РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
А В Т О М Е Т Р И Я
№ 5 1996

УДК 681.3

В. А. Райхлин
(Казань)

АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ ПРОЦЕССОРНЫХ МАТРИЦ ПРИ РАСПОЗНАВАНИИ ДВОИЧНЫХ ОБРАЗОВ

Предложены алгоритмы идентификации объектов структуризованных двоичных изображений для последовательной ЭВМ и процессорных матриц с байтово-последовательной обработкой. Дается сравнительная оценка скорости реализации процедуры в обоих случаях.

Введение. Повышение производительности нечисловой обработки массивов данных (поиск в таблицах, обработка булевых матриц и различных автоматных объектов, распознавание двоичных образов, сравнение и сортировка кодов, текстовая обработка, поддержка функций матобеспечения) связывается с развитием параллельных ОКМД (один поток команд и много потоков данных)-архитектур. Основными среди них являются [1] ассоциативные процессоры и процессорные матрицы с элементами ассоциативности: бит активности процессорного элемента (ПЭ), механизмы поиска и маскирования. Такие матрицы имеют хорошую перспективу реализации на современной технологической основе.

Параллельные ОКМД-архитектуры следует подразделять на системы с горизонтальной и вертикальной обработкой. В обоих случаях имеем параллельную обработку по словам, но в первом — параллельную по разрядам, во втором — разрядно-последовательную. Оба принципа совмещены в DAP-подобных матрицах [3, 4]. Для матрицы 64 × 64 обычным режимом обработки является разрядно-последовательный с параллельной обработкой 4 K слов (матричный режим). Однако возможна и параллельная обработка 64 слов разрядностью 64 (векторный режим).

За основу анализа в работе взят процессор массивов ассоциативного типа (ПМА) [5]. Для него характерны похожие режимы: векторный (параллельная обработка 32 слов по 4 байта) и байтово-последовательный (параллельная обработка 128 слов последовательно по байтам). В данном случае процессорный элемент байтовый (8-разрядный) с памятью 1 Кбайт. Состояние активности ПЭ (индикатор байта) устанавливается по итогам предыдущей операции сравнения либо задается из программы. Оно может передаваться от одного ПЭ к соседнему при сканировании матрицы индикаторов как единой строки. Адреса данных во всех ПЭ одинаковы. Поэтому каждый адрес отвечает определенному слою трехмерной памяти данных ПМА.

Выбор режима обработки процессорной матрицы неизменных размеров зависит от решаемой задачи. Вертикальная обработка наиболее хороша для цифровой обработки изображений и конечно-разностных вычислений [3], а горизонтальная — предпочтительна для нечисловых применений. Полученные для этого случая оценки на множестве представительных программ, частично отраженные в [5, 6], достаточно высоки. Вместе с тем существует класс нечисловых задач, для которых байтово-последовательная обработка представляется наиболее адекватной. Это идентификация объектов двоичных изображений в случае байтовой сетки кадра, что является предметом настоя-

щей статьи. Случай битовой сетки имеет свою специфику, которая отражена в [7].

1. **Постановка задачи.** Задача состоит в получении сравнительных оценок производительности при распознавании двоичных образов на последовательной ЭВМ (далее просто ЭВМ) и процессорных матрицах с байтово-последовательной обработкой на примере ПМА [5]. Эта задача решается при следующих ограничениях.

1.1. Анализируемое изображение трактуется как множество не пересекающихся в пространстве плоских объектов (двоичных матриц), координаты и типы которых заведомо неизвестны. Часть из них, принадлежащая определенному классу, идентифицируется в терминах объекты — координаты. Этот класс задан набором взаимонепокрываемых троичных эталонов таких, что каждый эталон покрывает все допустимые реализации соответствующего объекта. Размеры матриц эталонов $m^t \times n^t$, $t \in \{0, ..., \gamma - 1\}$ — номер объекта; γ — число типов объектов, кратны байту: $m^t = c^t n_1$, $n^t = d^t n_1$, $n_1 = 8$.

1.2. Изображение разбивается на кадры одинаковых размеров $Q \times L$ байт. Анализ выполняется последовательно, кадр за кадром. В каждом кадре распознаются только такие объекты, которые полностью входят в кадр. Чтобы избежать возможных вследствие этого потерь полезной информации, предусмотрено соответствующее перекрытие кадров [7]. С учетом возможностей ПМА предполагается: $Q \leq 112$, $L \leq 128$, т. е. максимальные размеры кадра 896×1024 бит.

1.3. Время формирования отдельных кадров анализируемого изображения и обменов между компонентами комплекса «базовая ЭВМ — ПМА» не учитывается. Влияние обменов в данном случае сравнительно невелико [7]. Формирование очередного кадра на базовой ЭВМ успевает завершиться за время обработки текущего кадра на процессоре массивов.

1.4. Каждый эталон естественным образом разбивается на элементарные фрагменты размерами $(1 \times 1)n_1$. Задача решается в два этапа. На первом этапе выполняется идентификация на множестве элементарных фрагментов. В итоге происходит сжатие (кодирование) кадра: каждый квадрат, 1×1 байт, заменяется двоичной байтной подстрокой, в которой записывается код (номер $s \in \{0, ..., \eta - 1\}$) символа, соответствующего данному элементарному фрагменту. На втором этапе решается частная задача распознавания пространственных лексем из $c^t \times d^t$ символов, соответствующих принятой кодировке эталонов.

1.5. Размеры всех объектов одинаковы и равны $c \times d$ байт. Расположение объектов «этажное» (рис. 1, *a*). Количество объектов на каждом «этаже» одинаково и равно ε, а сами объекты различны. Выделяются случаи «шашечного» расположения объектов (рис. 1, *b*) и предельно плотной упаковки (рис. 1, *c*). Между объектами действует фоновая помеха в принятом алфавите символов.

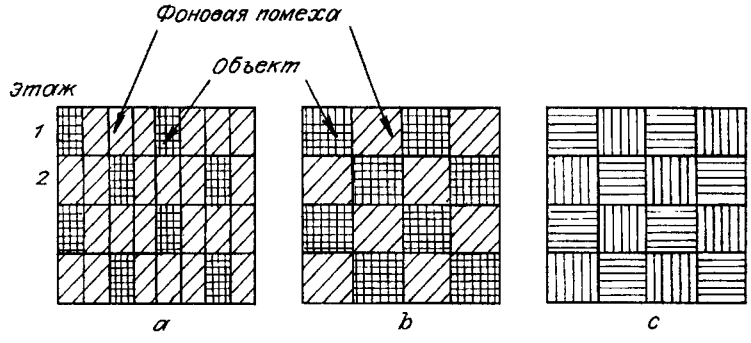

Рис. 1. Варианты расположения объектов:
a — «этажное» расположение, *b* — «шашечное» расположение объектов, *c* — предельно плотная упаковка

Варьируются размеры объектов, мощность алфавита символов, число распознаваемых объектов, величина ε.

1.6. ЭВМ имеет ту же длительность такта, что и ПМА. При этом короткие операции выполняются в ней однотактно, а работа с оперативной памятью занимает не менее пяти тактов.

2. **Алгоритмы первого этапа.** *Алгоритм для ЭВМ.* По условию двоичная развертка анализируемого кадра располагается в последовательных ячейках памяти. По аналогии со страницей машинописного текста (Q строк по L символов в строке) кадр рассматривается как плоская совокупность символов определенного алфавита мощностью η с координатами (j, i), $j = 0, ..., Q - 1$, $i = 0, ..., L - 1$. Двоичная развертка i-символа j-строки представлена последовательностью байтов памяти $V = V_0 + (8L)j + i + (L)k$, $k = 0, ..., 7$. Здесь k — порядковый номер строки в двоичном представлении символа. Каждый символ занимает 8 байт. Для k-строки любого символа в памяти ЭВМ хранится своя матрица инциденций (рис. 2). Элемент этой матрицы $\alpha^k_{s,z} = 1$, если z-содержимое k-строки анализируемого символа, представляет одну из реализаций k-строки s-эталонного символа. Число единиц в каждой строке матрицы соответствует множеству реализаций k-строки s-эталона. Например, для $[k]_s = 01{-}010{-}1$ имеем реализации $z = 73, 75, 105, 107$.

Содержимое k-анализируемой строки выделяет z-столбец k-матрицы инциденций. Следующая строка анализируемого символа определяет свой столбец $(k + 1)$-матрицы, а конъюнкция выделенных столбцов — номера эталонов, которым может соответствовать анализируемый символ. В шаге $k = 7$ конъюнктивный столбец будет содержать в точности одну единицу (это следует из условия взаимной непокрываемости эталонов), координата которой (s) укажет номер символа. Найденный код символа (j, i) заносится в байт памяти $W = W_0 + (L)j + i$, и так для каждого из QL анализируемых символов. Если итоговый конъюнктивный столбец оказывается нулевым, в байт W заносится код отсутствия идентификации.

Алгоритм для ПМА. По условию в каждый слой памяти данных заносится полная двоичная строка анализируемого кадра. Поскольку $Q \leq 112$, то для размещения исходной информации используется не более 896 слоев. Некоторый резерв памяти необходим для размещения эталонов, соответствующих им масок и кодов. Дополнительно выделяется один рабочий слой для последовательного формирования результата идентификации по каждой символьной строке.

Первые восемь слоев памяти данных последовательно записываются в матрицу и опрашиваются побайтно соответствующими строчными фрагментами s-эталона, $s = 0, ..., \eta - 1$, с учетом маски. Наличие в индикаторе байтового модуля единицы после последнего опроса свидетельствует о том, что в соответствующей позиции исходного текста стоит символ, который покрывается данным эталоном. Принятый код символа записывается в отмеченные индикаторами аккумуляторы матрицы и далее помещается в соответствующие байты рабочего слоя памяти.

Процедура повторяется для нового эталона всего η раз. В итоге в рабочем слое будут записаны побайтно коды всех идентифицированных символов анализируемой строки. Код отсутствия идентификации определен начальной установкой рабочего слоя. По окончании анализа j-строки символов содержимое рабочего слоя переписывается в уже отработанный j-слой для хранения результата, и так для каждой символьной строки.

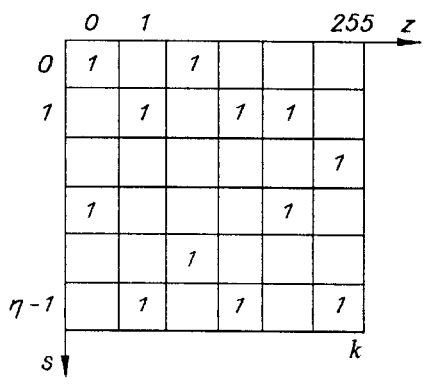

Рис. 2. Матрица k инциденций первого прохода

Характерно, что в данном случае анализ любой символьной строки проводится параллельно по i и последовательно по s. На ЭВМ, напротив, обработка ведется последовательно по i и параллельно по s. Поэтому сравнительная крайним верхним левым байтом. Сканирование области выполняется по байтам слева направо последовательно от верхней к нижней строке (рис. 3, *а*). Поэтому адрес (k, l)-байта области с координатами (j, i) $W = W_0 + (j + k)L + (i + l)$. Здесь $j = 0, ..., Q - c$, $i = 0, ..., L - d$, $k = 0, ..., c - 1$, $l = 0, ..., d - 1$.

Байту (k, l) любой области соответствует своя матрица инциденций (рис. 3, *b*). Элемент этой матрицы $\alpha_{t,s}^{k,l} = 1$, если s-символ, будучи расположен по координате (k, l)-области, входит в t-объект. В каждой строке матрицы содержится ровно одна единица. Коды символов и объектов отождествляются с их номерами. Содержимое каждого анализируемого байта дает номер некоторого символа и выделяет тем самым столбец соответствующей матрицы инциденций.

Следующий байт определяет один столбец своей матрицы инциденций, а конъюнкция выделенных столбцов — номера объектов, которые могут располагаться в данной области анализируемого кадра. Если на некотором шаге содержимое конъюнктивного столбца становится нулевым, анализ прекращается. При положительном исходе анализа конъюнктивный столбец на шаге cd будет содержать в точности одну единицу, координата которой укажет номер объекта. Каждому t-объекту отводится своя область памяти, в ней указываются координаты (j, i) локализации этого объекта на анализируемом кадре, по 2 байта на каждую локализацию.

Для уменьшения числа опросов используется двоичная матрица масок A размерами $Q \times L$ бит. Если при некоторых (j, i) происходит идентификация какого-либо объекта, в эту матрицу вводится единичный минор размерами $c \times d$ бит, левые верхние координаты которого суть (j, i). Матрица A заполняется единицами последовательно по мере сканирования строк символьного кадра. Поэтому, чтобы убедиться в отсутствии фрагментов ранее идентифицированных объектов в рассматриваемой области кадра, достаточно провести сравнение при $k = 0$. Если для некоторого l содержимое соответствующего элемента матрицы A равно 1, то просмотр области (j, i) следует прекратить и перейти к анализу области с координатой i, большей на $(l + d)$. Иначе про-

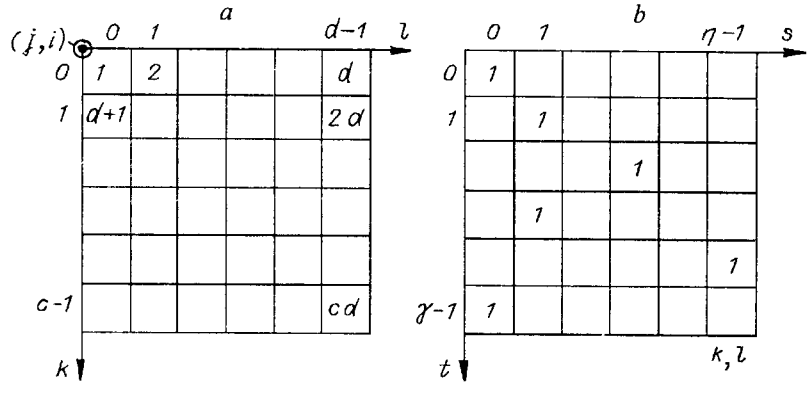

Рис. 3. Алгоритм второго прохода для ЭВМ:
а — (j, i)-область возможной локализации объекта; *b* — (k, l)-матрица инциденций второго прохода

водится описанная ранее процедура идентификации в данной области с использованием матриц инциденций. При отсутствии идентификации переходим к анализу области $(j, i + 1)$. В случае успешной идентификации очередная выбранная область — $(j, i + d)$, $i \leq L - d$. Невыполнение последнего условия требует перехода к строке $(j + 1)$.

Алгоритм для ПМА. Результаты первого этапа содержатся в j-слоях памяти данных, $j = 0, ..., Q - 1$. Это сжатое представление исходного кадра в символьном виде. Для каждого эталона идентификация проводится последовательно по байтам его символьного представления. При этом сканирование ведется сверху вниз и вправо (рис. 4).

Массив индикаторов условно развертывается в строку (сканирование слева направо и сверху вниз). Используются операции сдвига состояний индикаторов по этой строке на одну позицию вправо и записи содержимого некоторого слоя памяти или регистра признаков [5] в аккумуляторы матрицы по состояниям индикаторов. Эти состояния определяют и операции чтения матрицы в память, и операции сдвига содержимого матрицы вниз, которые также используются.

При идентификации t-объекта, $t = 0, ..., \gamma - 1$, в качестве признака для сравнения сначала берется байт **1** эталона. Поиск ведется параллельно по всем байтам слоя **0**. Результаты сравнения конъюнктивно отмечаются в индикаторах байтов, первоначально установленных в **1**. Затем аналогичная операция проводится для байта **2** по слою **1** и т. д. вплоть до испытания байта c на слое $c - 1$. По окончании сканирования первого байтового столбца эталона t состояния индикаторов сдвигаются на одну позицию вправо. Процесс повторяется для второго столбца вновь со слоя **0**.

Всякий раз перед переходом к очередному столбцу косвенным образом проводится анализ массива индикаторов на «$\neq 0$». Переход осуществляется только при положительном исходе анализа. В итоге cd-сравнений в индикаторах байтов получаем отметки о локализации t-объекта в срезе анализируемого кадра, включающем строки $0, ..., c - 1$. В соответствующие байты матрицы записывается код (номер) объекта t. Полученная информация заносится по состояниям индикаторов в рабочий слой памяти данных, первоначально установленный в 0, и так для каждого объекта. В итоге рабочий слой будет содержать полную информацию о типах и позициях объектов, локализованных в j-срезе кадра. Эта информация заносится в использованный слой j.

Указатели координат $i \in \{0, ..., L - 1\}$, по которым произошла идентификация при данном j, после специальной операции уплотнения сохраняются в отведенной для этого j-области памяти программ ПМА [5]. Затем значение j увеличивается на 1, что определяет новый начальный слой. Описанная ранее процедура повторяется $(Q - c + 1)$ раз при $j = 0, ..., Q - c$.

4. Сравнительные оценки быстродействия. Разработанные алгоритмы были детализованы до уровня машинных команд. Это позволило получить приближенные аналитические оценки числа команд ЭВМ ($N_{э1, 2}$) и ПМА ($N_{п1, 2}$) для первого и второго этапов. С учетом состава команд на множестве рассмотренных примеров найдено среднее число тактов выполнения одной команды первого ($\tau_{э1} = 4{,}4$; $\tau_{п1} = 1{,}61$) и второго ($\tau_{э2} = 3{,}3$; $\tau_{п2} = 1{,}56$) этапов. Искомая сравнительная оценка быстродействия:

$$\frac{T_э}{T_п} = \frac{T_{э1} + T_{э2}}{T_{п1} + T_{п2}} = \frac{\tau_{э1}N_{э1} + \tau_{э2}N_{э2}}{\tau_{п1}N_{п1} + \tau_{п2}N_{п2}},$$

ибо степень приближения в случаях ЭВМ и ПМА примерно одинакова.

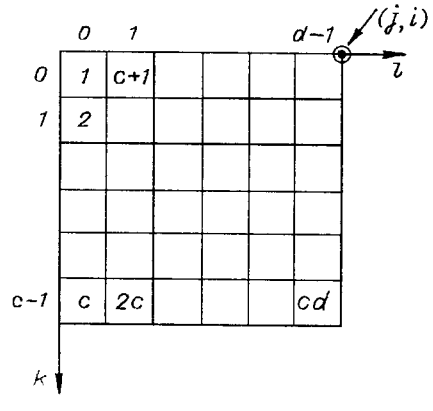

Рис. 4. Область (j, i) возможной локализации объекта для ПМА

Результаты вычислений для разных вариантов на рис. 1 приведены в табл. 1—4. Во всех случаях размеры анализируемого кадра максимальны: $Q = 112$, $L = 128$. Для табл. 1—3 значения $\eta = 54$, $\gamma = 64$, при этом расположение объектов «шашечное», с предельно плотной упаковкой и «этажное» ($c = d = 4$) соответственно, для табл. 4 — «шашечное» ($c = d = 4$). Полученные данные свидетельствуют о снижении эффективности использования ПМА с ростом размеров объектов (см. табл. 1, 2), плотности их упаковки (см. табл. 3), числа типов и мощности алфавита символов (см. табл. 4). Более высокая эффективность ПМА на первом этапе объясняется тем, что для этого этапа органичны операции с масками, которые хорошо реализуются на ПМА. Известный же параллелизм разработанного для ЭВМ алгоритма первого этапа начинает себя проявлять при достаточно больших η. На втором этапе маски отсутствуют. Эффект от использования матриц инциденций оказывается вы-

Таблица 1

$c \times d$	$N_{э1}$	$N_{э2}$	$N_{п1}$	$N_{п2}$	$\frac{T_{э1}}{T_{п1}}$	$\frac{T_{э2}}{T_{п2}}$	$\frac{T_э}{T_п}$
4 × 4	3971860	2810200	461000	611166	23,5	9,8	15,8
8 × 8	3971860	8486498	461000	1855720	23,5	9,7	12,5
16 × 16	3971860	27531510	461000	6157629	23,5	9,5	10,5
28 × 32	3971860	79424143	461000	18062254	23,5	9,4	9,7
56 × 64	3971860	129045105	461000	47082866	23,5	5,8	6

Таблица 2

$c \times d$	$N_{э1}$	$N_{э2}$	$N_{п1}$	$N_{п2}$	$\frac{T_{э1}}{T_{п1}}$	$\frac{T_{э2}}{T_{п2}}$	$\frac{T_э}{T_п}$
4 × 4	3971860	762666	461000	233260	23,5	7	18,1
16 × 16	3971860	597694	461000	498973	23,5	2,6	12,8
56 × 64	3971860	565214	461000	1703226	23,5	0,7	5,7
112 × 128	3971860	559449	461000	3263526	23,5	0,4	3,3

Таблица 3

ε	$N_{э1}$	$N_{э2}$	$N_{п1}$	$N_{п2}$	$\frac{T_{э1}}{T_{п1}}$	$\frac{T_{э2}}{T_{п2}}$	$\frac{T_э}{T_п}$
4	3971860	4279384	461000	578630	23,5	15,7	19,2
8	3971860	3785176	461000	589438	23,5	13,7	18,0
16	3971860	2810200	461000	611166	23,5	9,8	15,8

Таблица 4

η	γ	$N_{э1}$	$N_{э2}$	$N_{п1}$	$N_{п2}$	$\frac{T_{э1}}{T_{п1}}$	$\frac{T_{э2}}{T_{п2}}$	$\frac{T_э}{T_п}$
256	64	10709780	2810200	2180424	611166	13,4	9,8	12,6
54	64	3971860	2810200	461000	611166	23,5	9,8	15,8
54	256	3971860	7144624	461000	2306334	23,5	6,6	9,5

ше. Кроме того, удельный вес матричных команд [5] на первом этапе ПМА больше.

Заключение. Использованный способ оценки эффективности параллельного процессора путем сопоставления времени решения представительных программ на базовой ЭВМ и комплексе в целом при условии равенства длительностей тактов ЭВМ и ПМА технологически независим. Совершенствование технологии должно равно касаться всех компонентов комплекса, и это сохраняет сравнительные оценки для матрицы неизменных размеров.

Проведенный анализ показывает, что в случае байтово-последовательного распознавания двоичных образов производительность процессорных матриц размерами 32 × 32 бит и при глубине (числе слоев) памяти данных 1 К превышает минимум на порядок производительность универсальной ЭВМ с той же длительностью такта, если размеры объектов не более 128 × 128 бит на кадре 896 × 1024 бит. Дальнейшее укрупнение объектов снижает эффективность параллельного процессора. Влияние таких параметров задачи, как плотность расположения объектов в кадре, мощность алфавита символов, число типов объектов, менее значительно.

Необходимая глубина памяти (число строк Q обрабатываемого кадра) определяется максимальной «высотой» объекта c_{max}. Согласно найденным оценкам числа команд, выбор $Q > c_{max}$ слабо влияет на сравнительную оценку производительности. Влияние длины строки кадра L (размеров матрицы) весьма существенно, ибо значение $N_э$ растет примерно пропорционально L, а $N_п$ от L практически не зависит. Современные технологии допускают компактное исполнение процессорных матриц размерами до 128 × 128 бит [2], что соответствует значению $L = 2$ К. При этом, как показывают расчеты, в случае $c \times d = 112 \times 128$ (см. табл. 2) оценка $T_э/T_п$ улучшается от 3,3 до 44,9. Однако реальная оценка для комплекса будет несколько хуже из-за роста влияния времени формирования кадра на базовой ЭВМ и межпроцессорных обменов.

СПИСОК ЛИТЕРАТУРЫ

1. **Тербер К. Дж.** Архитектура высокопроизводительных вычислительных систем. М.: Наука, 1985.
2. **Blevins D. W., Davis E. W., Heaton R. A. et al.** BLITZEN: A highly integrated massively parallel machine // J. Parallel and Distributed Computing. 1990. 8, N 2. P. 399.
3. **Паркинсон Д.** Параллельные процессоры и их применение // Системы параллельной обработки /Под ред. Д. Ивенса. М.: Мир, 1985.
4. **Хокни Р., Джессхоуп К.** Параллельные ЭВМ: Архитектура, программирование и алгоритмы. М.: Радио и связь, 1986.
5. **Райхлин В. А., Медведев А. С., Мотягин В. Г. и др.** К исследованию эффективности комплектования универсальных ЭВМ средней производительности матричными процессорами ассоциативного типа // УСиМ. 1985. № 3.
6. **Райхлин В. А., Медведев А. С., Мотягин В. Г.** Вопросы разработки матричных компиляторов // Вычислительные системы. Новосибирск, 1981. № 89.
7. **Райхлин В. А.** Об использовании аппарата двумерного ассоциативного поиска в процессе распознавания // Проблемно-ориентированные средства повышения эффективности вычислительных систем. Казань: КАИ им. А. Н. Туполева, 1991.

Поступила в редакцию 17 июля 1995 г.