

# Параллельный алгоритм нахождения максимального паросочетания в графе

Шунгаров Х. Д. ([hamidsh@rambler.ru](mailto:hamidsh@rambler.ru))

Карачаево-Черкесский государственный университет.

## 1. Введение

Задача нахождения максимальных паросочетаний в графе имеет множество приложений[1]. Известно [2, 3], что для решения задачи о паросочетании имеются эффективные алгоритмы. В настоящей работе для решения задачи нахождения максимального паросочетания в графе предлагается параллельный алгоритм  $\alpha$ , вместе с обоснованием его эффективности.

Все недостающие определения понятий, относящихся к теории параллельных алгоритмов можно найти в [4]; в [5, 6] представлены определения понятий, относящихся к теории графов.

Пусть имеется параллельная вычислительная система, состоящая из любого нужного числа процессоров и произвольно большой памяти, одновременно доступной всем процессорам. Каждый процессор за единицу времени может выполнить любую унарную или бинарную операцию. Время выполнения всех вспомогательных операций, а также время взаимодействия с памятью и время, затрачиваемое на управление процессом, считаются пренебрежимо малыми. Все входные данные перед началом вычислений записаны в памяти. Каждый процессор считывает свои операнды из памяти и после выполнения операции записывает результат в память. После окончания вычислительного процесса все результаты остаются в памяти. Все процессоры и устройство памяти объединяются в единую систему, связанную каналами передачи информации. Такие системы называются параллельными вычислительными системами[4].

Пусть имеется некоторый алгоритм  $\alpha$ , операции которого разбиты на группы, упорядоченные так, что каждая операция любой группы зависит либо от начальных условий алгоритма, либо от результатов операций, находящихся в предыдущих группах. Представление алгоритма в подобном виде называется параллельной формой алгоритма или параллельным алгоритмом[4].

Каждая группа операций называется ярусом параллельной формы, число групп – высотой, а максимальное число операций в ярусе – шириной параллельного алгоритма. Высоту (ширину) параллельного алгоритма  $\alpha$  обозначим через  $H(\alpha)$  ( $S(\alpha)$ ).

Для сравнения параллельных алгоритмов между собой, а также для сравнения параллельных и последовательных алгоритмов, т.е. алгоритмов решения задач для однопроцессорных ЭВМ, используются различные характеристики. Ускорением  $A(\alpha)$  называется величина, показывающая, во сколько раз быстрее конкретная задача может быть решена при помощи параллельного алгоритма по сравнению с последовательной реализацией такого алгоритма[4].

Актуальным и практически важным является нахождение верхних и нижних оценок высот параллельных алгоритмов для решения классов задач.

Концепция математических исследований, связанная с изучением и построением параллельных алгоритмов, называется концепцией неограниченного параллелизма, основными методами которой являются:

- расщепление задачи на независимые подзадачи;
- выделение в задаче потока независимых однотипных задач;
- применение принципа «разделяй и властвуй» с рекурсивным расщеплением задачи на две или несколько задач[4].

## 2. Постановка задачи

Пусть задан связный неориентированный граф  $G=(V, E)$ , с  $n=|V|$  вершинами и  $|E|=m$  ребрами, состоящий из  $\Theta$  компонент связности  $G_t=(V_t, E_t)$ ,  $t=1, 2, \dots, \Theta$  и  $\Theta-1$  точек сочленения,  $|V_t|=n_t$ ,  $|E_t|=m_t$ ,  $\sum_{t=1}^{\Theta} m_t = m$ ,  $\sum_{t=1}^{\Theta} n_t = n - \Theta + 1$ . Напомним, что паросочетанием графа  $G=(V, E)$  называется подграф  $x=(V_x, E_x)$ , порожденный на множестве попарно-несмежных ребер  $E_x \subseteq E$ ,  $V_x \subseteq V$ . Паросочетание  $x^0$  называется максимальным, если число ребер в нем наибольшее среди всех паросочетаний графа  $G$ . Задача о паросочетании состоит в нахождении максимального паросочетания  $x^0$  графа  $G$ .

В настоящей работе для решения задачи о паросочетании предлагается параллельный алгоритм  $\alpha$ , трудоемкость которого составляет  $O\left(\frac{n^3}{\Theta^3}\right)$  операций, где параметр  $\Theta \geq 2$  удовлетворяет неравенству  $2 \leq \Theta \leq \left\lceil \frac{n}{\ln n} \right\rceil$ . Уменьшение времени работы предлагаемого алгоритма получено за счет его представления в параллельной форме и при том условии, что задача о паросочетании будет решаться на параллельной

вычислительной системе, состоящей из  $\Theta$  процессоров. Также исследованы основные характеристики параллельного алгоритма  $\alpha$ , такие как высота, ширина, ускорение.

В дальнейшем нам понадобятся следующие определения [5]. Вершина  $v \in V$  называется насыщенной паросочетанием  $M$  графа  $G$ , если она является концевой вершиной некоторого ребра  $e \in M$ . Чередующейся цепью  $P$  в графе  $G$  называется цепь, ребра которой поочередно входят в  $M$  или  $E \setminus M$ . Пусть  $P$  - чередующаяся цепь между двумя ненасыщенными вершинами в паросочетании  $M$ . Тогда симметрическая разность двух множеств ребер  $M \oplus P$  является паросочетанием с числом ребер на единицу больше, чем в паросочетании  $M$ . При этом цепь  $P$  называется увеличивающейся цепью по отношению к  $M$ .

### 3. Описание параллельного алгоритма $\alpha$

Пусть граф  $G=(V, E)$ , с  $|V|=n$ ,  $|E|=m$ , состоящий из  $\Theta$  компонент связности  $G_t=(V_t, E_t)$ ,  $t=1, 2, \dots, \Theta$  и  $\Theta-1$  точек сочленения,  $|V_t|=n_t$ ,  $|E_t|=m_t$ ,  $\sum_{t=1}^{\Theta} m_t = m$ ,  $\sum_{t=1}^{\Theta} n_t = n - \Theta + 1$ , задан списками смежности  $N_v$ , где  $N_v$  – список вершин, смежных вершине  $v \in V$ .

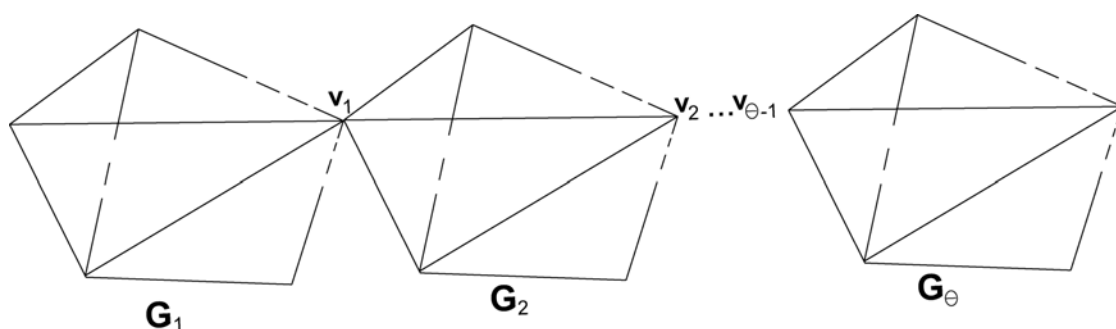


Рис. 1. Граф  $G=(V, E)$  и его компоненты связности  $G_1, G_2, \dots, G_{\Theta}$ .

В дальнейшем нам понадобится следующее определение. Расщепление точки сочленения  $v$  графа  $G=(V, E)$ , на  $k$  вершин такое, что получаются  $k$  реберно-непересекающихся подграфов  $G_k=(V_k, E_k)$ ,  $k=1, 2, \dots, \Theta$ , будем называть  $k$ -расщеплением точки сочленения.

Параллельный алгоритм  $\alpha$  состоит из трех частей:  $\alpha^1, \alpha^2, \alpha^3$ . Часть  $\alpha^1$  состоит из  $\alpha^1_r$  шагов, перенумерованных индексом  $r=1, 2, \dots, \Theta-1$ . На каждом шаге  $\alpha^1_r$  фиксируется

точка сочленения  $v_r$  графа  $G_r$  и эта вершина  $v_r$  2-расщепляется,  $r \in \{1, 2, \dots, \Theta-1\}$ . В результате работы части  $\alpha^1$  исходный граф  $G=(V, E)$  разбивается на  $\Theta$  подграфов  $G_t=(V_t, E_t)$ ,  $t \in \{1, 2, \dots, \Theta\}$ .

Часть  $\alpha^2$  алгоритма  $\alpha$  состоит из этапов  $\alpha_t^2$ , перенумерованных индексом  $t=1, 2, \dots, \Theta$ . На вход каждого из этапов  $\alpha_t^2$ ,  $t \in \{1, 2, \dots, \Theta\}$  соответственно подается граф  $G_{\Theta-t+1}=(V_{\Theta-t+1}, E_{\Theta-t+1})$ , где  $|V_{\Theta-t+1}|=n_{\Theta-t+1}$ ,  $|E_{\Theta-t+1}|=m_{\Theta-t+1}$ . Каждый этап  $\alpha_t^2$ ,  $t \in \{1, 2, \dots, \Theta\}$  начинает свою работу с нумерации ребер графа  $G_{\Theta-t+1}$  числами  $n_{\Theta-t+1}+2$ ,  $n_{\Theta-t+1}+4$ , ...,  $n_{\Theta-t+1}+2m_{\Theta-t+1}$ . Номер ребра  $(u, v)$  будем обозначать через  $N(u, v)$ . Множество  $E_{\Theta-t+1}$  ребер графа  $G_{\Theta-t+1}$  хранятся в массиве END, элементы которого пронумерованы числами  $n_{\Theta-t+1}+1$ ,  $n_{\Theta-t+1}+2$ , ...,  $n_{\Theta-t+1}+2m_{\Theta-t+1}$ . На каждом этапе  $\alpha_t^2$ ,  $t \in \{1, 2, \dots, \Theta\}$  работает алгоритм Габова [2], который находит максимальное паросочетание  $M_{\Theta-t+1}$  графа  $G_{\Theta-t+1}=(V_{\Theta-t+1}, E_{\Theta-t+1})$ . Каждое паросочетание  $M_r$ ,  $r=1, 2, \dots, \Theta$ , хранится в массиве MAT, который имеет по одному элементу для каждой вершины графа  $G_r$ . Ребро  $(v, w) \in E_r$  входит в паросочетание  $M_r$ , если  $MAT(v)=w$  и  $MAT(w)=v$ .

Вершина  $v \in V_r$  графа  $G_r$ ,  $r \in \{1, 2, \dots, \Theta\}$  называется внешней по отношению к фиксированной ненасыщенной вершине  $w$  тогда и только тогда, когда в этом графе существует чередующаяся цепь четной длины от  $w$  к  $v$ . Эта цепь  $P(v)=(v, v_1, v_2, \dots, w)$  начинается с ребра  $(v, v_1)$ , вошедшего в паросочетание  $M_r$  графа  $G_r$ ,  $r \in \{1, 2, \dots, \Theta\}$ . Если на каком-либо шаге этапа  $\alpha_t^2$ ,  $t \in \{1, 2, \dots, \Theta\}$  рассматривается ребро, соединяющее внешнюю вершину  $w$  с ненасыщенной вершиной  $w' \neq w$ , то при этом выделяется увеличивающаяся цепь вида  $w' * P(v)=(w', v, v_1, \dots, w)$ , где  $*$  - обозначение конкатенации. Если такое ребро в графе  $G_{\Theta-t+1}$  на этапе  $\alpha_t^2$ ,  $t \in \{1, 2, \dots, \Theta\}$  не обнаружено, то вершина  $w'$  не войдет ни в одну увеличивающую цепь.

На каждом этапе  $\alpha_t^2$ ,  $t \in \{1, 2, \dots, \Theta\}$  используется массив LABEL, состоящий из  $n_{\Theta-t+1}=|V_{\Theta-t+1}|$  элементов такой, что каждой вершине графа  $G_{\Theta-t+1}$  соответствует один элемент. При этом элемент массива, соответствующий внешней вершине  $v$ , используется для определения увеличивающей цепи  $P(v)$ . Элемент массива LABEL интерпретируется как начальная метка, или как метка вершины, или как метка ребра. Начальная вершина  $v_0$  имеет начальную метку. Увеличивающая цепь при этом  $P(v_0)=v_0$ .

Каждый этап  $\alpha_t^2$ ,  $t \in \{1, 2, \dots, \Theta\}$  начинается с того, в графе  $G_{\Theta-t+1}$  в качестве начальной вершины выбирается точка сочленения  $v_{\Theta-t} \in V_{\Theta-t+1}$ , для которой  $\text{LABEL}(v_{\Theta-t})=0$ .

Если  $\text{LABEL}(v)=i$ , где  $1 \leq i \leq n_t$ ,  $n_t = |V_t|$ ,  $t=1, 2, \dots, \Theta$ , то говорят, что вершина  $v$  имеет метку вершины. В этом случае  $v$  является внешней вершиной и элемент  $\text{LABEL}(v)$  равен номеру другой внешней вершины. Цепь  $P(v) = (v, \text{MAT}(V)) * P(\text{LABEL}(v))$ .

Если  $\text{LABEL}(v) = m_t + 2i$ , где  $1 \leq i \leq n_t$ ,  $m_t = |E_t|$ ,  $t \in \{1, 2, \dots, \Theta\}$ , то говорят, что вершина  $v$  имеет метку ребра. Если  $v$  – внешняя вершина и  $\text{LABEL}(v)$  содержит номер ребра, соединяющего две внешние вершины, то  $\text{LABEL}(v) = N(x, y)$ . Метка ребра  $N(x, y)$  вершины  $v$  указывает, что существует чередующаяся цепь четной длины из  $v$  в начальную вершину  $u$ , которая проходит через ребро  $(x, y)$ . Пусть  $v$  принадлежит цепи  $P(x)$ , а  $P(x, v)$  – часть цепи  $P(x)$  из  $x$  в  $v$ , тогда цепь  $P(v) = \text{rev}P(x, v) * P(y)$ , где  $\text{rev}P(x, v)$  – обращение цепи из  $x$  в  $v$ .

Если  $\text{LABEL}(v) < 0$ , то вершина  $v$  не является внешней. В начале каждого этапа  $\alpha_t^2$ ,  $t \in \{1, 2, \dots, \Theta\}$  полагаем  $\text{LABEL}(v) = -1$  для всех вершин  $v \in V_{\Theta-t+1}$  графа  $G_{\Theta-t+1}$ .

На каждом этапе  $\alpha_t^2$ ,  $t \in \{1, 2, \dots, \Theta\}$  алгоритма  $\alpha$  используются массивы FIRST и OUTER. Если  $v$  – внешняя вершина, то  $\text{FIRST}(v)$  – первая не внешняя вершина в цепи  $P(v)$ ; если  $v$  – не внешняя вершина, то  $\text{FIRST}(v) = 0$ . Массив OUTER используется для хранения внешних вершин, встречающихся при поиске увеличивающей цепи. Граф поиска растет во внешних вершинах в порядке их появления. В этих внешних вершинах графа  $G_{\Theta-t+1}$ ,  $t \in \{1, 2, \dots, \Theta\}$  на этапе  $\alpha_t^2$  производится поиск в ширину.

Каждый этап  $\alpha_t^2$ ,  $t \in \{1, 2, \dots, \Theta\}$  части  $\alpha^2$  алгоритма  $\alpha$  представляет собой алгоритм Габова [2] и состоит из трех процедур: PROC-EDMONS, PROC-LABEL, PROC-REMATCH. Процедура PROC-EDMONS является главной. На каждом этапе  $\alpha_t^2$ ,  $t \in \{1, 2, \dots, \Theta\}$  эта процедура начинает поиск увеличивающей цепи из каждой ненасыщенной вершины графа  $G_{\Theta-t+1}$ , просматривая его ребра с целью расширить паросочетание и приписать метки. При построении увеличивающей цепи на каждом этапе  $\alpha_t^2$ ,  $t \in \{1, 2, \dots, \Theta\}$  вызывается процедура PROC-REMATCH, которая в графе  $G_{\Theta-t+1}$  находит новое паросочетание, на одно ребро большее текущего паросочетания.

Напомним [5, 6], что «цветком» называется замкнутая чередующаяся цепь нечетной длины.

Если на каком-либо этапе  $\alpha_t^2$ ,  $t \in \{1, 2, \dots, \Theta\}$  при рассмотрении некоторого ребра  $(v, u) \in E_{\Theta-t+1}$  графа  $G_{\Theta-t+1}$  образуется «цветок», то вызывается процедура PROC-LABEL. В этом случае вершины  $v$  и  $u$  являются внешними. Процедура PROC-LABEL выполняет следующие операции:

1) вводится переменная JOIN, значением которой является первая не внешняя вершина, которая принадлежит как  $P(v)$  так и  $P(u)$ ;

2) все не внешние вершины, предшествующие JOIN в  $P(v)$  или  $P(u)$  становятся внешними и помечаются меткой ребра  $N(v, u)$ ; эта метка указывает, что к каждой из этих вершин существует чередующаяся цепь четной длины из начальной вершины, который проходит через ребро  $(v, u)$ ;

3) после этого JOIN – первая не внешняя вершина как в  $P(v)$  так и в  $P(u)$ , поэтому элементы массива FIRST, которые соответствуют вершинам, предшествующим JOIN в  $P(v)$  или  $P(u)$ , полагаются равными JOIN.

Этап  $\alpha_t^2$ ,  $t \in \{1, 2, \dots, \Theta\}$  заканчивает свою работу, как только не окажется увеличивающей цепи по отношению к текущему паросочетанию графа  $G_{\Theta-t+1} = (V_{\Theta-t+1}, E_{\Theta-t+1})$ . Каждый следующий этап  $\alpha_{t+1}^2$ ,  $t \in \{1, 2, \dots, \Theta-1\}$  начинает свою работу с того, что фиксируется точка сочленения  $v_{\Theta-t} \in V_{\Theta-t}$  графа  $G_{\Theta-t} = (V_{\Theta-t}, E_{\Theta-t})$ . Если эта вершина  $v_{\Theta-t}$  оказалась насыщенной паросочетанием  $M_{\Theta-t+1}$  графа  $G_{\Theta-t+1}$ , полученным на этапе  $\alpha_t^2$ , то  $v_{\Theta-t}$  считается насыщенной в графе  $G_{\Theta-t}$ ,  $t \in \{1, 2, \dots, \Theta\}$  на этапе  $\alpha_{t+1}^2$ .

Часть  $\alpha^3$  параллельного алгоритма  $\alpha$  состоит в объединении паросочетаний  $M_{\Theta}, M_{\Theta-1}, \dots, M_1$ , графов  $G_{\Theta}, G_{\Theta-1}, \dots, G_1$  соответственно полученных на этапах  $\alpha_1^2, \alpha_2^2, \dots, \alpha_{\Theta}^2$  с помощью операции замыкания вершин  $v_r$ ,  $r \in \{1, 2, \dots, \Theta-1\}$  и получении максимального паросочетания  $M$  исходного графа  $G$ .

Работу параллельного алгоритма  $\alpha$  покажем на графе  $G$ , изображенном на рис. 2. Граф  $G$  состоит из  $\Theta=2$  компонент связности  $G_1$  и  $G_2$ ;  $v=4$  – точка сочленения. На вход части  $\alpha^1$  подается граф  $G=(V, E)$  и точка сочленения  $v=4$  2-расщепляется. На вход этапа  $\alpha_1^2$  части  $\alpha^2$  подается граф  $G_2$  (рис 3.).

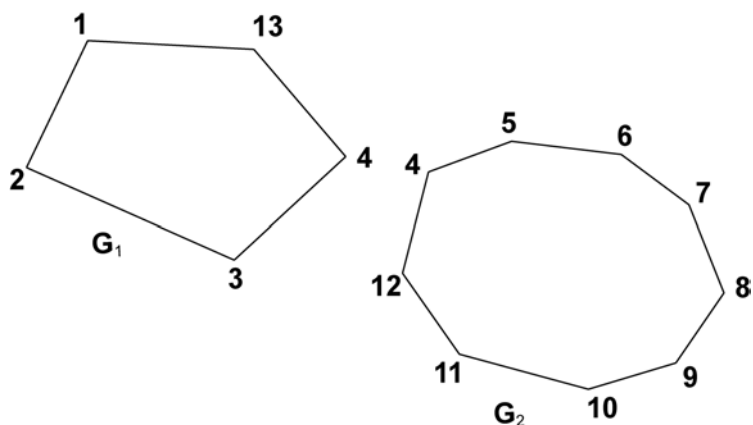


Рис.2.Работа части  $\alpha^1$  алгоритма  $\alpha$  на графе  $G$ , состоящем из компонент связности  $G_1$  и  $G_2$ .

Массив OUTER состоит сначала из вершин 4, 6, 8, 9, 11. Значения массивов LABEL и FIRST занесем в таблицу 1.

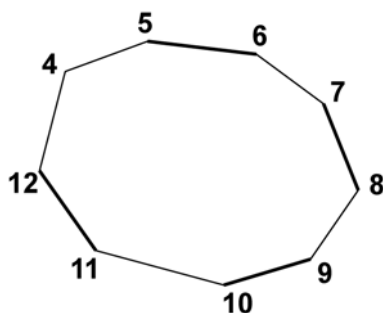


Рис. 3. Работа этапа  $\alpha_1^2$  части  $\alpha^2$ . Ребра паросочетания  $M_2$  графа  $G_2$  выделены жирными линиями.

Таблица 1.

N	LABEL	FIRST	LABEL	FIRST
4	0	0	0	0
5	-1	0	N(9, 8)	0
6	4	5	4	0
7	-1	0	N(9, 8)	0
8	6	7	6	0
9	11	10	11	0
10	-1	0	N(9, 8)	0
11	4	12	4	0
12	-1	0	N(9, 8)	0

Зафиксируем вершину  $v=5$ , которая имеет реберную метку  $N(9, 8)$ . Поэтому для определения цепи  $P(5)$  нам потребуются цепи  $P(9)$  и  $P(8)$ . Поскольку  $v=9$  – внешняя вершина, которая имеет метку вершины, то  $P(9) = (9, MAT(9)) * P(LABEL(9)) = (9,10) * P(11) = (9,10,11, MAT(11)) * P(LABEL(11)) = (9,10,11,12) * P(4) = (9,10,11,12,4)$ .

Вершина  $v=8$  также является внешней и имеет метку вершины, поэтому  $P(8) = (8, MAT(8)) * P(LABEL(8)) = (8,7) * P(LABEL(8)) = (8,7) * P(6) = (8,7,6, MAT(6)) * P(LABEL(6)) = (8,7,6,5) * P(4) = (8,7,6,5,4)$ .

Определим теперь цепь  $P(5)$ :  $P(5) = revP(8,5) * P(9) = (5,6,7,8) * (9,10,11,12,4) = (5,6,7,8,9,10,11,12,4) = ((5,6), (7,8), (9,10), (11,12), 4)$ . По отношению к выделенному паросочетанию нет увеличивающей цепи, поэтому оно является максимальным паросочетанием.

На вход этапа  $\alpha_2^2$  подается граф  $G_1$  (рис. 4). Массив OUTER содержит сначала вершины 1, 3, 4. Значения массивов LABEL и FIRST занесем в таблицу 2:

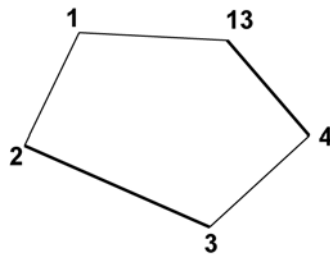


Рис. 4. Работа этапа  $\alpha_2^2$ .

Таблица 2.

N	LABEL	FIRST	LABEL	FIRST
1	0	0	0	0
2	-1	0	N(3, 4)	0
3	1	2	1	0
4	1	13	1	0
13	-1	0	N(3, 4)	0

Зафиксируем вершину  $v=2$ , которая имеет метку ребра  $N(3, 4)$ . Определим цепь  $P(2)$ :  $P(2) = revP(3,2) * P(4)$ . Вершина  $v=4$  имеет метку вершины, поэтому  $P(4) = (4, MAT(4)) * P(LABEL(4)) = (4,13) * P(1) = (4,13,1)$ . Следовательно,  $P(2) = revP(3,2) * P(4) = (2,3) * (4,13,1) = (2,3,4,13,1) = ((2,3), (4,13), 1)$ . По отношению к выделенному паросочетанию нет увеличивающей цепи, поэтому полученное паросочетание является максимальным.

Часть  $\alpha^3$  алгоритма  $\alpha$  состоит в объединении максимальных паросочетаний, полученных на этапах  $\alpha_1^2$  и  $\alpha_2^2$  с помощью операции замыкания в вершине  $v=4$ . Это будет

паросочетание  $M = ((5,6); (7,8); (9,10); (11,12); 4) * ((2,3); (4,13); 1) = ((5,6); (7,8); (9,10); (11,12); 4; (2,3); (4,13); 1)$ ,  
 которое является максимальным паросочетанием  $M$  исходного графа  $G$ . Убедимся в этом  
 рассмотрев работу алгоритма Габова [2] на графе  $G$  (рис. 5).

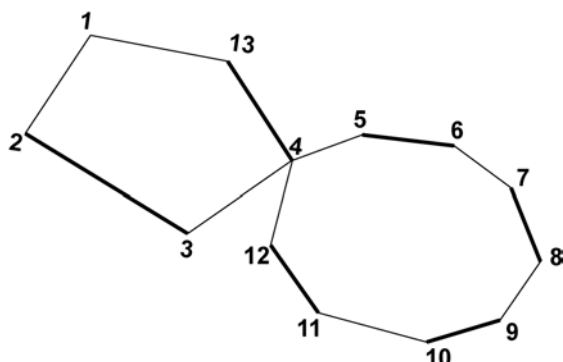


Рис. 5. Работа алгоритма Габова нахождения максимального паросочетания в графе  $G$ .

Начальной вершиной является  $v=1$ . Массив OUTER содержит сначала вершины 1, 3, 4, 11, 6, 9, 8. Значения массивов LABEL и FIRST занесены в таблицу 3.

Таблица 3.

N	LABEL	FIRST	LABEL	FIRST
1	0	0	0	0
2	-1	0	N(3, 4)	0
3	1	2	1	0
4	1	13	1	0
5	-1	0	N(9, 8)	0
6	4	5	4	0
7	-1	0	N(9, 8)	0
8	6	7	6	0
9	11	10	11	0
10	-1	0	N(9, 8)	0
11	4	12	4	0
12	-1	0	N(9, 8)	0
13	-1	0	N(3, 4)	0

Зафиксируем вершину  $v=12$ , которая имеет метку ребра  $N(9, 8)$ , тогда цепь  $P(12) = revP(9,12) * P(8)$ . Вершина  $v=9$  имеет метку вершины, поэтому  $P(9) = (9, MAT(9)) * P(LABEL(9)) = (9,10) * P(11) = (9,10,11, MAT(11)) * P(LABEL(11)) = (9,10,11,12) * P(4) = (9,10,11,12,4, MAT(4)) * P(LABEL(4)) = (9,10,11,12,4,13,1)$ .

Вершина  $v=8$  имеет метку вершины, поэтому  $P(8) = (8, MAT(8)) * P(LABEL(8)) = (8,7) * P(6) = (8,7,6, MAT(6)) * P(LABEL(6)) = (8,7,6,5) * P(4) = (8,7,6,5,4, MAT(4)) * P(LABEL(4)) = (8,7,6,5,4,13) * P(1) = (8,7,6,5,4,13,1)$ . Следовательно, цепь  $P(12) = revP(9,12) * P(8) = ((12,11); (10,9); (8,7); (6,5); (4,13); 1)$ . После дополнения получим новое паросочетание, состоящее из ребер  $(12, 11); (10, 9); (8, 7); (6, 5); (4, 13); 1$  и  $(3, 2)$ . По отношению к полученному паросочетанию нет увеличивающей цепи, поэтому это паросочетание – максимальное (рис. 5).

#### 4. Анализ и обоснование алгоритма $\alpha$

В процессе анализа алгоритмов особое место занимает оценка их трудоемкостей. Под трудоемкостью алгоритма будем понимать время выполнения соответствующей программы на ЭВМ. Предположим, что параллельная вычислительная система состоит из абстрактных вычислительных машин. Такая машина в состоянии выполнять арифметические операции, сравнения, пересылки и операции условной и безусловной передач управления. Такие операции называются элементарными. Каждая из элементарных операций выполняется за единицу времени и, следовательно, время работы алгоритма равно числу выполненных им элементарных операций.

Если всем исходным данным задачи присвоить конкретные значения и разместить их в памяти ЭВМ, то эти исходные данные называются входом задачи. Размером (длиной) входа называется число ячеек памяти ЭВМ, занятых входом[5].

Трудоемкостью (сложностью) алгоритма решения данной задачи называется функция  $f$ , ставящая в соответствие каждому натуральному числу  $n$  время работы  $f(n)$  алгоритма в худшем случае на входах длины  $n$ . Иначе, функция  $f(n)$  является максимальным временем работы алгоритма по всем входам задачи длины  $n$ [5].

При нахождении оценок сложности алгоритмов будем использовать  $O$ -символику. Будем говорить, что неотрицательная функция  $f(n)$  не превосходит по порядку функцию  $g(n)$ , если существует такая константа  $c$ , что  $f(n) \leq cg(n)$  для всех  $n \geq 0$  и при этом будем писать  $f(n) = O(g(n))$ . Иногда вместо «трудоемкость алгоритма есть  $O(g(n))$ » будем говорить «алгоритм решает задачу за время  $O(g(n))$ »[5].

Теорема 1. Пусть задан граф  $G=(V, E)$ ,  $|V|=n$ ,  $|E|=m$ , состоящий из  $\Theta$  компонент связности  $G_r=(V_r, E_r)$ ,  $r=1, 2, \dots, \Theta$  и  $\Theta-1$  точек сочленения,  $|V_r|=n_r$ ,  $|E_r|=m_r$ ,  $\sum_{r=1}^{\Theta} m_r = m$ ,

$\sum_{r=1}^{\Theta} n_r = n - \Theta + 1$ . Тогда паросочетание  $M$  графа  $G=(V, E)$ , полученное параллельным алгоритмом  $\alpha$  является максимальным.

Доказательство. Согласно конструкции параллельного алгоритма  $\alpha$ , в результате работы части  $\alpha^1$  исходный граф  $G=(V, E)$  разбивается на  $\Theta$  связанных подграфов  $G_r=(V_r, E_r)$ ,  $r=1, 2, \dots, \Theta$ . На вход каждого этапа  $\alpha_t^2$ ,  $t \in \{1, 2, \dots, \Theta\}$  подается граф  $G_{\Theta-t+1}=(V_{\Theta-t+1}, E_{\Theta-t+1})$ . На каждом этапе  $\alpha_t^2$ ,  $t=1, 2, \dots, \Theta$ , части  $\alpha^2$  работает алгоритм Габова [2], который находит максимальное паросочетание  $M_{\Theta-t+1}$ .

Паросочетание  $M$  графа  $G$ , полученное в части  $\alpha^3$  алгоритма  $\alpha$  равно объединению всех максимальных паросочетаний  $M_{\Theta-t+1}$ , полученных на этапах  $\alpha_t^2$ ,  $t=1, 2, \dots, \Theta$ .

Предположим, что  $M$  – не является максимальным паросочетанием графа  $G$ , т. е. в  $G$  существует другое паросочетание  $M'$  с  $|M'| > |M|$ . Рассмотрим ребра, входящие в  $M \oplus M'$ , где  $\oplus$  - знак симметрической разности. Эти ребра образуют подграф  $G' \subset G'$  графа  $G$ . Поскольку никакие два ребра не могут быть инцидентны одной и той же вершине в паросочетании  $M$ , то подграф  $G'=(V, M \oplus M')$  имеет специальную структуру – все его вершины имеют степень 2 и меньше. Если степень вершины в  $G'$  равна 2, то одно из инцидентных ей ребер входит в  $M$ , а другое - в  $M'$ . Поэтому компоненты в  $G'$  должны быть либо цепями, либо циклами четной длины с чередованием ребер из  $M$  и из  $M'$ . Поскольку  $M' > M$ , то среди этих компонентов необходимо есть цепь нечетной длины, начинающаяся и оканчивающаяся ребрами  $M'$ , которая является увеличивающей цепью. Однако из конструкции алгоритма  $\alpha$  следует, что среди компонентов графа  $G'$  нет цепей нечетной длины, начинающихся и оканчивающихся ребрами  $M'$ . Получили противоречие, следовательно,  $M$  – максимальное паросочетание графа  $G$ . Таким образом теорема 1 доказана. Докажем теперь, что справедлива

Теорема 2. Пусть задан граф  $G=(V, E)$ ,  $|V|=n$ ,  $|E|=m$ , состоящий из  $\Theta$  компонент связности  $G_r=(V_r, E_r)$ ,  $r=1, 2, \dots, \Theta$  и  $\Theta-1$  точек сочленения,  $|V_r|=n_r$ ,  $|E_r|=m_r$ ,  $\sum_{r=1}^{\Theta} m_r = m$ ,

$\sum_{r=1}^{\Theta} n_r = n - \Theta + 1$ ,  $\Theta \geq 2$ . Тогда высота  $H(\alpha)$  параллельного алгоритма  $\alpha$  нахождения максимального паросочетания в графе  $G$  равна  $\Theta+2$ , т. е.  $H(\alpha)=\Theta+2$ .

Доказательство. Разобьем операции параллельного алгоритма  $\alpha$  на группы и упорядочим их так, что каждая операция любой группы зависит либо от начальных

условий алгоритма, либо от результатов выполнения операций, находящихся в предыдущих группах:

- 1) операции расщепления точек сочленения графа  $G=(V, E)$ ;
- 2) операции алгоритма Габова [2] нахождения максимального паросочетания в каждом графе  $G_r=(V_r, E_r)$ ,  $r \in \{1, 2, \dots, \Theta\}$ ;
- 3) операции объединения паросочетаний  $M_t$  графов  $G_t$ ,  $t \in \{1, 2, \dots, \Theta\}$  полученных соответственно на этапах  $\alpha_t^2$  с помощью операции замыкания вершин  $v_r, r \in \{1, 2, \dots, \Theta-1\}$ .

Поскольку, согласно конструкции параллельного алгоритма  $\alpha$ , каждая из групп 1), 3) выполняется 1 раз, а группа операций 2) -  $\Theta$ -раз, то число ярусов параллельного алгоритма  $\alpha$  равно  $\Theta+2$ . Отсюда и из определения высоты параллельного алгоритма  $\alpha$  следует, что  $H(\alpha)=\Theta+2$ , где  $\Theta \geq 2$ . Таким образом, теорема 2 доказана.

Теорема 3. Пусть задан граф  $G=(V, E)$ ,  $|V|=n$ ,  $|E|=m$ , состоящий из  $\Theta$  компонент связности  $G_r=(V_r, E_r)$ ,  $r=1, 2, \dots, \Theta$  и  $\Theta-1$  точек сочленения,  $|V_r|=n_r$ ,  $|E_r|=m_r$ ,  $\sum_{r=1}^{\Theta} m_r = m$ ,

$$\sum_{r=1}^{\Theta} n_r = n - \Theta + 1, \quad n_* = \min_{1 \leq r \leq \Theta} n_r, \quad n^* = \max_{1 \leq r \leq \Theta} n_r, \quad n_* \geq \lceil \ln n \rceil, \quad n^* \leq \left\lfloor \frac{n}{2} \right\rfloor, \quad 2 \leq \Theta \leq \left\lceil \frac{n}{\ln n} \right\rceil. \quad \text{Тогда ширина}$$

$S(\alpha)$  параллельного алгоритма  $\alpha$  нахождения максимального паросочетания графа  $G$  удовлетворяет соотношению  $S(\alpha) = O\left(\frac{n^3}{\Theta^3}\right)$ .

Доказательство. Согласно конструкции параллельного алгоритма  $\alpha$ , ярус 1 представляет собой группу операций 2-расщепления точек сочленения графа  $G$ . Поэтому трудоемкость такого яруса составляет  $O(\Theta)$  элементарных операций. Каждый из ярусов 2, 3, ...,  $\Theta+1$  представляет собой группу операций алгоритма Габова [2] выделения максимального паросочетания графа  $G_r=(V_r, E_r)$ ,  $r \in \{1, 2, \dots, \Theta\}$ , поэтому трудоемкость каждого такого яруса составляет  $O(n_r^3)$  элементарных операций. Ярус  $\Theta+2$  представляет собой группу операций объединения паросочетаний  $M_r$ ,  $r \in \{1, 2, \dots, \Theta\}$  полученных в части  $\alpha^2$ . Поэтому трудоемкость такого яруса составляет  $O(\Theta)$  элементарных операций. Для нумерации ребер графа  $G_r=(V_r, E_r)$  потребуется  $O(n_r d)$  элементарных операций [5]. Используя теперь определение ширины параллельного алгоритма  $\alpha$  и условия теоремы 3, имеем

$$S(\alpha) = \max \left( O(\Theta), O\left(\frac{n^3}{\Theta^3}\right), O(\Theta), O\left(\frac{nd}{\Theta}\right) \right) = O\left(\frac{n^3}{\Theta^3}\right). \quad (1)$$

Таким образом, теорема 3 доказана.

Из теорем 2, 3 вытекает

Следствие 1. Пусть выполняются условия теоремы 3. Тогда высота  $H(\alpha)$  параллельного алгоритма  $\alpha$  выделения максимального паросочетания графа  $G$  удовлетворяет неравенству

$$4 \leq H(\alpha) \leq \left\lceil \frac{n}{\ln n} \right\rceil + 2.$$

Теорема 4. Пусть задан граф  $G=(V, E)$ ,  $|V|=n$ ,  $|E|=m$ , состоящий из  $\Theta$  компонент связности  $G_r=(V_r, E_r)$ ,  $r=1, 2, \dots, \Theta$  и  $\Theta-1$  точек сочленения,  $|V_r|=n_r$ ,  $|E_r|=m_r$ ,  $\sum_{r=1}^{\Theta} m_r = m$ ,

$$\sum_{r=1}^{\Theta} n_r = n - \Theta + 1, \quad n_r \leq \frac{n}{\Theta}, \quad n_* = \min_{1 \leq r \leq \Theta} n_r, \quad n^* = \max_{1 \leq r \leq \Theta} n_r, \quad 2 \leq \Theta \leq \left\lceil \frac{n}{\ln n} \right\rceil, \quad n_* \geq \lceil \ln n \rceil, \quad n^* \leq \frac{n}{2}.$$

Тогда ускорение  $A(\alpha)$  параллельного алгоритма  $\alpha$  нахождения максимального паросочетания графа  $G$  удовлетворяет неравенству

$$A(\alpha) = O(\Theta + 2).$$

Доказательство. Обозначим через  $n_p(\alpha)$  – число операций (трудоемкость) параллельного алгоритма  $\alpha$ , а через  $n(\alpha)$  – число операций последовательной реализации параллельного алгоритма  $\alpha$ . Из определения ускорения параллельного алгоритма следует, что

$$A(\alpha) = \frac{n(\alpha)}{n_p(\alpha)}. \tag{2}$$

Принимая во внимание равенство  $n_p(\alpha) = S(\alpha)$  и соотношение (1), получим

$$n_p(\alpha) = O\left(\frac{n^3}{\Theta^3}\right), \tag{3}$$

где  $2 \leq \Theta \leq \left\lceil \frac{n}{\ln n} \right\rceil$ . Величина  $n(\alpha)$  согласно определению удовлетворяет равенству

$$n(\alpha) = O\left(\frac{n^3}{\Theta^3}\right) \cdot (\Theta + 2). \tag{4}$$

Подставляя теперь (3), (4) в правую часть равенства (2), получим  $A(\alpha) = O(\Theta + 2)$ . Таким образом, теорема 4 доказана.

Из теорем 3, 4 вытекает

Следствие 2. Пусть выполняются условия теоремы 3. Тогда параллельный алгоритм  $\alpha$  находит максимальное паросочетание графа  $G$  за время  $O\left(\frac{n^3}{\Theta^3}\right)$  элементарных

операций, используя параллельную вычислительную систему, состоящую из  $\Theta+2$  процессоров.

#### Литература.

1. Ловас Л., Пламмер М. Прикладные задачи теории графов. Теория паросочетаний в математике, физике, химии.-М.:Мир, 1988.-653с.
2. H. N. Gabow. An Efficient Implementation of Edmons Algorithm for Maximum Matching of Graphs. J. ASM, vol. 23, 221-234 (1976).
3. M. I. Balinski. Labeling to Obtain a Maximum Matching in Combinatorial Mathematics and Its Applications (R. C. Bose and T. A. Dowling, Eds), Univ., North Carolina Press Chappel Hill, N. C., 1967, p. p. 585-602.
4. Воеводин В. В. Математические модели и методы в параллельных вычислительных системах. М.: Наука, 1986. – 296 с.
5. Емеличев В. А. и др. Лекции по теории графов. М.: Наука, 1990. – 384 с.
6. Свами М., Тхуласираман К. Графы, сети и алгоритмы. М.: Мир, 1984. – 455 с.