

Один метод построения циклов в графе

Е. П. Емельченков, В. И. Мунерман*, Д. В. Мунерман, Т. А. Самойлова

ФГБОУ ВО «Смоленский государственный университет», г. Смоленск, Российская Федерация
Адрес: 214000, Российская Федерация, г. Смоленск, ул. Пржевальского, д. 4

* vimoona@gmail.com

Аннотация

Задачи нахождения циклов в графе – неотъемлемая составная часть геоинформационных, логистических, и навигационных информационных систем. Задачи нахождения гамильтоновых и эйлеровых циклов, задача коммивояжера настолько важны в современных информационных системах, обеспечивающих решение задач в различных предметных областях от традиционных транспортных до научных проблем в химии и биологии, что этому посвящено большое число публикаций. Современная особенность задач нахождения циклов в графе заключается в необходимости обработки больших и сверхбольших данных (Big Data). Методы, которые дают точное решение, сводятся к алгоритмам с экспоненциальной вычислительной сложностью. Для уменьшения сложности предлагаются эвристические методы, например, генетические алгоритмы. Предложен многомерно-матричный подход к обработке больших графов, ориентированный на построение всех возможных циклов независимо от способа вычисления их свойств (например, стоимости, гамильтоновости и прочих). Этот подход обеспечивает эффективное распараллеливание алгоритмов построения всех циклов и использование технологии in database для построения всех циклов, что возможно в силу изоморфизма алгебры многомерных матриц и реляционной алгебры для рассматриваемого класса задач.

Ключевые слова: ориентированные графы, циклы в графе, многомерные матрицы, базы данных, параллельное программирование

Авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Один метод построения циклов в графе / Е. П. Емельченков [и др.] // Современные информационные технологии и ИТ-образование. 2021. Т. 17, № 4. С. 814-823. doi: <https://doi.org/10.25559/SITITO.17.202104.814-823>

© Емельченков Е. П., Мунерман В. И., Мунерман Д. В., Самойлова Т. А., 2021



Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.



Some Method for Constructing Cycles in a Graph

E. P. Yemelchenkov, V. I. Munerman*, D. V. Munerman, T. A. Samoylova

Smolensk State University, Smolensk, Russian Federation

Address: 4 Przhevalsky St., Smolensk 214000, Russian Federation

* vimoon@gmail.com

Abstract

The problem of finding cycles in a graph is an integral part of geoinformation, logistic, and navigation information systems. The problems of finding Hamiltonian and Euler cycles, the traveling salesman problem are so important in modern information systems that provide solutions to problems in various subject areas from traditional transport to scientific problems in chemistry and biology that a large number of publications are devoted to this. A modern feature of the tasks of finding cycles in a graph is the need to process large and super-large data (Big Data). Methods that give an exact solution are reduced to algorithms with exponential computational complexity. To reduce complexity, heuristic methods are proposed, for example, genetic algorithms. A multidimensional matrix approach to the processing of large graphs is proposed, focused on constructing all possible cycles, regardless of the method for calculating their properties (for example, cost, Hamiltonianity, and others). This approach provides efficient parallelization of algorithms for constructing all loops and the use of in database technology for constructing all loops, which is possible due to the isomorphism of the algebra of multidimensional matrices and relational algebra for the class of problems under consideration.

Keywords: directed graphs, cycles in a graph, multidimensional matrices, databases, parallel programming

The authors declare no conflict of interest.

For citation: Yemelchenkov E.P., Munerman V.I., Munerman D.V., Samoylova T.A. Some Method for Constructing Cycles in a Graph. *Sovremennye informacionnye tehnologii i IT-obrazovanie = Modern Information Technologies and IT-Education*. 2021; 17(4):814-823. doi: <https://doi.org/10.25559/SITITO.17.202104.814-823>



Введение

Задачи нахождения циклов в графе – неотъемлемая составная часть геоинформационных, логистических, и навигационных информационных систем. Задачи нахождения гамильтоновых и эйлеровых циклов, задача коммивояжера настолько важны в современных информационных системах, обеспечивающих решение задач в различных предметных областях от традиционных транспортных до научных проблем в химии и биологии, что этому посвящено большое число публикаций, например, [1-6], [23]. Современная особенность задач нахождения циклов в графе заключается в необходимости обработки больших и сверхбольших данных (Big Data). Традиционный подход к их решению основан на математическом моделировании с использованием теории графов. При всей своей естественности для представления объектов предметных областей, их свойств и связей, этот подход обладает одним существенным недостатком. Традиционные методы, которые дают точное решение, сводятся к переборным алгоритмам с экспоненциальной вычислительной сложностью. Для уменьшения сложности предлагаются различные способы, среди которых можно отметить эвристические методы, такие как, например, генетические алгоритмы [7-11], [22]. Причем последние ориентированы на случай графа, полностью помещающегося в оперативной памяти. Недостатки такого рода решений заключаются в том, что, во-первых, часто они основаны на эвристиках, применимых исключительно к условиям конкретной предметной области, что требует существенной переработки алгоритмов при переходе к другим предметным областям; во-вторых, для повышения их эффективности с использованием методов параллельного программирования приходится использовать сложные искусственные приемы. Поэтому в статье предложен универсальный алгебраический подход к обработке больших графов, ориентированный на построение всех возможных циклов независимо от способа вычисления их свойств (например, стоимости, гамильтоновости и прочих). В основу подхода положены теория многомерных матриц¹ [24], [25] и специальные двухосновные алгебраические системы (абстрактные алгебраические машины). Этот подход позволяет:

- эффективное распараллеливание алгоритмов построения всех циклов;
- использование технологии in database для построения всех циклов, что возможно в силу изоморфизма алгебры многомерных матриц и реляционной алгебры для рассматриваемого класса задач.

Построение маршрутов в графе

Решение задачи построения всех циклов в ориентированном графе легко реализуется на основе применения одного из вариантов операции умножения многомерных матриц – (0) -свернутого произведения, при котором не производится суммирование элементов (отсутствуют кэлиевы индексы). Для этой цели сначала рассматривается алгоритм построения всех возможных маршрутов в графе [12], [13].

Для построения маршрутов в графе справедливо следующее утверждение. Если матрица $G = \|w_{i_1 i_2}\|$ – матрица смежности графа, а $(k+1)$ -мерная матрица ${}^{(0,1)}G^k$ – степень матрицы G по $(1, 0)$ -свернутому произведению, то набор значений индексов отличного от нейтрального элемента этой матрицы i_1^*, \dots, i_{k+1}^* соответствует номерам вершин пути, который связывает вершины i_1^* и i_{k+1}^* . Пусть матрица $G = \|w_{i_1 i_2}\|$ – матрица смежности (или весов ребер) графа. Элементы матрицы $w_{i_1 i_2}$ принадлежат некоторому типу, например, логическому или числовому и имеют значения отличные от нейтрального элемента этого типа, если в графе есть ребро $i_1 i_2$, и есть нейтральный элемент в противном случае. После умножения матрицы G на себя получается трехмерная матрица ${}^{(1,0)}G^2 = \|w_{i_1 i_2 i_3}\|$. Здесь индекс i_1 обеих матриц-операндов – свободный (в матрице-результате он повторяется и второе его вхождение получает новое обозначение – i_3), а индекс i_2 – скоттов. Эта матрица содержит сведения обо всех маршрутах, проходящих по двум ребрам. После умножения матрицы ${}^{(1,0)}G^2$ на матрицу G получается четырехмерная матрица ${}^{(1,0)}G^3 = \|w_{i_1 i_2 i_3 i_4}\|$. В этой матрице индексы i_1, i_2 матрицы ${}^{(1,0)}G^2$ и индекс i_1 матрицы G (в матрице-результате он обозначается i_4) – свободные. В качестве скоттовых используются индекс i_3 матрицы G^2 и индекс i_2 матрицы G . Матрица ${}^{(1,0)}G^3$ содержит сведения обо всех маршрутах, проходящих по трем ребрам. Процесс продолжается до тех пор, пока не будет получена матрица, все элементы которой – нейтральные. Предшествующая ей матрица ${}^{(1,0)}G^k$ содержит сведения обо всех маршрутах, состоящих из k ребер. Тогда, если элемент $w_{i_1^* i_2^* \dots i_k^*}$ отличен от нейтрального элемента, то последовательность значений индексов $i_1^*, i_2^*, \dots, i_k^*$ есть последовательность вершин, через которые проходит маршрут, начинающийся в вершине i_1^* заканчивающийся в вершине i_k^* проходящий по $k-1$ ребрам и обладающий свойством, определяемым значением $w_{i_1^* i_2^* \dots i_k^*}$. Вычислительная сложность предложенного метода полиномиальная и имеет порядок $O(n^k + n^{k-1} + \dots + n^3)$, где n – число вершин графа, k – число ребер в маршруте, проходящем через наибольшее число вершин.

Это утверждение имеет конструктивный характер, так как фактически задает алгоритм построения маршрутов в графе. Как известно, цикл в графе – это маршрут, у которого начальная и конечная вершины совпадают². Получить все циклы на очередном этапе выполнения алгоритма построения маршрутов можно выполнив операцию сечения вычисленной многомерной матрицы – $(1, 0)$ -свернутой степени матрицы смежности графа, при условии, что значение первого индекса совпадает со значением последнего индекса этой многомерной матрицы. Перед вычислением следующей $(1, 0)$ -свернутой степени матрицы смежности графа необходимо в последней вычисленной многомерной матрице всем элементам, у которых значения первого и последнего индексов совпадают присвоить значение нейтрального элемента. Это позволит на следующем этапе избежать построения циклов, содержащих повторяющиеся вершины. Если в решаемой задаче имеются ограничения на количество вершин, через которые проходит цикл, то в этом случае достаточно вычислить все $(1, 0)$ -свернутые степени матрицы смежности графа до ${}^{(1,0)}G^k$, где k – наибольшее число вершин в цикле. При построении всех циклов

¹ Соколов Н. П. Введение в теорию многомерных матриц. Киев: Наукова думка, 1972. 176 с.

² Christofides N. Graph Theory: An Algorithmic Approach (Computer science and applied mathematics). Academic Press, Inc., USA, 1975. 400 p.



необходимо строить все $(1, 0)$ -свернутые степени матрицы смежности графа до получения матрицы, содержащей гамильтонов цикл, если такой существует в графе, то есть до получения матрицы $(1, 0)G^n$, где n – количество вершин графа.

Алгебраический метод построения циклов в графе

Далее рассматривается построение программно-аппаратных комплексов для построения циклов в графе на основе алгебры многомерных матриц и реляционной алгебры.

Алгебра многомерных матриц. Используя предложенный метод, можно строить циклы с различными свойствами, например, выполняя одновременное вычисление суммы весов ребер. Очевидно, что для каждого конкретного случая специфическим будет только тип элементов матриц. С другой стороны, параллельные алгоритмы умножения многомерных матриц достаточно сложны. Следовательно, разработка в полном объеме программного обеспечения для каждой задачи трудоемка и нецелесообразна. Проблема легко решается средствами объектно-ориентированного программирования. Поскольку в общем случае абстрактный тип данных (класс) – это многомерная алгебраическая система, можно использовать метод построения абстрактной, в данном случае многомерно-матричной, алгебраической машины [14], которая представляет собой двухосновную алгебраическую систему, первая основа которой задает структуру данных, а вторая тип элементов структуры. В рассматриваемом случае абстрактная многомерно-матричная машина имеет вид $EM = \langle M, X; \Omega; \Pi \rangle$ где M – множество многомерных матриц, все индексы которых принимают значения от 1 до n – числа вершин графа, а X – тип элементов матриц, определяемый исходными требованиями. Минимальное требование к типу X состоит в том, чтобы на X были определены две алгебраические операции, одна из которых трактуется как аддитивная, а вторая – как мультипликативная. То есть, тип X должен быть по каждой из этих операций, по крайней мере, алгебраической структурой, называемой группоидом. В реальных условиях типами могут быть достаточно сложные алгебраические структуры, такие как кольца и поля. В таблице 1 приведены формальные определения и описания операций сигнатуры Ω абстрактной матричной машины EM .

Таблица 1. Сигнатура операций EM
Table 1. EM operation signature

Операция	Описание операции
$\oplus : X \times X \rightarrow X$	аддитивная операция над элементами матриц
$\otimes : X \times X \rightarrow X$	мультипликативная операция над элементами матриц
$\overset{\circ}{\circ} : M \rightarrow M$	транспонирование матрицы
$\overset{\mu}{\mu} : M \rightarrow M$	свертка многомерной матрицы
$+$: $M \times M \rightarrow M$	сумма матриц;
\times : $M \times M \rightarrow M$	$(1, 0)$ -свернутое произведение многомерных матриц

Операция транспонирования матрицы – это вспомогательная операция, используемая при разработке программного обеспечения и для удобного вывода результатов. Операция свертки позволяет производить дополнительные действия для

готовых маршрутов, позволяющие выбирать маршруты с заданными свойствами. В реальных задачах в роли типа X могут быть, такие алгебраические системы, например, множество $\{0, 1\}$ с аддитивной операцией дизъюнкции и мультипликативной операцией конъюнкции, $\Omega = \{\vee, \wedge\}$ при построении циклов в графе, или множество положительных действительных чисел R^+ с аддитивной операцией min и мультипликативной операцией сложения. Первая алгебраическая система позволяет решить задачу построения циклов, а вторая – поиск циклов с минимальной стоимостью.

Операции над матрицами, входящие в сигнатуру операций Ω , реализуются хорошо известными последовательными и параллельными стандартными алгоритмами, о которых речь пойдет далее. Практическая ценность предложенного подхода состоит в том, что суть операций над элементами структуры S не изменяется при изменении сути операций над элементами типа X . Это свойство абстрактных многомерно-матричных машин полезно в практическом программировании. Если типы X_p, \dots, X_n – гомоморфные или изоморфные универсальные алгебраические системы, то становится возможной отладка операций над структурой на наиболее простом типе данных. Отлаженная таким образом структура становится базовым классом, от которого можно породить конкретные объекты (реализации), предназначенные для решения задач на различных, в том числе и сложных, типах данных. Эти реальные многомерно-матричные машины, работающие с конкретными типами данных, разрабатываются как наследники абстрактных матричных машин, у которых они наследуют операции над структурами и включают в себя операции над реальными элементами этих структур.

Реляционная алгебра. Предложенный метод на основе алгебры многомерных матриц особенно эффективно работает в том случае, когда матрица смежности графа не разрежена, то есть содержит незначительное количество нейтральных элементов. В случае разреженных матриц эффективность существенно падает, так как вычислительная сложность остается без изменения. С другой стороны, многие методы организации и хранения разреженных матриц достаточно сложны и требуют серьезных временных затрат на доступ к элементам матриц, а распараллеливание операции умножения разреженных матриц достаточно трудоемко. Выход из положения возможен за счет применения технологии in database. Доказано, что алгебра многомерных матриц с элементами логического и числовых типов и операцией (λ, μ) -свернутого произведения и реляционная алгебра с отношениями со схемами $R(i_p, \dots, i_p, w)$ (i_p, \dots, i_p – атрибуты, домены которых положительные целые числа, w – атрибут с доменом логического или числового типа) и операцией Inner Join над этими отношениями **изоморфны** [15], [16].

В том случае когда матрица смежности графа разреженная, целесообразно представить ее в виде отношения со схемой $R(i_p, i_p, w)$. Тогда вычисление многомерной матрицы $(1, 0)G^{k+1} = (1, 0)G^k \times G$ реализуется запросом

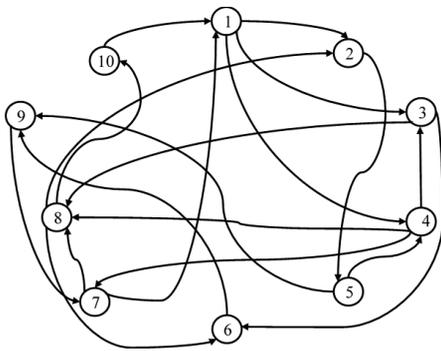
```
SELECT [Gi].[i1]A Si 1, ..., [Gi].[ik-1]A Si k, [Gi].[ik+1]A Si k+1, [Gi].[i2] AS ik+2, ([Gi].[w]@G.[w]) AS w
INTO Gk+1
FROM Gi Inner Join G On Gk+1 = Gi
WHERE ((([Gi].[i2] > [Gi].[i2]) And ... And (([Gi].[ik] > [Gi].[ik]) And (([Gi].[ik] > [Gi].[ik]))
```

Предикат, заданный в разделе WHERE обеспечивает отсутствие маршрутов, проходящих через одну вершину более одного раза.



Примеры построения циклов в графе

Пример 1 (алгебра многомерных матриц). Пусть дан ориентированный граф (рисунок 1), и его матрица смежности G (таблица 2). Требуется найти все возможные циклы в этом графе. Для решения поставленной задачи на этом 10-вершинном графе нужно построить реальную многомерно-матричную машину $EM = \langle M, \{0, 1\}; +, \wedge, \vee; \rangle$. Здесь структура – это множество M многомерных матриц, индексы которых принимают значения от 1 до 10, для которых определены операции сложения и $(1, 0)$ -свернутого произведения, а тип – это множество $\{0, 1\}$ с операциями дизъюнкции и конъюнкции. После выполнения операции $(1, 0)G \times G$ получается трехмерная матрица G^2 (таблица 3), которая содержит все маршруты, проходящие по двум ребрам. Эта таблица не содержит ни одного цикла, что соответствует рассматриваемому графу. В таблице 3 (и всех последующих таблицах) полностью приведены только содержащие единицы сечения этой матрицы.



Р и с. 1. Ориентированный граф с десятью вершинами
F i g. 1. Directed graph of ten vertices

Таблица 2. Матрица G смежности графа
Table 2. Graph adjacency matrix G

$\begin{matrix} \rightarrow i_2 \\ \downarrow i_1 \end{matrix}$	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0
3	0	0	0	0	0	1	0	1	0	0
4	0	0	1	0	0	0	1	1	0	0
5	0	0	0	4	0	0	0	0	1	0
6	0	0	0	0	0	0	0	0	1	0
7	1	0	0	0	0	0	0	1	0	0
8	0	1	0	0	0	1	0	0	0	1
9	0	0	0	0	0	0	1	0	0	0
10	1	0	0	0	0	0	0	0	0	0

В таблице 3 приведены циклы, которые построены в степенях матрицы смежности графа. В ее третьем столбце приведены примеры наборов значений индексов элементов $(1, 0)$ -свернутых степеней матрицы смежности графа, которые имеют значение 1. Следует отметить, что некоторые наборы значений индексов представляют собой циклические перестановки номеров вершин одного и того же цикла. Последняя степень $(0, 1)G^{10}$ содержит все циклические перестановки гамильтонова цикла.

Таблица 3. Циклы $(1, 0)$ -свернутых степеней матрицы смежности графа
Table 3. Cycles of $(1, 0)$ -folded powers of the adjacency matrix of a graph

Степень матрицы смежности	Количество циклов	Примеры наборов значений индексов
$(0, 1)G^2$	0	
$(0, 1)G^3$	3	(1,4,7,1); (4,7,1,4); (7,1,4,7)
$(0, 1)G^4$	16	(1,3,8,10,1); (4,8,2,5,4); (5,4,8,2,5); (7,8,6,9,7); (8,10,1,4,8); (10,1,3,8,10)
$(0, 1)G^5$	40	(1,4,3,8,10,1); (3,6,9,7,1,3); (4,7,1,2,5,4); (5,4,3,8,2,5); (7,1,2,5,9,7)
$(0, 1)G^6$	24	(1,4,3,6,9,7,1); (2,5,4,8,10,1,2); (6,9,7,1,3,8,6); (9,7,1,3,8,6,9); (10,1,2,5,4,8,10)
$(0, 1)G^7$	56	(1,4,3,8,6,9,7,1); (2,5,9,7,1,3,8,2); (5,4,7,8,10,1,2,5); (10,1,3,6,9,7,8,10)
$(0, 1)G^8$	40	(1,4,3,8,2,5,9,7,1); (3,6,9,7,8,10,1,4,3); (6,9,7,1,2,5,4,8,6); (9,7,8,10,1,4,3,6,9)
$(0, 1)G^9$	9	(2,5,4,3,8,6,9,7,1,2); (5,4,3,8,6,9,7,1,2,5); (8,6,9,7,1,2,5,4,3,8)
$(0, 1)G^{10}$	10	(1,2,5,4,3,6,9,7,8,10,1)



Пример 2 (реляционная алгебра). Матрице G ставится в соответствие таблица G и создается ее копия $G1$. После выполнения запроса

```
SELECT [G].[i1] AS i1, [G1].[i1] AS i2, [G1].[i2] AS i3, ([G].[w] OR
[G1].[w]) AS w INTO G2
FROM G INNER JOIN G1 ON G.[i2] = G1.[i1]
ORDER BY [G].[i1], [G1].[i1], [G1].[i2];
```

получается таблица $G2$, содержащая все маршруты, проходящие по двум ребрам. Эта таблица не содержит ни одного цикла поскольку в графе отсутствуют ребра, соединяющие две вершины в разных направлениях.

Следующий запрос

```
SELECT [G2].[i1] AS i1, [G2].[i2] AS i2, [G1].[i1] AS i3, [G1].[i2] AS
i4, ([G2].[w] OR [G1].[w]) AS w INTO G3
FROM G2 INNER JOIN G1 ON G2.i3 = G1.i1
WHERE ((([G2].[i2])<>[G1].[i2]))
ORDER BY [G2].[i1], [G2].[i2], [G1].[i1], [G1].[i2];
```

создает таблицу $G3$, содержащую все маршруты, проходящие по трем ребрам. Циклы, содержащиеся в этой $(0,1)G^3$ степени матрицы смежности представлены в таблице 4. На самом деле это цикл, проходящий через одни и те же вершины, но поскольку вершины, в которых приведенные в таблице циклы начинаются и заканчиваются различны, этим циклам соответствуют различные равные 1 элементы матрицы $(0,1)G^3$.

Таблица 4. Циклы, проходящие по трем ребрам
Table 4. Cycles passing through three edges

i1	i2	i3	i4	w
1	4	7	1	True
4	7	1	4	True
7	1	4	7	True

Запрос

```
SELECT [G3].[i1] AS i1, [G3].[i2] AS i2, [G3].[i3] AS i3, [G1].[i1] AS
i4, [G1].[i2] AS i5, ([G3].[w] OR [G1].[w]) AS w INTO G4
FROM G3 INNER JOIN G1 ON G3.i4 = G1.i1
WHERE ((([G3].[i2])<>[G1].[i2]) AND (([G3].[i3])<>[G1].[i2]) AND
(([G1].[i1])<>[G1].[i2]))
```

ORDER BY [G3].[i1], [G3].[i2], [G3].[i3], [G1].[i1], [G1].[i2];
создает таблицу $G4$, содержащую все маршруты, проходящие по четырем ребрам. Циклы, содержащиеся в этой $(0,1)G^4$ степени матрицы смежности представлены в таблице 5.

Таблица 5. Циклы, проходящие по четырем ребрам
Table 5. Cycles passing through four edges

i1	i2	i3	i4	i5	w
1	3	8	10	1	True
1	4	8	10	1	True
2	5	4	8	2	True
3	8	10	1	3	True
4	8	2	5	4	True
4	8	10	1	4	True
5	4	8	2	5	True
6	9	7	8	6	True
7	8	6	9	7	True

i1	i2	i3	i4	i5	w
8	2	5	4	8	True
8	6	9	7	8	True
8	10	1	3	8	True
8	10	1	4	8	True
9	7	8	6	9	True
10	1	3	8	10	True
10	1	4	8	10	True

Запросы, вычисляющие $(0,1)G^5 - (0,1)G^8$ (1, 0)-свернутые степень матрицы смежности графа, в статье не приводятся, так как содержат большое количество циклов.

Запрос

```
SELECT [G8].[i1] AS i1, [G8].[i2] AS i2, [G8].[i3] AS i3, [G8].[i4] AS
i4, [G8].[i5] AS i5, [G8].[i6] AS i6, [G8].[i7] AS i7, [G8].[i8] AS i8, [G1].
[i1] AS i9, [G1].[i2] AS i10, ([G8].[w]+[G1].[w]) AS w INTO G9
FROM G8 INNER JOIN G1 ON G8.i9 = G1.i1
WHERE ((([G8.i2]<>G1.i2) And (([G8.i3]<>G1.i2) And (([G8.i4]<>G1.
i2) And (([G8.i5]<>G1.i2) And (([G8.i6]<>G1.i2) And (([G8.i7]<>G1.
i2) And (([G8.i8]<>G1.i2) And (([G1.i1]<>G1.i2))
```

ORDER BY [G8].[i1], [G8].[i2], [G8].[i3], [G8].[i4], [G8].[i5], [G8].[i6], [G8].[i7], [G8].[i8], [G1].[i1], [G1].[i2];

создает таблицу $G9$, содержащую все маршруты, проходящие по девяти ребрам. Циклы, содержащиеся в этой $(0,1)G^9$ степени матрицы смежности представлены в таблице 6.

Таблица 6. Циклы, проходящие по девяти ребрам
Table 6. Cycles passing through nine edges

i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	w
1	2	5	4	3	8	6	9	7	1	True
2	5	4	3	8	6	9	7	1	2	True
3	8	6	9	7	1	2	5	4	3	True
4	3	8	6	9	7	1	2	5	4	True
5	4	3	8	6	9	7	1	2	5	True
6	9	7	1	2	5	4	3	8	6	True
7	1	2	5	4	3	8	6	9	7	True
8	6	9	7	1	2	5	4	3	8	True
9	7	1	2	5	4	3	8	6	9	True

Последний выполняемый запрос

```
SELECT [G9].[i1] AS i1, [G9].[i2] AS i2, [G9].[i3] AS i3, [G9].[i4] AS
i4, [G9].[i5] AS i5, [G9].[i6] AS i6, [G9].[i7] AS i7, [G9].[i8] AS i8, [G9].
[i9] AS i9, [G1].[i1] AS i10, [G1].[i2] AS i11, ([G9].[w]+[G1].[w]) AS
w INTO G10
FROM G9 INNER JOIN G1 ON G9.i10 = G1.i1
WHERE ((([G9.i2]<>G1.i2) And (([G9.i3]<>G1.i2) And (([G9.i4]<>G1.
i2) And (([G9.i5]<>G1.i2) And (([G9.i6]<>G1.i2) And (([G9.i7]<>G1.
i2) And (([G9.i8]<>G1.i2) And (([G9.i9]<>G1.i2) And (([G1.i1]<>G1.
i2))
```

ORDER BY [G9].[i1], [G9].[i2], [G9].[i3], [G9].[i4], [G9].[i5], [G9].[i6], [G9].[i7], [G9].[i8], [G9].[i9], [G1].[i1], [G1].[i2];

создает таблицу $G10$, содержащую все маршруты, проходящие по десяти ребрам. Циклы, содержащиеся в этой $(0,1)G^{10}$ степени матрицы смежности представлены в таблице 7. На самом деле это единственный гамильтонов цикл рассматриваемого графа.



Таблица 7. Циклы, проходящие по десяти ребрам
Table 7. Cycles passing through ten edges

i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11	w
1	2	5	4	3	6	9	7	8	10	1	True
2	5	4	3	6	9	7	8	10	1	2	True
3	6	9	7	8	10	1	2	5	4	3	True
4	3	6	9	7	8	10	1	2	5	4	True
5	4	3	6	9	7	8	10	1	2	5	True
6	9	7	8	10	1	2	5	4	3	6	True
7	8	10	1	2	5	4	3	6	9	7	True
8	10	1	2	5	4	3	6	9	7	8	True
9	7	8	10	1	2	5	4	3	6	9	True
10	1	2	5	4	3	6	9	7	8	10	True

Параллельная реализация алгоритмов построения циклов в графе

Известно, что один из способов повышения эффективности вычислительных средств при обработке больших данных алгоритмами с высокой вычислительной сложностью заключается в распараллеливании алгоритмов [17], [18].

– Применение многомерно-матричного подхода позволяет использовать хорошо известные алгоритмы параллельного блочного умножения матриц. В [19] приведено обобщение

алгоритма Кэннона для (Представленные в виде отношений матрица смежности графа и последующие ее (1, 0)-свернутые степени требуют значительно меньше памяти, чем соответствующие им разреженные матрицы;

– в большинстве современных СУБД операция Inner Join реализуется параллельными алгоритмами;

– в случае, когда объемы отношений велики, возможно применение принципа симметричного горизонтального распределения, что увеличивает степень параллелизма операции Inner Join [21].

Заключение

Из сказанного можно сделать следующие выводы:

1. Применение предложенного подхода существенно упрощает решение задач построения циклов, поскольку позволяет разрабатывать простые и удобные для понимания и программирования алгоритмы.
2. Простота алгоритмов, в свою очередь, обеспечивает возможность достаточно легкого распараллеливания на основе применения хорошо известных и проверенных методов.
3. Алгебраический подход обеспечивает эффективное сочетание технологий in memory и in database, что позволяет проектировать программно-аппаратные комплексы, наилучшим образом приспособленные для решения задач построения маршрутов.

Список использованных источников

- [1] Bast H. Car or Public Transport – Two Worlds // Efficient Algorithms. Lecture Notes in Computer Science; S. Albers, H. Alt, S. Näher (eds.). Vol. 5760. Springer, Berlin, Heidelberg, 2009. P. 355-367. doi: https://doi.org/10.1007/978-3-642-03456-5_24
- [2] Борознов В. О. Исследование решения задачи коммивояжера // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2009. № 2. С. 147-151. URL: <https://elibrary.ru/item.asp?id=12929954> (дата обращения: 20.08.2021).
- [3] Ozden S. G., Smith A. E., Gue K. R. Solving large batches of traveling salesman problems with parallel and distributed computing // Computers & Operations Research. 2017. Vol. 85. P. 87-96. doi: <https://doi.org/10.1016/j.cor.2017.04.001>
- [4] Ageyev D., Ignatenko A., Wehbe F. Design of Information and Telecommunication Systems with the Usage of the Multi-Layer Graph Model // Proceedings of the XIIth International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM). Lviv-Polyana, Ukraine, 2013. P. 1-4. doi: <https://doi.org/10.48550/arXiv.1307.1730>
- [5] Марголис Б. И., Музанна М. М. Синтез магистральных телекоммуникационных сетей // Программные продукты и системы. 2014. № 1(105). С. 162-168. URL: <https://elibrary.ru/item.asp?id=22807997> (дата обращения: 20.08.2021).
- [6] Спивак С. И., Исмагилова А. С., Хамитова И. А. Теоретико-графовый метод определения маршрутов сложных химических реакций // Доклады Академии наук. 2010. Т. 434, №. 4. С. 499-501. URL: <https://elibrary.ru/item.asp?id=15524859> (дата обращения: 20.08.2021).
- [7] Меламед И. И., Сергеев С. И., Сигал И. Х. Задача коммивояжера. Точные методы // Автоматика и телемеханика. 1989. № 10. С. 3-29. URL: <http://mi.mathnet.ru/at6433> (дата обращения: 20.08.2021).
- [8] Solving the Vehicle Routing Problem using Genetic Algorithm / K. Masum [и др.] // International Journal of Advanced Computer Science and Applications. 2011. Vol. 2, no. 7. P. 126-131. doi: <http://dx.doi.org/10.14569/IJACSA.2011.020719>
- [9] Research Directions for Big Data Graph Analytics / J. A. Miller [и др.] // 2015 IEEE International Congress on Big Data. New York, NY, USA: IEEE Press, 2015. P. 785-794. doi: <https://doi.org/10.1109/BigDataCongress.2015.132>
- [10] Real-time constrained cycle detection in large dynamic graphs / X. Qiu [и др.] // Proceedings of the VLDB Endowment. 2018. Vol. 11, no. 12. P. 1876-1888. doi: <https://doi.org/10.14778/3229863.3229874>
- [11] Прокопенков В. Ф. Полиномиальный алгоритм поиска гамильтонова цикла на графе // Вестник Национального технического университета «ХПИ». Серия: Стратегическое управление, управление портфелями, программами и проектами. 2021. № 1(3). С. 55-65. doi: <https://doi.org/10.20998/2413-3000.2021.3.8>
- [12] Мунерман В. И., Самойлова Т. А. Алгебраический подход к алгоритмизации задач маршрутизации // Системы высокой доступности. 2018. Т. 14, № 5. С. 50-56. doi: <https://doi.org/10.18127/j20729472-201805-08>



- [13] Grigoryeva G., Khodchenkov V. Implementation of Operations of a Quantum Computer in the Language of the Algebra of Multidimensional Matrices // 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus). Press, St. Petersburg, Moscow, Russia: IEEE Press, 2021. P. 2181-2184. doi: <https://doi.org/10.1109/ElConRus51938.2021.9396119>
- [14] Мунерман В. И. Архитектура программно-аппаратного комплекса для массовой обработки данных на базе многомерно-матричной модели // Системы высокой доступности. 2015. Т. 11, № 2. С. 13-18. URL: <https://www.elibrary.ru/item.asp?id=23819273> (дата обращения: 20.08.2021).
- [15] Мунерман В. И., Мунерман Д. В. О соответствии моделей данных и моделей вычислений // Системы компьютерной математики и их приложения. 2021. № 22. С. 146-152. URL: <https://www.elibrary.ru/item.asp?id=46649891> (дата обращения: 20.08.2021).
- [16] Объектно-ориентированный подход к разработке моделей данных / Е. П. Емельченков // Современные информационные технологии и ИТ-образование. 2020. Т. 16, № 3. С. 564-574. doi: <https://doi.org/10.25559/SITITO.16.202003.564-574>
- [17] Трещев И. А. Построение многопоточных приложений для распараллеливания алгоритмов перебора // Информатика и системы управления. 2008. № 1(15). С. 151-159. URL: <https://www.elibrary.ru/item.asp?id=10366217> (дата обращения: 20.08.2021).
- [18] Developing parallel programming and soft skills: A project based learning approach / A. A. Younis [и др.] // Journal of Parallel and Distributed Computing. 2021. Vol. 158. P. 151-163. doi: <https://doi.org/10.1016/j.jpdc.2021.07.015>
- [19] Захаров В. Н., Мунерман В. И. Параллельный алгоритм умножения многомерных матриц // Современные информационные технологии и ИТ-образование. 2015. Т. 11, № 2. С. 384-390. URL: <https://www.elibrary.ru/item.asp?id=26167519> (дата обращения: 20.08.2021).
- [20] Ильин П. Л., Мунерман В. И. Рекурсивное вычисление детерминанта многомерной матрицы // Системы компьютерной математики и их приложения. 2019. № 20-1. С. 162-167. URL: <https://www.elibrary.ru/item.asp?id=39103176> (дата обращения: 20.08.2021).
- [21] Munerman V., Munerman D., Samoiloва T. The Heuristic Algorithm For Symmetric Horizontal Data Distribution // 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus). St. Petersburg, Moscow, Russia: IEEE Press, 2021. P. 2161-2165. doi: <https://doi.org/10.1109/ElConRus51938.2021.9396510>
- [22] Genetic algorithm based test data generation for MPI parallel programs with blocking communication / T. Tian [и др.] // Journal of Systems and Software. 2019. Vol. 155, no. C. P. 130-144. doi: <https://doi.org/10.1016/j.jss.2019.04.049>
- [23] A novel cooperative navigation algorithm based on factor graph with cycles for AUVs / Y. Ben [и др.] // Ocean Engineering. 2021. Vol. 241. Article number: 110024. doi: <https://doi.org/10.1016/j.oceaneng.2021.110024>
- [24] Efficient Partitioning Algorithm for Parallel Multidimensional Matrix Operations by Linearization / K. S. Alam [и др.] // Information and Communication Technology for Intelligent Systems. ICTIS 2020. Smart Innovation, Systems and Technologies; T. Senjyu, P. N. Mahalle, T. Perumal, A. Joshi (eds.). Vol. 195. Springer, Singapore, 2021. P. 141-149. doi: https://doi.org/10.1007/978-981-15-7078-0_13
- [25] Sokolov N. P. Functions of multidimensional matrices and their applications for the solutions of linear systems of partial differential equations // Ukrainian Mathematical Journal. 1970. Vol. 22, issue 6. P. 657-674. doi: <https://doi.org/10.1007/BF01086271>

Поступила 20.08.2021; одобрена после рецензирования 05.10.2021; принята к публикации 15.11.2021.

Об авторах:

Емельченков Евгений Петрович, заведующий кафедрой информатики физико-математического факультета, ФГБОУ ВО «Смоленский государственный университет» (214000, Российская Федерация, г. Смоленск, ул. Пржевальского, д. 4), кандидат физико-математических наук, доцент, ORCID: <https://orcid.org/0000-0002-6589-694X>, uru1101@gmail.com

Мунерман Виктор Иосифович, доцент кафедры информатики физико-математического факультета, ФГБОУ ВО «Смоленский государственный университет» (214000, Российская Федерация, г. Смоленск, ул. Пржевальского, д. 4), кандидат технических наук, доцент, ORCID: <https://orcid.org/0000-0002-9628-4049>, vimoona@gmail.com

Мунерман Даниил Викторович, лаборант-стажер кафедры информатики физико-математического факультета, ФГБОУ ВО «Смоленский государственный университет» (214000, Российская Федерация, г. Смоленск, ул. Пржевальского, д. 4), ORCID: <https://orcid.org/0000-0002-5139-6645>, danvmoon@gmail.com

Самойлова Татьяна Аркадьевна, доцент кафедры информатики физико-математического факультета, ФГБОУ ВО «Смоленский государственный университет» (214000, Российская Федерация, г. Смоленск, ул. Пржевальского, д. 4), кандидат технических наук, доцент, ORCID: <https://orcid.org/0000-0002-3712-327X>, tatsamoilova24@gmail.com

Все авторы прочитали и одобрили окончательный вариант рукописи.



References

- [1] Bast H. Car or Public Transport – Two Worlds. In: Albers S., Alt H., Näher S. (eds.). *Efficient Algorithms. Lecture Notes in Computer Science*. Vol. 5760. Springer, Berlin, Heidelberg; 2009. p. 355-367. (In Eng.) doi: https://doi.org/10.1007/978-3-642-03456-5_24
- [2] Boroznov V.O. Research of the task solution of the traveling salesman. *Vestnik of Astrakhan State Technical University. Series: Management, Computer Sciences and Informatics*. 2009; (2):147-151. Available at: <https://elibrary.ru/item.asp?id=12929954> (accessed 20.08.2021). (In Russ., abstract in Eng.)
- [3] Ozden S.G., Smith A.E., Gue K.R. Solving large batches of traveling salesman problems with parallel and distributed computing. *Computers & Operations Research*. 2017; 85:87-96. (In Eng.) doi: <https://doi.org/10.1016/j.cor.2017.04.001>
- [4] Ageyev D., Ignatenko A., Wehbe F. Design of Information and Telecommunication Systems with the Usage of the Multi-Layer Graph Model. *Proceedings of the XIIth International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*. Lviv-Polyana, Ukraine; 2013. p. 1-4. (In Eng.) doi: <https://doi.org/10.48550/arXiv.1307.1730>
- [5] Margolis B.I., Muzanna M.M. Synthesis gas telemedicine networks. *Software & Systems*. 2014; (1):162-168. Available at: <https://elibrary.ru/item.asp?id=22807997> (accessed 20.08.2021). (In Russ., abstract in Eng.)
- [6] Spivak S.I., Ismagilova A.S., Khamitova I.A. Graph-theoretical method for determining routes of complex chemical reactions. *Doklady Physical Chemistry*. 2010; 434:169-171. (In Eng.) doi: <https://doi.org/10.1134/S0012501610100040>
- [7] Melamed I.I., Sergeev S.I., Sigal I.Kh. The traveling salesman's problem. Exact methods. *Avtomatika i Telemekhanika = Automation and Remote Control*. 1989; (10):3-29. Available at: <http://mi.mathnet.ru/at6433> (accessed 20.08.2021). (In Russ., abstract in Eng.)
- [8] Masum A.K.M., Shahjalal M., Faruque F., Sarker I.H. Solving the Vehicle Routing Problem using Genetic Algorithm. *International Journal of Advanced Computer Science and Applications*. 2011; 2(7):126-131. (In Eng.) doi: <http://dx.doi.org/10.14569/IJACSA.2011.020719>
- [9] Miller J.A., Ramaswamy L., Kochut K.J., Fard A. Research Directions for Big Data Graph Analytics. *2015 IEEE International Congress on Big Data*. IEEE Press, New York, NY, USA; 2015. p. 785-794. (In Eng.) doi: <https://doi.org/10.1109/BigDataCongress.2015.132>
- [10] Qiu X., Cen W., Qian Z., Peng Y., Zhang Y., Lin X., Zhou J. Real-time constrained cycle detection in large dynamic graphs. *Proceedings of the VLDB Endowment*. 2018; 11(12):1876-1888. (In Eng.) doi: <https://doi.org/10.14778/3229863.3229874>
- [11] Prokopenkov V. Polynomial algorithm for finding a Hamiltonian cycle on a graph. *Вісник Національного технічного університету «ХПІ». Серія: Стратегічне управління, управління портфелями, програмами та проектами = Bulletin of the National Technical University "KhPI". Series: Strategic management, portfolio, program and project management*. 2021; (1):55-65. (In Russ., abstract in Eng.) doi: <https://doi.org/10.20998/2413-3000.2021.3.8>
- [12] Munerman V.I., Samoylova T.A. Algebraic approach to algorithmization of routing problems. *Highly available systems*. 2018; 14(5):50-56. (In Russ., abstract in Eng.) doi: <https://doi.org/10.18127/j20729472-201805-08>
- [13] Grigoryeva G., Khodchenkov V. Implementation of Operations of a Quantum Computer in the Language of the Algebra of Multidimensional Matrices. *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*. IEEE Press, St. Petersburg, Moscow, Russia; 2021. p. 2181-2184. (In Eng.) doi: <https://doi.org/10.1109/ElConRus51938.2021.9396119>
- [14] Munerman V.I. Construction of hardware-software complexes architecture to improve massively data processing. *Highly available systems*. 2015; 11(2):13-18. Available at: <https://www.elibrary.ru/item.asp?id=23819273> (accessed 20.08.2021). (In Russ., abstract in Eng.)
- [15] Munerman V.I., Munerman D.V. About the correspondence of data models and calculation models. *Sistemy komp'yuternoj matematiki i ih prilozheniya = Computer Mathematics Systems and Their Applications*. 2021; (22):146-152. Available at: <https://www.elibrary.ru/item.asp?id=46649891> (accessed 20.08.2021). (In Russ., abstract in Eng.)
- [16] Emelchenkov E.P., Munerman V.I., Munerman D.V., Samoilova T.A. The object oriented approach to designing data models. *Sovremennye informacionnye tehnologii i IT-obrazovanie = Modern Information Technologies and IT-Education*. 2020; 16(3):564-574. (In Russ., abstract in Eng.) doi: <https://doi.org/10.25559/SITITO.16.202003.564-574>
- [17] Treschev I.A. Construction of multiflow supplements for unparallelizing of excess algorithms. *Informatika i sistemy upravleniya = Information Science and Control Systems'*. 2008; (1):151-159. Available at: <https://www.elibrary.ru/item.asp?id=10366217> (accessed 20.08.2021). (In Russ.)
- [18] Younis A.A., Sunderraman R., Metzler M., Bourgeois A.G. Developing parallel programming and soft skills: A project based learning approach. *Journal of Parallel and Distributed Computing*. 2021; 158:151-163. (In Eng.) doi: <https://doi.org/10.1016/j.jpdc.2021.07.015>
- [19] Zakharov V.N., Munerman V.I. *Parallel'nyj algoritm umnozhenija mnogomernyh matric* [Parallel algorithm for multiplying multidimensional matrices]. *Sovremennye informacionnye tehnologii i IT-obrazovanie = Modern Information Technologies and IT-Education*. 2015; 11(2):384-390. Available at: <https://www.elibrary.ru/item.asp?id=26167519> (accessed 20.08.2021). (In Russ.)
- [20] Iljin P.L., Munerman V.I. Recursive computation of the multidimensional matrix determinant. *Sistemy komp'yuternoj matematiki i ih prilozheniya = Computer Mathematics Systems and Their Applications*. 2019; (20-1):162-167. Available at: <https://www.elibrary.ru/item.asp?id=39103176> (accessed 20.08.2021). (In Russ., abstract in Eng.)
- [21] Munerman V., Munerman D., Samoilova T. The Heuristic Algorithm For Symmetric Horizontal Data Distribution. *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*. IEEE Press, St. Petersburg, Moscow, 2021. p. 2161-2165. (In Eng.) doi: <https://doi.org/10.1109/ElConRus51938.2021.9396510>



- [22] Tian T., Gong D., Kuo F.-C., Liu H. Genetic algorithm based test data generation for MPI parallel programs with blocking communication. *Journal of Systems and Software*. 2019; 155(C):130-144. (In Eng.) doi: <https://doi.org/10.1016/j.jss.2019.04.049>
- [23] Ben Y., Sun Y., Li Q., Zang X. A novel cooperative navigation algorithm based on factor graph with cycles for AUVs. *Ocean Engineering*. 2021; 241:110024. (In Eng.) doi: <https://doi.org/10.1016/j.oceaneng.2021.110024>
- [24] Alam K.S., Shishir T.A., Azharul Hasan K.M. Efficient Partitioning Algorithm for Parallel Multidimensional Matrix Operations by Linearization. In: Senjyu T., Mahalle P.N., Perumal T., Joshi A. (eds.) *Information and Communication Technology for Intelligent Systems. ICTIS 2020. Smart Innovation, Systems and Technologies*. Vol. 195. Springer, Singapore; 2021. p. 141-149. (In Eng.) doi: https://doi.org/10.1007/978-981-15-7078-0_13
- [25] Sokolov N.P. Functions of multidimensional matrices and their applications for the solutions of linear systems of partial differential equations. *Ukrainian Mathematical Journal*. 1970; 22(6):657-674. (In Eng.) doi: <https://doi.org/10.1007/BF01086271>

Submitted 20.08.2021; approved after reviewing 05.10.2021; accepted for publication 15.11.2021.

About the authors:

Evgeny P. Yemelchenkov, Head of the Chair of Computer Science, Faculty of Physics and Mathematics, Smolensk State University (4 Przhevalsky St., Smolensk 214000, Russian Federation), Cand.Sci. (Phys.-Math.), Associate Professor, **ORCID: <https://orcid.org/0000-0002-6589-694X>**, ypy1101@gmail.com

Victor I. Munerman, Associate Professor of the Chair of Computer Science, Faculty of Physics and Mathematics, Smolensk State University (4 Przhevalsky St., Smolensk 214000, Russian Federation), Cand.Sci. (Eng.), Associate Professor, **ORCID: <https://orcid.org/0000-0002-9628-4049>**, vimoon@gmail.com

Daniel V. Munerman, Laboratory Assistant of the Chair of Computer Science, Faculty of Physics and Mathematics, Smolensk State University (4 Przhevalsky St., Smolensk 214000, Russian Federation), **ORCID: <https://orcid.org/0000-0002-5139-6645>**, danvmoon@gmail.com

Tatyana A. Samoylova, Associate Professor of the Chair of Computer Science, Faculty of Physics and Mathematics, Smolensk State University (4 Przhevalsky St., Smolensk 214000, Russian Federation), Cand.Sci. (Eng.), Associate Professor, **ORCID: <https://orcid.org/0000-0002-3712-327X>**, tatsamoilova24@gmail.com

All authors have read and approved the final manuscript.

