Системный анализ

УДК 681.3

Т. Б. Мартынюк, В. В. Хомюк

МУЛЬТИОБРАБОТКА МАССИВОВ ДАННЫХ ПО РАЗНОСТНЫМ СРЕЗАМ

Ключевые слова: мультиобработка данных, пороговая обработка векторного массива данных, системы алгоритмических алгебр, разностный срез.

Key words: data multiprocessing, threshold processing of vector data array, systems of algorithmic algebras, difference cut.

Введение

Вопросы структурного программирования алгоритмов сортировки достаточно подробно освещены в ряде работ [1-5]. Эти публикации ещё раз подтвердили универсальность аппарата структурированных схем программ, его широкие возможности в плане анализа, классификации и создания новых алгоритмов с оптимальными характеристиками [2-4,6]. Дальнейшее развитие алгебраической алгоритмики способствует созданию и исследованию семейств алгебр, которые адекватны основным парадигмам программирования и предназначены для формализованного описания различных дискретных моделей – логических, теоретико-автоматных, грамматических, алгоритмических [7-9]. В известных публикациях системы алгоритмических алгебр (САА) В. М. Глушкова и их модификации в основном использованы для формализованного представления задач символьной мультиобработки, а именно, различных методов сортировки, которые являются наиболее наглядным и разнообразным примером мультиобработки.

Вместе с тем, одной из распространенных задач мультиобработки массива данных является параллельное суммирование элементов векторного массива [10], которое ещё называют операцией свертки вектора [11] или оператором группового суммирования [12]. Один из вариантов реализации этой операции с использованием разностных срезов рассмотрен в работе [13]. Особенностью такого подхода является дальнейшее развитие принципа многооперандности обработки векторного массива чисел, а также особое сочетание последовательного и параллельного выполнения базисных операций, что позволяет использовать специфические функциональные возможности разностно-срезовой обработки массивов данных [14,15].

В данной работе представлены результаты описания в базисе модифицированной САА В. М. Глушкова алгоритмов параллельного суммирования элементов векторного массива данных.

Модификация базиса САА В. М. Глушкова

Особенности операции суммирования элементов векторного массива M необходимо учесть, дополняя аппарат модифицированных САА [1-3]. Здесь также определяющим будет понятие размеченного массива M [2,3]. Пусть массив M имеет вид:

$$M = \ddot{I}\,\ddot{E} \rceil a_1 a_2 \ldots a_n *, \tag{1}$$

где ПЛ, $*$ – специальные маркеры, отмечающие соответственно левый и правый концы размеченного массива; $\rceil$ – указатель; n – размерность (длина) массива.

По классификации способа обработки элементов массива [1,3] в дальнейшем будет рассмотрена левосторонняя обработка массива M в результате циклического сдвига указателя $\rceil$ слева направо.

Наряду с такими известными конструкциями структурного программирования, как композиция, альтернатива и цикл [2,7], активно будет использоваться декомпозиция $(A + B)$ как операция параллельного применения операторов A и B, которая нашла применение при описании синхронных алгоритмов сортировки парным обменом [4]. Базисные условия (предикаты) включают известные по работам [2,3], а именно: а) $l \le r$ – истинное при выполнении указанного отношения для соседних элементов l и r массива; б) $d\,(*)$ – истинное при достижении указателем $\rceil$ маркера $*$, а также новое условие; в) γ - истинное при выполнении соотношения $\min = 0$ для минимального значения в массиве M .

На составе базисных операторов остановимся подробнее. Широко использоваться будут такие известные операторы, как оператор начальной расстановки указателей (НРУ), оператор завершения работы РС (ФИН), оператор сдвига указателя $\urcorner$ на один элемент вправо ($\vec{C}$). Для описания в терминах регулярных схем алгоритмов разностно-срезовой обработки и, в частности, для параллельного суммирования элементов векторного массива необходимо ввести следующие базисные операторы:

$C\acute{O}\grave{I}_2(b,c)$ – суммирование элементов b и c массива;

$\hat{A}\hat{U}\times_2(b,c)$ – вычитание (последовательное) элементов b и c массива;

$\hat{A}\hat{U}\times(M,a)$ – вычитание (параллельное) из всех элементов массива M элемента a;

$\hat{A}\hat{U}\hat{A}(R)$– вывод результата.

Представленный математический аппарат позволяет описать в терминах операторных представлений САА и проанализировать все способы (алгоритмы) реализации обработки векторного массива данных с использованием разностных срезов (РС).

Разностно-срезовая обработка

Основным понятием при разностно-срезовой обработке является понятие разностного среза (РС), данное в работе [13], которое определяет его в текущем цикле обработки как векторный массив, каждый элемент которого формируется как разность соответствующего элемента текущего массива и текущего внутреннего порога обработки, в качестве которого может использоваться элемент фиксированной позиции текущего массива, например, наименьший элемент.

В дальнейшем будем рассматривать векторные массивы чисел в качестве РС, обозначать их как A_j, где j – j-й цикл обработки РС, а для исходного РС вида A_0 примем, что все его элементы $a_{i,0}$ ненулевые и положительные числа.

Базовыми операциями для мультиобработки по РС являются [13, 14]:

а) формирование элементов вектора внутреннего порога обработки вида

$$\mathbf{q} = \left\{ q_j \right\}_{j=1}^{N}, \tag{2}$$

где N – количество циклов обработки, причем элемент q_j определяется следующим образом:

$$q_j = \min_i A_{j-1} = \min_i \left\{ a_{i,j-1} \right\}_{i=1}^{n}, \tag{3}$$

б) формирование текущего РС A_j вида:

$$A_j = A_{j-1} - q_j = \{a_{i,j-1} - q_j\}_{i=1}^{n}, \tag{4}$$

в) накопление частичных сумм S_j вида [14]:

$$S_N = \sum_{j=1}^{N} S_j = \sum_{j=1}^{N} q_j \left(\sum_{i=1}^{n} f_{i,j} \right), \tag{5}$$

где $f_{i,j}$ - i-й элемент j-го вектора бинарной матрицы $\boldsymbol{F}$ масок, который формируется одновременно с РС A_j следующим образом [13, 14]:

$$f_{i,j} = \begin{cases} 1, \ \text{åñëè} \ a_{i,j} \geq 0, \\ 0, \ \text{åñëè} \ a_{i,j} < 0. \end{cases} \tag{6}$$

Заканчивается процесс многооперандного суммирования элементов исходного РС A_0 при выполнении условия

$$q_j = 0. \tag{7}$$

Последовательное применение указанных базовых алгоритмов позволяет получить следующие результаты:

а) сформировать окончательную сумму элементов исходного РС A_0 как результат мультисуммирования [10]:

$$S_N = \sum_{i=1}^{n} a_{i,0} \tag{8}$$

в виде векторно-матричного умножения [13, 14]:

$$S_N = \left(\mathbf{n}^T \cdot \mathbf{F} \right) \cdot \mathbf{q}, \tag{9}$$

где $\mathbf{n}$ – единичный вектор; T – символ транспонирования;

б) отсортировать элементы исходного РС A_0 в порядке возрастания [14]:

$$A_0^s = \mathbf{G}^T \cdot A_0, \tag{10}$$

где $\mathbf{G}$ – матрица сортировки, элементы $g_{i,j}$ которой определяются следующим образом

$$g_{i,j} = \begin{cases} 1, \ \text{åñëè } a_{i,j} = 0, \\ 0, \ \text{åñëè } a_{i,j} \neq 0, \end{cases} \tag{11}$$

в) восстановить элементы исходного РС A_0 [14]:

$$A_0 = \mathbf{F} \cdot \mathbf{q} \, . \tag{12}$$

Кроме того, учитывая возможность совмещения в каждом j-м цикле операции формирования частичной суммы S_j и сравнения ее с текущим внешним порогом θ_j обработки [13], целесообразно ввести в рассмотрение пороговую функцию активации с формированием выходного сигнала Y вида:

$$\theta_j = \theta_{j-1} - S_j, \tag{13}$$

$$Y = \begin{cases} 1, \ \text{åñëè } \mathrm{S_N} \geq \theta_0, \\ 0, \ \text{åñëè } \mathrm{S_N} < \theta_0, \end{cases} \tag{14}$$

где θ_0 – начальный внешний порог обработки.

Соответственно, окончанием пороговой обработки является выполнение условия:

$$\theta_j \leq 0. \tag{15}$$

Таким образом, базовыми операциями разностно-срезовой обработки векторного массива данных являются операции вида (3) – (5), (13) и (14), а базовыми условиями – (7) и (15).

Описание алгоритмов разностно-срезовой обработки в терминах САА В. М. Глушкова

Принимая во внимание то, что в данной работе будут рассмотрены из выше перечисленных только процедуры мультисуммирования (8), мультисортировки (10) и пороговой обработки (14), а также базовая операция формирования РС (4), то целесообразно определить соответствующие составные операторы, которые базируются на использовании уже выделенных базисных операторов и базисных предикатов. В дальнейшем необходимо ввести обозначение для массива M с указанием его крайних элементов в виде $\left(\overline{a_1, a_n}\right)$ для выделения зоны составных операторов.

Для мультисуммирования (8) и мультисортировки (10) составные операторы имеют следующий вид:

а) составной оператор выделения минимального числа среди элементов $\left(\overline{a_1, a_n}\right)$ РС A_j:

$$\text{Ì ÈÍ}_2\left(\overline{a_1, a_n}\right) ::= \left[d(*)\right]\Big\{\left[l \le r\right]\Big(\text{ÓÑÒ}\big(\min = l\big) \vee \text{ÓÑÒ}\big(\min = r\big)\vec{C}\Big)\Big\}, \qquad (16)$$

с помощью которого реализуется последовательное выделение минимального элемента в каждой паре элементов массива $\left(\overline{a_1, a_n}\right)$ со сдвигом на один элемент вправо по массиву и с установкой (назначением) элементу min одного из двух значений;

б) составной оператор накопления частичных сумм элементов $\left(\overline{a_1, a_n}\right)$ РС A_j:

$$\text{Í ÀÊÎ Ï}_2\left(\overline{a_1, a_n}\right) ::= \left[d(*)\right]\Big\{\left[f_i = 0\right]\Big(E \vee \text{ÑÓÌ}_2\left(S, \min\right)\vec{C}\Big)\Big\}, \qquad (17)$$

с помощью которого реализуется последовательное накопление частичных сумм S_j вида (5) для чего используется проверка равенства нулю соответствующего элемента f_i (6) в текущем векторе бинарной матрицы $\mathbf{F}$ масок;

в) составной оператор формирования текущего вектора бинарной матрицы $\mathbf{F}$ масок:

$$\text{Ì ÀÑÊÀ}\left(\overline{a_1, a_n}\right) ::= \left[dif\left(r_i\right) < 0\right]\Big(\text{ÓÑÒ}\big(f_i = 0\big) \vee \text{ÓÑÒ}\big(f_i = 1\big)\Big), \qquad (18)$$

который включает в себя проверку условия отрицательного результата r_i вычитания (4) и соответствующую установку элемента f_i (6).

Для пороговой обработки (14) необходимо ввести составной оператор для реализации пороговой функции активации вида:

$$\text{ÀÊÒÈÂ}\left(\theta, S\right) ::= \left[dif\left(r\right) < 0\right]\Big(\text{ÓÑÒ}\big(Y = 1\big) \vee \text{ÓÑÒ}\big(Y = 0\big)\Big), \qquad (19)$$

который включает в себя проверку условия отрицательного результата r вычитания (13) и соответствующую установку выходного сигнала Y (14).

Поскольку формирование разностного среза РС A_j в каждом j-м цикле мультисуммирования является базовой операцией, то запишем эту операцию с помощью составных и базовых операторов следующим образом:

$$\text{ÑÐÅÇ}\left(\overline{\grave{a}_1,\grave{a}_n}\right)::=\text{Í ÐÓ}\times\text{Ì ÈÍ}_2\left(\overline{\grave{a}_1,\grave{a}_n}\right)\times\left(\text{ÂÛ}\times\left(M,\min\right)+\text{Ì ÀÑÊÀ}\left(\overline{a_1,a_n}\right)\right) \qquad (20)$$

с учётом одного оператора последовательного действия (16) и двух операторов параллельного действия - $\text{ÂÛ}\times\left(M,\min\right)$ и (18), которые выполняются одновременно.

Алгоритм сортировки элементов исходного РС A_0 в процессе мультисуммирования, который формирует дополнительный результат (отсортированный массив) по отношению к основному результату (сумме элементов массива), можно представить следующим образом:

$$\begin{aligned}&\text{ÑÎ ÐÒ}^n\left(\overline{a_1,a_n}\right)::=[\gamma]\Big\{\text{Í ÐÓ}\times\text{Ì ÈÍ}_2\left(\overline{a_1,a_n}\right)\times\text{Í ÐÓ}\times\left(\text{ÂÛ}\times\left(M,\min\right)+\right.\\&\left.+\text{Ì ÀÑÊÀ}\left(\overline{a_1,a_n}\right)+\text{Í ÀÊÎ Ï}_2\left(\overline{a_1,a_n}\right)\right)\times\text{ÂÛÂ}\left(S\right)\Big\}\times\text{ÔÈÍ}\,.\end{aligned} \qquad (21)$$

В записи (21) используется оператор $\text{ÂÛÂ}\left(S\right)$ для вывода результата операции накопления в каждом $j-$м цикле обработки. Выходом из процесса сортировки является выполнение условия (7), представленное базовым предикатом γ.

Базовый алгоритм формирования суммы элементов исходного РС A_0 можно представить в виде такой записи:

$$\begin{aligned}&\text{ÑÓÌ Ì À}^n\left(\overline{a_1,a_n}\right)::=[\gamma]\Big\{\text{Í ÐÓ}\times\text{Ì ÈÍ}_2\left(\overline{a_1,a_n}\right)\times\left(\text{ÂÛ}\times\left(M,\min\right)+\right.\\&\left.+\text{Ì ÀÑÊÀ}\left(\overline{a_1,a_n}\right)\right)\times\text{Í ÐÓ}\times\text{Í ÀÊÎ Ï}_2\left(\overline{a_1,a_n}\right)\times\text{ÑÓÌ}_2\left(R,S\right)\Big\}\times\text{ÂÛÂ}\left(R\right)\times\text{ÔÈÍ}\,,\end{aligned} \qquad (22)$$

которую можно сократить, воспользовавшись операцией $\text{ÑÐÅÇ}\left(\overline{a_1,a_n}\right)$ вида (20):

$$\begin{aligned}&\text{ÑÓÌ Ì À}^n\left(\overline{a_1,a_n}\right)::=[\gamma]\Big\{\text{ÑÐÅÇ}\left(\overline{a_1,a_n}\right)\times\text{Í ÐÓ}\times\text{Í ÀÊÎ Ï}_2\left(\overline{a_1,a_n}\right)\times\text{ÑÓÌ}_2\left(R,S\right)\Big\}\times\\&\times\text{ÂÛÂ}\left(R\right)\times\text{ÔÈÍ}\,.\end{aligned} \qquad (23)$$

Это первый вариант алгоритма суммирования, в котором постепенно за N циклов накапливаются текущие частичные суммы S_j вида (5).

Второй вариант для алгоритма суммирования можно представить в следующем виде:

$$\begin{aligned}&\text{ÑÓÌ Ì À}^n\left(\overline{a_1,a_n}\right)::=[\gamma]\Big\{\text{Í ÐÓ}\times\left(\text{Ì ÈÍ}_2\left(\overline{a_1,a_n}\right)+\text{Í ÀÊÎ Ï}_2\left(\overline{a_1,a_n}\right)\right)\times\\&\times\left(\text{ÂÛ×}\left(M,\min\right)+\text{Ì ÀÑÊÀ}\left(\overline{a_1,a_n}\right)+\text{ÑÓÌ}_2\left(R,S\right)\right)\Big\}\times\text{ÂÛÂ}\left(R\right)\times\text{ÔÈÍ}.\end{aligned}\tag{24}$$

Для второго варианта характерным является наличие в первом цикле двух «пустых» операций: $\text{Í ÀÊÎ Ï}_2\left(\overline{a_1,a_n}\right)$ и $\text{ÑÓÌ}_2\left(R,S\right)$, поскольку они выполняются в данном случае с опережением.

Для третьего варианта алгоритма суммирования характерна запись вида:

$$\begin{aligned}&\text{ÑÓÌ Ì À}^n\left(\overline{a_1,a_n}\right)::=\text{ÑÐÅÇ}\left(\overline{a_1,a_n}\right)\times[\gamma]\Big\{\text{Í ÐÓ}\times\text{Í ÀÊÎ Ï}_2\left(\overline{a_1,a_n}\right)\times\\&\times\text{ÑÓÌ}_2\left(R,S\right)\times\text{ÑÐÅÇ}\left(\overline{a_1,a_n}\right)\Big\}\times\text{ÂÛÂ}\left(R\right)\times\text{ÔÈÍ},\end{aligned}\tag{25}$$

в которой операция формирования первого РС выведена за пределы цикла.

Четвёртый вариант записи алгоритма суммирования имеет следующий вид:

$$\begin{aligned}&\text{ÑÓÌ Ì À}^n\left(\overline{a_1,a_n}\right)::=\text{Í ÐÓ}\times\text{Ì ÈÍ}_2\left(\overline{a_1,a_n}\right)\times[\gamma]\Big\{\Big(\text{ÂÛ×}\left(M,\min\right)+\\&+\text{Ì ÀÑÊÀ}\left(\overline{a_1,a_n}\right)+\text{ÑÓÌ}_2\left(R,S\right)\Big)\times\text{Í ÐÓ}\times\Big(\text{Ì ÈÍ}_2\left(\overline{a_1,a_n}\right)+\\&+\text{Í ÀÊÎ Ï}_2\left(\overline{a_1,a_n}\right)\Big)\Big\}\times\text{ÂÛÂ}\left(R\right)\times\text{ÔÈÍ},\end{aligned}\tag{26}$$

где за пределы цикла выведена операция определения минимального элемента в исходном РС A_0.

Таким образом, все четыре варианта записи в терминах САА В.М.Глушкова алгоритма суммирования с использованием разностных срезов отличаются только последовательностью применения введённых базисных и составных операторов и предикатов в пределах и за пределами цикла обработки.

Для пороговой обработки по РС первый вариант соответствующего алгоритма имеет вид:

$$\text{ПОРОГ}^{n}\left(\overline{a_1,a_n}\right)::=\text{СУММ}^{n}\left(\overline{a_1,a_n}\right)\times\text{ВЫЧ}_2\left(\theta,R\right)\times\text{АКТИВ}\left(\theta,R\right)\times\text{ВЫВ}\left(Y\right)\times\text{ФИН} \tag{27}$$

или при детальном описании с использованием записи (23)

$$\begin{aligned}&\text{ПОРОГ}^{n}\left(\overline{a_1,a_n}\right)::=[\gamma]\Big\{\text{СРЕЗ}\left(\overline{a_1,a_n}\right)\times\text{НРУ}\times\text{НАКОП}_2\left(\overline{a_1,a_n}\right)\times\\&\times\text{СУМ}_2\left(R,S\right)\Big\}\times\text{ВЫЧ}_2\left(\theta,R\right)\times\text{АКТИВ}\left(\theta,R\right)\times\text{ВЫВ}\left(Y\right)\times\text{ФИН}\ .\end{aligned} \tag{28}$$

Поскольку при обработке по РС возможно совмещённое выполнение некоторых операций пороговой обработки, то ниже приведены ещё три возможных варианта представления алгоритма пороговой обработки.

Второй вариант:

$$\begin{aligned}&\text{ПОРОГ}^{n}\left(\overline{a_1,a_n}\right)::=[\gamma]\Big\{\text{НРУ}\times\Big(\text{МИН}_2\left(\overline{a_1,a_n}\right)+\text{НАКОП}_2\left(\overline{a_1,a_n}\right)\Big)\times\\&\times\Big(\text{ВЫЧ}\left(M,\min\right)+\text{МАСКА}\left(\overline{a_1,a_n}\right)+\text{СУМ}_2\left(R,S\right)+\text{ВЫЧ}_2\left(\theta,R\right)\Big)\times\\&\times\text{АКТИВ}\left(\theta,R\right)\Big\}\times\text{ВЫВ}\left(Y\right)\times\text{ФИН}\ ,\end{aligned} \tag{29}$$

третий вариант:

$$\begin{aligned}&\text{ПОРОГ}^{n}\left(\overline{a_1,a_n}\right)::=\text{СРЕЗ}\left(\overline{a_1,a_n}\right)\times[\gamma]\Big\{\text{НРУ}\times\text{НАКОП}_2\left(\overline{a_1,a_n}\right)\times\\&\times\left(\text{СУМ}_2\left(R,S\right)+\text{ВЫЧ}_2\left(\theta,S\right)\right)\times\Big(\text{СРЕЗ}\left(\overline{a_1,a_n}\right)+\text{АКТИВ}\left(\theta,S\right)\Big)\Big\}\times\\&\times\text{ВЫВ}\left(Y\right)\times\text{ФИН}\ ,\end{aligned} \tag{30}$$

четвёртый вариант:

$$\begin{aligned}&\text{ПОРОГ}^{n}\left(\overline{a_1,a_n}\right)::=\text{НРУ}\times\text{МИН}_2\left(\overline{a_1,a_n}\right)\times[\gamma]\Big\{\Big(\text{ВЫЧ}\left(M,\min\right)+\\&+\text{МАСКА}\left(\overline{a_1,a_n}\right)+\text{ВЫЧ}_2\left(\theta,S\right)+\text{СУМ}_2\left(R,S\right)\Big)\times\text{НРУ}\times\Big(\text{МИН}_2\left(\overline{a_1,a_n}\right)+\\&+\text{НАКОП}_2\left(\overline{a_1,a_n}\right)+\text{АКТИВ}\left(\theta,R\right)\Big)\Big\}\times\text{ВЫВ}\left(Y\right)\times\text{ФИН}\ .\end{aligned} \tag{31}$$

Анализ представлений алгоритма пороговой обработки (27) – (31) свидетельствует о том, что в них задействованы все ранее рассмотренные варианты (22) – (26) алгоритма суммирования как основной составляющей мультиобработки по РС.

Выводы

1. Функциональная мощность базиса САА В.М. Глушкова позволяет на его основе, дополнив его базисными и составными операторами и предикатами, учитывающими

специфику обработки данных, представить в терминах регулярных схем любой алгоритм мультиобработки и, в частности, с использованием разностных срезов.

2. Всё многообразие реализаций любого алгоритма мультиобработки в модифицированном базисе САА В.М. Глушкова можно представить в компактной и наглядной форме, что не только упрощает процесс анализа таких представлений, но и способствует их дальнейшему развитию (оптимизации и модернизации).

Литература

1. Цейтлин Г.Е. Структурное программирование задач символьной мультиобработки // Кибернетика. – 1983. – № 5. – С. 22-30.
2. Цейтлин Г.Е. Проектирование последовательных алгоритмов сортировки: классификация, трансформация, синтез // Программирование. – 1989. – № 3. – С. 3-24.
3. Цейтлин Г.Е. Распараллеливание алгоритмов сортировки // Кибернетика. – 1989. – № 6. – С. 67-74.
4. Кожемяко В.П., Мартынюк Т.Б., Хомюк В.В. Особенности структурного программирования синхронных алгоритмов сортировки // Кибернетика и системный анализ. – 2006. – № 5. – С. 122 – 133.
5. Яценко Е.А. Регулярные схемы алгоритмов адресной сортировки и поиска // Управляющие системы и машины. – 2004. – № 5. – С. 61 – 66.
6. Цейтлин Г.Е., Амонс А.А., Головин О.В., Зубцов А.Ю. Интегрированный инструментарий проектирования и синтеза классов алгоритмов и программ // Кибернетика и системный анализ. – 2000. – № 3. – С. 165 – 169.
7. Цейтлин Г.Е. Алгебры Глушкова и теория клонов // Кибернетика и системный анализ. – 2003. – № 4. – С. 48 – 58.
8. Борисов Е.С. Полуавтоматическая система декомпозиции последовательных программ для параллельных вычислителей с распределённой памятью // Кибернетика и системный анализ. – 2004. – № 3. – С. 139 – 150.

9. Цейтлин Г.Е., Иванов Е.А. Специализированные информационные технологии для лиц с физическими ограничениями // Управляющие системы и машины. – 2008. – № 5. – С. 62 – 69, 74.
10. Мартинюк Т.Б., Хом'юк В.В. Методи та засоби паралельних перетворень векторних масивів даних. Монографія. – Вінниця: УНІВЕРСУМ – Вінниця, 2005. – 203 с. – ISBN 966-641-114-8.
11. Ни Л.М., Джейн А.К. Двухуровневый конвейерный систолический массив для кластерного анализа / В кн. СБИС для распознавания образов и обработки изображений: Пер. с англ./Под ред. К. Фу.– М.: Мир, 1988. – 248с.
12. Справочник по цифровой вычислительной технике / Под ред. В.Н. Малиновского. – К.: Техніка, 1980. – 320 с.
13. Мартынюк Т.Б. Модель порогового нейрона на основе параллельной обработки по разностным срезам // Кибернетика и системный анализ. – 2005. – № 4. – С. 78 – 89.
14. Мартинюк Т.Б. Рекурсивні алгоритми багатооперандної обробки інформації. Монографія. – Вінниця: УНІВЕРСУМ – Вінниця, 2000. – 216 с. – ISBN 966-7199-98-3.
15. Васюра А.С., Мартинюк Т.Б., Куперштейн Л.М. Методі та засоби нейроподібної обробки даних для систем керування. Монографія. – Вінниця: УНІВЕРСУМ – Вінниця, 2008. – 175 с. – ISBN 978-966-641-279-2.

1) «В печать»;
2) «На перевод на английский язык согласен».