Системный анализ

УДК 681.3.06

В. П. Кожемяко, Т. Б. Мартынюк, В. В. Хомюк

ОСОБЕННОСТИ СТРУКТУРНОГО ПРОГРАММИРОВАНИЯ СИНХРОННЫХ АЛГОРИТМОВ СОРТИРОВКИ

Ключевые слова: синхронные алгоритмы сортировки, сортировка методом парного обмена, система алгебраических алгебр.

Key words: synchronous sorting algorism, sorting of matching exchange method, systems of algebraic algebras.

Введение

В последнее время резко возрос интерес к методам и средствам сортировки больших массивов информации, что не в последнюю очередь связано как с новыми достижениями в области компьютерных технологий, так и с расширением сферы применимости самих алгоритмов сортировки [1-6]. Особенно привлекательной выглядит реализация методов сортировки на таких перспективных структурах, как ассоциативные процессоры [7], нейронные сети [4], процессоры на базе клеточных автоматов [8]. Несмотря на обилие алгоритмов и усовершенствований известных методов сортировки, основными среди них считаются семь методов: линейный выбор, линейный выбор с обменом, линейный выбор с подсчетом, парный обмен, стандартный обмен, просеивание и линейная вставка [9], а все остальные рассматриваются как производные от них [10]. Детальный анализ перечисленных методов позволяет разделить их на две группы с учетом эффективности реализации аппаратно или программно, а именно, на параллельные (синхронные) и последовательные методы сортировки соответственно [9]. Среди параллельных методов сортировки метод парного обмена считается одним из тех, которые обеспечивают максимально возможный параллелизм обработки данных [9,11]. В то же время сложность классификации алгоритмов сортировки связана с отсутствием формализованного способа их описания. Среди наиболее распространенных способов представления процесса сортировки можно отметить описание в виде программы [9,10] и в виде сортирующей сети [11]. В этой связи заслуживает внимание использование абстрактных моделей мультипроцессоров [12], а именно, дальнейшее развитие операторных представлений в системах алгоритмических алгебр (САА) В.М. Глушкова [13,14]. В ряде работ [13,15,16] в базисе регулярных (структурированных) схем (РС) в рамках теории САА не только подробно проанализированы наиболее распространенные последовательные алгоритмы сортировки, но и исследованы способы их распараллеливания. В то же время отсутствуют публикации, где этот подход был бы применен к известным параллельным алгоритмам сортировки. В данной работе приводятся результаты использования данной методики формализованного описания для конкретного параллельного метода, а именно, для сортировки методом парного обмена.

Аппарат модифицированных САА

Математический аппарат САА имеет достаточную универсальность, поскольку может быть использован для описания как последовательных алгоритмов сортировки, так и параллельных (синхронных) [15,16]. В последнем случае необходимо ввести некоторые коррективы в базисе САА.

В работах [15,16] определено понятие базиса САА вида

$$\sum \; = \widetilde{A} \cup \widetilde{X}\,, \tag{1}$$

где $\widetilde{A} = \{ A_i \mid i \in I \}$ - набор элементарных операторов, а $\widetilde{X} = \{ \alpha_j \mid j \in J \}$ - набор элементарных условий. Тогда регулярная схема (РС) оператора A определяется как суперпозиция вида *A=F(* $\sum$ *)*, которая представляет составной оператор A с привлечением базисных операторов и условий, входящих в $\sum$, а также используя операции, определяемые сигнатурой САА. Так для представления составных логических условий применяются логические операции: дизъюнкция $\alpha \vee \beta$, конъюнкция $\alpha \wedge \beta$ и отрицание $\overline{\alpha}$, а для составных операторов – операции, соответствующие известным конструкциям структурного программирования, а именно [15]: а) композиция $(A \times B)$ - последовательное применение операторов A и B; б) альтернатива $\underset{\alpha}{(A \vee B)}$ или $[\alpha](A \vee B)$ - если α, то A, иначе B; в) фильтр $\underline{\alpha}$ - если α, то E, иначе N, где E - тождественный оператор, N - неопределенный, т.е. обрывающий вычисления оператор; г) цикл $\underset{\alpha}{\{A\}}$ или $[\alpha]\{A\}$ - выполнять A пока α ложно, при α истинном – конец цикла.

В отличие от введенного [15,16] определения размеченного массива в дальнейшем (с учетом синхронной обработки) целесообразно использовать понятие сортируемого массива вида $M ::= D = a_1\ a_2 \ldots a_m$, где D- множество элементов сортируемого массива с определенным на нем отношением порядка $\leq$, т.е. отказаться от использования множества разделителей (указателей V и маркеров W), которые отмечают определенные позиции между элементами в процессе сортировки. При этом сохраняется определение расстояния между элементами a_i и a_j массива *M* как количества элементов массива *M*, расположенных между элементами a_i и a_j, которое равно величине $k = (i - j - 1)$. На сортирующих массивах определены логические условия (предикаты) и операторы, которые составляют базис (1) РС–алгоритмов сортировки. С учетом описания синхронных алгоритмов в дальнейшем будут задействованы следующие предикаты [15]: а) $l \underset{k}{\leq} s, l \underset{k}{>} s$ - истинные при выполнении указанных отношений для элементов l и s данного массива *M*, расположенных на расстоянии k друг от друга; б) $l \leq r, l > r$ - истинные при выполнении указанных отношений для соседних элементов массива *M*, т.е. при $k = 0$; в) $\pi(k)$ - истинное, если для любой пары элементов a и a' массива *M*, расположенных на расстоянии k друг от друга, выполняется отношение $a \leq a'$; г) $\pi = \pi(0)$ - истинное, если выполняется отношение $a \leq a'$ для любых соседних элементов a и a' массива *M*. При этом, если условия $\pi(k)$ и π истинны для всего множества *M* или некоторого его фрагмента, то можно говорить, что данный массив *M* или его фрагмент отсортирован по отношению $\underset{k}{\leq}$ или $\leq$. Среди базисных операторов практический интерес представляют следующие [15]: а) $ТРАНСП_k(l, s)$ - оператор перестановки (транспозиции) элементов l и s массива *M*, расположенных на расстоянии k друг от друга; б) $ТРАНСП(l, r)$ - оператор перестановки соседних элементов l и r массива *M*; в) $УСТ(z)$ - оператор установки последовательности z указателей и маркеров, где $z \in (V \cup W)$.

Для синхронных алгоритмов сортировки большую роль играет понятие зоны предиката или оператора, под которой подразумевается фрагмент массива, на котором осуществляется проверка истинности предиката или выполнение оператора соответственно. Так, истинность предиката π определяется наличием *n*-мерного единичного набора логических значений, *i*-ая компонента которого соответствует выполнимости соотношения

$$s_i \leq s_{i+1} \tag{2}$$

для любого $i = \overline{1, m}$, где $M = s_1\ s_2\ ...\ s_m$ - отсортированный массив.

Таким образом, массив *M* или его фрагмент отсортирован, если на соответствующей зоне соотношение (2) выполняется синхронно для всех пар соседних элементов [16].

Кроме перечисленных базисных операторов для синхронной сортировки основным является составной оператор вида

$$СОРТ_p\left(\overset{\frown}{v_1, v_2}\right) ::= \left[l \underset{k}{\leq} s\right]\left(E \vee ТРАНСП_k(l, s)\right), \tag{3}$$

который осуществляет сортировку в зоне $\overset{\frown}{v_1, v_2}$ размера p. При $p = 2$ оператор (3) можно представить так

$$СОРТ_2\left(\overset{\frown}{v_1, v_2}\right) ::= [l \leq r]\left(E \vee ТРАНСП(l, r)\right). \tag{4}$$

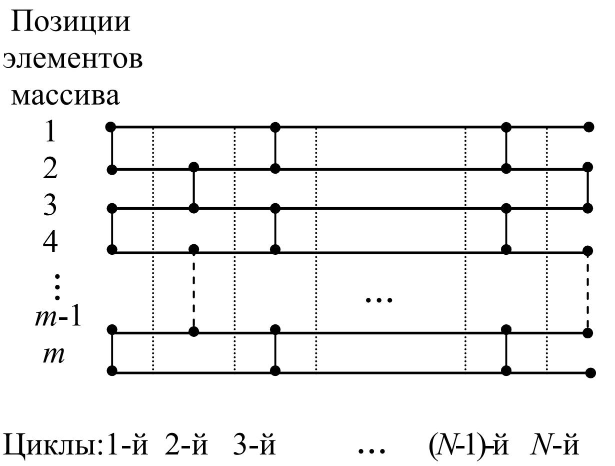
Рассмотренный математический аппарат структурированных схем программ представляет собой базу для формализованного описания синхронных алгоритмов сортировки и, в частности, алгоритмов сортировки методом парного обмена.

Синхронная сортировка методом парного обмена и ее интерпретация

В основе метода парного обмена лежит последовательность парных просмотров, в которых элемент сравнивается со своим ближайшим соседом, а возможными перестановками (транспозициями) являются перемещения больших по значению элементов на одну позицию вниз [9]. Другими словами, в нечетных просмотрах каждый элемент нечетной $(2k-1)$-й позиции сравнивается с соседним элементом четной $2k$-й позиции и больший из них занимает четную $2k$-ю позицию, а меньший – нечетную $(2k-1)$-ю позицию, где $k=\overline{1,K}$. В четных просмотрах каждый элемент четной $2k$-й позиции сравнивается с соседним элементом нечетной $(2k+1)$-й позиции и больший из них занимает нечетную $(2k+1)$-ю позицию, а меньший – четную $2k$-ю позицию. Сортировка заканчивается, когда при двух соседних просмотрах не будет перемещений ни в одной паре элементов.

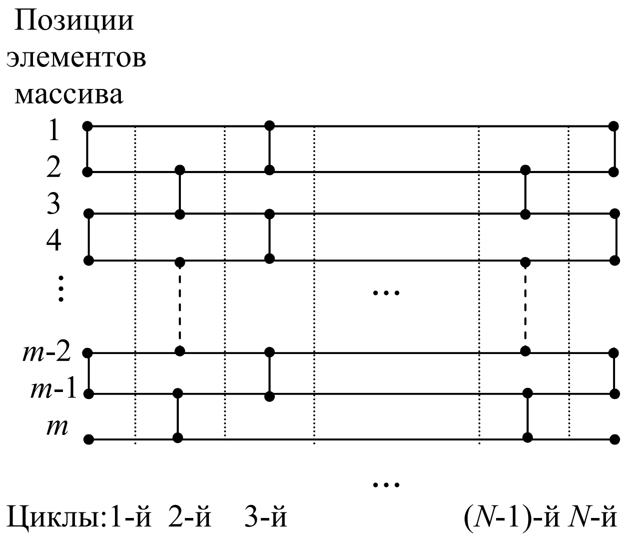
На рис. 1 а,б проиллюстрированы два варианта формирования пар элементов для сравнения в виде сортирующей сети [11] соответственно при четном и нечетном количестве m элементов массива M.

Размерность массива чисел: m-четное число



а)

Размерность массива чисел: m-нечетное число



б)

Рис. 1. Сортирующая сеть с учетом размерности массива чисел

На рис. 2 а, б приведены примеры сортировки методом парного обмена с указанием соотношения между количеством циклов (просмотров) N обработки и мощностью m массива M. Здесь используются следующие обозначения:

] – признак пары сравниваемых элементов;

⟶ - признак транспозиции элементов в соответствующей паре.

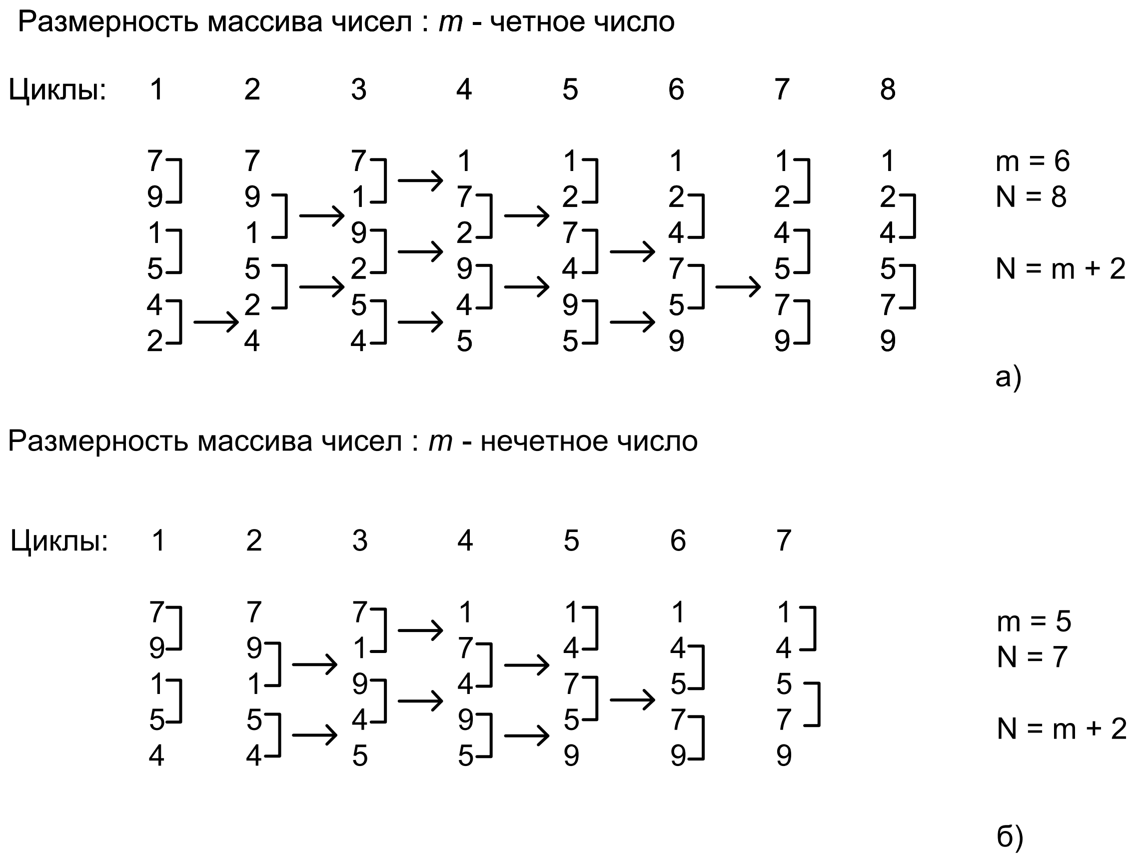


Рис. 2. Примеры сортировки методом парного обмена

Таким образом, в каждом цикле формируется и параллельно обрабатывается K пар элементов, где $K = \left[m/2 \right]$, а минимальное и максимальное количество циклов соответственно равно [9]:

$$N_{\min} = 2, \quad N_{\max} = m + 2 . \tag{5}$$

Для формализованного описания данного синхронного алгоритма сортировки необходимо добавить несколько базисных предикатов и ввести некоторые операторы. Эта необходимость диктуется тем, что в процессе обработки выполняется не последовательная, а многослойная сортировка массива *M*, который в данном случае разбивается на M_i подмассивы $R(M)=\left\{M_i \,\middle|\, i=\overline{1,K}\right\}$, которые подвергаются параллельному упорядочению с привлечением процедур $СОРТ$ вида (3), (4) [13]. При этом особенность синхронной сортировки состоит в том, что обработка не требует применения процедуры слияния полученных отсортированных подмассивов M_i, но требует введения вместо базисного оператора $УСТ(z)$ базисного оператора $УСТ\left(\overline{v_1, v_2}\right)$ установки подмассивов M_i с зоной $\overline{v_1, v_2}$ размером p. Предполагается в дальнейшем рассматривать зону операторов $УСТ\left(\overline{v_1, v_2}\right)$ и $СОРТ$ (4) как подмассив M_i, ограниченный элементами с индексами v_1, v_2 соответственно, т.е. использовать сортируемую зону $\left(a_{v_1}, a_{v_2}\right)$ постоянной размерности $p=2$ (по определению метода сортировки). Кроме того, для законченности формализованной записи процесса сортировки необходимо использование оператора $ФИН$ завершения работы [15].

Одновременно множество базисных условий необходимо расширить за счет введения таких отношений: а) π^K - истинное, если выполняется базисное условие π параллельно для всех K пар соседних элементов массива *M* или параллельно для всех подмассивов M_i; б) $\pi_Ч^K, \pi_Н^K$ - истинные, если выполняется предикат π^K соответственно в четном и нечетном циклах сортировки.

В результате синхронный алгоритм сортировки методом парного обмена можно представить следующим образом:

$$\begin{gathered}ОБМЕН ::= \left[\pi_Н^K \wedge \pi_Ч^K\right] \left\{\left(УСТ\left(\overline{2k-1, 2k}\right) \times СОРТ_2^K\left(\overline{2k-1, 2k}\right)\right) \times \right. \\ \left. \times \left(УСТ\left(\overline{2k, 2k+1}\right) \times СОРТ_2^K\left(\overline{2k, 2k+1}\right)\right)\right\} \times ФИН, \quad k=\overline{1,K}\end{gathered} \tag{6}$$

где $COPT_2^K\left(\overline{v_1, v_2}\right)$ - оператор вида (4), который выполняется параллельно для K пар элементов массива *М*.

Таким образом, алгоритм содержит один составной цикл, который, в свою очередь, состоит из двух циклов: первого (нечетного) и второго (четного). Завершение составного цикла происходит только при выполнении сложного предиката в виде конъюнкции $\left(\pi_Н^K \wedge \pi_Ч^K\right)$, т.е. при условии выполнения предиката π^K в двух соседних циклах.

Статистические исследования результатов моделирования данного алгоритма [17] показали, что после обязательной проверки выполнения предиката π^K в первых двух циклах в дальнейшем можно ограничиться проверкой истинности предиката π^K только в очередном цикле сортировки. Объяснить такую возможность можно следующим образом.

Пусть n –й цикл был последним, в котором выполнялась транспозиция (обмен) в определенных парах элементов массива *М*. В соответствии с классическим методом парного обмена [9] необходимо выполнить еще два проверочных цикла, чтобы подтвердить выполнение условия (2), т.е. упорядочение массива *М*. Однако, если в $(n+1)$- м цикле транспозиции в парах отсутствуют, т.е. предикат π^K истинный, то можно утверждать, что в $(n+2)$- м цикле транспозиций в парах также не будет, поскольку будут сравниваться элементы в парах, которые уже были упорядочены (отсортированы) в n – м цикле. Таким образом, признаком того, что массив *М* является отсортированным, можно считать наличие хотя бы одного (кроме первого), а не двух соседних циклов, в которых отсутствуют транспозиции во всех парах элементов массива *М*.

В результате максимальное количество циклов $N_{\max}$ уменьшается как минимум на один цикл, т.е.

$$N_{\max} = m + 1, \tag{7}$$

а сам алгоритм можно записать следующим образом:

$$\begin{aligned}&ОБМЕН ::= \left(УСТ\left(\overline{2k-1,\,2k}\right)\times СОРТ_2^K\left(\overline{2k-1,\,2k}\right)\right)\times\\&\times\left[\pi_Н^K \vee \pi_Ч^K\right]\left\{\left(УСТ\left(\overline{2k,\,2k+1}\right)\times СОРТ_2^K\left(\overline{2k,\,2k+1}\right)\right)\times\right.\\&\left.\times\left(УСТ\left(\overline{2k-1,\,2k}\right)\times СОРТ_2^K\left(\overline{2k-1,\,2k}\right)\right)\ \right\}\times ФИН,\quad k=\overline{1,\,K}\end{aligned} \tag{8}$$

т.е. завершение второго составного цикла происходит при выполнении сложного предиката в виде дизъюнкции $\left(\pi_Н^K \vee \pi_Ч^K\right)$, т.е. при условии истинности предиката π^K хотя бы в одном цикле после первого цикла сортировки.

Трансформация синхронной сортировки методом парного обмена

Из рис. 1 видно, что при сортировке данным методом не все элементы задействованы в парах, а именно, при четном m это элементы первой и m – й позиций в четных циклах, а при нечетном m это элемент m – й позиции в нечетных циклах и элемент первой позиции в четных циклах. В этой связи представляет интерес трансформация данного алгоритма, которая заключается в формировании дополнительной пары элементов, что позволяет, по крайней мере, увеличить не только уровень параллелизма сортировки, но и однородность структурной организации вычислительных средств при реализации алгоритма. При этом для формирования K - й пары, где $K = m/2$, для случая, когда m – нечетное число, необходимо ввести нулевой элемент с нулевой позицией в массив *M*. В результате модель сортирующей сети примет вид “кольца” (рис. 3 а, б) в отличие от модели сортирующей сети в виде “ленты” (рис. 1 а, б).

На рис. 4 а,б приведены примеры модифицированного процесса сортировки методом парного обмена с указанием соотношения $N = f(m)$ и учетом одного проверочного цикла. Здесь дополнительно используются следующие обозначения:

[- признак дополнительной пары крайних элементов;

[$\rightrightarrows$ - признак транспозиции элементов в дополнительной паре.

В результате максимальное количество циклов составит

$$N_{\max} = m + 1\,. \tag{9}$$

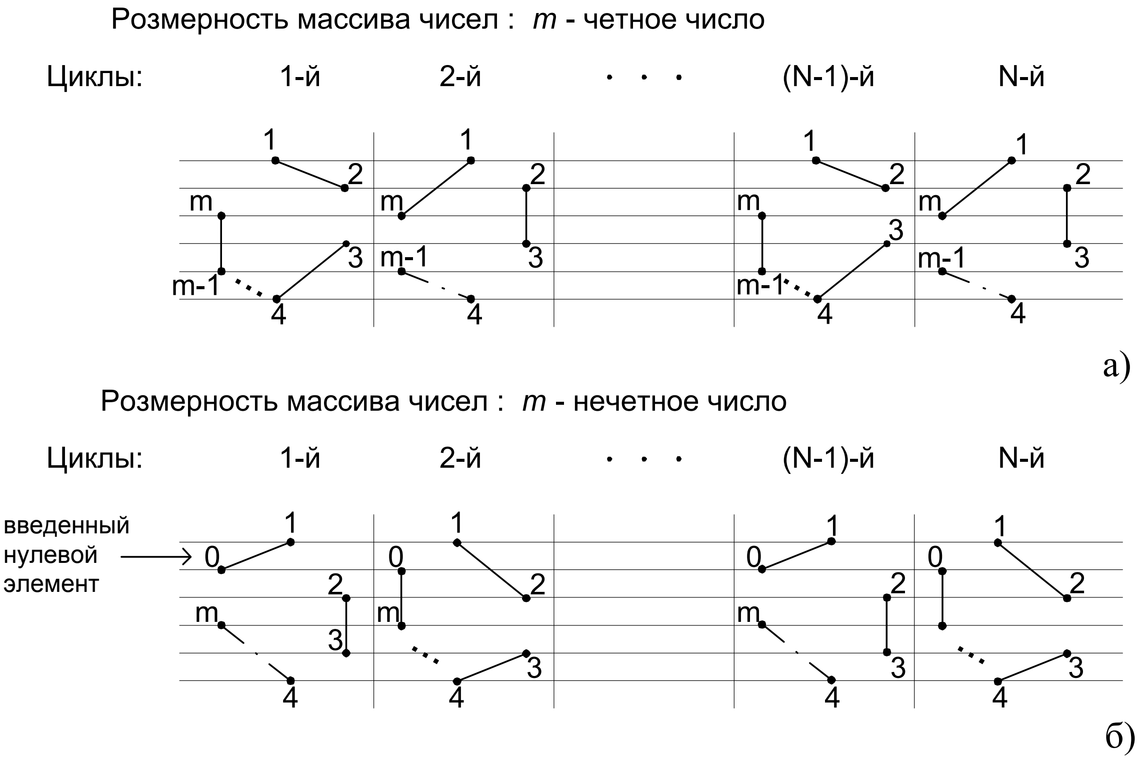


Рис. 3. Модель сортирующей сети типа “кольцо”

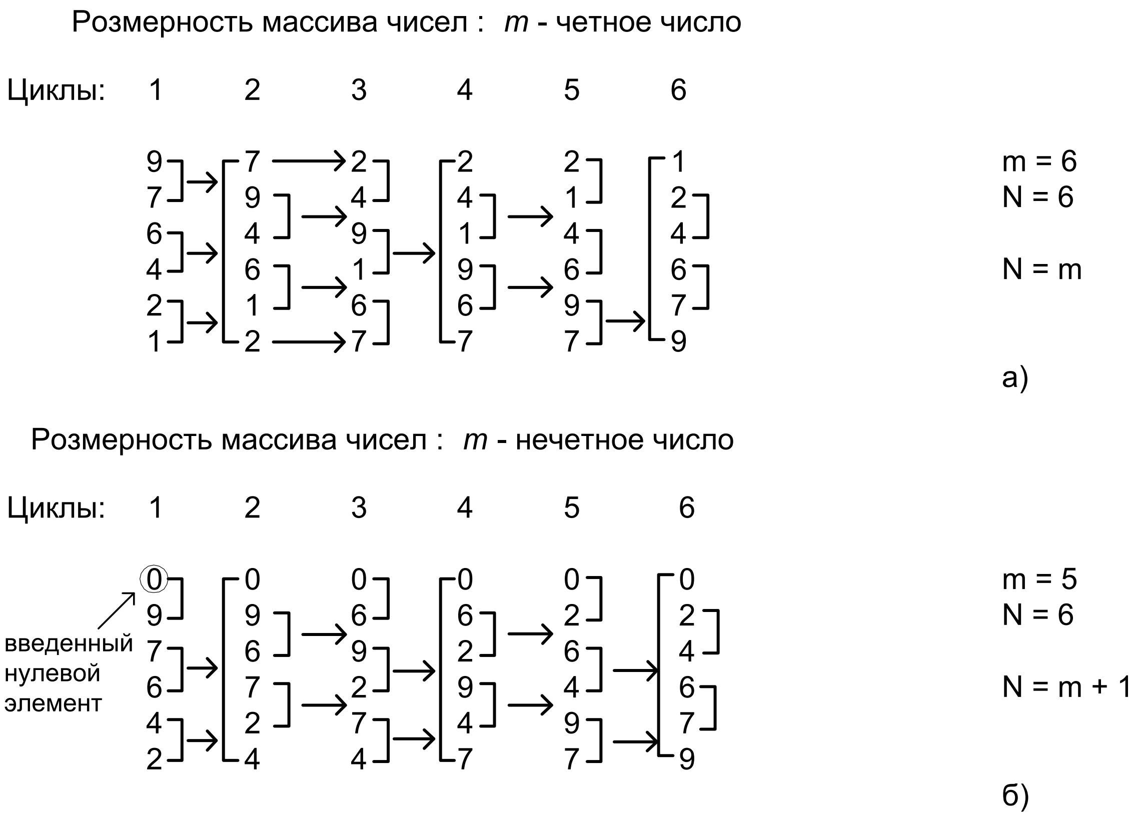


Рис. 4. Примеры модифицированного процесса сортировки

Формализованное описание данного алгоритма сортировки требует введения дополнительных предикатов вида: а) ρ - истинное, если для дополнительной K- й пары элементов массива M выполняется соотношение $a_1 \le a_m$ или $a_0 \le a_m$; б) $\mu = \mu(m)$ - истинное при выполнении отношения m – четное число; в) $\eta = \eta(m)$ - истинное при выполнении отношения m – нечетное число; а также соответствующих базисных операторов: а) $ВВОД(0)$ - оператор введения нулевого элемента с нулевым индексом; б) $КОЛЬЦО\left(\overline{v_1, v_2}\right)$ - оператор формирования K- й пары зоны $\overline{v_1, v_2}$ размером $p = 2$, а именно, пары элементов массива M с индексами $(1, m)$ или $(0, m)$.

Кроме того, в этом случае возникает необходимость введения дополнительной операции, а именно, композиции $A + B$, которая определяет параллельное применение операторов A и B. Для замыкания крайних элементов массива M необходимо введение соответствующих составных операторов:

$$КОЛЬЦО^{\mu}\left(\overline{1, m}\right) ::= УСТ\left(\overline{2k, 2k+1}; \overline{1, m}\right) \times \left(СОРТ_2^K\left(\overline{2k, 2k+1}\right) + СОРТ_2\left(\overline{1, m}\right)\right), \quad (10)$$

$$КОЛЬЦО^{\eta}\left(\overline{0, m}\right) ::= УСТ\left(\overline{2k-1, 2k}; \overline{0, m}\right) \times \left(СОРТ_2^K\left(\overline{2k-1, 2k}\right) + СОРТ_2\left(\overline{0, m}\right)\right), \quad (11)$$

где сложный оператор $УСТ\left(\overline{v_1, v_2}, \overline{v_1^K, v_2^K}\right)$ формирует все K пары элементов массива M или все K подмассивы M_i. По аналогии с операторами вида (10), (11) имеет смысл ввести обозначение составного оператора вида:

$$ЛЕНТА\left(\overline{v_1, v_2}\right) ::= УСТ\left(\overline{v_1, v_2}\right) \times СОРТ_2^K\left(\overline{v_1, v_2}\right), \quad (12)$$

что позволит представить соответственно два варианта описания предыдущего алгоритма вида (6) и (8) в более компактной форме:

$$ОБМЕН ::= \left[\pi_Н^K \wedge \pi_Ч^K\right]\left\{ЛЕНТА\left(\overline{2k-1, 2k}\right) \times ЛЕНТА\left(\overline{2k, 2k+1}\right)\right\} \times ФИН, \quad k = \overline{1, K} \quad (13)$$

$$ОБМЕН ::= ЛЕНТА\left(\overline{2k-1,\, 2k}\right)\times\left[\pi_Н^К \vee \pi_Ч^К\right]\left\{ЛЕНТА\left(\overline{2k,\, 2k+1}\right)\times\right.$$
$$\left.\times ЛЕНТА\left(\overline{2k-1,\, 2k}\right)\right\}\times ФИН,\quad k=\overline{1,\, K}. \tag{14}$$

В результате модифицированный алгоритм сортировки методом парного обмена можно записать в общем виде следующим образом:

$$ОБМЕН ::= [\mu]\left(ОБМЕН^{\mu} \vee ОБМЕН^{\eta}\right), \tag{15}$$

причем

$$ОБМЕН^{\mu} ::= \left(УСТ\left(\overline{2k-1,\, 2k}\right)\times СОРТ_2^К\left(\overline{2k-1,\, 2k}\right)\right)\times$$
$$\times\left[\rho\wedge\left(\pi_Н^К \vee \pi_Ч^К\right)\right]\left\{КОЛЬЦО^{\mu}\left(\overline{1,\, m}\right)\times\left(УСТ\left(\overline{2k-1,\, 2k}\right)\times\right.\right. \tag{16}$$
$$\left.\left.\times СОРТ_2^К\left(\overline{2k-1,\, 2k}\right)\right)\right\}\times ФИН,\quad k=\overline{1,\, K}$$

$$ОБМЕН^{\eta} ::= ВВОД(0)\times\left(УСТ\left(\overline{2k-2,\, 2k-1}\right)\times СОРТ_2^К\left(\overline{2k-2,\, 2k-1}\right)\right)\times$$
$$\times\left[\rho\wedge\left(\pi_Н^К \vee \pi_Ч^К\right)\right]\left\{КОЛЬЦО^{\eta}\left(\overline{0,\, m}\right)\times\left(УСТ\left(\overline{2k-2,\, 2k-1}\right)\times\right.\right. \tag{17}$$
$$\left.\left.\times СОРТ_2^К\left(\overline{2k-2,\, 2k-1}\right)\right)\right\}\times ФИН,\quad k=\overline{1,\, K}$$

Таким образом, в обоих случаях (для четного m (16) и нечетного m (17)) выполняется обязательно первый нечетный цикл, а затем составной цикл, который содержит выполнение соответственно оператора $КОЛЬЦО$ вида (10) или (11) и последовательности операторов нечетного цикла, входящих в состав составного оператора $ЛЕНТА$ вида (12). В результате запись операторов вида (16), (17) можно представить следующим образом:

$$ОБМЕН^{\mu} ::= ЛЕНТА\left(\overline{2k-1,\, 2k}\right)\times\left[\rho\wedge\left(\pi_Н^К \vee \pi_Ч^К\right)\right]\left\{КОЛЬЦО^{\mu}\left(\overline{1,\, m}\right)\times\right.$$
$$\left.\times ЛЕНТА\left(\overline{2k-1,\, 2k}\right)\right\}\times ФИН,\quad k=\overline{1,\, K} \tag{18}$$

$$ОБМЕН^{\eta} ::= ВВОД(0)\times ЛЕНТА\left(\overline{2k-2,\, 2k-1}\right)\times\left[\rho\wedge\left(\pi_Н^К \vee \pi_Ч^К\right)\right]\left\{КОЛЬЦО^{\eta}\left(\overline{0,\, m}\right)\times\right.$$
$$\left.\times ЛЕНТА\left(\overline{2k-2,\, 2k-1}\right)\right\}\times ФИН,\quad k=\overline{1,\, K} \tag{19}$$

Завершение составного цикла в обоих случаях (18) и (19) выполняется при истинности сложного предиката в виде конъюнкции, т.е. при условии выполнения обязательно предиката

ρ, а предиката π^{K} хотя бы в одном операторе *КОЛЬЦО* или *ЛЕНТА*, что входят в составной цикл.

Имитационное моделирование синхронной сортировки методом парного обмена

На сегодняшний день проведены многочисленные эксперименты по моделированию последовательных алгоритмов сортировки. В качестве иллюстрации можно привести следующие два примера.

Первый. Для сравнения предложено четыре алгоритма сортировки [18]. На вход подаются одинаковые массивы случайных чисел в диапазоне 0..1000. Объем массива – 50 000 чисел. Время выполнения приведено в табл. 1.

Таблица 1.

Алгоритм сортировки	Время выполнения алгоритма на персональных ЭВМ, сек.		Количество выполненных команд процессора
	Pentium – S 133	Pentium MXX 166	
Методом “пузырька”	72,5	64,8	$23{,}0*10^{9}$
Выбором	69,0	46,3	$21{,}9*10^{9}$
Шейкера	57,8	47,2	$17{,}1*10^{9}$
Вставками	28,5	22,3	$4{,}7*10^{9}$

Второй. Чтобы оценить время работы алгоритмов сортировки использовалась программа, которая формировала массив псевдослучайных равномерно распределенных на отрезке [0,100000] целых чисел, а затем сортировала этот массив каждым из пяти выбранных алгоритмов сортировки [19]. Были получены результаты для массивов с количеством элементов: 5000, 10000, 25000, 50000, 100000. Для каждой размерности было проведено по 5 экспериментов, в табл.2 представлены усредненные результаты, полученные в ходе их проведения.

Таблица 2.

Количество элементов	Алгоритм сортировки				
	Методом “пузырька”	Простых вставок	Бинарных вставок	Шелла	Бинарных деревьев
5000	570	242	154	54	1
10 000	2 352	990	570	208	1
25 000	17 160	9 062	4 710	1 194	1
50 000	77 234	44 940	22 322	5 978	98
100 000	321 169	190 062	102 228	24 648	208

Проанализировав полученные результаты, можно сделать такие выводы.

Во-первых, несложно заметить, что наилучшую скорость работы показал алгоритм сортировки бинарными деревьями для случая, когда количество элементов массива не превышало 50 000. Во-вторых, время работы первых четырех алгоритмов (“пузырька”, Шелла, простых и бинарных вставок) растет квадратично относительно роста размера упорядочиваемого массива, что подтверждается и теоретическими соображениями, так как известно, что скорость работы данных алгоритмов оценивается как $O(n^2)$ [9]. Таким образом, при разработке программ, требующих упорядочивания массивов большого объема, следует применять сортировку методом бинарных деревьев, как имеющую существенно лучшие временные показатели.

При этом следует обратить внимание на то, что исследовались конвейерные (последовательные) алгоритмы сортировки. Рассмотрим моделирование предложенного в данной работе параллельного алгоритма сортировки.

Для решения задачи имитационного моделирования синхронной сортировки методом парного обмена использовались векторные массивы данных с нормально распределенными элементами в диапазоне 0..1000. При этом объем массивов изменялся в диапазоне 64..1000 и для каждого случая генерировалось по сто массивов. Программная реализация проводилась на языке C++ в программной среде C++Builder 5. Выбор последнего обусловлен тем, что

C++Builder 5 позволяет работать с произвольными базами данных, создавать прикладные программы для работы с INTERNET и т. д. Результаты моделирования алгоритма синхронной сортировки методом парного обмена представлены в табл. 3 и на трехмерной диаграмме (рис. 5).

Таблица 3

Метод парного обмена

Размерность входного массива данных	**Стандартное отклонение**											
	120	110	100	90	80	70	60	50	40	30	20	10
64	59.64	60.37	59.64	59.82	58.91	62.37	60.35	62.55	59.28	59.45	60.18	60.19
128	121.45	121.81	121.82	123.27	123.09	122.1	119.46	123.45	120.9	119.1	123.46	120.1
192	182.72	184.36	180.55	183.26	180.37	184.91	184.55	183.82	182.73	182.18	181.09	182.55
256	246.91	243.27	244.91	242.18	244.37	242.19	241.81	247.64	240.9	244.18	239.82	246.11
320	305.09	306.37	308.55	307.27	312.73	305.82	310.73	307.09	306.26	308.55	307.21	309.09
384	370.0	369.63	368.91	369.09	373.82	366.91	368.35	371.63	373.46	369.09	370.37	367.45
448	432.37	432.91	429.27	436.55	432.0	434.18	431.64	428.19	433.27	430.39	435.09	432.73
512	490.55	496.21	494.73	496.55	494.37	492.0	497.82	499.27	487.82	495.09	496.9	495.64
576	562.21	558.91	556.0	554.91	559.09	552.91	557.27	559.27	556.0	547.64	558.73	559.09
640	617.64	625.81	621.09	620.37	624.18	620.18	622.18	620.37	615.46	617.63	624.72	625.09
704	684.0	678.73	684.18	679.64	680.55	696.18	680.0	680.18	692.53	683.46	684.54	688.54
768	751.27	749.81	748.0	745.64	748.73	745.82	743.81	744.0	748.0	754.0	742.37	744.19
832	809.32	808.0	807.82	817.09	811.92	813.09	810.55	808.18	814.38	816.55	806.73	806.55
896	880.37	871.27	873.09	874.37	868.37	876.54	871.63	876.37	870.18	874.36	869.45	870.55
960	929.64	939.45	939.64	938.54	940.73	940.0	928.55	930.0	932.37	934.37	941.82	935.45

Для решения задачи имитационного моделирования предложенного модифицированного алгоритма сортировки методом парного обмена были применены те же условия, что и для моделирования синхронной сортировки. Результаты этого моделирования представлены в табл. 4 и на трехмерной диаграмме (рис. 5).

Таблица 4

Модифицированный метод парного обмена

Размерность входного массива данных	Стандартное отклонение											
	120	110	100	90	80	70	60	50	40	30	20	10
64	49.64	50.37	49.64	49.82	48.91	52.37	50.35	52.55	49.28	49.45	50.18	50.19
128	101.45	101.81	101.82	103.27	103.09	102.1	99.46	103.45	100.9	99.1	103.46	100.1
192	162.72	164.36	160.55	163.26	160.37	164.91	164.55	163.82	162.73	162.18	161.09	162.55
256	226.91	223.27	224.91	222.18	224.37	222.19	221.81	227.64	220.9	224.18	219.82	226.11
320	275.09	276.37	278.55	277.27	282.73	275.82	280.73	277.09	276.26	278.55	277.21	279.09
384	340.0	339.63	338.91	339.09	343.82	336.91	338.35	341.63	343.46	339.09	340.37	337.45
448	402.37	402.91	399.27	406.55	402.0	404.18	401.64	398.19	403.27	400.39	405.09	402.73
512	460.55	466.21	464.73	466.55	464.37	462.0	467.82	469.27	457.82	465.09	466.9	465.64
576	532.21	528.91	526.0	534.91	529.09	522.91	527.27	529.27	526.0	517.64	528.73	529.09
640	587.64	595.81	591.09	590.37	594.18	590.18	592.18	590.37	585.46	587.63	594.72	595.09
704	654.0	648.73	654.18	649.64	650.55	666.18	650.0	650.18	662.53	653.46	654.54	658.54
768	721.27	719.81	718.0	715.64	718.73	715.82	713.81	714.0	718.0	724.0	712.37	714.19
832	779.32	778.0	777.82	787.09	781.92	783.09	780.55	788.18	784.38	786.55	776.73	776.55
896	850.37	841.27	843.09	844.37	838.37	846.54	841.63	846.37	840.18	844.36	839.45	840.55
960	899.64	909.45	909.64	908.54	910.73	910.0	898.55	900.0	902.37	904.37	911.82	905.45

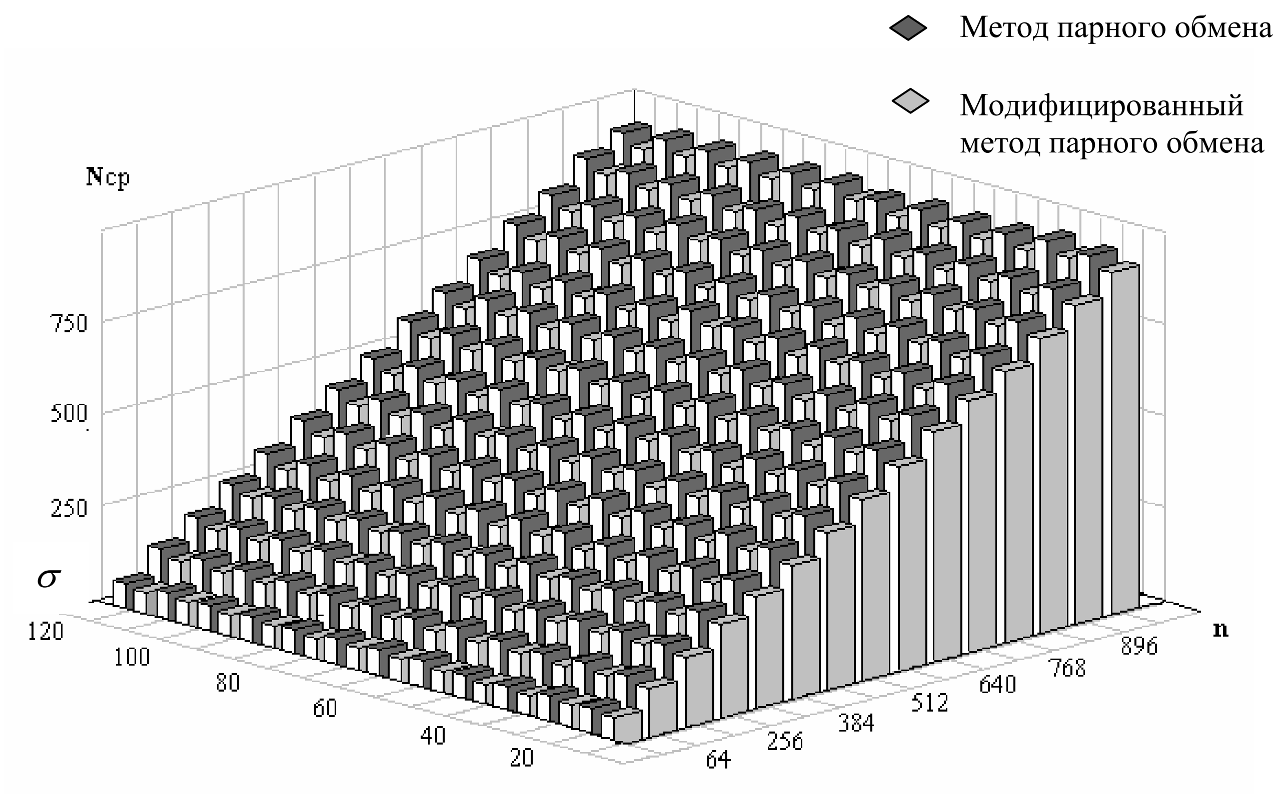


Рис. 5 Временные затраты в процессе сортировки методом парного обмена нормально распределенных элементов векторного массива данных

Выводы

1. Возможность формализованного описания одного из синхронных алгоритмов сортировки подтверждает универсальность метода многоуровневого структурного проектирования программ с использованием аппарата соотношений, который обеспечивает трансформационную сводимость проектируемых алгоритмов.
2. Использование полученных регулярных схем (РС) обеспечивает программную и аппаратную реализацию соответствующих сортировщиков, а также позволяет выполнить синтез, классификацию и сравнительный анализ новых эффективных алгоритмов сортировки массива чисел.
3. Проведенное имитационное моделирование позволило экспериментальным путем проверить предложенные математические модели синхронной сортировки методом парного обмена и подтвердить полученные временные оценки для модифицированного алгоритма сортировки данным методом.

Литература

1. Кучеренко К.И., Очин Е.Ф. Процессоры двумерной медианной фильтрации изображений на основе сортирующих сетей // Автометрия. – 1988. - № 2. – С. 13-19.
2. Кустов В.Н. Параллельная сортировка в сетях коммутации // Автоматика и вычислительная техника. – 1989. - № 3. – С. 76-81.
3. Гузик В.Ф., Золотовский В.Е.,Чиненов С.А. Организация различных методов сортировки в вычислительных системах. // Электронное моделирование. – 1992. – Т. 14. - № 3. – С. 25-28.

4. Григорьев В.Р., Наумов С.П. Нейросетевая организация алгоритмов сортировки на трехмерном оптическом нейрочипе // Автометрия. – 1993. - № 3.–С. 28-37.
5. Ромм Я.Е. Параллельная сортировка слиянием по матрицам сравнений. I // Кибернетика и системный анализ. – 1994. - № 5. – С. 3-23.
6. Вышинский В.А. Сортировка чисел в матрично-алгебраической ЭВМ // УСиМ. – 2001. - № 2. – С. 50-52.
7. Мартынюк Т.Б. Организация ассоциативного процессора с поразрядно-последовательной обработкой информации // Электронное моделирование. – 1996. – Т.18. - № 3. – С.28-31.
8. Зербино Д.Д., Цмоць И.Г. Процессор сортировки чисел на базе клеточных автоматов // 3 – я украинская конференция по автоматическому управлению “Автоматика - 96”. – Севастополь: СевГТУ, 1996. – Т.1. – С. 177-179.
9. Лорин Г. Сортировка и системы сортировки.: Пер. с англ. – М.: Мир.1983.– 384с.
10. Кнут Д. Искусство программирования для ЭВМ. Т.3. Поиск и сортировка. – М.: Мир, 1978. – 844 с.
11. Параллельные вычислительные системы с общим управлением / И.В. Прангишвили, С.Я. Виленкин, Л.И. Медведев. – М.: Энергоатомиздат, 1983. – 312 с.
12. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Методы символьной мультиобработки. – К.: Наук. думка, 1980. – 242 с.
13. Цейтлин Г.Е. Структурное программирование задач символьной мультиобработки // Кибернетика. – 1983. - № 5. – С. 22-30.
14. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра. Языки. Программирование. – К.: Наук. думка, 1989. – 332 с.
15. Цейтлин Г.Е. Проектирование последовательных алгоритмов сортировки: классификация, трансформация, синтез // Программирование. – 1989. - № 3. – С. 3-24.

16. Цейтлин Г.Е. Распараллеливание алгоритмов сортировки // Кибернетика. – 1989. - № 6. – С. 67-74.
17. Мартинюк Т.Б., Хом'юк В.В., Расенко Р.А., Емін С.А. Аналіз часових характеристик при сортуванні випадково розподілених даних // Вісник Вінницького політехнічного інституту. – 2003. - № 1. – С. 34 – 38
18. Шинкаренко В. И.. Об оценке эффективности алгоритмов с учетом архитектуры ЭВМ. // Тези доповідей Міждержавної науково-методичної конференції "Комп'ютерне моделювання". – Дніпродзержинськ, 2000. – С. 268 – 269.
19. http://alglib.chat.ru/sort/index.html.