МАТЕМАТИКА

ЭЛЕМЕНТЫ ТЕОРИИ ГРАФОВ

Методические указания к практическим занятиям для студентов всех специальностей и направлений подготовки

САНКТ-ПЕТЕРБУРГ
2020

ДОКУМЕНТ ПОДПИСАН УСИЛЕННОЙ
КВАЛИФИЦИРОВАННОЙ
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 00F3 503F 985D 6537 76D4 6643 BD9B 6D2D 1C
Владелец: Пашкевич Наталья Владимировна
Действителен: с 27.12.2022 по 21.03.2024

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
Санкт-Петербургский горный университет

Кафедра высшей математики

МАТЕМАТИКА

ЭЛЕМЕНТЫ ТЕОРИИ ГРАФОВ

Методические указания к практическим занятиям
для студентов всех специальностей и направлений подготовки

САНКТ-ПЕТЕРБУРГ
2020

УДК 519.17 (073)

МАТЕМАТИКА. Элементы теории графов: Методические указания к практическим занятиям / Санкт-Петербургский горный университет. Сост.: *Л.В. Бакеева*, *М.Б. Шабаева*. СПб, 2020. 44 с.

Методические указания разработаны в соответствии с требованиями государственного образовательного стандарта высшего образования.

Содержат основные теоретические сведения теории графов, некоторые алгоритмы решения задач на графах. Изложение теоретического материала сопровождается разобранными типовыми примерами.

Методические указания могут быть использованы для работы на практических занятиях в соответствии с программами подготовки специалистов, бакалавров и магистров всех специальностей и направлений подготовки по дисциплинам «Математика», «Математика (специальные разделы)», «Прикладная математика».

Научный редактор проф. *А.П. Господариков*

Рецензент доц. *Л.В. Боброва* (Национальный открытый институт)

© Санкт-Петербургский
горный университет, 2020

МАТЕМАТИКА

ЭЛЕМЕНТЫ ТЕОРИИ ГРАФОВ

Методические указания к практическим занятиям для студентов всех специальностей и направлений подготовки

Сост.: *Л.В. Бакеева, М.Б. Шабаева*

Печатается с оригинал-макета, подготовленного кафедрой высшей математики

Ответственный за выпуск *Л.В. Бакеева*

Лицензия ИД № 06517 от 09.01.2002

Подписано к печати 15.06.2020. Формат 60×84/16.
Усл. печ. л. 2,6. Усл.кр.-отт. 2,6. Уч.-изд.л. 2,4. Тираж 150 экз. Заказ 360. С 34.

Санкт-Петербургский горный университет
РИЦ Санкт-Петербургского горного университета
Адрес университета и РИЦ: 199106 Санкт-Петербург, 21-я линия, 2

ВВЕДЕНИЕ

Во многих прикладных задачах системы связей между различными объектами изучаются с помощью графов. Это фигуры, состоящие из точек и их соединяющих линий. В этих задачах несущественно, соединены ли точки отрезками прямых или криволинейными линиями, какова длина линий и другие геометрические характеристики конфигурации линий. Важно лишь то, что каждая линия соединяет какие-либо две из заданных точек.

Примером графа может служить схема линий электропередач, схема дорог района или схема печатной платы.

Первая работа по теории графов была опубликована в 1736 году академиком Леонардом Эйлером в Трудах Академии наук. В работе решалась задача о Кенингсбергских мостах. Два острова соединялись с берегами реки и между собой семью мостами (рис. 1). Требовалось пройти каждый мост по одному разу и вернуться в

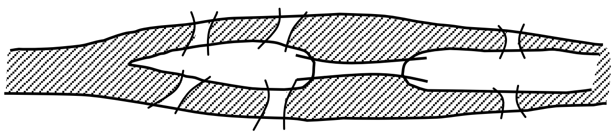


Рис. 1

исходную часть города. С помощью графа Эйлер показал, что задача решения не имеет.

Теория графов в последнее время широко используется в различных отраслях науки и техники, экономике и социологии. Графические изображения наглядно представляют различные взаимосвязи: пространственное расположение объектов, временные зависимости процессов и явлений, логические, структурные причинно-следственные и другие взаимосвязи. На языке теории графов формулируются и решаются многие задачи управления, в том числе задачи сетевого планирования, задачи принятия решений в условиях неопределенности и другие.

1. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Пусть V – непустое множество элементов $\{v_1, v_2, \ldots , v_n\}$ и U – множество, состоящее из неупорядоченных пар элементов множества V: $U=\{(v_i, v_j)\}$. *Графом* G (или *неориентированным графом*) называется совокупность двух множеств $G = (V,U)$. Элементы множества V называют вершинами, элементы U – ребрами графа. Геометрически вершины изображают точками, рёбра – линиями (рис. 2).

В теории графов изучают два основных вида графов: ориентированные и неориентированные. Ребро (v_i, v_j) называется *ориентированным*, или *дугой*, если v_i - начало, а v_j – его конец. Граф, все рёбра которого ориентированы, называется *ориентированным* (*орграфом*). Если граф ориентированный, то говорят, что дуга (v_i, v_j) *исходит* из вершины v_i и *заходит* в вершину v_j. При изображении орграфа дуги обозначают стрелками, указывающими их направление (рис. 3). Заметим, что любой неориентированный граф может быть представлен в виде орграфа путем замены каждого ребра двумя противоположно направленными дугами.

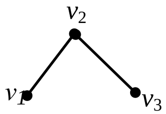


Рис. 2

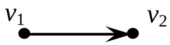


Рис. 3

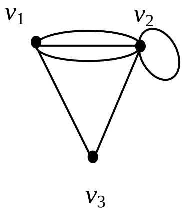


Рис. 4

Две вершины графа называются *смежными*, если существует соединяющее их ребро (дуга). Два ребра (дуги) называются *смежными*, если они имеют общую вершину. Если ребро (дуга) соединяет вершины v_i и v_j, то говорят, что ребро (дуга) (v_i, v_j) инцидентно вершинам v_i и v_j, а вершины v_i и v_j *инцидентны* ребру (дуге). Рёбра, инцидентные одной и той же паре вершин называются *параллельными*, или *кратными*. Ребро, концевые вершины которого совпадают, называется *петлей*.

Граф, содержащий петли и кратные рёбра (дуги), называется *псевдографом* (рис. 4). Граф с кратными рёбрами (дугами) и без петель называется *мультиграфом*.

Подграфом графа $G = (V,U)$ называется граф $G' = (V',U')$ такой, что V' является подмножеством множества V, а U' – подмножеством множества U. Если при этом V' совпадает с V, то G' называется *остовным подграфом* графа G.

Граф без петель и кратных рёбер называется *полным*, если для любых двух его вершин существует соединяющее их ребро (рис. 5).

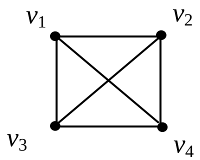


Рис. 5

Граф $\overline{G} = \left(V, \overline{U}\right)$ называется *дополнением графа* $G = \left(V, U\right)$, если множества вершин G и $\overline{G}$ совпадают, а множество $\overline{U}$ содержит только те рёбра, которые нужно добавить к графу G, чтобы получить полный граф (рис. 6).

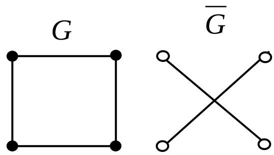


Рис. 6

Объединением графов $G_1 = (V_1, U_1)$ и $G_2 = (V_2, U_2)$ называется граф $G = G_1 \cup G_2 = \left(V_1 \cup V_2, U_1 \cup U_2\right)$.

Пересечением графов $G_1 = (V_1, U_1)$ и $G_2 = (V_2, U_2)$, где $V_1 \cap V_2 \neq \phi$, называется граф $G = G_1 \cap G_2 = \left(V_1 \cap V_2, U_1 \cap U_2\right)$.

Степенью вершины ν графа G называется число $\mu(\nu)$ рёбер, инцидентных этой вершине. Вершина графа, степень которой равна 0, называется *изолированной*.

В графе сумма степеней всех вершин равна удвоенному числу рёбер m графа (предполагается, что петля дает вклад 2 в степень вершины):

$$\sum_{v \in G} \mu\left(\nu\right) = 2m \,.$$

Для вершин орграфа определяются две степени:

$\mu_1(\nu)$ – число дуг с началом в вершине ν, или число дуг, исходящих из вершины ν;

$\mu_2(\nu)$ – число дуг, для которых эта вершина является концом, или число заходящих ребер.

Петля дает вклад 1 в обе эти степени. В орграфе суммы степеней всех вершин $\mu_1(\nu)$, $\mu_2(\nu)$ равны количеству рёбер m этого графа, а значит, равны между собой:

$$\sum_{v \in G} \mu_1(\nu) = \sum_{v \in G} \mu_2(\nu) = m .$$

Пример 1. Найти степени вершин графов G_1 и G_2, представленных на рис. 7.

Решение. Оба графа имеют по четыре вершины: $V = \{1,2,3,4\}$. Степени вершин неориентированного графа G_1: $\mu(1) = 3, \mu(2) = 4,$ $\mu(3) = 3, \quad \mu(4) = 4$,

$$\sum_{\nu \in G_1} \mu(\nu) = \mu(1) + \mu(2) + \mu(3) + \mu(4) = 3 + 4 + 3 + 4 = 14 .$$

Сумма степеней всех вершин вдвое больше числа рёбер графа:

$$14 = 2m , \; m = 7 \; \text{– число рёбер графа.}$$

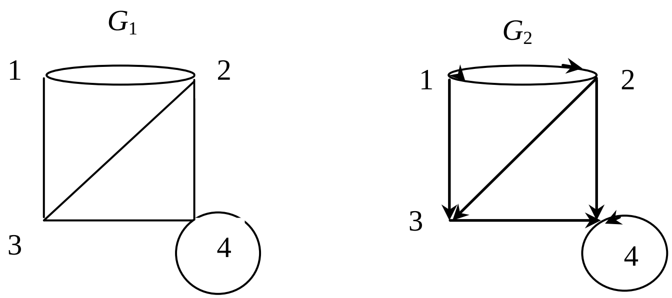


Рис. 7

Суммы степеней вершин первого и второго типа графа G_2 совпадают и равны числу рёбер графа:

$$\sum_{\nu \in G_2} \mu_1(\nu) = 2 + 3 + 1 + 1 = 7,$$

$$\sum_{\nu \in G_2} \mu_2(\nu) = 1 + 1 + 2 + 3 = 7,$$

$$\sum_{\nu \in G_2} \mu_1(\nu) = \sum_{\nu \in G_2} \mu_2(\nu) = 7 = m .$$

Граф называется *регулярным*, если степени всех его вершин равны.

Часто требуется уточнить, какие графы считаются различными, а какие не различаются. Это уточнение обычно связывается с понятием изоморфизма графов. Два графа G_1 и G_2 называются *изоморфными*, если существуют взаимно однозначное соответствие между их вершинами и взаимно однозначное соответствие между их рёбрами такие, что соответствующие рёбра соединяют соответствующие вершины. При этом записывают $G_1 \cong G_2$.

Пример 2. Показать, что графы G_1 и G_2, представленные на рис. 8, изоморфны.

Решение. Запишем множества вершин и рёбер

для графа G_1: $V_1 = \{1,2,3,4\}$ $U_1 = \{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}$;

для графа G_2: $V_2 = \{1,2,3,4\}$ $U_2 = \{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}$.

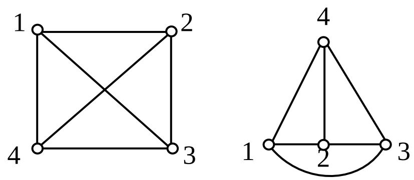


Рис. 8

Так как $V_1 = V_2$ и $U_1 = U_2$, то $G_1 \cong G_2$.

2. СПОСОБЫ ЗАДАНИЯ ГРАФОВ

Граф может быть задан одним из следующих способов.

1. Указанием множества вершин и списка концов всех рёбер, в котором для каждого ребра указывается пара вершин, являющихся концами этого ребра. Такое задание отвечает приведенному определению графа. Однако этот способ мало используется на практике.

2. Графическим способом, при котором вершины графа изображаются точками, а ребра – линиями, форма и длина которых не имеют значения.

3. Матрицей смежности. *Матрицей смежности* графа G, имеющего n вершин, называется квадратная матрица A порядка n, элементы которой, определяются по правилу:

$$a_{ij}=\begin{cases}1, \text{ если } (v_i, v_j)\in U;\\ 0, \text{ если } (v_i, v_j)\notin \mathrm{U}.\end{cases}$$

Матрица смежности является симметричной для любого неориентированного графа.

В случае псевдографа $a_{ij}=k$, где k – кратность дуги (ребра) (v_i, v_j) в этом псевдографе.

Число рёбер графа, заданного матрицей смежности, равно сумме элементов a_{ij}, расположенных на диагонали и выше, то есть равно $\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\geq i}}^{n} a_{ij}$. Число ребер орграфа определяется всеми элементами a_{ij} матрицы смежности и равно $\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}$.

4. Списком рёбер. Список ребер графа представляет собой два столбца: в левом столбце перечисляются все рёбра $u_k\in U$, а в правом – инцидентные им вершины. Для графа порядок вершин в строке произвольный, для орграфа первым стоит номер вершины - начала ребра.

Построение списка рёбер по матрице смежности. Элементу матрицы a_{ij} соответствует k строк с номерами i, j списка ребер. Элементу $a_{ij}=0$ – ни одной строки. Для графа строки сопоставляют только элементам над главной диагональю матрицы смежности, для орграфа нужно рассматривать все элементы a_{ij}.

5. Матрицей инцидентности. *Матрицей инцидентности* графа G, имеющего n вершин и m рёбер, называется матрица размерности $n \times m$, строки которой соответствуют вершинам, а столбцы – рёбрам, элементы которой, определяются по правилу:

$$b_{ij} = \begin{cases} 1, & \text{если вершина } v_i \text{ инцидентна ребру } \nu_j, \\ 0, & \text{в противном случае.} \end{cases}$$

В случае орграфа матрица задается так:

$$b_{ij} = \begin{cases} -1, & \text{если вершина } \nu_i - \text{начало дуги } u_j; \\ 1, & \text{если вершина } \nu_i - \text{конец ребра } u_j; \\ 0, & \text{в остальных случаях.} \end{cases}$$

Пример 3. Задать граф G, представленный на рис. 9, матрицей смежности, списком рёбер и матрицей инцидентности.

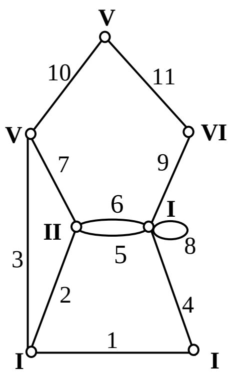


Рис. 9

Решение.

Матрица смежности имеет вид:

G	*I*	*II*	*III*	*IV*	*V*	*VI*	*VII*
I	0	1	1	0	1	0	0
II	1	0	0	1	0	0	0
III	1	0	0	2	1	0	0
IV	0	1	2	1	0	0	1
V	1	0	1	0	0	1	0
VI	0	0	0	0	1	0	1
VII	0	0	0	1	0	1	0

Список рёбер:

Ребра	Вершины		Ребра	Вершины	
1	I	II	6	III	IV
2	I	III	7	III	V
3	I	V	8	IV	IV
4	II	IV	9	IV	VII
5	III	IV	10	V	VI
			11	VI	VII

Матрица инцидентности имеет вид:

	1	2	3	4	5	6	7	8	9	10	11
I	1	1	1	0	0	0	0	0	0	0	0
II	1	0	0	1	0	0	0	0	0	0	0
III	0	1	0	0	1	1	1	0	0	0	0
IV	0	0	0	1	1	1	0	1	1	0	0
V	0	0	1	0	0	0	1	0	0	1	0
VI	0	0	0	0	0	0	0	0	0	1	1
VII	0	0	0	0	0	0	0	0	1	0	1

Список рёбер графа является, по существу, сокращенным представлением матрицы инцидентности (в каждой её строке только два элемента отличны от 0 или один, если ребро – петля).

Число строк в списке равно числу рёбер, а число вершин равно максимальному значению номера из перечисленных вершин.

3. МАРШРУТЫ, ЦЕПИ, ЦИКЛЫ НА ГРАФАХ

Пусть $G = (V,U)$ – неориентированный граф.

Маршрутом в графе G, соединяющим вершины v_1 и v_n, называется последовательность ребер $(v_1,v_2),(v_2,v_3),...,(v_{n-1},v_n)$, в которой конец каждого предыдущего ребра совпадает с началом следующего. Маршрут также можно задать перечислением входящих в него вершин: $l(v_1, v_2, \dots v_n)$.

Аналогом маршрута для орграфа является *ориентированный маршрут* из v_1 в v_n, представляющий собой последовательность дуг, в которой конец предыдущей дуги совпадает с началом следующей.

Число рёбер (дуг) в маршруте называется *длиной* маршрута.

Маршрут называется *замкнутым*, если его начальная вершина совпадает с конечной, и *незамкнутым* – в противном случае.

В маршруте одно и то же ребро (дуга) может встречаться несколько раз. Незамкнутый маршрут, в котором все ребра (дуги) различны, называется *цепью* (*путем*). Цепь (путь), в которой все вершины, кроме, быть может, первой и последней, различны, называется *простой цепью* (*простым путем*). Простая цепь (путь), начальная и конечная вершины которой совпадают, называется *циклом* (*контуром*).

Вершина $v_n \in G$ называется *достижимой* из вершины $v_1 \in G$, если существует цепь (путь) $(v_1, \dots, v_n)$ с началом в вершине v_1 и концом в вершине v_n.

Граф (орграф) G называется *связным*, если для любых двух его вершин v_i и v_j вершина v_j достижима из вершины v_i и вершина v_i достижима из вершины v_j. Таким образом, неориентированный граф будет связным, если любые две его вершины соединены цепью.

Деревом называется связный граф, не содержащий циклов.

Расстоянием $\rho(v_i,v_j)$ между вершинами v_i и v_j графа G называется минимальная длина простой цепи с началом в вершине v_i и концом в вершине v_j. Матрица с элементами $\rho_{ij} = \rho(v_i, v_j)$ называется *матрицей расстояний*.

Эксцентриситетом вершины v_i называется величина

$$\varepsilon(v_i) = \max\{\rho(v_i, v_j) | v_j \in G\},$$

т.е. эксцентриситет вершины равен расстоянию от данной вершины до наиболее удаленной от нее.

Максимальный среди всех эксцентриситетов вершин графа называется *диаметром графа* и обозначается $d(G)$:

$$d(G) = \max\{\varepsilon(v) | v \in G\}.$$

Вершина v называется *периферийной*, если $\varepsilon(v) = d(G)$.

Минимальный из эксцентриситетов графа G называется *радиусом графа* и обозначается $r(G)$:

$$r(G) = \min\{\varepsilon(v) | v \in G\}.$$

Вершина v называется *центральной*, если $\varepsilon(v) = r(G)$. Множество всех центральных вершин называется *центром графа*.

Пример 4. Построить матрицу расстояний для графа, изображенного на рис. 9. Определить диаметр, радиус и центр графа.

Решение. Матрица расстояний имеет вид

	I	*II*	*III*	*IV*	*V*	*VI*	*VII*	$\varepsilon(v)$
I	0	1	1	2	1	2	3	3
II	1	0	2	1	2	3	2	2
III	1	2	0	1	1	2	2	2
IV	2	1	1	1	2	2	1	2
V	1	2	1	2	0	1	2	2
VI	2	3	2	2	1	0	1	3
VII	3	2	2	1	2	1	0	2

Сравнивая значения эксцентриситетов вершин, получим, что $d(G) = 3$, $r(G) = 2$; центром графа является множество вершин

$\{\mathrm{II}, \mathrm{III}, \mathrm{IV}, \mathrm{V}, \mathrm{VII}\} \subset V(G)$, а $\{\mathrm{I}, \mathrm{VII},\} \subset V(G)$ – множество периферийных вершин.

Граф, каждому ребру которого поставлено в соответствие некоторое число, называемое весом, называется *взвешенным*. В качестве веса могут выступать самые различные физические величины. Например, если графом представлена схема дорог, то весом может быть расстояние между населенными пунктами или стоимость проезда между ними; если граф представляет связи складов с магазинами, то весом может быть объем поставок в магазины или стоимость доставки единицы товара от данного склада до магазина.

Взвешенные графы могут быть описаны матрицей весов. Матрицей весов графа, имеющего *n* вершин, называется квадратная матрица порядка *n*, каждый элемент которой определяется по правилу:

$$a_{ij} = \begin{cases} l_{ij} \text{ - вес ребра, соединяющего вершины } v_i \text{ и } v_j; \\ \infty, \text{ если не существует ребра, соединяющего вершины } v_i \text{ и } v_j. \end{cases}$$

Особый практический интерес представляют так называемые эйлеров и гамильтонов циклы.

Цикл (контур) в графе *G* называется *эйлеровым*, если он проходит по одному разу через каждое ребро этого графа. Связный граф называется *эйлеровым*, если он содержит эйлеров цикл. Если в графе сущетвует эйлеров цикл, то его можно нарисовать, не отрывая карандаша от бумаги и не повторяя линий.

Теорема Эйлера. Связный граф является эйлеровым тогда и только тогда, когда степень каждой его вершины четная.

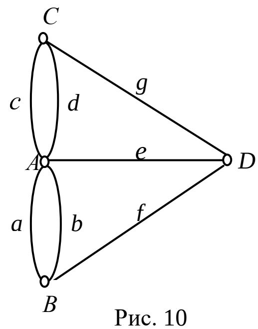


Рис. 10

Данная теорема служит критерием существования эйлерова цикла в графе. Эта теорема была сформулирована в процессе решения Леонардом Эйлером задачи о кенигсбергских мостах. На графе к задаче о кенигсбергских мостах нельзя построить эйлеров цикл, так как этот граф имеет вершины нечетной степени (см. рис. 10).

Теорема. Граф имеет эйлеров путь тогда и только тогда, когда он связный и ровно две его вершины имеют нечетную степень.

Согласно этой теореме задача о кенигсбергских мостах также не имеет эйлерова пути.

Теорема. Ориентированный граф имеет эйлеров цикл тогда и только тогда, когда он связный и степень захода каждой вершины равна ее степени исхода.

Цикл (контур) в графе G называется *гамильтоновым*, если он проходит через каждую вершину графа ровно один раз. Граф G называется *гамильтоновым*, если он содержит гамильтонов цикл.

Одной из практических задач, связанных с построением гамильтонова цикла, является *задача коммивояжера (развозчика продукции или бродячего торговца):* коммивояжер должен посетить один и только один раз каждый из n городов и вернуться в исходный пункт. Его маршрут должен минимизировать суммарную длину пройденного пути.

4. АЛГОРИТМЫ НА ГРАФАХ

Рассмотрим задачи, в которых производится поиск такой конфигурации рёбер и вершин заданного графа, которая обладает некоторым оптимальным свойством.

4.1. Задача о кратчайшем пути. Метод Дейкстры

Многие прикладные задачи на языке теории графов могут быть сформулированы следующим образом.

Пусть задан связный взвешенный граф $G(X, U)$, не содержащий кратных рёбер, в котором зафиксирована вершина, считающаяся исходной (x_1). Требуется определить кратчайший путь от вершины x_1 до каждой из оставшихся вершин.

Метод Дейкстры поиска кратчайшего пути между двумя вершинами состоит в том, что вершинам графа присваиваются временные метки, которые затем, по определенному правилу, заменяются постоянными. Постоянная метка вершины и есть кратчайшее расстояние от начальной вершины до рассматриваемой. Заметим, что этот метод применим как в случае неориентированного, так и в случае ориентированного графа. Опишем алгоритм метода.

Вначале вершине, которая является началом пути, присваивается постоянная метка $L^*(x_1) = 0$ (знак * у метки означает, что данная метка постоянная), а каждой из остальных вершин присваивается временная метка $L(x_k) = \infty$, $k = 2, \dots n$.

Следующая постоянная метка появляется в результате выполнения двух шагов, составляющих итерацию.

1. Для вершины x_i, которой на предыдущей итерации была присвоена постоянная метка $L^*(x_i)$, выделяют множество $\Gamma(x_i)$ вершин-последователей, имеющих с x_i общие ребра: $\Gamma(x_\mathrm{i}) = \{x_{\mathrm{j}1},\ x_{\mathrm{j}2}, \dots\}$. Для каждой из этих вершин с временной меткой вычисляют новую метку по формуле

$$L^н\left(x_j\right) = \min\left\{L^c\left(x_j\right), L^*\left(x_i\right) + R_{ij}\right\}, \qquad (4.1)$$

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

2. Из *всех* имеющихся временных меток выбирается наименьшая, которая становится постоянной для своей вершины.

Решение заканчивается после того, как вершине, до которой требуется найти кратчайший путь, будет присвоена постоянная метка. По постоянным меткам восстанавливается кратчайший путь.

Для построения остовного дерева кратчайших путей нужно найти кратчайшее расстояние от заданной вершины до всех остальных вершин, а значит, процесс присваивания постоянных меток продолжается до тех пор, пока не определятся постоянные метки для всех вершин.

Пример 5. Для графа, изображенного на рис. 11, найти длины кратчайших путей от вершины x_1 до остальных вершин; построить остовное дерево кратчайших путей.

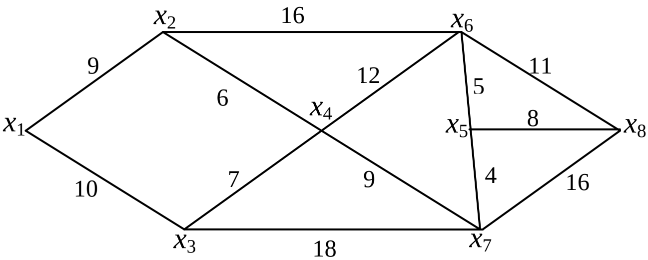


Рис. 11

Решение. Найдем кратчайшие пути от вершины x_1 до остальных вершин графа.

Нулевая итерация. $L^*(x_1) = 0$ (знак * у метки означает, что данная метка постоянная); $L(x_i) = \infty$, $i = 2, 3, 4, 5, 6,7,8$. Решение задачи удобно представить в виде таблицы, в которую будем заносить результаты каждой итерации. Во второй строке таблицы будем указывать вершину, получившую постоянную метку на предыдущей итерации.

Первая итерация.

1. Запишем множество вершин, связанных с вершиной x_1 рёбрами, то есть множество последователей вершины x_1: $\Gamma(x_1) = \{x_2, x_3\}$. Поскольку вершины из $\Gamma(x_1)$ имеют временные метки, вычислим новые временные метки этих вершин по формуле (4.1):

$$L^н(x_2) = \min\{ L^c(x_2) ; L^*(x_1) + R_{12}\}=\min\{\propto;\ 0+9\}= 9,$$

$$L^н(x_3) = \min\{ L^c(x_3) ; L^*(x_1) + R_{13}\}=\min\{\propto;\ 0 + 10\}= 10.$$

Найденные новые метки заносим в столбец таблицы, соответствующий первой итерации.

Итерации	0	1	2	3	4	5	6	7
x_i		x_1						
x_1	0^*							
x_2	∞	9^*						
x_3	∞	10						
x_4	∞							
x_5	∞							

Итерации	0	1	2	3	4	5	6	7
x_6	∞							
x_7	∞							
x_8	∞							

2. Наименьшая из всех временных меток

$$\min_{2\le j\le 8} L(x_j) = \min\{9,10,\infty,\infty,\infty,\infty,\infty\} = 9 .$$

Следовательно, вершина x_2 получает постоянную метку $L^*(x_2) = 9$.

Вторая итерация.

1. $\Gamma(x_2)=\{x_1, x_4, x_6\}$. Вершины x_4, x_6 имеют временные метки. Обновляем метки.

$$L^н(x_4) = \min\{ L^c(x_4) ; L^*(x_2) + R_{24}\}=\min\{\propto;\ 9 + 6\}= 9,$$

$$L^н(x_6) = \min\{ L^c(x_6) ; L^*(x_2) + R_{26}\}=\min\{\propto;\ 9 + 16\}= 25.$$

2. Наименьшая из всех временных меток:

$$\min\{10, 15, 25, \propto \} = 10.$$

Вершина x_3 получает постоянную метку $L^*(x_3) = 10$.

Третья итерация.

1. $\Gamma(x_3) = \{x_1,\ x_4,\ x_7\}$. Вершины x_4 и x_7 имеют временные метки. Обновляем метки:

$$L^н(x_4) = \min\{ L^c(x_4) ; L^*(x_3) + R_{43}\}=\min\{15;\ 10 + 7\}= 15,$$

$$L^н(x_7) = \min\{ L^c(x_7) ; L^*(x_3) + R_{37}\}=\min\{\propto;\ 10 + 18\}= 28.$$

2. Наименьшая из всех временных меток:

$$\min\{15, 25, 28 \}=15 \quad \Rightarrow L^*(x_4) = 15.$$

Четвертая итерация.

1. $\Gamma(x_4) = \{x_2, x_3, x_6, x_7\}$. Новые временные метки:

$$L^н(x_6) = \min\{ L^c(x_6) ; L^*(x_4) + R_{46}\}=\min\{25;\ 15 + 12\}= 25,$$

$$L^н(x_7) = \min\{ L^c(x_7) ; L^*(x_4) + R_{47}\}=\min\{28;\ 15 + 9\}= 24.$$

2. Наименьшая из всех временных меток:

$$\min\{24, 25 \} = 24 \quad \Rightarrow L^*(x_7) = 24.$$

Пятая итерация.

1. $\Gamma(x_7) = \{x_3, x_4, x_5, x_8\}$. Новые временные метки:

$$L^{н}(x_5) = \min\{ L^{c}(x_5)\,;\, L^*(x_7) + R_{57}\}=\min\{\propto;\ 24 + 4\}= 28,$$

$$L^{н}(x_8) = \min\{ L^{c}(x_8)\,;\, L^*(x_7) + R_{78}\}=\min\{\propto;\ 24 + 16\}= 40.$$

2. Наименьшая из всех временных меток:

$$\min\{25, 28, 40\} = 25 \quad \Rightarrow L^*(x_6)=25.$$

Шестая итерация.

1. $\Gamma(x_6) = \{x_2, x_4, x_5, x_8\}$. Новые временные метки:

$$L^{н}(x_5) = \min\{ L^{c}(x_5)\,;\, L^*(x_6) + R_{65}\}=\min\{28;\ 25 + 5\}= 28,$$

$$L^{н}(x_8) = \min\{ L^{c}(x_8)\,;\, L^*(x_6) + R_{68}\}=\min\{40;\ 25 + 11\}= 36.$$

2. Наименьшая из всех временных меток:

$$\min\{28, 36\} = 28 \quad \Rightarrow L^*(x_5) = 28.$$

Седьмая итерация.

1. Единственная вершина с временной меткой – это x_8.

$$L(x_8) = \min\{36, 28+8\} = 36.$$

2. Наименьшая из всех временных меток:

$$\min\{36\} = 36.$$

Вершина x_8 получает постоянную метку $L^*(x_8) = 36$.

В таблице заполняем последний столбец, соответствующий седьмой итерации. Вычисление постоянных меток закончено, а значит, кратчайшие расстояния от вершины x_1 до каждой из остальных найдены.

итерации	0	1	2	3	4	5	6	7
x_i		x_1	x_2	x_3	x_4	x_7	x_6	x_5
x_1	0^*							
x_2	∞	9^*						
x_3	∞	10	10^*					
x_4	∞		15	15^*				
x_5	∞					28	28^*	
x_6	∞		25			25^*		
x_7	∞			28	24^*			
x_8	∞					40	36	36^*

Построение дерева кратчайших путей. Построить дерево кратчайших путей означает, что надо найти связный подграф, в котором длина пути от вершины x_1 до каждой из вершин будет наименьшей по сравнению с любым другим путем.

На первой итерации постоянную метку получила вершина x_2, как последователь вершины x_1. Это означает, что дерево кратчайших путей содержит ребро (x_1, x_2).

На второй итерации постоянную метку получила вершина x_2. Эта метка была найдена на первой итерации для последователя вершины x_1. А значит, в дерево кратчайших путей включаем ребро (x_1, x_3).

На третьей итерации постоянную метку получила вершина x_4. Эта метка была найдена на второй итерации для последователя вершины x_2. А значит, дерево кратчайших путей содержит ребро (x_2, x_4). Определив аналогичным образом оставшиеся рёбра, получим дерево кратчайших путей, изображенное на рис. 12.

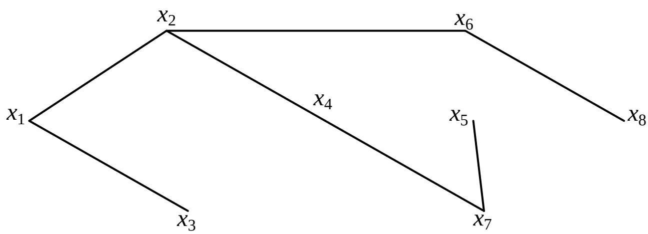


Рис. 12

Убедимся, что длина кратчайшего пути, например, из x_1 в x_8 $l(x_1, x_2, x_6, x_8)$ действительно равна постоянной метке вершины x_8:

$l(x_1, x_2) + l(x_2, x_6) + l(x_6, x_8) = 9+16+11=36.$

4.2. Задача о минимальном остовном дереве

На практике часто решается задача о построении дорог (газопроводов, линий электропередач) с минимальной стоимостью. Причем под стоимостью понимают не обязательно финансовые затраты, это может быть необходимое количество материала, временные затраты и пр. На языке теории графов задача формулируется следующим образом: для связного взвешенного графа $G(X, U)$ с n вершинами требуется построить остовное дерево с минимальным суммарным весом ребер; такое дерево называется минимальным остовным деревом.

Рассмотрим алгоритм «ближайшего соседа».

Построение минимального остовного дерева начинается с произвольной вершины, например, x_1. Среди всех ребер, инцидентных этой вершине, выбираем ребро с наименьшим весом и включаем в искомое дерево. Другую вершину выбранного ребра обозначим x_2.

Затем просматриваем все ребра, отличные от (x_1, x_2), инцидентные вершинам x_1 и x_2, выбираем из них ребро с наименьшим весом и включаем его в будущее дерево. Другую вершину, инцидентную этому ребру, обозначим через x_3.

На каждом из следующих шагов добавляем ребро с наименьшим весом, не образующее циклов с уже выбранными рёбрами.

Процесс продолжается до тех пор, пока все вершины графа G не будут включены в дерево, т. е. пока дерево не станет остовным.

Оставим без доказательства два следующих утверждения:

1. В процессе построения никогда не появятся циклы;

2. Описанная процедура действительно порождает минимальное остовное дерево.

Пример 6. Найти минимальное остовное дерево для взвешенного графа, изображенного на рис. 13.

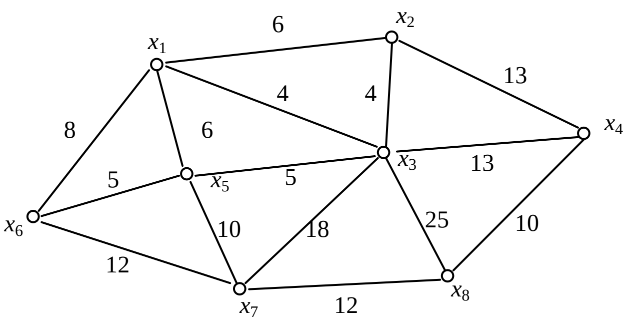


Рис. 13

Решение. Пусть исходной вершиной будет вершина x_1, присвоим ей метку 1. Инцидентными этой вершине являются ребра (x_1, x_2), (x_1, x_3), (x_1, x_5), (x_1, x_6). Наименьшую длину имеет ребро (x_1, x_3). Включаем это ребро в искомое дерево. Вершина x_3 получает метку 2, который указываем в скобках (см. рис. 14).

Инцидентными вершинам x_1 и x_3 являются ребра (x_1, x_2), (x_1, x_5), (x_1, x_6). (x_3, x_2), (x_3, x_4), (x_3, x_5), (x_3, x_7), (x_3, x_8). Ребро (x_1, x_3), включенное в дерево, естественно, не рассматривается. Наименьший вес имеет ребро (x_3, x_2), которое включаем в минимальное дерево. Вершина x_2 получает метку 3.

Рассмотрим ребра, инцидентные вершинам x_1 и x_2. Это ребра (x_1, x_5), (x_1, x_6), (x_2, x_4). Поскольку наименьший вес имеет ребро (x_1, x_5), включаем это ребро в минимальное дерево. Вершина x_5 получает метку 4.

Ребра, инцидентные вершинам x_2 и x_5: (x_2, x_4), (x_5, x_6), (x_5, x_7). Наименьший вес имеет ребро (x_5, x_6). Вершина x_6 получает метку 5. Мы не рассматриваем рёбра, которые создают циклы.

Проделанные и дальнейшие действия отражены на рис. 14.

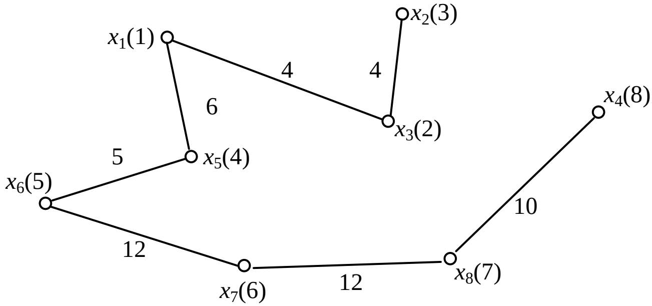


Рис. 14

Таким образом, минимальная суммарная длина построенного минимального дерева, обеспечивающего достижение всех вершин графа, равна 4+4+6+5+12+12+10 = 53.

5. СЕТИ

Сетью называется связный взвешенный орграф $G(V, U)$ без петель, некоторые вершины которого выделены. Выделенные вершины называются *полюсами* сети.

Двухполюсной сетью называется связный взвешенный орграф $S(V, U)$ без петель, для которого выполняются условия:

1) существует одна и только одна вершина v_0, называемая *истоком* или *входом* сети, такая, что ни одна дуга не заходит в v_0;

2) существует одна и только одна вершина s, называемая *стоком* или *выходом*, такая, что из s не исходит ни одной дуги;

3) каждой дуге $u \in U$ поставлено в соответствие число $c(u) \geq 0$, называемое пропускной способностью дуги.

Для наглядности можно представить, что по дугам из истока в сток направляется ресурс, груз, информация, вещество, тогда пропускная способность дуги характеризует максимальное количество вещества, которое может пропустить за единицу времени дуга $u \in U$.

Функция $\varphi(u)$, заданная на множестве дуг U, называется *потоком* (или *допустимым потоком*) в сети $S(V, U)$, если:

1) для любой дуги $u \in U$ величина $\varphi(u)$, называемая *потоком по дуге* u, удовлетворяет условию $0 \leq \varphi(u) \leq c(u)$;

2) для вершины v, $v \neq v_0, v \neq s$, сумма потоков по дугам, заходящим в вершину, равна сумме потоков по дугам, исходящим из этой вершины (условие сохранения потока в вершинах сети):

$$\sum_{(v_i, v) \in U} \varphi(v_i, v) = \sum_{(v, v_j) \in U} \varphi(v, v_j).$$

Условие сохранения потока в сети: сумма потоков по дугам, исходящих из истока, равна сумме потоков по дугам, входящих в сток:

$$\sum_{(v_0, v_i) \in U} \varphi(v_0, v_i) = \sum_{(v_j, s) \in U} \varphi(v_i, s).$$

Величина $\varphi_s = \sum_{(v_0, v_i) \in U} \varphi(v_0, v_i) = \sum_{(v_j, s) \in U} \varphi(v_i, s)$ называется *вели-*

чиной потока в сети.

Дуга $u \in U$ называется *насыщенной,* если поток по ней равен ее пропускной способности, т. е. $\varphi(u) = c(u)$. Поток φ_s называется *полным*, если любой путь в сети из v_0 в s содержит, по крайней мере, одну насыщенную дугу. Поток φ_s называется *максимальным*, если его величина принимает максимальное значение по сравнению с другими допустимыми потоками сети. Очевидно, что максимальный поток обязательно является полным.

Задача о максимальном потоке: на сети сформировать максимальный по величине поток $\varphi_{\max}$ между ее истоком и стоком. Иначе говоря, нужно найти совокупность потоков $\{\varphi^*(v_i, v_j)\}$ по всем дугам, которые максимизируют функцию φ_s.

Предположим, что дана некоторая сеть. Разобьем множество вершин V этой сети на два непересекающихся подмножества A и B ($A \cup B = V; A \cap B = \varnothing$) так, чтобы исток v_0 попал в подмножество A, а сток s – в подмножество B, т.е. $v_0 \in A,\ s \in B$. *Разрезом сети* $S(V,\ U)$ относительно множества вершин A, называется множество дуг $R(A/B) = \{u = (v_i, v_j) \in U\}$ таких, что $v_i \in A,\ v_j \in B$.

Иными словами, разрезом сети является множество дуг сети, обладающих следующим свойством: любой путь от истока к стоку сети пройдет хотя бы по одной дуге разреза. При удалении разреза исток сети отделяется от стока, т.е. сеть как бы разрезается, орграф становится несвязным.

Пусть $R(A/B)$ - разрез на сети, представляющий совокупность дуг, которые связывают подмножества вершин A и B. В разрез входят дуги, обозначим их R^+, начальные вершины которых принадлежат подмножеству A, а конечные – подмножеству B, т.е. $R^+ = \{(v_i, v_j) \mid v_i \in A, v_j \in B\}$. А также в разрез входят дуги, обозначим их R^-, начальные вершины которых принадлежат подмножеству *B*, а конечные – подмножеству *A*, т.е

$$R^- = \{(v_i, v_j) \mid v_i \in B, v_j \in A\}.$$

Пропускной способностью разреза $R(A/B)$ называется величина $C(A/B)$, которая определяется следующей формулой:

$$C(A/B) = C(R^+) - C(R^-) = \sum_{v_i \in A, v_j \in B} c(v_i, v_j) - \sum_{v_i \in B, v_j \in A} c(v_i, v_j).$$

Минимальным разрезом сети называется разрез с минимальной пропускной способностью.

Потоком через разрез $R(A/B)$ называется величина $\Phi(A/B)$, которая определяется следующей формулой:

$$\Phi(A/B) = \Phi(R^+) - \Phi(R^-) = \sum_{v_i \in A, v_j \in B} \varphi(v_i, v_j) - \sum_{v_i \in B, v_j \in A} \varphi(v_i, v_j).$$

Теорема Форда-Фалкерсона. Величина каждого потока от входа к выходу сети не превосходит пропускной способности минимального разреза, разделяющего исход и сток, причем существует максимальный поток, величина которого равна пропускной способности минимального разреза.

Алгоритм построения максимального потока в сети.

Вначале зададим начальное значение потока, если оно не задано. Обычно в качестве начального потока φ_0 выбирают нулевой поток: $\varphi_0(v_i, v_j) = 0$.

Максимальный поток определяется в результате итерационной процедуры. Итерацию составляют два следующих шага.

1. Выделим подмножество вершин *A*, достижимых из истока по ненасыщенным дугам. Если сток не войдет в подмножество *A*, то построенный поток максимальный и задача решена. Если сток попадет в подмножество *A*, то следует перейти к шагу 2.

Процедуру выделения подмножества вершин *A* начинают с истока – вершины v_0: составляют список вершин, в которые можно попасть из v_0 по ненасыщенным ребрам. Затем такие же списки составляют для вершин, которые вошли в предыдущий список и т.д. При этом вершины, встречавшиеся в прежних списках, повторно не выписывают.

2. *Построение насыщенного пути*. Выделим путь из истока в сток, состоящий из ненасыщенных ребер, и увеличим значение потока по каждой дуге пути на величину

$$\Delta = \min_{u}(c_{ij} - \varphi_{ij}) .$$

Величина Δ называется коэффициентом насыщения. Для построения нового потока рекомендуется выбирать путь, для которого коэффициент насыщения принимает наибольшее значение и/или одновременно соответствует нескольким дугам пути.

Процедура продолжается до получения максимального потока.

Пример 6. Для заданной сети S (рис.15) сформировать максимальный поток, направленный из истока v_0 в сток s. Определить минимальный разрез сети. Рядом с каждой дугой указана величина её пропускной способности.

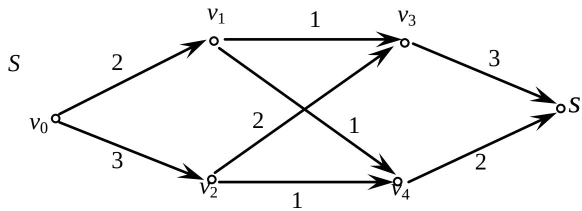


Рис. 15

Решение. Сформируем начальный поток: $\varphi(u) = 0$. Значение потока по дуге будем указывать в скобках рядом с пропускной способностью (рис.16).

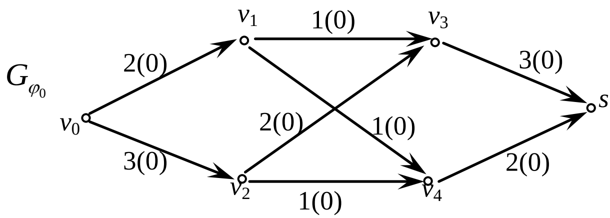


Рис. 16

Первая итерация.

1. Составим список вершин, в которые можно попасть из вершины v_0 по ненасыщенным ребрам:

$$v_0 \,||\, v_1, v_2.$$

Затем составим список вершин, в которые можно попасть из вершин v_1 и v_2 по ненасыщенным ребрам:

$$v_1 \,||\, v_3, v_4; \qquad v_2 \,||\, v_3, v_4.$$

И наконец, $v_3 \,||\, s$; $v_4 \,||\, s$. Поскольку вершина u вошла в подмножество вершин, достижимых из вершины v_0, значит поток не является максимальным. Можно построить путь с большим значением потока.

2. По спискам строим путь, содержащий ненасыщенные дуги:

$$v_0 \to v_1 \to v_3 \to s.$$

Определим величину, на которую можно увеличить значение потока

$$\Delta = \min_u \{2-0; 1-0; 3-0\} = 1.$$

Увеличиваем поток по построенному пути на 1 единицу (рис. 17).

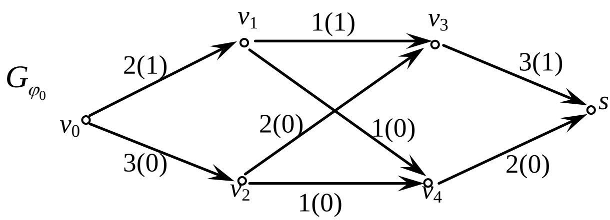


Рис. 17

Вторая итерация.

1. Составим список вершин, в которые можно попасть из вершин v_0, v_1, v_2, v_3, v_4:

$$v_0 \,||\, v_1, v_2; \quad v_1 \,||\, v_4; \quad v_2 \,||\, v_3, v_4; \quad v_3 \,||\, s; \quad v_4 \,||\, s.$$

2. По спискам строим путь, содержащий ненасыщенные дуги:

$$v_0 \to v_2 \to v_3 \to s;$$

Определим величину, на которую можно увеличить значение потока

$$\Delta = \min_u \{3-0; 2-0; 3-1\} = 2\,.$$

Увеличиваем поток по построенному пути на 2 единицы (рис.18).

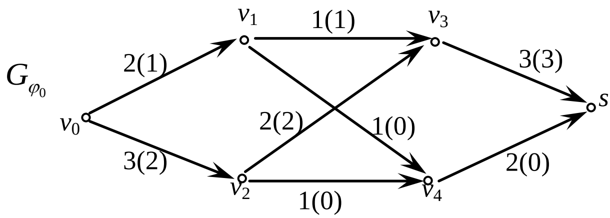


Рис. 18

Третья итерация.

1. Составим список вершин, в которые можно попасть из вершин v_0, v_1, v_2, v_4:

$$v_0 \,||\, v_1, v_2; \quad v_1 \,||\, v_4; \quad v_2 \,||\, v_4; \quad v_4 \,||\, s.$$

2. По спискам строим путь:

$$v_0 \to v_2 \to v_4 \to s.$$

$$\Delta = \min_u \{3-2; 1-0; 2-0\} = 1\,.$$

Увеличиваем поток по построенному пути на 1 единицу (рис. 19).

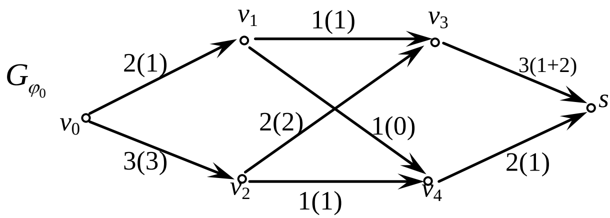


Рис. 19

Четвертая итерация.

1. Составим список вершин, в которые можно попасть из вершин v_0, v_1, v_4:

$$v_0 \,||\, v_1; \quad v_1 \,||\, v_4; \quad v_4 \,||\, s.$$

2. По спискам строим увеличивающую цепь:

$$v_0 \to v_1 \to v_4 \to s;$$

$$\Delta = \min_{u}\{2-1; 1-0; 2-1\} = 1.$$

Увеличиваем поток по построенному пути на 1 единицу (рис. 20).

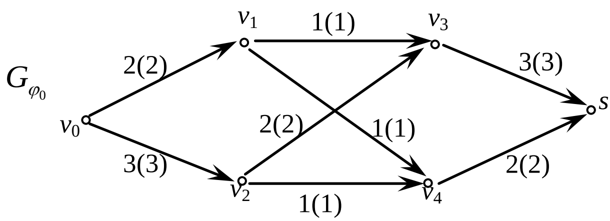


Рис. 20

Пятая итерация.

1. Составим список вершин, в которые можно попасть из вершин v_0:

$$v_0 \,||\, \ldots.$$

Поскольку все дуги, исходящие из v_0, насыщенные, дальнейшее увеличение потока невозможно и, значит, максимальный поток найден. Его величина $\varphi_{\max} = 3 + 2 = 5$.

Проверить правильность полученного ответа можно, определив величину минимального разреза. Его значение должно совпадать со значением максимального потока.

Поскольку $A = \{v_0\}$; $B=\{v_1,\ v_2,\ v_3,\ v_4,\ s\}$, в минимальный разрез входят дуги R(A/B)=$\{(v_0,\ v_1),\ (v_0;\ v_2)\}$. Сумма пропускных способностей этих дуг: $C(A/B) = c(v_0, v_1) + c(v_0; v_2) = 3+2=5$.

6. МАРКОВСКИЕ ПРОЦЕССЫ С ДИСКРЕТНЫМИ СОСТОЯНИЯМИ И НЕПРЕРЫВНЫМ ВРЕМЕНЕМ

Для описания работы стохастических систем, выходные параметры которых случайным образом зависят от входных параметров, используются графы и аппарат, в основе которого лежат марковские случайные процессы (СП). Примерами таких систем являются системы массового обслуживания: телефонные станции, ремонтные мастерские, вычислительные комплексы, билетные кассы, магазины и т.д.

Под *случайным процессом* понимается процесс изменения во времени состояния какой-либо системы в соответствии с вероятностными закономерностями. Примеры случайных процессов: динамика изменения курса валют или акций, выручка или прибыль организации с течением времени, объемы продаж товара и т. д.

Рассмотрим некоторую систему S, в которой в данный момент времени t_0 протекает СП с дискретными состояниями и непрерывным временем, т. е. случайный процесс может изменить свое состояние в произвольный момент времени и его возможные состояния S_1, S_2, S_3, … можно заранее перечислить. Этот процесс называется *марковским* или *случайным процессом без последствия*, если для любого момента времени t_0 поведение системы в будущем зависит только от его состояния в данный момент t_0 и не зависит от того, когда и как система пришла в это состояние.

Простейшим видом СП с дискретными состояниями и непрерывным временем является поток событий. *Потоком событий* называется некоторая последовательность однотипных событий, которые происходят в случайные моменты времени (например, звонки по телефону, посетители магазина, автомобили, проезжающие перекресток, и т. д.).

Важнейшей характеристикой любого потока событий является его интенсивность λ – среднее число событий, произошедших в потоке за одну единицу времени.

С интенсивностью тесно связана величина $T = \frac{1}{\lambda}$, которая имеет смысл среднего интервала времени между двумя событиями.

Если события в потоке происходят поодиночке и для любых непересекающихся интервалов времени число событий в одном интервале никак не влияет на то, сколько и каким образом будут происходить события в другом интервале (поток без последствия), то поток называется потоком Пуассона или пуассоновским потоком.

Все возможные переходы между состояниями описывают с помощью графа состояний системы. Граф состояний представляет собой взвешенный ориентированный граф, вершинами которого являются возможные состояния S_i. Две вершины соединяются дугой, если возможен непосредственный переход между состояниями, а вес дуги, ведущей из состояния S_i в состояние S_j, равен интенсивности $\lambda_{ij}(t)$, $\lambda_{ij}(t) \neq 0$, пуассоновского потока. Интенсивности переходных потоков λ_{ij} указываются на графе состояний весом соответствующей дуги. Главная задача в таких моделях состоит в определении вероятностей состояний p_i, которые имеют смысл средней доли времени, которое система проводит в этом состоянии.

Рассмотрим 2 состояния S_i и S_j (рис. 21). Интенсивностью переходного потока λ_{ij} называется среднее число переходов из состояния S_i в состояние S_j за единицу времени, которое система проводит в состоянии S_i. Если известно среднее время T_{ij}, которое система проводит в S_i до того как перейдет в S_j, то можно записать: $\lambda_{ij} = \frac{1}{T_{ij}}$.

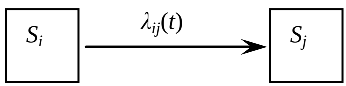


Рис. 21

Будем считать, что переход системы из состояния S_i в состояние S_j осуществляется под воздействием пуассоновского потока с интенсивностью $\lambda_{ij}(t)$. В этом случае для нахождения вероятностей состояний составляется система дифференциальных уравнений Колмогорова:

$$\frac{dp_i(t)}{dt} = \sum_{j=1}^{n} p_j(t)\lambda_{ji}(t) - p_i(t)\sum_{j=1}^{n}\lambda_{ij}(t), \qquad i=1,\ 2,\ \ldots,\ n.$$

Первая сумма в правой части формулы распространяется на те значения j, для которых возможен непосредственный переход из состояния S_j в состояние S_i, т. е. для которых $\lambda_{ji}(t) \neq 0$, а вторая – на те значения j, для которых возможен непосредственный переход из состояния S_i в состояние отказа S_j, т. е. $\lambda_{ij}(t) \neq 0$.

Систему дифференциальных уравнений решают при начальных условиях, задающих вероятности состояний в начальный момент при $t = 0$:

$$p_1(0), p_2(0),..., p_n(0),$$

причем для любого момента времени t выполняется нормировочное условие:

$$\sum_{i=1}^{n} p_i(t) = 1,\ (t \geq 0).$$

Это следует из того, что в любой момент t события $\{S(t) = S_1\}, \{S(t) = S_2\},..., \{S(t) = S_n\}$ образуют полную группу несовместных событий. Система уравнений является вырожденной, поэтому для нахождения единственного решения системы дифференциальных уравнений нужно заменить одно любое уравнение на условие нормировки.

Уравнения Колмогорова составляют по следующему правилу: *для каждого из возможных состояний системы записывается уравнение, в левой части которого стоит* $\frac{dp_i(t)}{dt}$, *а справа – столько слагаемых, сколько дуг графа соприкасается с данным состоянием. Если дуга направлена в данное состояние, то перед слагаемым* *ставится плюс, если дуга направлена из данного состояния – минус. Каждое из слагаемых будет равно произведению интенсивности перехода из данного состояния (либо в данное состояние) на вероятность состояния, из которого выходит стрелка.*

Заметим, что для стационарного процесса производные $\dfrac{dp_i(t)}{dt}$ равны нулю (вероятности состояний не меняются с течением времени). Система дифференциальных уравнений переходит при этом в систему алгебраических уравнений, описывающих стационарный режим.

Пример 7. Взвешенный граф состояний системы имеет вид (рис. 22)

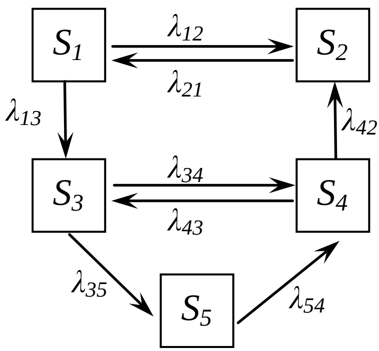


Рис. 22

Написать уравнения Колмогорова для вероятностей состояний и указать, при каких начальных условиях их нужно решать, если в начальный момент система S с вероятностью $\dfrac{1}{2}$ находится в состоянии S_1 и с вероятностью $\dfrac{1}{2}$ - в состоянии S_2.

Решение. Система дифференциальных уравнений Колмогорова имеет вид

$$
\begin{cases}
\dfrac{dP_1(t)}{dt} = P_2(t)\lambda_{21} - P_1(t)(\lambda_{12} + \lambda_{13}),\\
\dfrac{dP_2(t)}{dt} = P_1(t)\lambda_{12} + P_4(t)\lambda_{42} - P_2(t)\lambda_{21},\\
\dfrac{dP_3(t)}{dt} = P_1(t)\lambda_{13} + P_4(t)\lambda_{43} - P_3(t)(\lambda_{34} + \lambda_{35}),\\
\dfrac{dP_4(t)}{dt} = P_3(t)\lambda_{34} + P_5(t)\lambda_{54} - P_4(t)(\lambda_{43} + \lambda_{42}),\\
\dfrac{dP_5(t)}{dt} = P_3(t)\lambda_{35} - P_5(t)\lambda_{54}.
\end{cases}
$$

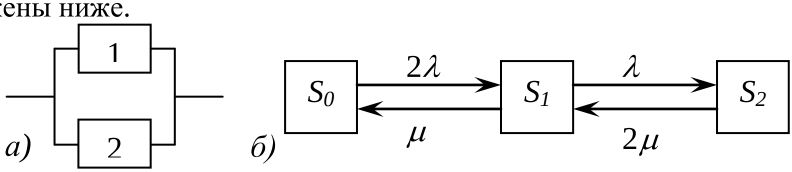
Любое из этих уравнений может быть отброшено, а соответствующая ему вероятность $P_i(t)$ (i=1,2,3,4,5) выражена через остальные с помощью нормировочного условия:

$$P_1(t) + P_2(t) + P_3(t) + P_4(t) + P_5(t) = 1.$$

Начальные условия, при которых решается система дифференциальных уравнений, имеют вид:

$$P_1(0) = P_2(0) = 0{,}5,$$
$$P_3(0) = P_4(0) = P_5(0) = 0.$$

Пример 8. Определить вероятности состояний системы, структурная схема и взвешенный граф состояний которой изображены ниже.



Известны интенсивности отказов элементов $\lambda = 0{,}02$, а интенсивности восстановления $\mu = 1{,}0$.

Решение. Система может находиться в одном из трёх состояний. Состояние S_0 – два элемента, входящие в систему, работоспособны. Состояние S_1 – один из элементов, входящих в систему, в отказовом состоянии. Состояние S_2 – оба элемента отказали. Интенсивности переходов системы в состояния S_0, S_1 и S_2 равны:

$$\lambda_{01} = 2\lambda_1 = 2 \cdot 0{,}02 = 0{,}04;$$
$$\lambda_{12} = \lambda_1 = 0{,}02;$$
$$\mu_{21} = 2\mu_1 = 2 \cdot 1{,}0 = 2{,}0;$$
$$\mu_{10} = \mu_1 = 1{,}0.$$

Составим систему дифференциальных уравнений, с помощью которых можно определить вероятности состояний системы:

$$\begin{cases} \dfrac{dP_0(t)}{dt} = -\lambda_{01}P_0(t) + \mu_{10}P_1(t), \\ \dfrac{dP_1(t)}{dt} = \lambda_{01}P_0(t) - (\lambda_{12} + \mu_{10})P_1(t) + \mu_{21}P_2(t), \\ \dfrac{dP_2(t)}{dt} = -\mu_{21}P_2(t) + \lambda_{12}P_1(t). \end{cases}$$

Будем считать рассматриваемый марковский процесс стационарным и производные $dP_i(t)/dt$ примем равными нулю. Полученная система алгебраических уравнений примет вид

$$\begin{cases} 0 = -\lambda_{01}P_0(t) + \mu_{10}P_1(t); \\ 0 = \lambda_{01}P_0 - (\lambda_{12} + \mu_{10})P_1(t) + \mu_{21}P_2(t); \\ 0 = -\mu_{21}P_2(t) + \lambda_{12}P_1(t); \\ P_0(t) + P_1(t) + P_2(t) = 1. \end{cases}$$

Четвертое уравнение для этой системы (при трех неизвестных) становится необходимым потому, что первые три уравнения сводятся к двум. Решая систему уравнений, получим значения искомых вероятностей

$$P_0(t) = \frac{1}{1 + \frac{\lambda_{01}}{\mu_{10}} + \frac{\lambda_{01}\lambda_{12}}{\mu_{21}\mu_{10}}} \approx 0{,}9612,$$

$$P_1(t) = \frac{P_0(t)\lambda_{01}}{\mu_{10}} \approx 0{,}0384,$$

$$P_2(t) = \frac{P_0(t)\lambda_{01}\lambda_{12}}{\mu_{10}\mu_{21}} \approx 0{,}0004.$$

Пример 9. Автоматизированная сборочная линия предприятия в среднем 1 раз в месяц выходит из строя и ремонтируется в среднем 3 дня. Кроме того, в среднем 2 раза в месяц она проходит техническое обслуживание, которое длится в среднем 1 день. В среднем в одном случае из трех при техническом обслуживании обнаруживается неполадка, и линия ремонтируется. Определить, какую среднюю прибыль приносит линия за месяц, если за один день безотказной работы прибыль равна 15 тыс. р. Один день технической обработки обходится в 20 тыс. р., а один день ремонта — 30 тыс. р.

Решение. Найдем вероятности состояний, равные долям времени работы, ремонта и технического обслуживания. Пусть:

S_1 — линия работает;

S_2 — техническое обслуживание;

S_3 — ремонт.

Граф состояний будет иметь вид (рис. 23):

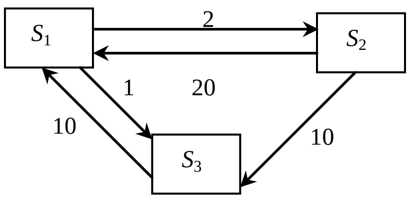


Рис. 23

Составим систему уравнений. В состояние S_1 входят 2 стрелки: из S_2 с интенсивностью 20 и из S_3 с интенсивностью 10, поэтому левая часть первого уравнения имеет вид $20P_2 + 10P_3$. Из состояния

S_1 выходят две стрелки с интенсивностями 2 и 1, поэтому правая часть первого уравнения системы примет вид $(2 + 1)\cdot P_1$. Аналогично, на основании состояний S_2 и S_3 составляем второе и третье уравнения. В результате система будет иметь вид:

$$\begin{cases} 20P_2 + 10P_3 = (2+1)P_1, \\ 2P_1 = (20+10)P_2, \\ P_1 + 10P_2 = 10P_3. \end{cases}$$

Данная система является вырожденной, и для ее решения нужно заменить одно любое (например, первое) уравнение условием нормировки $P_1 + P_2 + P_3 = 1$. В результате получим систему:

$$\begin{cases} 2P_1 = 30P_2, \\ P_1 + 10P_2 = 10P_3, \\ P_1 + P_2 + P_3 = 1. \end{cases}$$

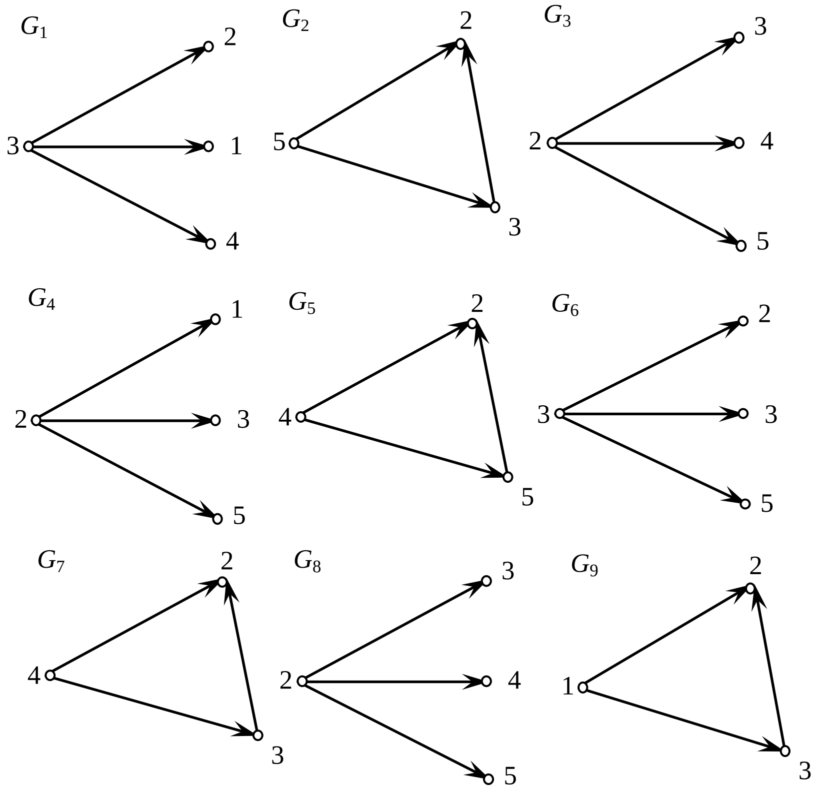
Выразив из 1-го и 2-го уравнений P_1, P_3 и, подставив результат в 3-е уравнение, получим:

$$P_1 = \frac{30}{37}; \quad P_2 = \frac{2}{37}; \quad P_3 = \frac{5}{37}.$$

Умножаем вероятности на 30 дней месяца и получим, что в среднем в месяц линия работает 24,3 дня, техническое обслуживание — 1,6 дней, ремонт — 4,1 дня. Отсюда следует, что средняя прибыль будет $24{,}3 \cdot 15 - 1{,}6 \cdot 20 - 4{,}1 \cdot 30 = 209{,}5$ тыс. руб.

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

Задача 1. Даны графы:



Построить матрицы смежности и инцидентности графа G, если

1. $G = G_1 \cup G_2 \cup G_3$;
2. $G = G_1 \cup G_2 \cup G_6$;
3. $G = G_1 \cup G_5 \cup G_8$;
4. $G = G_1 \cup G_7 \cup G_3$;
5. $G = G_1 \cup G_7 \cup G_6$;
6. $G = G_1 \cup G_9 \cup G_3$;
7. $G = G_1 \cup G_9 \cup G_8$;
8. $G = G_4 \cup G_7 \cup G_8$.

Задача 2. В таблице указаны декартовы координаты вершин графа и перечислены ребра графа. Построить граф на плоскости Oxy и найти:

1) таблицу степеней вершин;
2) матрицу смежности;
3) матрицу инцидентности;
4) таблицу расстояний в графе;
5) определить эксцентриситет вершин графа;
6) определить диаметр графа и периферийные вершины;
7) определить радиус и центр графа;
8) определить количество маршрутов длины 2;
9) выписать все маршруты длины 3 с началом в вершине x_3.

Вариант		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	координаты вершин	(1;3)	(3;5)	(6;5)	(2;2)	(3;3)	(1;0)	(3;0)	(6;2)
	Список ребер	$(x_1;x_2)$, $(x_2;x_5)$, $(x_2;x_3)$, $(x_2;x_4)$, $(x_1;x_6)$, $(x_2;x_7)$, $(x_6;x_7)$, $(x_7;x_8)$, $(x_3;x_8)$							
2	координаты вершин	(4;6)	(2;4)	(4;4)	(6;4)	(2;0)	(4;1)	(6;0)	(9;2)
	Список ребер	$(x_1;x_2)$, $(x_2;x_5)$, $(x_2;x_3)$, $(x_1;x_4)$, $(x_4;x_7)$, $(x_6;x_7)$, $(x_1;x_3)$, $(x_3;x_4)$, $(x_5;x_6)$, $(x_3;x_8)$							
3	координаты вершин	(2;3)	(2;6)	(3;7)	(3;5)	(5;6)	(5;4)	(6;6)	(4;1)
	Список ребер	$(x_1;x_2)$, $(x_2;x_3)$, $(x_4;x_6)$, $(x_3;x_4)$, $(x_5;x_6)$, $(x_3;x_5)$, $(x_5;x_7)$, $(x_7;x_8)$							
4	координаты вершин	(1;1)	(2;2)	(2;4)	(2;5)	(3;5)	(5;5)	(3;2)	(5;2)
	Список ребер	$(x_1;x_2)$, $(x_2;x_3)$, $(x_5;x_6)$, $(x_3;x_5)$, $(x_6;x_8)$, $(x_2;x_7)$, $(x_7;x_8)$, $(x_5;x_7)$, $(x_1;x_4)$, $(x_3;x_4)$							
5	координаты вершин	(1;4)	(3;5)	(5;4)	(1;2)	(5;2)	(1;0)	(5;0)	(7;1)
	Список ребер	$(x_1;x_2)$, $(x_2;x_4)$, $(x_2;x_5)$, $(x_2;x_3)$, $(x_4;x_5)$, $(x_6;x_7)$, $(x_5;x_7)$, $(x_4;x_6)$, $(x_7;x_8)$, $(x_3;x_8)$							

Задача 3.

3.1. По матрице весов построить граф и найти кратчайший путь между вершинами x_1 и x_8, используя метод Дейкстры.

3.2. С помощью алгоритма «ближайшего соседа» определить минимальное остовное дерево в графе из задания 3.1.

1.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	10	6	7	11	21		
x_2	10	0	5				15	
x_3	6	5	0				11	19
x_4	7			0	9	13	10	
x_5	11			9	0			
x_6	21			13		0	18	10
x_7		15	11	10		18	0	4
x_8			19			10	4	0

2.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	3	2	8	6		15	
x_2	3	0	4					
x_3	2	4	0	3				6
x_4	8		3	0			3	4
x_5	6				0	3	2	
x_6					3	0		2
x_7	15			3	2		0	6
x_8			6	4		2	6	0

3.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	5		4	6			
x_2	5	0						10
x_3			0	4		6	5	7
x_4	4		4	0	8		12	9
x_5	6			8	0	4		
x_6			6		4	0	3	
x_7			5	12		3	0	5
x_8			7	9			5	0

4.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	5	5		6			
x_2	5	0		7				
x_3	5		0		6	8		16
x_4		7		0	3	6	9	
x_5	6		6	3	0	4		
x_6			8	6	4	0	4	6
x_7				9		4	0	8
x_8			16			6	8	0

5.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	4	6			21		
x_2	4	0		10		5		
x_3	6		0		9	5		
x_4		10		0		8		8
x_5			9		0	6	4	6
x_6	21	5	5	8	6	0	10	11
x_7					4	10	0	5
x_8				8	6	11	5	0

6.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	10	11					
x_2	10	0	20	26	14			
x_3	11	20	0	16		25		
x_4		26	16	0	21	26	6	
x_5		14		21	0	4	28	
x_6			25	26	4	0		13
x_7				6	28		0	15
x_8						13	15	0

7.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	8	14	13	16			
x_2	8	0			14	6		
x_3	14		0	5		8		10
x_4	13		5	0			4	12
x_5	16	14			0		8	
x_6		6	8			0		15
x_7				4	8		0	9
x_8			10	12		15	9	0

Задача 4.

Определить существование насыщающего потока сети (рис. 18), построить существующий (максимальный или насыщенный) поток.

В таблице для каждого номера задана величина пропускной способности соответствующей дуги сети.

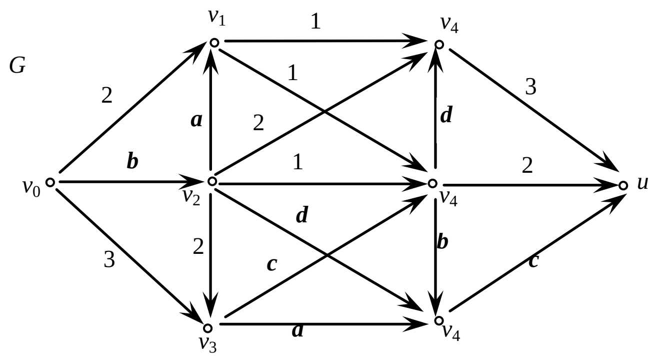


Рис. 18.

№ задания	*a*	*b*	*c*	*d*	№ задания	*a*	*b*	*c*	*D*
1	2	3	4	2	***5***	3	4	2	3
2	3	2	4	3	***6***	4	4	3	2
3	4	2	3	4	***7***	2	3	3	4
4	3	4	4	3	***8***	4	2	3	3

Задача 5. Техническая система S – вычислительный центр (ВЦ), состоящий из трех ЭВМ: 1, 2, 3. Каждая из ЭВМ выходит из строя (отказывает) независимо от других. Потоки отказов ЭВМ – пуассоновские с переменными интенсивностями, равными $\lambda_1(t), \lambda_2(t), \lambda_3(t)$. После отказа каждая ЭВМ восстанавливается; потоки восстановлений – пуассоновские с интенсивностями $\mu_1(t), \mu_2(t), \mu_3(t)$; потоки восстановлений тоже независимы. Рас-

сматриваются следующие состояния системы:

S_1 – все ЭВМ исправны

S_2 – ЭВМ 1 отказала, ЭВМ 2 и ЭВМ 3 исправны;

S_3 – ЭВМ 2 отказала, ЭВМ 1 и ЭВМ 3 исправны;

S_4 - ЭВМ 3 отказала, ЭВМ 1 и ЭВМ 2 исправны;

S_5 - ЭВМ 1 и ЭВМ 2 отказали, а ЭВМ 3 исправна;

S_6 - ЭВМ 1 и ЭВМ 3 отказали, а ЭВМ 2 исправна;

S_7 - ЭВМ 2 и ЭВМ 3 отказали, а ЭВМ 1 исправна;

S_8 – все три ЭВМ отказали.

Построить взвешенный граф состояний ВЦ. Составить уравнения Колмогорова для вероятностей состояний $p_1(t),\ldots, p_8(t)$. Записать нормировочное условие, позволяющее указать, при каких начальных условиях решается систему дифференциальных уравнений, если известно, что в начальный момент $t = 0$ все ЭВМ исправны.

Задача 6. Система представляется в виде технического устройства (аппаратура, производственный агрегат и т.п.), которое имеет три узла (элемента). Для работы технического устройства достаточно, чтобы работал хотя бы один узел. Система может находиться в следующих четырех состояниях:

e_1 – все узлы системы работают исправно;

e_2 – только один узел системы вышел из строя и подлежит восстановлению (ремонтируется или планируется его замена);

e_3 – два узла системы вышли из строя и восстанавливаются;

e_4 – все три узла системы вышли из строя и восстанавливаются.

Граф системы приведен на рис. 19.

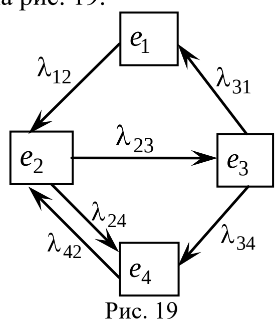


Рис. 19

Интенсивности переходов λ_{ij} из состояния e_i в состояние e_j .равны: $\lambda_{12} = 2$, $\lambda_{23} = 2$, $\lambda_{24} = 3$, $\lambda_{31} = 1$, $\lambda_{34} = 2$, $\lambda_{42} = 1$.

Определить:

1. Распределение вероятностей состояний для любого момента времени на интервале $t \in [0;5]$ с шагом $h = 0{,}5$;
2. Вектор финальных вероятностей системы;
3. Эффективность работы системы, если вектор стоимостей состояний системы: $e_1 = 18$, $e_2 = 8$, $e_3 = -5$, $e_4 = -18$.

РЕКОМЕНДАТЕЛЬНЫЙ БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Яблонский С.В. Введение в дискретную математику: Учеб. пособие для вузов / Под ред. В.А. Садовничего. 3-е изд., стер. М.: Высш.шк., 2002. – 384 с.
2. Белоусов А.И., Ткачев С.Б. Дискретная математика: Учеб. для вузов / Под ред. Зарубина В.С., Крищенко *А.П.* М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. 744 с.
3. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженера. М.: Энергия, 1980. 344 с.
4. Гончарова Г. А., Мочалин А. А. Элементы дискретной математики: Учебное пособие. М.: ФОРУМ: ИНФРА-М, 2003. 128 с.
5. Основина О.Н. Надежность информационных систем. Методические указания к практическим занятиям. С.: СТИ МИСиС, 2006. 68 с.
6. Олехник С. Н., Нестеренко Ю. В., Потапов М. К. «Старинные занимательные задачи». М.: «Наука», 1988. 160 с.
7. Полак Л. С. Уильям Гамильтон, 1805-1865 / Под ред. А. Т. Григорьян. М.: Наука, 1993. 267 с.

СОДЕРЖАНИЕ