КАЧЕСТВО ГЕНЕРАЦИИ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ В СИСТЕМАХ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ OPENGPSS, GPSS\WORLD И ANYLOGIC

Д. Г. Диденко (Киев, Украина)

Введение

Одним из современных методов анализа работы сложных систем является имитационное моделирование. Для проведения повторяющихся компьютерных прогонов стохастических моделей используются события, полученные от генераторов псевдослучайных чисел (ГПЧ). Поэтому качество псевдослучайных последовательностей имеет большое значение для получения достоверных результатов. В докладе рассматриваются современные системы имитационного моделирования OpenGPSS (http://www.simulation.kiev.ua) [1, 2], GPSS World [3] и AnyLogic [4], которые работают с дискретными моделями. Системы моделирования поддерживают работу с большим количеством вероятностных распределений – от Бернулли до Вейбула: 29 распределений в OpenGPSS, в помощи AnyLogic описано 29 распределений (хотя в [4] идет речь о 37), 24 распределения в GPSS World [5, 6, 7]. Все виды распределений строятся на равномерном распределении, качество которого и оценивается далее.

Тесты псевдослучайных последовательностей

На сегодня существует много графических и статистических тестов для проверки псевдослучайных последовательностей. Среди статистических тестов широко используются следующие: NIST, TEST-U01, CRYPT-X, The pLab Project, Diehard, ENT подобное. В настоящем докладе рассматривается применение набора тестов Diehard (автор George Marsaglia) для трёх систем имитационного моделирования.

Для каждой среды моделирования была написана собственная небольшая программа на соответствующем языке (GPSS или Java), которая позволяет получить большую выборку псевдослучайных чисел с равномерным распределением. Далее с помощью пакета программ Diehard, бинарные файлы которого взяты с сайта http://stat.fsu.edu/pub/diehard/, выполнен анализ последовательностей чисел для каждой системы моделирования отдельно.

Описание лабораторного оборудования

Для всех практических экспериментов использовался один компьютер класса Intel Core Duo с процессором 2,1 ГГц и оперативной памятью 2 Гб, операционная система Windows XP SP3. Эксперименты ставились на системах моделирования следующих версий: OpenGPSS 1.2.1.0 (1.2.2.0), GPSS World 5.2.2. и AnyLogic 6.4.1. Все эксперименты проводились над генератором псевдослучайных чисел № 1 со следующими начальными смещениями: 300 (OpenGPSS), 200 (GPSS World) и 100 (AnyLogic).

Особенности проведения вычислительного эксперимента в системах моделирования

Для системы моделирования GPSS World при малом количестве чисел (до 4 млн) программа Diehard выдает ошибку: "run-time error F6501: Read (имя файла) – end of file encountered". Это происходит на тесте № 2 «Перестановки, которые пересекаются», хотя в тексте самого теста написано, что он обрабатывает последовательность из 1 млн. 32-битных чисел дважды.

К сожалению, сразу сгенерировать данные в бинарный файл не позволяют среды моделирования, так как поддерживается чтение и запись только текстового файла. По-

лучить конечный файл можем лишь за два шага: первый – создание текстового файла с 4 млн. строк, в каждой строке одно число, которое записано текстом, и второй – перевод текстового файла в бинарный.

Обычная программа генерирования последовательности в GPSS World в текстовый файл сначала даже для 40 тыс. чисел длится 2 ч. И только после модификации программы на GPSS генерирование такого файла происходит за 2 мин. Дело в том, что блок открытия файла OPEN долго работает. Поэтому его не нужно каждый раз вызывать для каждого транзакта, а следует перенести в отдельный часовой сегмент GENERATE, OPEN, ADVANCE, CLOSE и TERMINATE, тем самым вызвав один раз для всей программы.

Для системы моделирования OpenGPSS текстовый файл получен за 30 мин.

Промежуточный текстовый файл занимает 46 Мбайт. Затем на конвертацию в нужный бинарный файл размером 15 Мбайт с помощью VisualBasic-сценария тратится 3 мин.

Каждый тест на выходе формирует специальное число p-value из интервала [0, 1]. Результаты проведенных экспериментов приведены в табл. 1. Количество сгенерированных чисел для каждой системы моделирования – 4 млн. В таблице «+» означает, что тест пройден (если p-value принадлежит отрезку [0,025; 0,975]), а «–» – тест не пройден.

Таблица 1

Значение p-value тестов DIEHARD

№	Тест	OpenGPSS		GPSS World		AnyLogic	
1	Дни рождения (Birthday Spacings)	0,359018	+	1,000000	–	0,431397	+
2	Пересекающиеся перестановки (Overlapping Permutations)	1,000000	–	0,060425	+	0,008494	–
3	Ранги матриц (Ranks of matrices)	0,950574/ 0,991835/ 0,999961	+/–/–	0,413445/ 0,536567/ 0,097624	+/+/+	0,382749/ 0,342521/ 0,249165	+/+/+
4	Поток битов (The bitstream test)	0,805974*	+	1,000000	–	0,4825392	+
5	Обезьяньи тесты (Monkey Tests)	1,000000	–	0,585412*	+	0,494274*	+
6	Подсчёт единичек (Count the 1's)	1,000000	–	1,000000/ 0,495902*	–/+	0,558042*/ 0,587793	+/+
7	Тест на парковку (Parking Lot Test)	0,979816	–	0,751581	+	0,221977	+
8	Тест на минимальное расстояние (Minimum Distance Test)	0,992231	–	0,545258	+	0,506258	+
9	Тест случайных сфер (Random Spheres Test)	0,173505	+	0,162508	+	0,278340	+
10	Тест сжатия (The Squeeze Test)	1,000000	–	0,475504	+	0,985307	-
11	Тест пересекающихся сумм (Overlapping Sums Test)	0,919356	+	0,681207	+	0,899422	+
12	Тест последовательностей (Runs Test)	0,312858	+	0,458301*	+	0,549636*	+
13	Тест игры в кости (The Craps Test) (for no. of wins/ for throws/game)	0,406554/ 1,000000	+/–	0,214467/ 0,807587	+/+	0,343878/ 0,396355	+/+

Были полностью использованы все тринадцать тестов этого пакета. Но, конечно, прохождения (или непрохождения) тестов недостаточно, чтобы принять или отклонить гипотезу о случайности потока данных. Тесты Diehard формируют на выходе числа

* Для этого теста считается среднее арифметическое, так как выводилось много значений p-value в результирующем файле.

p-value, которые равномерно распределены в интервале [0, 1], если входной поток чисел действительно случайный. Проверяем нашу «нулевую» гипотезу о входной поток через статистическую значимость по критерию Пирсона (критерий «Хи-квадрат»). Для критерия Пирсона нужно много реализаций, а тестов всего 13, поэтому используем все значения p-value из результирующего файла, там их около 240 (!). Результаты расчетов по критерию показаны в табл. 2.

Таблица 2

Проверка статистической гипотезы о случайности потока данных

Генератор псевдослучайных чисел	Количество пройденных тестов из набора Diehard	Критерий Пирсона («Хи-квадрат»)		Анализ результата
		полученное	табличное	
OpenGPSS	5	90,00	36,2	Поток нельзя считать случайным
GPSS World	10	29,45	36,2	Поток можно считать случайным
AnyLogic	11	28,17	36,2	Поток можно считать случайным

Улучшение ГПЧ в системе моделирования OpenGPSS

Первым вариантом улучшения ГПЧ является использование встроенного ГПЧ из СУБД Oracle. Работа с системным пакетом dbsm_random состоит из задания начального смещения для ГПЧ и получения следующего числа. При этом пакет уже встроен в Oracle и широко используется, что является большим преимуществом. К недостатком здесь следует отнести невозможность получить текущее смещение.

К другим способам улучшения ГПЧ относится самостоятельная реализация по заданным формализациям:

- линейный конгруэнтный метод $X_{n+1} = (aX_n + c) \bmod m$;
- квадратичный конгруэнтный метод $X_{n+1} = (dX_n^2+aX_n+c) \bmod m$;
- генератор на основе объединения путём сложения по mod 2^{32} двух генераторов: запаздывающего генератора Фибоначчи $X_n = X_{n-99}\ X_{n-33} \bmod 2^{32}$ и генератора на основе произведения с переносом $Y_n = 30903\ Y_n\text{-}1$ carry mod 2^{16};
- генератор М-последовательностей;
- вихрь Мерсена.

При этом возможны модификации линейного конгруэнтного метода:

- расширенный конгруэнтный генератор – $X_n = 2^{13} (X_{n-1} + X_{n-2} + X_{n-3}) \bmod (2^{32} – 5)$;
- алгоритм “Marsaglia-Multicarry” (Джордж Марсаглия);
- алгоритм “xor-shift” (Джордж Марсаглия);
- алгоритм Блюма-Блюма-Шуба;
- генератор на базе произведения с переносом – $X_n = (2111111111\ X_{n-4} + 1492\ X_{n-3} + 1778\ X_{n-2} + 5115\ X_{n-1})$ carry mod 2^{32};
- генератор на базе произведения с переносом – $X^n = a\ X_{n-1}$ carry mod 2^{32}.

Для улучшения качества сгенерированной последовательности решено использовать линейный конгруэнтный метод с различными значениями *a*, *m* и *c* для широкораспространённых библиотек и языков программирования, приведённых в табл. 3.

После замены алгоритма генерирования случайных чисел получаем следующие прогоны батареи тестов Diehard (табл. 4) в новой версии системы моделирования OpenGPSS 1.2.2.0.

Таблица 3

Примеры Линейного Конгруэнтного Метода

№	Источник	m	a	c	Биты
1	ANSI C: Open Watcom, Digital Mars, Metrowerks, IBM VisualAge C/C++	2^{32}	1103515245	12345	16..30
2	Apple CarbonLib	$2^{31}-1$	16807	0	0..31
3	Borland C/C++	2^{32}	22695477	1	0..30
4	Borland Delphi, Virtual Pascal	2^{32}	134775813	1	0..31
5	glibc (используется в GCC)	2^{32}	1103515245	12345	0..30
6	GNU Compiler Collection	2^{32}	69069	5	16..30
7	LC53 из языка программирования Forth	$2^{32}-5$	2^{32}- 333333333	0	0..31
8	Microsoft Visual Basic (версия 6 и ниже)	2^{24}	1140671485	12820163	0.23
9	Microsoft Visual/Quick C/C++	2^{32}	214013	2531011	16..30
10	MMIX Дональда Кнута	2^{64}	6364136223846793005	1442695040888963407	0..63
11	MS Fortran	$2^{31}-1$	48271	0	0..31
12	Numerical Recipes	2^{32}	1664525	1013904223	0..31
13	Random class in Java API	2^{48}	25214903917	11	16..47
14	RtlUniform from Native API	$2^{31}-1$	2147483629	2147483587	0..31
15	(старая версия библиотеки glibc)	2^{32}	69069	1	0..31

Таблица 4

Сведенные результаты прохождения тестов DIEHARD

ГПЧ	№ теста из батареи Diehard													Всего пройдено тестов
	1	2	3	4	5	6	7	8	9	10	11	12	13	
1. ANSI C	-	+	-/-/-	-	-	+	-	-	-	-	-	+	-/-	3
2. Apple CarbonLib	-	+	+/+/+	-	+	+	+	+	+	+	+	+	+/+	11
3. Borland C/C++	-	+	-/-/-	-	+	+	-	-	-	-	-	+	-/-	4
4. Borland Delphi, Virtual Pascal	-	+	+/+/-	-	+	+	-	-	-	-	-	+	-/-	4
5. glibc	-	+	-/-/-	-	+	+	-	-	-	-	-	+	-/-	4
6. GNU Compiler Collection	-	-	-/-/-	-	+	+	-	-	-	-	-	+	-/-	2
7. LC53	-	+	+/+/+	-	+	+	-	-	-	-	-	+	-/-	5
8. Microsoft Visual Basic	!*	-	-/-/-	-	-	+	-	-	-	-	-	+	-/-	2
9. Microsoft Visual/Quick C/C++	-	-	-/-/-	-	+	-	-	-	-	-	-	+	-/-	2
10. MMIX Дональда Кнута	+	+	-/-/+	-	+	+	-	-	-	-	-	+	-/-	6
11. MS Fortran	+	+	+/+/+	-	+	+	+	+	+	+	+	+	+/+	12
12. Numerical Recipes	-	+	+/+/-	-	+	+	-	-	-	-	-	+	-/-	4
13. Random class in Java API	-	-	-/-/-	-	+	-	-	-	-	-	-	+	-/-	2
14. RtlUniform from Native API	-	-	+/+/-	-	-	-	-	-	-	-	+	-	+/+	2
15. VAX's MTH\$RANDOM	-	+	+/+/-	-	+	+	-	-	-	-	-	+	-/-	2
16. Пакет dbsm_random из Oracle XE	-	+	+/+/+	+	+	+	+	+	+	+	+	+	+/+	12

Из таблицы видно, что перспективнее всего использовать алгоритмы "MS Fortran" и встроенный пакет dbsm_random. Для них и считается критерий Пирсона, как было уже описано ранее, а результаты заносятся в табл. 5.

Далее реализуются все 15 ГПЧ, при этом по умолчанию используется датчик "MS Fortran". Генератор из пакета dbsm_random решено не использовать из-за невозможности его работы с несколькими датчиками одновременно. Теперь настройку ГПЧ

* Данный тест не выполняется из-за программной ошибки.

в системе OpenGPSS пользователь может произвести на странице «Настройки» в браузере, выбрав из списка один из 15 генераторов, приведенных в табл. 3.

Таблица 5

Проверка статистической гипотезы о случайности потока данных

Генератор псевдослучайных чисел	Количество пройденных тестов из набора Diehard	Критерий Пирсона («Хи-квадрат»)		Анализ результата
		полученное	табличное	
OpenGPSS 1.2.1.0	5	90,00	36,2	Поток нельзя считать случайным
OpenGPSS 1.2.2.0 MS Fortran	12	24,46	36,2	Поток можно считать случайным
OpenGPSS 1.2.2.0 dbsm_random	12	12,06	36,2	Поток можно считать случайным

Заключение

1. Системы моделирования OpenGPSS, GPSS World и AnyLogic можно использовать для построения стохастических моделей исследуемых систем. В новой версии системы OpenGPSS работа ГПЧ улучшена, на что указывают значения критерия Пирсона.

2. Одним из вариантов улучшения равномерности сгенерированных псевдослучайных чисел является использование не одного алгоритма получения равномерного распределения, а нескольких, отличающихся по отношению качество/скорость. Перед компьютерным прогоном экспериментатор имеет возможность выбирать режим генерирования равномерного распределения в зависимости от своих потребностей и аппаратных возможностей. К сожалению, это реализовано только в системе моделирования OpenGPSS.

3. Для улучшения поведения стохастических моделей следует использовать несколько ГПЧ. Например, количество ГПЧ в GPSS World – 10^3, в OpenGPSS – 10^{38}, а в AnyLogic (с учетом датчиков пользователя) – не ограничено.

Литература

1. **Діденко Д. Г.** Порівняння генераторів псевдовипадкових чисел в системах імітаційного моделювання OpenGPSS, GPSS World та AnyLogic. //Шоста науково-практична конференція з міжнародною участю "Математичне та імітаційне моделювання систем. МОДС'2011". Чернігів, 2011. С. 315–318.
2. **Томашевский В. Н., Диденко Д. Г.** Агентная архитектура распределенной дискретно-событийной системы имитационного моделирования OpenGPSS // Системні дослідження та інформаційні технології. 2006. № 4, К.: ВПК «Політехніка», 2006. С.123–133.
3. **Бражник А. Н.** Имитационное моделирование: возможности GPSS WORLD. СПб.: Реноме, 2006. 439 с.
4. **Карпов Ю.** Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. СПб.: БХВ-Петербург, 2005. 400 с.
5. **Рыжиков Ю. И.** Имитационное моделирование. Теория и технологии. СПб.: КОРОНА-принт; М.: Альтекс-А, 2004. 384 с.
6. **Боев В. Д.** Моделирование систем. Инструментальные средства GPSS World: учеб. пособие. СПб.: БХВ-Петербург, 2004. 368 с.
7. **Алиев Т. И.** Основы моделирования дискретных систем. СПб: СПбГУ ИТМО, 2009. 363 с.