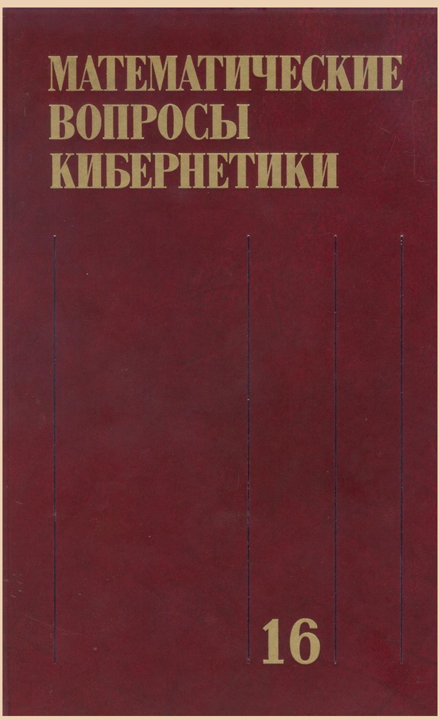

Р. Г. Бухараев

Сети вероятностных процессоров

СЕТИ ВЕРОЯТНОСТНЫХ ПРОЦЕССОРОВ

Р. Г. БУХАРАЕВ

(КАЗАНЬ)

Вероятностный автомат (ВА) — это автоматическое устройство, управляемое системой специальных команд и реализующее наперед заданное статистически стабильное случайное поведение выходных параметров. Вероятностные вычисления имеют ряд преимуществ перед детерминированными, в особенности в тех случаях, когда речь идет о вычислениях (моделировании) в сетях автоматов. В работе предлагается краткий обзор сравнительных характеристик при вычислениях на детерминированных и вероятностных автоматах.

Используя идею ВА, можно конструировать вероятностные машины, реализущие случайное поведение с заданными статистическими характеристиками с большой эффективностью. Еще большей эффективностью обладают вероятностные машины, реализованные конструктивно на сетях ВА. Существует большое разнообразие конструктивных средств и приемов, позволяющих многократно, в миллионы раз, увеличивать скорость и вычислительную мощь таких машин, применяя вероятностные вычислительные и моделирующие алгоритмы и обеспечивая высокую эффективность решения или моделирования.

Перечисляются возможные приемы и средства, которые могут быть использованы для ускорения процесса вычислений или моделирования. Эти свойства вероятностных процессоров (ВП) и сетей ВП определяются характеристиками математической модели — ВА. Описываются различные методы и подходы к их синтезу.

Введение

В применении компьютеров для решения вычислительных задач или моделирования сложных систем вероятностные алгоритмы занимают особое место. Первыми в этом ряду являются методы статистического моделирования, которые, возникнув практически одновременно с вычислительными машинами как так называемые методы Монте-Карло [11, 12], получили широкое распространение и развитие. В основе метода лежит моделирование некоторой случайной величины $x(T)$, подобранной таким образом, что ее математическое ожидание $E(x)$ есть решение задачи T, или просто с ним связано. Таким образом, аппроксимация решения задачи T получается как среднее значение большого числа выборочных значений случайной величины $x(T)$. Используя этот подход, можно вычислять кратные и функциональные интегралы, решать системы линейных алгебраических уровнений высоких порядков и, поэтоиу, реализовать бесчисленные вычислительные методы, сводящиеся к решению подобных систем, решать интегральные уравнения и краевые задачи для дифференциальных уравнений в частных производных различного типа и т. п.

Другой класс таких задач составляют многие проблемы физического моделирования, в которых случайность представляет собой поведенческую характеристику процесса, такие, как задачи массового обслуживания или задача моделирования работы ядерного реактора. Наконец, можно организовать вероятностный вычислительный процесс таким образом, что вероятность случайного кода сама рассматривается как количественная характеристика, или, иначе говоря, вероятность случайного кода рассматривается в процессе как аналоговый носитель числовой информации [10]. Для оценки сравнительной эффективности вероятностных алгоритмов можно привести ряд простых соображений, которые, тем не менее, проясняют проблему.

К очевидным преимуществам вероятностных (статистических) вычислений относятся:

— их малая связность, т. е. объем оперативной памяти, необходимой для промежуточных вычислений (это в существенной степени увеличивает возможности производить сложные вычисления);

— простота и высокая степень цикличности вычислительных (или моделирующих) алгоритмов, что существенно увеличивает возможности аппаратной реализации алгоритмов, чем в детерминированном случае (это дает дополнительный вклад в возможности усложнения рещаемых задач и в упрощение процессов программирования и отладки);

— высокая степень надежности вычислений (или моделирования), поскольку многие категории сбоев не оказывают заметного влияния на продуктивность алгоритма (такой новый тип надежности ведет к тому, что последняя очень мало зависит от сложности вычислительной схемы, обеспечивая в целом практически неограниченное усложнение схемы вычислений при сохранении функциональной надежности).

Таблица 1

Некоторые методы ускорения процессов моделирования

1. Аппаратурная генерация случайных кодов η	$n(\eta)$ раз
2. Параллельная реализация экспериментов	N_1 раз (степень параллелизма)
3. Конвейерное использование траекторий	T_η (средняя длина траекторий)
4. Использование ассоциативной памяти для проверки логических условий w	$n(w)$ раз
5. Использование адаптивного табличного процессирования для вычисления φ	$n(\varphi)$ раз
6. Гибридные методы вычислений благодаря стохастическому представлению чисел	$n(f)/N_\xi$ раз, (f вычисляется как $p(\xi)$)
7. Использование суперпозиций вероятностных алгоритмов	N_0 раз (число траекторий)
8. Использование принципа N-трансформация ВА	M раз (степень дублирования копий ВА).

Общий вывод состоит в том, что чем сложнее решаемая проблема, тем эффективнее в ее решении использование вероятностных методов. Это обстоятельство не случайно. Согласно гипотезе А. Н. Колмогорова, случайность можно рассматривать в некотором смысле как проявление сложности. С этой точки зрения, когда проблема не допускает аналитического решения (или его детерминированное описание «слишком сложно»), вероятностный результат можно рассматривать как частичную вероятностную «заплату», заменяющую недоступное аналитическое решение. В рассматриваемом аспекте это означает, что с возрастанием сложности возникающих проблем сфера применимости вероятностных алгоритмов будет расширяться, все более и более заменяя детерминированные методологии.

В табл. 1 приводится ряд возможных методов, которые могут использоваться для ускорения выполнения вычислительных и моделирующих алгоритмов с применением вероятностных процессоров.

Анализ показывает, что, в дополнение к простоте и надежности, вероятностные вычисления имеют и другие замечательные свойства. Они обладают рядом существенных преимуществ перед детерминированными алгоритмами, в особенности когда они используются в вычислительных структурах из сетей ВП. Табл. 2 дает краткий сравнительный анализ.

Таблица 2

Некоторые положительные свойства вероятностных вычислений

Случайность как конструктивный элемент машины	A. Резкое увеличение надежности B. Возможности строить более сложные модели
Вероятность как данное	A. Высокая точность B. Высокая дополнительная надежность C. Вычисление в реальное время D. Возможность универсализации схемы
Слабая взаимозависимость	A. Высокая надежность управления B. Широкие возможности параллельных вычислений
Новые средства вычислений и моделирования	A. Более тонкие и мощные статистические методы B. базирующиеся на нетривиальных свойствах ВА

Следет заметить, что вероятностные методы имеют и свои недостатки. Основных недостатков два. Слабая сходимость метода — первый из них. Число экспериментов N, требующееся для аппроксимации среднего значения $E(x)$ с точностью ε, обратно пропорционально только квадратному корню из ε, поэтому для получения достаточной точности число экспериментов должно быть очень большим.

Второй — это высокая «стоимость» машинного времени, расходуемого на один эксперимент. Сюда включаются время получения одного значения случайного кода $n(x)$ и, как правило, время $n(j, \omega)$ некоторого вычисления j и проверки некоторого логическиго условия ω. В виду специфики алгоритма M, общее время $T(M)$ получения одного среднего значения оказывается сравнимым с числом

$$T(M) = N(n(x) + n(j, \omega)),$$

что чрезвычайно чувствительно для общего времени получения решения задачи. Легко видеть, что оба отмеченных недостатка определяются не существом статистических методов, но неадекватностью структуры детерминированных компьютеров эффективной реализации вероятностных алгоритмов.

Если коротко сформулировать требования к эффективной конструкции вероятностного компьютера, то очевидны два главных требования.

1. Максимально гибкая и требующая минимум времени генерация случайного кода.

2. Максимальная элиминация вычислительных процедур в процессе реализации статистического моделирования.

Первое требование означает необходимость особого внимания к конструкции вероятностного процессора как специального генератора случайных кодов, управляемого адекватной системой команд. Второе требование является также глобальным и означает, что все вычислительные процедуры, относящиеся к решению конкретной задачи, должны в максимальной степени выноситься за пределы самого процесса статистического моделирования. Другими словами, если даже мы рассматриваем универсальный вероятностный компьютер, его подготовка к решению задачи состоит в предварительной «настройке» его памяти информацией, годной для немедленного использования в ходе реализации вероятностного алгоритма.

Рассмотрим еще одно важное свойство вероятностных алгоритмов, а именно их способность к распараллеливанию вычислений. Распараллеливание алгоритма решения задачи связано с параллельной работой управляющего алгоритма, реализующего расщепление процесса решения на подалгоритмы и координацию их выполнения. Между детерминированными и вероятностными алгоритмами в этом аспекте имеется существенное различие.

Жесткая определенность детерминированных алгоритмов влечет за собой строгую координацию подпроцессов и вызывает «перекачку» сложности распараллеливания в сложность управляющего алгоритма, контролирующего это распараллеливание.

В случае вероятностных алгоритмов наблюдается совершенно иная картина. Ввиду, как правило, слабой статистической зависимости между частями вычислительного процесса или даже их полной статистической независимости, распараллеливание процесса реализуется много проще. Принцип параллельных вычислений для вероятностных алгоритмов является совершенно естественным принципом, адекватным их природе.

Математическая модель задачи

Вероятностный процессор — это устройство, являющееся частью многопроцессорного компьютера, предназначенное для генерирования в определенном формате случайных кодов и их последовательностей с заданными вероятностными характеристикаии, обладающее управляемым входом, позволяющим изменять эти характеристики со скоростью, сопоставимой со скоростью операций самого устройства и, возможно, обладающее некоторыми другими подходящими функциями.

Мы рассматриваем абстрактный синтез ВП, сопровождаемый примерами решения некоторых конкретных задач. Ограничимся подчеркиванием только новых моментов в задаче синтеза. После того, как на уровне логического синтеза получена таблица функционирования детерминированного автомата, дальше используются традиционные методы синтеза.

Уточним постановку задачи. Проблема синтеза генератора случайных кодов всегда ставится как проблема синтеза некоторого детерминированного преобразователя начального стохастического процесса в другой процесс или множество процессов. В соответствии с разными типами «размещения» источников случайности в структуре устройства и уровня анализа информационных потоков в нем, возможно два подхода к описанию вероятностного процессора.

При первом подходе («разделенный» синтез) ВП рассматривается как последовательное соединение стандартного исходного источника случайных кодов и детерминированного автомата с конечным числом состояний, обладающего дополнительным управляющим входом. При втором подходе («распределенный» синтез) положение источника случайных кодов не локализовано. Синтезируемое устройство представляет собой конечный вероятностный автомат.

Вероятностный автомат, как это уже отмечалось выше, может быть представлен эквивалентным образом в виде последовательной композиции, так что основной является задача разделенного синтеза. Имеется большое число вариантов его решения. Мы рассмотрим здесь только некоторую классификацию этих подходов и некоторые примеры. Можно рассматривать следующие подходы к синтезу ВП.

— Приближенный синтез, когда задача считается решенной, если полученный ВП в состоянии генерировать случайные коды с вероятностными распределениями из заданного множества с наперед заданной точностью в определенном смысле.

— Предельный синтез, если полученный ВП генерирует последовательность случайных кодов, предельное распределение которой совпадает с некоторым заданным.

— Точный синтез, если полученный ВП генерирует случайный код, теоретическое распределение которого совпадает с требуемым из заданного семейства.

Легко видеть, что в конечном счете все подходы могут быть сведены к задаче точного синтеза, так что последний является основной задачей синтеза ВП.

Целесообразно рассмотреть все три варианта задачи точного синтеза: разделенный синтез, синхронный распределенный синтез и асинхронный распределенный синтез. В качестве достаточно удовлетворительной модели класса устройств для решения задачи выбраны классы детерминированных и синхронных вероятностных автоматов. Метод синтеза для случая асинхронного вероятностного автомата также разработан, но в этой работе он не рассматривается.

Сформулируем основную задачу синтеза управляемого источника. Пусть $\Sigma = \{\mathbf{p}(z), z \in Z\}$ — конечное множество стохастических векторов (дискретных распределений вероятностей), M — класс устройств, G — система критериев оптимальности устройства. Требуется определить:

1. подходящее множество случайных кодов $\boldsymbol{S} = \{\eta(z), z \in Z\}$, такое, что множество их распределений вероятностей совпадает с множеством Σ;
2. входной случайный код z;
3. детерминированный преобразователь $L(x, z)$, однотактный или многотактный, такой, что случайный код $L(\xi, z)$ совпадает со случайным кодом $\eta(z)$ для каждого значения $z \in Z$;
4. реализация преобразователя $L(x, z)$ в классе устройств M, удовлетворяющая критериям G.

Если случайные коды x и z с вероятностными распределениями соответственно $\mathbf{q}$ и $\mathbf{p}$ связаны между собой детерминированным преобразованием $L : L(z) = x$, тогда для их вероятностных распределений это означает существование подходящей простой матрицы H, такой, что $\mathbf{q}{=}\mathbf{p}H$. В таком случае мы говорим, что стохастический вектор $\mathbf{p}$ *имплицирует* стохастический вектор $\mathbf{q}$.

Отношение «$\mathbf{p}$ имплицирует $\mathbf{q}$» известно в теории вероятностей как «$\mathbf{q}$ измеримо относительно $\mathbf{p}$». Для логико-алгебраических целей термин «импликация» представляется более удобным и мнемоничным.

Таким образом, в этой терминологии задача синтеза сводится к конструированию подходящего стохастического вектора $\mathbf{p}$ и управляемого детерминированного преобразователя $L(\xi, z)$ (системы простых матриц $H(z)$), который имплицирует заданное множество стохастических векторов Σ.

Два аспекта синтеза ВП

Концепция вероятностной импликации дает средство для решения проблемы синтеза вероятностных преобразователей. К решению этой проблемы возможно два различных подхода. Для пояснения рассмотрим пример синтеза генератора случайных кодов без памяти.

Случайный код представляет собой некоторое соответствие между значениями кода и их вероятностями:

$$\xi = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_n \\ p_1 & p_2 & \dots & p_n \end{bmatrix}.$$

Преобразование одного случайного кода ξ в другой код η сопровождается параллельным преобразованием вероятностного распределения ξ в вероятностное распределение η:

$$\eta = \varphi(\xi),$$
$$p(\eta) = \varphi^*(p(\xi)).$$

Таким образом, импликация $(\mathbf{p} \longrightarrow \mathbf{q})$ связана с двумя типами функций. Первой является функция $\varphi^*(\mathbf{p})$ (она мультилинейна), которая реализует преобразование распределений. Вторая имеет форму $\varphi(\xi)$ и реализует преобразование случайных кодов η и ξ соответственно с вероятностными распределениями $\mathbf{p}$ и $\mathbf{q}$.

Эта двойственность позволяет ставить задачу синтеза вероятностного преобразователя и решать ее двумя путями. Обычно сначала синтезируется функция φ^*, а затем, с использованием знаний о функции φ^* синтезируется функция φ.

Некоторый произвол в выборе кодирования может быть использован для оптимизации решения. Но возможно, что кодирование задано заранее. Если импликация определена для случайных кодов ξ и η, тогда вначале синтезируется функция φ, а уже затем функция φ^*. Так что имеется два разных типа задач.

З а д а ч а 1. Пусть $\Sigma = \{\mathbf{q}(z), z \in Z\}$ конечное или счетное множество стохастических векторов. Требуется построить стохастичский вектор $\mathbf{p}$, такой, что

$$\mathbf{p} \longrightarrow \{\mathbf{q}(z), \quad z \in Z\}.$$

З а д а ч а 2. Требуется построить случайные коды — ξ с вероятностным распределением $\mathbf{p}$ и η с вероятностным распределением $\mathbf{q}$, такие, что

$$\eta = \varphi(\xi)$$

и преобразователь случайных кодов φ реализует преобразование φ^* их распределений.

Вначале строится вероятностный преобразователь (в форме функции φ^*, или, в случае преобразователя с памятью, в форме системы переходных матриц вероятностного автомата).

Затем конструируется сам преобразователь кодов (преобразующая функция вида $\varphi(\xi)$, или, в случае преобразователей с памятью, система функций переходов детерминированного автомата).

Таким образом, существуют следующие варианты синтеза ВП.

А. В семействе случайных кодов (устройства без памяти, т. е. класс контактных схем) имеется два варианта:

— F^*-синтез, затем кодирование или

— кодирование, затем F-синтез.

В. В семействе последовательностей случайных кодов (устройства с памятью, т. е. детерминированные или вероятностные автоматы) имеется много вариантов. Так же возможно рассматривать F^*-, или F-синтезы, или специальные смешанные формы, в случае вероятностных автоматов, возможно дополнительно рассматривать синхронный или асинхронный синтезы.

Кроме того, при структурном синтезе вероятностных преобразователей нужно различать «разделенный» и «распределенный» синтезы. В первом случае ВА конструируется как последовательное соединение управляемого генератора случайных кодов и детерминированного автомата (случайность и детерминированное преобразование «разделены»), во втором случае ВП конструируется из действительных вероятностных элементов «без выделения» случайности.

Общая методология синтеза

Разнообразие возможностей позволяет применять различные методы в соответствии с обстоятельствами конкретной задачи. Все эти методы представляют не только теоретический интерес, но применимы к синтезу реальных конструкций ВП. В частности, в основе методологии «распределенного синтеза» лежит следующая теорема.

Определение. Композиции ВА A и ВА B называются *структурно-изоморфными*, если между множествами подавтоматов

$$(A_1, A_2, \ldots, A_s) \text{ и } (B_1, B_2, \ldots, B_s)\,,$$

которые составляют соответственно автоматы A и B, и между множествами входов и выходов этих подавтоматов можно установить такое взаимнооднозначное соответствие, что входы и выходы соответствующих подавтоматов соответствуют друг другу.

Определение. Структура каскадной декомпозиции ВА A называется *детерминированно-порожденной*, если она имеет форму последовательного соединения некоторого управляемого генератора случайных кодов и детерминированного автомата B таким образом, что декомпозиция ВА A в вероятностные подавтоматы

$$(A_1, A_2, \ldots, A_s)$$

и ВА B в детерминированные подавтоматы

$$(B_1, B_2, \ldots, B_s)$$

структурно изоморфны.

Теорема. *Структура произвольной каскадной декомпозиции конечного ВА может быть сделана детерминированно-порожденной.*

Эта теорема позволяет реализовать любую каскадную декомпозицию ВА в форме последовательного соединения управляемого генератора случайных кодов и аналогичной каскадной декомпозиции детерминированного автомата.

Синтез на базе детерминированного автомата

Мы рассмотрим в этом разделе методологию синтеза на уровне преобразователей случайных кодов. Доказательства приведенных теорем, которые определяют метод синтеза ВП на основе множества состояний, можно найти в [9].

Пусть дана последовательность случайных кодов

$$\widetilde{\xi} : \xi_1, \xi_2, \ldots, \xi_n, \ldots,$$

где коды ξ_i принимают значения в одном и том же алфавите Y. Введем вероятности

$$P(\xi_1 = y_1, \xi_2 = y_2, \ldots, \xi_n = y_n) = \mu(y_1 y_2 \ldots y_n) = \mu(w),$$

где $w = y_1 y_2 \ldots y_n$.

О п р е д е л е н и е. Объект

$$J = < \mu(w), w \in Y^* >$$

называется *последовательностью случайных кодов*, если словарная функция $\mu(w)$

$$\mu : Y^* \longrightarrow [0, 1],$$

удовлетворяет условиям:

$$1) \mu(e) = 1,$$
$$2) \sum_{y \in Y} \mu(wy) = \mu(w).$$

Легко видеть, что эти условия эквивалентны условиям совместности Колмогорова и определяют случайную последовательность в обычном смысле.

Пусть даны две последовательности случайных кодов

$$\widetilde{\xi} : \xi_1, \xi_2, \ldots, \xi_n, \ldots,$$
$$\widetilde{\eta} : \eta_1, \eta_2, \ldots, \eta_n, \ldots$$

с системами вероятностей $< \mu_1(v), v \in X^* >$ и $< \mu_2(w), w \in Y^* >$, соответственно. Будем говорить, что последовательность случайных кодов $\widetilde{\xi}$ *последовательностно имплицирует* последовательность случайных кодов $\widetilde{\eta}$, если для каждого целого числа $n, n > 0$, начальная конечная подпоследовательность $\widetilde{\xi}_n = \xi_1, \xi_2, \ldots, \xi_n$ имплицирует начальную конечную подпоследовательность $\widetilde{\eta}_n = \eta_1, \eta_2, \ldots, \eta_n$

$$\widetilde{\xi}_n \longrightarrow \widetilde{\eta}_n, \quad n = 1, 2, \ldots$$

таким образом, что

1. для каждого целого n и каждой пары слов $v, v \in X^*$ и $w, w \in Y^*$, $|v| = |w|$ существует словарная функция φ_n такая, что

$$\mu_2(w) = \sum_{w = \varphi_n(v)} \mu_1(v),$$

2. для каждого целого числа n, слова v и символа $x, x \in X$

$$\varphi_{n+1}(vx) = \varphi_n(v)\varphi\prime(x).$$

Второе условие в определении последовательностной импликации последовательностей случайных кодов эквивалентно условию, что система словарных отображений φ_n определяет над целой полугруппой X^* такое автоматное отображение, что для каждого слова v_1, v_2 из X^* верны соотношения

$$\varphi(v_1 v_2) = \varphi(v_1)\varphi\prime(v_2).$$

Таким образом, может быть сформулирована следующая теорема.

Теорема. *Последовательность случайных кодов* $\widetilde{\xi}$ *последовательностно имплицирует последовательность случайных кодов* $\widetilde{\eta}$ *тогда и только тогда, когда существует детерминированный автомат, преобразующий последовательность* $\widetilde{\xi}$ *как входную последовательность в последовательность случайных кодов* $\widetilde{\eta}$ *как выходную последовательность.*

Другая форма этой теоремы (см. стр. 00?) описывает условия представимости последовательности пар случайных кодов в ВА в том частном случае, когда он является детерминированным.

Замечание. Для любого конечного (или счетного) множества конечных вероятностных распределений (стохастических векторов) $\Sigma = \{\bar{\boldsymbol{q}}_1, \bar{\boldsymbol{q}}_2, \ldots, \bar{\boldsymbol{q}}_s\}$ существует конечное вероятностное распределение (стохастический вектор) $\bar{\boldsymbol{q}}$, который имплицирует каждое распределение из множества Σ.

В выборе этого имплицирующего вектора $\bar{\boldsymbol{q}}$ существует большой произвол, который может быть использован для оптимизации решения. Аналогично, справедливо следующее замечание.

Замечание. Для любой последовательности случайных кодов $J = < \mu(w), w \in Y^* >$ с конечным алфавитом Y существует стационарная последовательность случайных кодов со взаимно-независимыми значениями $S = < \mu(v), v \in X^* >$, которая последовательностно имплицирует последовательность J.

Далее мы рассматриваем эту последовательность случайных кодов $J = < \mu(w), w \in Y^* >$ как выходную последовательность автономного вероятностного автомата. Другими словами, мы рассматриваем эту последовательность как многотактный канал, представленный в вероятностном автомате с единственной буквой входного алфавита.

Из последнего замечания вытекает следующая теорема.

Теорема. *Любая последовательность случайных кодов* $J = < \mu(w), w \in Y^* >$ *автоматна.*

Структура преобразования с конечным числом состояний

Интересно исследовать процедуру получения представляющего вероятностного автомата по данной последовательности случайных кодов. Мы руководствуемся представлением вероятностного автомата как последовательной композиции генератора случайных кодов и некоторого детерминированного автомата Мили. В случае автономного ВА управляемый генератор представляет собой стационарную последовательность случайных кодов со статистически независимыми значениями. Пусть последовательность случайных кодов

$$< \mu(w), w \in Y^* >$$

имеет стационарную имплицирующую последовательность

$$< \mu(v), v \in X^* >,$$

где для каждого слова $v = x_1 x_2 \ldots x_k$

$$\mu(x_1 x_2 \ldots x_k) = \mu(x_1)\mu(x_2)\ldots\mu(x_k)$$

и ДА D

$$D = < X, Y, S, \delta(s,x), \lambda(s,x) >$$

есть имплицирующий автомат. Представим пару функций (δ, λ) в форме $n\times n$ матрицы $C(y/x)$, где

$$c_{s_1,s_2}(y/x)=\begin{cases}1, & \text{если } s_2=\delta(s_1,x), y=\lambda(s_1,x),\\ 0 & \text{в противном случае.}\end{cases}$$

Тогда автономный ВА A, представляющий последовательность J как свою выходную последовательность, имеет переходную матрицу

$$A(y)=\sum_{x\in X}\mu(x)C(y/x).$$

Какова структура действий линейного оператора H_φ в случае последовательностной импликации $\widetilde{\xi}\longrightarrow\widetilde{\eta}$? Это преобразование реализуется некоторым детерминированным автоматом, который имеет функцию преобразования состояний $s\prime=\delta(s,x)$ и функцию выхода $y=\lambda(s,x)$. Поскольку здесь переменная x принимает случайные значения с распределением вероятностей $\bar{\boldsymbol{p}}(\xi_n)$ на n-м шаге процесса, мы получаем, что первая функция определяет стохастическую матрицу A, производящую перестановку состояний автомата. Функция выхода определяет некоторую простую (т. е. стохастическую, с элементами, равными 0 или 1) матрицу H_s, зависящую от текущего состояния автомата и реализующую линейный оператор преобразования входного распределения в выходное:

$$\bar{\boldsymbol{q}}(\eta_n)=\bar{\boldsymbol{p}}(\xi_n)H_s.$$

Таким образом, процесс последовательностной импликации одной последовательности случайных кодов в другую определяется двумя видами линейных операторов: стохастическая матрица A описывает процесс изменения состояний (индексов матриц H_s), а система простых матриц H_s определяет операторы типа φ^*, реализующие импликацию на текущем шаге процесса. Очевидно, что в общем случае преобразующий автомат имеет счетное число состояний, так что матрица A имеет счетный порядок, и мощность множества $\{H_s\}$ также счетна.

Мы видим, что произвольный автономный ВА порождает следующую систему матриц

$$\Xi=\{A, H_s, s\in S\},$$

где

1) A — стохастическая $|S|\times|S|$-матрица и

2) $H_s, s\in S$ суть $|Y|\times|X|$-простые матрицы.

Более того, верна следующая теорема:

Т е о р е м а. *Для произвольной системы матриц*

$$PA=\{A, H_s, s\in S\},$$

удовлетворяющей условиям 1) *и* 2), *существует ВА, порождающий эту систему* Ξ.

Таким образом, видно, что вместо системы $\Xi=\{A, H_s, s\in S\}$ можно рассматривать систему

$$\Delta=\{\bar{\boldsymbol{\alpha}},\quad C_x, x\in X,\quad H_s, s\in S\}$$

и можно сформулировать следующее следствие.

З а м е ч а н и е. Произвольная последовательность случайных кодов (или произвольный автономный ВА) взаимно-однозначно определяется системой типа Δ, где Y это алфавит последовательности (выходной алфавит ВА), X — некоторое конечное, а S — некоторое конечное или счетное множества, $\bar{\boldsymbol{\alpha}} = (\alpha_{x_1}, \alpha_{x_2}, \ldots, \alpha_{x_k})$, $x_i \in X$ — стохастический вектор, $C_{x_i}, x_i \in X$ и $H_s, s \in S$ — соответственно системы $|S| \times |S|$ и $|Y| \times |X|$ простых матриц.

Методология множества состояний

Следующее замечание будет играть в будущем изложении важную роль. Введем в рассмотрение словарную функцию $\mu_p(q)$ для фиксированного слова p из X^*, $\mu(p) \neq 0$, условием

$$\mu_p(q) = \mu(pq)/\mu(p).$$

Последовательность случайных кодов

$$J_p =< \mu_p(q), q \in X^* >$$

называется *p-состоянием* последовательности J.

П р и м е р 1. Рассмотрим стационарную последовательность случайных кодов со взаимно-независимыми значениями, т. е. для каждого слова $q = x_1 x_2 \ldots x_s$ выполняется равенство

$$\mu(x_1 x_2 \ldots x_s) = \mu(x_1)\mu(x_2) \ldots \mu(x_s).$$

Тогда для каждого слова p мы имеем

$$\mu_p(q) = \mu(pq)/\mu(p) = \mu(q),$$

т. е. последовательность J имеет единственное состояние.

П р и м е р 2. Рассмотрим однородную цепь Маркова с n состояниями, т. е., существует такое начальное вероятностное распределение $\mu(x)$ и стохастическая $n \times n$ матрица $A = (a_{x_i x_j})$, такая, что для любого слова p мы имеем

$$\mu(p) = \mu(x_1 x_2 \ldots x_s) = \mu(x_1) a_{x_1 x_2} a_{x_2 x_3} \ldots a_{x_{s-1} x_s}.$$

Тогда для состояния $J_p x$ получаем

$$\mu_p x(q) = a_{x x_{s+1}} a_{x_{s+1} x_{s+2}} \ldots a_{x_{k-1} x_k},$$

так что вероятность зависит только от значения x. Цепь имеет n состояний.

Из этих примеров не следует, что класс случайных последовательностей с конечным числом состояний совпадает с классом однородных марковских цепей.

З а м е ч а н и е. Последовательность случайных кодов J имеет конечное число состояний тогда и только тогда, когда существует такой конечный детерминированный автомат $D =< X, A, \delta(a, x) >$, что вероятности каждого начала p этой последовательности вычисляются в форме

$$\mu(p) = \mu(x_1 x_2 \ldots x_s) = \mu_{a_0 x_1} \mu_{a_1 x_2} \ldots \mu_{a_{s-1} x_s},$$

где последовательность индексов $a_0, a_1, \ldots a_{s-1}$ есть последовательность состояний автомата D для входного слова p.

Теорема, на которой базируется методология синтеза, формулируется так [9].

Теорема. *Детерминированный автомат Мили A, который преобразует данную последовательность случайных кодов $J_1 =< \mu_1(p), p \in \in X^* >$ в последовательность $J_2 =< \mu_2(p), p \in Y^* >$, существует тогда и только тогда, когда для каждого слова q из Y^* верно соотношение*

$$\mu_2(q) = \sum_{q=\varphi(p)} \mu(p),$$

где φ есть некоторое автоматное отображение. Число состояний отображения φ конечно тогда и только тогда, когда конечен автомат D.

Следствие. *Очевидно, что детерминированный автомат, преобразующий случайную последовательность J_1 в случайную последовательность J_2 конечен тогда и только тогда, когда автоматное отображение φ имеет конечное число состояний.*

Из этой теоремы вытекает метод синтеза детерминированного автомата, который преобразует последовательность J_1 в последовательность J_2. Для этого необходимо проанализировать отображение φ на соответствие условиям теоремы. Этот анализ приводит к определению всех состояний отображения. Далее задача сводится к синтезу детерминированного автомата D по его автоматному отображению.

В том случае, когда дано преобразоваание лишь начального сегмента последвательности J_1 в соответствующий начальный сегмент последовательности J_2, возможно доопределить это преобразование в полное автоматное отображение. Пусть усеченное отображение дано для всех слов p и q, имеющих длину не более n, и на этом фрагменте условия теоремы выполняются.

Например, можно доопределить отображение φ таким образом. Пусть слово p произвольной длины представлено в виде $p = p_1 p_2 \dots p_s p'$, где $|p_i| = n$ и $|p'| \neq n$. Тогда полагаем

$$\varphi(p_1 p_1 \dots p_s p') = \varphi(p_1)\varphi(p_2) \dots \varphi(p_s)\varphi(p').$$

Далее строится детерминированный автомат D, реализующий отображение φ.

Синтез на основе вероятностного автомата

Для детального обсуждения второго подхода к определению модели ВП рассмотрим определение вероятностного автомата.

Вероятностный автомат (общего вида) это объект

$$< X, Y, S, p(s', y/s, x) >,$$

где множества $X = \{x_1, x_2, \dots, x_n\}$ и $Y = \{y_1, y_2, \dots, y_n\}$ — соответственно множества входных и выходных букв, $S = \{s_1, s_2, \dots, s_k\}$ — конечное или счетное множество состояний автомата. Условное вероятностное распределение $p(s', y/s, x)$ определяет вероятность события, что ВА перейдет в последующее состояние s' и определит выходной символ y при условии, что автомат получил входной символ x будучи в текущем состоянии s. Ясно, что функция $p(s', y/s, x)$ удовлетворяет следующей системе условий:

$$0 \leqslant p(s', y/s, x) \leqslant 1, \qquad \sum_{s \in S, y \in Y} p(s', y/s, x) = 1, \quad (s, x) \in (S \times X).$$

Введем в рассмотрение $k \times k$-матрицы $A(y/x)$, где

$$a_{s,s'} = p(s', y/s, x),$$
$$A(y/x) = (') \,.$$

Элементы матриц $A(y/x)$ неотрицательны и не больше единицы. Очевидно, что

$$\sum_{y \in Y} A(y/x) = A(x)$$

— стохастическая матрица.

Пусть $w = x_1 x_2 \ldots x_t$ и $v = y_1 y_2 \ldots y_t$ — слова одинаковой длины в алфавитах соответственно X и Y. Введем обозначение

$$A(v/w) = A(y_1/x_1)A(y_2/x_2) \ldots A(y_t/x_t).$$

Если μ_0 — стохастический вектор, представляющий вероятностное распределение начальных состояний ВА, тогда вероятностное распределение состояний ВА после получения на вход символа x и при условии, что выходной буквой является y, описывается вектором

$$\mu(y/x) = \mu_0 A(y/x).$$

Для слов wx и vy соответствующий вектор $\mu(vy/wx)$ вычисляется так, как показано ниже:

$$\mu(vy/wx) = \mu(v/w)A(y/x).$$

Словарная функция двух аргументов

$$\tau(v/w) = \mu(e)A(v/w)\varepsilon.$$

называется входно-выходным отношением ВА $(A, \mu(e))$ или многотактным каналом, представленным в ВА A начальным вектором состояний $\mu(e)$. Число $\tau(v/w)$ есть вероятность того, что что ВА A отвечает реакцией v на входное слово w, при условии, что начальным вектором состояний является $\mu(e)$. Заметим, что $\tau(e/e) = 1$.

Введем обозначение $\tau(v/w) = A(v/w)\varepsilon$. Вектор-столбец $\tau(v/w)$ называется *поствектором состояний* ВА A. s-координата $\tau_s(v/w)$ поствектора $\tau(v/w)$ есть вероятность реакции ВА словом v, при получении на вход слова w и начале функционирования с фиксированного состояния s. Для каждой пары слов $|w_1| = |v_1|$, $|w_2| = |v_2|$, мы имеем

$$\tau(v_1 v_2/w_1 w_2) = A(v_1/w_1)\tau(v_2/w_2).$$

Задача синтеза ВП в соответствии с этой математической моделью ведет к проблеме синтеза ВА, представляющего данный многотактный канал $\tau_A(q/p) = \mu_0 A(q/p)e$.

Для каждой пары слов $|p| = |q|$ отношение

$$\tau_{p,q}(r/s) = \frac{\tau(qr/ps)}{\tau(q/p)}$$

называется (p, q)-*состоянием* многотактного канала τ_A, если $\tau_A(q/p) \neq 0$.

Рассмотрим два метода синтеза. Первый предполагает, что канал $\tau(q/p)$ имеет конечное число состояний. Во втором случае мы исходим из того, что существует конечный вероятностный автомат, представляющий канал τ.

Итак, предположим, что данный многотактный канал J имеет конечное число различных состояний.

Лемма. *Для каждой пары слов* $|p_1| = |q_1|$, $|p_2| = |q_2|$, $p_1, p_2 \in X^*$, $q_1, q_2 \in Y^*$ *и любой пары символов* (x, y) , $x \in X$, $y \in Y$ *из равенства* (p_1, q_1)*- и* (p_2, q_2)*-состояний многотактного канала* J *вытекает равенство состояний* $(p_1 x, q_1 y)$ *и* $(p_2 x, q_2 y)$.

Таким образом, можно ввести эквивалентность, сопоставляющую всем равным (p, q)-состояниям канала J некоторый один класс эквивалентности $a(J_{p,q})$. Возьмем это конечное множество классов эквивалентности в качестве множества состояний автомата A. Тогда переходные матрицы автомата $A(y/x)$ можно определить как

$$p_{aa'}(y/x) = \mu(a'/a, x, y)\mu(y/a, x),$$

где предполагается, что

$$\mu(a'/a, x, y) = \begin{cases} 1, & \text{если } a = \tau_{a(p,q)}, a' = \tau_a(px, qy), \\ 1, & \text{если } a = \tau, a' = \tau_a(p, q), \\ 0 & \text{в противном случае} \end{cases}$$

и

$$\mu(y/a, x) = \begin{cases} \tau_{p,q}(y/x), & \text{если } a = \tau_a(p, q), \\ \tau(y/x), & \text{если } a = \tau. \end{cases}$$

В качестве начального состояния выбирается класс a_0, который содержит сам канал J.

Другой метод синтеза более сложен, но дает ВА, имеющий минимальное число состояний. Для его рассмотрения необходимо ввести еще ряд определений.

Определим систему матриц $D(q/p)$ над линейным пространством $E(J)$, натянутым на множество всех состояний многотактного канала $J = <\tau(q/p)>$ условиями

$$d_{(p_1,q_1)(p_2,q_2)} = \begin{cases} 1, & \text{если } p_1 = pp_2, q_1 = qq_2, \\ 0 & \text{в противном случае.} \end{cases}$$

Эти матрицы определяют линейные преобразования многотактного канала τ таким образом, что для каждой пары слов (p, q), $|p| = |q|$, $p \in X^*$, $q \in Y^*$ верно равенство

$$\tau D(q/p) = \tau(q/p)\tau_{p,q}.$$

Это преобразование назовем (p, q)-*вращением* $E(J)$ относительно свободной полугрппы $(X, Y)^*$.

Определение. Множество многотактных каналов Ω, $\Omega \subset E(J)$ называется *выпуклым относительно* (p, q)*-вращения*, если результат (p, q)-вращения каждого элемента множества Ω принадлежит выпуклой линейной оболочке Ω.

Определение. Множество $\Gamma(\Omega)$ называется *опорным множеством* для множества Ω в линейном пространстве $E(J)$, если выпуклая линейная оболочка Γ содержит множество Ω.

Тогда критерий представимости многотактного канала τ в конечном ВА описывает следующая теорема.

Теорема. *Для того, чтобы многотактный канал τ был представим в конечном ВА, необходимо и достаточно, чтобы для множества состояний канала существовало конечное опорное множество F, выпуклое относительно полугруппы всех (p, q)-вращений.*

Если множество F конечно и имеет N элементов, тогда можно построить ВА с N состояниями, представляющий канал τ. Таким образом, проблема минимизации числа состояний автомата, представляющего канал τ , сводится к проблеме конструирования множества Г минимальной мощности. Если какое-либо конечное опорное множество уже известно, то проблема дальнейшей минимизации решена полностью.

Заключение

Несмотря на кажущуюся неконструктивность случайности, она может эффективно использоваться в позитивных целях. Более того, наличие большого числа степеней свободы в поведении случайных объектов дает ряд преимуществ, которые очень важны в моделировании поведения сложных систем. Возможности беспрепятственной реализации в вероятностных вычислениях параллельной и конвейерной обработки данных, возможности свободной суперпозиции вероятностных компьютеров в сетях ВА и ВП настолько эффективны, что это многократно перекрывает их недостатки. Если сказать кратко, можно видеть, что наоборот, вероятностный подход к вычислениям позволяет элиминировать некоторые недочеты, присущие детерминированным компьютерам, а именно, следующие.

1. Слабая функциональная надежность. С увеличением сложности детерминированного устройства, его надежность неизбежно снижается. Основное качество вероятности состоит в том, что «сбой», являющийся серьезной и неудалимой помехой при детерминированных вычислениях, является конструктивно учитываемым элементом в случае вероятностных вычислений. Как следствие, мы имеем высокую и регулируемую надежность выхода в случае вероятностных вычислений. На основе вероятностных устройств мы имеем возможность конструировать сколь угодно сложные системы без потери надежности. Этот принцип работает, в частности, в природе в процессе естественной эволюции.

2. Трудности распараллеливания детерминированных алгоритмов общего вида. Как следствие, с ростом сложности алгоритмов возрастает время вычислений, их скорость и надежность падают. Несмотря на растущую скорость современных вычислительных процессоров, ситуация в целом остается обманчивой и существуют пределы сложности, скорости и надежности. В противоположность этому, вероятностные алгоритмы очень легко поддаются мультипроцессированию и это обстоятельство создает предпосылки для создания сколь угодно сложных и сколь угодно надежных устройств.

3. Слабые возможности использования избыточности в детерминированных вычислениях. В вероятностных вычислениях эффективное использование избыточности является основой всех их позитивных качеств. Этот принцип также работает в природных процессах.

Вероятностный компьютер, представляющий собой сеть универсальных или специализированных вероятностных процессоров, был бы свободен от отмеченных недостатков. Это был бы гибридный компьютер в том смысле, что данные циркулировали в его каналах в виде вероятности случайных сигналов, тогда как конструкция машины была бы дискретной и детерминированной. Такой компьютер имеет гибкую функциональную систему и может функционировать как частично специализированный компьютер в соответствии с программой, связанной с ней в двух формах — как предварительная настройка ее памяти и структуры и как некоторый алгоритм функционирования, реализуемый в ходе выполнения машиной программы решения. Процесс решения состоит в получении статуса памяти, удовлетворяющего специальному критерию (самоуправляемый случайный поиск).

Вероятностных компьютеров описанного типа пока не существует, однако, имеются примеры отдельных опытов аппаратной реализации вероятностных алгоритмов [6–8]. Теория вероятностных автоматов, представляющая собой фундамент для конструривания вероятностных процессоров, к настоящему времени практически завершена [9]. Имеется значительное число проектов специализированных вероятностных компьютеров [1–5], в частности, разработан проект универсального вероятностного процессора (1992, [5]). В Казанском госуниверситете разработан комплекс технических требований и система комманд для такого вероятностного процессора, но практическая реализация этих условий требует более концентрированных усилий, чем это имеется сегодня. Цель настоящей статьи, в частности, состоит в том, чтобы обратить внимание на эту задачу.

Современная жизнь характерзуется постоянным нарастанием сложности возникающих проблем. В изучении поведения суперсложных систем методы статистического моделирования, в противоположность аналитическим методам, будут все более выступать на передний план.

Ориентация на моделирование в противовес численным решениям, имеет также философское основание. Природа не решает своих проблем методом вычислений. В частности, живые системы решают свои проблемы моделированием — посредством направленного случайного поиска на основе оценки точности результата.

Второй философский аргумент состоит в том, что в основе эволюции, как некоторого процесса усложнения, лежит случайность, а детерминированность проявляет себя лишь в качестве среднего значения параметров.

С этой точки зрения развитие теории конструирования вероятностных компьютеров является не только перспективной задачей, но и неизбежным делом. Теоретические основы для их конструирования полностью разработаны. Но переход к проявлению технологического интереса тормозится победным шествием современных компьютеров. Тем не менее, необходимо осознать, что движение в направлении вероятностных компьютеров и их сетей обосновано.

СПИСОК ЛИТЕРАТУРЫ

1. Бухараев Р. Г. Авторское свидетельство № 212628 от 13.12.67, СССР.
2. Бухараев Р. Г., Захаров В. М. Авторское свидетельство № 348112 от 7.5.70, СССР.
3. Бухараев Р. Г., Захаров В. М. Авторское свидетельство № 1223227 от 8.12.85, СССР.
4. Бухараев Р. Г., Захаров В. М. и др. Авторское свидетельство № 1524048 от 25.12.87, СССР.
5. Бухараев Р. Г., Захаров В. М. Авторское свидетельство № 1810888 от 10.10.92, Россия.
6. Гладкий В. С. Вероятностные вычислительные модели. — М.: Наука, 1973.
7. Хамитов Г. П. Имитация случайных процессов. — Иркутск, ИГУ, 1983.
8. Яковлев В. В., Федоров Р. Ф. Стохастические вычислительные машины. — Ленинград, «Машиностроение», 1974.
9. Bukharaev R. G. Theorie der stochastischen Automaten. — B. G. Teubner, Stuttgart, 1995.
10. Gaines B. R. Stochastic computing machinery // Electronics. — 1967. — № 14.
11. Metropolis N., Ulam S. The Monte-Carlo method // Journ. Am. Stat. Ass. — V. 44, № 247. — 1949. — P. 335–341.
12. Ulam S. On the Monte-Carlo method // Proc. of a 2nd Symposium on large scale digital calculating machinery. — Harvard Univ.Press. — 1949. — P. 207–212.

Поступило в редакцию 25 V 2006