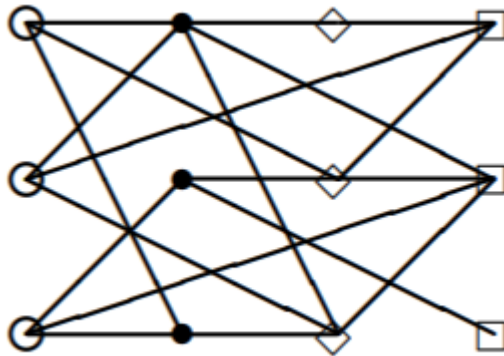


**Т.Р. Косовцева, В.В. Беляев**

**МОДЕЛИРОВАНИЕ СЕТЕВЫХ ЗАДАЧ  
СРЕДСТВАМИ СИСТЕМЫ R-PROJECT**



**Санкт-Петербург  
2021**

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

**Т.Р. Косовцева, В.В. Беляев**  
**МОДЕЛИРОВАНИЕ СЕТЕВЫХ ЗАДАЧ**  
**СРЕДСТВАМИ СИСТЕМЫ R-PROJECT**

УЧЕБНОЕ ПОСОБИЕ

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО  
по направлению подготовки 38.04.01 Экономика, 38.04.05 Бизнес-  
информатика, 27.04.05 Инноватика в качестве учебного пособия для  
реализации основных профессиональных образовательных программ  
высшего образования магистратуры

 УНИВЕРСИТЕТ ИТМО

**Санкт-Петербург**  
**2021**

Косовцева Т.Р., Беляев В.В., Моделирование сетевых задач средствами системы R-project – СПб: Университет ИТМО, 2021. – 62 с.

Рецензент(ы):

Цуканова Ольга Анатольевна, доктор экономических наук, профессор, профессор (квалификационная категория "ординарный профессор") факультета технологического менеджмента и инноваций, Университета ИТМО.

Приведены необходимые теоретические сведения и примеры выполнения заданий по построению минимального остовного дерева и нахождению кратчайшего пути в графе. Все решения выполнены как аналитически, путем использования методов «ближайшего соседа» и Дейкстры, так и с использованием электронных таблиц MS Excel, с помощью надстройки «Пакет анализа». Приведен пример решения задачи с использованием системы R-project.



**Университет ИТМО** – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2021  
© Косовцева Т.Р., Беляев В.В., 2021

## Содержание

Введение.....	4
Лабораторная работа 1. Задача о минимальном остовном дереве .....	4
Решение задачи в системе R-project .....	14
Задание .....	21
Лабораторная работа 2. Поиск кратчайшего пути на графе .....	24
Задание .....	47
Список литературы .....	48
Приложение 1 .....	49
Приложение 2 .....	51
Приложение 3 .....	53
Приложение 4 .....	55
Приложение 5 .....	57
Приложение 6 .....	59
Приложение 7 .....	60

## ВВЕДЕНИЕ

Теория графов - один из разделов дискретной математики, который широко применяется в решении множества задач. К их числу относятся экономические и управленческие задачи, задачи в программировании, химии, конструировании и изучении электрических цепей, коммуникации, психологии, социологии, лингвистике, других областях знаний. Теория графов систематически и последовательно изучает свойства графов. Граф - это объект, который состоит из множества вершин и множества ребер (дуг), которые отражают отношения между вершинами. Удобно считать, что граф состоит из множества точек и множества линий, отображающих связи между этими точками. Такая модель позволяет моделировать транспортные системы, в том числе дорожную сеть, где вершины графа играют роль городов, а ребра – роль дорог их соединяющих. Такого рода модели являются классикой теории графов, к их числу, в частности, относятся задачи об остовном дереве и кратчайшем пути. В учебном пособии рассмотрены алгоритмы ближайшего соседа (Прима) и Дейкстры для их решения.

### ЛАБОРАТОРНАЯ РАБОТА 1. ЗАДАЧА О МИНИМАЛЬНОМ ОСТОВНОМ ДЕРЕВЕ

Цель: Освоить на практике построение минимального остовного дерева методом ближайшего соседа.

Пусть дан связный неориентированный взвешенный граф  $G(X, U)$  с  $n$  вершинами. Каждому ребру, соединяющему вершины  $i$  и  $j$ , поставлено в соответствие число  $c_{ij}$ , называемое *весом*. Требуется построить для графа  $G$  минимальное остовное дерево.

*Деревом* называется связный граф без циклов и петель.

Остовным графом  $D(X1, U1)$  графа  $G(X, U)$  является граф, у которого множество вершин  $X1$  совпадает с множеством  $X(X1 \equiv X)$ , а множество ребер  $U1$  является подмножеством  $U (U1 \subset U)$ .

*Остовным деревом* называется связный остовной граф без циклов и петель.

Остовное дерево содержит  $(n - 1)$  ребер, где  $n$  – количество вершин.

*Кратчайшим остовом графа* называется такой остов, у которого сумма весов всех его ребер минимальна.

Другими словами, задача о кратчайшем остова дерева заключается в выборе таких ребер (дуг) заданного графа, что их суммарный вес («стоимость») минимальна, и для любой пары узлов найдется маршрут (или путь), соединяющий их. Этого можно достигнуть, выбирая дуги таким образом, что образованное ими дерево соединит все узлы данной сети.

Задача о кратчайшем остова имеет широкое практическое применение. Например, требуется разработать распределительную систему

для поставки природного газа с места разработки к  $n$  заказчикам. Тогда кратчайший остов сети подачи газа определит такую распределительную систему, которая свяжет всех заказчиков, и при этом затраты (или расстояние) будут минимальными.

Аналогичную задачу можно решать для построения линий электропередач. Возможны и другие применения. При этом в качестве весов графа можно использовать не только финансовые затраты, но и расстояния, количество материала и т.д.

Одним из наиболее простых алгоритмов, реализующих решение данной задачи, является алгоритм, основанный на «поглощении» кратчайших дуг, который часто называют алгоритмом «ближайшего соседа».

### АЛГОРИТМ «БЛИЖАЙШЕГО СОСЕДА»

Обозначим номер шага переменной  $k$ . На каждом шаге множество вершин, включённых в остовное дерево, будем обозначать  $S_k$ . Начальный шаг будем считать нулевым.

0. Присваиваем  $k$  значение равное нулю ( $k=0$ ).

1. Формируем множество  $S_k$ , для этого включаем в него произвольную вершину графа, например,  $x_1$ , таким образом  $S_0 = \{x_1\}$ .

2. Выписать все ребра графа, инцидентные вершинам, принадлежащим множеству  $S_k$ , но не оканчивающиеся в нем, и их веса.

3. Выбрать из них ребро с минимальным весом и включить его в будущее дерево.

4. Занести конечную вершину этого ребра в множество  $S_k$ , получим множество  $S_{k+1}$ .

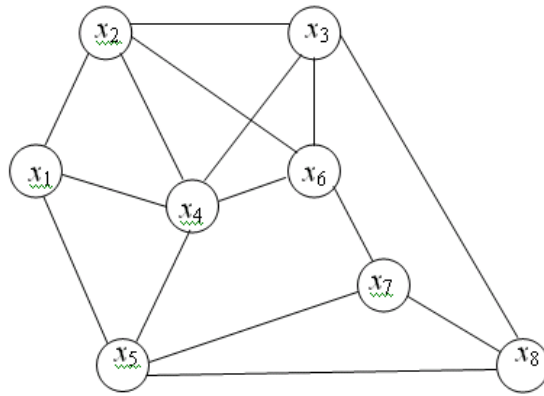
5. Проверить, все ли вершины исходного графа вошли в множество  $S_{k+1}$ , т.е. проверить, выполняется ли условие  $n_1 < n$ , где  $n_1$  – количество элементов множества  $S_{k+1}$ .

6. Если  $n_1 = n$ , процесс построения дерева завершен. Если  $n_1 < n$ , процесс построения дерева не окончен, следует  $k$  увеличить на единицу ( $k = k+1$ ) и продолжить решение, выполняя пункты, начиная с п. 2.

Заметим, что на каждом шаге количество вершин, включенных в остовное дерево, увеличивается на единицу, поэтому для завершения построения дерева потребуется  $n-1$  шаг.

### Пример

Пусть дан связный неориентированный взвешенный граф  $G(X, U)$  с количеством вершин  $n = 8$  (рис. 1.1).



**Рисунок 1.1 - Связный неориентированный взвешенный граф**

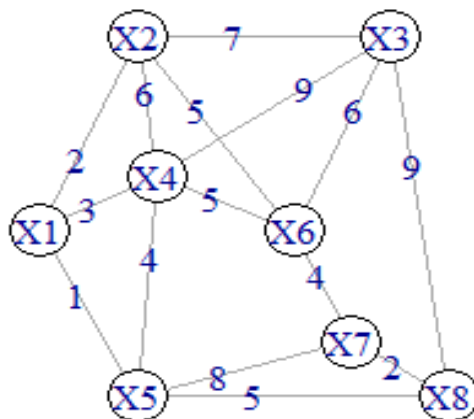
Каждому ребру, соединяющему вершины  $i$  и  $j$ , поставлено в соответствие число  $c_{ij}$ , называемое *весом*. Матрица весов графа  $G(X, U)$  представлена в табл.1.1. Для простоты знак  $\infty$  (вес несуществующего ребра) в матрицу не записан.

Построить для графа  $G$  минимальное остовное дерево.

Таблица 1.1

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$
$x_1$		2		3	1			
$x_2$	2		7	6		5		
$x_3$		7		9		6		9
$x_4$	3	6	9		4	5		
$x_5$	1			4			8	5
$x_6$		5	6	5			4	
$x_7$					8	4		2
$x_8$			9		5		2	

На рисунке 1.2 представлен граф с указанием веса каждого его ребра.



**Рисунок 1.2 - Граф с указанием веса каждого его ребра**

Воспользуемся для решения задачи описанным выше алгоритмом. Для этого выполним следующие действия:

1.1. Выберем произвольную вершину графа, например,  $x_1$ . и включим её в будущее дерево:  $S_0 = \{x_1\}$ .

1.2. Выпишем все ребра графа, инцидентные данной вершине, и их веса:

$$\{u_{12}, u_{14}, u_{15}\},$$

$$\{c_{12} = 2, c_{14} = 3, c_{15} = 1\}.$$

1.3. Выберем из них ребро с минимальным весом –  $u_{15}$  ( $c_{15} = 1$ ), и включим его в будущее дерево: внесем конечные вершины ребра  $u_{15}$  в множество  $S_0$ :

$$S_0 = \{x_1, x_5\}$$

и выделим его на графе (рис. 1.2).

1.4. Проверим, все ли вершины графа  $G(X, U)$  вошли в множество  $S_1$ , т.е. равно ли количество элементов  $n_1$  множества  $S_1$  величине  $n$ .

На данном шаге ( $n_1 = 2$ ) < ( $n = 8$ ), поэтому продолжим построение дерева (п. 2).

Геометрический смысл указанных действий показан на рис. 1.3. На этом и последующих рисунках ребра, включенные в остовное дерево обозначены жирными линиями, ребра, которые являются соседями обозначены точками, лучшее ребро (среди соседей) – жирным пунктиром.

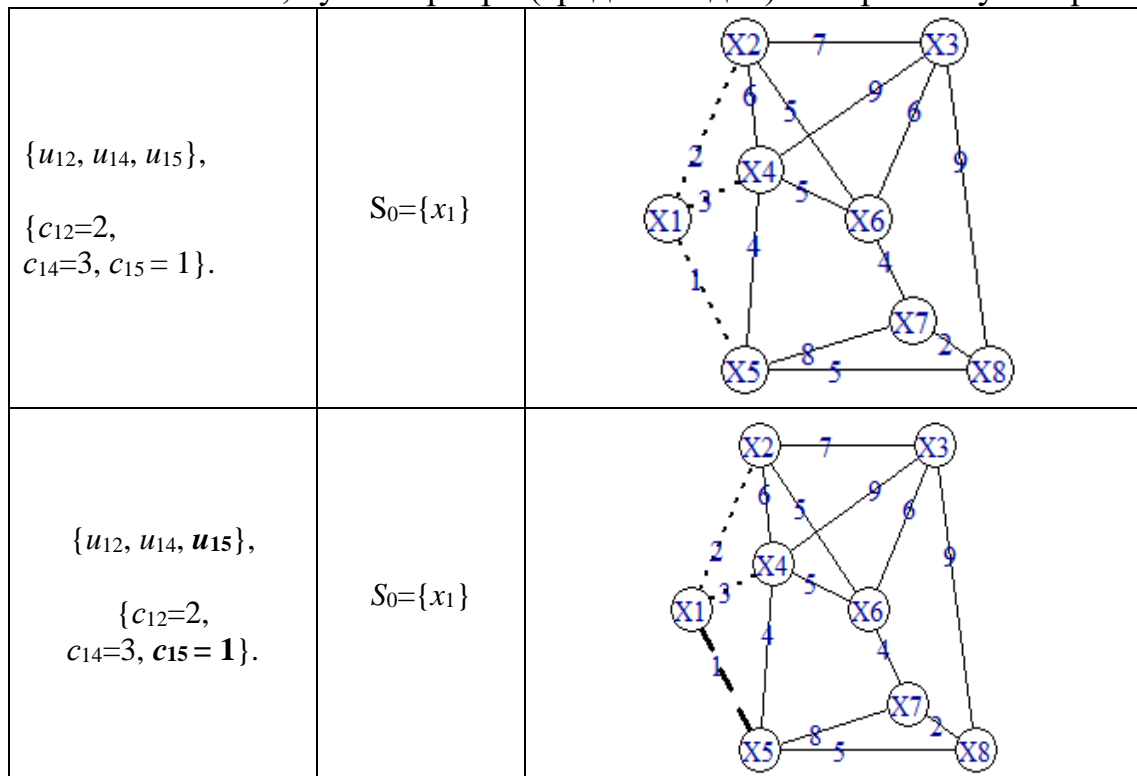


Рисунок 1.3 (начало)



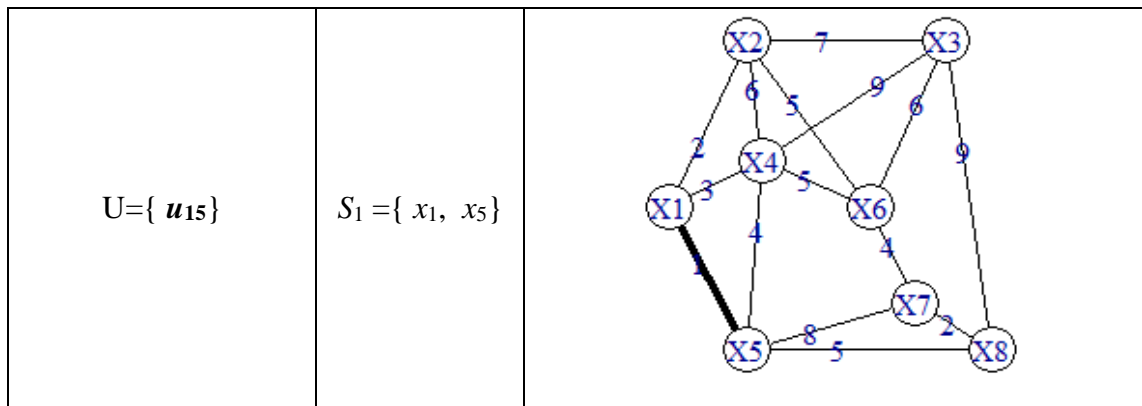


Рисунок 1.3 (окончание)

2.1. Выпишем все ребра, кроме ребра, выбранного ранее ( $u_{15}$ ), инцидентные вершинам, принадлежащим множеству  $S_1$ , но не оканчивающиеся в нем, и их веса.

$$\{u_{12}, u_{14}, u_{54}, u_{57}, u_{58}\},$$

$$\{c_{12} = 2, c_{14} = 3, c_{54} = 4, c_{57} = 8, c_{58} = 5\}.$$

2.2. Выберем из них ребро с минимальным весом –  $u_{12}$  ( $c_{12} = 2$ ), и включим его в будущее дерево. Выделим его на графе (рис. 1.4).

2.3. Внесем конечные вершины ребра  $u_{12}$  в множество  $S_1$ . Вершина  $x_1$  уже имеется в множестве  $S_1$ , остается внести вершину  $x_2$ . Получим множество  $S_2$ :

$$S_2 = \{x_1, x_2, x_5\}.$$

2.4. Проверим, все ли вершины графа  $G(X, U)$  вошли в множество  $S_2$ , т.е. равно ли количество элементов  $n_1$  множества  $S_2$  величине  $n$ .

На данном шаге ( $n_1 = 3$ ) < ( $n = 8$ ), поэтому продолжим построение дерева (п. 2).

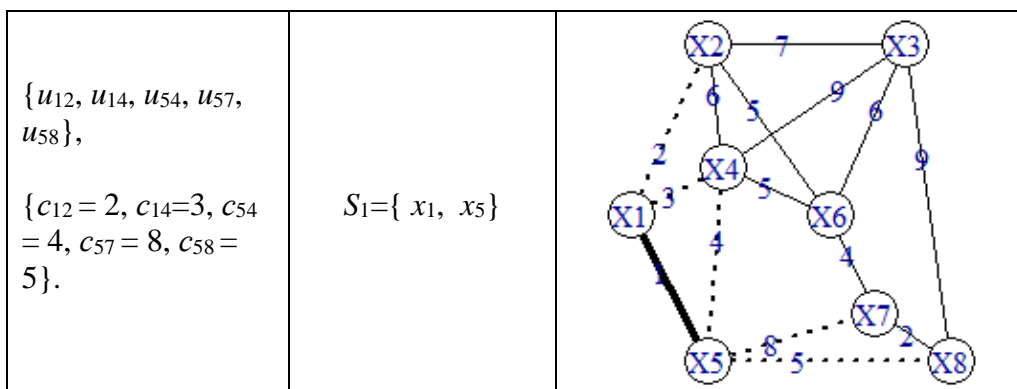


Рисунок 1.4 (начало)

$\{u_{12}, u_{14}, u_{54}, u_{57}, u_{58}\},$ $\{c_{12}=2, c_{14}=3, c_{54}=4, c_{57}=8, c_{58}=5\}.$	$S_1 = \{x_1, x_5\}$	
$U = \{u_{15}, u_{12}\}$	$S_2 = \{x_1, x_2, x_5\}.$	

Рисунок 1.4 (окончание)

3.1. Выпишем все ребра, кроме ребер, выбранных ранее ( $u_{12}, u_{15}$ ), инцидентные вершинам, принадлежащим множеству  $S_1$ , но не оканчивающиеся в нем, и их веса.

$$\{u_{14}, u_{23}, u_{24}, u_{26}, u_{54}, u_{57}, u_{58}\},$$

$$\{c_{14} = 3, c_{23} = 7, c_{24} = 6, c_{26} = 5, c_{54} = 4, c_{57} = 8, c_{58} = 5\}.$$

3.2. Выберем из них ребро с минимальным весом –  $u_{14}$  ( $c_{14} = 3$ ) и включим его в будущее дерево.

3.3. Внесем конечные вершины ребра  $u_{14}$  (которых нет в этом множестве) в множество  $S_1$ . Получим множество  $S_3$ :

$$S_3 = \{x_1, x_2, x_4, x_5\}.$$

Выделим его на графе (рис. 1.5).

3.4. Так как  $(n_2 = 4) < (n = 8)$ , необходимо продолжить построение дерева (рис.1.6).

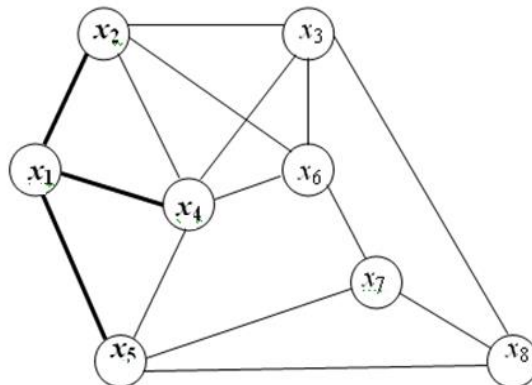


Рисунок 1.5

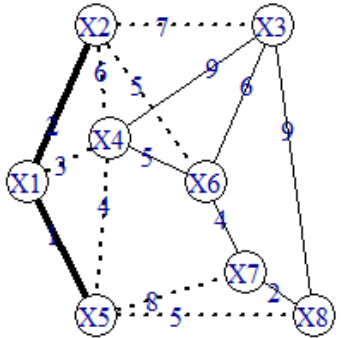
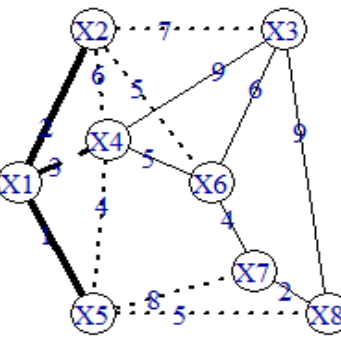
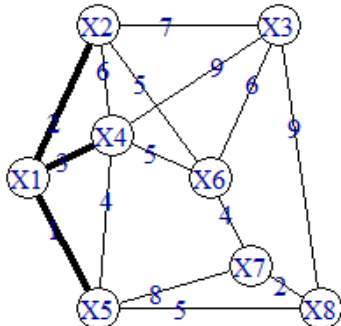
$\{u_{14}, u_{23}, u_{24}, u_{26}, u_{54}, u_{57}, u_{58}\},$  $\{c_{14} = 3, c_{23} = 7, c_{24} = 6, c_{26} = 5, c_{54} = 4, c_{57} = 8, c_{58} = 5\}.$	$S_2 = \{x_1, x_2, x_5\}.$	
$\{u_{14}, u_{23}, u_{24}, u_{26}, u_{54}, u_{57}, u_{58}\},$  $\{c_{14} = 3, c_{23} = 7, c_{24} = 6, c_{26} = 5, c_{54} = 4, c_{57} = 8, c_{58} = 5\}.$	$S_2 = \{x_1, x_2, x_5\}.$	
$U = \{u_{15}, u_{12}, u_{14}\}.$	$S_3 = \{x_1, x_2, x_4, x_5\}.$	

Рисунок 1.6

4.1. Выпишем все ребра, кроме ребер, выбранных ранее ( $u_{12}, u_{14}, u_{15}$ ), инцидентные вершинам, принадлежащим множеству  $S_3$ , но не оканчивающиеся в нем, и их веса.

$$\{u_{23}, u_{26}, u_{43}, u_{46}, u_{57}, u_{58}\},$$

$$\{c_{23} = 7, c_{26} = 5, c_{43} = 9, c_{46} = 5, c_{57} = 8, c_{58} = 5\}.$$

4.2. Выберем из них ребро с минимальным весом. Таких ребер три:  $u_{26}$  ( $c_{26} = 5$ ),  $u_{46}$  ( $c_{46} = 5$ ),  $u_{58}$  ( $c_{58} = 5$ ). В будущее дерево можно включить любое из них. Включим ребро  $u_{26}$ . Выделим его на графе (рис. 1.7).

4.3. Внесем конечные вершины ребра  $u_{26}$  в множество  $S_3$  (которых нет в этом множестве). Получим множество  $S_4$ :

$$S_4 = \{x_1, x_2, x_4, x_5, x_6\}.$$

4.4. Так как  $(n_4 = 5) < (n = 8)$ , необходимо продолжить построение дерева.

$\{u_{23}, u_{26}, u_{43}, u_{46}, u_{57}, u_{58}\},$ $\{c_{23} = 7, \quad c_{26} = 5, c_{43} = 9, c_{46} = 5, c_{57} = 8, c_{58} = 5\}.$	$S_3 = \{x_1, x_2, x_4, x_5\}$	
$\{u_{23}, \mathbf{u_{26}}, u_{43}, u_{46}, u_{57}, u_{58}\},$ $\{c_{23} = 7, \quad c_{26} = \mathbf{5}, c_{43} = 9, c_{46} = 5, c_{57} = 8, c_{58} = 5\}.$	$S_3 = \{x_1, x_2, x_4, x_5\}$	
$U = \{u_{15}, u_{12}, u_{14}, u_{26}\}.$	$S_4 = \{x_1, x_2, x_4, x_5, x_6\}$	

Рисунок 1.7

5.1. Выпишем все ребра, кроме ребер, выбранных ранее ( $u_{12}, u_{14}, u_{15}, u_{26}$ ), инцидентные вершинам, принадлежащим множеству  $S_4$ , но не оканчивающиеся в нем, и их веса.

$$\{u_{23}, u_{43}, u_{57}, u_{58}, u_{63}, u_{67}\},$$

$$\{c_{23} = 7, c_{43} = 9, c_{57} = 8, c_{58} = 5, c_{63} = 6, c_{67} = 4\}.$$

5.2. Выберем из них ребро с минимальным весом:  $u_{67}$  ( $c_{67} = 4$ ) и включим его в будущее дерево. Выделим его на графе (рис. 1.8).

5.3. Внесем конечные вершины ребра  $u_{67}$  в множество  $S_4$ . Получим множество  $S_5$ :

$$S_5 = \{x_1, x_2, x_4, x_5, x_6, x_7\}.$$

5.4. Так как ( $n_5 = 6$ ) < ( $n = 8$ ), необходимо продолжить построение дерева.

$\{u_{23}, u_{43}, u_{57}, u_{58}, u_{63}, u_{67}\},$ $\{c_{23} = 7, c_{43} = 9, c_{57} = 8, c_{58} = 5, c_{63} = 6, c_{67} = 4\}$	$S_4 = \{x_1, x_2, x_4, x_5, x_6\}..$	
$\{u_{23}, u_{43}, u_{57}, u_{58}, u_{63}, u_{67}\},$ $\{c_{23} = 7, c_{43} = 9, c_{57} = 8, c_{58} = 5, c_{63} = 6, c_{67} = 4\}$	$S_4 = \{x_1, x_2, x_4, x_5, x_6\}..$	
$U = \{u_{15}, u_{12}, u_{14}, u_{26}, u_{67}\}.$	$S_5 = \{x_1, x_2, x_4, x_5, x_6, x_7\}$	

Рисунок 1.8

6.1. Выпишем все ребра, кроме ребер, выбранных ранее ( $u_{12}, u_{14}, u_{15}, u_{26}, u_{67}$ ), инцидентные вершинам, принадлежащим множеству  $S_4$ , но не оканчивающиеся в нем, и их веса.

$$\{u_{23}, u_{58}, u_{63}, u_{78}\},$$

$$\{c_{23} = 7, c_{58} = 5, c_{63} = 6, c_{78} = 2\}.$$

6.2. Выберем из них ребро с минимальным весом:  $u_{78}$  ( $c_{78} = 2$ ) и включим его в будущее дерево. Выделим его на графе (рис. 1.9).

6.3. Внесем конечные вершины ребра  $u_{78}$  в множество  $S_5$  (которых нет в этом множестве). Получим множество  $S_6$ :

$$S_6 = \{x_1, x_2, x_4, x_5, x_6, x_7, x_8\}.$$

6.4. Так как  $(n_6 = 7) < (n = 8)$ , необходимо продолжить построение дерева.

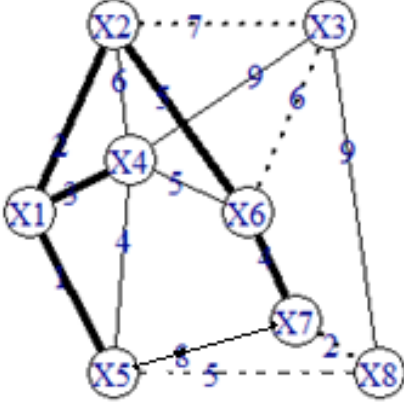
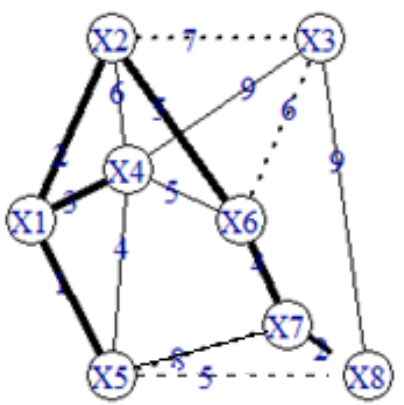
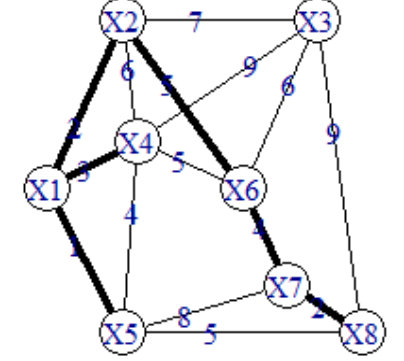
$\{u_{23}, u_{58}, u_{63}, u_{78}\},$ $\{c_{23} = 7, c_{58} = 5, c_{63} = 6, c_{78} = 2\}.$	$S_5 = \{x_1, x_2, x_4, x_5, x_6, x_7\}$	
$\{u_{23}, u_{58}, u_{63}, u_{78}\},$ $\{c_{23} = 7, c_{58} = 5, c_{63} = 6, c_{78} = 2\}.$	$S_5 = \{x_1, x_2, x_4, x_5, x_6, x_7\}$	
$U = \{u_{15}, u_{12}, u_{14}, u_{26}, u_{67}, u_{78}\}.$	$S_6 = \{x_1, x_2, x_4, x_5, x_6, x_7, x_8\}.$	

Рисунок 1.9

7.1. Выпишем все ребра, кроме ребер, выбранных ранее ( $u_{12}, u_{14}, u_{15}, u_{26}, u_{67}, u_{78}$ ), инцидентные вершинам, принадлежащим множеству  $S_6$ , но не оканчивающиеся в нем, и их веса.

$$\{u_{23}, u_{43}, u_{63}, u_{83}\},$$

$$\{c_{23} = 7, c_{43} = 9, c_{63} = 6, c_{83} = 9\}.$$

7.2. Выберем из них ребро с минимальным весом:  $u_{63}$  ( $c_{63} = 6$ ) и включим его в будущее дерево (рис. 1.10).

7.3. Внесем конечные вершины ребра  $u_{63}$  в множество  $S_6$  (которых нет в этом множестве). Получим множество  $S_7$ :

$$S_7 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}.$$

7. 4. Так как  $(n_7 = 8) = (n = 8)$ , построение дерева закончено.

Получили минимальное остовное дерево, количество ребер которого равно  $n - 1 = 8 - 1 = 7$ .

Перечень ребер, входящих в остовное дерево

$$U = \{u_{15}, u_{12}, u_{14}, u_{26}, u_{67}, u_{78}, u_{63}\}.$$

Суммарный вес ребер равен

$$c_{15} + c_{12} + c_{14} + c_{26} + c_{67} + c_{78} + c_{63} = 1 + 2 + 3 + 5 + 4 + 2 + 6 = 23.$$

$\{u_{23}, u_{43}, u_{63}, u_{83}\},$ $\{c_{23} = 7, c_{43} = 9, c_{63} = 6, c_{83} = 9\}.$	$S_6 = \{x_1, x_2, x_4, x_5, x_6, x_7, x_8\}.$	
$\{u_{23}, u_{43}, u_{63}, u_{83}\},$ $\{c_{23} = 7, c_{43} = 9, c_{63} = 6, c_{83} = 9\}.$	$S_6 = \{x_1, x_2, x_4, x_5, x_6, x_7, x_8\}.$	
$U = \{u_{15}, u_{12}, u_{14}, u_{26}, u_{67}, u_{78}, u_{63}\}.$	$S_7 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}.$	

Рисунок 1.10

## РЕШЕНИЕ ЗАДАЧИ В СИСТЕМЕ R-PROJECT

Содержание данного раздела является результатом применения технологии R Markdown. Это позволяет объединить строки кода, заголовки, пояснения, комментарии, результаты расчетов и рисунки в единый форматированный документ. Текст, приведённый ниже,

фактически получен «автоматически», т.е. путем выполнения команд, содержащихся в файле типа *rmd*, поэтому его форматирование отличается от форматирования других разделов пособия. Оформление текста, приведенного ниже, выполнено с незначительным количеством ручной правки .

Строки кода написаны на светло сером фоне, результаты содержатся в строках, которые начинаются с двух символов # (решётка или шарп [англ. sharp]). Комментарии содержатся в строках, которые начинаются с символа # и отображаются курсивом.

Для работы с графами используем пакет `igraph`

```
library("igraph")
```

Функции для отображения MST в исходном графе

Текст функций `vectorsInVectorSeq` и `plotMSTwithinG` был размещен на ресурсе <https://www.smartana.co.uk/mst-tutorial/MST.html>

Вспомогательная функция для проверки вхождения матрицы в матрицу

```
# Tests whether row vectors are within a matrix; identifies rows in matrix accordingly
vectorsInVectorSeq <- function(testVectorMatrix, seqVectorMatrix) {
  r <- rep(FALSE,nrow(seqVectorMatrix))
  for (k in 1:nrow(testVectorMatrix)) {
    testElement = testVectorMatrix[k,]
    r <- r | apply(seqVectorMatrix, 1, function(x, test)
isTRUE(all.equal(x, test)), testElement)
  }
  return(r)
}
```

Функция для вывода графа

Аргументом функции `plotMSTwithinG` является `e_mst` - граф, который является подграфом графа `G`.

Функция позволяет выводить исходный граф `G`, при этом ребра, входящие в подграф `e_mst`, выделяются цветом и шириной.

```
plotMSTwithinG <- function(e_mst)
{
  e.G = get.edges(G,1:ecount(G)) # matrix of edges of G

  mst_idx = vectorsInVectorSeq(e_mst, e.G) # row indicis in G of MST
  #Now we can finally highlight the MST within the original graph G.
```



```

ecol <- rep("SkyBlue2", ecount(G)) # default edge colour
ecol[mst_idx] <- "red" # colour for MST

ew <- rep(2,ecount(G)) # default edge width
ew[mst_idx] <- 5 # width for MST edges

plot(G, layout = coords, edge.color=ecol, edge.width=ew)
return(TRUE)
}

```

Создаем переменную `edges`, содержащую данные о ребрах заданного графа

```

edges <- read.table(textConnection(
  "from to weight
  1 2 2
  1 4 3
  1 5 1
  2 3 7
  2 4 6
  2 6 5
  3 4 9
  3 6 6
  3 8 9
  4 5 4
  4 6 5
  5 7 8
  5 8 5
  6 7 4
  7 8 2

  "),header=TRUE)
# смотрим результат
# edges

```

Создаём неориентированный граф с именем `G` из множества ребер `edges`.

Для этого используем функцию **`graph.data.frame`**, которая может использовать в качестве аргумента набор данных, который описывает ребра сети. В первых двух колонках содержатся идентификаторы начальной и конечной вершины для каждого ребра. В следующих колонках находятся параметры ребра (вес, тип, метка, другое).

Приписываем каждому ребру заданный вес.

```

G = graph.data.frame(edges[,1:2], directed = FALSE)
E(G)$weight = edges[,3] # specify the actual weight

```

Результат работы функции - объект типа `igraph`.

Описание объекта **igraph** начинается с четырех букв:

**D** или **U** — для направленного или ненаправленного графа соответственно.

**N** — для именованного графа (где у узлов есть атрибут `name`).

**W** — для взвешенного графа (где у связей есть атрибут `weight`).

**B** — для двустороннего графа (где у узлов есть атрибут `type`).

Следующие два числа (8 и 15) указывают количество вершин и ребер в графе. В описании также приводятся свойства вершин и ребер, например:

(g/c) — свойство-строка на уровне графа;

(v/c) — свойство-строка на уровне вершины;

(e/n) — свойство-число на уровне ребра.

**G**

```
## IGRAPH e446e34 UNW- 8 15 --
## + attr: name (v/c), weight (e/n)
## + edges from e446e34 (vertex names):
## [1] 1--2 1--4 1--5 2--3 2--4 2--6 3--4 3--6 3--8 4--5 4--6 5--7
5--8 6--7 7--8
```

Для улучшения наглядности изменяем некоторые свойства вершин графа:

```
V(G)$size = 30 # размер вершины
V(G)$color='white' # цвет вершины
V(G)$name <- paste("X",1:8,sep="") # имя вершины
```

В качестве меток ребер используем их вес

```
E(G)$label = E(G)$weight # labels for the weight
```

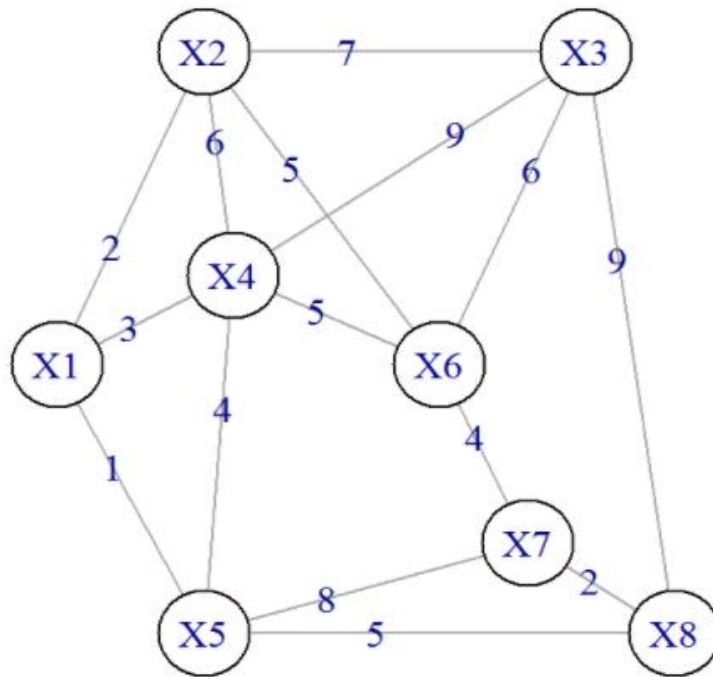
Координатам вершин присваиваем значения:

```
L<-20
##### 1 2 3 4 5 6 7 8
x = c( 0, 5, 18, 6, 5, 13, 16, 21)*L
y = c( 6, 13, 13, 8, 0, 6, 2, 0)*L
coords = cbind(x,y)
```

**Строим “график” графа G**

Параметр `layout` определяет расположение вершин на графике.

```
plot(G, layout = coords)
```



Наличие установленных свойств можно посмотреть

G

```
## IGRAPH e446e34 UNW- 8 15 --
## + attr: name (v/c), size (v/n), color (v/c), weight (e/n), label
(e/n)
## + edges from e446e34 (vertex names):
## [1] X1--X2 X1--X4 X1--X5 X2--X3 X2--X4 X2--X6 X3--X4 X3--X6 X3--
X8 X4--X5
## [11] X4--X6 X5--X7 X5--X8 X6--X7 X7--X8
```

В описании добавились новые свойства

Свойства вершин :

name и color - (v/c) — свойство-строка;

size - (v/n) — свойство-число;

Свойства ребер : weight и label - (e/n) — свойство-число.

### Встроенная функция `minimum.spanning.tree`

Для нахождения MST используем функцию **minimum.spanning.tree**.

Функция содержится в пакете `igraph`.

Результатом работы является подграф, т.е. граф, вершины которого являются подмножеством вершин исходного графа.

Подграф связного графа является минимальным остовным деревом, если этот подграф является деревом, включающим все вершины исходного

графа, и сумма его весов ребер является минимальной среди всех подграфов исходного графа, обладающих указанными свойствами.

**Аргументы функции:**

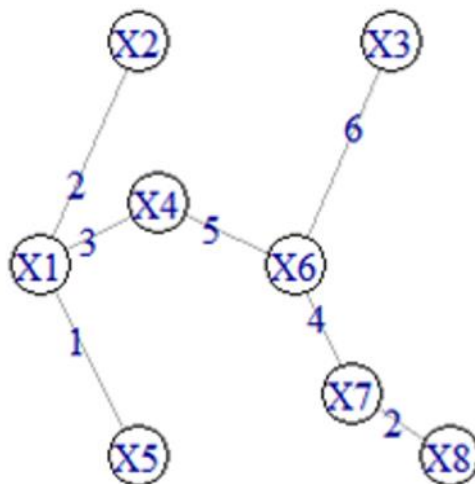
**graph** - исходный граф.

**weights** - вес ребер графа. Он может быть пропущен, если будет выбран режим работы “unweighted”.

**algorithm** - алгоритм, используемый для расчетов. Значение “unweighted” может быть использовано для невзвешенных графов, значение “prim” может быть указывается для использования алгоритма Прима для взвешенных графов. Если этот параметр не определен (NULL), функция пытается выбрать алгоритм автоматически.

```
mst = minimum.spanning.tree(G)
```

Можно построить “график” полученного остовного дерева  
`plot(mst, layout = coords)`

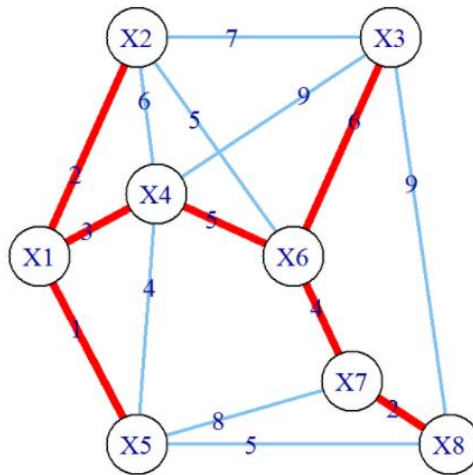


Полученная картинка не является удачной, поскольку на ней не изображен исходный граф.

### Визуализация MST

Для визуализации используем функцию **plotMSTwithinG**, описанную выше.

```
resplotMSTwithinG<-FALSE  
# Формируем матрицу из ребер найденного MST.  
e.mst = get.edges(mst,1:ecount(mst)) # matrix of edges of MST  
# Активируем функцию  
resplotMSTwithinG<-plotMSTwithinG(e.mst)
```



*# Результат работы функции. Если TRUE(Истина), картинка успешно построена.*

```
resplotMSTwithinG
```

```
## [1] TRUE
```

#### Общие свойства найденного MST

```
mst
```

```
## IGRAPH e48aa43 UNW- 8 7 --
```

```
## + attr: name (v/c), size (v/n), color (v/c), weight (e/n), label (e/n)
```

```
## + edges from e48aa43 (vertex names):
```

```
## [1] X1--X2 X1--X4 X1--X5 X3--X6 X4--X6 X6--X7 X7--X8
```

Как и следовало ожидать, количество вершин, совпало с количеством вершин исходного графа, количество ребер на единицу меньше количества вершин.

*# Перечень ребер найденного MST*

```
E(mst)
```

```
## + 7/7 edges from e48aa43 (vertex names):
```

```
## [1] X1--X2 X1--X4 X1--X5 X3--X6 X4--X6 X6--X7 X7--X8
```

*# Веса ребер найденного MST*

```
E(mst)$weight
```

```
## [1] 2 3 1 6 5 4 2
```

*# Сумма весов ребер найденного MST*

```
sum(E(mst)$weight)
```

```
## [1] 23
```

Заметим, что решение, полученное с помощью встроенной функции MST, отличается от решения, полученного *методом ближайшего соседа*. Однако суммарный вес ребер, входящих в каждый из этих графов совпадает. Очевидно, что при аналитическом решении задачи на четвертом шаге можно было включить в решение любое ребро из трех, имеющих одинаковый вес, в результате будем иметь три варианта решения.

### ЗАДАНИЕ

Лабораторная работа содержит одну задачу. Решить задачу двумя способами:

1. Аналитически (методом «ближайшего соседа»);
2. Средствами R-project (можно использовать встроенные функции).

### ЗАДАЧА

Пусть дан связный неориентированный взвешенный граф  $G(V,E)$  с 9 вершинами рис.1.11. Построить для графа  $G$  минимальное остовное дерево. Матрицы весов приведены на рис. 1.12 – 1.15.

Исходные данные каждого варианта определяются в соответствии с таблицей 1.2.

Таблица 1.2

Вариант №	Схема	Матрица весов
1	1	1-1
2	1	1-2
3	1	1-3
4	1	1-4
5	2	2-1
6	2	2-2
7	2	2-3
8	2	2-4
9	3	3-1
10	3	3-2
11	3	3-3
12	3	3-4
13	4	4-1
14	4	4-2
15	4	4-3
16	4	4-4

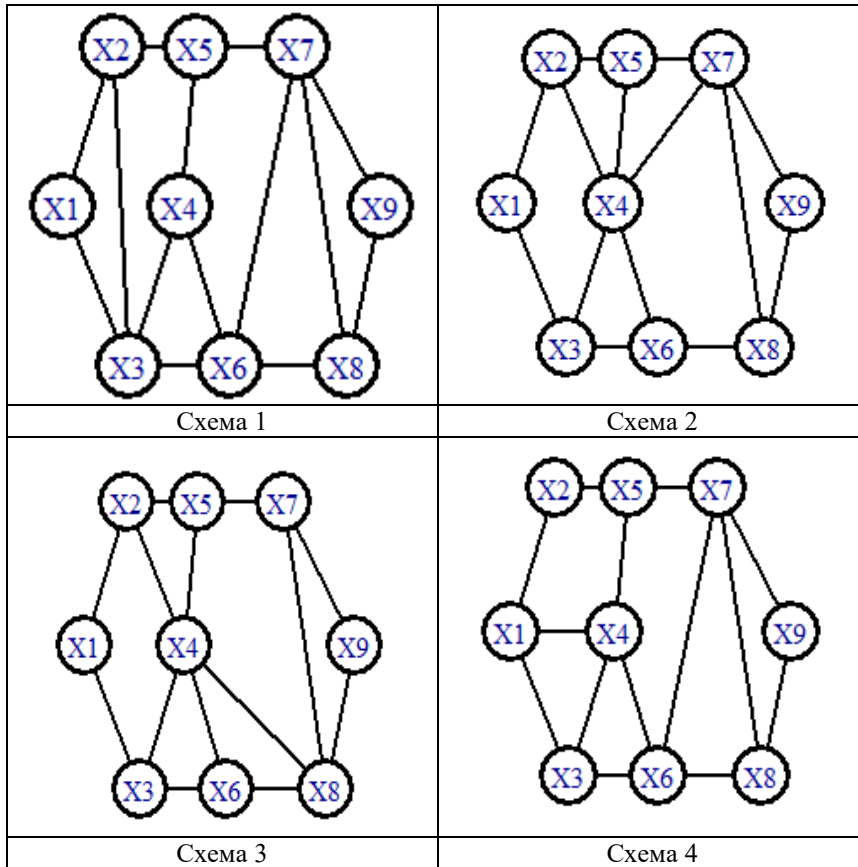


Рисунок 1.11

Матрица 1-1										Матрица 1-2										
	X1	X2	X3	X4	X5	X6	X7	X8	X9		X1	X2	X3	X4	X5	X6	X7	X8	X9	
X1	.	15	15	.	.	.	.	.	.	X1	.	19	18	.	.	.	.	.	.	.
X2	15	.	8	.	4	.	.	.	.	X2	19	.	10	.	8	.	.	.	.	.
X3	15	8	.	8	.	16	.	.	.	X3	18	10	.	9	.	17	.	.	.	.
X4	.	.	8	.	7	7	.	.	.	X4	.	.	9	.	7	10	.	.	.	.
X5	.	4	.	7	.	.	8	.	.	X5	.	8	.	7	.	.	9	.	.	.
X6	.	.	16	7	.	.	8	17	.	X6	.	.	17	10	.	.	13	19	.	.
X7	.	.	.	.	8	8	.	10	7	X7	.	.	.	.	9	13	.	12	10	.
X8	.	.	.	.	.	17	10	.	19	X8	.	.	.	.	.	19	12	.	18	.
X9	.	.	.	.	.	.	7	19	.	X9	.	.	.	.	.	.	10	18	.	.

Матрица 1-3										Матрица 1-4										
	X1	X2	X3	X4	X5	X6	X7	X8	X9		X1	X2	X3	X4	X5	X6	X7	X8	X9	
X1	.	21	20	.	.	.	.	.	.	X1	.	21	21	.	.	.	.	.	.	.
X2	21	.	12	.	8	.	.	.	.	X2	21	.	15	.	10	.	.	.	.	.
X3	20	12	.	9	.	18	.	.	.	X3	21	15	.	11	.	20	.	.	.	.
X4	.	.	9	.	9	11	.	.	.	X4	.	.	11	.	12	11	.	.	.	.
X5	.	8	.	9	.	.	10	.	.	X5	.	10	.	12	.	.	13	.	.	.
X6	.	.	18	11	.	.	13	19	.	X6	.	.	20	11	.	.	15	21	.	.
X7	.	.	.	.	10	13	.	13	11	X7	.	.	.	.	13	15	.	15	13	.
X8	.	.	.	.	.	19	13	.	21	X8	.	.	.	.	.	21	15	.	22	.
X9	.	.	.	.	.	.	11	21	.	X9	.	.	.	.	.	.	13	22	.	.

Рисунок 1.12

Матрица 2-1										Матрица 2-2										
	X1	X2	X3	X4	X5	X6	X7	X8	X9		X1	X2	X3	X4	X5	X6	X7	X8	X9	
X1	.	15	15	.	.	.	.	.	.	X1	.	19	18	.	.	.	.	.	.	.
X2	15	.	.	8	4	.	.	.	.	X2	19	.	.	10	8	.	.	.	.	.
X3	15	.	.	8	.	16	.	.	.	X3	18	.	.	9	.	17	.	.	.	.
X4	.	8	8	.	17	7	11	.	.	X4	.	10	9	.	17	10	12	.	.	.
X5	.	4	.	17	.	.	6	.	.	X5	.	8	.	17	.	.	11	.	.	.
X6	.	.	16	7	.	.	.	17	.	X6	.	.	17	10	.	.	.	19	.	.
X7	.	.	.	11	6	.	.	10	17	X7	.	.	.	12	11	.	.	12	20	.
X8	.	.	.	.	.	17	10	.	19	X8	.	.	.	.	.	19	12	.	18	.
X9	.	.	.	.	.	.	17	19	.	X9	.	.	.	.	.	.	20	18	.	.

Матрица 2-3										Матрица 2-4										
	X1	X2	X3	X4	X5	X6	X7	X8	X9		X1	X2	X3	X4	X5	X6	X7	X8	X9	
X1	.	21	20	.	.	.	.	.	.	X1	.	21	21	.	.	.	.	.	.	.
X2	21	.	.	12	8	.	.	.	.	X2	21	.	.	15	10	.	.	.	.	.
X3	20	.	.	9	.	18	.	.	.	X3	21	.	.	11	.	20	.	.	.	.
X4	.	12	9	.	19	11	13	.	.	X4	.	15	11	.	22	11	16	.	.	.
X5	.	8	.	19	.	.	11	.	.	X5	.	10	.	22	.	.	13	.	.	.
X6	.	.	18	11	.	.	.	19	.	X6	.	.	20	11	.	.	.	21	.	.
X7	.	.	.	13	11	.	.	13	21	X7	.	.	.	16	13	.	.	15	23	.
X8	.	.	.	.	.	19	13	.	21	X8	.	.	.	.	.	21	15	.	22	.
X9	.	.	.	.	.	.	21	21	.	X9	.	.	.	.	.	.	23	22	.	.

Рисунок 1.13

Матрица 3-1										Матрица 3-2										
	X1	X2	X3	X4	X5	X6	X7	X8	X9		X1	X2	X3	X4	X5	X6	X7	X8	X9	
X1	.	15	15	.	.	.	.	.	.	X1	.	19	18	.	.	.	.	.	.	.
X2	15	.	.	8	14	.	.	.	.	X2	19	.	.	10	18	.	.	.	.	.
X3	15	.	.	18	.	16	.	.	.	X3	18	.	.	19	.	17	.	.	.	.
X4	.	8	18	.	17	7	.	11	.	X4	.	10	19	.	17	10	.	12	.	.
X5	.	14	.	17	.	.	6	.	.	X5	.	18	.	17	.	.	11	.	.	.
X6	.	.	16	7	.	.	.	17	.	X6	.	.	17	10	.	.	.	19	.	.
X7	.	.	.	.	6	.	.	10	17	X7	.	.	.	.	11	.	.	12	20	.
X8	.	.	.	11	.	17	10	.	19	X8	.	.	.	12	.	19	12	.	18	.
X9	.	.	.	.	.	.	17	19	.	X9	.	.	.	.	.	.	20	18	.	.

Матрица 3-3										Матрица 3-4										
	X1	X2	X3	X4	X5	X6	X7	X8	X9		X1	X2	X3	X4	X5	X6	X7	X8	X9	
X1	.	21	20	.	.	.	.	.	.	X1	.	21	21	.	.	.	.	.	.	.
X2	21	.	.	12	18	.	.	.	.	X2	21	.	.	15	20	.	.	.	.	.
X3	20	.	.	19	.	18	.	.	.	X3	21	.	.	21	.	20	.	.	.	.
X4	.	12	19	.	19	11	.	13	.	X4	.	15	21	.	22	11	.	16	.	.
X5	.	18	.	19	.	.	11	.	.	X5	.	20	.	22	.	.	13	.	.	.
X6	.	.	18	11	.	.	.	19	.	X6	.	.	20	11	.	.	.	21	.	.
X7	.	.	.	.	11	.	.	13	21	X7	.	.	.	.	13	.	.	15	23	.
X8	.	.	.	13	.	19	13	.	21	X8	.	.	.	16	.	21	15	.	22	.
X9	.	.	.	.	.	.	21	21	.	X9	.	.	.	.	.	.	23	22	.	.

Рисунок 1.14



Матрица 4-1										Матрица 4-2									
	X1	X2	X3	X4	X5	X6	X7	X8	X9		X1	X2	X3	X4	X5	X6	X7	X8	X9
X1	.	15	15	8	.	.	.	.	.	X1	.	19	18	10	.	.	.	.	.
X2	15	.	.	.	14	.	.	.	.	X2	19	.	.	.	18	.	.	.	.
X3	15	.	.	18	.	16	.	.	.	X3	18	.	.	19	.	17	.	.	.
X4	8	.	18	.	7	7	.	.	.	X4	10	.	19	.	7	10	.	.	.
X5	.	14	.	7	.	.	8	.	.	X5	.	18	.	7	.	.	9	.	.
X6	.	.	16	7	.	.	8	17	.	X6	.	.	17	10	.	.	13	19	.
X7	.	.	.	.	8	8	.	10	7	X7	.	.	.	.	9	13	.	12	10
X8	.	.	.	.	.	17	10	.	19	X8	.	.	.	.	.	19	12	.	18
X9	.	.	.	.	.	.	7	19	.	X9	.	.	.	.	.	.	10	18	.
Матрица 4-3										Матрица 4-4									
	X1	X2	X3	X4	X5	X6	X7	X8	X9		X1	X2	X3	X4	X5	X6	X7	X8	X9
X1	.	21	20	12	.	.	.	.	.	X1	.	21	21	15	.	.	.	.	.
X2	21	.	.	.	18	.	.	.	.	X2	21	.	.	.	20	.	.	.	.
X3	20	.	.	19	.	18	.	.	.	X3	21	.	.	21	.	20	.	.	.
X4	12	.	19	.	9	11	.	.	.	X4	15	.	21	.	12	11	.	.	.
X5	.	18	.	9	.	.	10	.	.	X5	.	20	.	12	.	.	13	.	.
X6	.	.	18	11	.	.	13	19	.	X6	.	.	20	11	.	.	15	21	.
X7	.	.	.	.	10	13	.	13	11	X7	.	.	.	.	13	15	.	15	13
X8	.	.	.	.	.	19	13	.	21	X8	.	.	.	.	.	21	15	.	22
X9	.	.	.	.	.	.	11	21	.	X9	.	.	.	.	.	.	13	22	.

Рисунок 1.15

## ЛАБОРАТОРНАЯ РАБОТА 2. ПОИСК КРАТЧАЙШЕГО ПУТИ НА ГРАФЕ

**Цель:** Освоить на практике решение задачи нахождения кратчайшего пути в графе методом Дейкстры

Задача о кратчайшем пути формулируется следующим образом: дан связный неориентированный взвешенный граф  $G = (V, E)$ , не имеющий параллельных ребер. Для любых двух выделенных вершин графа требуется найти кратчайший путь между ними и указать этот путь.

Данный метод позволяет решать многие практические задачи. Одной из них является задача организации перевозки грузов от одного поставщика к нескольким потребителям при разветвленной сети дорог, при которых суммарный прогон транспортных средств будет минимальным (или суммарная стоимость перевозок будет минимальной). Знание кратчайших путей в сети улиц позволяет планировщикам транспортных сетей более точно моделировать потоки движения автомобилей по улицам города.

Для решения задачи нахождения кратчайших путей разработано много методов. Наиболее популярным является метод Дейкстры, который применим для графов с положительными весами ребер.

## МЕТОД ДЕЙКСТРЫ

Пусть  $x_j, j = \overline{1, n}$  – вершины графа. Пусть начальная вершина<sup>1</sup> – это вершина  $x_1$ . Метод заключается в присвоении вершинам графа временных меток  $L(x_j)$ , а затем по определенному правилу замены их постоянными метками  $L^*(x_j)$ . *Постоянные метки* – кратчайшие расстояния от заданной вершины ( $x_1$ ) до всех остальных ( $x_j$ ).

Предварительно всем вершинам графа, кроме вершины  $x_1$  (источник), требуется присвоить временные метки, равные  $\infty$ :  $L(x_j) = \infty, j = \overline{2, n}$ , а вершине  $x_1$  – постоянную метку  $L^*(x_1) = 0$ . Затем следует выполнить последовательность следующих действий:

1. Определить множество вершин графа, смежных с вершиной  $x_1$ :

$$S(x_1) = \{x_j\}, j = \overline{1, k}.$$

2. Для каждой вершины, принадлежащей множеству  $S(x_1)$ , вычислить новые временные метки  $L(x_j)$ , равные

$$\min\{L(x_j), L^*(x_i) + R_{ij}\},$$

где  $L(x_j)$  – старая временная метка вершины  $x_j$ ,  $L^*(x_i)$  – новая постоянная метка вершины  $x_i$ ,  $R_{ij}$  – вес ребра  $(x_i, x_j)$ .

3. Из всех имеющихся временных меток выбрать наименьшую и сделать ее постоянной для своей вершины.

4. Определить множество вершин графа, смежных с вершиной  $x_i$ , которой на предыдущем шаге была присвоена постоянная метка:

$$S(x_i) = \{x_j\}, j = \overline{1, k}.$$

Процесс повторяется до тех пор, пока все вершины не будут иметь постоянные метки.

Если требуется найти кратчайшее расстояние от вершины-источника до некоторой фиксированной вершины<sup>2</sup>, процесс вычисления новых временных и постоянных меток заканчивается после того, как вершине-цели будет присвоена новая постоянная метка.

Описанный выше алгоритм позволяет найти значения минимальных (кратчайших) расстояний от вершины-источника до остальных, чтобы найти кратчайший путь (последовательность смежных ребер), необходимо выполнить «обратный ход» от вершины-цели (target) до вершины-источника (source). Метод нахождения кратчайшего пути будет показан на примере.

### ПРИМЕР

Используя метод Дейкстры, определить кратчайшие пути графа, представленного на рис.2.1, от вершины  $x_1$  до всех остальных. Матрица смежности графа задана в табл. 2.1.

---

<sup>1</sup> Общеизвестным названием этой вершины является термин – вершина – источник (source).

<sup>2</sup> Общеизвестным названием этой вершины является термин – вершина – цель (target).

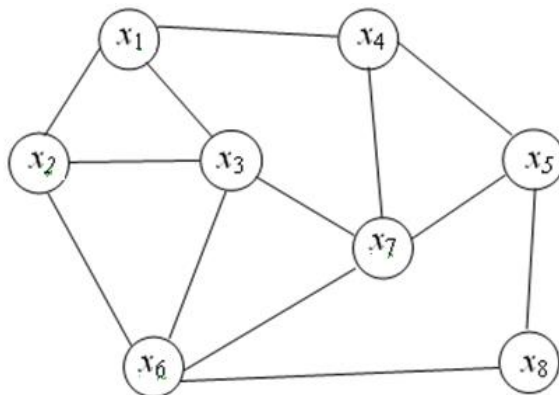


Рисунок 2.1

Таблица 2.1

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$
$x_1$		34	5	4				
$x_2$	34		26			60		
$x_3$	5	26				15	31	
$x_4$	4				9		18	
$x_5$				9			21	20
$x_6$		60	15				30	14
$x_7$			31	18	21	30		
$x_8$					20	14		

Результаты вычислений будем представлять в таблице (табл. 2.2).

Предварительно всем вершинам графа, кроме вершины  $x_1$  присвоим временные метки:  $L(x_2) = L(x_3) = L(x_4) = L(x_5) = L(x_6) = L(x_7) = \infty$ , а вершине  $x_1$  – постоянную метку  $L^*(x_1) = 0$ . Занесем их в нулевой столбец табл. 2.2.

Таблица 2.2

Шаги	0	1	2	3	4	5	6	7
$x_1$	0*							
$x_2$	$\infty$							
$x_3$	$\infty$							
$x_4$	$\infty$							
$x_5$	$\infty$							

Шаги \	0	1	2	3	4	5	6	7
$x_6$	$\infty$							
$x_7$	$\infty$							
$x_8$	$\infty$							

Выполним последовательность следующих шагов:

*Шаг 1*

1. Определим множество вершин графа, смежных с вершиной  $x_1$  (рис. 2.1, табл. 2.1):  $S(x_1) = \{x_2, x_3, x_4\}$ .

2. Для каждой вершины, принадлежащей множеству  $S(x_1)$ , вычислим новые временные метки  $L(x_j)$ , равные  $\min\{L(x_j), L^*(x_1) + R_{1j}\}$ , где  $L(x_j)$  – старая временная метка вершины  $x_j$ ,  $L^*(x_1) = 0$  – постоянная метка вершины  $x_1$ ,  $R_{1j}$  – вес ребра  $(x_1, x_j)$ .

$L(x_j)$		$L^*(x_1) + R_{1j}$		$\min\{L^*(x_1) + R_{1j}\}$
$L(x_2)$	$= \infty$	$L^*(x_1) + R_{12} =$	$0 + 34 = 34$	$\min\{\infty, 34\} = 34$
$L(x_3)$	$= \infty$	$L^*(x_1) + R_{13} =$	$0 + 5 = 5$	$\min\{\infty, 5\} = 5$
$L(x_4)$	$= \infty$	$L^*(x_1) + R_{14} =$	$0 + 4 = 4$	$\min\{\infty, 4\} = 4$ .

Вершинам  $x_2, x_3, x_4$  присвоим новые временные метки, которые вычислили выше; метки вершин  $x_5, x_6, x_7$  остаются без изменения, т.е. будем иметь:  $L(x_2) = 34, L(x_3) = 5, L(x_4) = 4, L(x_5) = L(x_6) = L(x_7) = L(x_8) = \infty$ . Занесем их в первый столбец табл. 2.3, причем обновленные метки выделим.

Таблица 2.3

Шаги \	0	1	2	3	4	5	6	7
$x_1$	<b>0*</b>							
$x_2$	$\infty$	<b>34</b>						
$x_3$	$\infty$	<b>5</b>						
$x_4$	$\infty$	<b>4</b>						
$x_5$	$\infty$	$\infty$						
$x_6$	$\infty$	$\infty$						
$x_7$	$\infty$	$\infty$						
$x_8$	$\infty$	$\infty$						

3. Из всех имеющихся временных меток выберем наименьшую и сделаем ее постоянной для своей вершины:  $\min\{34, 5, \mathbf{4}, \infty, \infty, \infty, \infty\}$ . Эта метка соответствует вершине  $x_4$ :  $L^*(x_4) = 4$ . Отметим ее звездочкой в столбце 1 (табл. 2.4).

Таблица 2.4

Шаги \	0	1	2	3	4	5	6	7
$x_1$	0*							
$x_2$	$\infty$	<b>34</b>						
$x_3$	$\infty$	<b>5</b>						
$x_4$	$\infty$	<b>4*</b>						
$x_5$	$\infty$	$\infty$						
$x_6$	$\infty$	$\infty$						
$x_7$	$\infty$	$\infty$						
$x_8$	$\infty$	$\infty$						

Геометрический смысл указанных действий показан в Приложении 1 на рисунках П.1.1 - П.1.3

На этом и последующих рисунках вершины с временными метками обозначены кружками, вершины с постоянными метками – квадратиками и треугольниками, при это треугольником обозначена вершина, которая стала постоянной на первом или предыдущем шаге.

### Шаг 2

1. Определим множество вершин графа, смежных с вершиной  $x_4$  (см. рис. 2.1, табл. 2.1):  $S(x_4) = \{x_1, x_5, x_7\}$ . Вершина  $x_1$  уже имеет постоянную метку, т.е.  $S(x_4) = \{x_1^*, x_5, x_7\}$ .

2. Для каждой вершины, принадлежащей множеству  $S(x_4)$  и не имеющей постоянной метки, вычислим новые временные метки  $L(x_j)$ , равные  $\min\{L(x_j), L^*(x_4) + R_{4j}\}$ , где  $L(x_j)$  – старая временная метка вершины  $x_j$ ,  $L^*(x_4) = 4$  – постоянная метка вершины  $x_4$ ,  $R_{4j}$  – вес ребра  $(x_4, x_j)$  (см. табл. 2.1).

$L(x_j)$		$L^*(x_4) + R_{4j}$		$\min\{L^*(x_4) + R_{4j}\}$
$L(x_5)$	$= \infty$	$L^*(x_4) + R_{45} =$	$4 + 9 = 13$	$\min\{\infty, 13\} = 13$
$L(x_7)$	$= \infty$	$L^*(x_4) + R_{47} =$	$4 + 18 = 22$	$\min\{\infty, 22\} = 22.$

3. Вершинам  $x_5, x_7$  присвоим новые временные метки, которые вычислили выше; метки вершин  $x_6, x_8$  остаются без изменения, т.е. будем иметь:  $L(x_2) = 34, L(x_3) = 5, L(x_5) = 13, L(x_7) = 22, L(x_6) = L(x_8) = \infty$ . Занесем их во второй столбец таблицы 2.5, причем обновленные метки выделим.

4. Из всех имеющихся временных меток выберем наименьшую и сделаем ее постоянной для своей вершины:  $\min\{34, 5, 13, \infty, 22, \infty\}$ . Эта метка соответствует вершине  $x_3$ :  $L^*(x_3) = 5$ . Отметим ее звездочкой в столбце 2 (табл. 2.6).

Таблица 2.5

Шаги	0	1	2	3	4	5	6	7
$x_1$	0*							
$x_2$	$\infty$	<b>34</b>	34					
$x_3$	$\infty$	<b>5</b>	5					
$x_4$	$\infty$	<b>4*</b>						
$x_5$	$\infty$	$\infty$	<b>13</b>					
$x_6$	$\infty$	$\infty$	$\infty$					
$x_7$	$\infty$	$\infty$	<b>22</b>					
$x_8$	$\infty$	$\infty$	$\infty$					

Таблица 2.6

Шаги	0	1	2	3	4	5	6	7
$x_1$	0*							
$x_2$	$\infty$	<b>34</b>	34					
$x_3$	$\infty$	<b>5</b>	5*					
$x_4$	$\infty$	<b>4*</b>						
$x_5$	$\infty$	$\infty$	<b>13</b>					
$x_6$	$\infty$	$\infty$	$\infty$					
$x_7$	$\infty$	$\infty$	<b>22</b>					
$x_8$	$\infty$	$\infty$	$\infty$					

Геометрический смысл указанных действий показан в Приложении 2 на рисунках П.2.1.-П.2.3

### Шаг 3

1. Определим множество вершин графа, смежных с вершиной  $x_3$  (рис.2.1, табл. 2.1):  $S(x_3) = \{x_1, x_2, x_6, x_7\}$ . Вершина  $x_1$  уже имеет постоянную метку, т.е.  $S(x_3) = \{x_1^*, x_5, x_6, x_7\}$ .

2. Для каждой вершины, принадлежащей множеству  $S(x_3)$  и не имеющей постоянной метки, вычислим новые временные метки  $L(x_j)$ , равные  $\min\{L(x_j), L^*(x_3) + R_{3j}\}$ , где  $L(x_j)$  – старая временная метка вершины  $x_j$ ,  $L^*(x_3) = 5$  – постоянная метка вершины  $x_3$ ,  $R_{3j}$  – вес ребра  $(x_3, x_j)$  (табл.2.1).

$L(x_j)$		$L^*(x_3) + R_{3j}$		$\min\{L^*(x_3) + R_{3j}\}$
$L(x_2)$	= 34	$L^*(x_3) + R_{32} =$	$5 + 26 = 31$	$\min\{34, 31\} = 31$
$L(x_6)$	= $\infty$	$L^*(x_3) + R_{36} =$	$5 + 15 = 20$	$\min\{\infty, 20\} = 20$
$L(x_7)$	= $\infty$	$L^*(x_3) + R_{37} =$	$5 + 31 = 36$	$\min\{22, 36\} = 22$ .

Вершинам  $x_2, x_6$  присвоим новые временные метки, которые вычислили выше; метки вершин  $x_5, x_7, x_8$  остаются без изменения, т.е. будем иметь:  $L(x_2) = 31, L(x_5) = 13, L(x_6) = 20, L(x_7) = 22, L(x_8) = \infty$ .

Занесем их в третий столбец табл. 2.7, причем обновленные метки выделим.

Таблица 2.7

Шаги	0	1	2	3	4	5	6	7
$x_1$	0*							
$x_2$	$\infty$	<b>34</b>	34	<b>31</b>				
$x_3$	$\infty$	<b>5</b>	5*					
$x_4$	$\infty$	<b>4*</b>						
$x_5$	$\infty$	$\infty$	<b>13</b>	13				
$x_6$	$\infty$	$\infty$	$\infty$	<b>20</b>				
$x_7$	$\infty$	$\infty$	<b>22</b>	22				
$x_8$	$\infty$	$\infty$	$\infty$	$\infty$				

3. Из всех имеющихся временных меток выберем наименьшую и сделаем ее постоянной для своей вершины:  $\min\{31, \mathbf{13}, 20, 22, \infty\} = 13$ . Эта метка соответствует вершине  $x_5$ :  $L^*(x_5) = 13$ . Отметим ее звездочкой в столбце 3 (табл. 2.8).

Геометрический смысл указанных действий показан в Приложении 3 на рисунке П.3.1.-П.3.3

#### Шаг 4

1. Определим множество вершин графа, смежных с вершиной  $x_5$  (см. рис. 2.1, табл. 2.1):  $S(x_5) = \{x_4, x_7, x_8\}$ . Вершина  $x_4$  уже имеет постоянную метку, т.е.  $S(x_5) = \{x_4^*, x_7, x_8\}$ .

Таблица 2.8

Шаги	0	1	2	3	4	5	6	7
$x_1$	0*							
$x_2$	$\infty$	<b>34</b>	34	<b>31</b>				
$x_3$	$\infty$	<b>5</b>	5*					
$x_4$	$\infty$	<b>4*</b>						
$x_5$	$\infty$	$\infty$	<b>13</b>	13*				
$x_6$	$\infty$	$\infty$	$\infty$	<b>20</b>				
$x_7$	$\infty$	$\infty$	<b>22</b>	22				
$x_8$	$\infty$	$\infty$	$\infty$	$\infty$				

2. Для каждой вершины, принадлежащей множеству  $S(x_5)$  и не имеющей постоянной метки, вычислим новые временные метки  $L(x_j)$ , равные  $\min\{L(x_j), L^*(x_5) + R_{5j}\}$ , где  $L(x_j)$  – старая временная метка вершины  $x_j$ ,  $L^*(x_5) = 13$  – постоянная метка вершины  $x_5$ ,  $R_{5j}$  – вес ребра  $(x_5, x_j)$  (см. табл. 2.1).

$L(x_j)$	$L^*(x_5) + R_{5j}$	$\min\{L^*(x_5) + R_{5j}\}$
$L(x_7) = 22$	$L^*(x_5) + R_{57} = 13 + 21 = 34$	$\min\{22, 34\} = 22$
$L(x_8) = \infty$	$L^*(x_5) + R_{58} = 13 + 20 = 33$	$\min\{\infty, 33\} = 33$

Вершине  $x_8$  присвоим новую временную метку, которую вычислили выше; метки вершин  $x_2, x_6, x_7$  остаются без изменения, т.е. будем иметь:  $L(x_2) = 31, L(x_6) = 20, L(x_7) = 22, L(x_8) = 33$ . Занесем их в четвертый столбец табл. 2.9, причем обновленные метки выделим.



4. Из всех имеющихся временных меток выберем наименьшую и сделаем ее постоянной для своей вершины:  $\min\{31, \mathbf{20}, 22, 33\} = 20$ . Эта метка соответствует вершине  $x_6$ :  $L^*(x_6) = 20$ . Отметим ее звездочкой в столбце 4 (табл. 2.10).

Таблица 2.9

Шаги	0	1	2	3	4	5	6	7
$x_1$	0*							
$x_2$	$\infty$	<b>34</b>	34	<b>31</b>	31			
$x_3$	$\infty$	<b>5</b>	5*					
$x_4$	$\infty$	<b>4*</b>						
$x_5$	$\infty$	$\infty$	<b>13</b>	13*				
$x_6$	$\infty$	$\infty$	$\infty$	<b>20</b>	20			
$x_7$	$\infty$	$\infty$	<b>22</b>	22	22			
$x_8$	$\infty$	$\infty$	$\infty$	$\infty$	<b>33</b>			

Таблица 2.10

Шаги	0	1	2	3	4	5	6	7
$x_1$	0*							
$x_2$	$\infty$	<b>34</b>	34	<b>31</b>	31			
$x_3$	$\infty$	<b>5</b>	5*					
$x_4$	$\infty$	<b>4*</b>						
$x_5$	$\infty$	$\infty$	<b>13</b>	13*				
$x_6$	$\infty$	$\infty$	$\infty$	<b>20</b>	20*			
$x_7$	$\infty$	$\infty$	<b>22</b>	22	22			
$x_8$	$\infty$	$\infty$	$\infty$	$\infty$	<b>33</b>			

### Шаг 5

1. Определим множество вершин графа, смежных с вершиной  $x_5$  (см. рис. 2.1, табл. 2.1):

$S(x_6) = \{x_2, x_3, x_7, x_8\}$ . Вершина  $x_3$  уже имеет постоянную метку, т.е.  $S(x_6) = \{x_2^*, x_3, x_7, x_8\}$ .

2. Для каждой вершины, принадлежащей множеству  $S(x_6)$  и не имеющей постоянной метки, вычислим новые временные метки  $L(x_j)$ , равные  $\min\{L(x_j), L^*(x_6) + R_{6j}\}$ , где  $L(x_j)$  – старая временная метка вершины  $x_j$ ,  $L^*(x_6) = 20$  – постоянная метка вершины  $x_6$ ,  $R_{6j}$  – вес ребра  $(x_6, x_j)$  (см. табл. 2.1).

$L(x_j)$		$L^*(x_6) + R_{6j}$		$\min\{L^*(x_6) + R_{6j}\}$
$L(x_7) = 22$		$L^*(x_6) + R_{67} =$	$20+30 = 50$	$\min\{22, 50\} = 22$
$L(x_8) = 33$		$L^*(x_6) + R_{68} =$	$20+14 = 34$	$\min\{33, 34\} = 33$

Все временные метки остаются без изменения. Перенесем их в пятый столбец табл. 2.11.

Таблица 2.11

Шаги	0	1	2	3	4	5	6	7
$x_1$	0*							
$x_2$	$\infty$	<b>34</b>	34	<b>31</b>	31	31		
$x_3$	$\infty$	<b>5</b>	5*					
$x_4$	$\infty$	<b>4*</b>						
$x_5$	$\infty$	$\infty$	<b>13</b>	13*				
$x_6$	$\infty$	$\infty$	$\infty$	<b>20</b>	20*			
$x_7$	$\infty$	$\infty$	<b>22</b>	22	22	22		
$x_8$	$\infty$	$\infty$	$\infty$	$\infty$	<b>33</b>	33		

3. Из всех имеющихся временных меток выберем наименьшую и сделаем ее постоянной для своей вершины:  $\min\{31, 22, 33\} = 22$ . Эта метка соответствует вершине  $x_7$ :  $L^*(x_7) = 22$ . Отметим ее звездочкой в столбце 5 (табл. 2.12).

Таблица 2.12

Шаги	0	1	2	3	4	5	6	7
$x_1$	0*							
$x_2$	$\infty$	<b>34</b>	34	<b>31</b>	31	31		

Шаги \	0	1	2	3	4	5	6	7
$x_3$	$\infty$	<b>5</b>	5*					
$x_4$	$\infty$	<b>4*</b>						
$x_5$	$\infty$	$\infty$	<b>13</b>	13*				
$x_6$	$\infty$	$\infty$	$\infty$	<b>20</b>	20*			
$x_7$	$\infty$	$\infty$	<b>22</b>	22	22	22*		
$x_8$	$\infty$	$\infty$	$\infty$	$\infty$	<b>33</b>	33		

### Шаг 6

1. Определим множество вершин графа, смежных с вершиной  $x_7$  (см. рис.2.1, табл. 2.1):  $S(x_7) = \{x_3, x_4, x_5, x_6\}$ . Все вершины имеют постоянные метки, т.е.  $S(x_7) = \{x_3^*, x_4^*, x_5^*, x_6^*\}$ .

2. Для вершин, принадлежащих множеству  $S(x_7)$ , вычислять новые временные метки не надо. Перенесем их в шестой столбец табл. 2.13.

Таблица 2.13

Шаги \	0	1	2	3	4	5	6	7
$x_1$	<b>0*</b>							
$x_2$	$\infty$	<b>34</b>	34	<b>31</b>	31	31	31	
$x_3$	$\infty$	<b>5</b>	5*					
$x_4$	$\infty$	<b>4*</b>						
$x_5$	$\infty$	$\infty$	<b>13</b>	13*				
$x_6$	$\infty$	$\infty$	$\infty$	<b>20</b>	20*			
$x_7$	$\infty$	$\infty$	<b>22</b>	22	22	22*		
$x_8$	$\infty$	$\infty$	$\infty$	$\infty$	<b>33</b>	33	33	

5. Из всех имеющихся временных меток выберем наименьшую и сделаем ее постоянной для своей вершины:  $\min\{31, 33\} = 31$ .

Эта метка соответствует вершине  $x_2$ :

$$L^*(x_2) = 31.$$

Отметим ее звездочкой в столбце 6 (табл. 2.14).

Таблица 2.14

Шаги	0	1	2	3	4	5	6	7
$x_1$	0*							
$x_2$	$\infty$	<b>34</b>	34	<b>31</b>	31	31	31*	
$x_3$	$\infty$	<b>5</b>	5*					
$x_4$	$\infty$	<b>4*</b>						
$x_5$	$\infty$	$\infty$	<b>13</b>	13*				
$x_6$	$\infty$	$\infty$	$\infty$	<b>20</b>	20*			
$x_7$	$\infty$	$\infty$	<b>22</b>	22	22	22*		
$x_8$	$\infty$	$\infty$	$\infty$	$\infty$	<b>33</b>	33	33	

*Шаг 7*

1. Определим множество вершин графа, смежных с вершиной  $x_2$  (см. рис. 2.1, табл. 2.1):  $S(x_2) = \{x_1, x_3, x_6\}$ .  
Все вершины имеют постоянные метки, т.е.  $S(x_2) = \{x_1^*, x_3^*, x_6^*\}$ .
2. Имеется одна старая временная метка. Перенесем ее в седьмой столбец табл. 2.15.

Таблица 2.15

Шаги	0	1	2	3	4	5	6	7
$x_1$	0*							
$x_2$	$\infty$	<b>34</b>	34	<b>31</b>	31	31	31*	
$x_3$	$\infty$	<b>5</b>	5*					
$x_4$	$\infty$	<b>4*</b>						
$x_5$	$\infty$	$\infty$	<b>13</b>	13*				
$x_6$	$\infty$	$\infty$	$\infty$	<b>20</b>	20*			
$x_7$	$\infty$	$\infty$	<b>22</b>	22	22	22*		
$x_8$	$\infty$	$\infty$	$\infty$	$\infty$	<b>33</b>	33	33	33

3. Сделаем последнюю временную метку постоянной. Отметим ее звездочкой в столбце 7 (табл. 2.16).

Таблица 2.16

Шаги	0	1	2	3	4	5	6	7
$x_1$	0*							
$x_2$	$\infty$	<b>34</b>	34	<b>31</b>	31	31	31*	
$x_3$	$\infty$	<b>5</b>	5*					
$x_4$	$\infty$	<b>4*</b>						
$x_5$	$\infty$	$\infty$	<b>13</b>	13*				
$x_6$	$\infty$	$\infty$	$\infty$	<b>20</b>	20*			
$x_7$	$\infty$	$\infty$	<b>22</b>	22	22	22*		
$x_8$	$\infty$	$\infty$	$\infty$	$\infty$	<b>33</b>	33	33	33*

Процесс расстановки меток закончен. Значения их дают кратчайшие расстояния от исходной вершины  $x_1$  до всех остальных:

$$L(x_2) = 31, L(x_3) = 5, L(x_4) = 4, L(x_5) = 13, L(x_6) = 20, L(x_7) = 33.$$

Чтобы найти кратчайшие пути от вершины  $x_1$  до всех остальных вершин, следует использовать соотношение:

$$L^*(x_j) = L^*(x_i) + R_{ij}, \quad (2.1)$$

в котором вершина  $x_i$  предшествует вершине  $x_j$ .

Найдем кратчайший путь в графе от вершины - источника  $x_1$  до вершины-цели  $x_8$ . Для этого выполним «обратный ход».

1. Определим, какая вершина предшествует вершине  $x_8$  в кратчайшем пути.

Для этого используем соотношение (2.1), где  $j = 8$ :

$$L^*(x_8) = L^*(x_i) + R_{i8} \quad (2.2)$$

Вершина  $x_8$  имеет только две смежные вершины:  $x_5$  и  $x_6$ , из которых можно попасть в вершину  $x_8$ :  $S(x_5, x_6)$ . Определим, какая из этих вершин удовлетворяет соотношению (2.2).

$$L^*(x_8) = \mathbf{33},$$

$$L^*(x_5) = 13 \quad R_{58} = 20 \quad L^*(x_8) = L^*(x_5) + R_{58} = \mathbf{33},$$

$$L^*(x_6) = 20 \quad R_{68} = 14 \quad L^*(x_8) = L^*(x_6) + R_{68} = 34.$$

Как видно, только вершина  $x_5$  удовлетворяет соотношению (2.2). Следовательно, в кратчайшем пути вершине  $x_8$  предшествует вершина  $x_5$  (рис.2.4).

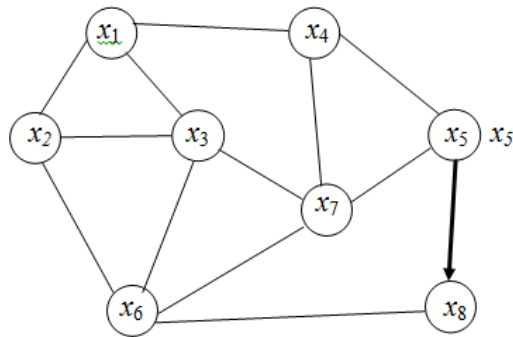


Рисунок 2.4

2. Определим, какая вершина предшествует вершине  $x_5$  в кратчайшем пути. Вершина  $x_5$  имеет две смежные вершины:  $x_4, x_7$  (вершина  $x_8$  уже вошла в искомый путь и поэтому не рассматривается):  $S(x_4, x_7)$ . Определим, какая из этих вершин удовлетворяет соотношению (2.1).

$$L^*(x_5) = \mathbf{13},$$

$$L^*(x_4) = 4 \quad R_{45} = 9 \quad L^*(x_5) = L^*(x_4) + R_{45} = \mathbf{13},$$

$$L^*(x_7) = 22 \quad R_{75} = 21 \quad L^*(x_8) = L^*(x_6) + R_{68} = 33.$$

Как видно, соотношению (2.1) удовлетворяет только вершина  $x_4$ . Следовательно, в кратчайшем пути вершине  $x_5$  предшествует вершина  $x_4$  (рис. 2.5).

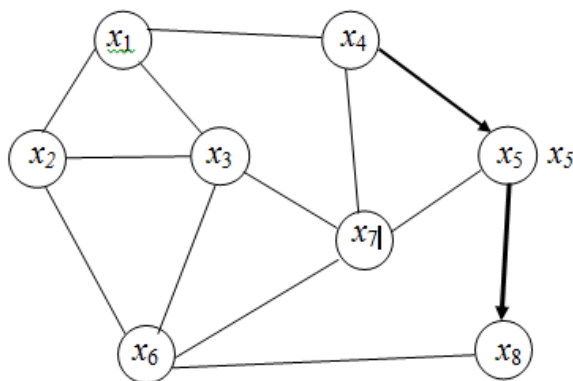


Рисунок 2.5

3. Определим, какая вершина предшествует вершине  $x_4$  в кратчайшем пути. Вершина  $x_4$  имеет две смежные вершины:  $x_1$  и  $x_7$  (вершина  $x_5$  уже вошла в искомый путь и поэтому не рассматривается):  $S$

$(x_1, x_8, x_5)$ . Определим, какая из этих вершин удовлетворяет соотношению (2.1).

$$L^*(x_4) = 4,$$

$$L^*(x_1) = 0 \quad R_{14} = 4 \quad L^*(x_4) = L^*(x_1) + R_{14} = 4,$$

$$L^*(x_7) = 22 \quad R_{74} = 21 \quad L^*(x_4) = L^*(x_7) + R_{74} = 40.$$

Как видно, соотношению (2.1) удовлетворяет только вершина  $x_4$ . Следовательно, в кратчайшем пути вершине  $x_4$  предшествует вершина  $x_1$  (рис. 2.6), которая является исходной.

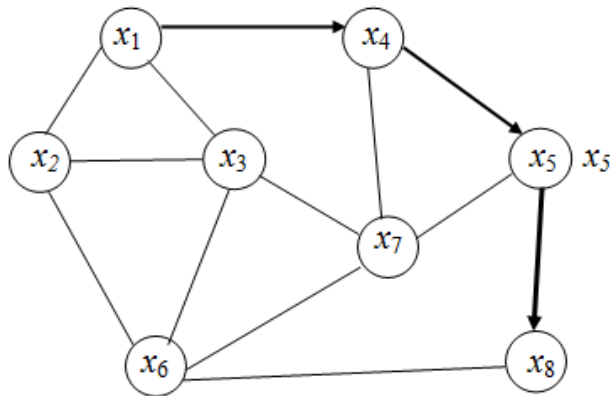


Рисунок 2.6

Таким образом, кратчайший путь от вершины  $x_1$  до вершины  $x_8$  проходит по вершинам  $x_1, x_4, x_5, x_8$  и длина этого пути равна **33**.

Очевидно, что каждый из путей из вершины  $x_1$  до любой вершины, входящей в построенный кратчайший путь от вершины  $x_1$  в вершину  $x_8$ , тоже будет оптимальным.

В качестве вершины-цели можно рассматривать любую вершину, кратчайший путь до неё от вершины - источника  $x_1$  может быть найден с помощью соответствующего «обратного хода».

### Решение задачи в табличном процессоре MS Excel

Для контроля задачу решим задачу в MS Excel с помощью надстройки «Поиск решения».

Для этого может быть использована следующая модель.

Пусть  $C$  - квадратная матрица, элементы которой определяются соотношением:

$$c_{i,j} = \begin{cases} w_{i,j} & \text{если вершины } x_i \text{ и } x_j \text{ соединены ребром} \\ M & \text{, в противоположном случае} \end{cases},$$

где  $j = \overline{1, n}$ ,  $i = \overline{1, n}$ ,  $w_{i,j}$  - вес ребра, соединяющего вершины  $x_i$  и  $x_j$ ,

$M$  – большое число, значительно превосходящее вес ребер.

Обозначим индекс начальной вершины символом  $S$  (source), индекс конечной вершины - символом  $T$  (target).

Введем матрицу  $\{x_{ij}\}$   $j = \overline{1, n}$ ,  $i = \overline{1, n}$ ,

где  $x_{ij}$  принимает значения ноль или единица.

$$x_{i,j} = \begin{cases} 1, & \text{если путь выходит из } x_i \text{ и входит в } x_j \\ 0, & \text{в противоположном случае} \end{cases}.$$

При таких обозначениях задача о кратчайшем пути может быть сформулирована в виде задачи на оптимизацию с целевой функцией, определяемой соотношением

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij} \cdot x_{ij} \rightarrow \min. \quad (2.3)$$

Ограничения определяются следующими соотношениями

$$\sum_{j=1}^n x_{sj} = 1 \quad (2.4)$$

$$\sum_{i=1}^n x_{iT} = 1 \quad (2.5)$$

$$\sum_{j=1}^n x_{kj} = \sum_{i=1}^n x_{ik} \quad \text{для } k \notin \{S, T\} \quad (2.6)$$

$$x_{ij} \in \{0,1\} \quad (2.7)$$

Выполнение ограничения (2.4) обеспечивает выход из стартовой вершины  $x_s$ , ограничения (2.5) – достижение вершины  $x_T$ .

Соотношение (2.6) содержит требование для всех остальных вершин (не стартовой и не финишной) – количество входов в вершину равно количеству выходов.

Рассмотрим решение приведенного выше примера в MS Excel с помощью надстройки «Поиск решения».

Расчеты приведены на рис.2.11- рис.2.15.

Искомые значения  $\{x_{ij}\}$   $j = \overline{1, n}$ ,  $i = \overline{1, n}$  - в диапазоне ячеек A8:I16.

В начальный момент в этом диапазоне находятся нули. В диапазоне ячеек C17: I17 находятся суммы указанных значений по столбцам, а в диапазоне ячеек J9:J16 находятся суммы указанных значений по строкам.

Значения матрицы весов  $c_{i,j}$   $j = \overline{1, n}$ ,  $i = \overline{1, n}$  находятся в диапазоне ячеек A23: I31.



В качестве числа  $M$  в задаче выбрано число 100. В ячейке I33 находится значение целевой функции, определяемое соотношением (2.3), которое может быть вычислено по формуле  

$$=СУММПРОИЗВ(B9:I16;B24:I31).$$

	A	B	C	D	E	F	G	H	I	J	K	L	
7	Пункты отправления	Пункты назначения								Количество исходов	Исток		
8		$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$				
9	$x_1$	0	0	0	0	0	0	0	0	0	1		
10	$x_2$	0	0	0	0	=СУММ(B9:I9)		0	0	0			
11	$x_3$	0	0	0	0	0	0	0	0	0			
12	$x_4$	0	0	0	0	0	0	0	0	0			
13	$x_5$	0	0	0	0	0	0	0	0	0			
14	$x_6$	=СУММ(C9:C16)			0	0	0	0	0	0			
15	$x_7$	0	0	0	0	0	0	0	0	0			
16	$x_8$	0	0	0	=ТРАНСП(J10:J15)		0	0	0	0			
17		0	0	0	0	0	0	0	0	Количество входов			
18									1	Сток			
19			0	0	0	0	0	0		Количество			
20										исходов			

Рисунок 2.11 - Фрагмент листа MS Excel с начальными данными.

	A	B	C	D	E	F	G	H	I	J	
21		Матрица весов									
22	Пункты отправления	Пункты назначения									
23		$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$		
24	$x_1$	100	34	5	4	100	100	100	100		
25	$x_2$	34	100	26	100	100	60	100	100		
26	$x_3$	5	26	100	100	100	15	31	100		
27	$x_4$	4	100	100	100	9	100	18	100		
28	$x_5$	100	100	100	9	100	100	21	20		
29	$x_6$	100	60	15	100	100	100	30	14		
30	$x_7$	100	100	31	18	21	30	100	100		
31	$x_8$	100	100	100	100	20	14	100	100		
32											
33		=СУММПРОИЗВ(B9:I16;B24:I31)								0 <-ЦФ	

Рисунок 2.12 - Фрагмент листа MS Excel с матрицей весов и целевой функцией.

Соотношения (2.4 - 2.7) учтены в ограничениях диалогового окна надстройки «Поиск решения».

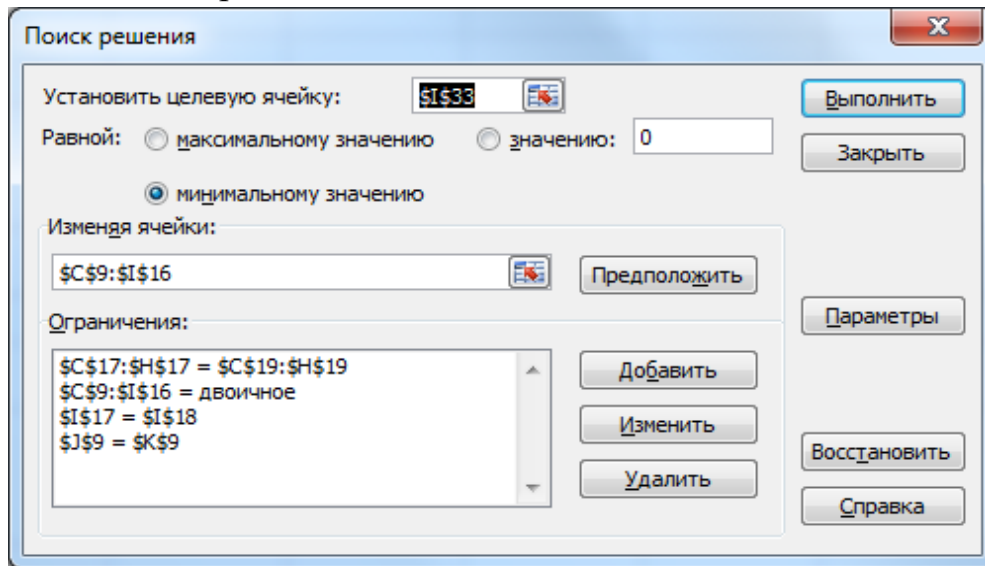


Рисунок 2.13 - Диалоговое окно надстройки «Поиск решения».

Результаты решения приведены на рис.2.14 и рис.2.15. Кратчайший путь читается с помощью анализа матрицы в диапазоне ячеек A8:I16. Наличие значений, равных единице, позволяет восстановить искомый путь от вершины  $x_1$  до вершины  $x_8$ , который проходит по вершинам  $x_1, x_4, x_5, x_8$ . Длина этого пути равна 33, соответствующее значение приведено в ячейке I33. Заметим, что полученное решение совпадает с решением, полученным методом Дейкстры.

	A	B	C	D	E	F	G	H	I	J	K	L	
7	Пункты отправления	Пункты назначения								Количество исходов	Исток		
8		$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$				
9	$x_1$	0	0	0	1	0	0	0	0	1	1		
10	$x_2$	0	0	0	0	0	0	0	0	0			
11	$x_3$	0	0	0	0	0	0	0	0	0			
12	$x_4$	0	0	0	0	1	0	0	0	1			
13	$x_5$	0	0	0	0	0	0	0	1	1			
14	$x_6$	0	0	0	0	0	0	0	0	0			
15	$x_7$	0	0	0	0	0	0	0	0	0			
16	$x_8$	0	0	0	0	0	0	0	0	0			
17		0	0	0	1	1	0	0	1	Количество входов			
18									1	Сток			
19			0	0	1	1	0	0		Количество исходов			

Рисунок 2.14 - Фрагмент листа MS Excel с результатами решения.

	A	B	C	D	E	F	G	H	I	J	
32											
33		=СУММПРОИЗВ(B9:I16;B24:I31)								33	<-ЦФ

Рисунок 2.15 - Фрагмент листа MS Excel с результатами решения.

## РЕШЕНИЕ ЗАДАЧИ В СИСТЕМЕ R-PROJECT

### Для работы с графами используем пакет `igraph`

Подключаем пакет

```
library("igraph")
```

```
##
## Attaching package: 'igraph'
##
## The following objects are masked from 'package:stats':
##
##   decompose, spectrum
##
## The following object is masked from 'package:base':
##
##   union
```

### Ввод исходных данных

Создаем переменную `edges`, содержащую данные о ребрах заданного графа:

```
edges <- read.table(textConnection(
"from to weight
1 2 34
1 3 5
1 4 4
2 3 26
2 6 60
3 6 15
3 7 31
4 5 9
4 7 18
5 7 21
5 8 20
6 7 30
6 8 14
"),header=TRUE)
```

Создаём неориентированный граф с именем `G` из множества ребер `edges`:

```
G <- graph.data.frame(edges[,1:2] , directed = FALSE)
```

Приписываем каждому ребру заданный вес

```
E(G)$weight <- edges[,3] # specify the actual weight
```

Для улучшения наглядности изменяем некоторые свойства вершин графа:

```
V(G)$size <- 30 # размер вершины  
V(G)$color <- 'white' # цвет вершины  
V(G)$name <- paste("X",1:8,sep="") # имя вершины
```

Изменяем некоторые свойства ребер графа:

```
E(G)$color <- "SkyBlue2"  
E(G)$width <- 2
```

В качестве меток ребер используем их вес

```
E(G)$label <- E(G)$weight # Labels for the weight
```

Координатам вершин присваиваем значения

```
L <- 20  
##### 1 2 3 4 5 6 7 8  
x <- c(10, 0, 20, 30, 55, 13, 40, 50)*L  
y <- c(30, 20, 20, 30, 20, 0, 12, 0)*L  
coords <- cbind(x,y)
```

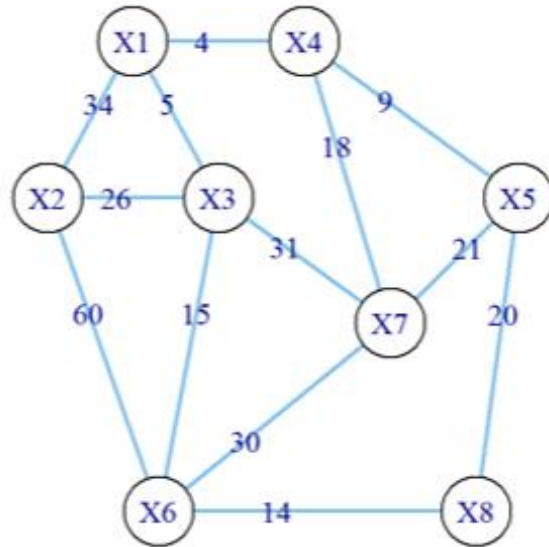
Смотрим свойства графа

```
G  
## IGRAPH 35b56f5 UNW- 8 13 --  
## + attr: name (v/c), size (v/n), color (v/c), weight (e/n), color  
(e/c),  
## | width (e/n), label (e/n)  
## + edges from 35b56f5 (vertex names):  
## [1] X1--X2 X1--X3 X1--X4 X2--X3 X2--X6 X3--X6 X3--X7 X4--X5 X4--  
X7 X5--X7  
## [11] X5--X8 X6--X7 X6--X8
```

## Строим “график” графа G

Параметр layout определяет расположение вершин на графике.

```
plot(G, layout = coords)
```



### “Кратчайший путь” в пакете igraph

В пакете `igraph` содержится ряд функций, позволяющих найти кратчайшие расстояния между вершинами графа. Эти функции объединены заголовком **“Shortest (directed or undirected) paths between vertices”**

Для нахождения нахождения кратчайшего пути можно использовать функцию `shortest_paths`.

Результатом работы этой функции является единственный кратчайший путь между двумя вершинами. Для взвешенного графа используется метод Дейкстры (Dijkstra’s algorithm).

#### Аргументы функции:

**graph** - исходный граф;

**from** - константа-число, вершина *от* которой пути вычисляются;

**to** - числовой вектор, определяет вершины, *до* которых кратчайшие пути будут вычисляться. По умолчанию равно  $V(\text{graph})$ , т.е. включает все вершины;

**mode** = `c(“out”, “all”, “in”)`, режим работы для ориентированного графа;

**weights** - вес ребер графа. Значение по умолчанию `NULL`.

**output** - символьный скаляр, определяющий вид отчета о кратчайшем пути. Возможные значения `output = c(“vpath”, “epath”, “both”)`. Значение `“vpath”` определяет вывод результата в виде последовательности вершин, значение `“epath”` определяет вывод результата в виде последовательности ребер. `“both”` определяет вывод результата в виде последовательности ребер и вершин;

**predecessors** - логический скаляр (FALSE или TRUE), определяющий вывод вершин, предшествующих данной в дереве, при построении кратчайших путей.

Находим кратчайший путь от вершины **X1** до вершины **X8** с помощью функции **shortest\_paths**.

```
ShortPth<-shortest_paths(G, 1, to = 8,weights = E(G)$weight,  
output="both", predecessors=TRUE)
```

Результатом работы является объект типа list  
typeof(ShortPth)

```
## [1] "list"
```

```
summary(ShortPth)
```

```
##          Length Class      Mode  
## vpath      1    -none-    list  
## epath      1    -none-    list  
## predecessors 8    igraph.vs numeric  
## inbound_edges 0    -none-    NULL
```

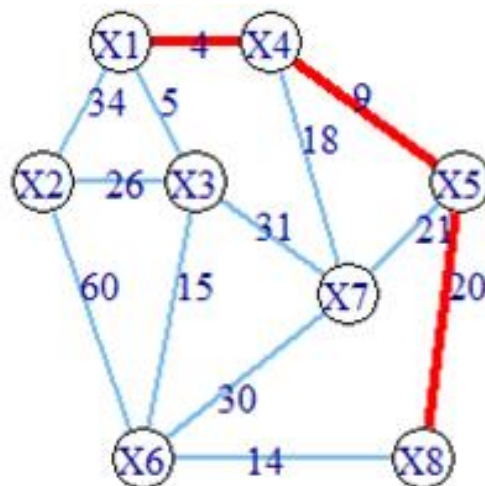
## Визуализация кратчайшего пути

Для всех ребер устанавливаем ширину и цвет.

```
E(G)$color <- "SkyBlue2"  
E(G)$width <- 2
```

Для ребер, входящих в кратчайший путь, устанавливаем особенную ширину и цвет.

```
E(G, path=unlist(ShortPth[[1]]))$color <- "red"  
E(G, path=unlist(ShortPth[[1]]))$width <- 4
```



Показываем кратчайший путь на графе

```
plot(G, edge.label = paste(" ", E(G)$weight), layout =
coords)
```

## Общие свойства найденного кратчайшего пути

Покажем список вершин, через которые проходит кратчайший путь  
ShortPth\$vpath

```
## [[1]]
## + 4/8 vertices, named, from 35b56f5:
## [1] X1 X4 X5 X8
```

Покажем список ребер, которые составляют кратчайший путь  
ShortPth\$epath

```
## [[1]]
## + 3/13 edges from 35b56f5 (vertex names):
## [1] X1--X4 X4--X5 X5--X8
```

Суммарный вес ребер, входящих в кратчайший путь  
sum(E(G, path=unlist(ShortPth[[1]]))\$weight)

```
## [1] 33
```

Покажем список предшествующих вершин  
ShortPth\$predecessors

```
## + 8/8 vertices, named, from 35b56f5:
## [1] X1 X3 X1 X1 X4 X3 X4 X5
```

Заметим, что такой список позволяет выполнить «обратный ход» от вершины-цели (target). до вершины-источника (source) при аналитическом решении.

## Кратчайший путь для всех вершин

Для нахождения кратчайшего пути между всевозможными парами вершин графа может использоваться функция **distances**.

**Аргументы функции:**

**algorithm** - алгоритм, используемый для расчетов. Возможные значения могут быть выбраны из списка с("automatic", "unweighted", "dijkstra", "bellman-ford", "johnson").

Другие аргументы в основном совпадают с параметрами функции **shortest\_paths**.

Найдем матрицу со всевозможными значениями расстояний

```
Gdistances<-distances(G, mode="out")
```

Матрица имеет вид, приведенный ниже.

```
Gdistances
```

```

##      X1 X2 X3 X4 X5 X6 X7 X8
## X1   0 31  5  4 13 20 22 33
## X2  31  0 26 35 44 41 53 55
## X3   5 26  0  9 18 15 27 29
## X4   4 35  9  0  9 24 18 29
## X5  13 44 18  9  0 33 21 20
## X6  20 41 15 24 33  0 30 14
## X7  22 53 27 18 21 30  0 41
## X8  33 55 29 29 20 14 41  0

```

Для сравнения этого результата с аналитическим решением выделим первую строку.

```
Gdistances[1,]
```

```

## X1 X2 X3 X4 X5 X6 X7 X8
##  0 31  5  4 13 20 22 33

```

Содержащиеся в ней числа равны расстояниям от вершины **X1** до остальных вершин, которые являются значениями постоянных меток в методе Дейкстры.

## Вывод

Полученное решение полностью совпадает с решением, полученным аналитически методом Дейкстры.

## ЗАДАНИЕ

Лабораторная работа содержит одну задачу.

### ЗАДАЧА

Определить кратчайший путь между вершинами графа. Исходные данные для построения графа определены в предыдущей работе (рис.2.11-2.15). Исходные данные для определения начальной (source) и конечной (target) вершин приведены в таблице 2. 17.

Таблица 2.17

Вариант №	Source	Target
1	X1	X8
2	X3	X9
3	X3	X7
4	X2	X8
5	X1	X8
6	X1	X9
7	X3	X7
8	X2	X8



Вариант №	Source	Target
9	X1	X8
10	X3	X9
11	X3	X7
12	X2	X8
13	X1	X8
14	X1	X9
15	X3	X7
16	X2	X8

Решить задачу тремя способами:

4. Аналитически (методом Дейкстры);
5. Средствами MS Excel (надстройка «Поиск решения»);
6. Средствами R-project (можно использовать встроенные функции).

### СПИСОК ЛИТЕРАТУРЫ

1. *Петухова Н.М.* Математическое моделирование на автомобильном транспорте [Электронный ресурс] : учеб. пособие / Н. М. Петухова. - СПб.: Горн. ун-т, 2013. - 225 с. - Б. ц.с [http://irbis.spmi.ru/jirbis2/index.php?option=com\\_irbis&view=irbis&Itemid=402&task=set\\_static\\_req&bns\\_string=NWPIB,ELC,ZAPIS&req\\_irb=<.>I=%2D078736<.>](http://irbis.spmi.ru/jirbis2/index.php?option=com_irbis&view=irbis&Itemid=402&task=set_static_req&bns_string=NWPIB,ELC,ZAPIS&req_irb=<.>I=%2D078736<.>)
2. *Оре О.* Теория графов.– М.: Наука, 1980. 336 с.
3. *Харари Ф.* Теория графов.– М.: Мир, 1973. 300 с.
4. R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. [Электронный ресурс] /Режим доступа URL <https://www.R-project.org/>, свободный. - Загл. с экрана.
5. <https://www.smartana.co.uk/mst-tutorial/MST.html>
6. Csardi G, Nepusz T: The igraph software package for complex network research, InterJournal, Complex Systems 1695. 2006. /Режим доступа <http://igraph.org> , свободный. - Загл. с экрана.
7. JJ Allaire and Yihui Xie and Jonathan McPherson and Javier Luraschi and Kevin Ushey and Aron Atkins and Hadley Wickham and Joe Cheng and Winston Chang and Richard Iannone (2020). rmarkdown: Dynamic Documents for R. R package version 2.6. /Режим доступа <https://rmarkdown.rstudio.com>, свободный. - Загл. с экрана

## ПРИЛОЖЕНИЕ 1

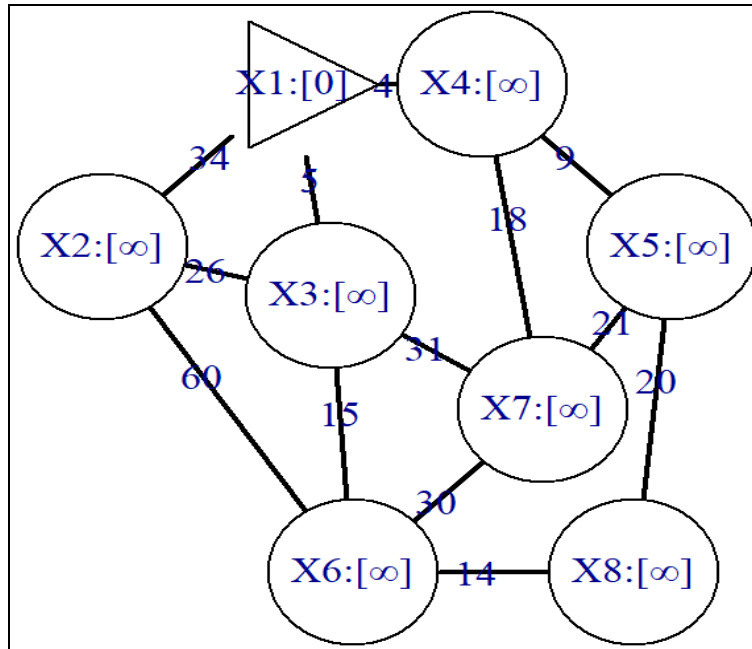


Рисунок П.1.1 - Шаг 1-0

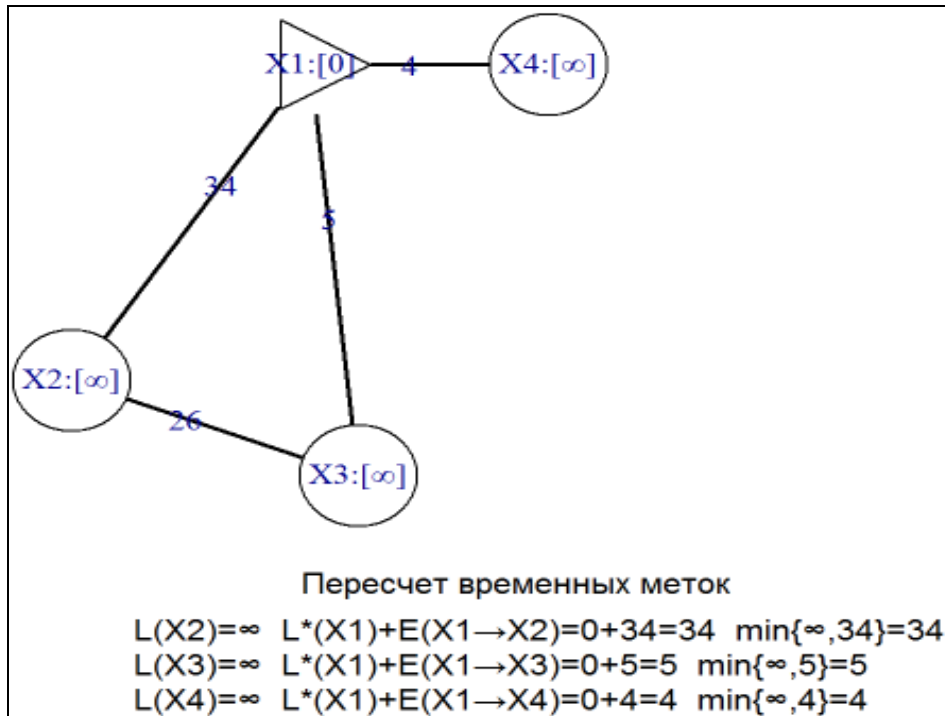


Рисунок П.1.2 - Шаг 1-1

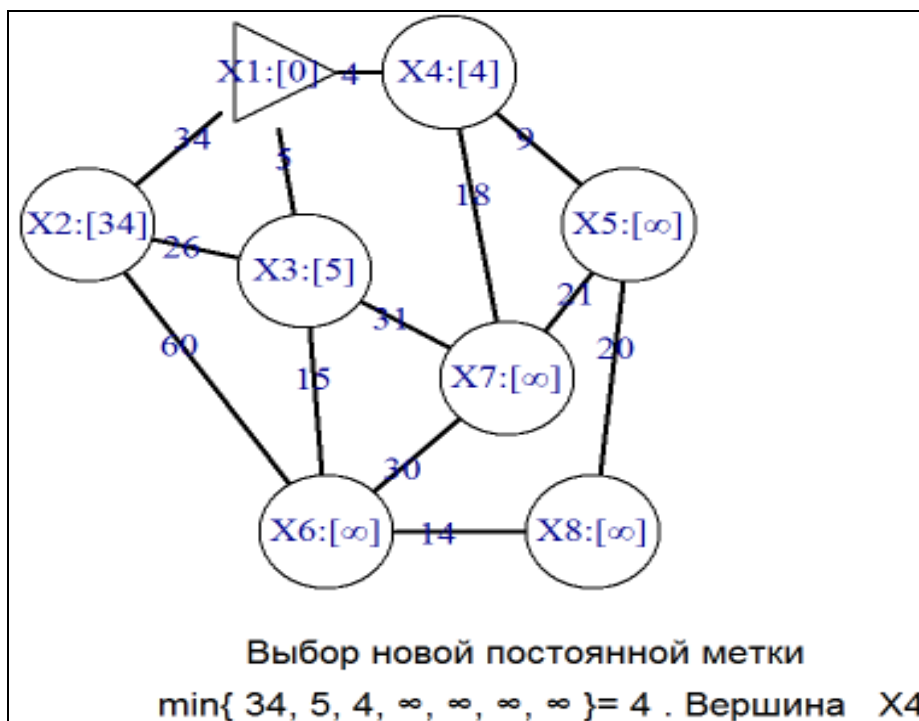


Рисунок П.1.3 - Шаг 1-2

## ПРИЛОЖЕНИЕ 2

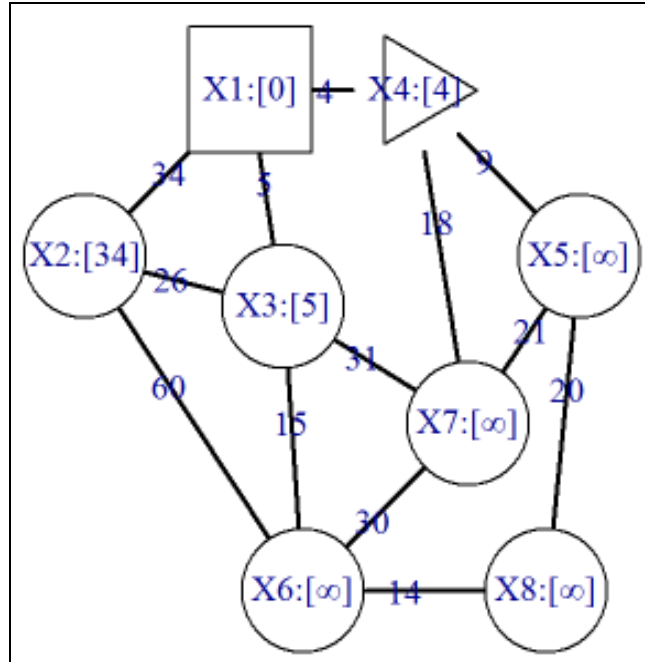


Рисунок П.2.1 - Шаг 2-0

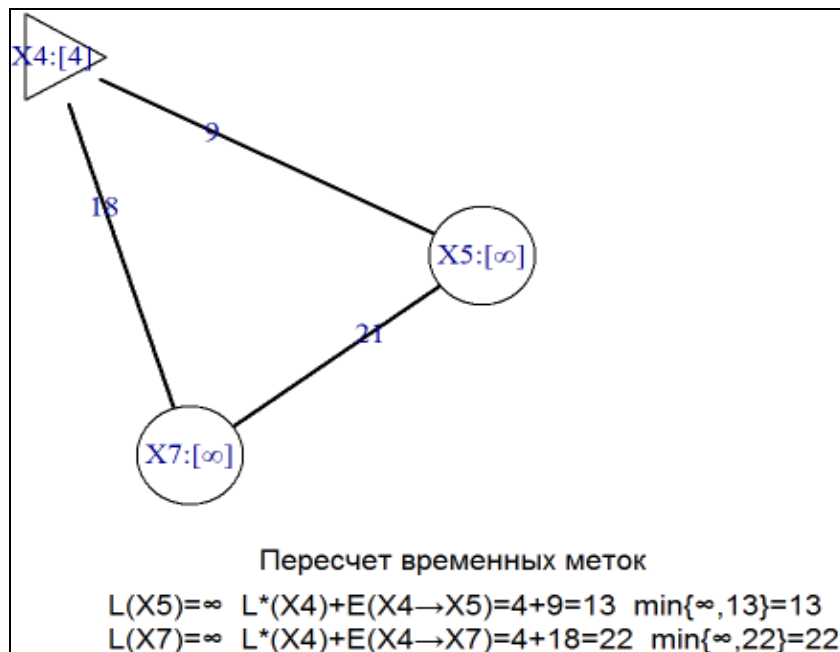


Рисунок П.2.2 - Шаг 2-1

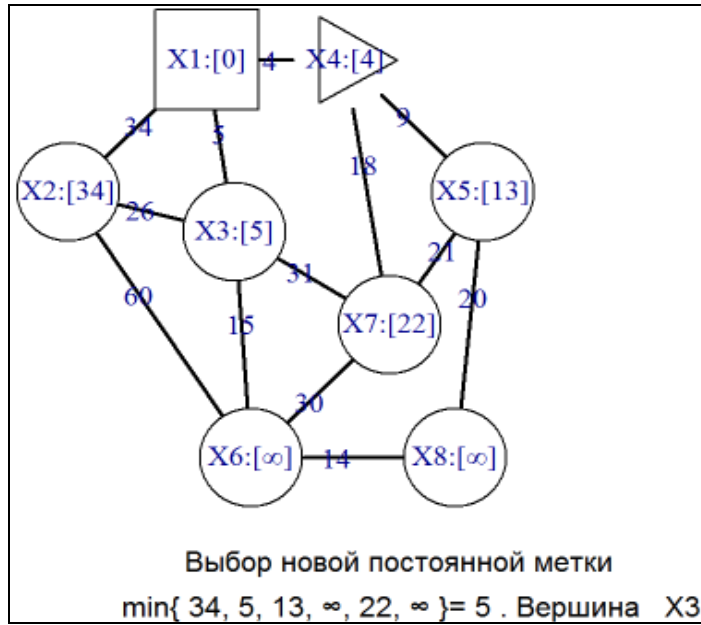


Рисунок П.2.3 - Шаг 2-2

### ПРИЛОЖЕНИЕ 3

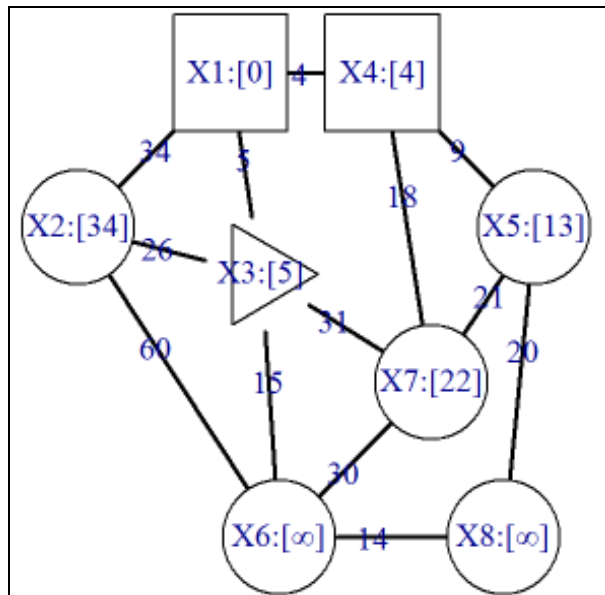


Рисунок П.3.1 - Шаг 3-0

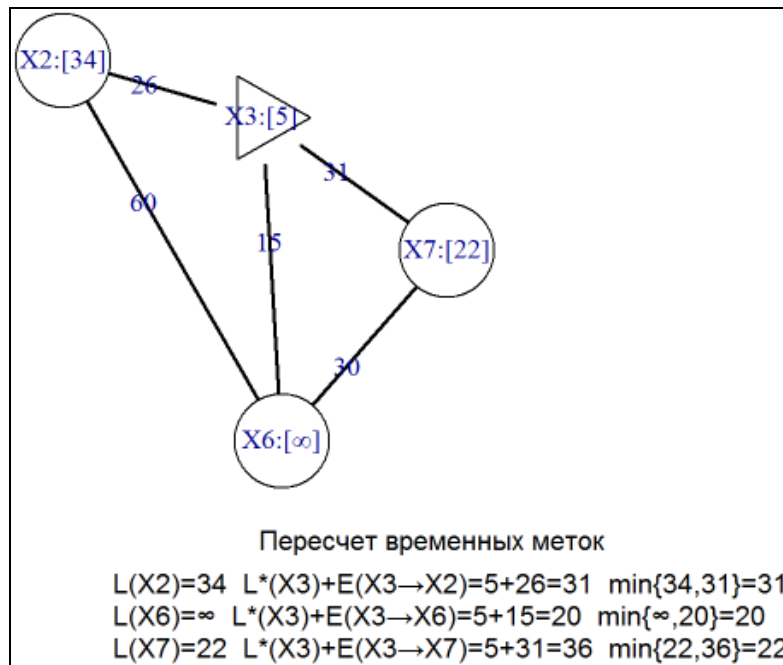


Рисунок П.3.2. - Шаг 3.1

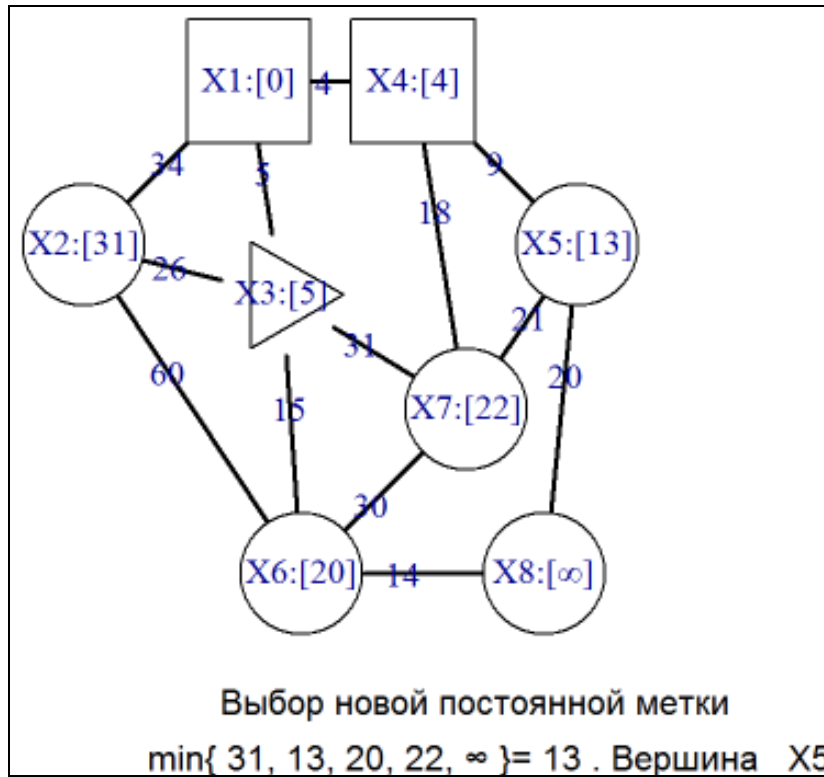


Рисунок П.3.3. - Шаг 3.2

## ПРИЛОЖЕНИЕ 4

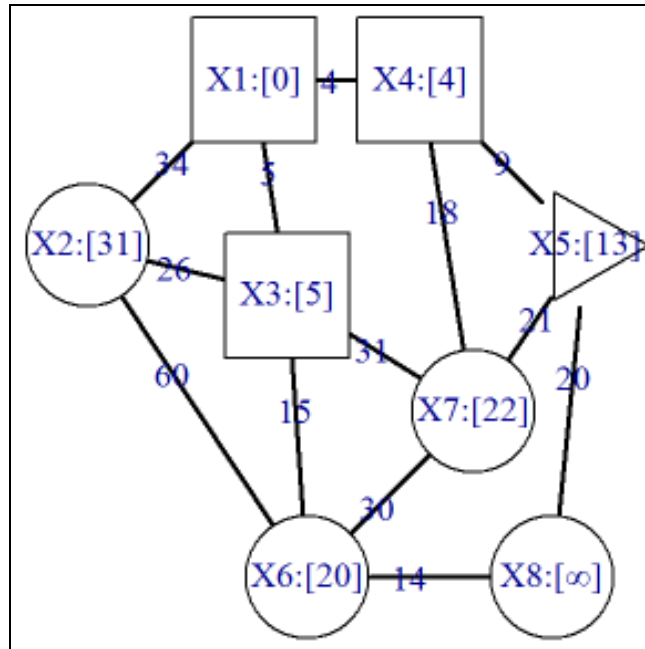


Рисунок П.4.1 - Шаг 4-0

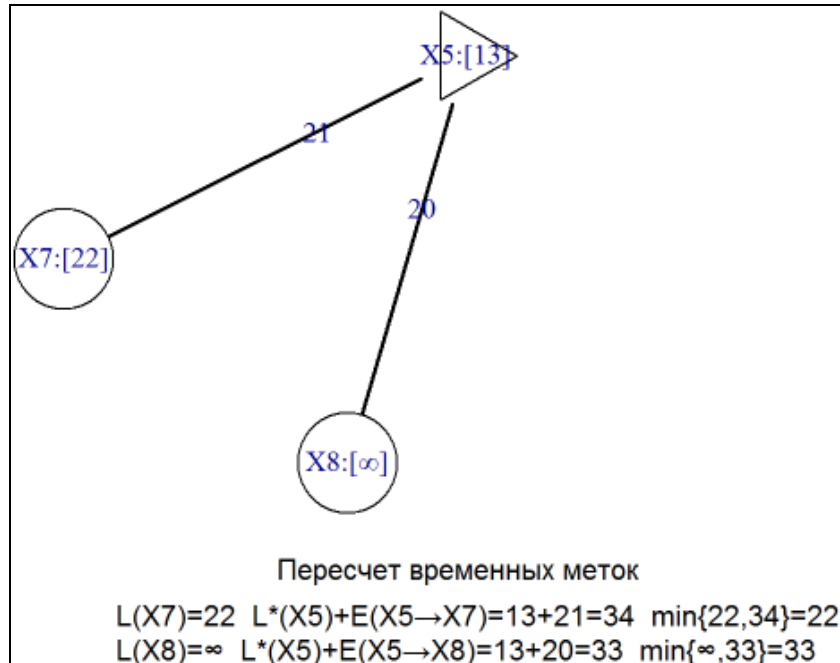


Рисунок П.4.2 - Шаг 4-1



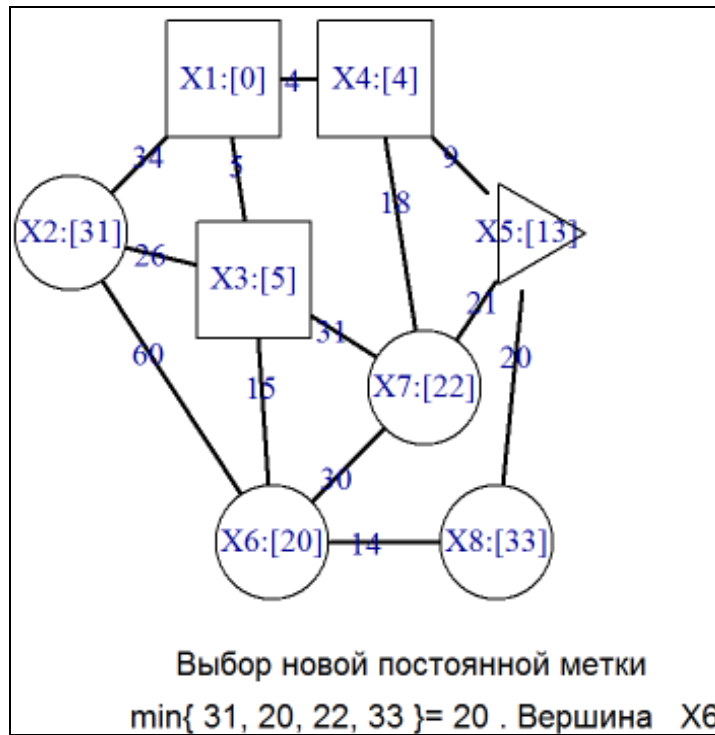


Рисунок П.4.3 - Шаг 4-2

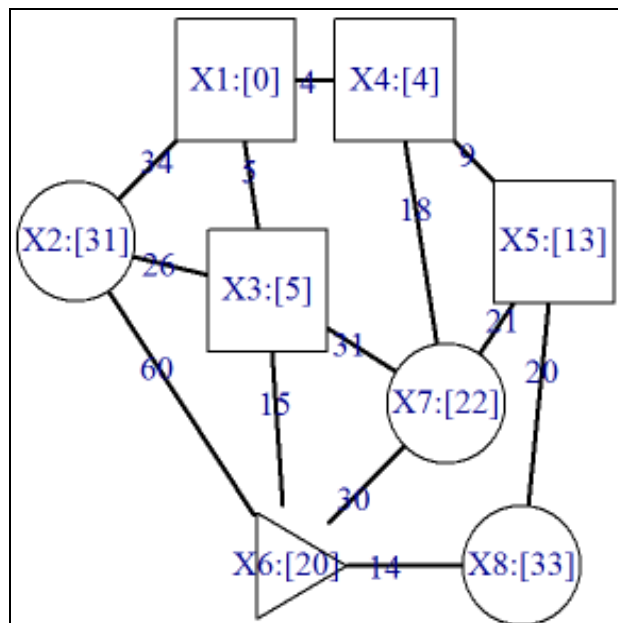
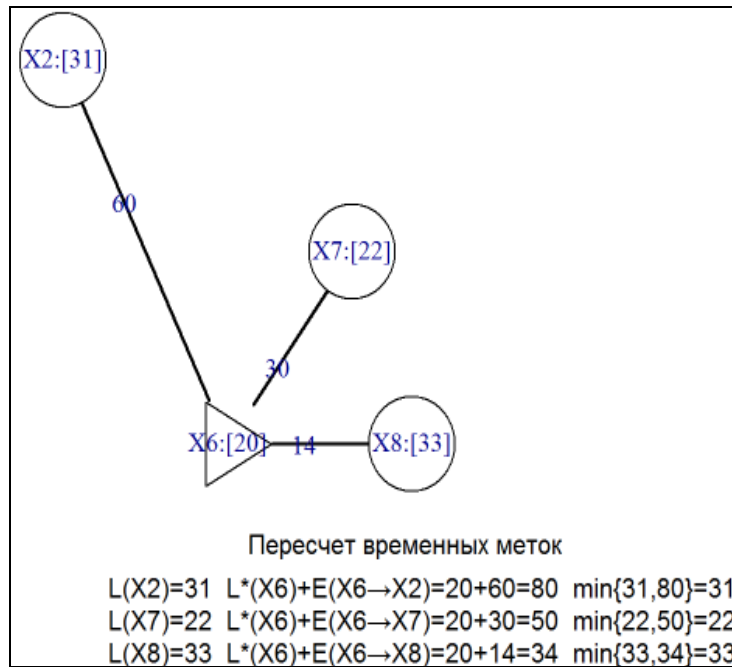
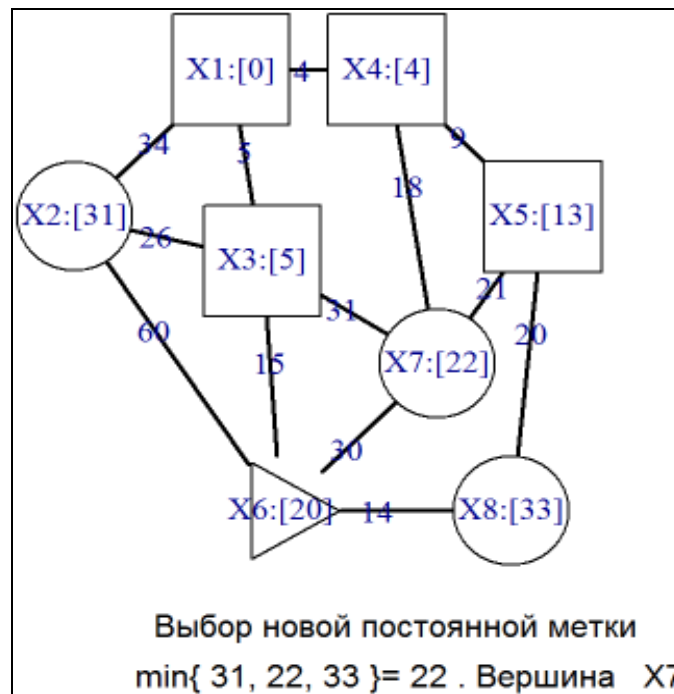


Рисунок П.4.4 - Пр. рез.4

## ПРИЛОЖЕНИЕ 5



**Рисунок П.5.1 - Шаг 5-1**



**Рисунок П.5.2 - Шаг 5-2**

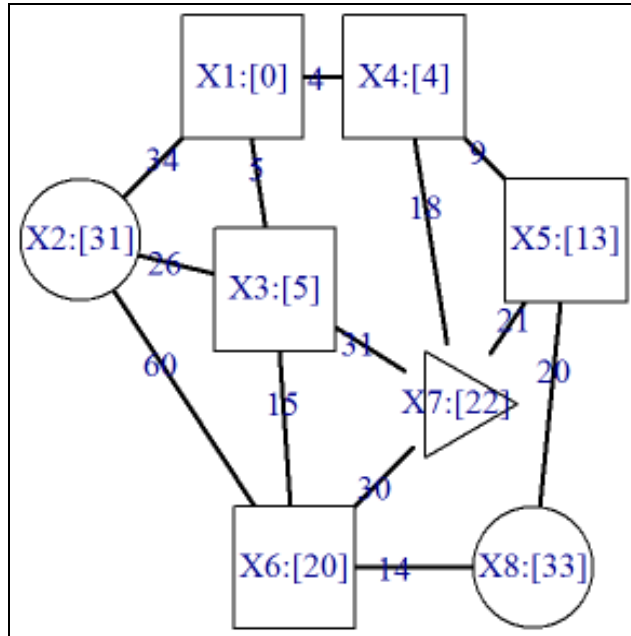


Рисунок П.5.3 - Пр. рез.5

## ПРИЛОЖЕНИЕ 6

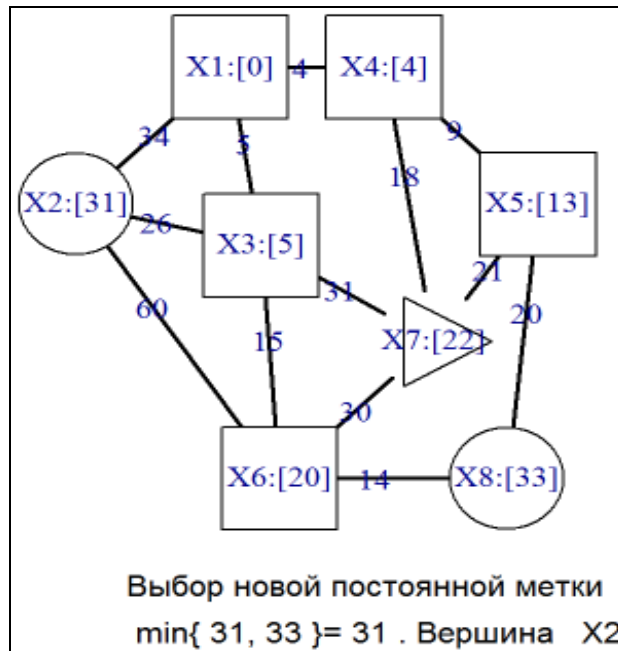


Рисунок П.6.1 - Шаг 6-2

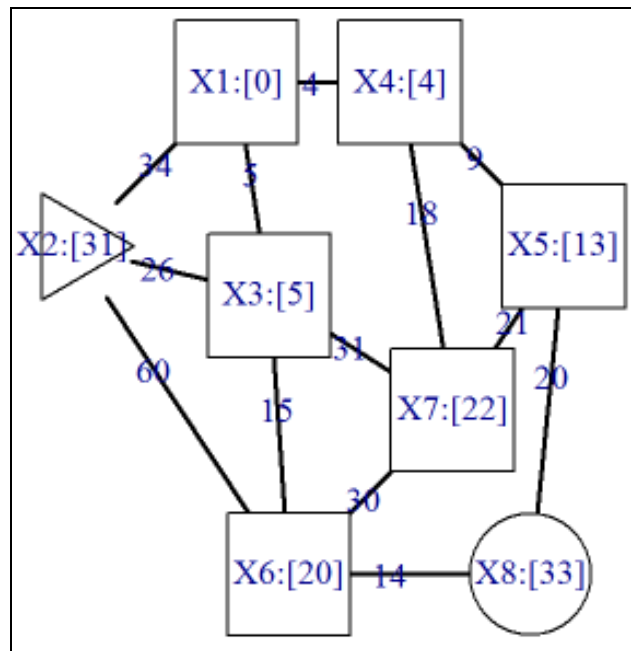


Рисунок П.6.2 - Пр. рез.6

## ПРИЛОЖЕНИЕ 7

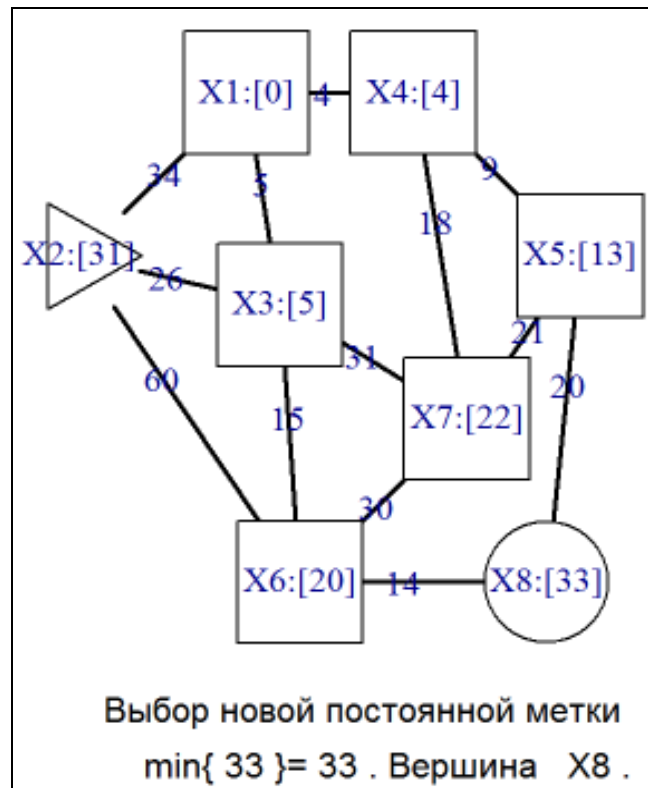


Рисунок П.7.1 - Шаг 7-2

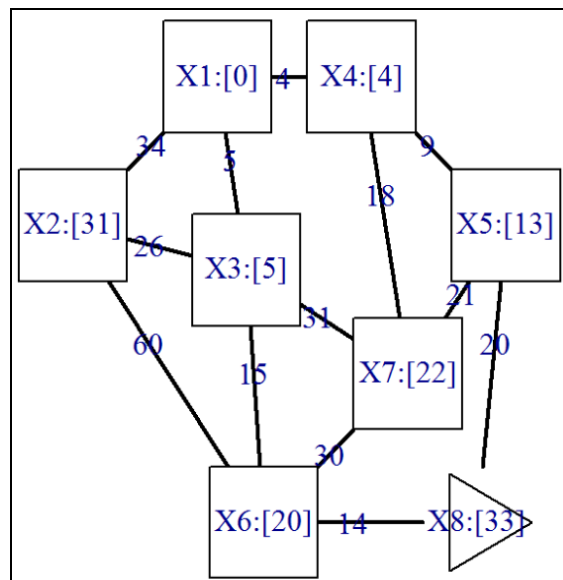


Рисунок П.7.2 – Результат применения метода Дейкстры.

Косовцева Татьяна Реональдовна  
Беляев Виктор Вениаминович

**Моделирование сетевых задач средствами  
системы R-project**

**Учебное пособие**

В авторской редакции

Редакционно-издательский отдел Университета ИТМО

Зав. РИО

Н.Ф. Гусарова

Подписано к печати

Заказ №

Тираж

Отпечатано на ризографе

**Редакционно-издательский отдел**  
**Университета ИТМО**  
197101, Санкт-Петербург, Кронверкский пр., 49, литер А