

19. Kryzhanovskiy A.V., Gushanskiy S.M. Issledovanie kvantovykh algoritmov obnaruzheniya i raspoznavaniya lits s pomoshch'yu kvantovoy neyroseti [Investigation of quantum algorithms for face detection and recognition using a quantum neural network], *Fundamental'nye i prikladnye aspekty komp'yuternykh tekhnologiy i informatsionnoy bezopasnosti: Mater. VII Vserossiyskoy nauchno-tekhnicheskoy konferentsii* [Fundamental and applied aspects of computer technology and information security: Materials of the VII All-Russian Scientific and Technical Conference]. Taganrog: Izd-vo YuFU, 2021, pp. 346-349.
20. Dunjko V. and Briegel H.J. Machine learning & artificial intelligence in the quantum domain: A review of recent progress, *Rep. Prog. Phys.*, June 2018, Vol. 81, pp. 074001.

Статью рекомендовал к опубликованию д.ф.-м.н. Г.В. Куповых.

Гушанский Сергей Михайлович – Южный федеральный университет; e-mail: smgushanskiy@sfedu.ru; г. Таганрог, Россия; кафедра вычислительной техники; к.т.н.; доцент.

Буглов Владислав Евгеньевич – e-mail: buglov@sfedu.ru; тел.: +79508674921; кафедра вычислительной техники; аспирант.

Gushanskiy Sergey Mikhaylovich – Southern Federal University; e-mail: smgushanskiy@sfedu.ru; Taganrog, Russia; the department of computer engineering; cand. of eng. sc.; associate professor.

Buglov Vladislav Evgen'evich – e-mail: buglov@sfedu.ru; phone: +79508674921; the department of computer engineering; graduate student.

УДК 519.178

DOI 10.18522/2311-3103-2021-7-177-188

Д.В. Михайлов

ПРЕОБРАЗОВАНИЕ ПОСЛЕДОВАТЕЛЬНОГО ИНФОРМАЦИОННОГО ГРАФА МЕТОДА ПРОГОНКИ В ПАРАЛЛЕЛЬНУЮ ФОРМУ

Множество вычислительных задач может быть представлено в виде последовательного информационного графа. В общем случае такой информационный граф не может быть приведён к параллельному виду с целью ускорения выполнения его операций. Но в случае если вершины этого графа обладают свойствами ассоциативности, дистрибутивности и т.д., такой граф можно преобразовать в параллельно-конвейерную форму. Эти преобразования могут быть произведены не только над графами, содержащими элементарные операции – сложение, умножение, логическое И и т.д. – но и над графами, содержащими макрооперации. Одним из примеров таких графов является информационный граф решения СЛАУ методом прогонки (методом Томаса). В статье рассмотрено решение для трёхдиагональных СЛАУ. Информационный граф метода прогонки состоит из двух частей: прямого хода, в котором выполняется переход от трёхдиагональной формы к двухдиагональной, и обратного хода, в котором непосредственно вычисляются значения неизвестных. Несмотря на то, что операции, составляющие базовую макрооперацию метода прогонки, обладают свойством ассоциативности, простое преобразование графа к пирамидальному виду не даст необходимого результата. Необходимо преобразовать базовые макрооперации особым образом и изменить то, какие данные на них поступают. После этого возможно будет привести граф к пирамидальному виду. Для обратного хода применяется аналогичное преобразование графа и составляющих его базовых подграфов. Поскольку для того, чтобы начать вычисления в обратном ходе, нам необходимо полное завершение вычислений прямого хода, следует перейти от двух специализированных типов вычислительных блоков к одному универсальному, и построить на его основе универсальную вычислительную структуру.

Метод прогонки; СЛАУ; последовательный информационный граф.

D.V. Mikhailov

CONVERSION OF THE SEQUENTIAL INFORMATION GRAPH OF THE THOMAS ALGORITHM INTO A PARALLEL FORM

Many computational tasks can be represented in the form of a sequential information graph. In the general case, such an information graph cannot be reduced to a parallel form in order to speed up the execution of its operations. But if the vertices of this graph have the properties of associativity, distributivity, etc., such a graph can be transformed into a parallel-pipeline form. These transformations can be performed not only on graphs containing elementary operations - addition, multiplication, logical AND, etc. - but also over graphs containing macro operations. One example of such graphs is the information graph for solving SLAEs by the sweep method (Thomas's method). The article considers a solution for tridiagonal linear systems. The information graph of the sweep method consists of two parts: the forward move, in which the transition from the three-diagonal form to the two-diagonal form is performed, and the reverse move, in which the values of the variables are directly calculated. Despite the fact that the operations that make up the basic macro-operation of the sweep method have the property of associativity, a simple transformation of the graph to a pyramidal form will not give the desired result. It is necessary to transform the basic macro operations in a special way and change what data is received on them. After that, it will be possible to bring the graph to a pyramidal form. For the reverse move, a similar transformation of the graph and its constituent base subgraphs is applied. Since in order to start computations in the reverse run, we need to complete the computations of the forward run, we should switch from two specialized types of computational blocks to one universal one, and build a universal computational structure on its basis.

Tridiagonal matrix algorithm; system of linear equations; sequential information graph.

Введение. В современном мире сложность задач, решаемых на вычислительной технике, постоянно растёт. Компьютерное моделирование природных и антропогенных процессов, необходимое для прогнозирования, задачи криптографии, задачи радиолокации, связанные с управлением в различных системах, поиск по базам данных и т.д. – всё это требует громадных вычислительных затрат, которые возрастают вместе с ростом требований к скорости получения результатов и их качеству. Многие описанные задачи могут быть описаны с помощью последовательного информационного графа.

В определённых условиях абсолютно последовательные информационные графы могут быть приведены к параллельной форме, что позволяет значительно сократить общее время выполнения операций в вычислительных системах, соответствующих этим графам. Подобные преобразования могут быть применены также и для случаев, когда операции в вершинах информационного графа представляют собой сложную комбинацию элементарных операций. Это может быть показано на примере информационного графа решения СЛАУ методом прогонки (алгоритмом Томаса).

Постановка задачи. Метод прогонки применяется для решения систем линейных уравнений вида $Ax = F$, равносильных соотношению:

$$A_i x_{i-1} + B_i x_i + C_i x_{i+1} = F_i.$$

Вычисления производятся в два этапа: на первом этапе, называемом прямым ходом, осуществляется переход от трёхдиагональной матрицы, соответствующей СЛАУ, к двухдиагональной.

Коэффициенты B'_i и вектор F'_i задаются формулами

$$B'_1 = B_1, B'_i = B_i - \frac{A_i}{B'_{i-1}} \cdot C_{i-1},$$

$$F'_1 = F_1, F'_i = F_i - \frac{A_i}{B'_{i-1}} \cdot F'_{i-1}.$$

На втором этапе (обратный ход) вычисляются корни уравнений системы

$$x_n = \frac{F'_n}{B'_n}, x_i = \frac{F'_i - C_i \cdot x_{i+1}}{B'_i}$$

Информационный граф прямого хода представлен на рис. 1.

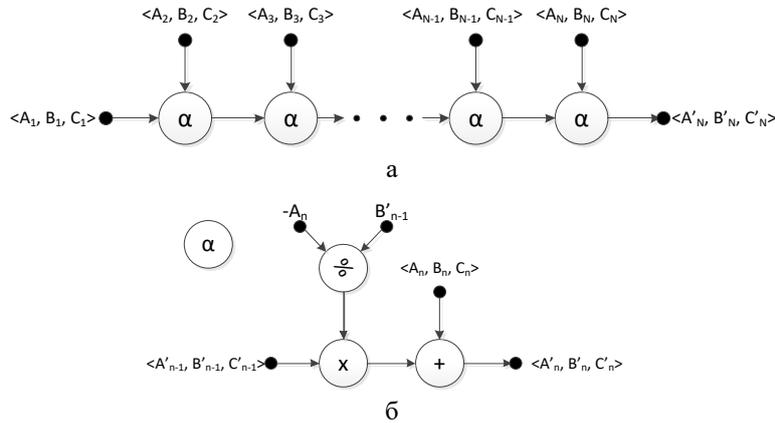


Рис. 1. Информационный граф (а) и подграф (б) прямого хода

Время выполнения всех операций графа составит

$$T = (N - 1) \cdot L_\alpha \cdot \tau, L_\alpha = L_S + L_M + L_D,$$

где N – количество строк в СЛАУ, L_S – латентность сумматора, L_M – латентность умножителя, L_D – латентность делителя.

Предлагаемое решение. Для повышения быстродействия вычислительной системы необходимо сократить это значение. Согласно теореме Иванова [1] в случае, если нам не хватает вычислительного ресурса для реализации всей схемы, максимальная эффективность будет достигаться при реализации единственного подграфа α (рис. 2), причём глубина обратной связи, охватывающей α будет ограничивать возможность добиться плотного потока данных. Операция сложения строк, представляющая собой сложение с весовыми коэффициентами, обладает свойством ассоциативности, и очевидным кажется применение преобразований, аналогичных преобразованиям графа из ассоциативных операций, чтобы перейти от последовательной формы к пирамидальной. Но здесь мы сталкиваемся с проблемой: вычисления B' и F' строго последовательны, для вычисления B'_i необходимо предварительно вычислить B'_{i-1} и т.д. В случае произвольного порядка выполнения операций над строками мы не получим сокращения количества коэффициентов, которое является необходимым элементом решения. Следовательно, этот способ неприменим. Необходима процедура, позволяющая выполнять операции над строками трёхдиагональной матрицы в произвольном порядке.

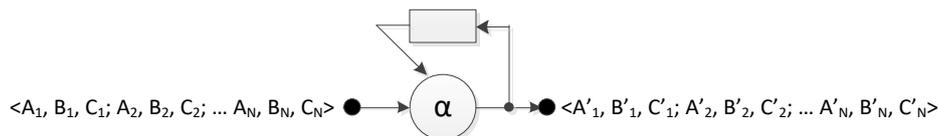


Рис. 2. Вычислительная структура с обратной связью

Как это уже было показано выше в разделе, посвящённом смешанным информационным графам, с помощью метода автоподстановки возможно добиться плотного потока в структурах с обратной связью, при условии, что имеющегося оборудования хватит для одновременной обработки нужного количества данных. На рисунке 3 показано преобразование графа α^2 , соответствующего формуле

$$\begin{aligned}
 B'_i &= B_i - \frac{A_i}{B'_{i-1}} \cdot C_{i-1} = B_i - \frac{A_i}{B_{i-1} - \frac{A_{i-1}}{B'_{i-2}} C_{i-2}} \cdot C_{i-1} =, \\
 &= B_i - \frac{A_i \cdot C_{i-1}}{B_{i-1} \cdot B'_{i-2} - A_{i-1} \cdot C_{i-2}} = B_i - \frac{A_i \cdot C_{i-1} \cdot B'_{i-2}}{B_{i-1} \cdot B'_{i-2} - A_{i-1} \cdot C_{i-2}} = \\
 &= \frac{B_i \cdot (B_{i-1} \cdot B'_{i-2} - A_{i-1} \cdot C_{i-2}) - A_i \cdot C_{i-1} \cdot B'_{i-2}}{B_{i-1} \cdot B'_{i-2} - A_{i-1} \cdot C_{i-2}},
 \end{aligned}
 \tag{1}$$

т.е. содержащего два подграфа α . На рис. 3 штриховой линией показан критический путь графа, определяющий временные затраты на выполнение всех операций графа.

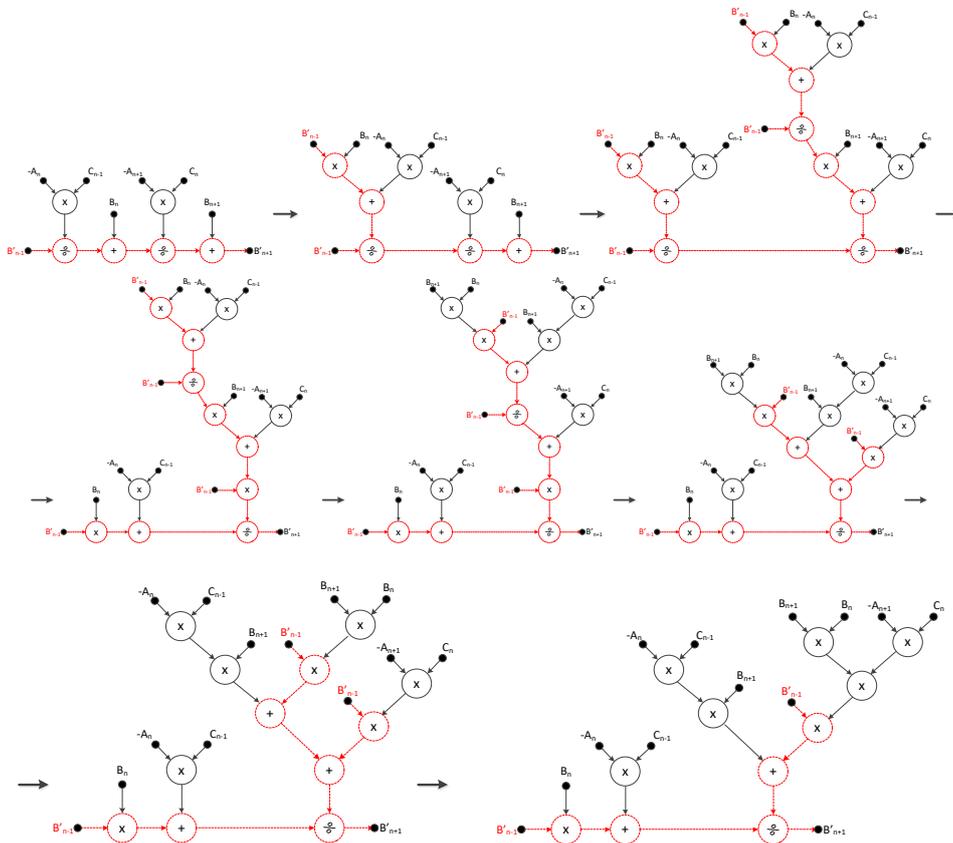


Рис. 3. Преобразование подграфа, содержащего два подграфа α

Ниже показаны попарное объединение подграфов α (рис. 4,а), сводящееся к соединению подграфов α^2 , и структура подграфа α^2 (рис. 4,б).

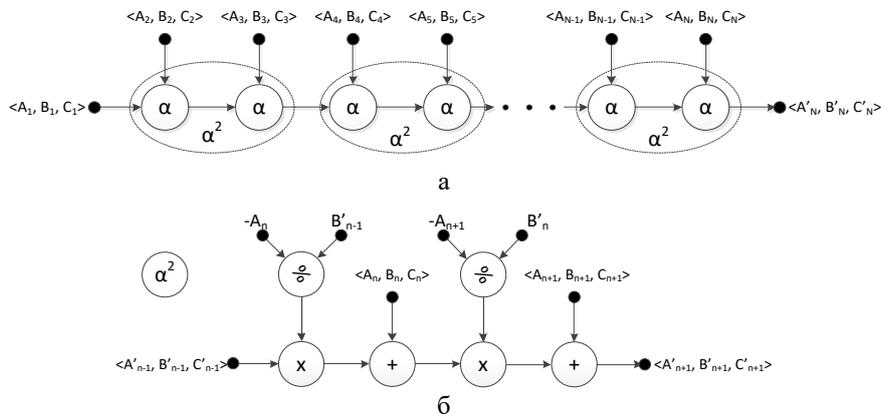


Рис. 4. Информационный граф с объединением подграфов (а) и подграф α^2 (б)

С помощью автоподстановки возможно сократить критический путь графа, содержащего любое количество подграфов α до одной вершины, выполняющей функцию умножения, одной – сложения и ещё одной – деления. Однако затраты оборудования при этом значительно возрастают. Также следует отметить, что для вычисления каждого B'_i необходимо будет организовывать отдельную вычислительную структуру (поскольку на выходе структуры, содержащей n подграфов α мы получим только B'_n , но не B'_{n-1} , B'_{n-2} и т.д.), что увеличит затраты оборудования ещё сильнее.

Таким образом, метод объединения подграфов α и автоподстановки принципиально не решает проблему сокращения времени вычислений.

Эту проблему удаётся решить в случае, если операции проводятся сразу над тремя строками, что позволяет исключить коэффициенты с нечётными номерами и нечётные строки матрицы.

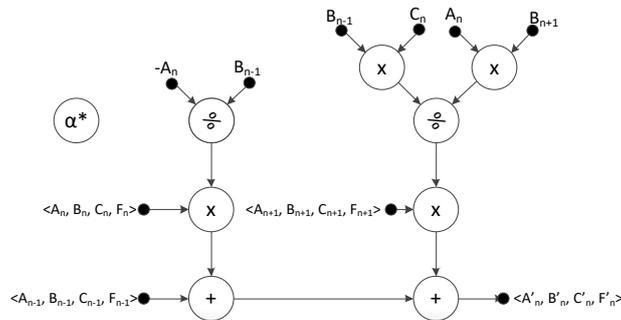


Рис. 5. Подграф α^*

Подграф α^2 может быть преобразован в подграф α^* , представляющий макрооперацию, которая может выполняться в произвольном порядке. Однако если ограничиться преобразованием подграфа α^2 в подграф α^* не меняя входных информационных вершин, мы не получим ускорения и время выполнения всех операций графа составит:

$$T = (N - 1) \cdot L_{\alpha^*} \cdot \tau, L_{\alpha^*} = L_S + L_M + L_D.$$

Причиной этого является сохраняющаяся зависимость вычисления очередного коэффициента от вычисления предыдущего (рис. 6).

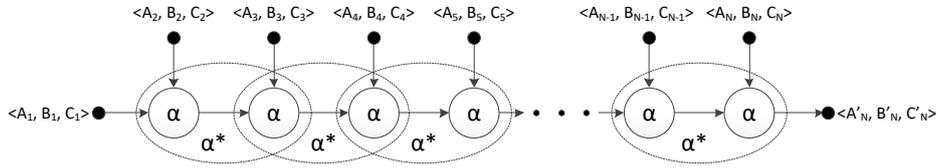


Рис. 6. Граф, состоящий из подграфов α^*

Для ускорения вычислений нам необходимо заменить ещё и входные информационные вершины подграфа, добившись информационной независимости каждого подграфа α^* от любого предыдущего подграфа. В результате мы получим граф, представленный на рис. 7.

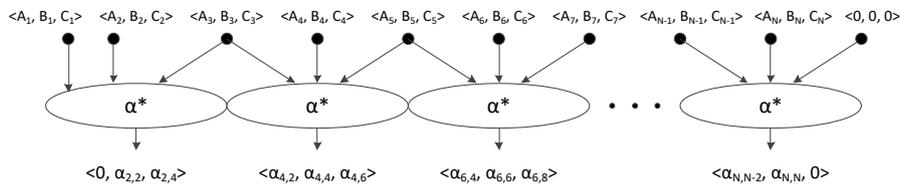


Рис. 7. Первая ступень преобразованного графа

Таким образом формируются коэффициенты новой матрицы, которая в два раза меньше исходной. Далее мы производим аналогичные преобразования уже над коэффициентами из этой матрицы и т.д. Все вычисления могут быть произведены независимо от предыдущих вычислений на этом же ярусе, следовательно, можно перейти к пирамидальной структуре. Итоговый граф представлен на рис. 8.

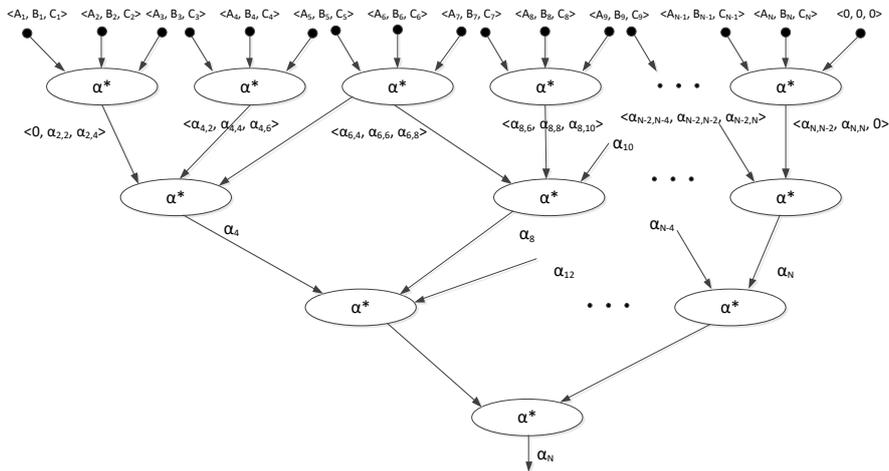


Рис. 8. Граф после преобразования

В итоге мы получим граф прямого хода решения СЛАУ, имеющий пирамидальную структуру. Граф, состоящий из подграфов α^* , имеющий пирамидальную структуру и представленный на рис. 9, с точностью до входных вершин соответ-

вует графу циклической редукции. Метод циклической редукции заключается в исключении из уравнений неизвестных, как и в методе прогонки, однако в ней исключение ведут одновременно по всей СЛАУ [2].

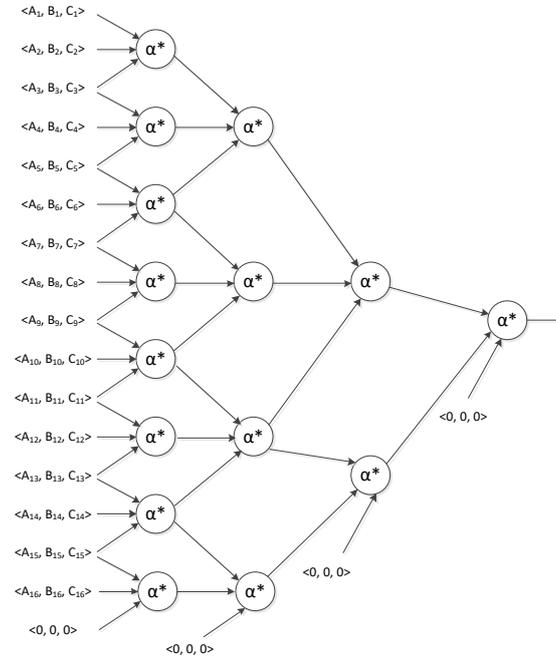


Рис. 9. Пример структуры графа, состоящего из подграфов α^*

Время выполнения всех операций в таком графе будет равно

$$T_{\text{прм}} = \lceil \log_2 N \rceil \cdot L_{\alpha^*} \cdot \tau.$$

Таким образом, в случае, если нам хватает оборудования для реализации всех операционных вершин прямого хода решения СЛАУ методом циклической редукции, мы получим ускорение решения примерно в $\frac{N}{\lceil \log_2 N \rceil}$ раз.

В случае же, если имеющегося оборудования недостаточно для реализации всего графа, нам придётся преобразовать граф для конвейерных вычислений. Для этого мы должны будем выделить одинаковые пирамидальные подграфы, содержащие R подграфов α^* , где R – доступный нам вычислительный ресурс. Эти пирамидальные подграфы должны охватывать весь массив входных информационных вершин, что приведёт к частичному перекрытию подграфов.

Затем мы переходим от информационного графа к вычислительной структуре. Пирамидальному подграфу, выделенному выше, соответствует пирамидальная структура из блоков, соответствующих подграфу α^* , а после этой структуры размещается вычислительный блок, соответствующий подграфу α^k , охваченный обратной связью. Для достижения плотного потока необходимо, чтобы k (степень автоподстановки подграфа α^k) равнялась латентности блока, соответствующего подграфу α^2 . В результате на выходе пирамидального вычислительного блока мы получаем последовательность наборов коэффициентов $\langle 0, a_{h,h}, a_{h,2h} \rangle, \langle a_{2h,h}, a_{2h,2h}, a_{2h,3h} \rangle, \dots, \langle a_{N-h,N-2h}, a_{N-h,N-h}, a_{N-h,N} \rangle, \langle a_{N,N-h}, a_{N,N}, 0 \rangle$, над которыми затем выполняется макрооперация α (прогонка). $h = 2^p$, где $p = \lceil \log_2 R \rceil$. В результате мы получим массив коэффициентов B'_{n^*h} , где n целое.

Особенностью как метода прогонки, так и метода циклической редукции является то, что невозможно перейти к вычислениям обратного хода не вычислив все коэффициенты B' и F' во время прямого хода. В случае, если нам хватает ресурса для реализации всей вычислительной структуры, это не создаёт неудобств. Однако в обратном случае эта особенность приводит к тому, что в вычислениях возникает разрыв – часть вычислительной структуры, отвечающая за обратный ход будет простаивать, ожидая, пока закончатся все вычисления прямого хода. Результат этих вычислений сохраняется в блок ОЗУ, откуда затем будут поданы на обратный ход.

Время выполнения всех операций прямого хода в этом случае составит

$$T_{\text{прм}} = (\lceil \log_2 R \rceil + \frac{2N}{r} \cdot L_{\alpha^*} + L_{\alpha} \cdot (\lceil \log_2 N \rceil - m)) \cdot \tau,$$

r – количество одновременно подаваемых строк, $r = \frac{R+1}{2}$

$$R \geq \sum_{k=1}^m (2^k) - m, m \in \mathbb{N}, m = \lceil \log_2(r - 1) \rceil,$$

m – количество слоёв, которые возможно реализовать на имеющемся оборудовании одновременно.

Вычисление свободного члена F' осуществляется на этой же вычислительной схеме, поэтому не требует дополнительных временных затрат, но требует дополнительных затрат оборудования. На рис. 10 представлен подграф α^* с указанием входных информационных вершин, включающий в себя вычисления свободного члена F' .

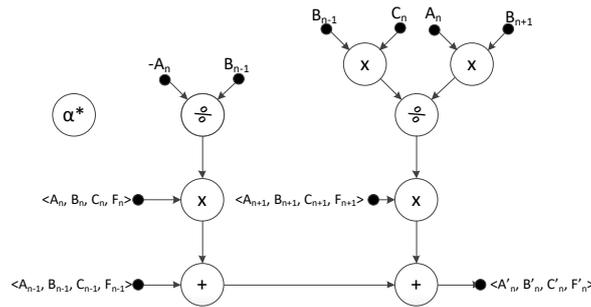
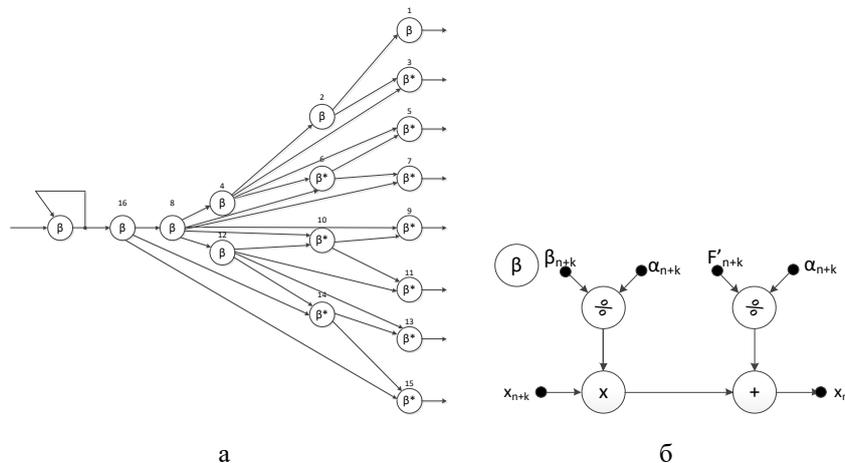


Рис. 10. Подграф α^*

Теперь рассмотрим обратный ход. Пример информационного графа обратного хода циклической редукции показан на рис. 11,а.



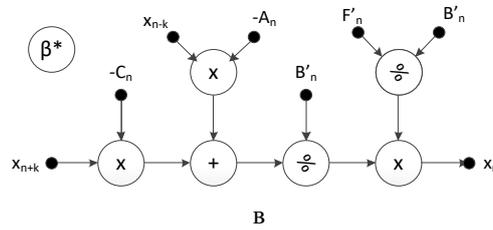


Рис. 11. Пример графа обратного хода (а) и подграфы β (б) и β^* (в)

Результаты вычислений прямого хода, которые мы сохранили в ОЗУ, теперь подаются на вход структуры обратного хода, где они сначала обрабатываются в блоке β , охваченном обратной связью, а затем подаются на пирамидальную структуру, включающую в себя блоки β и β^* (рис. 12). В результате на выходе мы получаем результаты $\langle x_1, x_2, \dots, x_r \rangle$, $\langle x_{r+1}, x_{r+2}, \dots, x_{2r} \rangle$ и т.д. вплоть до $\langle x_{N-r+1}, x_{N-r+2}, \dots, x_N \rangle$.

$$\text{Время выполнения } T_{\text{обр}} = (\lceil \log_2(r_0 + 1) \rceil + \frac{N}{r_0} \cdot L_{\beta^*}) \cdot \tau$$

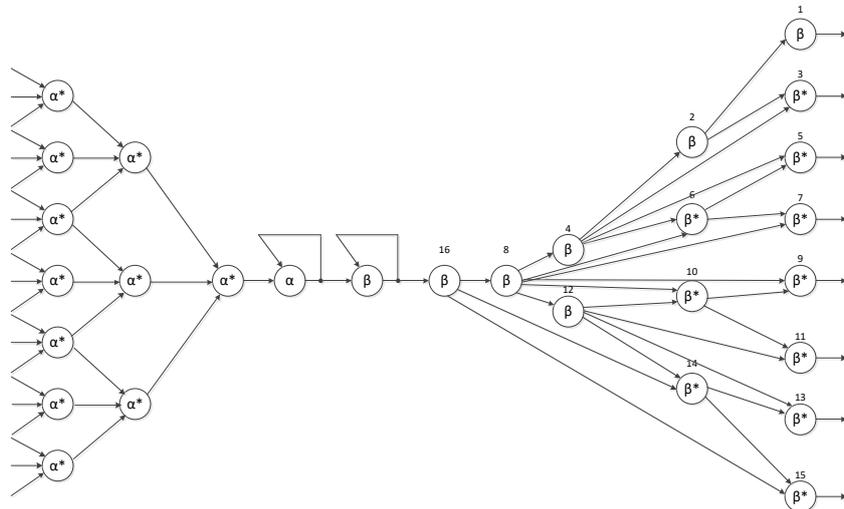


Рис. 12. Пример графа, включающего прямой и обратный ход

Поскольку для того, чтобы начать вычисления в обратном ходе, нам необходимо полное завершение вычислений прямого хода, следует перейти от двух специализированных типов вычислительных блоков (α и β) к одному универсальному α/β , и построить на его основе универсальную вычислительную структуру, которая будет последовательно осуществлять вычисления сначала прямого хода, а затем обратного хода. Сигналы на входы универсальных блоков будут подаваться через мультиплексоры, управляемые в зависимости от того, прямой или обратный ход вычисляется в данный момент. Вид универсального вычислительного блока α/β приведён на рис. 13,а. Пример вычислительной структуры на основе универсальных блоков приведён на рис. 13,б.

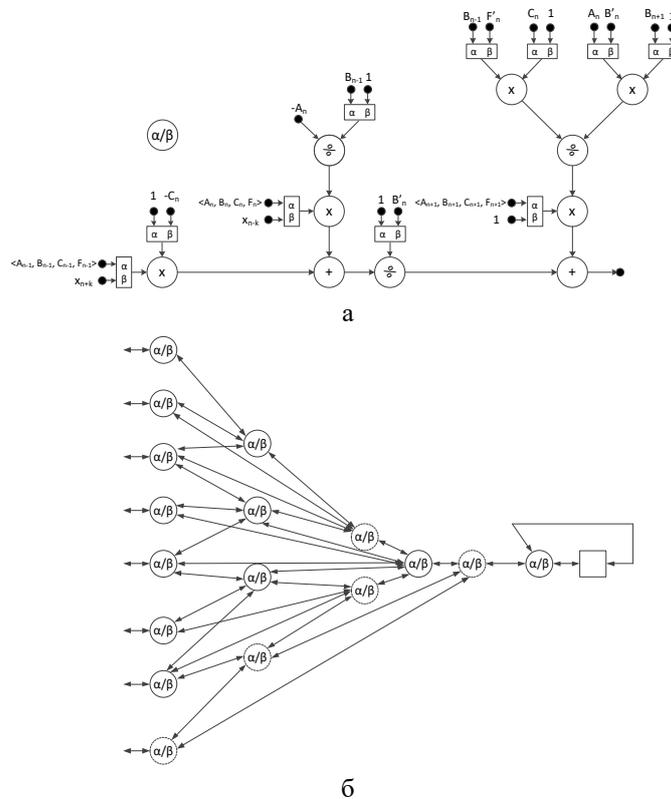


Рис. 13. Универсальный вычислительный блок α/β (а) и пример вычислительной структуры на его основе (б)

Время выполнения всех операций в такой структуре составит

$$T = \left((\lceil \log_2 R \rceil + 2 + \lceil \log_2 (R - 2) \rceil) \cdot L_{\alpha\beta} + 2 \cdot \left\lceil \frac{2N}{r + 1} \right\rceil \right) \cdot \tau.$$

Заключение. Применение описанных методов позволяет добиться значительного ускорения решения СЛАУ методом прогонки. Для произвольно выбранного случая $L_{\alpha\beta} = 25, L_{\alpha} = 20, L_{\beta} = 20, N = 2^{20}, R = 1024$ ускорение составит около пяти тысяч раз. Использование универсальной вычислительной структуры приводит к повышению удельной производительности вычислительной системы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Иванов А.И., Коновальчик П.М. Методы организации параллельно-конвейерных вычислений для решения расчетомких задач // Информационные технологии. – 2004. – № 12. – С. 38-43.
2. Ильин В.П., Кузнецов Ю.И. Трехдиагональные матрицы и их приложения. – М.: Наука. Главная редакция физико-математической литературы, 1985. – 208 с.
3. Ортега Джеймс. Введение в параллельные и векторные методы решения линейных систем. – М.: Изд-во Мир, 1991. – 367 с.
4. Касаркин А.В., Левин И.И. Структурная реализация задачи нахождения всех максимальных клик графа на реконфигурируемых вычислительных системах // Вестник компьютерных и информационных технологий – 2018. – № 10 (172). – С. 3-10.
5. Левин И.И., Пелипец А.В. Методология распараллеливания по итерациям при решении задач линейной алгебры на реконфигурируемых вычислительных системах // Вестник компьютерных и информационных технологий. – 2016. – № 7 (145). – С. 34-40.

6. *Левин И.И., Пелипец А.В., Сорокин Д.А.* Решение задачи LU-декомпозиции на реконфигурируемых вычислительных системах: оценка и перспективы // Известия ЮФУ. Технические науки. – 2015. – № 7 (168). – С. 62-70.
7. *Гуленок А.А.* Оптимизация блоков перекоммутации в вычислительных структурах параллельных программ для реконфигурируемых вычислительных систем // Десятая Всероссийская Мультиконференция по проблемам управления (МКПУ-2017): Матер. 10-й Всероссийской мультиконференции: в 3 т. – Ростов-на-Дону: Изд-во: ЮФУ, 2017. – С. 130-131.
8. *Каляев И.А., Левин И.И.* Реконфигурируемые мультиконвейерные вычислительные системы для решения потоковых задач // Информационные технологии и вычислительные системы. – 2011. – № 2. – С. 12-22.
9. *Левин И.И.* Структурно-процедурная организация параллельных вычислений // Матер. Четвертой Международной научной молодежной школы "Высокопроизводительные вычислительные системы". – Таганрог: Изд-во ТТИ ЮФУ, 2007. – С. 49-68.
10. *Михайлов Д.В.* Преобразование некоторых видов последовательных информационных графов в параллельно-конвейерную форму // Известия ЮФУ. Технические науки. – 2020. – № 7 (217). – С. 78-93.
11. Задача суммирования элементов массива // Лаборатории Параллельных информационных технологий НИВЦ МГУ. – URL: <https://parallel.ru/fpga/Summ2> (дата обращения: 25.10.2021).
12. *Ефимов С.С.* Обзор методов распараллеливания алгоритмов решения некоторых задач вычислительной дискретной математики // Математические структуры и моделирование. – 2007. – Вып. 17. – С. 72-93.
13. *Дасгупта С., Пападимитриу Х., Вазирани У.* Алгоритмы: пер. с англ. / под ред. А. Шеня. – М.: МЦНМО, 2014. – 320 с.
14. *Харари Ф.* Теория графов. – М.: Мир, 1973. – 300 с.
15. *Кормен Т.М. и др.* Часть VI. Алгоритмы для работы с графами // Алгоритмы: построение и анализ = Introduction to Algorithms. – 2-е изд. – М.: Вильямс, 2006. – С. 1296.
16. *Edward A. Bender.* Lists, Decisions and Graphs. With an Introduction to Probability. S. Gill Williamson. – 2010. – 251 p.
17. *Trudeau, Richard J.* Introduction to graph theory. – Dover Publications, Inc. New York, 1993. – 224 p.
18. *Старченко А.В., Берцун В.Н.* Методы параллельных вычислений. – Томск: Изд-во Том. ун-та, 2013. – 223 с.
19. *Фрич Р., Перегуда Е.Е., Мацневский С.В.* Избранные главы теории графов: учеб. пособие: пер. с нем. Е.Е. Перегуда / под ред. С.В. Мацневского. – Калининград: Изд-во РГУ им. И. Канта, 2008. – 204 с.
20. *Tamт У.* Теория графов: пер. с англ. Г.П. Гаврилова. – М.: Мир, 1988. – 424 с.

REFERENCES

1. *Ivanov A.I., Konoval'chik P.M.* Metody organizatsii parallel'no-konveyernykh vychisleniy dlya resheniya raschetnoemkikh zadach [Methods of organization of parallel-pipeline computations for solving computationally intensive tasks], *Informatsionnye tekhnologii* [Information Technologies], 2004, No. 12, pp. 38-43.
2. *Il'in V.P., Kuznetsov Yu.I.* Trekhdiagonal'nye matritsy i ikh prilozheniya [Tridiagonal matrices and their applications]. Moscow: Nauka. Glavnaya redaktsiya fiziko-matematicheskoy literatury, 1985, 208 p.
3. *Ortega Dzheymс.* Vvedenie v parallel'nye i vektornye metody resheniya lineynykh system [Introduction to parallel and vector methods for solving linear systems]. Moscow: Izd-vo Mir, 1991, 367 p.
4. *Kasarkin A.V., Levin I.I.* Strukturnaya realizatsiya zadachi nakhozheniya vseh maksimal'nykh klik grafa na rekonfiguriruemyykh vychislitel'nykh sistemakh [Structural realization of the problem of finding all maximal cliques of a graph on reconfigurable computing systems], *Vestnik komp'yuternykh i informatsionnykh tekhnologiy* [Bulletin of Computer and Information Technologies], 2018, No. 10 (172), pp. 3-10.
5. *Levin I.I., Pelipets A.V.* Metodologiya rasparallelivaniya po iteratsiyam pri reshenii zadach lineynoy algebrы na rekonfiguriruemyykh vychislitel'nykh sistemakh [Methodology of parallelization by iterations in solving linear algebra problems on reconfigurable computing systems], *Vestnik komp'yuternykh i informatsionnykh tekhnologiy* [Bulletin of Computer and Information Technologies], 2016, No. 7 (145), pp. 34-40.

6. Levin I.I., Pelipets A.V., Sorokin D.A. Reshenie zadachi LU-dekompozitsii na rekonfiguriruemyykh vychislitel'nykh sistemakh: otsenka i perspektivy [Solving the LU-decomposition problem on reconfigurable computing systems: assessment and prospects], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2015, No. 7 (168), pp. 62-70.
7. Gulenok A.A. Optimizatsiya blokov perekommutatsii v vychislitel'nykh strukturakh parallelnykh programm dlya rekonfiguriruemyykh vychislitel'nykh sistem [Optimization of recommutation blocks in computing structures of parallel programs for reconfigurable computing systems], *Desyataya Vserossiyskaya Mul'tikonferentsiya po problemam upravleniya (MKPU-2017): Mater. 10-y Vserossiyskoy mul'tikonferentsii* [The tenth all-Russian multimedia conference on management (mcpu-2017): proceedings of the 10th all-Russian conference]: in 3 vol. Rostov-on-Don: Izd-vo: YuFU, 2017, pp. 130-131.
8. Kalyaev I.A., Levin I.I. Rekonfiguriruemye mul'tikonveyernye vychislitel'nye sistemy dlya resheniya potokovykh zadach [Reconfigurable multiconveyor computing systems for solving streaming tasks], *Informatsionnye tekhnologii i vychislitel'nye sistemy* [Information technologies and computing systems], 2011, No. 2, pp. 12-22.
9. Levin I.I. Strukturno-protsedurnaya organizatsiya parallelnykh vychisleniy [Structural and procedural organization of parallel computing], *Mater. Chetvortoy Mezhdunarodnoy nauchnoy molodezhnoy shkoly "Vysokoproizvoditel'nye vychislitel'nye sistemy"* [Materials of the Fourth International Scientific Youth School "High-performance computing systems"]. Taganrog: Izd-vo TTI YuFU, 2007, pp. 49-68.
10. Mikhaylov D.V. Preobrazovanie nekotorykh vidov posledovatel'nykh informatsionnykh grafov v parallelnoy-konveyernuyu formu [Transformation of some types of sequential information graphs into a parallel-conveyor form], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2020, No. 7 (217), pp. 78-93.
11. Zadacha summirovaniya elementov massiva [The task of summing array elements], *Laboratorii Parallelnykh informatsionnykh tekhnologiy NIVTS MGU* [Laboratories of Parallel information technologies of the NIVC MSU]. Available at: <https://parallel.ru/fpga/Summ2> (accessed 25 October 2021).
12. Efimov S.S. Obzor metodov rasparallelivaniya algoritmov resheniya nekotorykh zadach vychislitel'noy diskretnoy matematiki [Review of methods for parallelizing algorithms for solving some problems of computational discrete mathematics], *Matematicheskie struktury i modelirovanie* [Mathematical structures and modeling], 2007, Issue 17, pp. 72-93.
13. Dasgupta S., Papadimitriou Kh., Vazirani U. Algoritmy [Algorithms]: transl. from engl., ed. by A. Shenya. Moscow: MTSNMO, 2014, 320 p.
14. Kharari F. Teoriya grafov [Graph theory]. Moscow: Mir, 1973, 300 p.
15. Kormen T.M. i dr. Chast' VI. Algoritmy dlya raboty s grafami [Part VI. Algorithms for working with graphs], *Algoritmy: postroyeniye i analiz = Introduction to Algorithms* [Algorithms: construction and analysis = Introduction to Algorithms]. 2nd ed. Moscow: Vil'yams, 2006, pp. 1296.
16. Edward A. Bender. Lists, Decisions and Graphs. With an Introduction to Probability. S. Gill Williamson, 2010, 251 p.
17. Trudeau, Richard J. Introduction to graph theory. Dover Publications, Inc. New York, 1993, 224 p.
18. Starchenko A.V., Bertsun V.N. Metody parallelnykh vychisleniy [Methods of parallel computing]. Tomsk: Izd-vo Tom. un-ta, 2013, 223 p.
19. Frich R., Peregud E.E., Matsievskiy S.V. Izbrannyye glavy teorii grafov: ucheb. posobie [Selected chapters of graph theory: studies. manual]: transl. from german by E.E. Pereguda, ed. by S.V. Matsievskogo. Kaliningrad: Izd-vo RGU im. I. Kanta, 2008, 204 p.
20. Tatt U. Teoriya grafov [Graph theory]: transl. from the engl. G.P. Gavrilova. Moscow: Mir, 1988, 424 p.

Статью рекомендовала к опубликованию д.т.н. А.В. Никитина.

Михайлов Денис Васильевич – ООО «НИЦ СЭ и НК»; e-mail: mixailow.den@gmail.com; г. Таганрог, Россия; тел.: 89287502869.

Mikhailov Denis Vasilevich – «SRC SC & NC» Co Ltd; e-mail: mixailow.den@gmail.com; Taganrog, Russia; phone: +79287502869.