

## ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ НАХОЖДЕНИЯ ДОСТИЖИМОСТЕЙ В ГРАФЕ

Князькова А.В., Волченская Т.В., Князьков В.С.

ФГБОУ ВПО «Вятский государственный университет», Киров, e-mail: kniazkov@list.ru

Приводятся области применения задач на графах и сетях, в которых используется понятие достижимости. Операция построения матрицы достижимости является ключевой в задаче разбиения графа на максимальные сильно связанные подграфы, в задаче построения кратчайших путей и в ряде других задач. Рассматриваются способы представления графовых моделей для компьютерной обработки. Приводится понятие достижимости и представление его через многозначные отображения и транзитивные замыкания. Рассматривается параллельный алгоритм нахождения матрицы достижимости, главной особенностью которого является то, что операция построения матрицы достижимостей может быть выполнена параллельно для всех вершин графа. Количество итераций зависит не от размерности графа, а от его структуры, а именно от длины простых цепей и контуров. В реальных сетевых моделях, где велика степень связности графа, скорость построения матрицы еще более возрастет.

**Ключевые слова:** граф, графовая модель, достижимость в графах, матрица достижимости, параллельный алгоритм

## PARALLEL ALGORITHM OF SEARCH ACCESSIBILITY IN THE GRAPH

Knyazkova A.V., Volchenskaya T.V., Knyazkov V.S.

Vyatka State University, Kirov, e-mail: kniazkov@list.ru

There are problems in the field of graph and network tasks in which the concept of reachability is used. The operation of reachability matrix construction is the key point in the problem of graph splitting up into maximal coherent subgraphs in a shortest construction problem and in a number of other problems. Many ways of performance graph models for computer processing are considered. The concept of reachability and its performance through multiple-valued displays and transitive short circuits is presented. The parallel algorithm is used to create the reachability matrix to be considered. The main characteristic is that construction of a reachability matrix operation can be executed in parallel for all graph nodes. The quantity of iterations depends on its structure, not the graph dimensions. It depends on simple circuits and paths. In real network models where the degree of connectivity in the graph is large, the speed of the construction of a matrix will increase.

**Keywords:** graph, graph model, accessibility in the graph, a matrix of the accessibility, parallel algorithm

Сетевые и графовые модели охватывают довольно широкий класс задач, встречающихся при проектировании систем, планировании работ, распределении продукции, организации транспортных перевозок, размещении различных центров обслуживания и т.п.

Задач, в которых используется понятие достижимости, довольно много. Вот одна из них. Граф может быть моделью какой-то организации или социальной сети, в которой люди представлены вершинами, а дуги интерпретируют каналы связи. При рассмотрении такой модели можно поставить вопрос, может ли информация от одного лица  $x_i$  быть передана другому лицу  $x_j$ , т.е. существует ли путь, идущий от вершины  $x_i$  к вершине  $x_j$ . Если такой путь существует, то говорят, что вершина  $x_j$  достижима из вершины  $x_i$ . Можно интересоваться достижимостью вершины  $x_j$  из вершины  $x_i$  только на таких путях, длины которых не превосходят заданной величины или длина которых меньше наибольшего числа вершин в графе и т.п. задачи.

Характерной особенностью таких задач является большая размерность, что обуслов-

ливает необходимость конструирования более скоростных алгоритмов. Плодотворной основой для построения таких алгоритмов могут служить их сетевые и графовые формализации [1, 7].

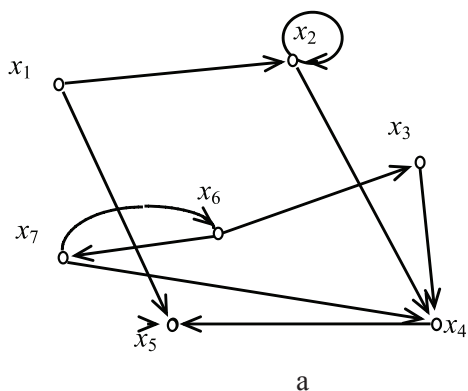
Программные реализации указанных задач весьма универсальны и просто программируемы при небольших размерностях графа. Однако при значительном увеличении количества узлов или дуг графа время программного решения при последовательном исполнении инструкций становится практически неприемлемым. Аппаратные реализации хотя и обеспечивают необходимую скорость за счет параллельной обработки информации, но не обладают достаточной алгоритмической гибкостью, в силу этого – их адаптация к изменениям логики вычислений не всегда возможна [2, 3, 4, 5]. Решением данной проблемы является использование таких вычислительных алгоритмов, которые бы совмещали скорость аппаратных вычислений с легкостью программной модификации в параллельных алгоритмах [6, 8, 9].

В рассматриваемых алгоритмах графы заданы с помощью матрицы смежности,

которая дает его компактное представление, особенно если есть возможность работать с двоичными битами в машинном слове. Анализ алгоритмов решения оптимизационных задач на графах приводит к выводу, что большинство операций на графах могут выполняться параллельно для каждой из

вершин, что позволяет значительно ускорить обработку данных.

Допустим, что мы имеем граф, структура которого описана матрицей смежности (рис. 1) размерностью  $n \times n$ , (где  $n$  – число вершин графа), однозначно представляющая его структуру:



	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
$x_1$	1	1	0	0	1	0	0
$x_2$	0	1	0	1	0	0	0
$x_3$	0	0	1	1	0	0	0
$x_4$	0	0	0	1	1	0	0
$x_5$	0	0	0	0	1	0	0
$x_6$	0	0	1	0	0	1	1
$x_7$	0	0	0	1	0	1	1

Рис. 1. Пример 1:  
а – граф; б – матрица смежности графа

$A = \{a_{ij}\}$ ,  $i, j = 1, 2, \dots, n$ , а каждый элемент матрицы определяется следующим образом:

$a_{ij} = 1$ , если существует дуга из вершины  $x_i$  в вершину  $x_j$ ,

$a_{ij} = 0$ , если нет дуги из вершины  $x_i$  в вершину  $x_j$ .

Достижимость в графе описывается матрицей достижимости

$R = [r_{ij}]$ ,  $i, j = 1, 2, \dots, n$ , где  $n$  – число вершин графа, а каждый элемент определяется следующим образом:

$r_{ij} = 1$ , если вершина  $x_j$  достижима из  $x_i$ ,  
 $r_{ij} = 0$ , в противном случае.

Множество вершин  $R(x_i)$  графа  $G$ , достижимых из заданной вершины  $x_i$ , состоит из таких элементов  $x_j$ , для которых  $(i, j)$ -й

элемент в матрице достижимостей равен 1. Очевидно, что все диагональные элементы в матрице  $R$  равны 1, поскольку каждая вершина достижима из себя самой путем длины 0. Поскольку прямое отображение 1-го порядка  $\Gamma^{+1}(x_i)$  является множеством таких вершин  $x_j$ , которые достижимы из  $x_i$  с использованием путей длины 1, то множество  $\Gamma^{+}(\Gamma^{+1}(x_i)) = \Gamma^{+2}(x_i)$  состоит из вершин, достижимых из  $x_i$  с использованием путей длины 2. Аналогично  $\Gamma^{+p}(x_i)$  является множеством вершин, которые достижимы из  $x_i$  с помощью путей длины  $p$ .

Так как любая вершина графа, которая достижима из  $x_i$ , должна быть достижима с использованием пути (или путей) длины 0 или 1, или 2, ..., или  $p$ , то множество вершин, достижимых для вершины  $x_i$ , можно представить в виде

$$R(x_i) = \{x_i\} \cup \Gamma^{+1}(x_i) \cup \Gamma^{+2}(x_i) \cup \dots \cup \Gamma^{+p}(x_i).$$

Как видим, множество достижимых вершин  $R(x_i)$  представляет собой прямое транзитивное замыкание вершины  $x_i$ , т.е.  $R(x_i) = T^{+}(x_i)$ . Следовательно, для построения матрицы достижимости находим достижимые множества  $R(x_i)$  для всех вершин  $x_i \in X$ . Полагая  $r_{ij} = 1$ , если  $x_j \in R(x_i)$  и  $r_{ij} = 0$  в противном случае.

Суть предлагаемого параллельного алгоритма нахождения матрицы достижимости заключается в следующем:

1. Так как каждая вершина достижима сама для себя, то в матрице достижимости  $R^1$  на главной диагонали должны стоять 1. Единицы можно предварительно занести на главную диагональ матрицы смежности, для того чтобы в п. 2 алгоритма использовать общую формулу преобразования.

Матрица R <sup>1</sup>							
	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	x <sub>7</sub>
x <sub>1</sub>	1	1	0	1	1	0	0
x <sub>2</sub>	0	1	0	1	1	0	0
x <sub>3</sub>	0	0	1	1	1	0	0
x <sub>4</sub>	0	0	0	1	1	0	0
x <sub>5</sub>	0	0	0	0	1	0	0
x <sub>6</sub>	0	0	1	1	1	1	1
x <sub>7</sub>	0	0	1	1	1	1	1

Матрица R <sup>2</sup>							
	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	x <sub>7</sub>
x <sub>1</sub>	1	1	0	1	1	0	0
x <sub>2</sub>	0	1	0	1	1	0	0
x <sub>3</sub>	0	0	1	1	1	0	0
x <sub>4</sub>	0	0	0	1	1	0	0
x <sub>5</sub>	0	0	0	0	1	0	0
x <sub>6</sub>	0	0	1	1	1	1	1
x <sub>7</sub>	0	0	1	1	1	1	1

2. По матрице смежности для каждой вершины графа  $x_i$  находим строку в матрице  $R^1$  как результат логического сложения тех ( $j-x$ ) строк матрицы смежности, для которых  $x_{ij} = 1$

$$\forall R_i^1 = A_{i1} \times (A_{11}, A_{12}, A_{13}, \dots, A_{1n}) \vee$$

$$\vee A_{i2} \times (A_{21}, A_{22}, A_{23}, \dots, A_{2n}) \vee \dots \vee$$

$$\vee A_{in} \times (A_{n1}, A_{n2}, A_{n3}, \dots, A_{nm}).$$

Например, для вершины  $x_1$  в матрице смежности элементы  $a_{11}$ ,  $a_{12}$  и  $a_{15}$  равны 1, следовательно, строка в матрице  $R^1$  для вершины  $x_1$  будет получена как результат логического сложения элементов соответствующих позиций первой, второй и пятой строк матрицы смежности.

Операция построения матрицы достижимостей  $R^1$  может быть выполнена параллельно для всех вершин графа.

3. По матрице  $R^1$  для каждой вершины графа находим матрицу следующего уровня  $R^2$  по аналогичным формулам:

$$\forall R_i^2 = R_{i1}^1 \times (R_{11}^1, R_{12}^1, R_{13}^1, \dots, R_{1n}^1) \vee$$

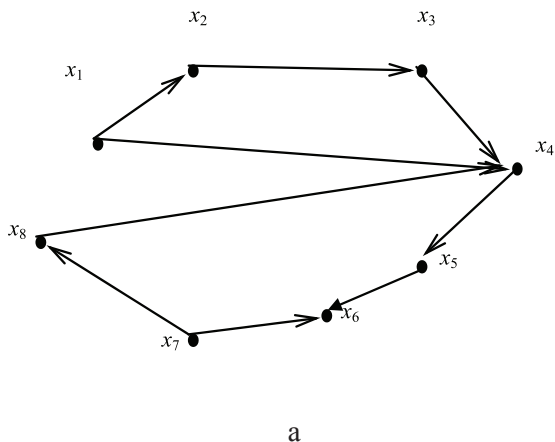
$$\vee R_{i2}^1 \times (R_{21}^1, R_{22}^1, R_{23}^1, \dots, R_{2n}^1) \vee \dots \vee$$

$$\vee R_{in}^1 \times (R_{n1}^1, R_{n2}^1, R_{n3}^1, \dots, R_{nm}^1).$$

4. Далее процедура построения матрицы достижимости  $R^k$  по матрице достижимости предыдущего уровня  $R^{k-1}$  повторяется до тех пор, пока  $R^k$  не будет равной  $R^{k-1}$ .

В нашем примере на рис. 1 построение матрицы достижимости произошло за одну итерацию, так как  $R^2 = R^1$ .

Количество итераций зависит от структуры графа, а именно от длины простых цепей и контуров. Рассмотрим второй пример, показанный на рис. 2. Как видим из примера, матрица достижимости строится за три итерации, то есть только  $R^4 = R^3$ . Таким образом, количество итераций равно длине самой длинной простой цепи и не зависит от количества вершин графа. В реальных сетевых моделях, где велика степень связности графа, скорость построения матрицы еще более возрастет.



	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	x <sub>7</sub>	x <sub>8</sub>
x <sub>1</sub>	1	1	0	1	0	0	0	0
x <sub>2</sub>	0	1	1	0	0	0	0	0
x <sub>3</sub>	0	0	1	1	0	0	0	0
x <sub>4</sub>	0	0	0	1	1	0	0	0
x <sub>5</sub>	0	0	0	0	1	1	0	0
x <sub>6</sub>	0	0	0	0	0	1	0	0
x <sub>7</sub>	0	0	0	0	0	0	1	1
X	0	0	0	1	0	0	0	1

Рис. 2. Пример 2:  
а – граф; б – матрица смежности графа

Матрица R<sup>1</sup>

	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	x <sub>7</sub>	x <sub>8</sub>
x <sub>1</sub>	1	1	1	1	1	0	0	0
x <sub>2</sub>	0	1	1	1	0	0	0	0
x <sub>3</sub>	0	0	1	1	1	0	0	0
x <sub>4</sub>	0	0	0	1	1	1	0	0
x <sub>5</sub>	0	0	0	0	1	1	0	1
x <sub>6</sub>	0	0	0	0	0	1	0	1
x <sub>7</sub>	0	0	0	0	0	1	1	1
x <sub>8</sub>	0	0	0	1	1	1	0	1

Матрица R<sup>2</sup>

	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	x <sub>7</sub>	x <sub>8</sub>
x <sub>1</sub>	1	1	1	1	1	1	0	1
x <sub>2</sub>	0	1	1	1	1	1	0	0
x <sub>3</sub>	0	0	1	1	1	1	0	1
x <sub>4</sub>	0	0	0	1	1	1	0	1
x <sub>5</sub>	0	0	0	0	1	1	0	1
x <sub>6</sub>	0	0	0	0	0	1	0	1
x <sub>7</sub>	0	0	0	0	0	1	1	1
x <sub>8</sub>	0	0	0	1	1	1	0	1

Матрица R<sup>3</sup>

	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	x <sub>7</sub>	x <sub>8</sub>
x <sub>1</sub>	1	1	1	1	1	1	0	1
x <sub>2</sub>	0	1	1	1	1	1	0	1
x <sub>3</sub>	0	0	1	1	1	1	0	1
x <sub>4</sub>	0	0	0	1	1	1	0	1
x <sub>5</sub>	0	0	0	0	1	1	0	1
x <sub>6</sub>	0	0	0	0	0	1	0	1
x <sub>7</sub>	0	0	0	0	0	1	1	1
x <sub>8</sub>	0	0	0	1	1	1	0	1

Матрица R<sup>4</sup>

	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	x <sub>7</sub>	x <sub>8</sub>
x <sub>1</sub>	1	1	1	1	1	1	0	1
x <sub>2</sub>	0	1	1	1	1	1	0	1
x <sub>3</sub>	0	0	1	1	1	1	0	1
x <sub>4</sub>	0	0	0	1	1	1	0	1
x <sub>5</sub>	0	0	0	0	1	1	0	1
x <sub>6</sub>	0	0	0	0	0	1	0	1
x <sub>7</sub>	0	0	0	0	0	1	1	1
x <sub>8</sub>	0	0	0	1	1	1	0	1

По сравнению с последовательным алгоритмом, в котором время выполнения зависит от размерности графа, предлагаемое решение позволяет значительно увеличить быстродействие.

Операция построения матрицы достижимости является ключевой в задаче разбиения графа на максимальные сильно связанные подграфы, в задаче построения кратчайших путей и в ряде других задач.

**Список литературы**

1. Волченская Т.В. Разработка алгоритмов решения задач на графах с применением клеточной логики // Новые информационные технологии и системы: тез. докл. II междунар. научн. техн. конф. октябрь 1996 г., г. Пенза. – Пенза, 1996. – С. 95–96.

2. Волченская Т.В., Князьков В.С. и др. Устройство для определения числа вершин подграфов графа. Авт. св. № 1341649 G06F15/20, БИ № 36, 1987.

3. Волченская Т.В., Князьков В.С. и др. Устройство для исследования подмножеств графа. Авт. св. № 1410051, G06F15/31, БИ № 26, 1988.

4. Волченская Т.В., Князьков В.С. и др. Устройство для исследования графов. Авт. св. № 1363237, G06F15/20, БИ № 48, 1987.

5. Волченская Т.В., Князьков В.С. и др. Многофункциональная ячейка однородной структуры. Патент № 1663609, G06F7/00, октябрь 1993.

6. Волченская Т.В., Князьков В. С. Спецпроцессоры для решения задач на графах // Научный журнал Advanced science № 3, ПРИП ФГБОУ ВПО «ВятГУ». – Киров, 2013. – С. 73–82.

7. Кристофидес Н. Теория графов. Аналитический подход. – М.: Мир, 1978.

8. Knyaz'kov V.S., Volchenskaya T.V. Maximally Parallel Algorithms for Raster Image Processing by Cellular VLSI Processors // Pattern Recognition and Image Analysis. – 1996. – Vol. 6, № 2. – P. 403–404.

9. Knyaz'kov V.S. An Assotiative Cellular VLSI Processor for Paralled Processing by Raster Image: the concept and Applied Computations // Pattern Recognition and Image Analysis. – 1996. – Vol. 6, № 2. – P. 401–402.

### References

1. Volchenskaya T.V. *Tezisy докладov II Mezhdunarodnoin-auchno-teshnicheskoi konferencii «Novye informacionnieteshnologii sistemy»* (Theses of reports of II international scientifically – technical conference «New information technologies and systems»), October 1996, Penza, pp. 95–96.

2. Volchenskaya T.V., Knyazkov V.S. The device for definition of number of tops of subgraphs the graph. The copyright certificate no. 1341649 G06F15/20, the bulletin of inventions no. 36, 1987.

3. Volchenskaya T.V., Knyazkov V.S. The device for research of subsets the graph. The copyright certificate no. 1410051, G06F15/31, the bulletin of inventions no. 26, 1988.

4. Volchenskaya T.V., Knyazkov V.S. The device for research the graph. The copyright certificate no. 1363237, G06F15/20, the bulletin of inventions no. 48, 1987.

5. Volchenskaya T.V., Knyazkov V.S. Multipurpose cell of homogeneous structure. The patent no. 1663609, G06F7/00, 1993.

6. Volchenskaya T.V., Knyazkov V.S. The specialized processors for the decision of problems on graphs. *Journal of Advanced science* no. 3, 2013, pp. 73–82.

7. Kristofides N. Graph theory. The analytical approach. Moscow, 1978.

8. Knyaz'kov V.S., Volchenskaya T.V. Maximally Parallel Algorithms for Raster Image Processing by Cellular VLSI Processors // Pattern Recognition and Image Analysis, Vol.6, no. 2, 1996, pp. 403–404.

9. Knyaz'kov V.S. An Assotiative Cellular VLSI Processor for Paralled Processing by Raster Image: the concept and Applied Computations // Pattern Recognition and Image Analysis, Vol. 6, no. 2, 1996, pp. 401–402.

### Рецензенты:

Шатров А.В., д.ф.-м.н., профессор, зав. кафедрой «Математическое моделирование в экономике» Вятского государственного университета, г. Киров;

Бутаев М.М., д.т.н., профессор, ученый секретарь ОАО «НПП Рубин», г. Пенза.

Работа поступила в редакцию 26.02.2014.