

Алгоритмы раскраски графа

Необходимо раскрасить вершины графа таким образом, чтобы смежные вершины были окрашены в разные цвета.

Минимальное число красок, в которые можно раскрасить граф называется *хроматическим числом графа*.

Задача раскраски вершин графа относится к NP-полным задачам. Различают точные и приближенные алгоритмы

Раскраски. Примером точных алгоритмов служит алгоритм Вейссмана.

Алгоритм состоит из двух частей:

1. Построение семейства максимальных внутренне устойчивых множеств (МВУМ) (метод Магу);
2. Выбор минимального числа МВУМ, покрывающих все вершины графа (метод Петрика).

Множество вершин X_s графа $G(X, U)$ называется *внутренне устойчивым (независимым)*, если никакие две вершины из этого множества не смежны, $X_s \subset X$ [$\Gamma X_s \cap X_s = \emptyset$]. Внутренне устойчивое множество называется *максимальным*, если оно не является собственным подмножеством некоторого другого

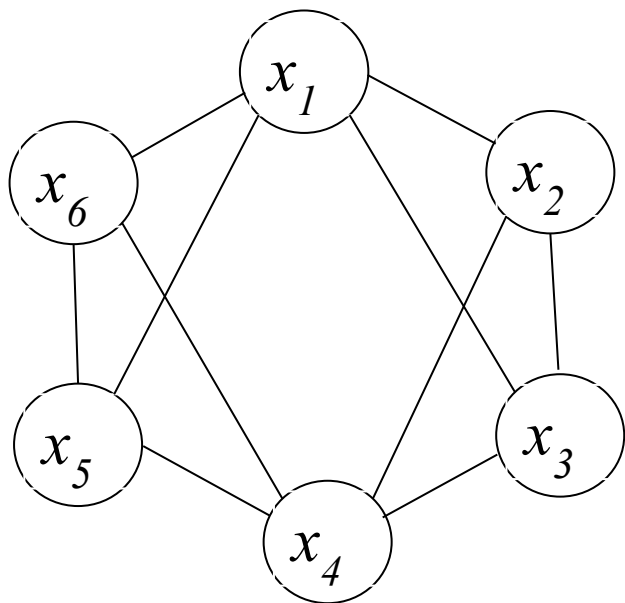
1. В матрице соединений R для каждой вершины подсчитывается число ненулевых элементов r_i ;
2. Находится вершина x_i с $\max r_i$, если таких вершин несколько, то выбирается любая;
3. Для выбранной вершины x_i записывается выражение $C_i = (x_i \vee x_a x_b \dots x_q)$, где $\Gamma x_i = \{x_a, x_b, \dots, x_q\}$;
4. Из матрицы R удаляются строка и столбец, соответствующие вершине x_i ;
5. Если $R \neq \emptyset$, то переход к п. 2, иначе к п. 6;
6. Составляется конъюнкция $\Pi = \bigwedge C_i$. Раскрываются скобки. В полученной дизъюнкции на основе законов булевой алгебры выполняется минимизация.
7. Результат минимизации записывается в виде $\Pi = \bigvee K_j$;
8. Для каждого K_j ищутся вершины графа, не вошедшие в него. Получено ϕ_j и семейство МВУМ $\Psi = \{\phi_1, \phi_2, \dots, \phi_l\}$;
9. Для каждой вершины $x_i \in X$ определяются подмножества ϕ_j , в которые входит вершина $x_i \in \phi_j$. Составляется дизъюнкция $t_i = \bigvee \phi$.

10. Составляется конъюнкция $\Pi' = \bigwedge t_j$. Раскрываются скобки. В полученной дизъюнкции на основе законов булевой алгебры выполняется минимизация;

11. Получена дизъюнкция конъюнктивных термов $\Pi' = \bigvee (\bigwedge \phi_j)$. Выбирается конъюнктивный терм $\bigwedge \phi_j$ с минимальным числом сомножителей.

Количество сомножителей в этом терме и есть хроматическое число графа. Число минимальных термов – число вариантов раскраски графа. А каждое ϕ_j – множество вершин, которые можно окрасить в один цвет.

Заметим, что п.п. 1-8 составляют метод Магу, а п.п. 9-11 – метод Петрика.



$$R = \begin{array}{c|cccccc|c} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & r_i \\ \hline x_1 & 0 & 1 & 1 & 0 & 1 & 1 & 4 \\ x_2 & & 0 & 1 & 1 & 0 & 0 & 3 \\ x_3 & & & 0 & 1 & 0 & 0 & 3 \\ x_4 & & & & 0 & 1 & 1 & 4 \\ x_5 & & & & & 0 & 1 & 3 \\ x_6 & & & & & & 0 & 3 \end{array}$$

1. В матрице R подсчитываем число ненулевых элементов r_i ;
2. $\max r_i = r_1 = r_4 = 4$, выбираем x_1 ;
3. $\Gamma x_1 = \{x_2, x_3, x_5, x_6\}$, записываем выражение

$$C_1 = (x_1 \vee x_2 x_3 x_5 x_6);$$
4. Из матрицы R удаляем строку и столбец, соответствующие вершине x_1 ;

	x_2	x_3	x_4	x_5	x_6	r_i
x_2	0	1	1	0	0	2
x_3		0	1	0	0	2
x_4			0	1	1	4
x_5				0	1	2
x_6					0	2

	x_2	x_3	x_5	x_6	r_i
x_2	0	1	0	0	1
x_3		0	0	0	1
x_5			0	1	1
x_6				0	1

	x_3	x_5	x_6	r_i
x_3	0	0	0	0
x_5		0	1	1
x_6			0	1

	x_3	x_6	r_i
x_3	0	0	0
x_6		0	0

5. $R \neq \emptyset$, $\max r_i = r_4 = 4$;

$\Gamma x_4 = \{x_2, x_3, x_5, x_6\}$,

$C_4 = (x_4 \vee x_2 x_3 x_5 x_6)$;

6. Из матрицы R удаляем строку и столбец, соответствующие вершине x_4 ;

7. $R \neq \emptyset$, $\max r_i = r_2 = r_3 = r_5 = r_6 = 1$, выбираем x_2 ;

$\Gamma x_2 = \{x_3\}$, $C_2 = (x_2 \vee x_3)$;

8. Из матрицы R удаляем строку и столбец, соответствующие вершине x_2 ;

9. $R \neq \emptyset$, $\max r_i = r_5 = r_6 = 1$, выбираем x_5 ;

$\Gamma x_5 = \{x_6\}$, $C_5 = (x_5 \vee x_6)$;

10. Из матрицы R удаляем строку и столбец, соответствующие вершине x_5 ;

11. $R = \emptyset$;

12. Составляем конъюнкцию C_i и выполняем минимизацию

$$\begin{aligned}
 \Pi = \bigwedge C_i = C_1 C_2 C_4 C_5 &= (x_1 \vee x_2 x_3 x_5 x_6)(x_2 \vee x_3)(x_4 \vee x_2 x_3 x_5 x_6)(x_5 \vee x_6) = \\
 &= x_1 x_2 x_4 x_5 \vee x_1 x_2 x_4 x_6 \vee x_1 x_3 x_4 x_5 \vee x_1 x_3 x_4 x_6 \vee x_2 x_3 x_5 x_6 = \bigvee K_j = \\
 &= \bigvee_{j=1}^5 K_j;
 \end{aligned}$$

13. Для каждого K_j ищем ϕ_j :

$$\phi_1 = \{x_3, x_6\}, \phi_2 = \{x_3, x_5\}, \phi_3 = \{x_2, x_6\}, \phi_4 = \{x_2, x_5\}, \phi_5 = \{x_1, x_4\}.$$

Получено семейство МВУМ Ψ ;

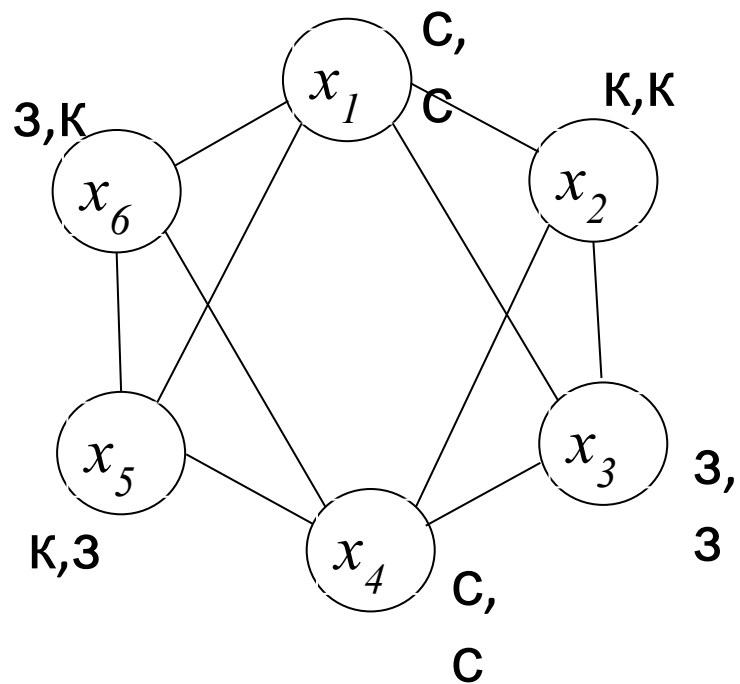
14. Для каждой вершины определим подмножества ϕ_j , в которые она входит. Строим дизъюнкцию $t_i = \bigvee \phi_j$;

$$t_1 = \phi_5; t_2 = \phi_3 \vee \phi_4; t_3 = \phi_1 \vee \phi_2; t_4 = \phi_5; t_5 = \phi_2 \vee \phi_4; t_6 = \phi_1 \vee \phi_3;$$

15. Составляем конъюнкцию и выполняем минимизацию булевой функции

$$\begin{aligned}
 \Pi' = \bigwedge t_i = t_1 t_2 t_3 t_4 t_5 t_6 &= \phi_5 (\phi_3 \vee \phi_4) (\phi_1 \vee \phi_2) \phi_5 (\phi_2 \vee \phi_4) (\phi_1 \vee \phi_3) = \\
 &= \phi_1 \phi_4 \phi_5 \vee \phi_2 \phi_3 \phi_5
 \end{aligned}$$

Хроматическое число графа $\chi(G) = 3$. Существует два варианта раскраски графа.



$$\phi_1 = \{x_3, x_6\}, \phi_2 = \{x_3, x_5\}, \phi_3 = \{x_2, x_6\},$$

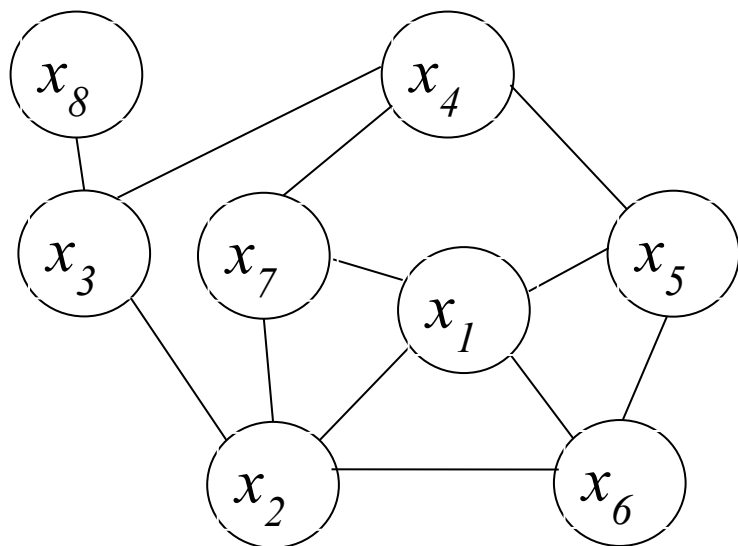
$$\phi_4 = \{x_2, x_5\}, \phi_5 = \{x_1, x_4\}.$$

$$\phi_1 \phi_4 \phi_5 \vee \phi_2 \phi_3 \phi_5$$

Недостатком точных алгоритмов является низкое быстродействие. Поэтому на практике используют приближенные алгоритмы, примером которых может служить

Алгоритм, использующий упорядочивание вершин

1. Положить $j = 1$;
2. В матрице R подсчитываем число ненулевых элементов r_i ;
3. Упорядочим вершины графа в порядке не возрастания r_i ;
4. Просматривая последовательность слева направо, красить в цвет j каждую неокрашенную вершину, не смежную с уже окрашенными в этот цвет;
5. Если остались неокрашенные вершины, то удалить из матрицы R строки и столбцы, соответствующие окрашенным вершинам. Положить $j = j + 1$ и перейти к п. 2, иначе, задача решена.



$R =$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	r_i
x_1	0	1	0	0	1	1	1	0	4
x_2		0	1	0	0	1	1	0	4
x_3			0	1	0	0	0	1	3
x_4				0	1	0	1	0	3
x_5					0	1	0	0	3
x_6						0	0	0	3
x_7							0	0	3
x_8								0	1

1. Положим $j = 1$;

2. Упорядочим вершины графа в порядке не возрастания r_i .

$x_{1'}, x_{2'}, x_{3'}, x_{4'}, x_{5'}, x_{6'}, x_{7'}, x_{8'}$;

3. Красим в первый цвет вершины x_1 и x_3 . Вершины x_4 и x_8 смежны вершине x_3 , остальные – смежны вершине x_1 ;

4. Остались неокрашенные вершины, поэтому удалим из матрицы R строки и столбцы, соответствующие вершинам x_1 и x_3 . Положим $j = j + 1 = 2$.

	x_2	x_4	x_5	x_6	x_7	x_8	r_i
x_2	0	0	0	1	1	0	2
x_4		0	1	0	1	0	2
x_5			0	1	0	0	2
x_6				0	0	0	2
x_7					0	0	2
x_8						0	0

5. Упорядочим вершины графа в порядке не возрастания r_i :

$x_2, x_4, x_5, x_6, x_7, x_8$;

6. Красим во второй цвет вершины x_2, x_4 и x_8 . Вершины x_5 и x_7 смежны вершине x_4 , вершина x_6 смежна вершине x_2 ;

	x_5	x_6	x_7	r_i
x_5	0	1	0	1
x_6		0	0	1
x_7			0	0

7. Остались неокрашенные вершины, удалим из матрицы R строки и столбцы, соответствующие вершинам x_2, x_4 и x_8 . Положим $j = j + 1 = 3$.

8. Упорядочим вершины графа в r_i : x_5, x_6, x_7 .

9. Красим в третий цвет вершины x_5 и x_7 . Вершины x_6 и x_5 смежны; осталась неокрашенная вершина, удалим из матрицы R строки и столбцы, соответствующие вершинам x_5 и x_7 .

10. В четвертый цвет окрашиваем вершину x_6 .

Все вершины окрашены.

Достоинство алгоритма – быстродействие. Недостаток – не оптимальность.

Для раскраски вершин графа приближенным алгоритмом потребовалось четыре цвета. А хроматическое число графа χ равно 3. Действительно, если в первый цвет окрасить вершины x_1, x_4 и x_8 , во второй – x_2 и x_5 , то в третий можно окрасить оставшиеся вершины x_3, x_6 и x_7 .

Кратчайшие пути

Пусть дан граф $G(X, \Gamma)$, ребрам которого приписаны веса, заданные матрицей $C = \|c_{ij}\|_{m \times m}$. **Задача о кратчайшем пути** состоит в нахождении пути с минимальным суммарным весом от начальной вершины $s \in X$ до конечной $t \in X$ или от начальной вершины $s \in X$ до всех остальных, при условии, что такие пути существуют.

Решением алгоритм Дейкстры. Он основан на приписывании вершинам временных пометок, дающих верхнюю границу длины пути от s к этой вершине. Эти пометки постепенно уточняются, и на каждом шаге итерации точно одна из временных пометок становится постоянной. Это указывает на то, что пометка уже не является верхней границей, а дает точную длину кратчайшего пути от s к рассматриваемой вершине.

Алгоритм работает только для графов без ребер отрицательного веса.

Пусть $l(x_i)$ пометка вершины x_i , а $l(x_i)^+$ - постоянная пометка вершины.

1. Положить $l(s)=0^+$ и считать эту пометку постоянной. Положить $l(x_i)=\infty$ для всех $x_i \neq s$ и считать их временными. Положить $p=s$.

2. Для всех $x_i \in \Gamma p$, пометки которых временные, изменить пометки в соответствии со следующим выражением

$$l(x_i) = \min[l(x_i), l(p) + c(p, x_i)].$$

3. Среди всех вершин с временными пометками найти такую, для которой $l(x_i^*) = \min[l(x_i)]$.

4. Считать пометку вершины x_i^* постоянной $l(x_i^*)^+$ и положить $p=x_i^*$. (Если надо найти лишь путь от s до t).

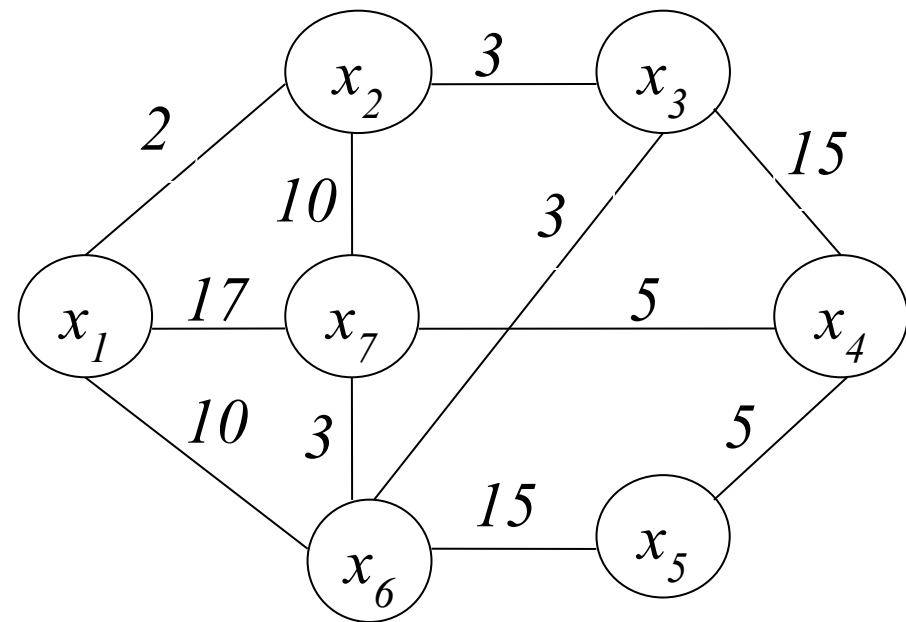
Если $p=t$, то $l(p)$ – длина кратчайшего пути, конец. Если $p \neq t$, перейти к п.2.

6. (Если надо найти путь от s до всех остальных вершин).

Если все вершины имеют постоянные пометки, то конец, если есть временные пометки, то перейти к п.2.

Сами пути можно получить при помощи рекурсивной процедуры с использованием соотношения: $l(x_i') + c(x_i', x_i) = l(x_i)$, где x_i' – вершина, непосредственно предшествующая вершине x_i в кратчайшем пути от s к x_i .

Заданы взвешенный граф $G(X, \Gamma)$ и матрица весов $C = \|c_{ij} 7 \times 7\|$.
 Необходимо найти кратчайшие пути от начальной вершины x_1 ко всем остальным вершинам.



$$C = \begin{array}{c|ccccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ \hline x_1 & 0 & 2 & & & & 10 & 17 \\ x_2 & 2 & 0 & 3 & & & & 10 \\ x_3 & & 3 & 0 & 15 & & 3 & \\ x_4 & & & 15 & 0 & 5 & & 5 \\ x_5 & & & & 5 & 0 & 15 & \\ x_6 & 10 & & 3 & & 15 & 0 & 3 \\ x_7 & 17 & 10 & & 5 & & 3 & 0 \end{array}$$

1. $l(x_1) = 0^+$; $l(x_i) = \infty$, для всех $i \neq 1$, $p = x_1$.

Результаты итерации запишем в таблицу.

			1	2			1	2	3
$L=$	x_1	0^+	x_1	0^+		x_1	0^+		
	x_2	∞	x_2	∞	2^+	x_2	∞	2^+	
	x_3	∞	x_3	∞	∞	x_3	∞	∞	5^+
	x_4	∞	x_4	∞	∞	x_4	∞	∞	∞
	x_5	∞	x_5	∞	∞	x_5	∞	∞	∞
	x_6	∞	x_6	∞	10	x_6	∞	10	10
	x_7	∞	x_7	∞	17	x_7	∞	17	12

2. $\Gamma p = \{x_2, x_6, x_7\}$ – все пометки временные, уточним их:

$$l(x_2) = \min[\infty, 0^+ + 2] = 2;$$

$$l(x_6) = \min[\infty, 0^+ + 10] = 10;$$

$$l(x_7) = \min[\infty, 0^+ + 17] = 17.$$

3. $l(x_i^*) = \min[l(x_i)] = l(x_2) = 2.$

4. x_2 получает постоянную пометку $l(x_2) = 2^+,$

5. Не все вершины имеют постоянные пометки, $\Gamma p = \{x_1, x_3, x_7\}$ – временные пометки имеют вершины x_3, x_7 уточняем их:

$$l(x_3) = \min[\infty, 2^+ + 3] = 5; \quad l(x_7) = \min[17, 2^+ + 10] = 12.$$

$$6. l(x_i^*) = \min[l(x_i)] = l(x_3) = 5.$$

$$7. l(x_3) = 5^+, p=x_3.$$

8. Не все вершины имеют постоянные пометки, $\Gamma p = \{x_2, x_4, x_6\}$ – временные пометки имеют вершины x_4, x_6 уточняем их:

$$l(x_4) = \min[\infty, 5^+ + 15] = 20; \quad l(x_6) = \min[10, 5^+ + 3] = 8.$$

$$9. l(x_i^*) = \min[l(x_i)] = l(x_6) = 8.$$

$$10. l(x_6) = 8^+, p=x_6.$$

11. $\Gamma p = \{x_1, x_5, x_7\}$ – временные пометки имеют вершины x_5, x_7 уточняем их: $l(x_5) = \min[\infty, 8^+ + 15] = 23; \quad l(x_7) = \min[12, 8^+ + 3] = 11.$

	1	2	3	4
x_1	0^+			
x_2	∞	2^+		
x_3	∞	∞	5^+	
x_4	∞	∞	∞	20
x_5	∞	∞	∞	∞
x_6	∞	10	10	8^+
x_7	∞	17	12	12

	1	2	3	4	5
x_1	0^+				
x_2	∞	2^+			
x_3	∞	∞	5^+		
x_4	∞	∞	∞	20	20
x_5	∞	∞	∞	∞	23
x_6	∞	10	10	8^+	
x_7	∞	17	12	12	11^+

12. $l(x_i^*) = \min[l(x_i)] = l(x_7) = 11.$

13. $l(x_7) = 11^+, p=x_7.$
14. Не все пометки постоянные, $\Gamma p = \{x_1, x_2, x_4, x_6\}$ – временную пометку имеет вершина x_4 уточняем ее: $l(x_4) = \min[20, 11^+ + 5] = 16.$
15. $l(x_i^*) = \min[l(x_i)] = l(x_4) = 16.$

16. $l(x_4) = 16^+, p=x_4.$
17. Не все пометки постоянные, $\Gamma p = \{x_3, x_5, x_7\}$ – временную пометку имеет вершина x_5 уточняем ее: $l(x_5) = \min[23,$

~~$16^+ + 5]$~~ $21. l(x_5) = 21.$

19. $l(x_5) = 21^+, p=x_5.$
20. Все пометки постоянные.

$L =$

	1	2	3	4	5	6
x_1	0^+					
x_2	∞	2^+				
x_3	∞	∞	5^+			
x_4	∞	∞	∞	20	20	16^+
x_5	∞	∞	∞	∞	23	23
x_6	∞	10	10	8^+		
x_7	∞	17	12	12	11^+	

$L =$

	1	2	3	4	5	6	7
x_1	0^+						
x_2	∞	2^+					
x_3	∞	∞	5^+				
x_4	∞	∞	∞	20	20	16^+	
x_5	∞	∞	∞	∞	23	23	21^+
x_6	∞	10	10	8^+			
x_7	∞	17	12	12	11^+		

Кратчайшие расстояния от вершины x_1 до всех вершин найдены.

Как найти кратчайший путь до конкретной вершины, покажем на примере вершины x_5 .

$$21 = l(x_4) + c(x_4, x_5) = 16 + 5, \quad 21 \neq l(x_6) + c(x_6, x_5) = 8 + 15.$$

Это означает, что в вершину x_5 мы попали из вершины x_4 .

$$\text{Далее, } l(x_4) = 16, \Gamma x_4 = \{x_3, x_5, x_7\}, \quad 16 \neq l(x_3) + c(x_3, x_4) = 5 + 15,$$

$$16 \neq l(x_5) + c(x_5, x_4) = 21 + 15, \quad 16 = l(x_7) + c(x_7, x_4) = 11 + 5.$$

Это означает, что в вершину x_4 мы попали из вершины x_7 .

$$\text{Далее, } l(x_7) = 11, \Gamma x_7 = \{x_1, x_4, x_6\}, \quad 11 \neq l(x_1) + c(x_1, x_7) = 0 + 17,$$

$$11 \neq l(x_4) + c(x_4, x_7) = 16 + 5, \quad 11 = l(x_6) + c(x_6, x_7) = 8 + 3.$$

Это означает, что в вершину x_7 мы попали из вершины x_6 .

$$\text{Далее, } l(x_6) = 8, \Gamma x_6 = \{x_1, x_3, x_5, x_7\}, \quad 8 \neq l(x_1) + c(x_1, x_6) = 0 + 10,$$

$$8 = l(x_3) + c(x_3, x_6) = 5 + 3, \quad 8 \neq l(x_5) + c(x_5, x_6) = 21 + 15, \quad 8 \neq l(x_7) + c(x_7, x_6) = 11 + 3.$$

Это означает, что в вершину x_6 мы попали из вершины x_3 .

$$\text{Далее, } l(x_3) = 5, \Gamma x_3 = \{x_2, x_4, x_6\}, \quad 5 = l(x_2) + c(x_2, x_3) = 2 + 3,$$

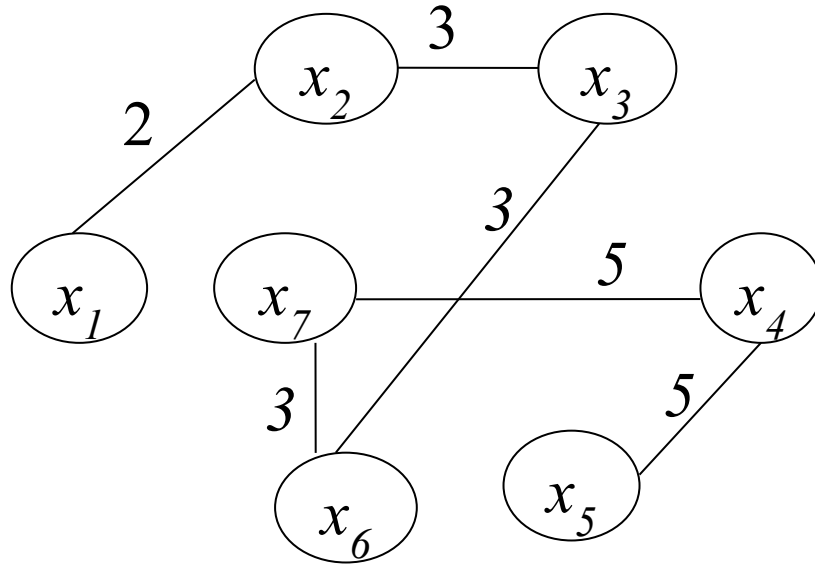
$$5 \neq l(x_4) + c(x_4, x_3) = 16 + 15, \quad 5 \neq l(x_6) + c(x_6, x_3) = 8 + 3.$$

Это означает, что в вершину x_3 мы попали из вершины x_2 .

Далее, $l(x_2) = 2$, $\Gamma x_2 = \{x_1, x_3, x_7\}$, $2 = l(x_1) + c(x_1, x_2) = 0 + 2$,
 $2 \neq l(x_3) + c(x_3, x_2) = 5 + 3$, $2 \neq l(x_7) + c(x_7, x_2) = 11 + 10$.

Это означает, что в вершину x_2 мы попали из вершины x_1 .

Кратчайший путь от вершины x_1 до вершины x_5 найден .



Путь с наибольшей пропускной способностью.

Каждое ребро графа имеет пропускную способность q_{ij} и требуется найти путь от s к t с наибольшей пропускной способностью. Пропускная способность пути P определяется ребром из P с наименьшей пропускной способностью, т.е.

$$Q(P) = \min_{(x_i, x_j) \in P} [q_{ij}].$$

Определение. Если множество вершин графа $G(X, U)$ разбить на два подмножества X_1 и X_2 (где $X = X_1 \cup X_2$), то множество ребер графа, одни концевые вершины которых лежат в X_1 , а другие в X_2 , называется **разрезом графа G** .

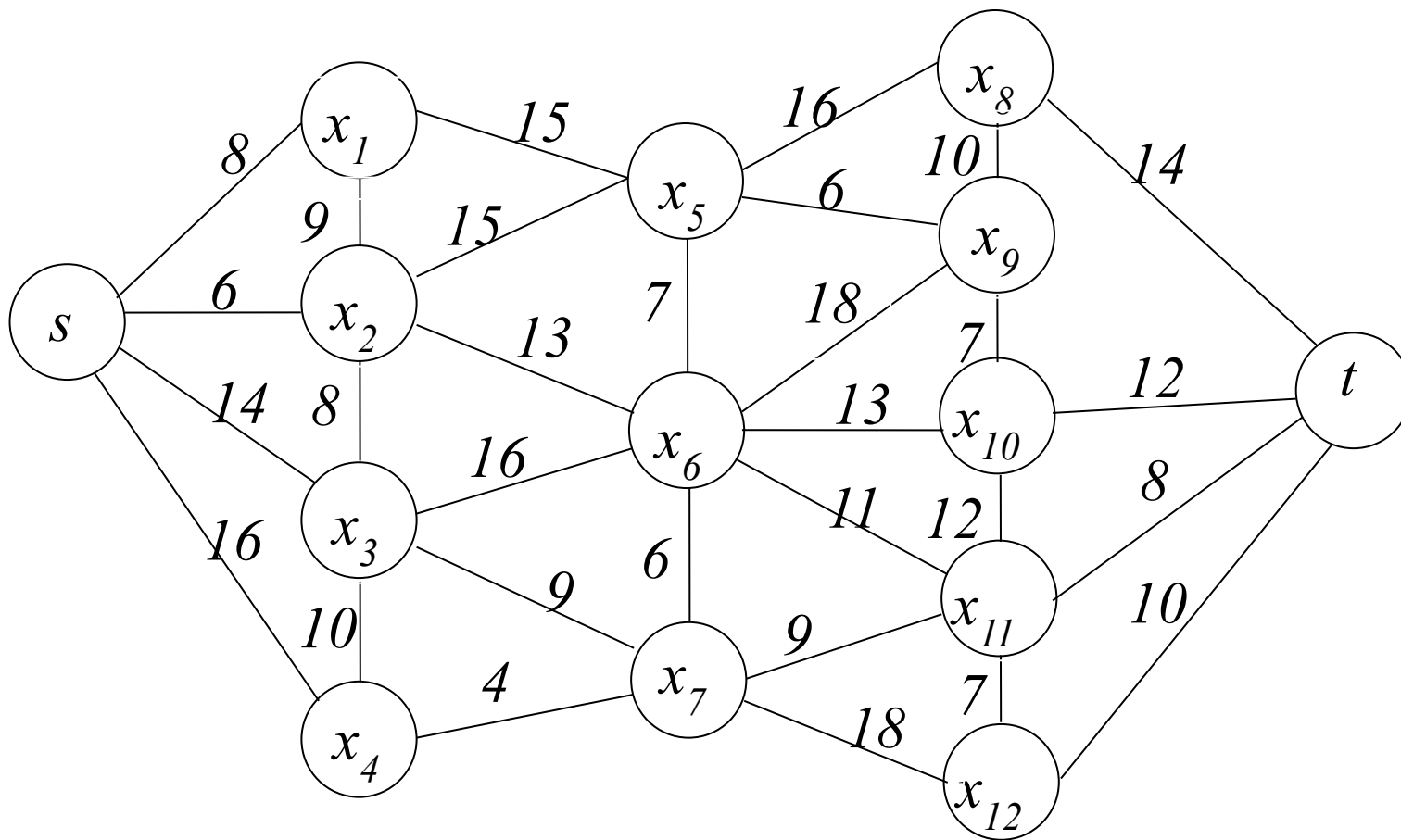
Теорема Форда – Фалкерсона. Пропускная способность пути с наибольшей пропускной способностью от s к t равна

$$Q(P) = \min_K \{ \max_{(x_i, x_j) \in K} [q_{ij}] \}, \quad \text{где } K - \text{любой } (s-t) \text{ разрез.}$$

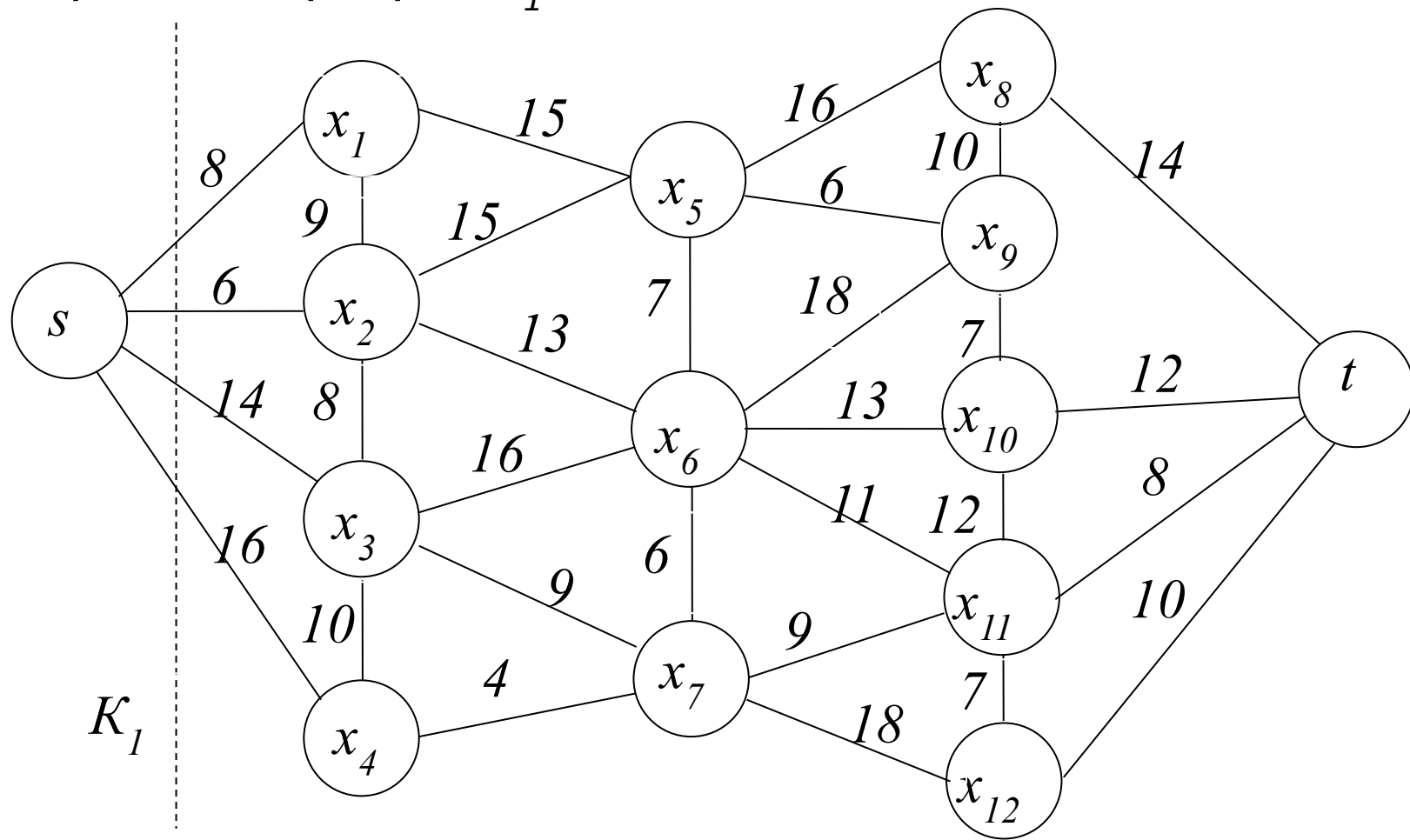
Алгоритм Франка – Фриша

1. Взять $(s-t)$ разрез $K_1 = (\{s\}, X \setminus \{s\})$ и найти $Q_1 = \max_{(x_i, x_j) \in K_1} [q_{ij}]$.
2. Закоротить все ребра графа (x_i, x_j) с $q_{ij} \geq Q_1$, т.е. заменить вершины x_i и x_j на вершину x , удалив ребро (x_i, x_j) , положить $x_i = x_j = x$.
3. Для полученного графа G_1 выбрать другой $(s-t)$ разрез K_2 и найти $Q_2 = \max_{(x_i, x_j) \in K_2} [q_{ij}]$.
4. Закоротить все ребра графа (x_i, x_j) с $q_{ij} \geq Q_2$. Получить граф G_2 ... и т.д., пока не будут объединены вершины $s-t$.
5. Теперь каждый $(s-t)$ путь в графе G' , образованный вершинами из G и теми ребрами, которые оказались закороченными, будет иметь максимальную пропускную способность.

Найти $(s-t)$ путь с наибольшей пропускной способностью в графе $G(X,U)$



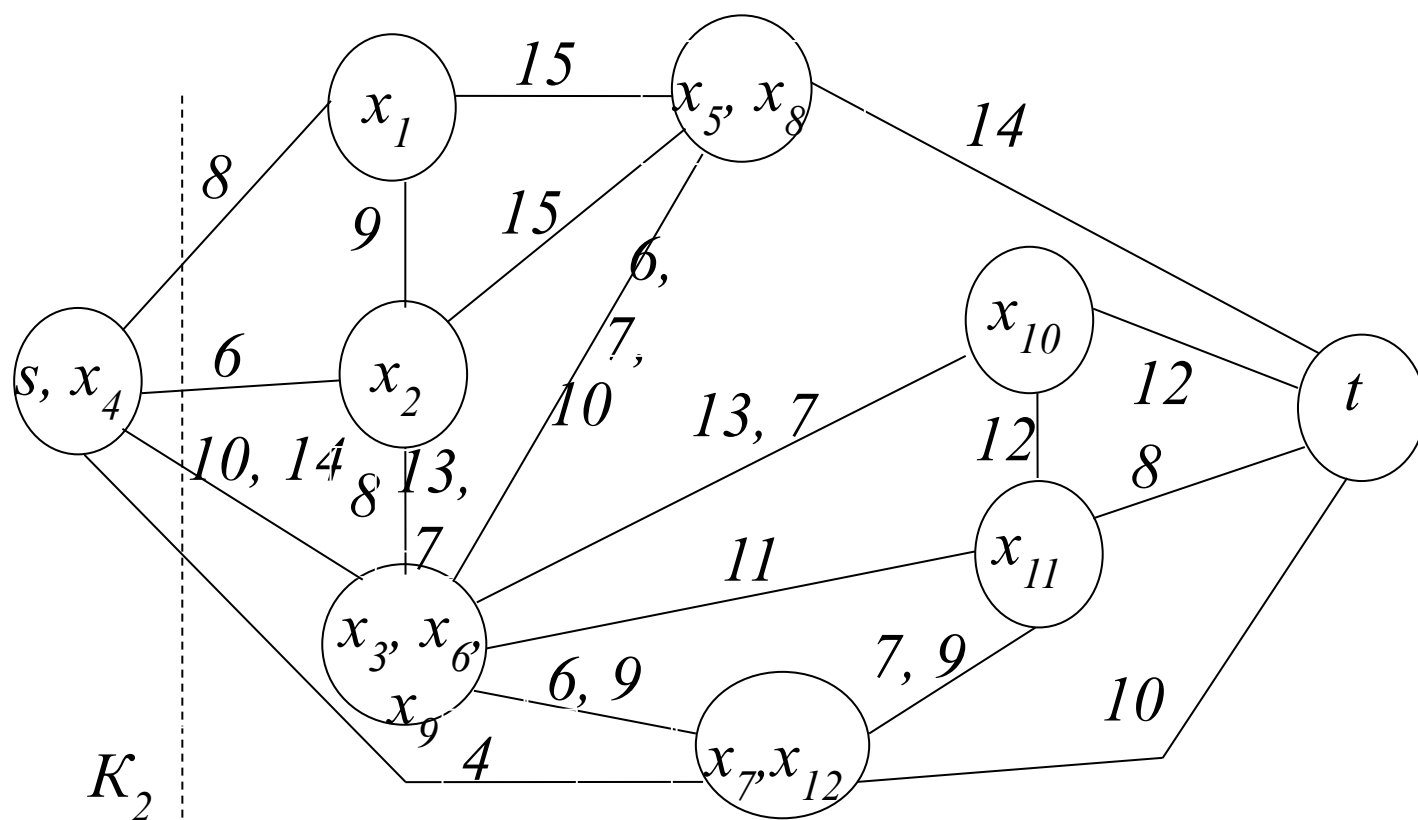
1. Проводим разрез $K_1 = (\{s\}, X \setminus \{s\})$



2. Находим $Q_1 = \max_{(x_i, x_j) \in K_1} [q_{ij}] = 16$.

3. Закорачиваем все ребра графа (x_i, x_j) с $q_{ij} \geq Q_1$.

4. Это ребра (s, x_4) , (x_3, x_6) , (x_5, x_8) , (x_6, x_9) и (x_7, x_{12}) . Получаем граф



5. Проводим разрез K_2 ,

$$Q_2 = \max_{(x_i, x_j) \in K_1} [q_{ij}] = 14.$$

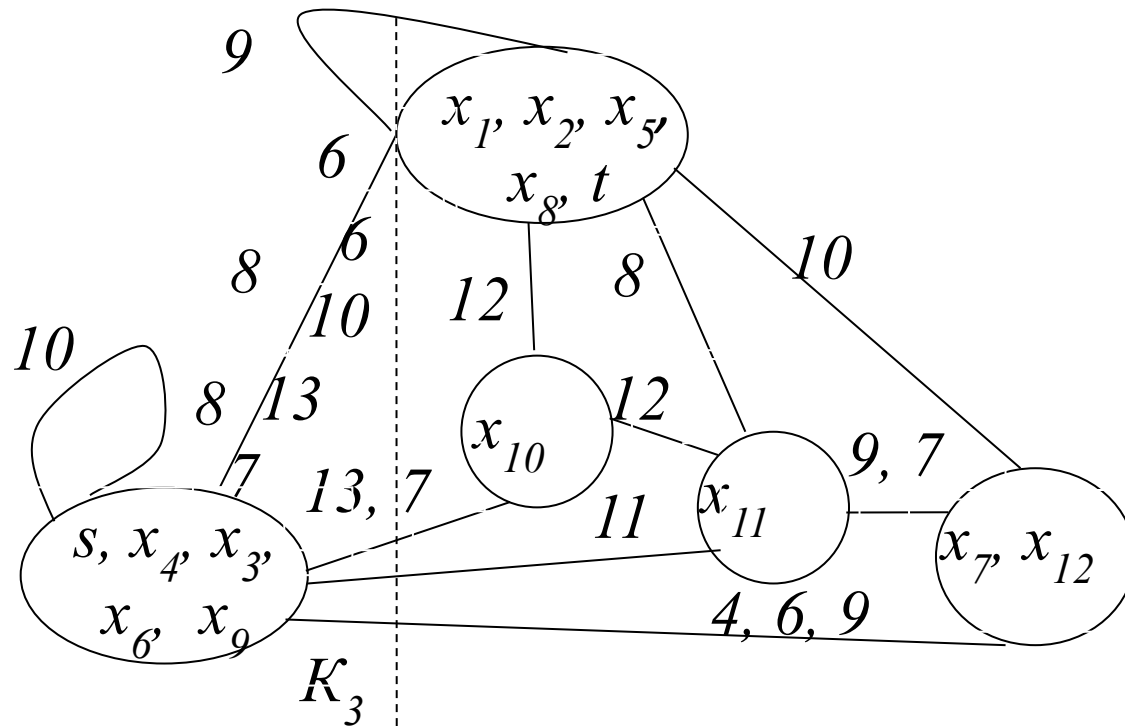
находим

6. Закорачиваем все ребра графа (x_i, x_j) с $q_{ij} \geq Q_2$. Это ребра $(s, x_4, x_3, x_6, x_9), (x_1, x_2, x_5, x_8, t)$. Получаем граф G_2 .

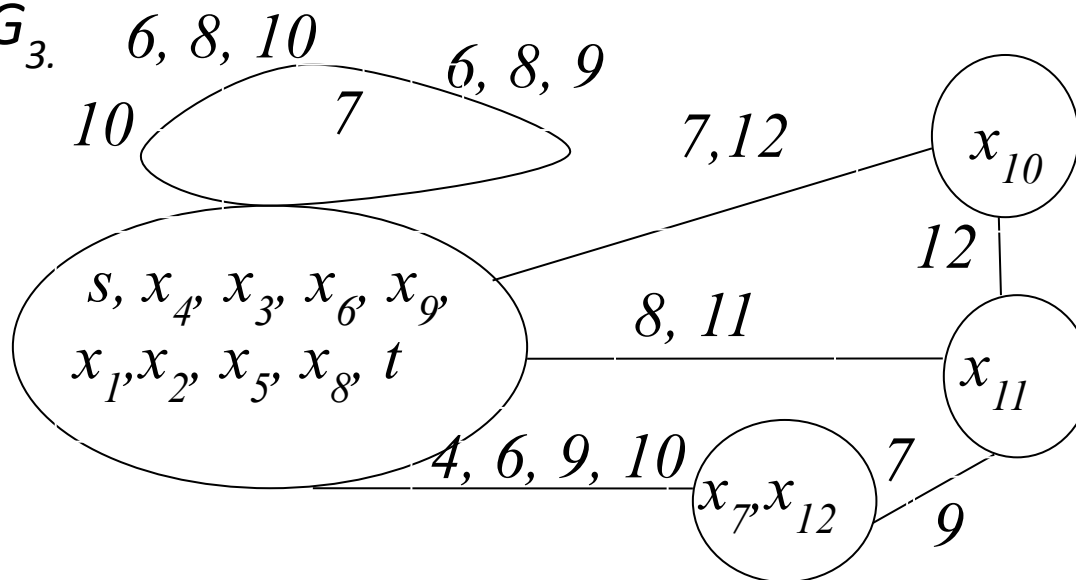
7. Проводим разрез K_3 ,

$$Q_3 = \max_{(x_i, x_j) \in K_1} [q_{ij}] = 13.$$

находим

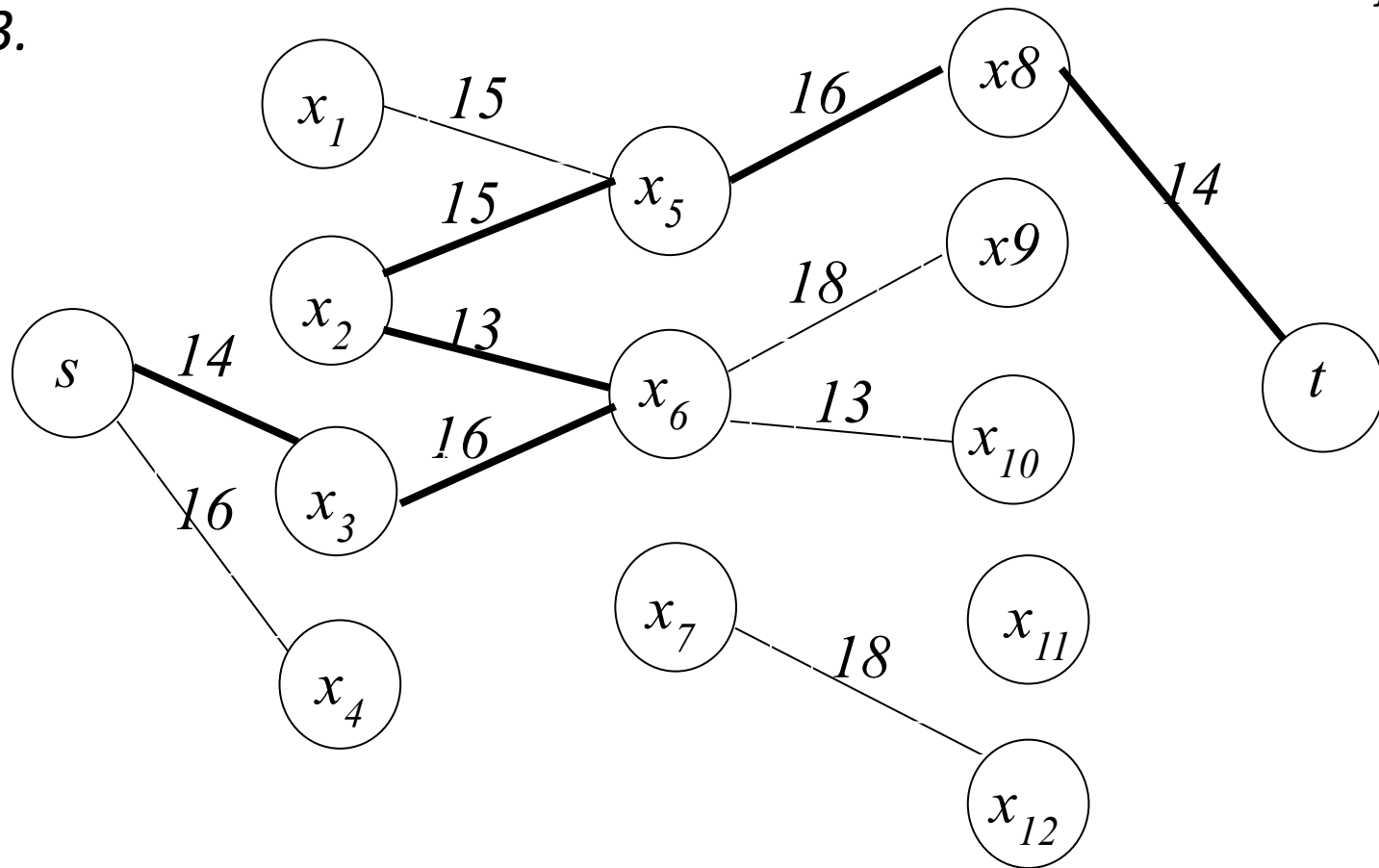


8. Закорачиваем все ребра графа (x_i, x_j) с $q_{ij} \geq Q_3$. Получаем граф G_3 .



9. Вершины s - t объединены. Пропускная способность искомого пути $Q(P)=13$.

10. Строим граф, вершины которого – вершины исходного графа G , а ребра – ребра с пропускной способностью $q_{ij} \geq Q(P)=13$.



Путь с наибольшей пропускной способностью.

Математическая модель задачи размещения

Пусть заданы множество элементов $E = \{e_1, e_2, \dots, e_m\}$ и множество фиксированных позиций для размещения элементов $P = \{p_1, p_2, \dots, p_l\}$ ($l \geq m$, будем считать, что $l = m$). Схема задана матрицей соединений $R = \|r_{ij}\|_{m \times m}$, а расстояние между позициями матрицей расстояний $D = \|d_{ij}\|_{l \times l}$. Для вычисления элементов матрицы D будем пользоваться ортогональной метрикой $d_{ij} = |x_i - x_j| + |y_i - y_j|$ где $(x_i, y_i), (x_j, y_j)$ — координаты позиций.

Учитывая симметричность матриц R и D , запишем выражение для суммарной длины соединений

$$F(P) = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m r_{ij} \times d_{p(i)p(j)}.$$

Таким образом, задача размещения по критерию суммарной длины соединений состоит в минимизации функционала $F(P)$ на множестве перестановок P . Данная задача называется **задачей квадратичного назначения**. Для решения этой задачи предложен ряд алгоритмов, основанных на **методе ветвей и границ**.

Метод ветвей и границ

Метод ветвей и границ – общий алгоритмический метод для нахождения оптимальных решений различных задач оптимизации, особенно дискретной и комбинаторной оптимизации. Метод был впервые предложен Ленд и Дойг в 1960 г. Для решения задач целочисленного линейного

Формализация метода заключается в разбиении всего множества допустимых решений на подмножества и просмотра каждого под-множества с целью выбора оптимального. Для всех решений вычисляется нижняя граница минимального значения целевой функции. Как только нижняя граница становится больше значения целевой функции для наилучшего из ранее известных, подмножество решений, соответствующее этой границе исключается из области решений. Это обеспечивает сокращение перебора.

Поиск продолжается до тех пор, пока не будут исключены все решения, кроме оптимального.

Различные модификации общего метода отличаются способом расчета нижних границ и способом разбиения поля решений. Для описания процесса поиска оптимального размещения строится дерево решений. Ребра первого яруса дерева соответствуют назначениям элемента e_1 в позиции, второго – e_2 и т. д. Произвольному размещению элементов соответствует в дереве решений некоторый путь, исходящий из начальной вершины. Для каждой вершины дерева можно рассчитать нижнюю границу целевой функции для множества путей, связанных с этой вершиной. Если эта граница больше значения целевой функции для известного размещения, то дальнейшее продвижение по дереву в этой вершине прекращается.

Для вычисления нижней границы можно воспользоваться следующим свойством: если $r = \{r_1, r_2, \dots, r_m\}$ и $d = \{d_1, d_2, \dots, d_m\}$ два вектора, то минимум скалярного произведения r и d , т. е. минимум на множестве всех перестановок P , соответствует расположению составляющих вектора r в невозрастающем порядке, а вектора d – в неубывающем.

Пусть скалярное произведение двух векторов rd минимально. В векторе r поменяем местами элементы r_i и r_j . Получим новое скалярное произведение $r'd$. Вычислим разницу между этими произведениями

$$rd - r'd = r_i d_i + r_j d_j - r_i d_j - r_j d_i = r_i (d_i - d_j) - r_j (d_i - d_j) =$$

$= (r_i - r_j)(d_i - d_j)$. Но, по условию $rd - r'd$ минимальное произведение, поэтому $(r_i - r_j)(d_i - d_j) \leq 0$. Это возможно только, если сомножители разных знаков, т.е. или

$$\begin{cases} (r_i - r_j) \leq 0 \\ (d_i - d_j) \geq 0 \end{cases} \text{ или } \begin{cases} (r_i - r_j) \geq 0 \\ (d_i - d_j) \leq 0. \end{cases}$$

Т.е. вектора упорядочены по разному (один по невозрастанию, а другой по неубыванию).

Таким образом, простейшая нижняя граница может быть получена при рассмотрении верхних половин матриц R и D в качестве составляющих векторов r и d длиной $m(m-1) / 2$ и при выполнении указанного выше упорядочивания.

Пусть имеется некоторое частичное размещение q множества элементов E_k на множестве позиций P_k . Тогда нижняя граница складывается из следующих частей

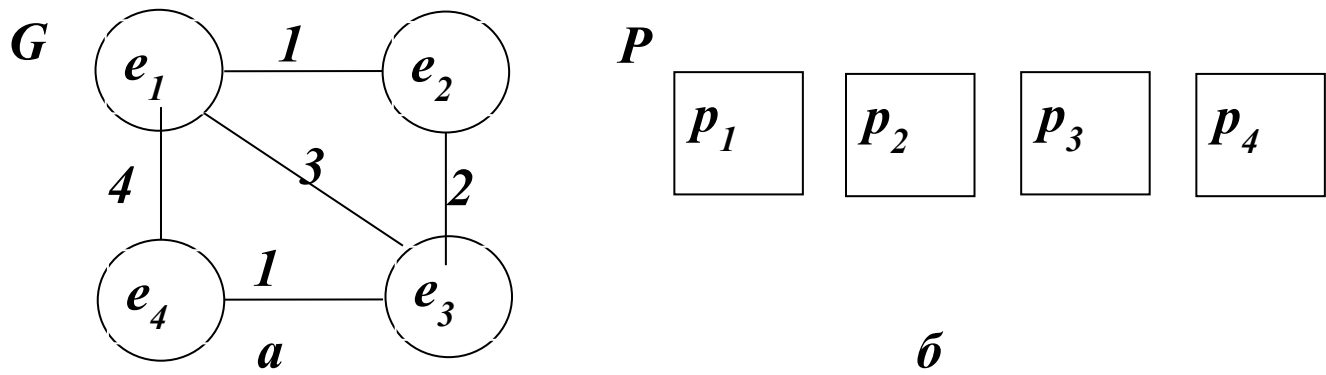
$F(P) = F(q) + w(P) + v(P)$, где

$F(q) = \sum_{e_i \in E_k} \sum_{e_s \in E_k} r_{is} d_{p(i)p(s)}$ - суммарная длина соединений между

размещенными элементами;
 $w(q) = \sum_{e_i \in E_k} \sum_{e_s \notin E_k} r_{is} d_{p(i)p(s)}$ - суммарная длина соединений

между размещенными и неразмещенными элементами;
 $v(q) = \sum_{e_i \notin E_k} \sum_{e_s \notin E_k} r_{is} d_{p(i)p(s)}$ - суммарная длина соединений между

Пример. Разместить элементы, заданные взвешенным графом G (рис. (а)) на множестве позиций P (рис. (б)).



Составим матрицы соединений R графа и расстояний D множества позиций.

	e_1	e_2	e_3	e_4
$R = e_1$	0	1	3	4
e_2		0	2	0
e_3			0	1
e_4				0

	p_1	p_2	p_3	p_4
$D = p_1$	0	1	2	3
p_2		0	1	2
p_3			0	1
p_4				0

Определим нижнюю границу целевой функции для этих исходных данных.

Для этого упорядочим составляющие вектора r в невозрастающем порядке, а вектора d – в неубывающем.

$$r = \{4 \quad 3 \quad 2 \quad 1 \quad 1 \quad 0\}$$

$$d = \{1 \quad 1 \quad 1 \quad 2 \quad 2 \quad 3\}$$

$$r \times d = 4 + 3 + 2 + 2 + 2 + 0 = 13.$$

Это значит, что для этих исходных данных значение целевой функции $F(P)$ не может быть меньше 13.

1. Помещаем элемент e_1 в позицию p_1 .

Т. к. размещен один элемент $F(q) = 0$.

Неразмещенные элементы $\{e_2, e_3, e_4\}$, свободные позиции $\{p_2, p_3, p_4\}$. Составим вектор, соответствующий первой строке матрицы R $r_1 = \{4 \quad 3 \quad 1\}$, и вектор, соответствующий первой строке матрицы D $d_1 = \{1 \quad 2 \quad 3\}$, суммарная длина соединений между размещенным и неразмещенными элементами

$$w(P) = r_1 \times d_1 = 4 + 6 + 3 = 13.$$

Для оценки $v(P)$ вычеркнем из матриц R и D первые строки и столбцы и образуем вектора: $r = \{2 \quad 1 \quad 0\}$ и $d = \{1 \quad 1 \quad 2\}$, соответствующие верхним половинам усеченных матриц R и D .

Получим $v(P) = r \times d = 2 + 1 + 0 = 3$.

Таким образом, нижняя граница $F(P) = 0 + 13 + 3 = 16$.

2. Помещаем элемент e_1 в позицию p_2 . По-прежнему $F(q)=0$.

Неразмещенные элементы $\{e_2, e_3, e_4\}$, свободные позиции $\{p_1, p_3, p_4\}$.

Составим вектор, соответствующий первой строке матрицы R $r_1 = \{4 \ 3 \ 1\}$, и вектор, соответствующий второй строке матрицы D $d_2 = \{1 \ 1 \ 2\}$, суммарная длина соединений между размещенным и неразмещенными элементами

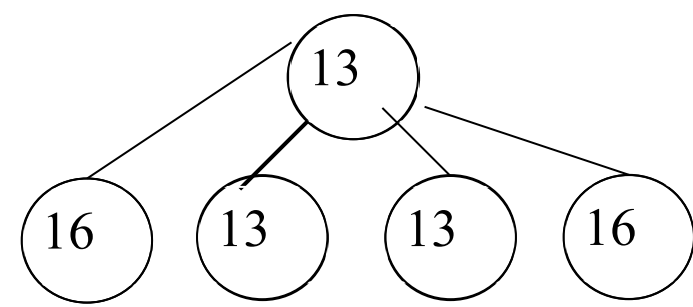
Для оценки $v(P)$ выберем из матрицы R первые строку и столбец, а из матрицы D вторые строку и столбец. Образует вектора:

$r = \{2 \ 1 \ 0\}$ и $d = \{1 \ 2 \ 3\}$, соответствующие верхним половинам усеченных матриц R и D . Получим $v(P) = r \times d = 2 + 2 + 0 = 4$. Таким

образом, нижняя граница $F(P) = 0 + 9 + 4 = 13$.
Очевидно, что ввиду симметричности позиций $(p_1 \text{ и } p_4)$ и $(p_2 \text{ и } p_3)$ будут получены те же результаты для симметричных позиций. На рисунке представлены результаты расчета нижних границ для первого яруса дерева решений.

Назначаем элемент e_1 в позицию p_2 .

3. Помещаем элемент e_2 в позицию p_1 . Размещены два элемента: e_1 в позиции p_2 и e_2 в позиции p_1 , $F(q) =$



Неразмещенные элементы $\{e_3, e_4\}$, свободные позиции $\{p_3, p_4\}$;

$r_1 = \{4 \ 3\}$ и $d_2 = \{1 \ 2\}$, $r_1 \times d_2 = 4 + 6 = 10$;

$r_2 = \{2 \ 0\}$ и $d_1 = \{2 \ 3\}$, $r_2 \times d_1 = 4 + 0 = 4$;

$w(P) = 10 + 4 = 14$.

$r = \{1\}$ и $d = \{1\}$, $v(P) = r \times d = 1$. $F(P) = 1 + 14 + 1 = 16$.

4. Помещаем элемент e_2 в позицию p_3 . Размещены два элемента: e_1 в позиции p_2 и e_2 в позиции p_3 , $F(q) = r_{12} d_{23} = 1$.

Неразмещенные элементы $\{e_3, e_4\}$, свободные позиции $\{p_1, p_4\}$;

$r_1 = \{4 \ 3\}$ и $d_2 = \{1 \ 2\}$, $r_1 \times d_2 = 4 + 6 = 10$;

$r_2 = \{2 \ 0\}$ и $d_3 = \{1 \ 2\}$, $r_2 \times d_3 = 2 + 0 = 2$;

$w(P) = 10 + 2 = 12$.

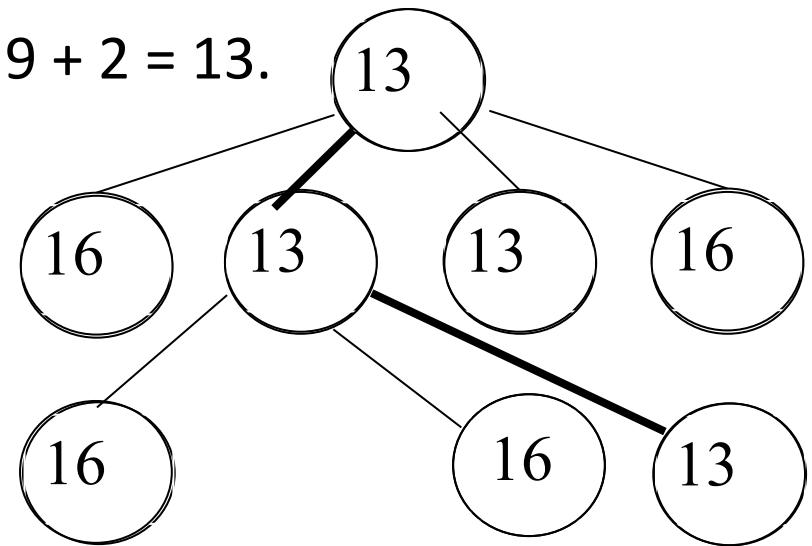
$r = \{1\}$ и $d = \{3\}$, $v(P) = r \times d = 3$. $F(P) = 1 + 12 + 3 = 16$.

5. Помещаем элемент e_2 в позицию p_4 . Размещены два элемента: e_1 в позиции p_2 и e_2 в позиции p_4 , $F(q) = r_{12}d_{24} = 2$.
 Неразмещенные элементы $\{e_3, e_4\}$, свободные позиции $\{p_1, p_3\}$;
 $r_1 = \{4 \ 3\}$ и $d_2 = \{1 \ 1\}$, $r_1 \times d_2 = 4 + 3 = 7$;
 $r_2 = \{2 \ 0\}$ и $d_4 = \{1 \ 3\}$, $r_2 \times d_4 = 2 + 0 = 2$;
 $w(P) = 7 + 2 = 9$.

$r = \{1\}$ и $d = \{2\}$, $v(P) = r \times d = 2$. $F(P) = 2 + 9 + 2 = 13$.

На рисунке представлены результаты расчета нижних границ для двух ярусов дерева решений.

Назначаем элемент e_2 в позицию p_4 .



6. Помещаем элемент e_3 в позицию p_1 . Размещены три элемента: e_1 в позиции p_2 , e_2 в позиции p_4 и e_3 в позиции p_1 ,
 $F(q) = r_{12}d_{24} + r_{13}d_{21} + r_{23}d_{41} = 2 + 3 + 6 = 11$.

Неразмещенный элемент $\{e_4\}$, свободная позиция $\{p_3\}$;

$r_1 = \{4\}$ и $d_2 = \{1\}$, $r_1 \times d_2 = 4$;

$r_2 = \{0\}$ и $d_4 = \{1\}$, $r_2 \times d_4 = 0$;

$r_3 = \{1\}$ и $d_1 = \{2\}$, $r_3 \times d_1 = 2$;

$w(P) = 4 + 0 + 2 = 6$.

Неразмещенный элемент один, $v(P) = 0$. $F(P) = 11 + 6 + 0 = 17$.

7. Помещаем элемент e_3 в позицию p_3 . Размещены три элемента: e_1 в позиции p_2 , e_2 в позиции p_4 и e_3 в позиции p_3 ,

$F(q) = r_{12}d_{24} + r_{13}d_{23} + r_{23}d_{43} = 2 + 3 + 2 = 7$.

Неразмещенный элемент $\{e_4\}$, свободная позиция $\{p_1\}$;

$r_1 = \{4\}$ и $d_2 = \{1\}$, $r_1 \times d_2 = 4$;

$r_2 = \{0\}$ и $d_4 = \{1\}$, $r_2 \times d_4 = 0$;

$r_3 = \{1\}$ и $d_3 = \{2\}$, $r_3 \times d_3 = 2$;

$w(P) = 4 + 0 + 2 = 6$.

Неразмещенный элемент один, $v(P) = 0$. $F(P) = 7 + 6 + 0 = 13$.

На рисунке представлены результаты расчета нижних границ для трех ярусов дерева решений.

Назначаем элемент e_3 в позицию p_3 .

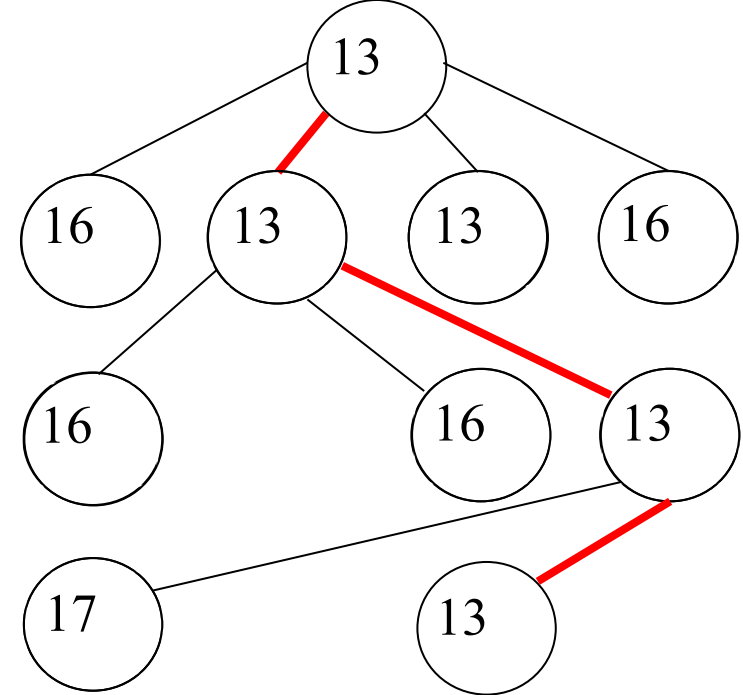
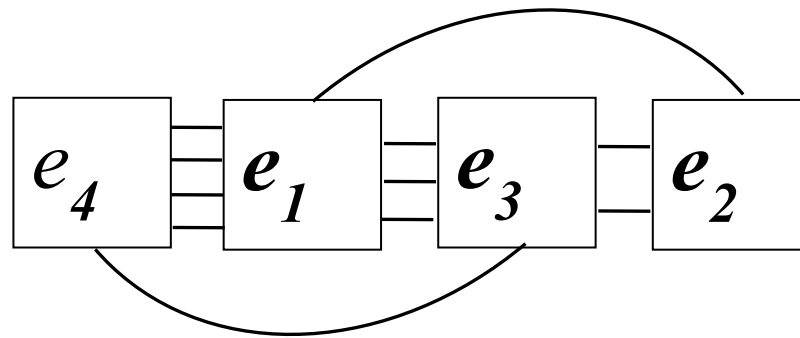
8. Неразмещенный элемент $\{e_4\}$,
свободная позиция $\{p_1\}$.

Помещаем $\{e_4\}$ в позицию $\{p_1\}$.

$$F(q) = r_{12}d_{24} + r_{13}d_{23} + r_{23}d_{43} + r_{14}d_{21} + r_{24}d_{41} + r_{34}d_{31} = 2 + 3 + 2 + 4 + 0 + 2 = 13.$$

$$w(P) = v(P) = 0. F(p) = 13.$$

Получено размещение



Нахождение гамильтонова цикла.

Алгоритм Робертса-Флореса

Цикл, включающий все вершины один раз, называется гамильтоновым.

Алгоритм Робертса-Флореса состоит в следующем.

Некоторая начальная вершина (x_1) выбирается в качестве отправной и образует первый элемент множества S , которое будет хранить уже найденные вершины цепи. К S добавляется первая вершина a , смежная с x_1 , $a \in \Gamma x_1$. Затем к множеству S добавляется первая вершина b , смежная с a , $b \in \Gamma a$, $b \notin S$.

1. $b \in \Gamma a$;

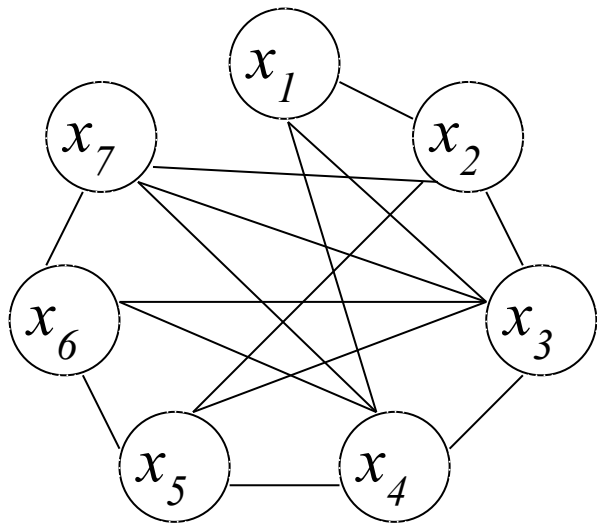
2. $b \notin S$.

Существует две причины, препятствующие включению некоторой вершины на шаге r в множество $S = \{x_1, a, b, \dots, x_r\}$:

1. у x_r нет "возможной" вершины;
2. цепь S имеет длину $|S|=n$. В этом случае:
 - а) есть ребро (x_r, x_1) и гамильтонов цикл найден;
 - б) такого ребра нет и найдена гамильтонова цепь.

В случаях 1 и 2 б) следует прибегнуть к возвращению, которое заключается в удалении из S последней включенной вершины x_r и добавлении к S новой первой "возможной", следующей за x_r вершины. Если не существует никакой возможной вершины, делается следующий шаг возвращения и т.д.

Поиск заканчивается в случае когда S состоит из одной x_1 и нет не рассмотренных "возможных" вершин.



$R(G) =$

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

Нахождение гамильтонова цикла

Включаем в S начальную вершину $S = \{x_1\}$.

Первая "возможная" вершина $x_2 \nabla x_1$, $S = \{x_1, x_2\}$ и т.д. до вершины $S = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$. Ребра (x_7, x_1) нет. Найдена гамильтонова цепь.

Прибегнем к возвращению. Удалим из S вершину x_7 .

$S = \{x_1, x_2, x_3, x_4, x_5, x_6\}$.

У x_6 больше нет "возможных" вершин. Удалим и ее. $S = \{x_1, x_2, x_3, x_4, x_5\}$.

У x_5 больше нет "возможных" вершин. Удалим ее. $S = \{x_1, x_2, x_3, x_4\}$. Следующая "возможная" вершина x_6 . $S = \{x_1, x_2, x_3, x_4, x_6\}$.

Следующая "возможная" вершина x_5 . $S=\{x_1, x_2, x_3, x_4, x_6, x_5\}$. Больше нет "возможных" вершин. Удалим ее.

$S=\{x_1, x_2, x_3, x_4, x_6\}$. Следующая "возможная" вершина x_7 . $S=\{x_1, x_2, x_3, x_4, x_6, x_7\}$.

У x_7 больше нет "возможных" вершин. Удалим из S вершину x_7 .

$S=\{x_1, x_2, x_3, x_4, x_6\}$. У x_6 больше нет "возможных" вершин. Удалим ее. $S=\{x_1, x_2, x_3, x_4\}$. Следующая "возможная" вершина x_7 . $S=\{x_1, x_2, x_3, x_4, x_7\}$.

Следующая "возможная" вершина x_6 . $S=\{x_1, x_2, x_3, x_4, x_7, x_6\}$. Следующая "возможная" вершина x_5 .

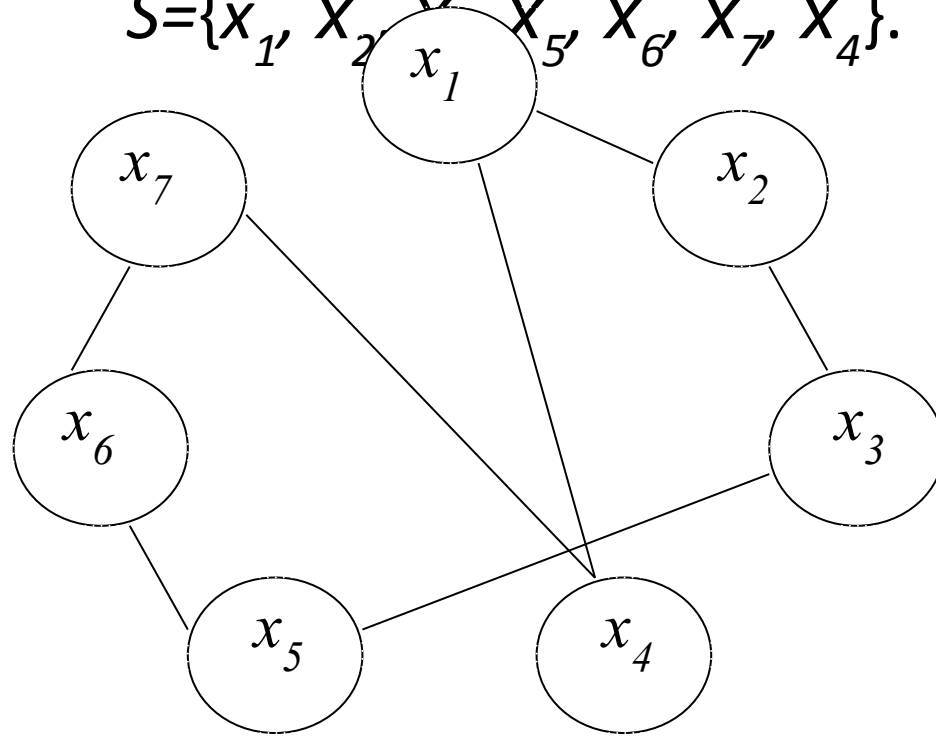
$S=\{x_1, x_2, x_3, x_4, x_7, x_6, x_5\}$.

Ребра (x_5, x_1) нет. Найдена гамильтонова цепь.

Удаляем вершины x_4, x_7, x_6, x_5 . $S=\{x_1, x_2, x_3\}$.

И т.д., пока последней не окажется вершина x_4 , которая образует цикл с вершиной x_1 . Гамильтонов цикл будет

$$S = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_1\}.$$



Нахождение эйлерова цикла. Алгоритм Флери

Эlegantный алгоритм нахождения эйлерова цикла был предложен М. Флери (M. *Fleury*) в 1883 году.

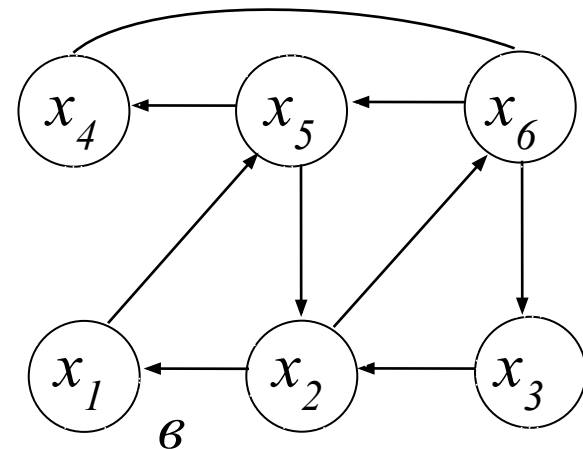
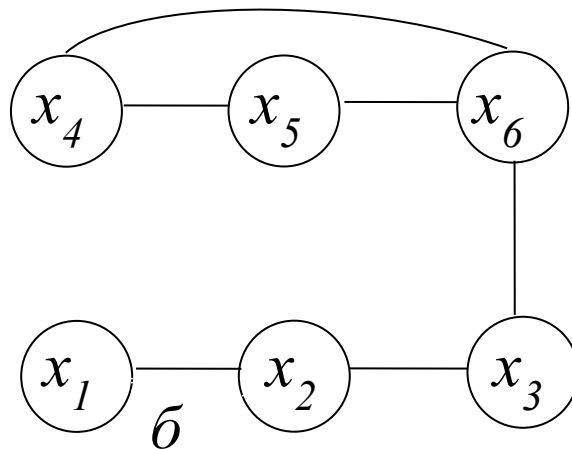
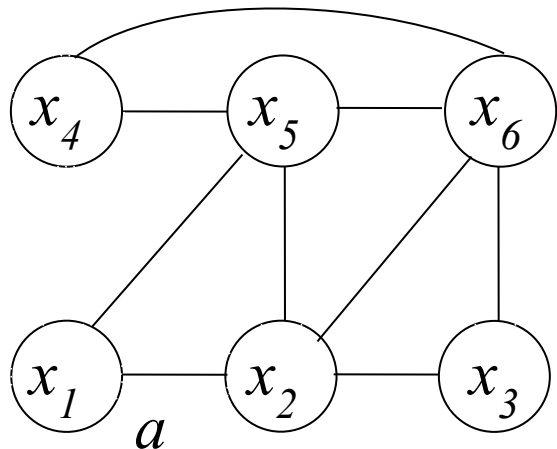
Алгоритм заключается в следующем:

1. Положить текущий граф равным $G(X, U)$, а текущую вершину равной произвольной вершине $x_i \in X$.
2. Выбрать произвольное ребро u_{ij} текущего графа, инцидентное текущей вершине x_i с учетом следующего ограничения: если степень текущей вершины в текущем графе больше 1, нельзя выбирать ребро, удаление которого из текущего графа увеличит число компонент связности в нем (т. е. ребро, являющееся мостом).

3. Назначить текущей x_j вершину, инцидентную ребру u_{ij} . Удалить u_{ij} из текущего графа и внести в список.

5. Если в текущем графе еще остались ребра, то положить $i=j$ и вернуться на шаг 2.

Сложность алгоритма $O(k)$, где $k = |U|$ – число ребер.



Пусть на шаге 1 выбрана вершина x_1 . При выборе на шаге 2 ограничение никак не сказывается; пусть выбрано ребро (x_1, x_5) . На двух следующих итерациях ограничений на выбор по-прежнему не возникает; пусть выбраны ребра (x_5, x_2) и (x_2, x_6) . Тогда текущим графом становится граф, изображенный на рис. (б) (текущая вершина – x_6).

На следующей итерации нельзя выбрать ребро (x_6, x_3) из-за ограничения; пусть выбрано ребро (x_6, x_5) .

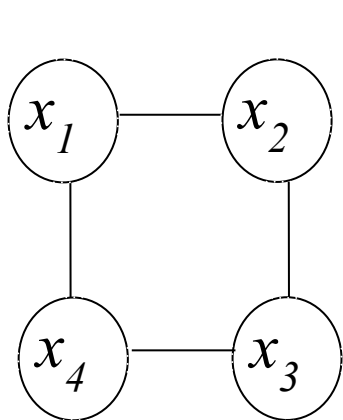
Дальнейший выбор ребер определен однозначно (текущая вершина всегда будет иметь степень 1), так что в итоге будет построен следующий эйлеров цикл (рис. (в)): $x_1 \rightarrow x_5 \rightarrow x_2 \rightarrow x_6 \rightarrow x_5 \rightarrow x_4 \rightarrow x_6 \rightarrow x_3 \rightarrow x_2 \rightarrow x_1$.

Сравнение эйлеровых и гамильтоновых циклов

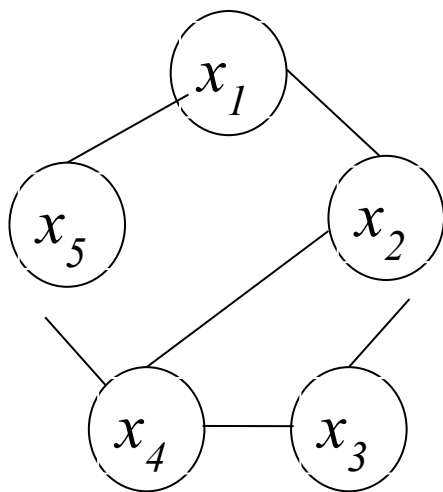
Несмотря на внешнюю схожесть определений эйлерова и гамильтонова циклов, задачи нахождения циклов этих двух типов в данном графе разительно отличаются по сложности.

Задача о нахождении эйлерова цикла – это простая с математической точки зрения задача: существует эффективный критерий существования эйлерова цикла (теорема Эйлера); если критерий выполнен, имеется эффективный алгоритм для нахождения цикла (например, алгоритм Флери). Ни критерия гамильтоновости графа, ни эффективного алгоритма нахождения гамильтонова цикла в произвольном гамильтоновом графе, неизвестно (и скорее всего, не существует). Задача о нахождении гамильтонова цикла – это NP-трудная задача.

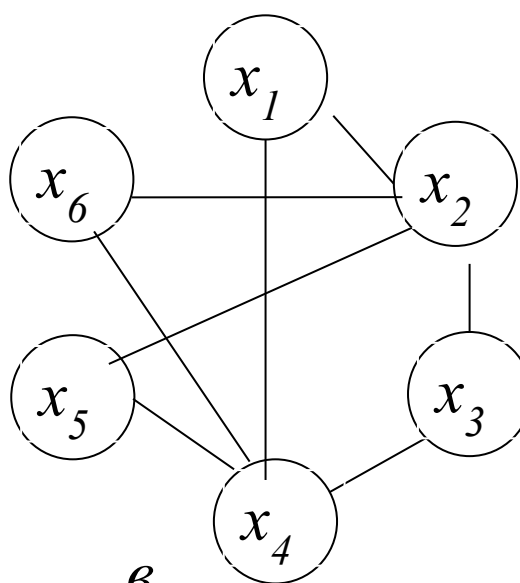
Следующие четыре графа демонстрируют отсутствие тесной взаимосвязи между существованием эйлеровых и гамильтоновых циклов .



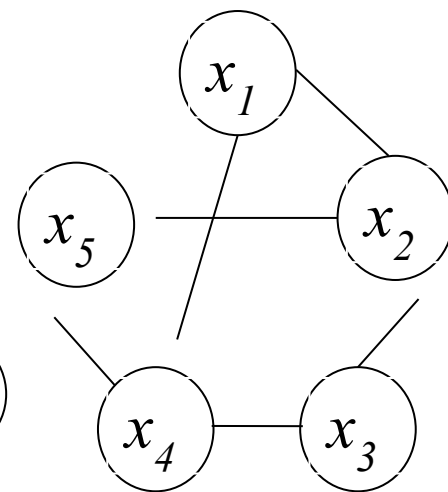
a



б



в



г

Графы: эйлеров и гамильтонов (а); неэйлеров и гамильтонов (б); эйлеров и негамильтонов (в); неэйлеров и негамильтонов (г).

Однако, двойственность между эйлеровыми и гамильтоновыми циклами (замена вершины на ребро и наоборот) приводит к тесной связи между этими двумя понятиями в применении к графу G и соответствующему ему реберному графу G_1 , определяемому ниже.

Реберный граф G_1 графа G имеет столько же вершин, сколько ребер у графа G . Ребро между двумя вершинами графа G_1 существует тогда и только тогда, когда ребра графа G , соответствующие этим двум вершинам, смежны.

Верны два следующих утверждения о взаимоотношении между эйлеровыми и гамильтоновыми циклами, принадлежащие Ф. Харари.

1. Если граф G имеет эйлеров цикл, то граф G_1 имеет как эйлеров, так и гамильтонов циклы.
2. Если граф G имеет гамильтонов цикл, то граф G_1 также имеет гамильтонов цикл.

Обращение этих утверждений неверно!

Алгоритмы построения минимальных связывающих деревьев

Для построения МСД разработан ряд полиномиальных алгоритмов.

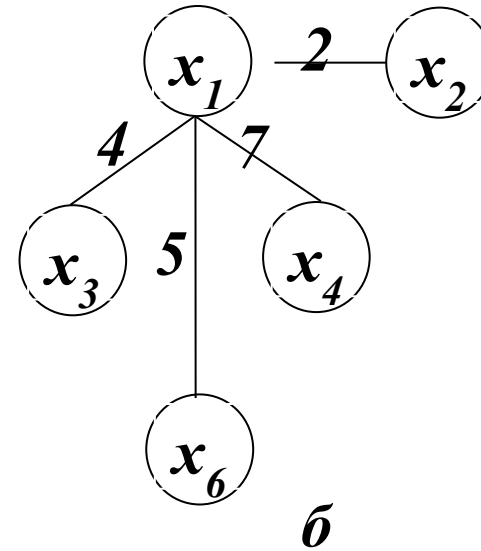
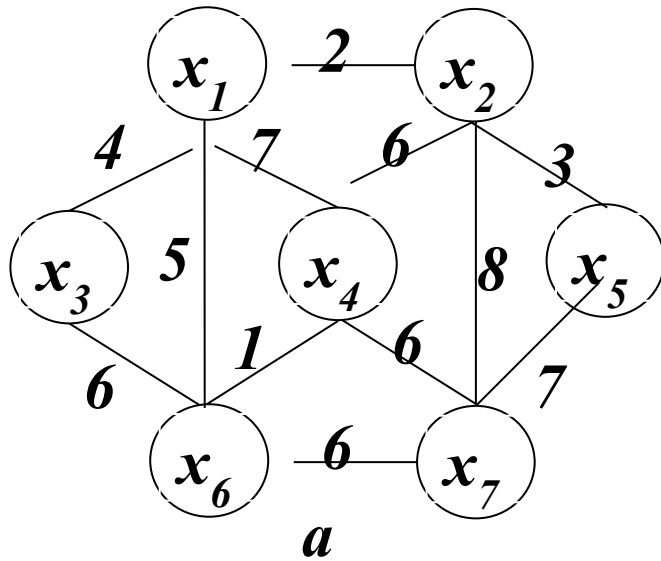
Алгоритм Прима

Алгоритм Прима (*R.C. Prim*) относится к так называемым "жадным" алгоритмам. Жадные алгоритмы действуют, используя в каждый момент лишь часть исходных данных и принимая лучшее решение на основе этой части. В нашем случае мы будем на каждом шаге рассматривать множество ребер, допускающих присоединение к уже построенной части связывающего дерева, и выбирать из них ребро с наименьшим весом. Повторяя эту процедуру, мы получим остовное дерево с наименьшим весом.

Начнем с произвольной вершины графа x_i и включим ее в остовное дерево. Из всех вершин, соединенных с данной ($x_j \in \Gamma x_i$), ищем вершину, соединенную ребром с наименьшим весом.

Это ребро вместе с новой вершиной добавляется в дерево. Из вершин, не вошедших в дерево, ищем вершину, соединенную с уже построенной частью остовного дерева ребром с наименьшим весом. Это ребро вместе с новой вершиной добавляется в дерево и т. д. После того, как в дерево попадут все вершины, работа будет закончена.

Пример. Исходный взвешенный граф изображен на рисунке.



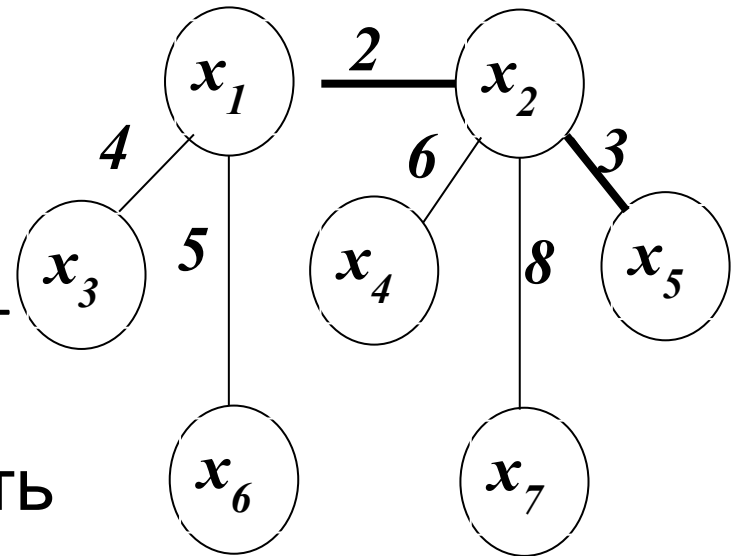
В начале процесса выбираем произвольную вершину, например, x_1 . Выбираем вершины, непосредственно связанные с x_1 .

Ребро наименьшего веса связывает вершины x_1 и x_2 , поэтому к уже построенной части МСД добавляется вершина x_2 вместе с ребром (x_1, x_2) .

При добавлении к дереву вершины x_2 необходимо определить, не следует ли добавить в кандидаты новые ребра. В результате обнаруживаем, что это необходимо проделать с ребрами $(x_2 x_5)$ и $(x_2 x_7)$. Здесь же необходимо проверить, являются ли ребра, ведущие из вершины x_1 , в x_3 , x_4 и x_7 , кратчайшими среди ребер, соединяющих эти вершины с деревом, или есть более удобные ребра, исходящие из x_2 .

Ребро $(x_2 x_4)$ короче ребра $(x_1 x_4)$, и поэтому необходимо его заменить

Наименьший вес из пяти ребер - кандидатов имеет ребро $(x_2 x_5)$, поэтому к дереву нужно добавить его и вершину x_5

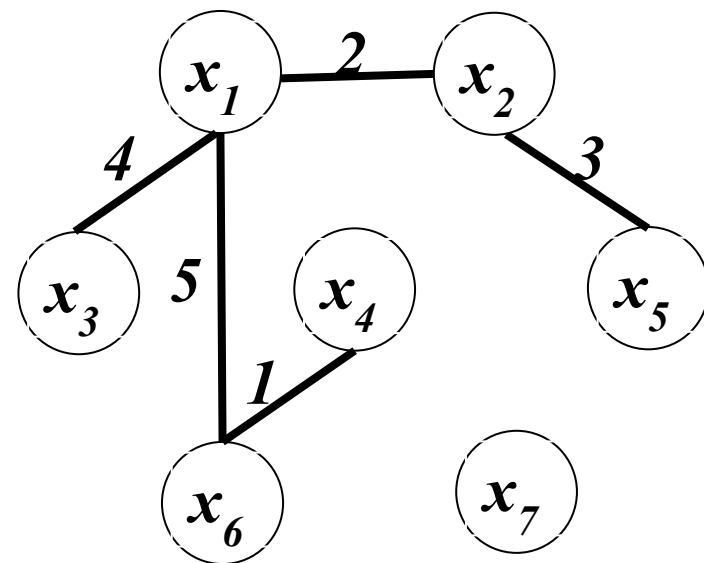
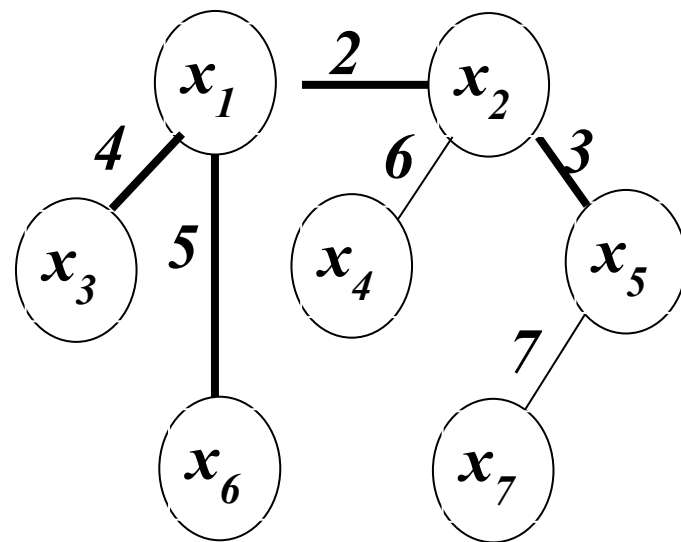


Вес ребра $(x_5 x_7)$ меньше веса ребра $(x_2 x_7)$, поэтому оно замещает последнее.

Из четырех ребер, инцидентных построенной части дерева, наименьший вес имеет ребро $(x_1 x_3)$, поэтому следующим к дереву добавляется оно и вершина x_3

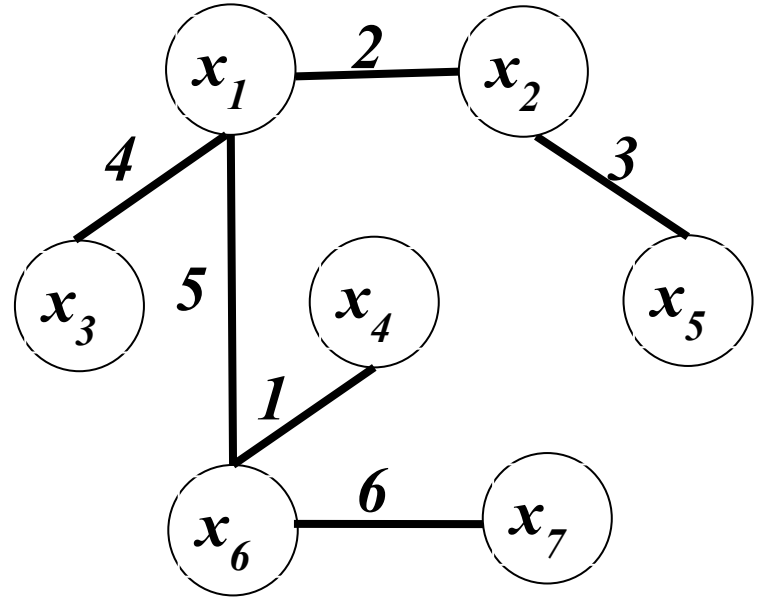
Затем к дереву добавляется ребро $(x_1 x_6)$ вместе с вершиной x_6 .

Поскольку вес ребра $(x_4 x_6)$ меньше веса ребра $(x_2 x_4)$, а вес ребра $(x_6 x_7)$ меньше веса ребра $(x_5 x_7)$, меняем ребра – кандидаты.



Ребро (x_4, x_6) имеет наименьший вес среди оставшихся, и оно добавляется следующим.

Теперь осталось добавить к дереву всего одно ребро (x_6, x_7)



После этого процесс завершается, построено МСД с суммарным весом ребер 21.

Сложность алгоритма Прима равна $O(m^3)$, где $m = |X|$.

Алгоритм Краскала

Алгоритм заключается в следующем.

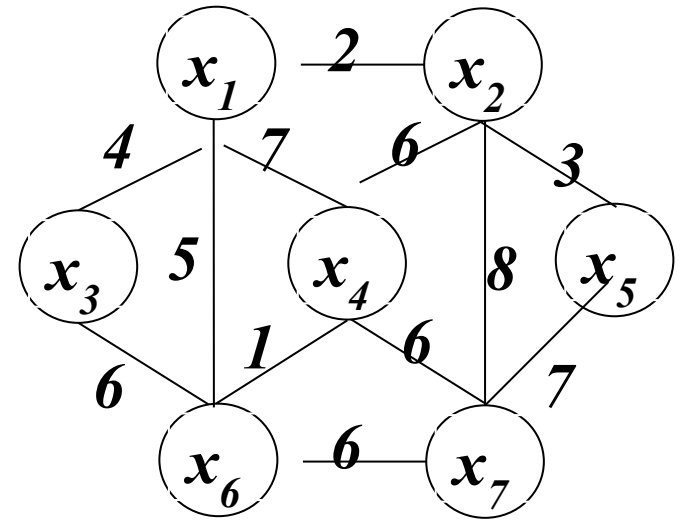
Упорядочим все ребра исходного графа по не убыванию весов.

Просматривая последовательность слева направо, включаем в де-рево каждое ребро, не образующее в дереве цикла. О том, что ребро (x_i, x_j) образует цикл можно судить по тому, что из вершины x_i есть путь в вершину x_j по другим ребрам дерева. Процесс заканчивается, когда в дерево включено $(m - 1)$ ребро.

Построим МСД для графа

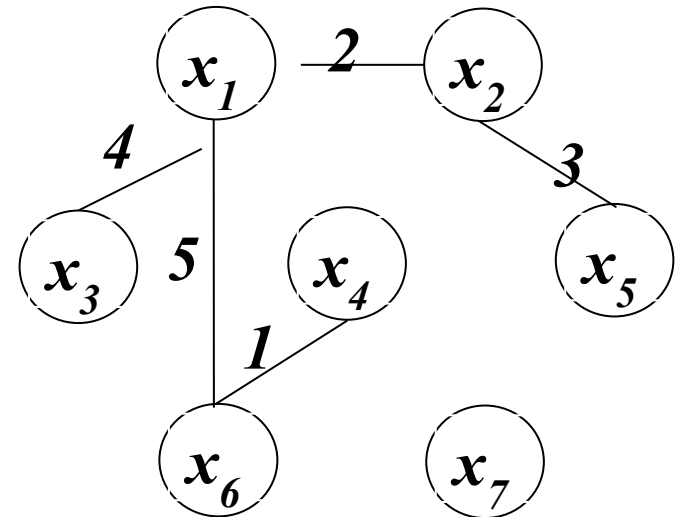
Упорядочим ребра:

$(x_4 x_6), (x_1 x_2), (x_2 x_5), (x_1 x_3), (x_1 x_6), (x_2 x_4), (x_3 x_6), (x_4 x_7), (x_6 x_7), (x_1 x_4), (x_5 x_7), (x_2 x_7).$



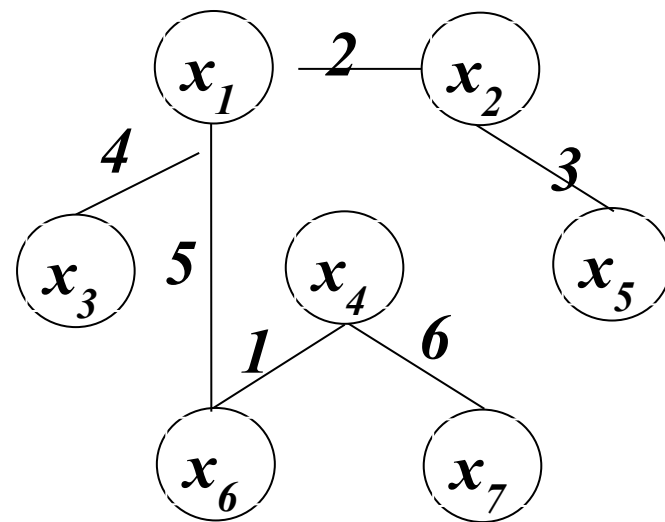
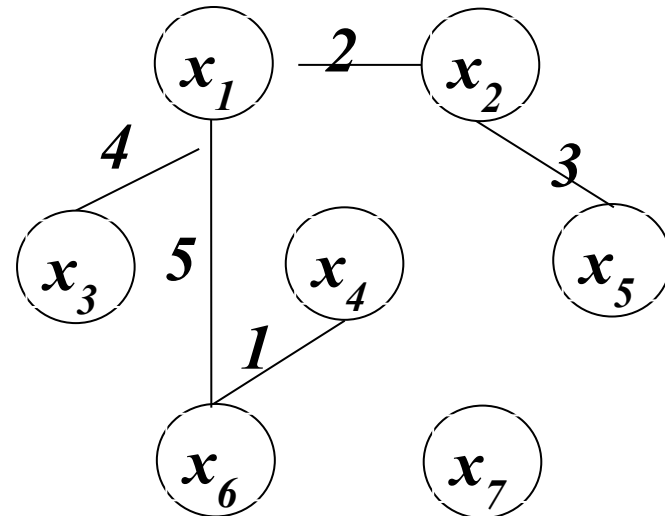
Просматривая список, включаем в дерево ребра $(x_4 x_6), (x_1 x_2), (x_2 x_5), (x_1 x_3), (x_1 x_6)$

Ребра $(x_2 x_4)$ и $(x_3 x_6)$ образуют цикл с уже построенными ребрами, поэтому в дерево не включаются.



Следующее включаемое в
дерево ребро $(x_4 x_7)$.

Дерево построено.



Суммарный вес ребер МСД равен
21.

Сложность алгоритма Краскала
равна $O(n \log n)$, где $n = |U|$.

Планаризация графа

Задача встает при проектировании электронных схем. Электрические цепи печатным способом наносятся на плату из изолирующего материала. Так как наносимые цепи не изолированы, то они не должны пересекаться. Отсюда вытекает вопрос, как расположить контакты на схеме, чтобы можно было без пересечений нанести цепи на плату. А если так сделать нельзя, то каким минимальным числом плат (слоев графа) можно обойтись.

Среди критериев планарности графа наиболее известен **критерий Понтрягина-Куратовского**:

Граф $G(X, U)$ – планарен тогда и только тогда, когда он не содержит подграфов гомеоморфных полному графу K_5 или полному двудольному графу $K_{3,3}$.

Стягивание ребра – это операция удаления ребра из графа, а вершины, инцидентные этому ребру
~~Введение вершины~~ – это операция добавления вершины w в ребро (u, v) , в результате которой получаются два ребра (u, w) (w, v) , а ребро (u, v)
Два графа ~~гомеоморфны~~ **гомеоморфны**, если они могут быть получены из одного и того же графа стягиванием ребер или включением вершин.

Однако критерий Понтрягина-Куратовского неконструктивен (требует перебора). Известны другие критерии, но их также трудно использовать. Разработан ряд алгоритмов определения планарности графа удобных для реализации на ЭВМ.

Критерий Бадера. Граф $G(X, U)$ планарен, если его граф пересечений G' является бихроматическим графом.

Критерий справедлив для графов, имеющих

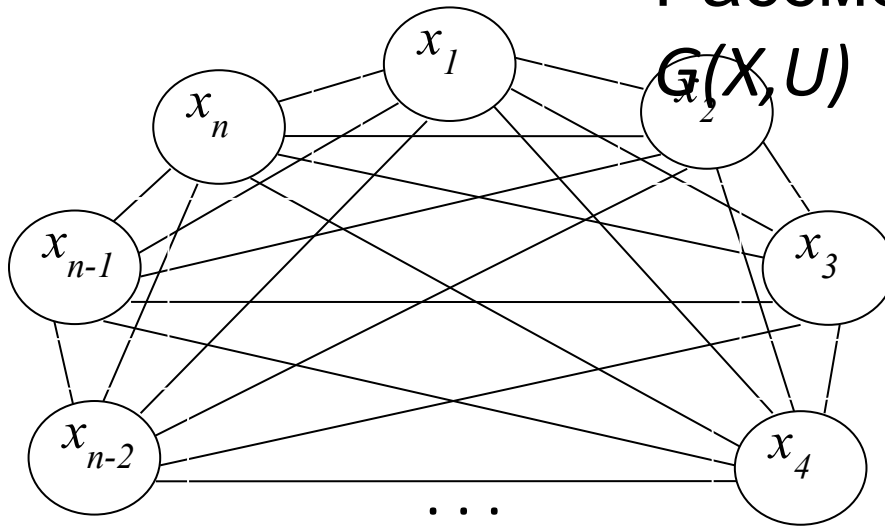
Граф пересечений G' – граф, вершины которого соответствуют ребрам и эти вершины соединены, если ребра пересекаются.

Построение графа пересечений G'

Приняты следующие допущения:

1. Граф G имеет гамильтонов цикл;
2. Два ребра пересекаются только один раз;
3. Ребра, инцидентные одной вершине, не пересекаются;
4. Ребра графа не пересекают ребер гамильтонова цикла;
5. Ребра вершины x могут пересекать ребра

Рассмотрим некоторый граф



p_i – число пересечений
ребер i -ой вершины.

Согласно п. 5 принятых допущений $p_1=0$.

Рассмотрим ребро (x_2x_n) . Число его пересечений с
ребрами вершины x_1

$$p_{2n} = r_{2n} \sum_{i=3}^{n-1} r_{1i}.$$

Ребро (x_2x_{n-1}) пересекает $p_{2(n-1)} = r_{2(n-1)} \sum_{i=3}^{n-2} r_{1i}$ ребер.

Общее число пересечений ребер
вершины x_2

$$p_2 = \sum_{j=4}^n r_{2j}.$$

Для вершины x_k

$$p_k = \sum_{j=k+2}^n r_{kj}.$$

Общее число пересечений ребер
графа

$$P(G) = \sum_{k=2}^{n-2} p_k.$$

Проще определять число пересечений по матрице соединений R . Будем использовать треугольную часть матрицы. Учитывая, что ребра графа не пересекают ребер гамильтонова цикла, в матрице "1", соответствующие гамильтонову циклу, заменим на "x".

$$R(G) = \begin{array}{c|cccccc} & 1 & 2 & 3 & \dots & n-1 & n \\ \hline 1 & 0 & \times & \boxed{R_{2n}} & & & \times \\ 2 & & 0 & \times & & & \boxed{r_{2n}} \\ 3 & & & 0 & \times & & \\ \dots & & & & & \times & \\ n-1 & & & & & 0 & \times \\ n & & & & & & 0 \end{array}$$

Определим p_{2n} , для чего в матрице R выделим подматрицу R_{2n} . Сумма единиц подматрицы R_{2n} соответствует p_{2n} . Для $p_{2(n-1)}$ выделим $R_{2(n-1)}$ и т.д. от R_{2n} до $R_{(n-2)n}$. Одновременно строим граф пересечений G' . Ребра графа G , не имеющие пересечений в G' не учитываются.

Для определения двудольности G' используем максимальные внутренне устойчивые множества.

Нахождение максимальных внутренне устойчивых

Для нахождения МВУМ при раскраске графа применялся метод Магу. Рассмотрим другой алгоритм, а именно - модифицированный алгоритм Г.А. Петухова.

1. В матрице R заменим все диагональные элементы на "1", $r_{jj}=1$. Положим $i=1$, $\alpha=1$;
2. В i -той строке находим элементы $r_{ij}=0$ ($j>i$). Номера нулевых элементов помещаем в список $J(j)$. Если все $r_{ij}=1$, то $\psi_\alpha=\{x_i\}$, $\alpha=\alpha+1$, перейти к п.7;
3. Записываем дизъюнкцию $M_{ij}=r_i \vee r_j$;
4. В строке M_{ij} находим $m_k=0$ ($k>j$), если все $m_k=1$, то перейти к п. 6;
5. Записываем дизъюнкцию $M_{ijk}=M_{ij} \vee r_k$; Переходим к п. 4.
6. Если в дизъюнкции нет ни одного нулевого элемента, $\psi_\alpha=\{x_i, x_j, x_k, \dots\}$, $\alpha=\alpha+1$.

Пока в списке $J(j)$ есть не рассмотренные элементы, выбрать следующий нулевой элемент и перейти к п. 3;

7. Положить $i = i+1$, пока $i < n$ перейти к п.2;

8. Семейство внутренне устойчивых множеств $\Psi_{G'}$ построено.

Выделение из G' максимального двудольного подграфа

Введем следующий критерий **H'**

$$\alpha_{\gamma\delta} = |\psi_\gamma| + |\psi_\delta| - |\psi_\gamma \cap \psi_\delta|.$$

Отсюда граф пересечений G' двудольный, а соответствующий ему граф G – планарен, если

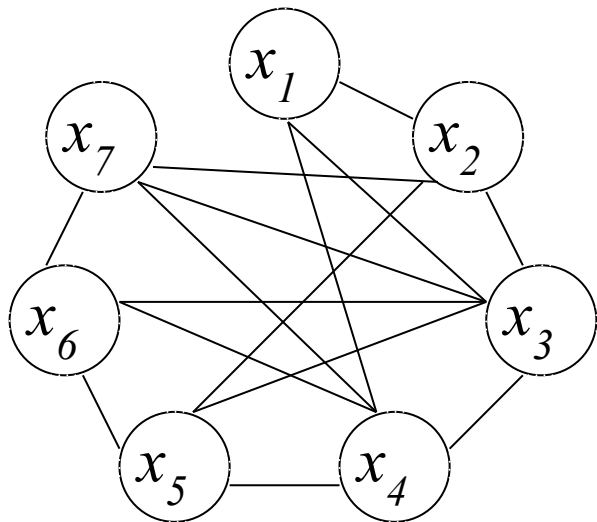
$$\alpha_{\gamma\delta} = |\psi_\gamma| + |\psi_\delta| = |X'|, \text{ а } |\psi_\gamma \cap \psi_\delta| = \emptyset.$$

Естественно, что чем ближе $\alpha_{\gamma\delta}$ к $|X'|$, тем большее число вершин содержит выделяемый двудольный подграф H' . Для его выделения необходимо определить $\alpha_{\gamma\delta}$ для всех пар $(\psi_\gamma, \psi_\delta)$ Ψ и выбрать ψ_γ и ψ_δ с $\max \alpha_{\gamma\delta}$.

Максимальному H' в исходном графе G соответствует суграф H , содержащий максимальное число непересекающихся ребер. В графе H ребра, вошедшие в одно внутренне устойчивое множество, проводятся внутри гамильтонова цикла, а во втором множестве ψ_δ исключаются ребра, вошедшие в ψ_γ и ψ_δ . Одинаковые множества объединяются в одно.

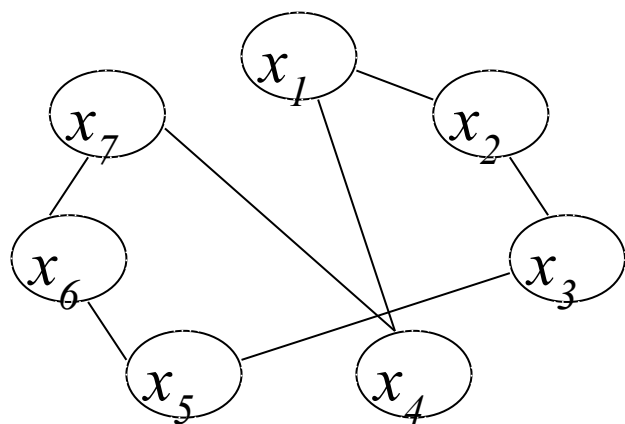
Описанная процедура повторяется до тех пор, пока Ψ_G не станет пусто.

Планаризовать граф $G(X, U)$



$R(G)=$

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

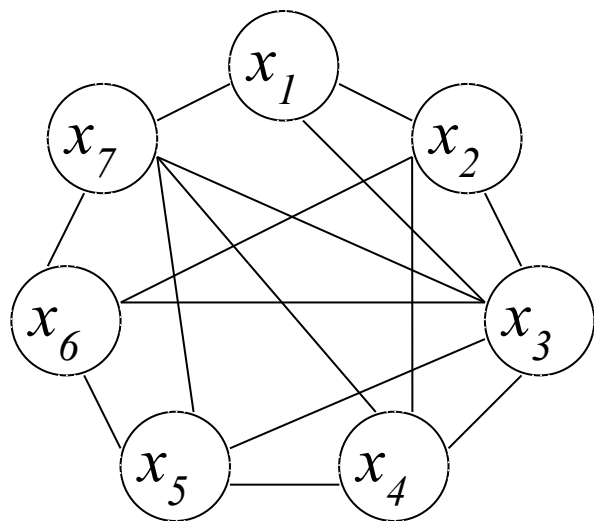


Построение графа пересечений G'

Перенумеруем вершины графа таким образом, чтобы ребра гамильтонова цикла были внешними.

до перенумерации	x_1	x_2	x_3	x_5	x_6	x_7	x_4
после перенумерации	x_1	x_2	x_3	x_4	x_5	x_6	x_7

Тогда граф $G(X, U)$ будет выглядеть так



$R(G) =$

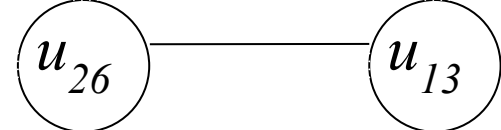
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	p_i
x_1	0	$\times$	1				$\times$	
x_2		0	$\times$	1		1		2
x_3			0	$\times$	1	1	1	4
x_4				0	$\times$		1	3
x_5					0	$\times$	1	2
x_6						0	$\times$	
x_7							0	

$$\sum p_i = 11$$

Определим p_{26} , для чего в матрице R

выделим подматрицу Ребро $(x_2 x_6)$ пересекается с ребром

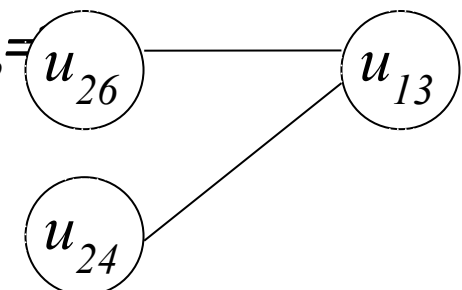
Строим граф пересечений G'



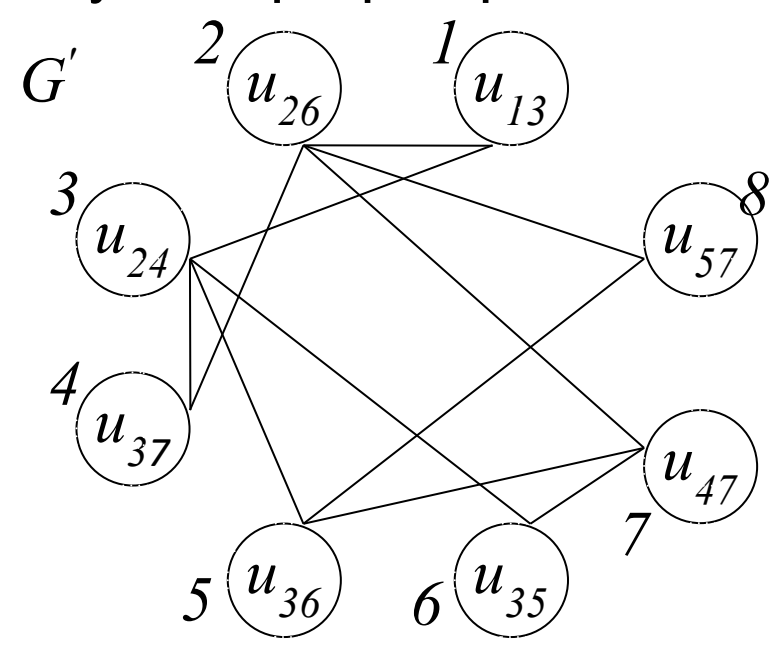
Определим p_{24} , для чего в матрице R выделим подматрицу R_{24} .

Ребро (x_2x_4) пересекается с ребром (x_1x_3) . $p_2 =$

Продолжаем строить граф пересечений G' .



После обработки остальных ребер
получим граф пересечений G'



$R(G') =$

	1	2	3	4	5	6	7	8
1	1	1	1					
2	1	1		1			1	1
3	1		1	1	1	1		
4		1	1	1				
5			1		1		1	1
6			1			1	1	
7		1			1	1	1	
8		1			1			1

Число пересечений ребер графа $P(G) = p_2 + p_3 + p_4 + p_5 = 11$.

Построение семейства $\Psi_{G'}$

В первой строке матрицы $R(G')$ находим номера нулевых элементов. Составляем список $J(j) = \{4, 5, 6, 7, 8\}$.

Для первого нулевого элемента составляем дизъюнкцию

$$M_{14} = r_1 \vee r_4 = \{11110000\}.$$

В строке M_{14} находим номера нулевых элементов. Составляем список $J'(j') = \{5, 6, 7, 8\}$.

Записываем дизъюнкцию $M_{145} = M_{14} \vee r_5 = \{11111011\}$.

В строке M_{145} находим $m_6 = 0$. Записываем дизъюнкцию

$$M_{1456} = M_{145} \vee r_6 = \{11111111\}.$$

В строке M_{1456} все «1».

Построено $\Psi_1 = \{u_{13}, u_{37}, u_{36}, u_{35}\}$.

Из списка $J'(j')$ выбираем следующий элемент. Записываем дизъюнкцию $M_{146} = M_{14} \vee r_6 = \{11110110\}$.

В строке M_{146} находим $m_8 = 0$. Записываем дизъюнкцию

$$M_{1468} = M_{146} \vee r_8 = \{11111111\}.$$

В строке M_{1468} все «1».

Построено $\Psi_2 = \{u_{13}, u_{37}, u_{35}, u_{57}\}$.

Из списка $J'(j')$ выбираем следующий элемент. Записываем дизъюнкцию $M_{147} = M_{14} \vee r_7 = \{11111110\}$.

И, наконец, $M_{1478} = M_{147} \vee r_8 = \{11111111\}$.

В строке M_{1478} все «1».

Построено $\Psi_3 = \{u_{13}, u_{37}, u_{47}, u_{57}\}$.

Из списка $J'(j')$ выбираем следующий элемент. Записываем дизъюнкцию $M_{148} = M_{14} \vee r_8 = \{11110001\}$.

В строке остались незакрытые «0».

Из списка $J(j)$ выбираем следующий нулевой элемент r_5 .

Составляем дизъюнкцию $M_{15} = r_1 \vee r_5 = \{11101011\}$.

В строке M_{15} находим нулевой элемент – $m_6 = 0$. Записываем дизъюнкцию $M_{156} = M_{15} \vee r_6 = \{11101111\}$.

В строке остался незакрытый «0». Понятно, что «0» в четвертой позиции элементами с номерами $j > 4$ закрыть не удастся.

Во второй строке ищем первый нулевой элемент – r_{23} .

Составляем дизъюнкцию $M_{23} = r_2 \vee r_3 = \{11111111\}$.

В строке M_{23} все «1».

Построено $\Psi_4 = \{u_{26}, u_{24}\}$.

Во второй строке ищем следующий нулевой элемент – r_{25} .

Составляем дизъюнкцию $M_{25} = r_2 \vee r_5 = \{11111011\}$.

И, наконец, $M_{256} = M_{25} \vee r_6 = \{11111111\}$.

Построено $\Psi_5 = \{u_{26}, u_{36}, u_{35}\}$.

Во второй строке ищем следующий нулевой элемент – r_{26} .

Составляем дизъюнкцию $M_{26} = r_2 \vee r_6 = \{11110111\}$.

В строке остался незакрытый «0».

В третьей строке ищем первый нулевой элемент – r_7 .

Составляем дизъюнкцию $M_{37} = r_3 \vee r_7 = \{11111110\}$.

И, наконец, $M_{378} = M_{37} \vee r_8 = \{11111111\}$.

Построено $\Psi_6 = \{u_{24}, u_{47}, u_{57}\}$.

В третьей строке ищем следующий нулевой элемент – r_{38} .

Составляем дизъюнкцию $M_{38} = r_3 \vee r_8 = \{1111101\}$.

В строке остался незакрытый «0».

Из матрицы $R(G')$ видно, что строки с номерами $j > 3$ «0» в первой позиции закрыть не смогут.

Семейство максимальных внутренне устойчивых множеств $\Psi_{G'}$ построено. Это

$$\psi_1 = \{u_{13}, u_{37}, u_{36}, u_{35}\};$$

$$\psi_2 = \{u_{13}, u_{37}, u_{35}, u_{57}\};$$

$$\psi_3 = \{u_{13}, u_{37}, u_{47}, u_{57}\};$$

$$\psi_4 = \{u_{26}, u_{24}\};$$

$$\psi_5 = \{u_{26}, u_{36}, u_{35}\};$$

$$\psi_6 = \{u_{24}, u_{47}, u_{57}\}.$$

Для каждой пары множеств вычислим значение критерия

$$\alpha_{\gamma\delta} = |\psi_\gamma| + |\psi_\delta| - |\psi_\gamma \cap \psi_\delta|.$$

Результаты вычислений запишем в матрицу $A = \|\alpha_{\gamma\delta}\|$.

$$\alpha_{12} = |\psi_1| + |\psi_2| - |\psi_1 \cap \psi_2| = 4 + 4 - 3 = 1.$$

$$\alpha_{13} = |\psi_1| + |\psi_3| - |\psi_1 \cap \psi_3| = 4 + 4 - 2 = 1 \text{ и т.д.}$$

	1	2	3	4	5	6
1	0	5	6	6	5	7
2		0	5	6	6	6
3			0	6	7	5
4				0	4	4
5					0	6
6						0

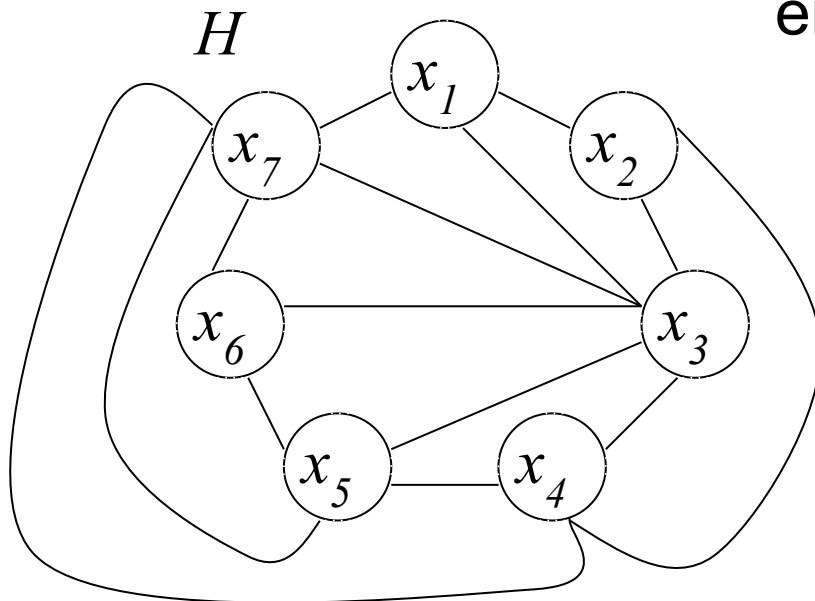
$\max_{y \in \delta} \alpha_{y\delta} = \alpha_{16} = \alpha_{35} = 7$, дают две пары множеств ψ_1, ψ_6 и ψ_3, ψ_5 .

Возьмем множества

$\psi_1 = \{u_{13}, u_{37}, u_{36}, u_{35}\}$ и

$\psi_6 = \{u_{24}, u_{47}, u_{57}\}$.

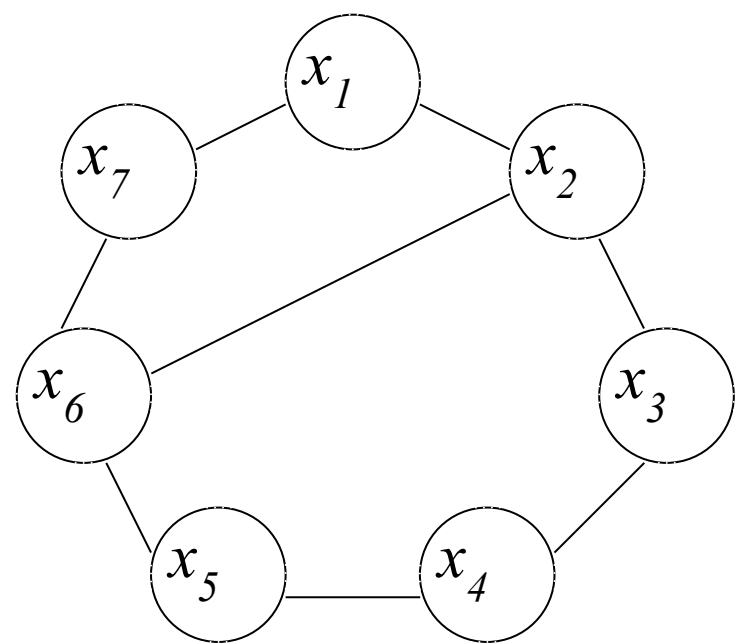
В суграфе H , содержащем максимальное число непересекающихся ребер, ребра, вошедшие в ψ_1 , проводим внутри гамильтонова цикла, а в ψ_6 – вне его



Удалим из ψ_G ребра, вошедшие в ψ_1 и ψ_6 $\psi_4 = \{u_{26}\}$; $\psi_5 = \{u_{26}\}$.

Объединим одинаковые множества, не реализованным осталось единственное ребро $\psi_4 = \{u_{26}\}$.

Проведем его.
Все ребра графа G
реализованы. Толщина графа m
 $= 2$.



Если взять множества $\psi_3 = \{u_{13}, u_{37}, u_{47}, u_{57}\}$
и $\psi_5 = \{u_{26}, u_{36}, u_{35}\}$, то не реализованным
окажется ребро $\{u_{24}\}$.