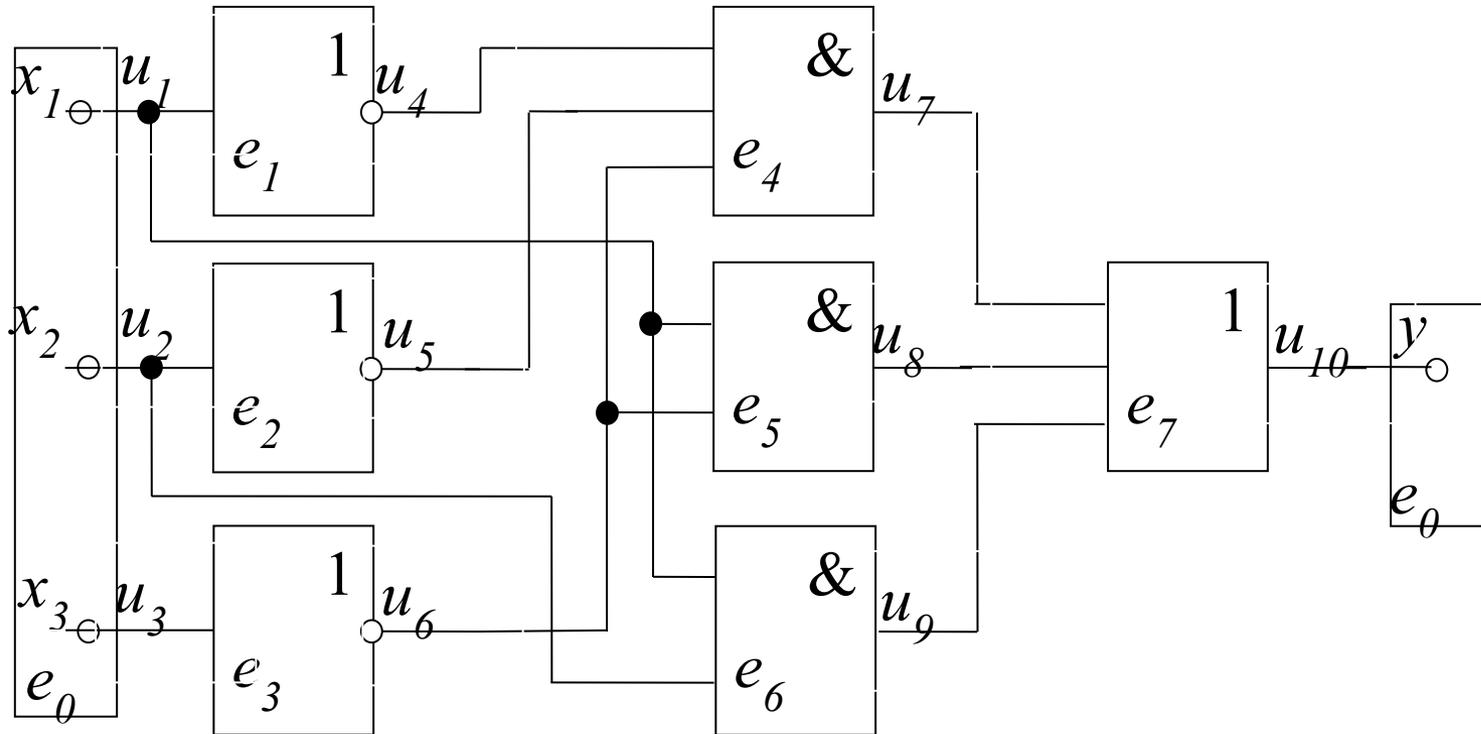
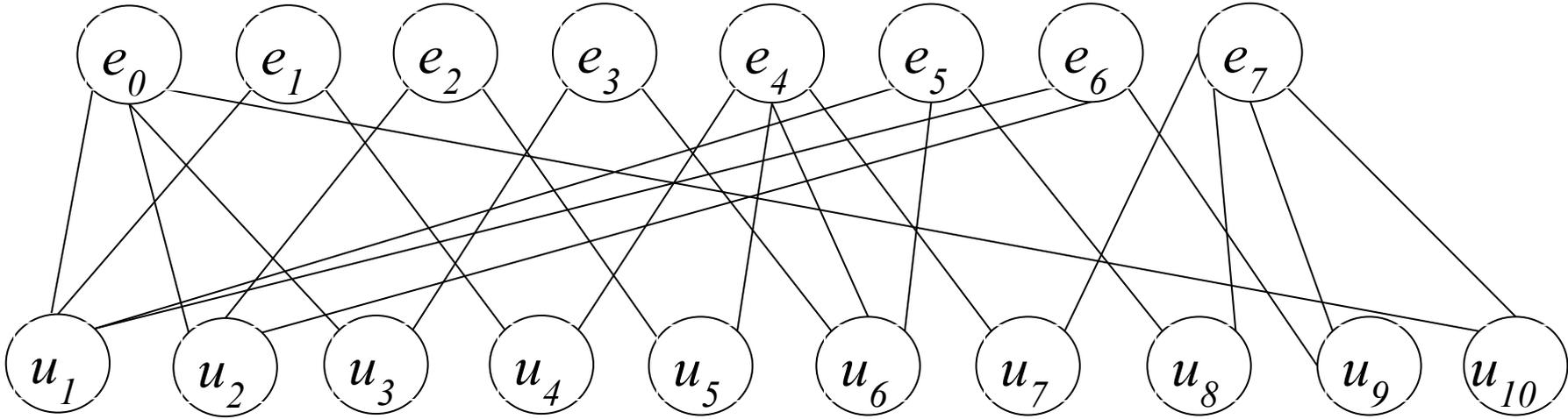


Математические модели электрических

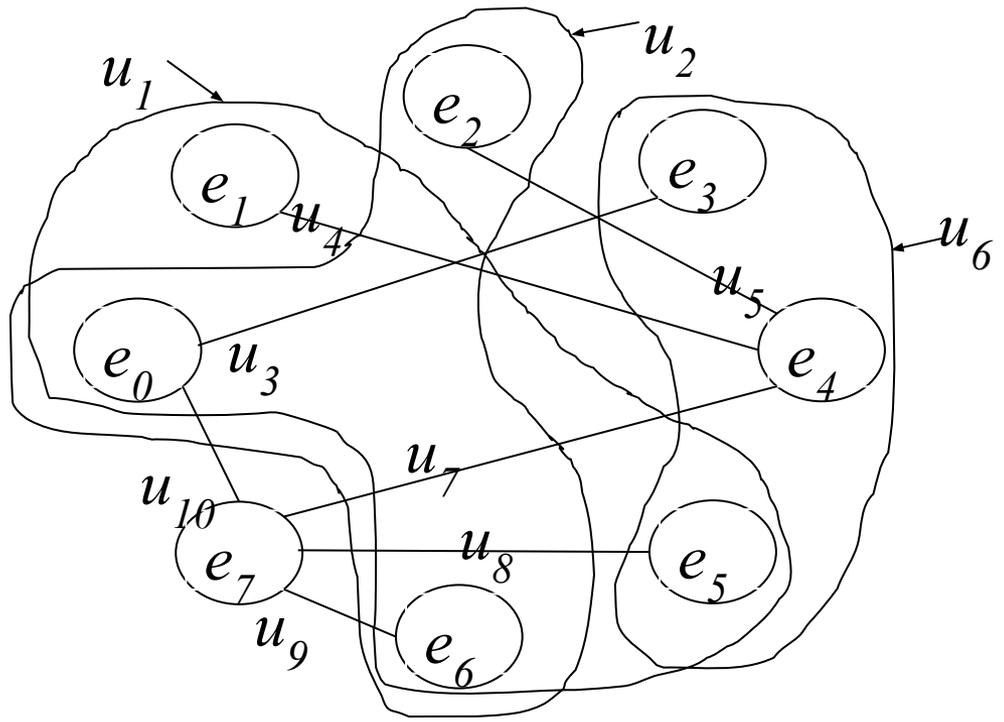
схем



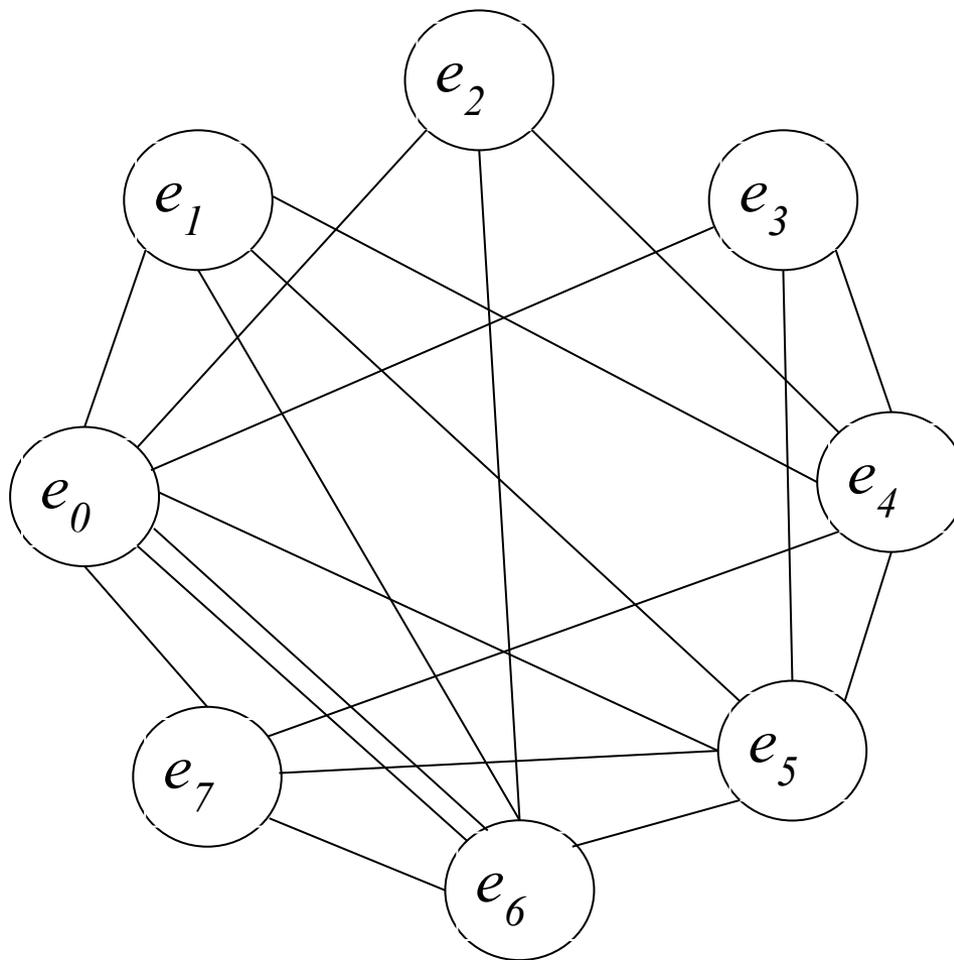
1. Модель схемы в виде двудольного графа $G(E, U, P)$



2. Модель схемы в виде гиперграфа $H(E, U)$



3. Модель схемы в виде мультиграфа $G(E, U)$



4. Модель схемы в виде взвешенного графа $G(E, U)$

Матрица соединений $R = ||r_{ij}||_{n \times n}$.

Матрицу соединений легко получить из матрицы
КОМПЛЕКСОВ

$$r_{ij} = \sum_{s=1}^k q_{is} q_{js}$$

	e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7
e_0	0	1	1	1	0	1	2	1
e_1	1	0	0	0	1	1	1	0
e_2	1	0	0	0	1	0	1	0
e_3	1	0	0	0	1	1	0	0
e_4	0	1	1	1	0	1	0	1
e_5	1	1	0	1	1	0	1	1
e_6	2	1	1	0	0	1	0	1
e_7	1	0	0	0	1	1	1	0

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

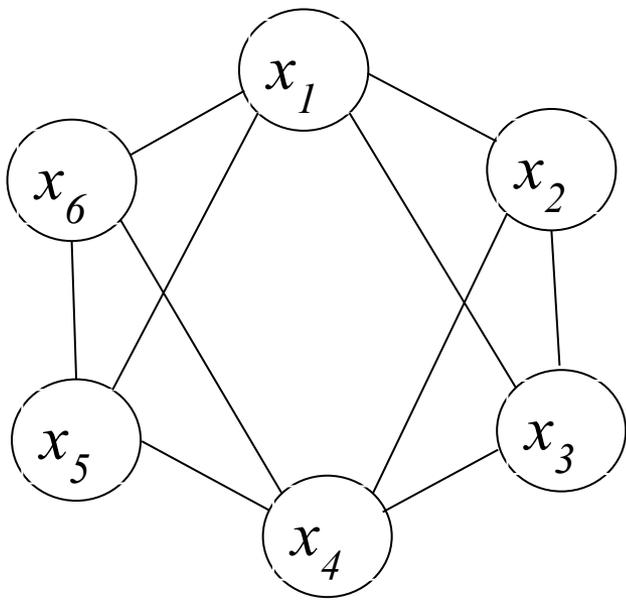
1. В матрице соединений R для каждой вершины подсчитывается число ненулевых элементов r_i ;
2. Находится вершина x_i с $\max r_i$, если таких вершин несколько, то выбирается любая;
3. Для выбранной вершины x_i записывается выражение $C_i = (x_i \vee x_a x_b \dots x_q)$, где $\Gamma x_i = \{x_a, x_b, \dots, x_q\}$;
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

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 ;

	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

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 ;

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

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

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

	x_3	x_6	r_i
x_3	0	0	0
x_6		0	0

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

11. $R = \emptyset$;

12. Составляем конъюнкцию C_i и выполняем минимизацию

$$P = \bigwedge C_i = C_1 C_2 C_4 C_5 = (x_1 \vee x_2 x_3 x_5 x_6)(x_2 \vee x_3)(x_4 \vee x_2 x_3 x_5 x_6)(x_5 \vee x_6) =$$

$$= x_1 x_2 x_4 x_5 \vee x_1 x_2 x_4 x_6 \vee x_1 x_3 x_4 x_5 \vee x_1 x_3 x_4 x_6 \vee x_2 x_3 x_5 x_6 = \bigvee K_j =$$

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 $t_i = \bigvee \phi_j$;

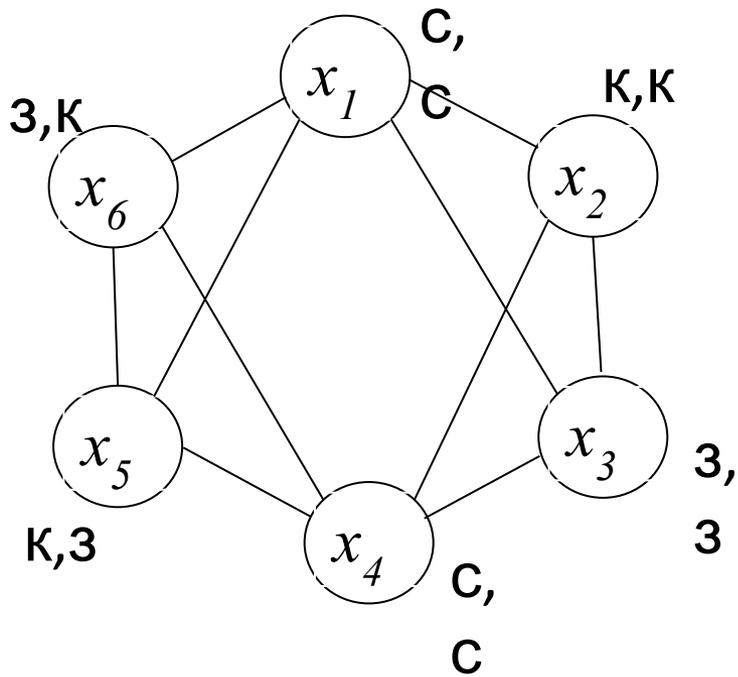
$$t_1 = \phi_5; t_2 = \phi_3 \vee \phi_4; t_3 = \phi_1 \vee \phi_2; t_4 = \phi_5; t_5 = \phi_2 \vee \phi_4; t_6 = \phi_1 \vee \phi_3;$$

15. Составляем конъюнкцию и выполняем минимизацию булевой функции

$$P' = \bigwedge t_i = t_1 t_2 t_3 t_4 t_5 t_6 = \phi_5(\phi_3 \vee \phi_4)(\phi_1 \vee \phi_2)\phi_5(\phi_2 \vee \phi_4)(\phi_1 \vee \phi_3) =$$

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

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



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

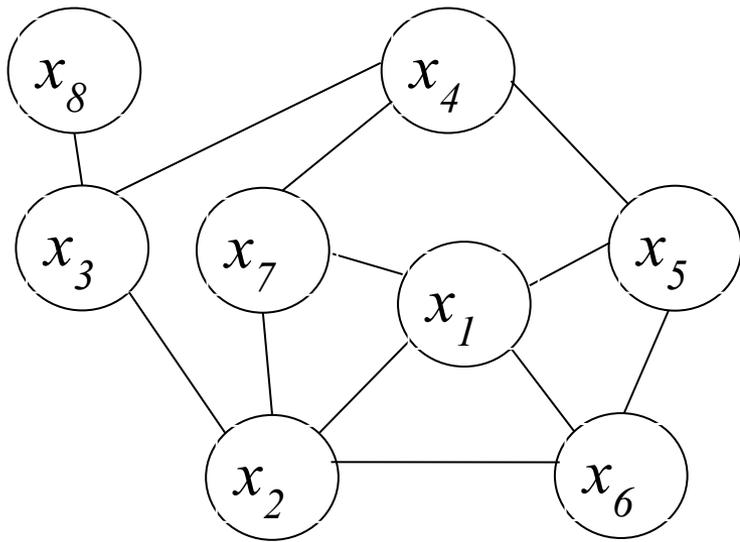
$$\phi_4 = \{x_2, x_5\}, \phi_5 = \{x_1, x_4\}.$$

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

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.



$$R = \begin{array}{c|cccccccc|c} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & r_i \\ \hline x_1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 4 \\ x_2 & & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 4 \\ x_3 & & & 0 & 1 & 0 & 0 & 0 & 1 & 3 \\ x_4 & & & & 0 & 1 & 0 & 1 & 0 & 3 \\ x_5 & & & & & 0 & 1 & 0 & 0 & 3 \\ x_6 & & & & & & 0 & 0 & 0 & 3 \\ x_7 & & & & & & & 0 & 0 & 3 \\ x_8 & & & & & & & & 0 & 1 \end{array}$$

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 .

11. 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 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 .

Алгоритмы размещения элементов

Постановка задачи

Задано множество конструктивных элементов, связанных между собой в соответствии с принципиальной электрической схемой узла. Требуется разместить элементы на некотором плоском коммутационном поле таким образом, чтобы некоторый функционал достигал экстремального значения.

Классическим критерием задачи размещения является критерий суммарной длины соединений, который интегральным образом учитывает многочисленные требования, предъявляемые к расположению элементов и трасс их соединений.

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 $(x_i, y_i), (x_j, y_j)$

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 СДС состоит в минимизации функционала $F(P)$ на множестве перестановок P . Данная задача называется *задачей квадратичного назначения*.

Алгоритм обратного размещения

Алгоритм обратного размещения принадлежит группе алгоритмов параллельно-последовательного размещения.

В методе обратного размещения осуществляются предварительные оценки каждого из размещаемых элементов и каждой свободной позиции, после чего все элементы размещаются одновременно заданы матрица соединений R и матрица расстояний D .

Для каждого элемента e_i найдем суммарное число соединений с другими элементами

$$r_i = \sum_{j=1}^m r_{ij}, \text{ где } i = 1, 2, \dots, m.$$

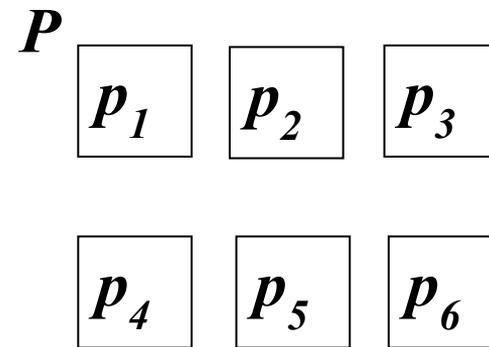
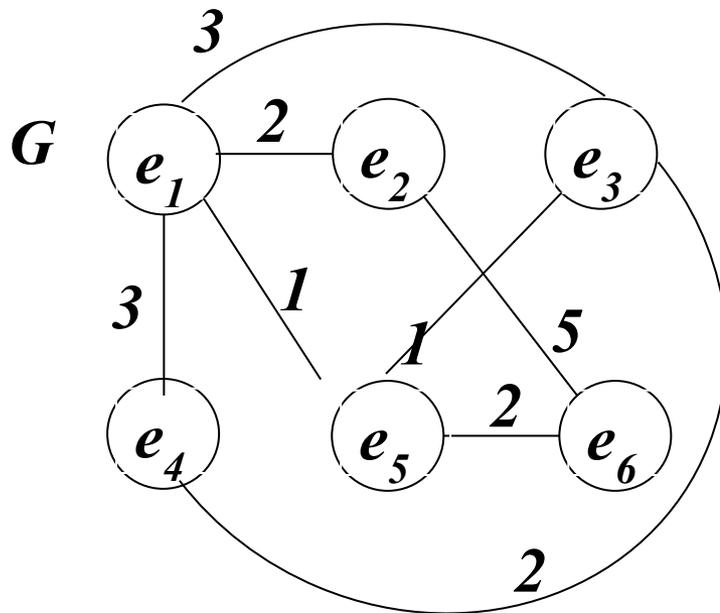
Для каждой позиции p_i найдем суммарное расстояние до остальных позиций

$$d_i = \sum_{j=1}^m d_{ij}, \text{ где } i = 1, 2, \dots, m.$$

Очевидно, что позиции в центральной части коммутационного поля имеют меньшую характеристику d_i , чем позиции на периферии. Естественно, что центральные позиции наиболее благоприятны для размещения сильно связанных элементов.

Учитывая условия минимальности скалярного произведения $r \times d$, получим следующий алгоритм:

1. Упорядочить элементы e_i в порядке не убывания r_i ;
 2. Упорядочить позиции p_i в порядке не возрастания d_i ;
 3. i -ый элемент из упорядоченного списка элементов помещается в i -ую позицию из упорядоченного списка позиций.
- Разместить элементы, заданные взвешенным графом G на множестве позиций P .



Составим матрицы соединений R и расстояний D .

	e_1	e_2	e_3	e_4	e_5	e_6	r_i		p_1	p_2	p_3	p_4	p_5	p_6	d_i
e_1	0	2	3	3	1	0	9	p_1	0	1	2	1	2	3	9
e_2		0	0	0	0	5	7	p_2		0	1	2	1	2	7
e_3			0	2	1	0	6	p_3			0	3	2	1	9
e_4				0	0	0	5	p_4				0	1	2	9
e_5					0	2	4	p_5					0	1	7
e_6						0	7	p_6						0	9

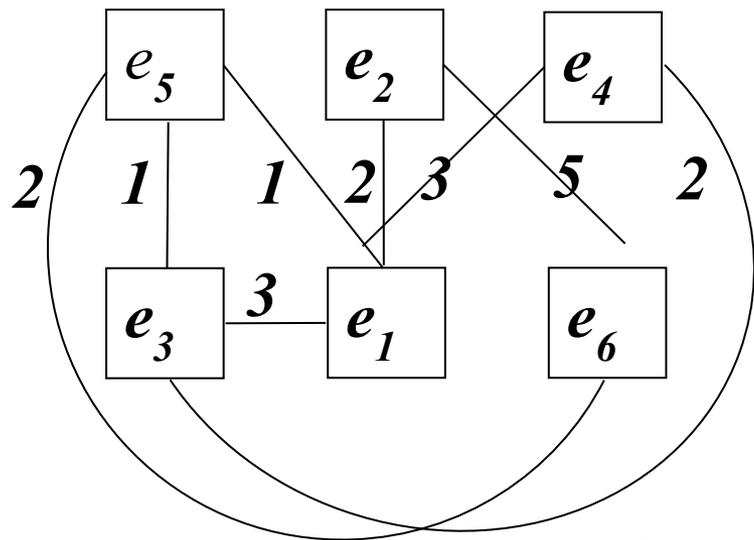
1. Упорядочим элементы e_i в порядке не убывания r_i

$$\{e_5, e_4, e_3, e_6, e_2, e_1\}.$$

2. Упорядочим позиции p_i в порядке не возрастания d_i

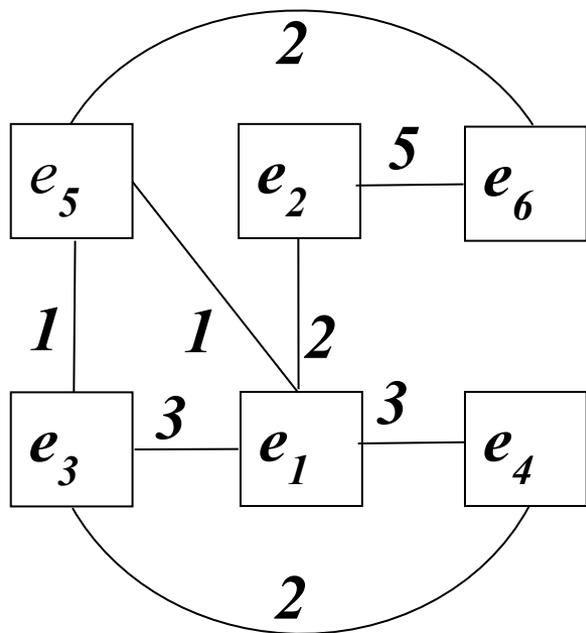
$$\{p_1, p_3, p_4, p_6, p_2, p_5\}.$$

3. Размещаем элементы в соответствии с упорядоченными списками

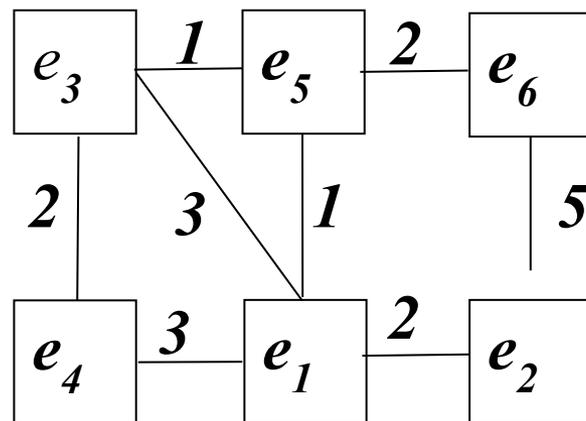


Значение целевой функции для полученного размещения $F(p)=36$.
 Можно поменять позиции вершин, размещенных в равноценные позиции. Так, можно переставить вершины e_4 и

e_6 .
 Целевая функция размещения $F(p)=24$.



У оптимального размещения значение целевой функции $F(p)=22$.



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.

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).

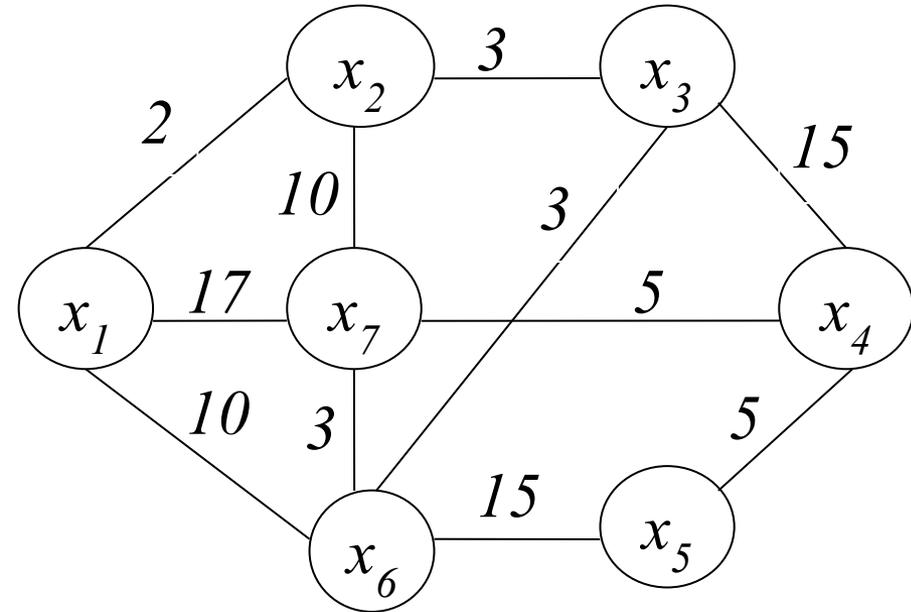
Если $p=t$, то $l(p)$ – длина кратчайшего пути, конец. Если $p \neq t$, перейти к п.2.

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.

$$L =$$

x_1	0^+
x_2	∞
x_3	∞
x_4	∞
x_5	∞
x_6	∞
x_7	∞

	1	2
x_1	0^+	
x_2	∞	2^+
x_3	∞	∞
x_4	∞	∞
x_5	∞	∞
x_6	∞	10
x_7	∞	17

	1	2	3
x_1	0^+		
x_2	∞	2^+	
x_3	∞	∞	5^+
x_4	∞	∞	∞
x_5	∞	∞	∞
x_6	∞	10	10
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 $l(x_2) = 2^+,$

$\Gamma = \{x_2\}$ – 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. $l(x_i^*) = \min[l(x_j)] = l(x_7) = 11$. 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. $l(x_i^*) = \min[l(x_j)] = l(x_4) = 16$. 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: $l(x_5) = \min[23,$
- ~~18. $l(x_i^*) = 21$.~~ 19. $l(x_5) = 21^+$, $p=x_5$.

20. Все пометки постоянные.

	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^+	

	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 .

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

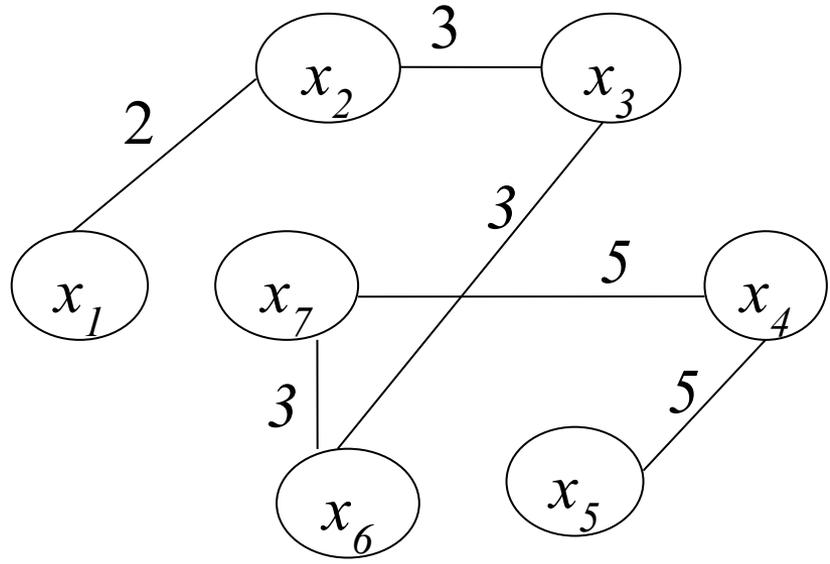
$$5 \neq l(x_4) + c(x_4, x_3) = 16 + 15, \quad 5 \neq l(x_6) + c(x_6, x_3) = 8 + 3.$$

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 .



Задачи, близкие к задаче о кратчайшем пути

1. Наиболее надежный путь.

В этом случае вес ребра представляет его надежность.

Надежность пути от s к t , составленного из ребер, взятых из

множества P , задается формулой $\prod_{(x_i, x_j) \in P} \rho_{ij}$ где ρ_{ij} – надежность ребра (x_i, x_j) .

2. *Самый длинный (критический) путь.*

Задача сетевого планирования, заключающаяся в нахождении самого длинного по временной протяженности пути в сетевом графике, определяющего продолжительность работ по выполнению проекта.

3. *Путь с наибольшей пропускной способностью.*

В этом случае 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* .

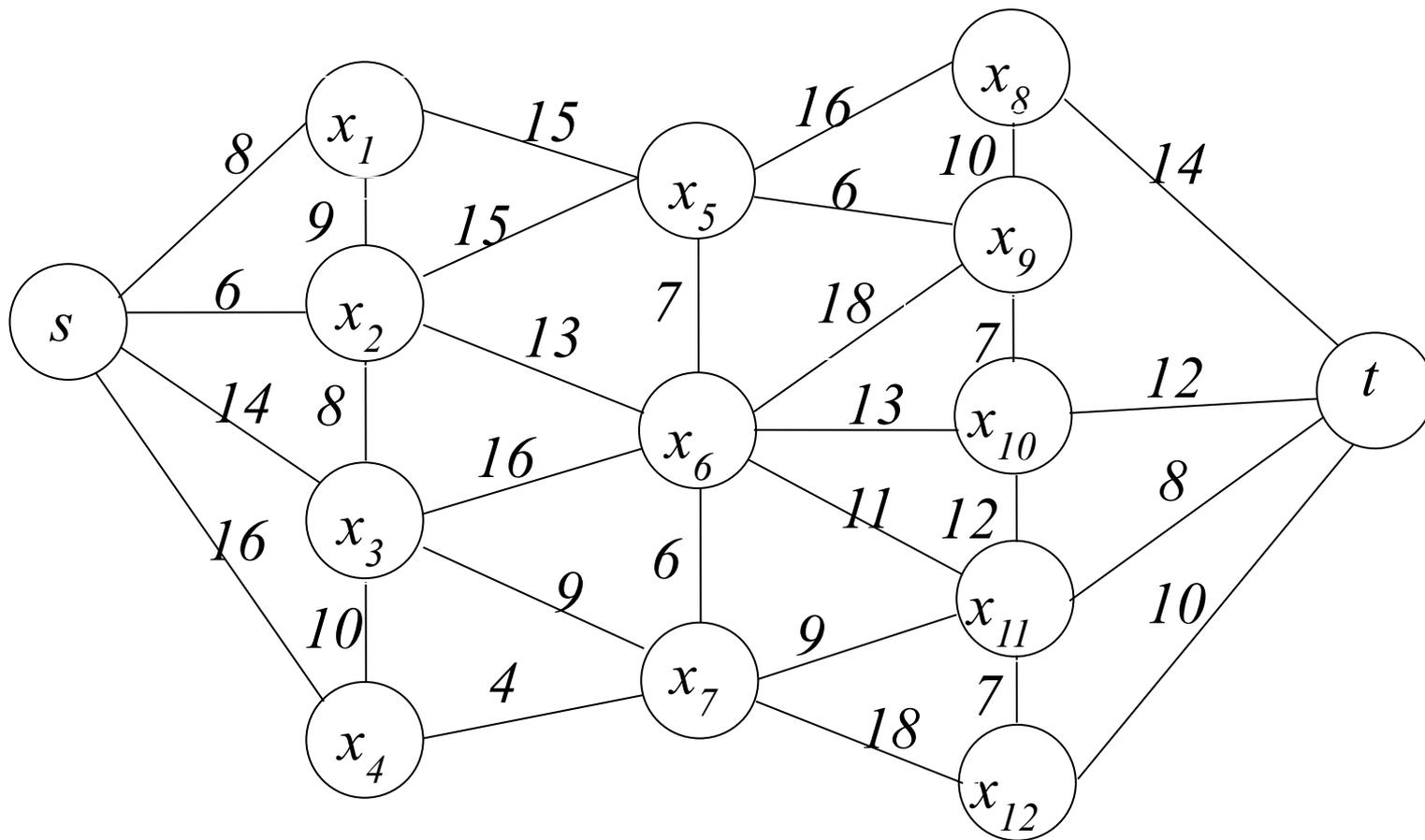
Теорема Форда – Фалкерсона. Пропускная способность пути с наибольшей пропускной способностью от s к t равна

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

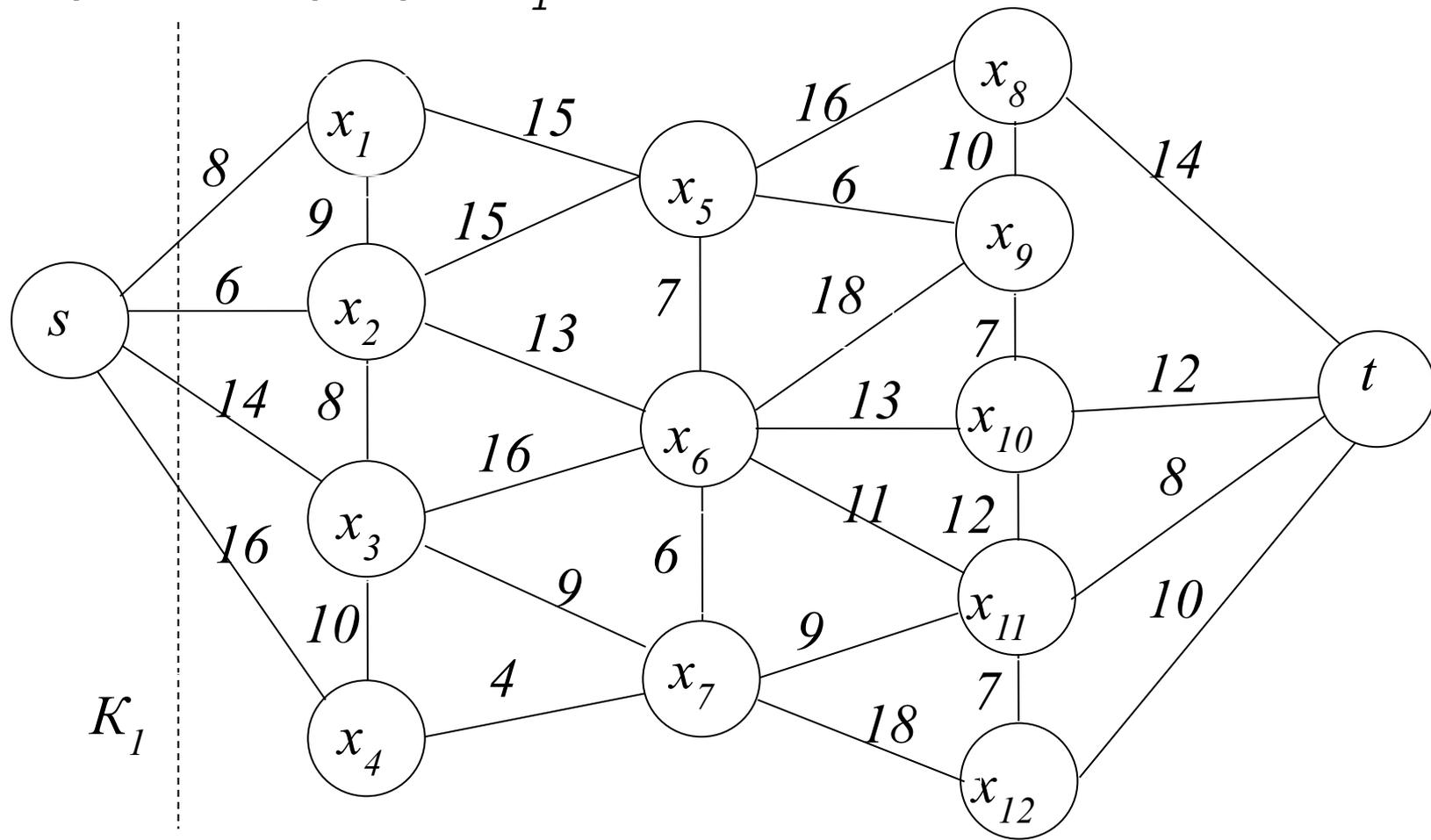
Алгоритм Франка – Фриша

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 $Q = Q_1$.
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.

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



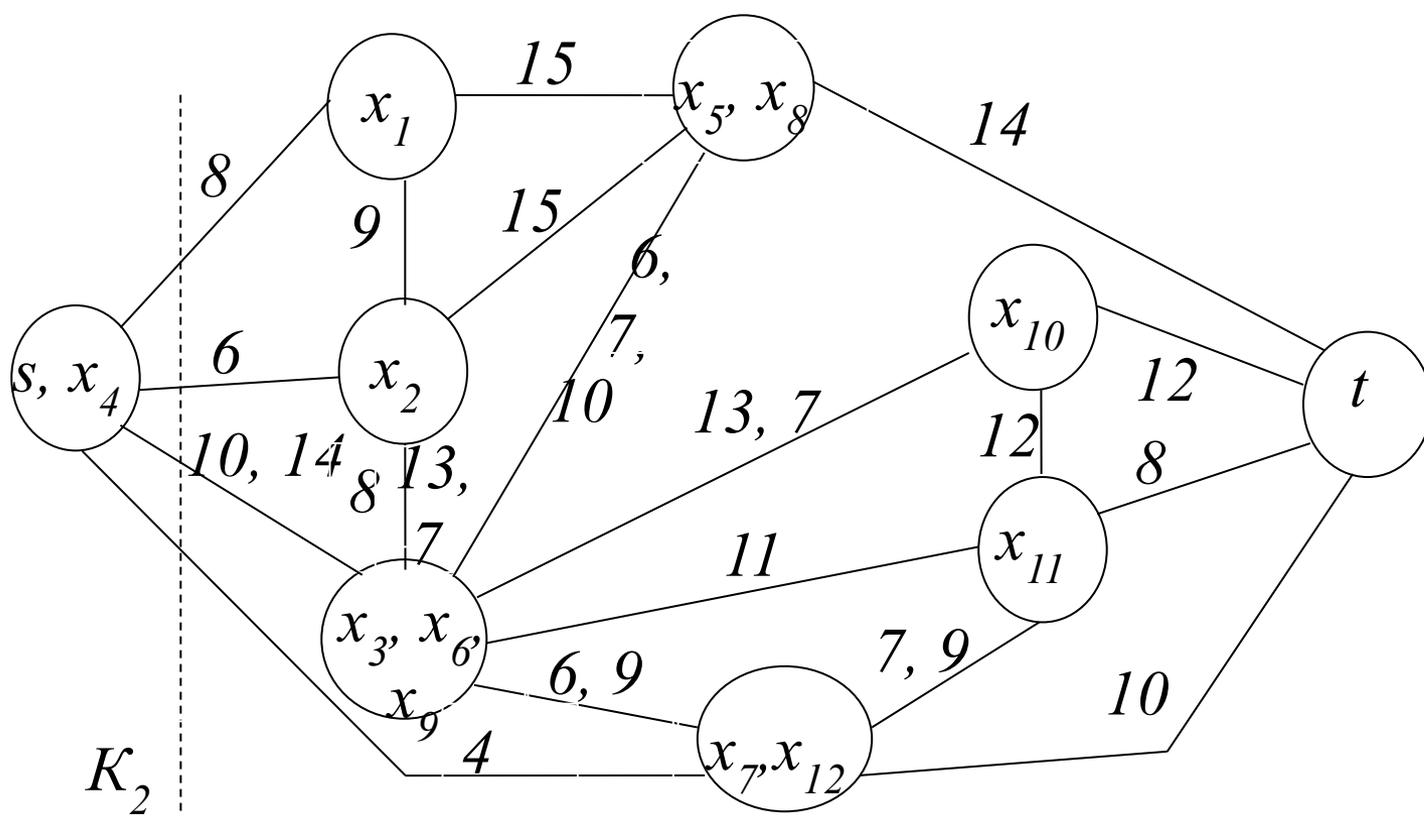
1. Проводим разрез $K_1 = (\{s\}, X \setminus \{s\})$



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 ,

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

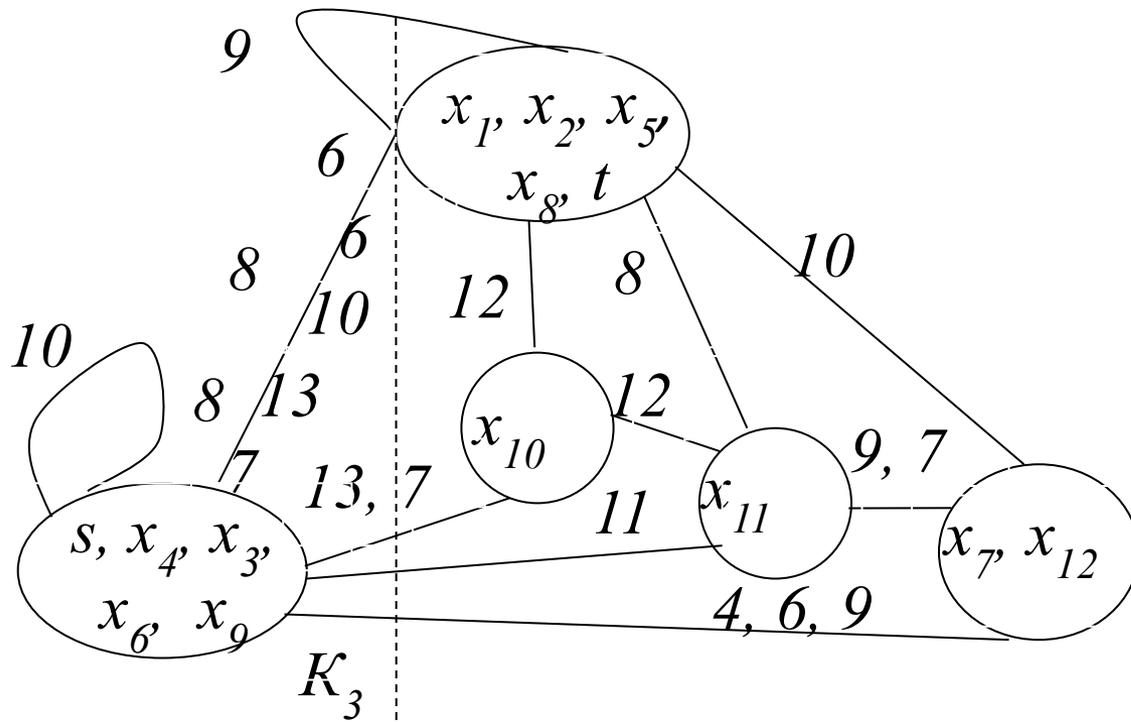
находим

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 .

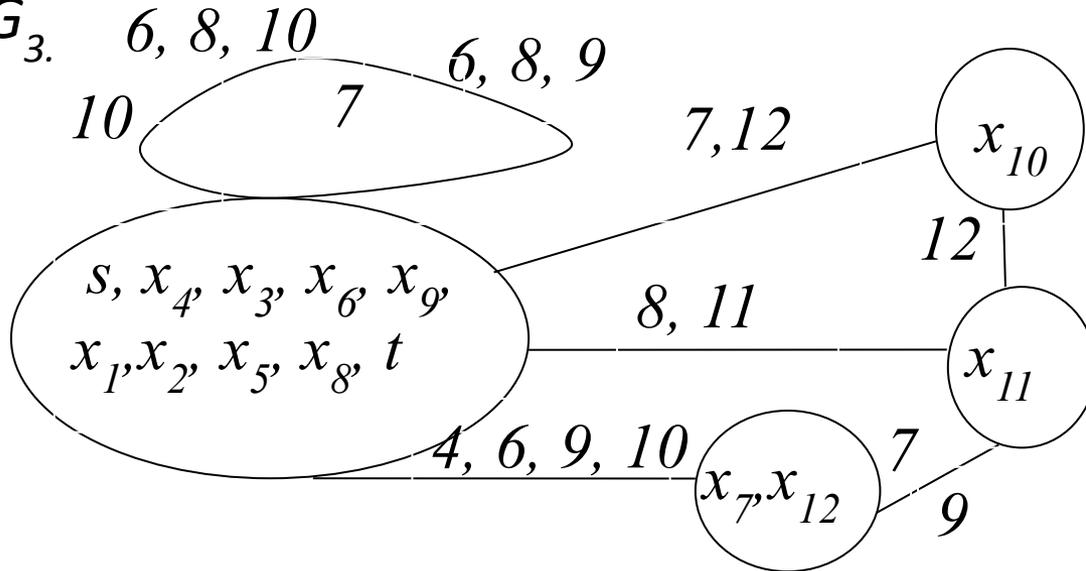
7. Проводим разрез K_3 ,

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

находим

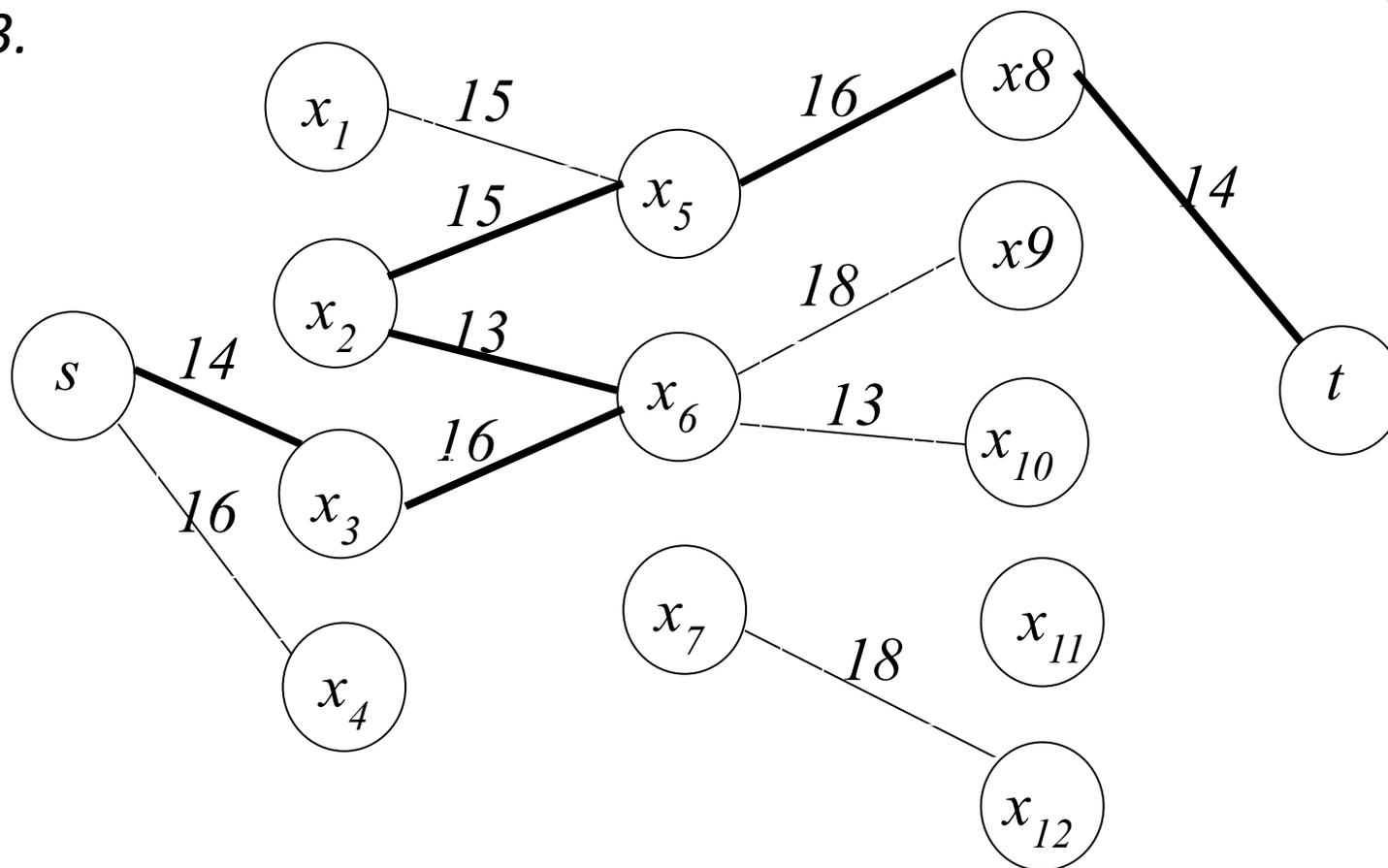


8. Закорачиваем все ребра графа (x_i, x_j) с $q_{ij} \geq Q_3$. Получаем граф G_3 .



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.

