УДК 681.32.01:51

ОБ ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ АССОЦИАТИВНОЙ ПАМЯТИ В ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕДУРАХ

С.Я.Беркович, Ю.Я.Кочин, В.А.Молчанов

Полезность ассоциативного принципа организации памяти представляется довольно очевидной, однако, ввиду относительной сложности создания ассоциативных запоминающих устройств (АЗУ) при современном уровне развития технологии и их более высокой стоимости, целесообразность применения АЗУ требует подробного анализа.

Ассоциативное свойство памяти фактически помогает эффективно реализовать всего две процедуры: выделение подмножества слов, отвечающих заданному критерию опроса, и одновременное одинаковое изменение содержимого определенных разрядов выделенных слов. Первая процедура - поиск по точному соответствию - дополняется в случае необходимости разделением многозначного ответа; на основании этой процедуры, путем объединения или пересечения поразрядно выделяемых подмножеств производится также поиск слов в соответствии с отношением упорядочения (максимального или минимального, больших или меньших заданного, ближайшего или меньшего и т.д.) [1]. Помимо отмеченных простейших операций с помощью АЗУ можно эффективно выполнять ряд групповых операций преобразования массивов данных, в частности алгебраическое сложение числа с массивом; эти операции в равной мере используют обе ассоциативные процедуры [2].

При оценке непосредственного выигрыша во времени решения задачи от использования АЗУ, надо учесть большое количество разнообразных факторов, связанных с определением количества обращений к памяти, со скоростью выполнения тех или иных операций и с особенностями возможных вариантов организации вычислительного процесса. Поскольку точный учет таких факторов в общем случае невозможен, то мы ограничимся только рассмотрением асимптотического поведения времени решения при возрастании размеров задачи; при сравнении вычислительных устройств мы будем характеризовать алгоритмическую эффективность ассоциативной памяти порядком скорости роста отношения главных членов в указанных асимптотических выражениях.

В то время как технической реализации АЗУ посвящено большое количество работ, сведений по поводу анализа эффективности применения АЗУ относительно немного. Оказывается, что для большинства задач линейной алгебры и линейного программирования использование АЗУ не приводит к значительному увеличению скорости решения. Это определяется тем, что алгоритмы данных задач в большинстве случаев используют поэлементные преобразования информационных массивов. Основными процедурами, на которых основаны прямые методы решения задач, связанных с системами линейных уравнений и неравенств, являются алгоритмы исключения Гаусса и Жордана и их модификации (см., например, [3,4]). Поскольку в этих алгоритмах на каждом элементарном шаге преобразуется весь двумерный массив, то ассоциативные операции поиска не могут сколь-нибудь существенно сократить время решения.

В данной статье рассматривается вопрос об эффективности использования АЗУ для ряда экстремальных комбинаторных задач, а также для статистического моделирования. В этих задачах наиболее часто повторяющиеся вычислительные процедуры связаны с затратой большого количества действий на различные довольно сложные операции поиска при относительно малом объеме арифметических операций.

I. Экстремальные задачи комбинаторного типа

А) Венгерский метод. Основу этого метода составляет процедура нахождения максимального числа независимых допустимых элементов в матрице [5]. Ставится следующая задача: рассматривается совокупность всех клеток некоторой прямоугольной таблицы, которая произвольным образом разбивается на два множества, одно из них называют множеством допустимых

клеток, другое – соответственно множеством недопустимых клеток; требуется на множестве допустимых клеток разместить единицы ("I") таким образом, чтобы в каждом столбце и в каждой строке матрицы находилось не более одной "I" (такие клетки называют независимыми) и чтобы количество "I" для данного разбиения было бы максимальным. Алгоритм решения этой задачи основывается на теореме венгерских математиков Кенига и Эгервари: максимальное число независимых допустимых клеток равно минимальному числу рядов, покрывающих все допустимые клетки.

После первоначального построения некоторой совокупности независимых "I" производится итеративная процедура, в результате выполнения которой обнаруживается что либо, количество "I" максимально, либо их число может быть увеличено на единицу. В преобразовании, применяемом в основном шаге этой итеративной процедуры, производятся последовательные просмотры элементов матрицы и делаются пометки соответствующих рядов. При использовании ассоциативной памяти оказывается удобным осуществлять непосредственно пометку элементов матрицы. Количество действий, которое надо затратить в адресной памяти на достижение результата, эквивалентного результату, получаемому с помощью такого преобразования, пропорционально количеству элементов в строке или столбце матрицы – n, поскольку производимые при этом пометки рядов матрицы связаны с последовательным просмотром всех элементов ряда. В ассоциативной памяти пометки элементов выполняются одновременно, так что основное время при выполнении рассматриваемого преобразования тратится на выбор какого-нибудь из элементов преобразуемого множества клеток. При использовании любой процедуры разделения многозначного ответа в ассоциативной памяти на выбор первого слова следует затратить $\sim \log_2 M$ действий (здесь под M можно понимать некоторую величину либо порядка размера используемого массива, либо порядка количества слов во всей памяти, так что логарифм этой величины во многих случаях практически одинаков). Таким образом, предлагаемая организация вычислительного процесса при использовании ассоциативной памяти дает выигрыш в скорости $\sim \dfrac{n}{\log_2 M}$ раз.

Венгерский метод решения комбинаторных задач строится на основе двух вычислительных процедур, одна из которых – нахождение максимального множества независимых допустимых клеток – была описана выше, другая является преобразованием к новому множеству допустимых клеток в случае, когда полученное из старого максимальное множество независимых допустимых клеток не удовлетворяет требуемым условиям. Последняя процедура основана на поиске минимального числа в некотором массиве и заключается в пометке всех элементов матрицы, бо́льших этого числа, как в задаче о назначении на узкие места, либо в сложении и вычитании этого числа с элементами определенного набора строк и столбцов матрицы, как в задачах транспортного типа.

По сравнению с указанными операциями удельный вес остальных операций является незначительным; выигрыш в скорости выполнения этих операций при использовании ассоциативной памяти будет $\sim \frac{n}{\tau}$, где τ - разрядность чисел. Для процедуры нахождения максимального числа независимых допустимых клеток была получена оценка $\sim \frac{n}{\log_2 M}$; поскольку обычно $\tau \geq \log_2 M$, то для общего выигрыша в скорости решения комбинаторных задач венгерским методом при использовании ассоциативной памяти должна быть справедлива оценка $\sim \frac{n}{\tau}$.

Б) М е т о д п о т е н ц и а л о в. Алгоритм решения транспортной задачи методом потенциалов складывается из предварительного этапа и конечного числа однотипных итераций [6,7].

В данном вычислительном процессе такие процедуры, как определение значений потенциалов и перераспределение поставок по цепи, требуют количества действий, пропорционального сумме числа строк и столбцов матрицы, что существенно меньше количества других действий, которое пропорционально общему числу элементов матрицы. Вычисление характеристик элементов матрицы в адресной памяти требует обращения к каждому из них и, таким образом, $\sim n^2$ действий. В ассоциативной памяти достаточно произвести групповое вычитание потенциалов из строк и столбцов матрицы, так что выигрыш в скорости будет $\sim \frac{n}{\tau}$. Построение цепи для перераспределения поставок осуществляют, просматривая строки и столбцы матрицы и вычеркивая те из них, которые содержат не более одной клетки с записанной поставкой. Очевидно, что выигрыш в скорости на этой операции от использования ассоциативной памяти будет $\sim n$. Наименьший выигрыш в скорости выполнения отдельных вычислительных операций в методе потенциалов

$\sim \frac{n}{\tau}$, и, таким образом, н зависимо от относительной доли той или иной операции общий выигрыш в скорости решения всей транспортной задачи при использовании ассоциативной памяти будет по крайней мере $\sim \frac{n}{\tau}$.

В) А л г о р и т м Д ж о н с о н а в т е о р и и р а с п и с а н и й. Одним из примеров эффективного применения ассоциативной памяти является решение частного случая общей задачи теории расписаний для двух машин. Задача ставится следующим образом: имеются две машины и n деталей, каждая из деталей должна пройти обработку сначала на первой, а затем на второй машине; время обработки i-й детали на первой машине обозначим через a_i, на второй - через b_i; требуется составить расписание, т.е. перестановку $(i_1, i_2, \ldots, i_n)$ чисел $1, 2, \ldots, n$, минимизирующую время от начала обработки детали i_1 на первой машине до окончания обработки детали i_n на второй машине.

Решает эту задачу известный алгоритм Джонсона [8], который заключается в следующем. Ищется минимум среди всех чисел a_i и b_i. Если минимальное значение найдено среди чисел a_i, то соответствующая деталь ставится в расписании первой, если же минимум относится к b_i, то эта деталь обрабатывается последней. Числа a_i и b_i, относящиеся к рассмотренной детали, вычеркиваются, и операция поиска минимума повторяется; найденная деталь ставится в расписании на соответствующее место, определяемое тем же правилом.

При наиболее простой программной организации в адресной памяти на поиск каждого минимального числа надо затратить порядка n действий, и общее число действий в адресной памяти будет $\sim n^2$. В ассоциативной памяти на поиск каждого минимального числа надо затратить, вообще говоря, порядка τ действий, так что общее число действий будет $\sim n\tau$. Таким образом, в этом случае при реализации алгоритма Джонсона ассоциативная память обеспечивает выигрыш в скорости $\sim \frac{n}{\tau}$.

Следует отметить, что за счет усложнения программной организации и повышения расхода памяти в адресном ЗУ можно реализовать алгоритм Джонсона за $\sim n \log n$ действий; для этого следует предварительно упорядочить каждый из массивов $\{a_i\}$ и $\{b_i\}$ (с использованием дополнительной памяти), а также установить связи между их соответствующими компонентами, например, путем введения n-мерного вектора для фиксирования вычеркиваемых номеров. Однако если в ассоциативной памяти использовать усложненный метод упорядоченной выборки, такой как метод Стецюры-Левина [I], то на выбор одного числа придется затрачивать всего два такта обращения к устройству. В случае применения обоих указанных усложнений преимущество использования ассоциативной памяти для реализации алгоритма Джонсона сохраняется.

Г) Метод ветвей и границ [8]. Вычислительная схема метода может быть построена различными способами. Однако во всех случаях в нее включаются следующие основные процедуры: I) разбиение соответствующего дерева, 2) вычисление оценок (границ), 3) исключение подмножеств, не содержащих оптимального варианта.

Основное затруднение в организации вычислительного процесса для метода ветвей и границ с помощью адресной памяти вызывается процедурой поиска минимальной оценки в случае её использования для определения подмножества наиболее "перспективного" для деления, и процедурой поиска оценок, больших определенной величины, для исключения ненужных подмножеств. Это затруднение связано с тем, что появляющиеся в вершинах дерева величины оценок подмножеств вариантов не упорядочены, так что для выполнения указанных процедур надо, вообще говоря, на каждом шаге либо заносить оценки упорядоченным образом, либо производить полный просмотр всего поля памяти, отведенного под оценки. Это требует порядка M действий, где M – размер данного поля памяти. В ассоциативной памяти для выполнения указанных процедур надо затратить порядка τ действий. Таким образом, выигрыш в скорости выполнения рассматриваемых процедур при использовании ассоциативной памяти будет порядка $\sim \frac{M}{\tau}$; размер дерева подмножества вариантов непредсказуем, однако в общем случае он может быть довольно значительным, так что повышение скорости может быть весьма существенным.

Общий выигрыш в скорости решения задач методом ветвей и границ при использовании ассоциативной памяти оценить трудно из-за сильной нерегулярности этого метода, однако, учитывая большой выигрыш в скорости, получаемый в процедуре поиска на дереве подмножеств вариантов, можно считать, что этот общий выигрыш будет значительным.

2. Статистическое моделирование

Функциональные возможности АЗУ значительно расширяются при построении памяти на трехзначных элементах. Трехзначные запоминающие элементы кроме основных состояний "0" и "I" имеют ещё и третье (М-состояние), безразличное по отношению к ассоциативному опросу (вырабатывается сигнал совпадения при опросе как по "0", так и по "I"). Очевидно, помимо специальных схемных решений, трехзначный элемент может быть смоделирован с помощью двух двузначных.

Оказывается весьма эффективным использование АЗУ на трехзначных элементах при статистическом моделировании, а именно для организации случайных блужданий, т.е. переходов с заданными вероятностями на определенном множестве состояний, что является основной вычислительной процедурой при моделировании марковских цепей [9].

Пусть рассматриваемая система имеет n состояний и матрицу вероятностей перехода $\| P_{ik} \|$. Действия, которые надо выполнить при моделировании случайных блужданий, состоят в следующем: в текущий момент система находится в некотором состоянии i, нужно выбрать состояние k, в которое система перейдет в следующий момент. Для организации этой процедуры надо решить две задачи: I) задачу размещения информации, описывающей связи между состояниями системы (в больших системах матрица $\| P_{ik} \|$ содержит, как правило, значительное число нулевых элементов) и 2) задачу определения состояния, куда должна попасть система в соответствии с совокупностью вероятностей перехода $\{ P_{ik} \}$ ($k = I, 2, \ldots, n$).

При использовании ассоциативной памяти решение первой задачи является очень удобным: индексы i можно взять в качестве ассоциативных признаков, так что при обращении по такому признаку будет выделен сразу набор из n_i состояний, в которые может перейти система, т.е. для которых $P_{ik} \neq 0$. Для решения второй задачи – выбора состояния в соответствии с заданной вероятностью – предлагается существенно использовать трехзначность запоминающих элементов.

При подаче на вход АЗУ в качестве кодов опроса случайных равномерно распределенных двоичных τ-разрядных чисел вероят –

ность выбора некоторого слова из АЗУ на двузначных элементах, равна 2^{-r}, а из АЗУ на трехзначных элементах – 2^{-r+m}, где m – количество разрядов слова, находящихся в M – состоянии.

Очевидно, в одной числовой линейке АЗУ можно представить состояние, вероятность перехода в которое равняется только целой отрицательной степени двойки. В случае произвольного значения величины заданной вероятности, надо последнюю разложить на отдельные слагаемые, каждое из которых выражается целой отрицательной степенью двойки.

Для определения избыточности объема памяти требуется оценить максимально возможное число единиц во всей совокупности ℓ заданных чисел, представленных r-разрядными двоичными кодами в форме с фиксированной запятой.

Поскольку сумма рассматриваемых чисел определена, то максимальное количество единиц в их совокупности может быть получено посредством заполнения единицами младших разрядов. В каждом разряде всех чисел не может быть единиц больше, чем ℓ; количество разрядов q, которое можно заполнить, начиная с самого младшего, оценивается из соотношения:

$$\ell \sum_{i=r-q+1}^{r} 2^{-i} \approx 1, \qquad (1)$$

откуда

$$q \approx r - \log_2 \ell . \qquad (2)$$

Для представления ℓ состояний может потребоваться примерно не более чем ℓq числовых линеек. Таким образом, q есть верхняя оценка для относительной величины избыточности памяти. Из формулы (2) видно, что q, как и следовало ожидать, с увеличением r – увеличивается, а с увеличением ℓ – уменьшается. При характерных значениях параметров $\ell \sim 50 - 200$ и $r \sim 15 - 20$ не требуется увеличения объема памяти более чем на порядок.

Если величины вероятностей P_i разлагаются в сумму небольшого числа отрицательных степеней двойки, то необходимая избыточность памяти резко снижается. В ряде задач, использующих модели конечных цепей Маркова, оказывается, что для их решения важно лишь сохранить структуру переходов между состояниями, а сами значения переходных вероятностей, вообще говоря, несущественны [10]. Для таких задач применение ассоциативной памяти с трехзначными запоминающими элементами получается весьма эффективным.

Литература

1. КРАЙЗМЕР Л.П., БОРОДАЕВ Д.А., ГУТЕНМАХЕР Л.И., КУЗЬМИН Б.П., СМЕЛЯНСКИЙ И.Л. Ассоциативные запоминающие устройства. Л., "Энергия", 1967.

2. ПОПОВА Г.М., ПРАНГИШВИЛИ И.В. Ассоциативный параллельный процессор для групповой обработки данных. –"Автоматика и телемеханика". 1972, № 1.

3. ВОЕВОДИН В.В. Численные методы алгебры. Теория и алгорифмы. М., "Наука", 1966.

4. ЗУХОВИЦКИЙ С.И., РАДЧИК И.А. Математические методы сетевого планирования. М., "Наука", 1965.

5. ФОРД Л.Р, ФАЛКЕРСОН Д.Р. Потоки в сетях. М., "Мир", 1966.

6. ГОЛЬШТЕЙН Е.Г., ЮДИН Д.Б. Задачи линейного программирования транспортного типа. М., "Наука", 1969.

7. БИРМАН И.Я. Оптимальное программирование. М., "Экономика", 1968.

8. КОРБУТ А.А., ФИНКЕЛЬШТЕЙН Ю.Ю. Дискретное программирование. М., "Наука", 1969.

9. БУСЛЕНКО Н.П., ГОЛЕНКО Д.И., СОБОЛЬ И.М., СТРАГОВИЧ В.Г. ШРЕЙДЕР Ю.А. Метод статистических испытаний (метод Монте-Карло). М., ГИФМЛ, 1962.

10. ЕРМАКОВ С.М. Метод Монте-Карло и смежные вопросы. М., "Наука", 1971.

Поступила в ред.-изд.отд.
15 февраля 1973 года