распознавания объектов, и, в-третьих, избавиться от проблемы получения отрицательных значений вероятностей, что характерно для метода аппроксимации Бахадура-Лазарсфельда.

Список литературы

1. Козюра В.Д. Статистическая модель распознавания речевых сигналов. Вісник ДУІКТ, Том 6, № 3, 2008. - с. 229-234.

2. Дуда Р., Харт П. Распознавание образов и анализ сцен. Пер с англ. – М.: Мир, 1976.

Chow C.K., Liu C.N. Approximating Discrete Probability Distributions with Dependence Trees. IEEE Transactions of Information Theory, Vol. IT-14, No. 3, May 1968. - p. 462-467.

Поступила 28.01.2009

УДК 004.681.3:519.67

Дахно Н.Б., Тискина Е.О., Хорошко В.А.

АЛГОРИТМЫ ОБРАЩЕНИЯ МАТРИЦ ДЛЯ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ СИСТЕМ ЗАЩИТЫ ИНФОРМАЦИИ

Вступление

Повышение эффективности математического моделирования сложных технических систем, к которым относятся и системы защиты информации, можно обеспечить за счет использования вычислительных структур (ВС) с параллельной обработкой данных. Эта необходимость стимулирует разработку методов и алгоритмов решения систем уравнений большой размерности и алгоритмов обращения матриц, допускающих распараллеливания вычислений. Однако вопросы параллельной реализации обращения матриц в литературе уделено недостаточное внимание.

Основная часть

Исходя из этого и для решения этой задачи рассматриваются алгоритмы обращения невыраженной квадратной матрицы $A=[a_{ij}](i,j=1,2,...,n)$ большой размерности. Эти алгоритмы базируются на использовании метода «цифра за цифрой» [1,2] с целью организации параллельных вычислений.

Алгоритм 1. Обращение матрицы с помощью алгоритма Гаусса по методу «цифра за цифрой».

Для нахождения обратной матрицы $A^{-1}=[x_{ij}]$ используется основное соотношение $AxA^{-1}=1$, где 1-единичная матрица размером nxn. Так, для матрицы размером 4x4 основное соотношение в матричной форме имеет вид:

$$A\times A^{-1}=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \times \begin{matrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{matrix} = \begin{matrix} 1 & - & - & - \\ - & 1 & - & - \\ - & - & 1 & - \\ - & - & - & 1 \end{matrix},$$

из которого получается четыре системы уравнений с 16 неизвестными элементами x_{ij} обратной матрицы $A^{-1}=[x_{ij}]$ $(i,j=1,2,3,4)$. Матричную форму основного соотношения представим следующим образом:

$$\begin{matrix} a_{11} & a_{12} & a_{13} & a_{14} & x_{11} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} & x_{41} \end{matrix} \times \begin{matrix} 1 \\ 0 \\ 0 \\ 0 \end{matrix} = \begin{matrix} a_{11} & a_{12} & a_{13} & a_{14} & x_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} & x_{44} \end{matrix} \times \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \end{matrix}; \ldots \quad (1)$$

Решение матричных форм (1) по методу «цифра за цифрой» [1,2] осуществляется по формуле

$$x_{ij}^{t} = \varepsilon_j^{t} \; sign \; a_{ij}[-sign(\varepsilon_i^{t})] \; ,$$

где

$$\varepsilon_j = \begin{cases} 0, & если \quad |\varepsilon_i^{t}| < 2^{-t} \quad при \quad i \neq j; \\ 1, & если \quad |\varepsilon_i^{t}| \geq 2^{-t} \quad при \quad i = j; \end{cases} \quad (2)$$

$$\varepsilon_i^{t+1} = (\varepsilon_i^{t} + \sum_{j=1}^{n} a_{ij} x_{ij}) 2^{-t}; \; \varepsilon_i^{0} = -1; \; t = 0,1,2,\ldots; \; i = \overline{1,n}.$$

Здесь ε_i - значение навязки обратной матрицы. Условием перехода на следующий (t+1)-й шаг процесса при соответствующем изменении модуля шага является выполнение соотношения

$$|\varepsilon_i^{t+1}| \leq 0,5|\varepsilon^{t}|,$$

т.е. переход на следующий шаг осуществляется в том случае, если на данном шаге значение каждой навязки уменьшается, как минимум, в два раза (старший разряд обнуляется: $\sum_{j=1}^{n} a_{ij} x_{ij}^{t}$). Такое поразрядное уменьшение навязок обратной матрицы, сводящее их к нулю с необходимой точностью, позволяет определять одну верную цифру в значениях обратной матрицы на каждом шаге.

Для структурной реализации формулы (2) приводятся к виду:

$$x_{ij}^{t} = \varepsilon_j^{t}[-sign(\varepsilon_i^{t})],$$

где

$$\varepsilon_j^{t} = \begin{cases} 0, & если \quad |\varepsilon_i^{t}| < 1,0 \quad при \quad i \neq j; \\ 1, & если \quad |\varepsilon_i^{t}| \geq 1,0 \quad при \quad i = j; \end{cases}$$

$$\varepsilon_i^{t+1} = (\varepsilon_i^{t} + \sum_{j=1}^{n} a_{ij} x_{ij}) 2^{-t}; \; \varepsilon_i = -1.$$

На первом этапе выполнения алгоритма 1 обращения матрицы определяется набор неизвестных элементов x_{ij}, на втором – значения новых навязок ε_i. Выполнение первого этапа сводится к анализу знакового и старшего значащего разрядов навязок ε_i обратной матрицы, а второго этапа – к суммированию значений коэффициентов a_{ij} в каждом уравнении со значением навязки ε_i и сдвигу значения этой суммы влево на один разряд. Все математические операции в алгоритме 1 сведены и легко выполнимы средствами цифровой вычислительной техники операциям сложения и сдвига, что позволяет использовать алгоритм 1 как в спецпроцессорах при структурной реализации, так и в микро ЭВМ при

программной реализации. При структурной реализации алгоритм 1 обеспечивает распараллеливание процесса выполнения верных цифр в значениях обратной матрицы на каждом шаге и, следовательно, представляет собой N одинаковых параллельных ВС.

Алгоритм 2. Обращение матрицы на основе LU-разложения по методу «цифра за цифрой».

Разложение матрицы A на множители, т.е. представление ее в виде произведения нижней треугольной матрицы L и верхней треугольной матрицы U осуществляется по формуле:

$$U_{ik} = a_{ik} - \sum_{j=1}^{i-1} l_{ij} u_{jk} \text{ для } i \le k \le n;$$

$$l_{ki} = (a_{ki} - \sum_{j=1}^{i-1} l_{kj} u_{ji}) / u_{ii} \text{ для } i < k \le n; \quad (3)$$

$$l_{ki} = 1;\ i = \overline{1,n}.$$

Выражение (3) может быть преобразовано к виду, удобному для реализации на параллельных ВС:

$$u_{tj} = a^t_{tj} \text{ при } j \ge t;$$

$$l_{tj} = a^t_{ik} / u_{tt} \text{ при } i > t;$$

$$a^{t+1}_{ij} = a^t_{ij} - l_{it} u_{tj} \text{ при } i, j > t.$$

Для обращения матрицы A на основании LU-разложения введем обозначения $L^{-1} = M = [m_{ij}]$ и $U^{-1} = V = [v_{ij}]$, о соотношение $L \times M = 1$ используем для определения матрицы. Элементы матрицы M вычисляются по формуле:

$$m_{kk} = 1 \text{ для } k = \overline{1,n};$$

$$m_{ij} = -\sum_{k=j}^{i-1} l_{ik} m_{kj} \text{ для } 1 \le j < i \le n. \quad (4)$$

Для организации параллельных вычислений выражение (4) можно представить в следующем рекуррентном виде:

$$m_{tt} = m^t_{tt} = 1;\ m^t_{tt} = 0 \text{ для } i > t;$$

$$m_{tj} = m^t_{tj} \text{ для } j < t; \quad (5)$$

$$m^{t+1}_{ij} = m^t_{ij} - l_{it} m_{tj} \text{ для } i > t;\ j \le t.$$

При определении матрицы V используем соотношение V-U=1, а элементы матрицы V выражаются формулами:

$$V_{kk} = 1 / U_{kk} \text{ для } k = \overline{1,n};$$

$$V_{ij} = (\sum_{k=1}^{j-1} V_{ik} U_{kj}) / U_{ij} \text{ для } 1 \le i < j \le n,$$

которая может быть представлена в следующем рекуррентном виде:

$$V^t_{tt} = 1;\ V^t_{ij} = 0 \text{ для } j < t;$$

$$V_{it} = V^t_{it} / U_{tt} \text{ для } i \le t; \quad (6)$$

$$V^{t+1}_{ij} = V^t_{ij} - V_{it} U_{tj} \text{ для } i \le t;\ j > t;\ t = \overline{1,n}.$$

При $t = n$ обратная матрица L,U вычисляется в результате решения уравнений (5) и (6).

Таким образом, обращение матрицы A можно выполнить:

$$A^{-1} = [x_{ij}] = \sum_{k=1}^{n} v_{ik} m_{kj} = \sum_{k=\max\{i,j\}}^{n} v_{ik} m_{kj} . \qquad (7)$$

Для параллельных выражений соотношение (6) представим в виде:

$$x_{it}^{t} = x_{tj}^{t} = 0 \text{ при } i, j \le t ;$$

$$x_{ij}^{t+1} = x_{ij}^{t} - v_{it} m_{tj} \text{ для } i, j \le t ;\ t = \overline{1,n} .$$

Следует отметить, что для нахождения обратной матрицы L^{-1} по выражению (5), обратной матрицы U^{-1} по выражению (6) и обратной матрицы $A^{-1} = U^{-1}L^{-1}$ по выражению (7) можно использовать метод «цифра за цифрой» так же, как и в алгоритме 1.

Алгоритм 3. Обращение матрицы на основе оптимального упорядочения разложения матрицы на треугольные сомножители.

Алгоритм обращения матрицы A рассмотрен в работе [3]. Суть его заключается в том, что выражение для обратной матрицы имеет вид:

$$A^{-1} = U_1, U_2, U_3, ..., U_{k-1}, D_k, L_{k-1} D_{k-1}, ..., L_1 D_1 . \qquad (8)$$

По выражению (8) на N процессорах вычисляются подматрицы U_i, L_i и D_i. Так же на N процессорах выполняются умножение пар подматриц $B_1 = U_1 U_2, ..., U_m$; $B_i = U_{k-1} D_k$; $B_{i+1} = L_k D_{k-1}, ...$; $B_N = L_1 D_1$, а затем умножение пар полученных подматриц $B_1 B_2, ..., B_i B_{i+1}, ..., B_{N-1} B_N$ и т.д.

В результате выполнения параллельно-последовательного умножения подматриц определяется обратная матрица A^{-1}. Очевидно, что умножение пар полученных подматриц $B_1 B_2, ..., B_i B_{i+1}, ..., B_{N-1} B_N$ можно осуществить по методу «цифра за цифрой» [4].

Алгоритм 4. Обращение матрицы на основе метода пополнения.

Метод пополнения для обращения матрицы [5] заключается в следующем. Исходная матрица $A=[a_{ij}]$ рассматривается как последний член последовательности $A_0 = 1, A_1, ..., A_n = A$, причем переход от матрицы A_{n-1} к матрице A_n осуществляется посредством замены n–й строки матрицы A_{n-1} на n–ю строку матрицы A. Таким образом, матрица A^{-1} получается в результате n–кратного применения описанного ранее вычислительного процесса. Пусть a_j^n есть j–й столбец матрицы A_n^{-1}, тогда переход от (n–1)-го шага к n шагу характеризуется выражением:

$$a_j^n = a_j^{n-1} - \frac{v_n^T a_j^{n-1}}{1 + v_n^T a_j^{n-1}} \cdot a_n^{n-1} , \qquad (9)$$

где $v_n = a_{n1}, a_{n,n-1}, ..., a_{nn}$; $j = \overline{1,n}$.

N значений $v_n^T a_j^{n-1}$ вычисляется параллельно (один раз), а результаты пересылаются сразу на все процессоры, работающие параллельно, а затем выполняются вычисления по выражению (9). Нетрудно показать, что умножение вектора на вектор, вектора на скаляр и скаляр на скаляр для параллельной организации вычислений можно осуществить по методу «цифра за цифрой», как показано в работе [4].

Алгоритм 5. Обращение матрицы на основе метода окаймления.

Вычислительная процедура для обращения матрицы A основана на идее окаймления [5] и представляется в виде:

$$A = \begin{matrix} a_{11} & a_{12} & \dots & a_{1,n-1} & a_{1n} \\ a_{21} & a_{22} & \dots & a_{1,n-1} & a_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n-1,1} & a_{n-1,2} & \dots & a_{n-1,n-1} & a_{n-1,n} \\ a_{n1} & a_{n2} & \dots & a_{n,n-1} & a_{nn} \end{matrix} = \begin{matrix} A_{n-1} & U_n \\ V_n & a_{nn} \end{matrix},$$

где $V_n = a_{n1}...a_{n,n-1}$; $U_n = a_{n1}...a_{n-1,n}$.

Обратную матрицу A^{-1} находим в виде окаймленной матрицы вида:

$$\begin{matrix} A_{n-1}^{-1} + \frac{A_{n-1}^{-1}U_nV_nA_{n-1}^{-1}}{\alpha_n} & \frac{-A_{n-1}^{-1}U_n}{\alpha_n} \\ \frac{-V_nA_{n-1}^{-1}}{\alpha_n} & \frac{1}{\alpha_n} \end{matrix},$$

где $\alpha_n = a_{nn} - V_nA_{n-1}^{-1}U_n$.

В алгоритме 5 умножение вектора на вектор, вектора на скаляр и скаляра на скаляр для параллельного вычисления в ВС можно осуществить по методу «цифра за цифрой», как показано в работе [4].

Оценка вычислительных затрат. Время вычисления обратной матрицы A^{-1} размером nxn по рассмотренным выше алгоритмам, с учетом времени выполнения ($t_{сл}$) операций сложения и сдвигов ($t_{сд}$) в машинных тактах определяется по формулам:

$$T_1 = n(t_{сл} + t_{сд})q;$$

$$T_2 = \frac{n}{2}(t_{сл} + t_{сд})q;$$

$$T_3 = 3n(t_{сл} + t_{сд})q;$$

$$T_4 = [(3n-1)t_{сл} + nt_{сд}]q;$$

$$T_5 = \frac{1}{2}[(n+2)(3n-1)(t_{сл} + t_{сд})]q,$$

где нижний индекс T указывает номер алгоритма. Для сравнения отметим, что традиционные прямые методы и алгоритмы обращения матрицы, требующие выполнения операций умножения и деления [5], реализуется в среднем за n^3q тактов, т.е. уступает приблизительно в n раз, например, алгоритму 1.

Достоинством предлагаемых алгоритмов является то, что они обладают регулярной и однородной структурой. Это обеспечивает гибкость при аппаратной реализации данных алгоритмов.

Выводы

Таким образом, приведенные результаты могут быть использованы при разработке эффективных алгоритмов математического моделирования сложных технических систем, следовательно, и технических систем защиты информации, а также для оценки эффективности использования различных параллельных ВС.

Список литературы

1. Скорик В.Н., Степанов А.Е., Хорошко В.А. Мультимикропроцессорные системы. – К.: Техніка, 1989. -192 с.

2. Михайлов В.А., Хорошко В.А. Принципы построения многопроцессорных систем и расчет их производительности /Зб. наук. праць Севастопольського військово-морського ОЧЗ інститут ім. П.С.Нахімова. Вип. 1(9), 2006. –С.102-106.

3. Нагорный Л.Я., Лебедев П.А. Устройство для решения системы линейных уравнений с разреженной матрицей. №813444 А.С. СССР. Опубл. 15.03.81. Бюл. №10.

4. Егоров Ф.И., Орленко В.С., Хорошко В.А. Вычислительные модули для системы защиты информации / Зб. наук. праць ВІ КНУ ім. Т.Шевченка. Вип. №11, 2008. –С.117-124.

Ковальова Ю.С., Плус Д.В., Хорошко В.О. Розподіл ресурсів у багаторубіжній системі захисту / Правове, нормативне та метрологічне забезпечення СЗІ, НТЗ. Вип. 8, 2004. –С.39-43.

Поступила 29.01.2009

УДК 511

Мохор В.В., Жилин А.В.

СВЕДЕНИЕ ЗАДАЧИ ФАКТОРИЗАЦИИ ЧИСЛА К РЕШЕНИЮ СИСТЕМЫ НЕЛИНЕЙНЫХ УРАВНЕНИЙ В АЛГЕБРЕ ЖЕГАЛКИНА

Одним из направлений в развитии современных средств защиты информации является использование асимметричных методов/алгоритмов шифрования. Классическим представителем таких алгоритмов является алгоритм RSA. При этом, как отмечает Б. Шнайер [1], «безопасность алгоритма RSA основана на трудоемкости разложения на множители (факторизации) больших чисел» и далее «предполагается, что восстановление открытого текста по шифртексту и открытому ключу равносильно разложению числа на два больших простых множителя». В этом контексте общие результаты в решении задачи факторизации представляют не только теоретический, но и практический интерес для сферы технической защиты информации

Представленный в работе [2] метод факторизации предполагает работу со структурой числа, которое представлено в двоичной форме, и основывается на правиле умножения чисел в двоичной системе счисления. Факторизуемое число Z представляется в виде вектора: $Z= \{ z_0, z_1, z_2,\ldots, z_i,\ldots, z_n\}$, где z_0 - старший разряд числа, z_n, - младший разряд числа. Факторизуемое число Z является произведением двух простых чисел X и Y, представляемых векторами $X= \{x_1, x_2,\ldots, x_j,\ldots, x_m\}$, и $Y= \{y_1, y_2,\ldots, y_j,\ldots, y_m\}$, где x_0, y_0 - старшие разряды, а x_m, y_m - младшие разряды чисел X и Y соответственно. Задача состоит в том, чтобы по известным значениям компонент вектора $Z= \{z_0, z_1, z_2,\ldots, z_i,\ldots, z_n\}$ найти неизвестные компоненты для векторов $X= \{x_0, x_1, x_2,\ldots, z_j,\ldots, x_m\}$, и $Y= \{y_0, y_1, y_2,\ldots, y_j,\ldots, y_m\}$. Классический метод умножения чисел «в столбик» $Z= X^* Y$ представим следующим образом:

						x_1	x_2	…	x_k	…	x_{m-1}	x_m
						y_1	y_2	…	y_k	…	y_{m-1}	y_m
						$x_1 y_m$	$x_2 y_m$	…	$x_k y_m$	…	$x_{m-1} y_m$	$x_m y_n$
					$x_1 y_{m-1}$	$x_2 y_{m-1}$	…	$x_k y_{m-1}$	…	$x_{m-1} y_{m-1}$	$x_m y_{m-1}$	
					…	…	…	…	…	…	…	
			$x_1 y_k$	$x_2 y_k$	…	$x_k y_k$	…	$x_{m-1} y_k$	$x_m y_k$			
		…	…	…	…	…	…	…	…			
		$x_1 y_2$	$x_2 y_2$	…	$x_k y_2$	…	$x_{m-1} y_2$	$x_m y_2$				
	$x_1 y_1$	$x_2 y_1$	…	$x_k y_1$	…	$x_{m-1} y_1$	$x_m y_1$					
z_0	z_1	z_2	…	…	…	…	z_m	…	…	…	…	z_{n-1}	z_n