УДК 519.856.2

Д.А. Рыбаков, Х.Д. Ламажапов, А.В. Ивченко

ХЭШ-ФУНКЦИИ КАК ИСТОЧНИК СЛУЧАЙНЫХ ЧИСЕЛ В МОДЕЛЯХ СО СТОХАСТИКОЙ

Параметры многих моделируемых процессов имеют две составляющие: закономерность и случайность. Если закономерность часто можно выразить с помощью аналитических функций, то случайность можно выразить с помощью хэш-функций (hash functions). Использование хэш-функций позволяет компактно хранить в программе массивы псевдослучайных чисел, доступных только для чтения. Извлечение подмножеств из этого массива позволяет осуществлять дальнейшую полноценную обработку. Преимущества такого подхода продемонстрировано при моделировании трехмерной перколяционной системы на решетке. Извлеченный из такой решетки перколяционный кластер требует $O(L^D)$ байт памяти, где L – линейный размер, D≈2,5 – фрактальная размерность кластера.

Хэш-функции; стохастика; фракталы; перколяция.

D.A. Rybakov, Kh.D. Lamazhapov, A.V.Ivchenko

HASH FUNCTIONS AS SOURCE OF RANDOM NUMBERS FOR STOCHASTICAL MODELS

Many simulated processes include regularity and chaos. Regular values are usually expressed using analytic functions meanwhile chaotic values may be expressed using hash functions. Using of hash functions in software program leads to compact representations for read-only arrays of pseudo random values. Subsequent extraction of subset from read-only array allows full data processing for extracted data. For example one may express random lattice of percolation model using hash functions. Lattice maximum size is not limited by computer memory. Percolation cluster extracted from such lattice consumes $O(L^D)$ bytes, where L – is cluster linear size, D ≈ 2,5 – is fractal dimension of cluster.

Hash-functions; stochastic; fractal; percolation.

Хэш-функции в качестве входного параметра получают массив байтов. В результате работы алгоритма на выходе получается число, которое зависит только от входных параметров. Среди хэш-функций существуют такие, свойства значений которых очень близки к свойствам равномерных распределений случайных чисел. Примером такой функции служит функция Message Digest 5 (MD5), которая возвращает числа от 0 до $2^{128}-1$. Если в качестве параметра этой функции подавать случайный набор байтов, то возвращаемые значения будут иметь равномерное псевдослучайное распределение в указанном диапазоне. При этом возвращаемые значения будут некоррелированными, даже если входные массивы коррелированны. Например, если два входных массива отличаются только на один бит, то возвращаемые два числа будут независимыми.

Такая особенность делает возможным использовать хэш-функции в качестве генератора псевдослучайных чисел. Чтобы получать числа в диапазоне от 0 до 1, требуется нормализация. Нормализированная функция

$$f(s) = MD5(s) / 2^{128}, \quad (1)$$

где s – массив байтов, возвращает псевдослучайное число с плавающей точкой с равномерным распределением в диапазоне от 0 до 1.

Замечательным свойством этой функции является то, что если требуется получить и использовать большой массив псевдослучайных чисел $\{x_i\}$, то отпадает необходимость хранить этот массив где-либо. Достаточно каждый раз при необходимости производить вычисления $x_i = f(i)$.

Для того чтобы получать разные массивы псевдослучайных чисел, следует сделать входной параметр зависимым от заранее заданной константы *R:*

$$x_i = f(s(R,i)), \tag{2}$$

где функция $s(R,i)$ производит символьное слияние текстового представления своих аргументов с разделителем. Например, $s(1,2) = "1,2"$, $s(1,2,3,4,5) = "1,2,3,4,5"$. В таком случае вся последовательность $\{x_i\}$ зависит только от величины *R*. Таким образом, достигается компактное представление последовательности псевдослучайных чисел любой длины с помощью только одного числа *R*.

Продемонстрируем эту особенность при моделировании перколяционных задач [1]. Теория перколяции изучает возникновение связанных структур в случайных средах. Примером такой среды может служить трехмерная кубическая решетка, в которой некоторые кубы являются проводящими с вероятностью p. Если две и более примыкающих ячеек являются проводящими, то они образуют проводящий кластер. Крайними случаями такой среды являются модель диэлектрика (при $p=0$) и модель сплошного проводника (при $p=1$). В промежуточных случаях возникает сложная стохастическая среда с интересными характеристиками с точки зрения математики и физики. В более общих случаях математическая модель перколяции может описывать такие физические явления, как фазовые переходы второго рода, горение, протекание и многие другие явления, в которых присутствует два и более компонента разной природы.

При моделировании такой системы в простом случае используется массив $P_{x,y,z}$, заполненный единицами с вероятность p и нулями с вероятностью $1-p$, где x, y, z – целые числа. Массив $P_{x,y,z}$ соответствует решетке и обычно располагается в ОЗУ вычислительной машины, и занимаемый объем ОЗУ растет по кубическому закону от линейного размера массива. При вычислениях программа обращается к ячейкам памяти $P_{x,y,z}$, которое можно заменить вызовом функции $P(x, y, z)$,

где

$$P(x, y, z) = \begin{cases} 1, & \text{если } MD5(s(x, y, z, R)) \le p, \\ 0. \end{cases} \tag{3}$$

При фиксированных параметрах x, y, z функция $P(x, y, z)$ всегда будет возвращать фиксированное значение без обращения к ячейкам большого массива. Таким образом, формируется некоторая решетка, доступная только для чтения. При этом параметр *R* является идентификатором, который однозначно задает структуру этой решетки.

Есть случаи, когда этого бывает достаточно. В процессе работы можно выделять подмножества и записывать их в память компьютера. Эти подмножества будут требовать меньшее количество ОЗУ, чем вся решетка. В случае перколяционной задачи можно выделять кластеры, пути внутри кластеров и т.д. Кластером считает набор связанных ячеек такой, что между любыми двумя ячейками существует проводящий путь.

Перколяционные кластеры являются фрактальными объектами с фрактальной размерностью $D \approx 2,5$ [2, 3] (рис. 1). Из курса геометрии известно, что объем фрактального объекта степенным образом зависит от его линейного размера

$$V \sim L^D, \tag{4}$$

где V – объем, L – линейный размер объекта [4].

Таким образом, один кластер включает $N \sim L^{2.5}$ ячеек решетки. Например, весь массив 5 000×5 000×5 000 потребовал бы ~125 Гбайт, в то время как перколяционный кластер – порядка 1,7 Гбайт, а другие подмножества, например пути внутри кластера, – еще меньше. При этом сокращение использования ОЗУ компенсируется более длительным выполнением программы, так как обращение к функции $P(x, y, z)$ более длительное, чем обращение к ячейке памяти.

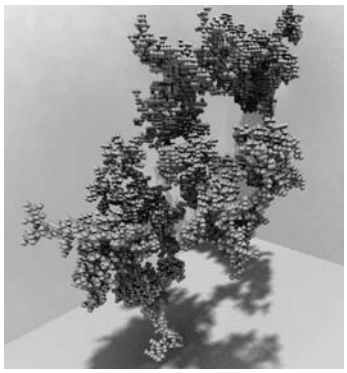

Рис. 1. Перколяционный кластер, найденный алгоритмом с использованием MD5

В завершение отметим, что кроме *MD5* существует множество других хэш функций. При использовании других функций для подобных вычислений следует убедиться, что возвращаемые значения являются некоррелированными и имеют равномерное распределение.

Выводы. При определенных условиях хэш-функция является аналогом массива псевдослучайных величин, доступного в программе только для чтения. При этом достигается компактное представление массива псевдослучайных величин любого размера. Различные приемы программирования могут значительно сократить использование ОЗУ и других хранилищ данных при моделировании стохастических процессов. При этом нехватка ОЗУ компенсируется бóльшим временем исполнения программы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Тарасевич Ю.Ю.* Перколяция: теория, приложения, алгоритмы. – М., 2002. – 122 с.
2. *Bagnich S.A., Konash A.V.* Computer investigation of the percolation processes in two- and three- dimensional systems with heterogeneous internal structure // Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. April 2003. – Vol. 502, № 2-3, 21. – P. 731-732.
3. *Ламажапов Х.Д., Прохоров С.А, Рыбаков Д.А.* Свойства трехмерных кластеров, составленных из параллелепипедов // Вестник Новосибирского государственного университета. Сер. Физика. – 2009. – Т. 4 , № 3. – С. 67-73.
4. *Мандельброт_Б.* Фрактальная геометрия природы. – М., 2002. – 656 с.

Статью рекомендовал к опубликованию д.т.н. Журавлева.

Рыбаков Дмитрий Александрович – Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования "Самарский государственный аэрокосмический университет им. акад. С.П. Королева (национальный исследовательский университет)"; e-mail: dim1r@yandex.ru; 445037, г. Тольятти, ул. Степана Разина, 32, кв. 84; тел.: +79023222099; кафедра радиоэлектроники и системотехники тольяттинского филиала СГАУ; к.т.н.

Ламажапов Хубита Доржиевич – Самарский государственный университет путей сообщения; e-mail: hubitalamazhapov@gmail.com; 443066, г. Самара, 1-й Безымянный пер., 18; тел.: 88469990656, +79272069528; кафедра физики и экологической теплофизики; к.ф.-м.н.; доцент.

Ивченко Алексей Владимирович – Институт акустики машин при Самарском государственном аэрокосмическом университете им. акад. С.П. Королева; 443086, г. Самара, Московское шоссе, 34, корп. 14; с.н.с.; к.т.н.

Rybakov Dmitry Alexandrobvich – Samara State Aerospace University named after academician S.P. Korolyov ; e-mail: dim1r@yandex.ru; 32-84, Stepana Razina street, Togliatti, 445037, Russia; phone: +79023222099; the department of electronics and system engineering of Togliatti branch of SSAY; cand. of eng. sc.

Lamazhapov Khubuta Dorzhievich – Samara State University of Transport; e-mail: hubitalamazhapov@gmail.com; 18, 1st Bezymjannyj pereulok, Samara, 443066, Russia; phone: +78469990656; the department of thermal and heat engines; cand. of phis.-math. sc.; associate professor.

Ivchenko Alexej Vladimirovich – Research Institute of Machine Acoustics at the S.P. Korolyov Samara State Aerospace University; e-mail: fgrt@yandex.ru ; 34, Moskovskoje Shosse, korpus 14; Samara, 443086, Russia; senior research; cand. of eng. sc.

УДК 621.396:517.9:518.6

В.Н. Бирюков

ОБУСЛОВЛЕННОСТЬ ПРАКТИЧЕСКИХ ЗАДАЧ ПАРАМЕТРИЧЕСКОЙ ОПТИМИЗАЦИИ

Ошибка параметрической оптимизации в условиях ограниченной точности исходных данных имеет случайный характер и существенно зависит как от жесткости задачи, так и от обусловленности по аргументу. Показано, что если первая составляющая ошибки может быть снижена существенно, то появляется возможность экспериментальной оценки второй составляющей ошибки. Обнаружено, что при малой точности исходных данных ошибка, вследствие плохой обусловленности, может стать доминирующей. Вероятность плохой обусловленности растет с увеличением размерности задачи, чем, в частности, и объясняется снижение эффективности методов численной оптимизации с ростом размерности.

Оптимизация; погрешность; обусловленность

V.N. Biryukov

CONDITIONALITY OF PARAMETRIC OPTIMIZATION

If the accuracy of a source data is limited, the error of parametric optimization is random. The error depends on the stiffness of a problem, and on the conditioning of the argument. This paper provided opportunity to experimental evaluation of the error. Component of the error associated with the high stiffness of the problem; in some cases it may be negligible. In these cases, the error component associated with poor conditioning becomes dominant. The article shows that for low accuracy of initial data ill-conditioning becomes a major factor of optimization error. The most important conclusion relates for a multi-dimensional problems, since the probability of ill-conditioning increases with increasing dimension.

Index Terms-optimization; error; conditionality