalidad de los métodos estudiados en la asignatura. Cada uno de ellos cuenta con dos tipos de salidas o retorno. El primero, es el definido para cada algoritmo y el segundo, es una salida de tipo texto que describe como sucedió la corrida del algoritmo, definiéndose paso a paso la forma en que fueron cambiando los parámetros y las variables involucradas, así como el estado en que se encuentra la ejecución en cada preciso momento. Esto último facilita el uso de la biblioteca y pone a disposición del profesor encargado de impartir la asignatura una valiosa herramienta, en la cual se puede apoyar para llevar a cabo el proceso de enseñanza-aprendizaje en el aula y constituye el valor agregado más importante que brinda esta herramienta.

Software R3.1.1 y RStudio 0.98

Para la implementación de la biblioteca se utilizó el lenguaje R. Es un lenguaje para análisis estadístico y gráfico. Se trata de un proyecto de software libre, resultado de la implementación GNU del lenguaje S. R es un lenguaje utilizado principalmente en la investigación por la comunidad estadística, biomédica, la bioinformática y las matemáticas financieras. A esto contribuye la posibilidad de cargar diferentes bibliotecas o paquetes con finalidades específicas de cálculo o gráfico. Otra de las características de R es su capacidad gráfica, que permite generar gráficos con alta calidad para llevar un análisis exhaustivo de los datos. Actualmente se encuentra en la versión 3.2.0.

Como IDE se hizo uso del software RStudio en su versión 0.98. RStudio es una aplicación completa que cuenta con las herramientas de desarrollo, optimizado para el lenguaje R. El programa te permite crear secuencias de comandos, compilar códigos, crear parcelas, así como el trabajo con varios conjuntos de datos, en un entorno altamente capaz. La aplicación le permite crear proyectos para el almacenamiento de datos. RStudio cuenta con una serie de herramientas de gran alcance diseñado para facilitar el desarrollo de aplicaciones en lenguaje R, como el software de análisis de datos. El lenguaje R se dedica a crear las dos líneas de código y generar representaciones gráficas. El programa le permite optimizar su trabajo mediante el almacenamiento y organización de datos en los proyectos. Por lo tanto, cada proyecto puede contener varios archivos, como scripts R, Rebajas R, texto plano, código fuente en C ++, R, HTML, Presentaciones R o Documentación R.

RStudio es capaz de integrar las herramientas que utiliza con R en un único entorno. El programa permite el trabajo con HTML, PDF, documentos y presentaciones de diapositivas. Por otra parte, puede fácilmente abrir proyectos existentes o interrumpir una sesión de R, con el fin de cambiar el directorio de trabajo, para cargar, guardar o borrar el espacio de trabajo actual.

Complejidad algorítmica.

Generalmente en los procesos simulados se utilizan gran cantidad de datos por lo que en la implementación de algoritmos para este fin, es necesario tener en cuenta criterios para medir su rendimiento o comportamiento. Estos criterios se centran principalmente en su simplicidad y en el uso eficiente de los recursos. Respecto al uso eficiente de los recursos, éste suele medirse en función de dos parámetros: el espacio, es decir, memoria que utiliza, y el tiempo, lo que tarda en ejecutarse. Ambos representan los costes que supone encontrar la solución al problema planteado mediante un algoritmo. Dichos parámetros van a servir además para comparar algoritmos entre sí, permitiendo determinar el más adecuado de entre varios que solucionan un mismo problema.

La unidad de tiempo a la que se debe hacer referencia para el cálculo de la complejidad temporal no puede ser expresada en segundos o en otra unidad de tiempo concreta, pues no existe un ordenador estándar al que puedan hacer referencia todas las medidas. Se denota por $T(n)$ el tiempo de ejecución de un algoritmo para una entrada de tamaño n (Leiserson, Cormen, Rivest, & Stein, 2009). Teóricamente $T(n)$ debe indicar el número de instrucciones ejecutadas por un ordenador idealizado. Se debe buscar por tanto medidas simples y abstractas, independientes del ordenador a utilizar. Para ello es necesario acotar de alguna forma la diferencia que se puede producir entre distintas implementaciones de un mismo algoritmo, ya sea del mismo código ejecutado por dos máquinas de distinta velocidad, como de dos códigos que implementen el mismo método. Para la descripción de la función que enuncia la complejidad temporal se hace uso en este caso del criterio de la cota superior. Luego de llevado a cabo en análisis de complejidad de los algoritmos implementados se construye la siguiente tabla con un resumen de los resultados (Leiserson, Cormen, Rivest, & Stein, 2009).

<i>Método</i>	<i>Nombre de la función</i>	<i>Parámetros</i>	<i>Descripción</i>	<i>CO</i>
Box y Muller	<i>boxmuller(r1,r2,m,d)</i>	<i>r1</i> : Número aleatorio. <i>r2</i> : Número aleatorio. <i>m</i> : media. <i>d</i> : Desviación típica.	Algoritmo para la generación de observaciones aleatorias que siguen una distribución normal.	$O(1)$
Congruencial lineal	<i>congruencial(a,m,x0,k,c)</i>	<i>a</i> : Multiplicador. <i>m</i> : Módulo de la división. <i>x0</i> : Semilla. <i>k</i> : Cantidad de números a generar. <i>c</i> : Incremento.	Algoritmo para la generación de números aleatorios.	$O(n)$
Tausworthe	<i>gtausworthe(r,q,iniciales,t,bit)</i>	<i>r</i> : Entero que cumple $0 < r < q$. <i>q</i> : Entero que cumple $0 < r < q$. <i>iniciales</i> : Arreglo ⁵ de valores binarios. <i>t</i> : Entero positivo. <i>bit</i> : Cantidad de bits.	Algoritmo para la generación de números aleatorios.	$O(n)$

⁵Arreglo: Conjunto de valores.

		generar.		
Trasformada inversa de la distribución exponencial	<i>invExponencial(numeroa, m)</i>	<i>numeroa</i> : Número aleatorio. <i>m</i> : media.	Algoritmo para la generación de observaciones aleatorias.	$O(n)$
Trasformada inversa de la distribución uniforme.	<i>invuniforme(numeroa, a, b)</i>	<i>numeroa</i> : Número aleatorio. <i>a</i> : Primer extremo del intervalo. <i>b</i> : Segundo extremo del intervalo.	Algoritmo para la generación de observaciones aleatorias.	$O(n)$
Teorema del límite central	<i>tlcentral(numerosa, m, d)</i>	<i>numerosa</i> : Arreglo de números aleatorios. <i>m</i> : Media. <i>d</i> : Desviación típica.	Algoritmo para la generación de observaciones aleatorias.	$O(n)$
Teorema del límite central	<i>tlcentralv(numerosa, m, d, cn)</i>	<i>numerosa</i> : Arreglo de números aleatorios. <i>m</i> : Media. <i>d</i> : Desviación típica. <i>cn</i> : arreglo de números enteros positivos.	Algoritmo para la generación de observaciones aleatorias.	$O(n^2)$

Prueba de Kolmogorov-Smirnov	<i>testks(numerosa, cantidad)</i>	<i>numerosa</i> : Arreglo de números aleatorios. <i>cantidad</i> : Entero positivo.	Algoritmo para la validación de uniformidad de un conjunto de números aleatorios.	$O(n^2)$
Prueba de "gaps"	<i>testgaps(numerosa)</i>	<i>numerosa</i> : Arreglo de números aleatorios.	Algoritmo para la validación de independencia de un conjunto de números aleatorios.	$O(n)$
Prueba de Chi-cuadrado	<i>testchic(numerosa)</i>	<i>numerosa</i> : Arreglo de números aleatorios.	Algoritmo para la validación de independencia de un conjunto de números aleatorios.	$O(n)$
Prueba de autocorrelación	<i>testautocorrelacion(numerosa)</i>	<i>numerosa</i> : Arreglo de números aleatorios.	Algoritmo para la validación de independencia de un conjunto de números aleatorios.	$O(n)$
Composición	<i>composición(pi, numsa, fnes)</i>	<i>pi</i> : Arreglo de coeficientes. <i>numsa</i> : Arreglo de números aleatorios.	Algoritmo para la generación de observaciones aleatorias.	$O(n^2)$

		<i>fnes</i> : Arreglo de funciones.		
Convolución	<i>convolución(numsa, fnes)</i>	<i>numsa</i> : Arreglo de números aleatorios. <i>fnes</i> : arreglo de funciones	Algoritmo para la generación de observaciones aleatorias.	$O(n^2)$
Aceptación-Rechazo	<i>aceptacionr(numerosa, cond)</i>	<i>numerosa</i> : Arreglo de números aleatorios. <i>cond</i> : Condición.	Algoritmo para la generación de observaciones aleatorias.	$O(n)$

Tabla 1 Análisis de la complejidad temporal de cada uno de los algoritmos de la biblioteca SIM.

Una vez descrito, se puede concluir que la complejidad de los algoritmos en general se encuentra entre las cotas $O(1)$ y $O(n^2)$. Nótese que no representan valores que pongan en cuestión su ejecución en sistemas de cómputos de bajas prestaciones.

Resultados

Carácter didáctico e instructivo

Para abordar los aspectos de carácter didáctico que propone SIM 1.0, se debe incursionar en los conceptos de didáctica. Se entiende por este término que es la rama de la Pedagogía que se encarga de buscar métodos y técnicas para mejorar la enseñanza, definiendo las pautas para conseguir que los conocimientos lleguen de una forma más eficaz a los educados (Definición.de, 2015).

El material didáctico es aquel que reúne medios y recursos que facilitan la enseñanza y el aprendizaje. Suelen utilizarse dentro del ambiente educativo para facilitar la adquisición de conceptos, habilidades, actitudes y destrezas. Es importante tener en cuenta que el material didáctico debe contar con los

elementos que posibiliten un cierto aprendizaje específico (Definición.de, 2015).

Sim 1.0 más allá de ser una biblioteca de algoritmo como otras que existen en el mundo, aporta al usuario una salida tipo texto en la cual queda la explicación del algoritmo recién ejecutado. Esto constituye una técnica para la comprensión del algoritmo por parte de los educandos, lo cual ayuda a que los conocimientos lleguen de una manera más eficaz, ya que pueden realizar comprobaciones y comparaciones de lo que están desarrollando de forma manual. Estas actividades de comprobación y comparación crean en el educando las habilidades necesarias para la correcta interpretación y adquisición del conocimiento. A continuación se muestra una salida en consola de uno de los algoritmos.

```
> congruencial()
Se inicia el generador congruencial lineal mixto con parámetros a = 3 m = 11 x0 = 1 y c = 5
Iteracion 1 se genera el valor 0.727272727272727
Iteracion 2 se genera el valor 0.636363636363636
Iteracion 3 se genera el valor 0.363636363636364
Iteracion 4 se genera el valor 0.545454545454545
[1] 0.09090909 0.72727273 0.63636364 0.36363636 0.54545455
>
```

Ilustración 1 Ejecución del algoritmo congruencial lineal mixto

Actualmente la biblioteca Sim 1.0 recién comenzó a ser utilizada como medio de enseñanza para la asignatura de Simulación, teniendo en cuenta de que esta no cuenta con un banco de medios, debido a que se imparte por primera vez en la carrera de Ingeniería en Ciencias Informáticas. En sus inicios ha tenido buena aceptación por el claustro que imparte la asignatura y por el estudiantado. La complejidad temporal de los algoritmos que propone es baja, por lo cual asegura su eficiente ejecución. Las bondades que brinda como material didáctico inciden positivamente en proceso enseñanza-aprendizaje en la asignatura.

Conclusiones

La biblioteca Sim 1.0 constituye un importante material de apoyo para los profesores que imparten la asignatura de simulación en la carrera de ingeniería en ciencias informáticas en la UCI. Contribuye a la adquisición del sistema de habilidades con que deben contar los estudiantes de la mencionada carrera. La correcta implementación de los algoritmos reduce la complejidad temporal de los mismos, contribuyendo al ahorro de recursos de hardware. El uso del R pone a disposición de desarrollador una amplia gama

de funcionalidades con las que se reduce considerablemente el tiempo de implementación.

Bibliografía

Azor Montoya, J. (2000). *Pruebas no paramétricas*. Mendoza.

Barceló, J. (1996). *Simulación de Sistemas Discretos*. Madrid: Isdefe.

Becerra Córdova, G. (2004). *UN SISTEMA GENERADOR DE NÚMEROS PSEUDO ALEATORIOS*. México.

Carvajal Carreño, W., Ordóñez Plata, G., Moreno Wuandurruga, A., & Duarte Gualdrón, C. (2011). Simulación de sistemas eléctricos con cargas no lineales y variantes en el tiempo. *Ingeniare. Revista chilena de ingeniería*, 76-92.

Definición.de. (2015). *Definición.de*. Obtenido de <http://definicion.de/didactica/>

Felicísimo, A. (2007). Conceptos básicos, modelos y simulación.

Harel, D. (2004). *Algorithmics. The Spirit of Computing*. London: Pearson Education.

Leiserson, C., Cormen, T., Rivest, R., & Stein, C. (2009). *Introduction to algorithms*. London.

Montes, F., & Corberán, A. (2001). *¡Qué difícil es la aleatoriedad!*

Moreno Parra, R. (2012). *Simulación: Conceptos y programación*. Colombia.

Parberry, I., & Gasarsh, W. (2002). *Problems on Algorithms*. Texas: Prentice-Hall, Inc.

Pérez Maqueo, O., Delfin, C., & Esquihua, M. (2006). Modelos de simulación para la elaboración y evaluación de los programas de los servicios ambientales hídricos.

Salas Perea, R., & Ardanza Zulueta, P. (1995). La simulación como método de enseñanza y aprendizaje. *Educación Médica Superior*.

The Art of Computer Programming. (2007). London.

Urquía Moraleda, A. (2010). *Simulación. Texto base de teoría*. Madrid.

Vazquez, E. (2014). *Programa analítico de la asignatura de Simulación*. La Habana.