

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Санкт-Петербургский  
государственный университет аэрокосмического приборостроения

---

Л. А. Прокушев

ДИСКРЕТНАЯ МАТЕМАТИКА  
Основы теории графов и алгоритмизации задач

Учебное пособие

Санкт-Петербург  
2000

УДК 519.2  
ББК 22.174  
П80

## **Прокушев Л. А**

П80 Дискретная математика (основы теории графов и алгоритмизации задач): Учеб. пособие / СПбГУАП. СПб., 2000. 82 с.: ил.

Рассмотрены основные определения и понятия теории графов, необходимые для решения некоторых прикладных задач дискретной математики (определение оптимальных расстояний между множеством объектов, поиск критического пути в задаче сетевого планирования и управления, выбор предпочтительных вариантов системы по множеству критериев). Обсуждаются подходы к разработке компьютерных алгоритмов задач на основе моделей теории графов.

Учебное пособие предназначено для студентов специальности “Системы автоматизированного проектирования”, а также для студентов других специальностей, использующих теорию графов для решения задач.

Рецензенты:  
кафедра промышленного менеджмента Санкт-Петербургского  
Балтийского государственного технического университета;  
кандидат технических наук доцент *В.П. Понов*

Утверждено  
редакционно-издательским советом университета  
в качестве учебного пособия

- © Санкт-Петербургский  
государственный университет  
аэрокосмического  
приборостроения, 2000
- © Л. А. Прокушев, 2000

## ВВЕДЕНИЕ

Теория графов, являясь разделом дискретной математики, используется для описания и изучения отношений между дискретными объектами. Геометрически объекты можно представить в виде множества точек, называемых вершинами графа, а отношения между ними – линиями, связывающими вершины. Если линии не ориентированы, то их называют ребрами, а сам граф – неориентированным, а если линии ориентированы, то их называют дугами, а сам граф – ориентированным, или оргграфом.

Имея в своей основе простейшие идеи и элементы (точки, соединенные линиями), теория графов строит из них огромное многообразие форм, наделяет эти формы различными свойствами и в результате становится полезным инструментом при исследовании самых разнообразных систем.

В самом понятии графа сочетаются теоретико-множественные, комбинаторные и топологические аспекты, что делает теорию графов удобным, простым языком для формулировки и построения моделей, а также эффективным и мощным инструментом решения задач, относящихся к широкому кругу научных и инженерных проблем. Можно упомянуть в этой связи вопросы конструирования систем автоматизированного проектирования, систем программирования для ЭВМ и создания операционных систем, исследования операций и управления, математической лингвистики и разработки трансляторов, календарное планирование промышленного производства, задачи сетевого планирования и управления (СПУ), тактические и логические задачи выбора лучших объектов, большой круг экономических задач, игровые задачи.

Вместо понятия графа часто используется понятие “сеть”, когда кроме основных, чисто структурных, отношений в графе задаются некоторые количественные характеристики точек и линий. При этом можно решать проблемы построения электрических сетей, систем связи и исследования процессов передачи информации, выбора оптимальных маршрутов и потоков в сетях, например построения сети выполнения ра-

бот при проектировании изделия. При этом ребрам или (и) дугам сети ставятся в соответствие определенные количественные характеристики энергии, затрат и потока.

Цель данного курса состоит в том, чтобы ознакомить студентов с основными понятиями теории графов и первыми представлениями о моделях теории графов, их приложениях к решению некоторых прикладных задач с использованием алгоритмического подхода и с возможностью применения ЭВМ.

## 1. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ И ПОНЯТИЯ

### 1.1. Элементы теории множеств

Для определения основных понятий теории графов будут использованы некоторые сведения из теории множеств. Множество образуется совокупностью дискретных объектов, являющихся элементами множества, и обозначаются  $A = \{a, b, c\}$ , где  $a, b, c$  – элементы множества  $A$ .

Предполагается, что для конкретного элемента  $a$  и множества  $A$  всегда можно определить, принадлежит элемент  $a$  множеству  $A$  (обозначается  $a \in A$ ) или не принадлежит ( $a \notin A$ ).

Множество  $A$  называется конечным, если в нем содержится конечное число элементов, например,  $A = \{a_1, a_2, \dots, a_n\}$ . Число  $n$  называется количеством элементов  $A$  и обозначается через  $|A|$ . Множество, состоящее из одного элемента, обозначается через  $\{a\}$ . Множество, не содержащее элементов, называется пустым и обозначается  $\emptyset$ , например  $A = \emptyset$  означает, что  $A$  – пустое множество.

#### 1.1.1. Операции над множествами

Вложенность множеств (подмножество) – операция  $\subseteq$ .

Если из  $x \in A$  (где  $x$  – любой элемент) следует  $x \in B$ , то  $A$  называют подмножеством  $B$  ( $A \subseteq B$ ) и говорят, что  $A$  вложено в  $B$  ( $A$  содержится в  $B$ ,  $B$  содержит  $A$ ). Например, если  $A = \{a, b\}$  и  $B = \{a, b, c\}$ , то  $A \subseteq B$ .

Дополнение множеств – операция  $\cup$  (сумма).

Множество  $A \cup B$  включает все элементы множеств  $A$  и  $B$  без повторения элементов, например пусть  $A = \{a, b\}$ ,  $B = \{b, c\}$ , тогда  $A \cup B = \{a, b, c\}$ .

Пересечение множеств – операция  $\cap$ .

Множество  $A \cap B$  включает элементы общие для множеств  $A$  и  $B$ , например:

1. Пусть  $A = \{a, b\}$ ,  $B = \{b, c\}$ , тогда  $A \cap B = \{b\}$ .

2. Пусть  $A=\{a, b\}$ ,  $B=\{c, d\}$ , тогда  $A\cap B=\emptyset$ .

Дополнение множеств – операция  $\setminus$  (вычитание).

Множество  $A\setminus B$  включает все элементы множества  $A$ , не принадлежащие  $B$ , т. е. из  $A$  вычитаются элементы общие с  $B$ . Например, пусть  $A=\{a, b\}$ ,  $B=\{b, c\}$ , тогда  $A\setminus B=\{a\}$ .

Теоретико-множественное произведение – операция  $\times$ .

Пусть  $A$  и  $B$  – два множества. Через  $A\times B$  обозначим множество, состоящее из упорядоченных пар  $(a, b)$ , таких, что  $a\in A$ ,  $b\in B$ . Иначе говоря,  $c\in A\times B$  тогда и только тогда, когда  $c$  есть пара  $(a, b)$ , причем  $a\in A$ ,  $b\in B$ .

Например, пусть  $A=\{a, b\}$ ,  $B=\{c, d\}$ , тогда  $A\times B=\{(a, c), (a, d), (b, c), (b, d)\}$ .

Для облегчения общего определения введем понятие *произведения множества самого на себя*.

Упорядоченным, или декартовым (прямым), произведением множества  $V$  самого на себя (которое обозначается  $V\times V$ ) называется множество всех упорядоченных пар  $(s, t)$ , где  $s\in V$  и  $t\in V$ . Здесь  $(s, t)$  и  $(t, s)$  рассматриваются как различные элементы, исключая случай  $s=t$ . Если число элементов  $|V|=n$ , то  $V\times V$  состоит из  $n^2$  упорядоченных пар. Например, пусть  $V=\{v_1, v_2, v_3\}$ , тогда  $V\times V=\{(v_1, v_1), (v_1, v_2), (v_1, v_3), (v_2, v_1), (v_2, v_2), (v_2, v_3), (v_3, v_1), (v_3, v_2), (v_3, v_3)\}$ .

Неупорядоченным произведением множества  $V$  самого на себя (обозначим как  $V\&V$ ) называется множество всех различных неупорядоченных пар  $[s, t]$ , при этом пары  $[s, t]$  и  $[t, s]$  эквивалентны и так же, как при декартовом произведении, допускается совпадение элементов пары, т. е.  $s=t$ . При  $|V|=n$  множество  $V\&V$  состоит из  $n(n+1)/2$  различных неупорядоченных пар. Например, пусть  $V=\{v_1, v_2, v_3\}$ , тогда  $V\&V=\{[v_1, v_1], [v_1, v_2], [v_1, v_3], [v_2, v_2], [v_2, v_3], [v_3, v_3]\}$ .

Справедливы отношения:  $A\cap B\subseteq A$ ,  $A\cap B\subseteq B$ ,  $A\subseteq A\cup B$ ,  $B\subseteq A\cup B$ .

Например, если  $A\subseteq B$ ,  $B\subseteq C$ , то  $A\subseteq C$ , поэтому из приведенных выше соотношений следует, что  $A\cap B\subseteq A\cup B$ .

Справедливы также формулы дистрибутивности:

$$(A\cup B)\cap C=(A\cap B)\cup(B\cap C), (A\cap B)\cup C=(A\cup C)\cap(B\cup C).$$

Равенство множеств – операция  $=$ .

$A=B$  справедливо тогда и только тогда, когда  $A\subseteq B$  и  $A\supseteq B$ .

### 1.1.2. Отношение

Введем понятие *отношения*. Говорят, что на  $A$  задано отношение  $R$ , если задано подмножество  $R$  декартова произведения  $A \times A$  ( $R \subseteq A \times A$ ). Элемент  $a \in A$  находится в отношении  $R$  к элементу  $b \in A$  (записывается как  $aRb$ ) тогда и только тогда, когда  $(a, b) \in R$ . При этом существен порядок элементов: может быть, что  $a$  находится в отношении  $R$  к элементу  $b$ , но  $b$  не находится в отношении  $R$  к  $a$ .

Отношение называется: а) *симметричным*, если  $aRb$  тогда и только тогда, когда  $bRa$ ; б) *рефлексивным*, если для любого  $a \in A$   $aRa$ ; в) *транзитивным*, если из  $aRb$  и  $bRc$  следует  $aRc$ ; г) *антисимметричным*, если из  $aRb$  и  $bRa$  следует  $a=b$ .

Из множества всех отношений выделяется класс отношений, одновременно симметричных, транзитивных и рефлексивных. Такие отношения называются отношениями эквивалентности.

### 1.1.3. Отображение

Термины *отображение* (как понятие однозначного соответствия) и *функция* будут рассматриваться как синонимы. Считаем, что задано отображение  $f: X \rightarrow Y$ , если каждому элементу  $x \in X$  сопоставлен элемент  $f(x) \in Y$ .

Пусть  $f: X \rightarrow Y$  – некоторое отображение. Если  $A \subset X$ , то образом  $A$  при отображении  $f$  называется множество элементов  $B \subset Y$  такое, что  $y \in B$  тогда и только тогда, когда существует  $x \in A$  такой, что  $y=f(x)$ . Прообразом  $C \subset Y$  при отображении  $f$  называется множество  $D \subset X$  такое, что  $x \in D$  тогда и только тогда, когда  $f(x) \in C$ . Образ и прообраз обозначим соответственно  $f(A)$  и  $f^{-1}(C)$ . Операция взятия образа или прообраза множества удовлетворяют соотношениям:

$$f^{-1}(A \cup B) = f^{-1}(A) \cup f^{-1}(B), f(A \cup B) = f(A) \cup f(B);$$

$$f^{-1}(A \cap B) = f^{-1}(A) \cap f^{-1}(B), f(A \cap B) \subset f(A) \cap f(B);$$

$$f^{-1}(Y \setminus A) = X \setminus f^{-1}(A), f(X \setminus A) \supset f(X) \setminus f(A).$$

Однако нельзя утверждать, что  $f(A \cap B) = f(A) \cap f(B)$  и  $f(X \setminus A) = f(X) \setminus f(A)$ . Достаточно рассмотреть постоянное отображение  $f: X \rightarrow Y$  (такое, что  $f(x) = y_0 \in Y$  для любого  $x \in X$ ).

Отображение называется *взаимно-однозначным*, если  $f(x) = Y$  и для любого  $y \in Y$  множество  $f^{-1}(\{y\})$  состоит из одного элемента. Нетрудно

видеть, что определено отображение  $f^{-1}: Y \rightarrow X$ . Именно,  $f^{-1}(y) = a$  тогда и только тогда, когда  $f^{-1}(\{y\}) = \{a\}$ .

## 1.2. Основные понятия для графов

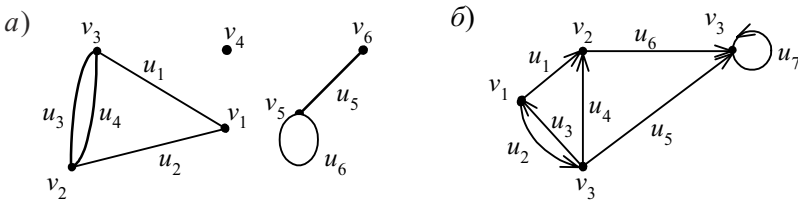
### 1.2.1. Геометрическое представление графов

Прежде чем определить понятие графа в наиболее общей форме, полезно рассмотреть геометрическую форму графов. Это позволит с самого начала получить удобное, наглядное представление различных понятий и структур, которые будут рассматриваться в дальнейшем. Любой граф в абстрактном смысле эквивалентен (по отношению к свойствам, изучаемым в теории графов) некоторому геометрическому графу. При некоторой *идеализации* многие известные структуры можно рассматривать как геометрические графы и изучать с помощью методов теории графов. Например, в виде графа можно представить систему или сеть улиц в городе, если пренебречь шириной последних, а пересечения считать точками.

Геометрический граф есть геометрическая конфигурация или структура в пространстве, состоящая из множества точек, взаимосвязанных множеством непрерывных, самонепересекающихся кривых.

На рис. 1.1 показано обычное представление геометрического графа, с помощью которого можно проиллюстрировать некоторые термины теории графов.

С позиции теории графов элементы  $v$  называются вершинами графа, а соединяющие их неориентированные линии ( $u$ ) ребрами или дугами, если линии снабжены стрелками, и тогда различают *неориентированный* граф (рис.1.1,а) или *ориентированный* граф (сокращенно орграф) (рис.1.1,б). Если множество вершин ( $V$ ) и ребер (дуг) ( $U$ ) конечно, то граф называется конечным. Мы будем рассматривать только конечные графы.



**Рис. 1.1. Геометрическое представление графов:**  
**а** – неориентированный граф; **б** – ориентированный граф



**Вершины**, связанные ребром или дугой, называются *смежными*. Если смежные вершины связаны более чем одним ребром или дугой, то такие ребра (дуги) называются кратными, или параллельными ( $u_3, u_4$  на рис. 1.1,*а*;  $u_2, u_3$  – на рис. 1.1,*б*), а сам граф называется мультиграфом, т. е. графы на рис. 1.1 являются мультиграфами. Вершины, несвязанные с другими, называются *изолированными*, например,  $v_4$  на рис. 1.1,*а*. Ребро (дуга), начинающееся и заканчивающееся в одной точке, называется петлей (петли  $u_6$  – на рис. 1.1,*а* и  $u_7$  – на рис. 1.1,*б*). Говорят, что ребро (дуга) и его граничные точки инцидентны друг другу, иначе говоря, вершины, инцидентные ребру (дуге), являются его концевыми точками, например, на рис. 1.1,*а* ребру  $u_1$  инцидентны вершины  $v_1$  и  $v_3$ , а вершине  $v_3$  инцидентны ребра  $u_1, u_3, u_4$ .

Вообще говоря, множество дуг  $U$  (но не вершин  $V$ ) может быть пустым, тогда говорят, что граф *вырожденный*.

### 1.2.2. Абстрактные графы

Геометрическое представление графов, являясь наглядным, трудно поддается математической обработке, поэтому необходимо его формализовать и представить определение графа в более общем абстрактном виде. Хотя многие графы, представляющие реальные объекты (после их идеализации), являются геометрическими графами, с точки зрения теории графов, их единственная структурная особенность состоит в том, что с каждым ребром связаны две (возможно совпадающие) вершины. Такие ребра (дуги) и вершины были определены выше как взаимно инцидентные. Понятие *инцидентности ребер (дуг) и их концевых вершин* является фундаментальным в теории графов, которая построена с учетом именно этой особенности и не учитывает реальной природы вершин и ребер (дуг).

Введение понятия *абстрактного графа* позволяет, во-первых, избавиться от случайных геометрических характеристик, сохранив наиболее существенные комбинаторные свойства графа, во-вторых, расширить возможности приложения теории, так как многие реальные структуры имеют комбинаторные свойства, которые полезно исследовать с помощью математического аппарата теории графов. Например, при проектировании сложного изделия (технического или программного) в виде графа можно задать соотношение между отдельными работами, которые составляют сложные проекты. В этом случае вершины представля-

ют отдельные работы, а отношения инцидентности графа отражают последовательность выполнения определенных работ.

Формализация графа может принимать разные формы, которые мы рассмотрим далее. Например, нижеследующая таблица содержит всю информацию, необходимую для описания геометрического графа (рис. 1.1,*a*):

Ребра	Вершины, соответствующие, т. е. инцидентные, ребрам
$u_1$	$v_1, v_3$
$u_2$	$v_1, v_2$
$u_3$	$v_1, v_3$
$u_4$	$v_2, v_3$
$u_5$	$v_5, v_6$
$u_6$	$v_5$
	$v_4$

Поскольку формализация и операции для неориентированных и ориентированных графов имеют отличия, приведем их абстрактные описания отдельно.

### 1.3. Основные понятия для неориентированных графов

#### 1.3.1. Абстрактные описания неориентированных графов

Пусть  $V$  – непустое множество вершин графа  $v \in V$ . Ребром, соединяющим две вершины  $v_i$  и  $v_j$ , называется неупорядоченная пара  $[v_i, v_j]$  или  $[v_j, v_i]$ . Ребро обозначается  $\bar{u} = (\overline{v_i, v_j})$  или  $\bar{u} = [v_j, v_i]$ , а множество ребер графа обозначается  $\bar{U}$ .

Множество ребер  $\bar{U}$  является конечным подмножеством неупорядоченного произведения  $V \times V$  с элементами вида  $[v, w]$ , где  $v, w \in V$  и допустимо совпадение элементов пары  $v = w$ .

Тогда граф  $G$  можно определить как совокупность непустого конечного множества вершин  $V$  и множества ребер  $\bar{U}$  (возможно пустого) и

обозначать  $G=(V, \bar{U})$ . При таком определении графа отношение инцидентности вершин и ребер задается не явно. Например, для геометрического графа (рис.1.2) соответствующий абстрактный граф можно описать как множества:

вершин  $V=\{v_1, v_2, v_3, v_4\}$  и

ребер  $\bar{U}=\{[v_1, v_2], [v_2, v_3], [v_3, v_4], [v_4, v_2], [v_4, v_1]\}$ .

Можно ввести отображение инцидентности (или инциденции) –  $\Gamma$  графа  $G$ . Если вершины  $v, w \in V$  являются граничными (концевыми) точками ребра  $\bar{u} \in \bar{U}$ , то это можно обозначить  $\bar{u} \sim [v, w]$  (читается “ $\bar{u}$  соединяет вершины  $v$  и  $w$ ”) и ввести отображение  $\Gamma(\bar{u})=[v, w]$  – это означает, что ребро  $\bar{u}$  инцидентно каждой из вершин  $v$  и  $w$  и наоборот – вершины инцидентны ребру.

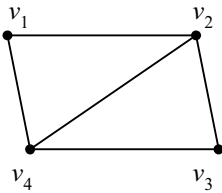


Рис. 1.2. Геометрический граф без обозначения ребер

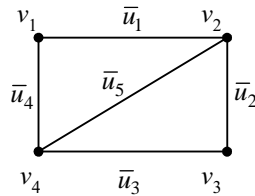


Рис. 1.3. Геометрический граф с обозначением ребер

Тогда граф  $G$  можно описать как математическую систему  $G=(V, \bar{U}, \Gamma)$ , состоящую из двух множеств  $V$  (вершин) и  $\bar{U}$  (ребер) и отображения  $\Gamma$  инцидентности множества  $\bar{U}$  в  $V \times V$  (множество неупорядоченных пар элементов вида  $[v, w] \in V \times V$ ).

Например, для геометрического графа (рис.1.3) соответствующий абстрактный граф можно описать как множества:

$V=\{v_1, v_2, v_3, v_4\}$ ,  $\bar{U}=\{\bar{u}_1, \bar{u}_2, \bar{u}_3, \bar{u}_4, \bar{u}_5\}$  и отображения инцидентности  $\Gamma=\{\bar{u}_1 \sim [v_1, v_2], \bar{u}_2 \sim [v_2, v_3], \bar{u}_3 \sim [v_3, v_4], \bar{u}_4 \sim [v_4, v_1], \bar{u}_5 \sim [v_1, v_2]\}$ .

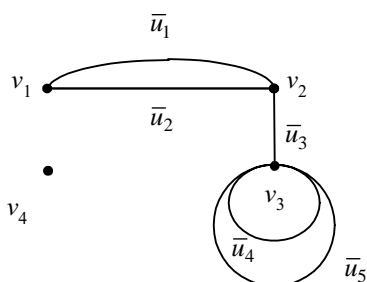
Поскольку отображение инцидентности графа  $\Gamma$  включает множество ребер графа  $\bar{U}$ , постольку описание графа как совокупности множества вершин  $V$  и отображения  $\Gamma$  множества  $\bar{U}$  в  $V \times V$  вида  $G=(V, \Gamma)$  полностью определяет граф.

### 1.3.2. Локальные свойства неориентированных графов

Для описания различных структурных свойств графов полезно ввести ряд дополнительных понятий, которые особенно наглядно иллюстрируются на примере геометрических графов, показанных выше, а также на рис. 1.4.

Если  $\bar{u} = [v, w]$ , то  $v$  и  $w$  называются *граничными точками*  $\bar{u}$  независимо от того, является граф геометрическим или нет.

Петлей  $\bar{u} = [v, v]$  называется ребро  $\bar{u}$  с единственной граничной точкой, т. е. петля инцидентна одной вершине, например петли  $\bar{u}_4$  и  $\bar{u}_5$  на рис. 1.4.



**Рис. 1.4. Пример неориентированного графа**

*Параллельными* (кратными) называются ребра, если они соединяют одни и те же вершины, например  $\bar{u}_1 \sim [v_1, v_2]$  и  $\bar{u}_2 \sim [v_1, v_2]$ . В частности, две петли, инцидентные одной и той же вершине, являются параллельными, например  $\bar{u}_4$  и  $\bar{u}_5$ .

Граф  $G$ , имеющий параллельные ребра, называется мультиграфом, например мультиграфы на рис. 1.1 и 1.4.

*Смежными вершинами* называются

вершины  $v$  и  $w$ , образующие граничные точки ребра  $\bar{u} \sim [v, w]$ . Одна вершина может иметь несколько смежных вершин, если в ней сходятся несколько ребер, например для вершины  $v_2$  смежными являются вершины  $v_1$  и  $v_3$  (рис. 1.4). В частности, вершина  $v$  смежна сама с собой, если существует петля, инцидентная  $v$ , например вершина  $v_3$  смежна сама с собой (рис. 1.4), в противном случае  $v$  не может быть смежной сама с собой.

*Смежными ребрами* называются такие, которые имеют, по крайней мере, одну общую граничную точку. Например, на рис. 1.4 смежными являются ребра  $\bar{u}_1$ ,  $\bar{u}_2$ ,  $\bar{u}_3$  с общей вершиной  $v_2$ . Смежными можно считать также параллельные ребра ( $\bar{u}_1$  и  $\bar{u}_2$ ) и петли ( $\bar{u}_4$  и  $\bar{u}_5$ ).

Заметим, что смежность является отношением между двумя подобными элементами (между вершинами или ребрами), тогда как инцидентность есть отношение между разнородными элементами (вершинами и ребрами).

### 1.3.3. Степень вершины

Степенью вершины  $\delta(v)$  называется число концов ребер, инцидентных вершине  $v$  (т. е. петля учитывается дважды). Например,  $\delta(v_3)=5$  (рис. 1.4).

Изолированной вершиной называется та, для которой  $\delta(v)=0$ , например  $\delta(v_4)=0$  (рис. 1.4).

Вырожденным называется граф, у которого все вершины изолированы.

Пусть дано  $|V|$  – число вершин, а  $|\bar{U}|$  – число ребер конечного графа  $G=(V, \bar{U})$ . Появление каждого нового ребра добавляет по единице к степеням двух вершин (или в случае петли добавляет два к степени одной вершины), поэтому справедливо выражение  $\sum_{v \in V} \delta(v) = 2|\bar{U}|$ , т. е.

это четное значение.

Если  $V_0$  и  $V_1$  – множества вершин, имеющих четные и нечетные степени соответственно, то значение  $\sum_{v \in V_0} \delta(v)$  четно, так как это конеч-

ная сумма четных чисел. Отсюда следует, что значение

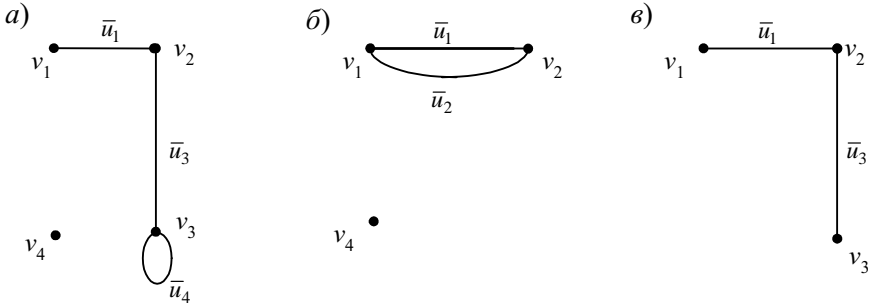
$$\sum_{v \in V} \delta(v) - \sum_{v \in V_0} \delta(v) = \sum_{v \in V_1} \delta(v)$$

также обязательно четно, что доказывает следующее утверждение: в конечном графе число вершин нечетной степени четно. Например, для графа рис.1.4 имеем  $\delta(v_1)=2$ ,  $\delta(v_2)=3$ ,  $\delta(v_3)=5$ ,  $\delta(v_4)=0$ , отсюда  $\delta(v_1)+\delta(v_4)=2$ ,  $\delta(v_2)+\delta(v_4)=8$ .

### 1.3.4. Разделение неориентированного графа на составляющие

Рассмотрим понятия, используемые для выделения важнейших частей графа. Пусть задан граф  $G=(V, \bar{U})$  (рис.1.4).

Суграфом графа  $G$  называется граф  $G'=(V, \bar{U}')$ , где  $\bar{U}' \subset \bar{U}$ , т. е. суграф получается из исходного графа удалением некоторого числа ребер (с сохранением вершин). Например, на рис. 1.5,а показан суграф графа (рис. 1.4) без кратных ребер и петель.



**Рис. 1.5. Примеры деления графа:**  
 а – суграф; б – подграф; в – частичный граф

*Подграфом* графа  $G$  называется граф  $G'=(V', \bar{U}')$ , где  $V' \subset V$ ,  $\bar{U}' = \bar{U} \cap (V' \times V')$ , т. е. подграф получается из исходного графа удалением некоторого числа вершин вместе со всеми ребрами, инцидентными удаленной вершине. Например, на рис. 1.5,б показан подграф графа (рис. 1.4) после удаления вершины  $v_3$  и инцидентных ей ребер и петель ( $\bar{u}_3, \bar{u}_4, \bar{u}_5$ ).

*Частью* (частичным графом) графа  $G$  называется граф  $G'=(V', \bar{U}')$ , где  $V' \subset V$ ,  $\bar{U}' \subset \bar{U}$ , т. е. часть графа получается из исходного графа применением обеих описанных операций. Например, на рис. 1.5,в показана часть графа (рис. 1.4) без кратных ребер и петель ( $\bar{u}_2, \bar{u}_4, \bar{u}_5$ ), без изолированной вершины  $v_4$ .

Исходный граф  $G$  называется *надграфом* для всех его частей.

### 1.3.5. Динамические свойства неориентированного графа

До сих пор мы рассматривали статические структурные свойства графа и отражающие их понятия. Однако граф обладает и динамическими свойствами, которые позволяют перемещаться по смежным ребрам от одной вершины к другой. Для изучения этих свойств, которые играют фундаментальную роль в теории графов, введем в рассмотрение соответствующие понятия.

*Маршрутом* называется конечная последовательность  $n$  ребер графа  $\bar{u}_1, \bar{u}_2, \dots, \bar{u}_n$  (не обязательно различных), если существует последовательность  $v_0, v_1, \dots, v_n$  из  $n+1$  (не обязательно различных) вершин таких, что

$$\bar{u}_i \sim [v_{i-1}, v_i], \text{ для } i=1, 2, \dots, n.$$

Длиной маршрута ( $n$ ) называется число ребер, которые он содержит.

Говорят, что маршрут замкнут, если  $v_0=v_n$ , и не замкнут, если  $v_0\neq v_n$ . В последнем случае можно сказать, что маршрут соединяет вершины  $v_0$  и  $v_n$ . Заметим, что одно ребро можно рассматривать как маршрут длины 1.

Для графа на рис. 1.6 последовательность ребер  $\bar{u}_1, \bar{u}_5, \bar{u}_2, \bar{u}_3, \bar{u}_4, \bar{u}_6$  образует незамкнутый маршрут длины 6, соединяющий вершины  $v_1$  и  $v_5$ . Ему соответствует последовательность вершин  $v_1, v_2, v_2, v_3, v_4, v_2, v_5$ . Если заменить в маршруте ребро  $\bar{u}_6$  на  $\bar{u}_1$ , то получим пример замкнутого маршрута с вершинами

$v_1, v_2, v_2, v_3, v_4, v_2, v_1$ .

Цепью называется незамкнутый маршрут, если все его ребра различны. Например, для графа на рис.1.6 можно выделить множество ребер  $\bar{u}_1, \bar{u}_5, \bar{u}_4, \bar{u}_3, \bar{u}_2$ , образующих цепь с последовательностью вершин  $v_1, v_2, v_2, v_4, v_3, v_2$ .

Циклом называется цепь, которая начинается и заканчивается в одной и той же вершине ( $v_0=v_n$ ). Например, для графа рис.1.6 цикл образует последовательность ребер  $\bar{u}_2, \bar{u}_3, \bar{u}_4, \bar{u}_5$  с последовательностью вершин  $v_2, v_3, v_4, v_2, v_2$ .

Простой цепью называется цепь, в которой все  $n+1$  вершин  $v_0, v_1, \dots, v_n$  различны (очевидно, что в этом случае ребра обязательно различны). Например, для графа на рис.1.6 простую цепь составляют ребра  $\bar{u}_6, \bar{u}_2, \bar{u}_3$  с вершинами  $v_5, v_2, v_3, v_4$ .

Простым циклом называется цикл, если  $v_0=v_n$ , но все остальные вершины различны. Например, для графа простой цикл образуют ребра  $\bar{u}_2, \bar{u}_3, \bar{u}_4$  с последовательностью вершин  $v_2, v_3, v_4, v_2$ .

### 1.3.6. Связность неориентированного графа

Граф  $G=(V, \bar{U})$  называется связным, если для всех  $v_i$  и  $v_j \in V$  ( $v_i \neq v_j$ ) всегда существует цепь из  $v_i$  в  $v_j$ , т. е. если каждая пара различных вершин может быть соединена, по крайней мере, одной цепью. В противном случае граф называется несвязным. Например, граф рис.1.6 является связным, а графы рис. 1.4 и 1.7 – несвязными.

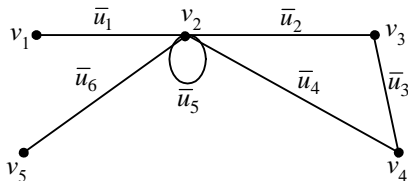
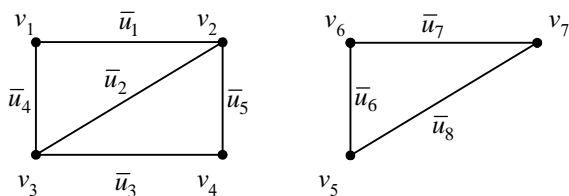


Рис. 1. 6. Пример графа



**Рис. 1.7. Пример несвязного графа**

Для вершины  $v_i$  обозначим через  $C(v_i)$  совокупность вершин графа, которые можно соединить цепью с  $v_i$ . *Компонентой связности*  $G(v_i)$  называется подграф графа  $G$ , порожденный  $C(v_i)$ . Например, граф  $G=(V, \bar{U})$  на рис. 1.7 имеет две компоненты связности с подмножествами вершин  $V_1=\{v_1, v_2, v_3, v_4\}$ ,  $V_2=\{v_5, v_6, v_7\}$ .

Пусть  $G_1, G_2, \dots$  – компоненты связности, порожденные подмножествами вершин  $V_1, V_2, \dots$ . Тогда

1.  $(\forall i \forall j) V_i \neq V_j$ , т. е. нет одинаковых подмножеств вершин.
2.  $(\forall i \forall j) V_i \neq V_j \Rightarrow V_i \cap V_j = \emptyset$ , т. е. подмножества вершин не содержат общих вершин.
3.  $\bigcup_i V_i = V$ , т. е. объединение подмножеств вершин составляет множество вершин графа, а также  $\bigcup C(v_i) = V$ .

$$v_i \in V.$$

На основе понятия компонент связности можно дать другое определение связности графа:

Граф  $G=(V, \bar{U})$  связан тогда и только тогда, когда множество его вершин нельзя разбить на два непустых подмножества  $V_1$  и  $V_2$  так, что обе граничные точки каждого ребра находятся в одном и том же подмножестве.

Пусть  $G=(V, \bar{U})$  – произвольный граф. Зададим бинарное отношение  $\rho$  между парами вершин следующим образом:  $v_1 \rho v_2$  тогда и только тогда, когда  $v_1 = v_2$  или когда существует цепь, соединяющая  $v_1$  с  $v_2$ . Отношение  $\rho$  *рефлексивно* ( $v \rho v$  для любого  $v$ ), *симметрично* (из  $v_1 \rho v_2$  следует  $v_2 \rho v_1$ ) и *транзитивно* (из  $v_1 \rho v_2$  и  $v_2 \rho v_3$  следует  $v_1 \rho v_3$ ). Таким образом,  $\rho$  есть отношение *эквивалентности*. Оно разбивает множество  $V$  единственным образом на классы эквивалентности взаимно связанных вершин. Для графа на рис. 1.7 такими клас-



сами эквивалентности являются  $V_1 = \{v_1, v_2, v_3, v_4\}$  и  $V_2 = \{v_5, v_6, v_7\}$ . Каждый класс эквивалентности вершин вместе с ребрами из  $\bar{U}$ , инцидентными этим вершинам, образует связный подграф, называемый просто компонентой связности графа  $G$ . Компонента  $G_i$  графа  $G$  является максимально связным подграфом в том смысле, что граф  $G_i$  не имеет связного собственного надграфа. Отсюда следует еще одно определение компоненты связности:

Подграф  $G'$  графа  $G$  называется компонентой связности графа  $G$ , если 1)  $G'$  связан, 2)  $G'$  обладает свойством максимальности, т. е. если  $G''$  — некоторый другой связный подграф  $G$  и  $G' \subset G''$ , то графы  $G'$  и  $G''$  совпадают.

### 1.3.7. Деревья и леса в графе

*Деревом* называется связный граф без циклов. Граф является деревом тогда и только тогда, когда каждая пара различных вершин соединяется одной и только одной цепью. Связность дерева означает существование, по крайней мере, одной такой цепи, а из отсутствия циклов следует существование единственной такой цепи.

*Лесом* из  $K$  деревьев называется граф без циклов, состоящий из  $K$  компонент.

Из графа на рис.1.7 путем удаления ребер, замыкающих циклы, можно получить различные варианты деревьев, например такие, как на рис.1.8.

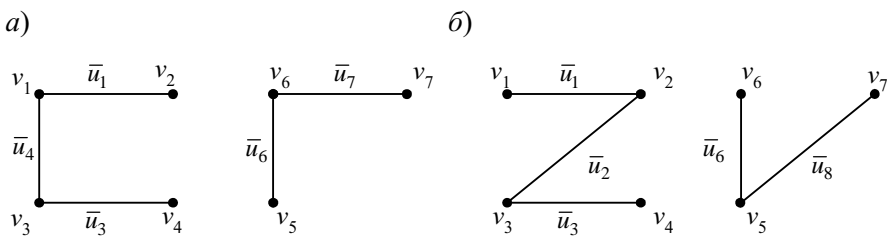


Рис. 1.8. Примеры несвязного графа в виде леса из двух деревьев

Удаление любого ребра из дерева делает его несвязным, так как удаляемое ребро составляет единственную цепь, соединяющую его граничные точки. Следовательно, дерево можно также определить как минимально связный граф, где минимальность понимается в том смысле, что

он не содержит подграфа, который состоит из всех его вершин и является связным.

Если дерево  $D$  является подграфом графа  $G$ , то ребра  $G$ , которые принадлежат дереву  $D$ , называются *ветвями* дерева  $D$ , а ребра, не принадлежащие дереву  $D$ , – *хордами* относительно дерева  $D$ . Если все вершины  $G$  принадлежат дереву  $D$ , то говорят, что дерево покрывает граф  $G$ . При этом только связные графы имеют покрывающие деревья и только деревья имеют единственные покрывающие деревья.

Рис. 1.8 иллюстрирует общее свойство деревьев: каждое дерево с  $n$  вершинами имеет в точности  $n - 1$  ребро. Применив этот результат к каждому дереву леса, можно сказать: лес из  $K$  деревьев, содержащий  $n$  вершин, имеет в точности  $n - k$  ребер.

### 1.3.8. Специальные классы неориентированных графов

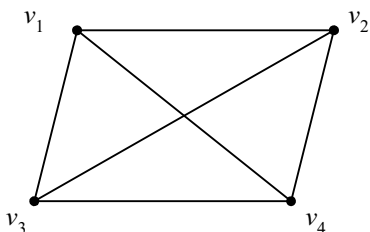
Классификация графов зависит от структурных признаков, которые используются в качестве основы для классификации, например, графы можно разбить на связные и несвязные. Возможны и другие разновидности графов.

*Обыкновенным* называется граф, который не содержит петель и параллельных ребер. Во многих случаях достаточно рассматривать только обыкновенные графы. Например, связность графа (или отсутствие ее) не меняется, если удалить все петли и параллельные ребра. Обыкновенный граф можно также определить как граф, не имеющий циклов, которые содержат менее трех ребер, так как цикл из двух ребер образуется параллельными ребрами.

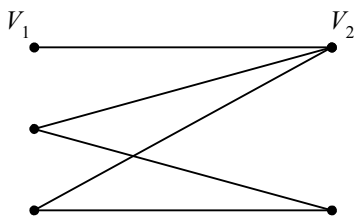
*Полным* называется граф, в котором любые две различные вершины являются смежными, т. е. соединяются ребром. Обычно этот термин применяется к обыкновенным графам, для которых существует только один полный граф с фиксированным числом вершин. Следовательно, выражение “полный граф с  $k$  вершинами” однозначно определяет граф. На рис. 1.9 изображен полный граф с четырьмя вершинами.

Если степень вершины  $\delta(v)=k$  (число ребер, инцидентных вершине, включая параллельные ребра и петли) для всех вершин графа, то граф называется однородным графом степени  $k$  или просто  $k$ -однородным. Заметим, что обыкновенный полный граф, имеющий  $k$  вершин, является  $(k-1)$ -однородным. Например, граф на рис. 1.9 является 3-однородным.

Граф называется  $k$ -связным, если любая пара различных вершин  $v$  и  $w$  соединена, по крайней мере,  $k$  цепями, которые не имеют общих вер-



**Рис. 1.9. Пример обыкновенного полного графа**



**Рис. 1.10. Двудольный граф**

шин, исключая, конечно,  $v$  и  $w$ . Например, любой простой цикл (исключая петли) образуют 2-связный граф. На рис.1.9 изображен 4-связный граф. При  $k=1$  это понятие совпадает с понятием обычной связности. Например, дерево – это односвязный граф.

*Двудольным* называется граф, в котором вершины могут быть разбиты на два непересекающихся подмножества  $V_1$  и  $V_2$  так, что каждое ребро имеет одну граничную точку в  $V_1$ , а другую в  $V_2$ . В общем случае граф называется  $k$ -дольным, если множество его вершин можно разбить на  $k$  непересекающихся подмножеств  $\{V_1, V_2, \dots, V_k\}$  так, что не существует ребер, соединяющих вершины одного и того же подмножества. Двудольный граф изображают, размещая множества вершин  $V_1$  и  $V_2$  в разных столбцах (или строках), как показано на рис. 1.10.

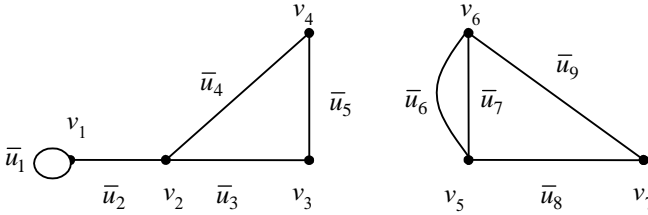
### 1.3.9. Матричное представление графов

#### *Матрица инциденций*

Для удобства обработки графов с использованием ЭВМ лучше всего подходит матричная форма представления графа. Пусть  $G$  – граф, имеющий  $n$  вершин и  $m$  ребер. Граф  $G$  можно описать матрицей инциденций  $A$  размером  $n \times m$ , строки и столбцы которой соответствуют вершинам и ребрам графа. Элемент матрицы  $a_{ij}$  принимает значение 1 или 0 в зависимости от того, инцидентно  $j$ -е ребро  $i$ -й вершине или нет. Для петли все элементы столбца считаются равными 0. Матрицы, содержащие только элементы 1 или 0, называются *булевыми матрицами*. На рис. 1.11 представлен двухкомпонентный граф и его матрица инциденций.

Некоторые свойства графа проявляются в его матрице инциденций. Например, так как ребро графа инцидентно точно двум вершинам, то каждый

а)



б)

$$A_{7 \times 9} = \begin{array}{c|cccccccc} & \bar{u}_1 & \bar{u}_2 & \bar{u}_3 & \bar{u}_4 & \bar{u}_5 & \bar{u}_6 & \bar{u}_7 & \bar{u}_8 & \bar{u}_9 \\ \hline v_1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_2 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ v_3 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ v_4 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline v_5 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ v_6 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ v_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

**Рис.1.11. Матричное представление графа**  
**а – граф; б – матрица инцидентий**

столбец матрицы инцидентий содержит ровно две 1, причем для кратных ребер в одинаковых строках. Исключение составляет петля, так как она дважды инцидентна одной и той же вершине. Столбец, соответствующий петле, состоит из нулей. Таким образом, матрица инцидентий не указывает на существование петель для конкретной вершины, поэтому при изучении графов с помощью таких матриц желательно исключать петли.

При соответствующей нумерации ребер и вершин графа каждая его компонента соответствует подматрице матрицы инцидентий, которая в этом случае имеет блочную структуру следующего вида:

$$A = \begin{pmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & A_n \end{pmatrix},$$

где  $A_i$  – матрица инцидентий, соответствующая  $i$ -й компоненте графа. Блочно-диагональное представление графа можно получить последова-

тельной нумерацией ребер и вершин внутри каждой компоненты и между компонентами, как показано в примере, или непосредственно с помощью подстановки строк и столбцов матрицы инциденций.

Можно утверждать, что два графа изоморфны, если они имеют одни и те же матрицы инциденций с точностью до перестановок строк и столбцов. Таким образом, матрица инциденций обеспечивает полное описание графа (если петли исключены).

### *Матрица смежности вершин*

Для неориентированных графов можно определить матрицу смежности вершин (или просто матрицу смежности). Это квадратная матрица  $S$ -размера  $n \times n$ , где строки и столбцы соответствуют вершинам графа, а элемент  $s_{ij}$  равен числу ребер, инцидентных одновременно  $i$ -й и  $j$ -й вершинам. Таким образом, для графа на рис. 1.11 имеем матрицу смежности

$$S_{7 \times 7} = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{matrix} .$$

Матрица смежности неориентированного графа (без петель) симметрична относительно главной диагонали, поскольку паре вершин  $(v_i, v_j)$  инцидентны одинаковые ребра  $[v_i, v_j]=[v_j, v_i]$ , т. е. достаточно рассматривать только половину матрицы выше главной диагонали. Очевидно, что матрица смежности обыкновенного графа (без петель и кратных ребер) является булевой матрицей.

### **1.3.10. Изоморфизм графов. Плоский граф**

Вернемся к понятию геометрического графа, расширив его. Обозначим  $n$ -мерное евклидово пространство  $\mathcal{E}^n$ . Евклидово  $n$ -мерное пространство есть множество последовательностей из  $n$  действительных чисел  $x = (x_1, \dots, x_n)$ , в котором расстояние между любыми двумя точками  $x = (x_1, \dots, x_n)$  и  $y = (y_1, \dots, y_n)$  определено следующим образом:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} .$$

Простой незамкнутой кривой в пространстве  $\mathcal{E}^n$  называется непрерывная, самонепересекающаяся кривая, соединяющая две различные точки в  $\mathcal{E}^n$  (т. е. кривая, полученная непрерывной деформацией прямолинейного отрезка). Аналогично *простой замкнутой кривой* называется непрерывная самонепересекающаяся кривая, конечные точки которой совпадают.

Геометрический граф в пространстве  $\mathcal{E}^n$  есть множество  $V = \{v_i\}$  точек пространства  $\mathcal{E}^n$  и множество  $U = \{u_i\}$  простых кривых, удовлетворяющих следующим условиям:

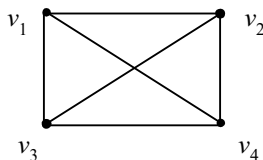
1. Каждая замкнутая кривая в  $U$  содержит только одну точку  $v \in V$ .
2. Каждая незамкнутая кривая в  $U$  содержит ровно две точки множества  $V$ , которые являются ее граничными точками.
3. Кривые в  $U$  не имеют общих точек, за исключением точек из множества  $V$ .

Ранее было отмечено, что любой граф в абстрактном смысле идентичен, или, используя более принятый термин, изоморфен некоторому геометрическому графу.

Говорят, что графы  $G = (V, \bar{U})$  и  $G' = (V', \bar{U}')$  изоморфны друг другу, если существует взаимно однозначное соотношение между  $V$  и  $V'$  и между  $\bar{U}$  и  $\bar{U}'$ , сохраняющее отношение инцидентности. Иначе говоря, ребро  $\bar{u}$  инцидентно вершине  $v$  графа  $G$  тогда и только тогда, когда инцидентны соответствующие элементы  $\bar{u}'$  и  $v'$  в графе  $G'$ . Если граф  $G$  изоморфен геометрическому графу  $G'$ , то  $G'$  называется геометрической реализацией графа  $G$ .

Граф называется *плоским* тогда и только тогда, когда он имеет геометрическую реализацию в пространстве  $\mathcal{E}^2$ , т. е. на плоскости. Например, граф  $G$  на рис. 1.12,а – плоский, так как он изоморфен графу  $G'$ , изображенному рядом.

а)  $G$ :



б)  $G'$ :

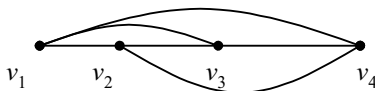


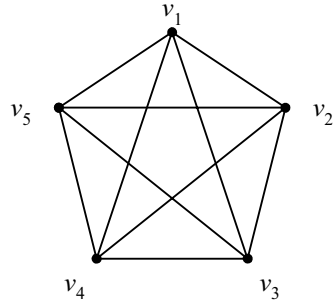
Рис. 1.12. Плоский граф

Рис. 1.12 иллюстрирует очевидный, но важный факт: геометрический граф может быть плоским, даже если его нельзя преобразовать в плоский граф с помощью непрерывной деформации. Хотя  $G$  и  $G'$  имеют существенные отличия с точки зрения топологии, с точки зрения теории графов они эквивалентны.

Обыкновенный полный граф, который имеет наименьшее число вершин и не является плоским, есть полный граф из пяти вершин (рис. 1.13).

Если в пространстве  $\mathbb{E}^2$  только ограниченный класс конечных графов имеет геометрическую реализацию, то для пространства  $\mathbb{E}^3$  справедливо следующее утверждение: *любой конечный граф  $G$  имеет геометрическую реализацию в  $\mathbb{E}^3$ .*

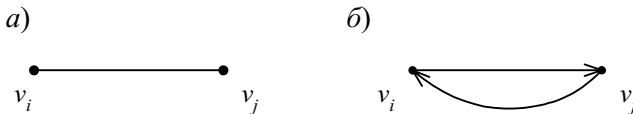
Очевидно, что если любой граф содержит такой пятивершинный граф (или, вообще, любой неплоский граф) в качестве подграфа, то он обязательно неплоский.



**Рис. 1.13. Неплоский граф**

### 1.4. Основные понятия для ориентированных графов

Во многих случаях при описании графа линиям, соединяющим вершины графа, необходимо задать ориентацию (направление), и тогда линии называются дугами (стрелками), а граф – ориентированным (орграфом). Структурное различие между неориентированным графом и орграфом состоит в том, что в первом случае граничные точки ребра образуют неупорядоченную, а во втором – упорядоченную пару вершин. На рис. 1.14 показано, что ребро эквивалентно противоположно направленным дугам.



**Рис. 1.14. Ребро и дуги графа**

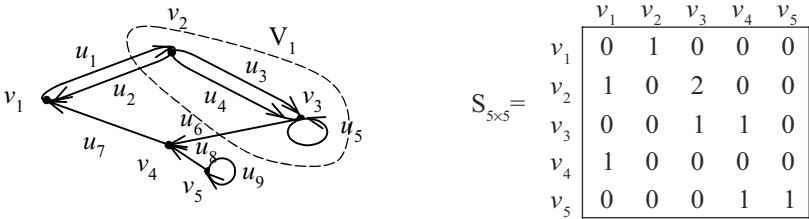
Необходимость введения ориентации линий графа возникает по двум причинам. Во-первых, дуга представляет отношение между парой не-

симметричных вершин. Например, в системе городских улиц необходимо изобразить улицы с односторонним движением. Во-вторых, введение ориентации необходимо для установления системы координат и устранения неоднозначностей. Например, при соединении электрических приборов одно направление необходимо обозначить как “положительное” для того, чтобы однозначно описать распределение электрического тока.

Представим формальное описание орграфа, опираясь на структурные термины, введенные для неориентированных графов.

Орграф  $G$  можно определить как совокупность  $G=(V, U)$ , где  $V=\{v_1, \dots, v_n\}$  – множество вершин графа, а  $U \subset V \times V$  – конечное подмножество упорядоченного (декартова) произведения  $V \times V$ , образующее множество дуг графа.

Дугу (стрелку), соединяющую вершины  $v_i, v_j$ , обозначают  $(v_i, v_j)$ , где  $v_i$  – начало дуги, а  $v_j$  – конец дуги. Говорят, что дуга направлена от вершины  $v_i$  к вершине  $v_j$ , или, что дуга исходит из вершины  $v_i$  и входит в вершину  $v_j$ . Для удобства дуги также обозначают одной буквой:  $u_{ij}=(v_i, v_j)$ ,  $u_k=(v_i, v_j)$  или  $u=(v_i, v_j)$ . Пример орграфа приведен на рис. 1.15.



**Рис. 1.15. Ориентированный граф и его матрица смежности**

Орграф на рис. 1.15 состоит из множества вершин  $V=\{v_1, v_2, v_3, v_4, v_5\}$  и множества дуг  $U=\{u_1=(v_1, v_2), u_2=(v_2, v_1), u_3=(v_2, v_3), u_4=(v_2, v_3), u_5=(v_3, v_3), u_6=(v_3, v_4), u_7=(v_4, v_1), u_8=(v_5, v_4), u_9=(v_5, v_5)\}$ .

*Матрица смежности вершин орграфа*

Орграф можно описать квадратной матрицей смежности  $S_{n \times n}$ , где  $n = |S|$  – количество вершин графа, а элемент  $s_{ij}$  равен количеству дуг  $(v_i, v_j)$ , т. е. стрелок, идущих от вершины  $v_i$  к  $v_j$  (рис. 1.15).

*Смежными дугами* называются дуги, имеющие общую граничную точку, например, дуги  $u_1, u_2, u_3, u_4$  смежные относительно вершины  $v_2$  (рис. 1.15).



*Смежными вершинами* называются две различные вершины  $v_i, v_j$ , если существует хотя бы одна дуга, идущая от одной из них к другой, например, вершины  $v_1$  и  $v_2$ ,  $v_1$  и  $v_4$  смежны, а  $v_1$  и  $v_3$  не смежны (рис. 1.15).

*Петлей* называется дуга  $(v_i, v_i)$ , где начало и конец дуги совпадают, например, петли  $u_5=(v_3, v_3)$  и  $u_9=(v_5, v_5)$  (рис. 1.15).

*Кратными* называются дуги, граничные точки которых совпадают. При этом если  $a_1=(v, w)$  и  $b_2=(v, w)$ , то дуги  $a_1$  и  $b_2$  называются *параллельными*, а если  $c_3=(w, v)$ , то дуги  $a_1$  и  $c_3$  будем называть *встречными*. Например, у графа на рис. 1.15 есть кратные параллельные дуги  $u_3$  и  $u_4$  и встречные дуги  $u_1$  и  $u_2$ . Граф без параллельных дуг описывается булевой матрицей смежности.

*Мультиграфом* называется граф с кратными дугами, т. е. граф на рис. 1.15 является мультиграфом.

Дуга и ее граничные вершины инцидентны друг другу. Дуга  $u=(v, w)$  считается положительно инцидентной ее конечной вершине  $w$ , говорят, что дуга (стрелка) заходит в вершину  $w$ . Дуга  $u$  считается отрицательно инцидентной ее начальной вершине  $v$ , говорят, что дуга исходит из этой вершины.

Число дуг, заходящих в вершину  $v$ , называется положительной степенью  $v$  и обозначается через  $\delta^+(v)$ . Отрицательная степень  $v$  определяется аналогично для исходящих дуг и обозначается  $\delta^-(v)$ . Ориентированная петля, инцидентная  $v$ , считается положительно и отрицательно инцидентной с  $v$ . Степень вершины равна  $\delta(v)=\delta^+(v)+\delta^-(v)$ .

Учитывая, что каждая дуга положительно инцидентна вершине и отрицательно инцидентна также одной вершине, получим

$$\sum_{v \in V} \delta^+(v) = \sum_{v \in V} \delta^-(v) = |U|,$$

где  $|U|$  означает число дуг графа. Выше было показано для неориентированного графа

$$\sum_{v \in V} \delta(v) = 2|\bar{U}|.$$

#### 1.4.1. Дуги, инцидентные подмножеству вершин

Пусть задан граф  $G=(V, U)$  и непустое подмножество  $V_1$  множества  $V$  ( $V_1 \subset V$ ). Говорят, что дуга  $u=(v_i, v_j)$  заходит в  $V_1$ , если  $v_i \notin V_1$ , а  $v_j \in V_1$

(конец дуги  $v_j$  находится в  $V_1$ ). Если же  $v_i \in V_1$ , а  $v_j \notin V_1$ , то говорят, что дуга  $u$  исходит из  $V_1$  (начало дуги  $v_i$  находится в  $V_1$ ). В обоих случаях дугу  $u$  называют инцидентной подмножеству  $V_1$ . Обозначим через  $U_{V_1}^+$  и  $U_{V_1}^-$  множества дуг, заходящих в  $V_1$  и исходящих из него. Например, для графа на рис.1.15

$$U_{V_1}^+ = \{u_1=(v_1, v_2)\}, \quad U_{V_1}^- = \{u_2=(v_2, v_1), u_6=(v_3, v_4)\}.$$

Если подмножество сводится к одной вершине, то дугу называют *инцидентной* этой вершине (заходящей в нее или исходящей из нее).

Обозначим через  $\Gamma v$  подмножество вершин  $V$ , смежных с  $v$ , в которые можно прийти по инцидентным дугам (стрелки исходят из  $v$ ) или иначе подмножество концов дуг, имеющих началом вершину  $v$ . Например, для графа на рис. 1.15

$$\Gamma v_1 = \{v_2\}, \Gamma v_2 = \{v_1, v_3\}, \Gamma v_3 = \{v_3, v_4\}, \Gamma v_4 = \{v_1\}, \Gamma v_5 = \{v_4, v_5\}.$$

По матрице смежности  $S$  можно выделить подмножество  $\Gamma v_i$  по  $i$ -й строке, где ненулевые элементы  $s_{ij}$  указывают на вершины  $v_j \in \Gamma v_i$ . Это отображение *инцидентности и смежности* позволяет определить граф как пару  $G=(V, \Gamma)$ , образованную множеством вершин  $V$  и многозначным отображением  $\Gamma$  множества  $V$  в себя. Элемент  $v_i \in V$  называют вершиной, а пару  $(v_i, v_j)$ , где  $v_j \in \Gamma v_i$ , дугой. Граф  $G$  имеет порядок  $n$ , если число вершин  $|V|=n$ .

Отображение  $\Gamma$  можно определить для любого подмножества из  $V$ , например для графа на рис. 1.15

$$\Gamma\{v_1, v_2\} = \Gamma v_1 \cup \Gamma v_2 = \{v_2\} \cup \{v_1, v_3\} = \{v_1, v_2, v_3\}.$$

Определим граф  $G^{-1}$  с помощью обратного отображения  $\Gamma^{-1}$ :

$$G^{-1} = (V, \Gamma^{-1}).$$

Соответствующие представления графа  $G^{-1}$  получаются, если  $\Gamma^{-1}v$  рассматривать как подмножества вершин, смежных с  $v$ , из которых можно зайти в вершину  $v$  по инцидентным дугам, при этом матрица смежности графа  $G^{-1}$  образуется транспонированием (перестановкой строк и столбцов) матрицы смежности графа  $G$ . Например, для графа на рис.1.15

$$\Gamma^{-1}v_1=\{v_2, v_4\}, \Gamma^{-1}v_2=\{v_1\}, \Gamma^{-1}v_3=\{v_2, v_3\}, \Gamma^{-1}v_4=\{v_3, v_5\}, \Gamma^{-1}v_5=\{v_5\}.$$

По транспонированной матрице смежности  $S^T$  можно выделить  $\Gamma^{-1}v_i$  по  $i$ -й строке, где ненулевые элементы  $s^T_{ij}$  указывают на вершины  $v_j \in \Gamma^{-1}v_i$ :

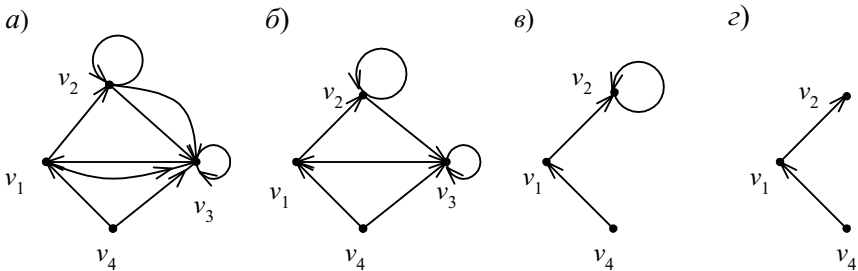
$$S_{5 \times 5}^T = \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \\ \hline 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{array} .$$

### 1.4.2. Разделение орграфа на составляющие

*Суграфом*  $G'=(V, \Gamma')$  графа  $G=(V, \Gamma)$  называется граф, если  $(\forall v_i \in V) (\Gamma'v_i \subseteq \Gamma v_i)$ , т. е. из исходного графа удаляется некоторое количество дуг (с сохранением вершин) (рис.1.16, а,б).

*Подграфом*  $G'=(V', \Gamma')$  графа  $G=(V, \Gamma)$  называется граф, если  $V' \subset V$  и  $(\forall v_i \in V) (\Gamma'v_i = V' \cap \Gamma v_i)$ , т. е. из исходного графа удаляются некоторые вершины вместе со всеми дугами, инцидентными удаленной вершине (рис.1.16, а,в).

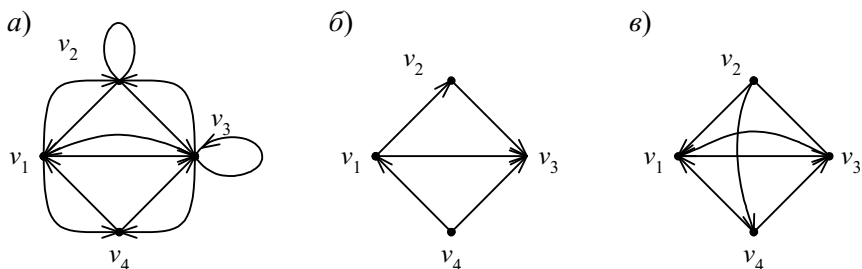
*Частичным подграфом*  $G'=(V', \Gamma')$  графа  $G=(V, \Gamma)$  называется граф, если  $V' \subset V, \Gamma' \subset \Gamma$ , т. е. к исходному графу применены обе описанные выше операции (рис.1.16, а, з).



**Рис. 1.16. Разделение орграфа на подчасти:**  
а – орграф ; б – суграф; в – подграф; з – частичный граф

### 1.4.3. Некоторые специальные классы орграфов

*Симметрическим* называется граф  $G=(V, U)$ , в котором  $(\forall v_i \in V, \forall v_j \in V) (v_i, v_j) \in U \Rightarrow (v_j, v_i) \in U$  ( т. е. любые две смежные вершины  $v_i$  и  $v_j$  со-



**Рис. 1.17. Разновидности орграфов:**  
**а – симметрический; б – антисимметрический; в – полный**

единены двумя встречными, т. е. противоположно направленными дугами) (рис. 1.17,а).

*Антисимметрическим* называется граф  $G=(V,U)$ , в котором  $(\forall v_i \in V, \forall v_j \in V) (v_i, v_j) \in U \Rightarrow (v_j, v_i) \notin U$  (пара смежных вершин может быть соединена лишь в одном направлении, петли отсутствуют) (рис.1.17,б).

*Полным* называется граф  $G=(V, U)$ , в котором  $(\forall v_i \in V, \forall v_j \in V) (v_i, v_j) \notin U \Rightarrow (v_j, v_i) \in U$  ( $i \neq j$ ) (любые две вершины соединены по крайней мере одной дугой, т. е. смежны) (рис. 1.17, в).

*Рефлексивным* будем называть граф, имеющий петлю в каждой вершине.

*Транзитивным* называется граф, в котором из наличия дуг  $u_1=(v, v')$  и  $u_2=(v', v'')$  следует существование дуги  $u_3=(v, v'')$ . Таким образом в транзитивном графе существование любого пути из вершины  $v$  в вершину  $w$  означает существование дуги из  $v$  в  $w$ .

*Полным графом с петлями* называется граф  $G=(V, \Gamma)$ , если  $(\forall v_i \in V) \Gamma_{v_i}=V$  (каждая пара вершин, различных или нет, соединяется дугой, т. е. это *симметрический, рефлексивный и транзитивный* граф). Очевидно, что каждому графу можно сопоставить полный граф с петлями.

#### 1.4.4. Перемещения в орграфе

Перемещения по дугам орграфа обусловлены их ориентацией.

Путем длины  $n$  является последовательность (не обязательно различных) дуг  $(u_1, u_2, \dots, u_n)$  таких, что для соответствующей последовательности  $n+1$  вершин  $v_0, v_1, \dots, v_n$  справедливо  $u_i=(v_{i-1}, v_i)$  для  $i=1, 2, \dots, n$  (т. е. конец предыдущей дуги совпадает с началом следующей). Путь можно задать последовательностью дуг или вершин,

через которые он проходит. Например, на рис.1.18 последовательность дуг  $(u_1, u_5, u_5, u_3)$  образует путь с соответствующей последовательностью вершин  $(v_3, v_2, v_2, v_2, v_1)$ .

Путь называется *замкнутым*, если  $v_0=v_n$ , и *незамкнутым*, если  $v_0 \neq v_n$ . В последнем случае говорят, что путь соединяет  $v_0$  и  $v_n$  или, точнее, что он идет из  $v_0$  в  $v_n$ .

*Простым* путем называется путь, в котором нет повторяющихся дуг, например путь  $(u_1, u_5, u_3, u_6)$  на рис.1.18. с последовательностью вершин  $(v_3, v_2, v_2, v_1, v_4)$ .

*Элементарным* путем называется путь, если все его вершины различны, например элементарный путь  $(u_1, u_3, u_6)$  с вершинами  $(v_3, v_2, v_1, v_4)$  на рис. 1.18.

*Контуром* называется замкнутый путь, когда  $v_0=v_n$ , например контур  $(u_7, u_2, u_4, u_2, u_6)$  с вершинами  $(v_4, v_3, v_1, v_3, v_1, v_4)$  на рис.1.18. *Простым* контуром называется контур, в котором все дуги различны, например контур  $(u_1, u_5, u_3, u_4)$  с вершинами  $(v_3, v_2, v_2, v_1, v_3)$ . Элементарным контуром называется контур с различными вершинами (кроме начальной и конечной, которые совпадают), например, контур  $(u_1, u_3, u_4)$  с вершинами  $(v_3, v_2, v_1, v_3)$  на рис.1.18.

Длиной пути  $\mu = (u_1, u_2, \dots, u_n)$  называют число его дуг и обозначают через  $L(\mu)$ , например на рис.1.18

$$\mu=(u_1, u_5, u_3, u_6); L(\mu)=4;$$

$$\mu=(u_1, u_3, u_4); L(\mu)=3.$$

Для удобства вводят понятие пути нулевой длины (изолированная вершина).

Орграф называется *циклическим* (контурным), если он содержит, по крайней мере, один контур, и *ациклическим* (бесконтурным) в противном случае. Заметим, что петля является специальным видом контура.

*Гамильтонов путь* – это элементарный путь, в котором число дуг на единицу меньше числа вершин графа  $|V|$ . Иначе говоря, такой путь проходит через все вершины в точности по одному разу. При записи с помощью вершин гамильтонов путь – это перестановка вершин графа. Например, в графе рис 1.18 есть два гамильтоновых пути:

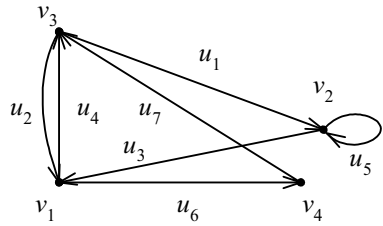


Рис.1.18. Орграф

- 1) путь  $(u_1, u_3, u_6)$  с последовательностью вершин  $(v_3, v_2, v_1, v_4)$ ;
- 2) путь  $(u_6, u_7, u_1)$  с последовательностью вершин  $(v_1, v_4, v_3, v_2)$ .

Аналогично *гамильтонов контур* – это контур, проходящий через все вершины в точности по одному разу (исключая произвольно выбранное начало). Например, на рис.1.18 есть один гамильтонов контур  $(u_1, u_3, u_6, u_7)$  с последовательностью вершин  $(v_3, v_2, v_1, v_4, v_3)$ ; причем любая вершина может быть выбрана в качестве начальной циклическим сдвигом.

### 1.4.5. Ориентированные деревья

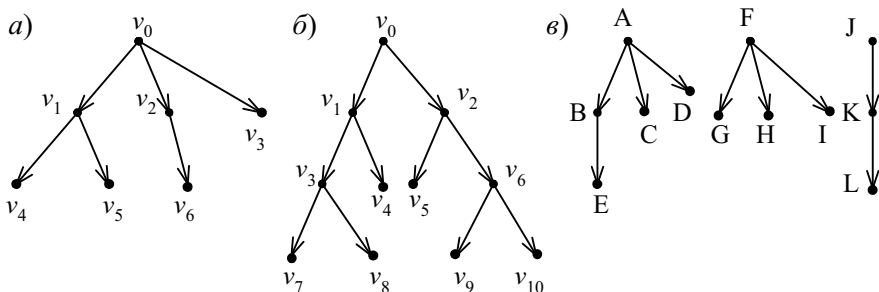
Ориентированным деревом (его называют также прадеревом), растущим из корня  $v_0$ , называется граф  $G=(V,U)$ , если:

- 1)  $(\exists! v_0 \in V) \Gamma^{-1}\{v_0\} = \emptyset$  (где символьное обозначение  $\exists!$  означает – “существует одна и только одна”), т. е. существует единственная вершина  $v_0$  (называемая корнем дерева), в которую не заходит ни одна дуга (нет предшествующих вершин);
- 2)  $(\forall v_i \in V) |\Gamma^{-1}\{v_i\}| = 1$  ( $v_i \neq v_0$ ), т. е. в каждую другую вершину заходит только одна дуга (только одна вершина предшествует вершине  $v_i$ );
- 3) граф не содержит контуров.

Вершина  $v_i$  называется *висячей*, если  $\Gamma v_i = \emptyset$ , т. е. из  $v_i$  не исходит дуга (нет следующей вершины), ее также называют листом.

На рис.1.19,а показано прадерево с корнем  $v_0$  и висячими вершинами  $v_3, v_4, v_5, v_6$ .

Таким образом, можно сказать, что прадеревом является граф без контуров и петель, в котором из корня дерева в любую другую вершину можно прийти только по одному элементарному пути.



**Рис. 1.19. Ориентированные деревья :**  
**а – прадерево; б – двоичное дерево; в – ветвящийся граф**

Двоичным деревом называется прадеерево, если  $(\forall v_i \in V) |\Gamma \{v_i\}| = 2$  или 0, т. е. из каждой вершины исходит две дуги, если это не висячая вершина (рис. 1.19,б).

Ветвящимся называется граф, все связные компоненты которого являются прадееревами (рис 1.19,в).

#### 1.4.6. Орграфы и бинарные отношения

Пусть  $V$  – множество объектов, подлежащих исследованию путем их попарного (бинарного) сравнения. Множество всех упорядоченных пар  $v, v' \in V$  обозначается  $V \times V$  (декартово произведение). Подмножество  $B \subseteq V \times V$  называется бинарным отношением, т. е.  $(v, v') \in B$ .

Бинарное отношение называется: рефлексивным, если  $(v, v') \in B$ ; антирефлексивным, если  $(v, v) \notin B$ ; симметричным, если  $(v, v') \in B \rightarrow (v', v) \in B$ ; антисимметричным, если  $(v, v') \in B$  и  $(v', v) \in B \rightarrow v = v'$ ; асимметричным, если  $(v, v') \in B \rightarrow (v', v) \notin B$ ; транзитивным, если  $(v, v') \in B$  и  $(v', v'') \in B \rightarrow (v, v'') \in B$ ; линейным, если  $(v, v') \in B$  или  $(v', v) \in B$ .

Бинарные отношения легко представляются в виде орграфа, при этом вершины графа соответствуют объектам сравнения  $v_i \in V$ , а дуги отображают бинарное отношение между объектами. Если  $(v, v') \in B$ , то проводится дуга из  $v$  в  $v'$ , при этом  $v$  – начало, а  $v'$  – конец дуги, т. е. дуга определяется парой  $(v, v')$ . Как показано выше, последовательность дуг  $(v_{i-1}, v_i)$ ,  $i=1, 2, \dots, n$ , составляет путь в графе. Конечный путь, у которого  $v_0 = v_n$ , образует контур. Дуга  $(v, v)$  называется петлей (отражает рефлексивное отношение).

В действительности, любое бинарное отношение  $B$ , определенное на множестве  $V$ , может быть представлено ориентированным графом, вершины которого соответствуют элементам  $V$ . Граф полностью описывает отношение перечислением всех связных элементов. Таким образом, орграф, в некотором смысле, является исчерпывающей формой представления отношения, т. е. он полностью перечисляет все пары, для которых рассматриваемое отношение имеет место.

Рассмотренные выше описания специальных классов орграфов, не имеющих параллельных ребер, аналогичны терминологии бинарных отношений. Действительно, рефлексивным называется граф, имеющий петлю  $(v, v)$  в каждой вершине, симметрическим является граф, в котором каждой дуге  $u=(v, w)$ , соответствует встречная дуга  $u'=(w, v)$ , транзитивным называется граф, в котором из существования дуг  $u_1=(v, v')$  и  $u_2=(v', v'')$  следует существование дуги  $u_3=(v, v'')$ .

### 1.4.7. Транзитивные замыкания

Для того чтобы дать формализованное определение сильно связного графа, играющего важную роль при решении многих задач методами теории графов, введем понятие транзитивного и обратного транзитивного замыкания.

Выше мы обозначили через  $\Gamma v$  подмножество смежных вершин-концов дуг, имеющих началом вершину  $v$ , а через  $\Gamma^{-}v$  подмножество смежных вершин-начала дуг, имеющих концом вершину  $v$ .

Обозначим через  $\Gamma^n v$  подмножество тех вершин, в которые можно прийти из вершины  $v$ , используя пути длины  $n$  (или, быть может, меньшей), а через  $\Gamma^{-n} v$  подмножество тех вершин, из которых можно прийти в вершину  $v$ , используя пути длины  $n$  (или, возможно, меньшей).

Например, для графа на рис.1.20,а имеем:

$$\Gamma v_1 = \{v_2, v_3\}, \Gamma v_2 = \{v_1, v_4, v_5\}, \Gamma v_3 = \{v_4\}, \Gamma v_4 = \{v_1\}, \Gamma v_5 = \{v_4\};$$

$$\Gamma^2 v_1 = \Gamma\{\Gamma v_1\} = \Gamma\{v_2, v_3\} = \Gamma v_2 \cup \Gamma v_3 = \{v_1, v_4, v_5\} \cup \{v_4\} = \{v_1, v_4, v_5\};$$

$$\Gamma^3 v_1 = \Gamma\{\Gamma^2 v_1\} = \Gamma\{v_1, v_4, v_5\} = \Gamma v_1 \cup \Gamma v_4 \cup \Gamma v_5 = \\ = \{v_2, v_3\} \cup \{v_1\} \cup \{v_4\} = \{v_1, v_2, v_3, v_4\};$$

$$\Gamma^{-1} v_1 = \{v_2, v_4\}; \Gamma^{-1} v_2 = \{v_1\}; \Gamma^{-1} v_3 = \{v_1\}; \Gamma^{-1} v_4 = \{v_2, v_3, v_5\}; \Gamma^{-1} v_5 = \{v_2\};$$

$$\Gamma^{-2} v_4 = \Gamma^{-}\{\Gamma^{-} v_4\} = \Gamma^{-}\{v_2, v_3, v_5\} = \Gamma^{-} v_2 \cup \Gamma^{-} v_3 \cup \Gamma^{-} v_5 = \\ = \{v_1\} \cup \{v_1\} \cup \{v_2\} = \{v_1, v_2\};$$

$$\Gamma^{-3} v_4 = \Gamma^{-}\{\Gamma^{-2} v_4\} = \Gamma^{-}\{v_1, v_2\} = \Gamma^{-} v_1 \cup \Gamma^{-} v_2 = \{v_2, v_4\} \cup \{v_1\} = \{v_1, v_2, v_4\}.$$

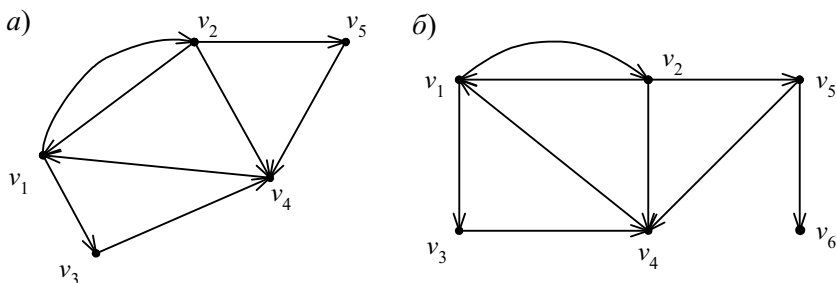


Рис. 1.20. Орграфы



*Транзитивным замыканием* называется многозначное отображение, определяемое формулой

$$\hat{\Gamma}v_i = \{v_i\} \cup \Gamma v_i \cup \Gamma^2 v_i \cup \dots,$$

т. е. транзитивное замыкание есть множество вершин, в которые можно прийти из вершины  $v_i$  по некоторому пути без повторения дуг.

Аналогично определяется обратное транзитивное замыкание:

$$\hat{\Gamma}^-v_i = \{v_i\} \cup \Gamma^{-1}v_i \cup \Gamma^{-2}v_i \cup \dots,$$

т. е. *обратное транзитивное замыкание* есть множество вершин, из которых попадают в вершину  $v_i$  по некоторому пути без повторения дуг.

Например, для графа на рис.1.20,*a*

$$\hat{\Gamma}v_1 = V, \hat{\Gamma}v_2 = V, \hat{\Gamma}v_3 = V, \hat{\Gamma}v_4 = V, \hat{\Gamma}v_5 = V,$$

аналогично

$$\hat{\Gamma}^-v_1 = V, \hat{\Gamma}^-v_2 = V, \hat{\Gamma}^-v_3 = V, \hat{\Gamma}^-v_4 = V, \hat{\Gamma}^-v_5 = V,$$

т. е. из каждой вершины можно прийти в любую другую вершину.

Для графа на рис. 1.20,*б*  $\Gamma v_6 = \{v_6\}$ , т. е. из вершины  $v_6$  нельзя попасть ни в одну из других вершин, тогда как  $\Gamma^-v_6 = V$ , т. е. в вершину  $v_6$  можно прийти из любых других вершин графа.

#### **1.4.8. Сильно связный орграф**

Сильно связным графом  $G = (V, \Gamma)$  называется граф, если

$$(\forall v_i \in V) \hat{\Gamma}v_i = V,$$

т. е. для любых двух вершин  $v_i, v_j$  существует путь, идущий из  $v_i$  в  $v_j$ . Например, граф на рис 1.20,*a* является сильно связанным, а его надграф на рис.1.20,*б* не является таковым. Сильно связный граф можно определить также с помощью условия

$$(\forall v_i \in V) \hat{\Gamma}^-v_i = V.$$

Бинарное отношение “существует путь из  $v_i$  в  $v_j$ ” является отношением частичного порядка на множестве вершин  $V$ , так как оно транзитивно и рефлексивно. Бинарное отношение “существует путь из  $v_i$  в  $v_j$  и путь из  $v_j$  в  $v_i$ ” задает отношение эквивалентности  $(R)$ , так как оно, кроме того, симметрично. Это бинарное отношение порождает контуры в графе. Значит, сильно связный граф – это граф, содержащий контуры.

## 1.5. Реализация алгоритмов теории графов на ЭВМ

Решение задач с применением методов теории графов должно подчиняться общей последовательности этапов решения задач с применением ЭВМ.

1-й этап – постановка задачи. Дается формулировка задачи, целей ее решения и ограничений на “содержательном” естественном языке.

2-й этап – формализация задачи. Содержательное описание задачи переводится на формализованный язык используемой математической теории, т. е. создается математическая модель задачи.

3-й этап – выбор численного метода решения задачи в соответствии с заданными критериями, например при использовании ЭВМ такими критериями могут быть время решения и объем занимаемой памяти.

4-й этап – алгоритмизация. При построении алгоритмов обработки графов должны соблюдаться общие свойства алгоритмов.

1. *Дискретность*. Применение алгоритма осуществляется в дискретном времени в соответствии с последовательностью правил, действий, операций, шагов, процедур по обработке графов.

2. *Детерминированность*. Действия над графом в соответствии с правилами обработки должны быть простыми и определенными, при этом могут использоваться известные процедуры.

3. *Результативность*. Алгоритм состоит из совокупности конечного числа правил и действий, которые должны от исходных данных привести к искомому результату за конечное число шагов. Выбор правил и действий зависит только от результатов предыдущих шагов, например от результата анализа вершин смежных с данной вершиной.

4. *Массовость*. Алгоритм должен применяться для целого класса задач, а не одной задачи.

Однако, если решение задачи представлено в виде алгоритма с соблюдением свойств 1–4, он может иметь свою специфику, связанную с языком теории графов и отражающуюся в его формулировке. Имеется определенное различие в степени подготовленности записи алгоритмов для их реализации на ЭВМ. Поэтому необходимо привести алгоритм, написанный на содержательном уровне и на уровне абстракций теории графов, к виду машинно-ориентированных алгоритмов.

Машинно-ориентированными, т. е. компьютерными, являются алгоритмы, которые можно записывать в виде программы для ЭВМ.

5-й этап – программирование. ЭВМ может быть применена для решения задачи теории графов только после того, как ее алгоритм будет

записан в виде программы, т. е. детальных инструкций по обработке графа на языке, понятном ЭВМ.

В программе должно быть предусмотрено, какие данные и в каком виде должны быть введены в ЭВМ для их последующей обработки. Чаще всего граф вводится в ЭВМ с помощью матрицы смежности вершин. Например, для графа  $G=(V,U)$  без параллельных дуг матрица смежности – это квадратная булева матрица  $\|S_{ij}\|$  размером  $n \times n$ , где  $n$  – число вершин графа. Элемент  $S_{ij} = 1$ , если дуга  $(v_i, v_j) \in U$ , и  $S_{ij} = 0$  в противном случае. Кроме того, может потребоваться введение дополнительной информации о графе в виде массивов (матриц и векторов). Возможно дугам и/или вершинам могут быть приписаны некоторые числа (условно говоря, длины дуг, веса вершин), отражающие дополнительные свойства связей между вершинами. Например, длина дуги – это расстояние между удаленными объектами, представленными вершинами графа.

Программа должна четко предписать последовательность операций, которые нужно произвести над матрицей смежности графа, другими входными данными и промежуточными результатами (матрицами, векторами, числами), формируемыми в процессе решения, чтобы получить выходные результаты, а также предусмотреть, в каком виде эти результаты будут выданы ЭВМ для пользователя.

6-й этап – отладка, т. е. поиск и исправление синтаксических (формальных) ошибок, а также семантических (смысловых) ошибок с использованием специальных тестовых данных.

7-й этап – использование отлаженной программы для расчетов с реальными исходными данными.

8-й этап – интерпретация результатов, полученных в ходе работы программы и рекомендации по их использованию.

После этих кратких замечаний нетрудно понять, какая большая дистанция, как правило, имеется между формулировкой алгоритма на языке теории графов и программой для ЭВМ. Именно разработка компьютерного алгоритма должна быть результатом этапов формализации и алгоритмизации при решении прикладных задач с применением теоретических моделей теории графов.

Ниже будет рассмотрено решение некоторых задач с использованием алгоритмов теории графов с возможным применением компьютерных алгоритмов.

## 2. ЗАДАЧИ О ПУТЯХ В ОРГРАФЕ

Алгоритмы этих задач играют большую роль в теории графов, поскольку с их помощью решаются многие прикладные задачи. Например, это задачи о существовании путей между вершинами графа, о количестве путей между двумя вершинами, о пути с наименьшим числом дуг между вершинами, о кратчайшем пути между двумя вершинами, о длиннейшем по времени пути от начальной вершины до конечной вершины графа (так называемый критический путь), о дереве кратчайших расстояний и другие задачи.

### 2.1. Существование путей в орграфе

Постановка задачи. В орграфе требуется определить, существуют ли какие-либо пути между вершинами графа.

Пусть  $N_0 = \{1, 2, 3, \dots\}$  – множество натуральных чисел без нуля. Для графа  $G=(V, \Gamma)$  путь положительной длины из вершины  $v_i$  в  $v_j$  существует, если

$$(\exists p \in N_0) \quad v_j \in \Gamma^p v_i. \quad (2.1)$$

Если рассматривать пути длины ноль, то (2.1) можно записать так:

$$v_j \in \hat{\Gamma} v_i, \quad (2.2)$$

т. е.  $v_j$  входит в множество транзитивного замыкания вершины  $v_i$ .

#### 2.1.1. Метод решения задачи о путях в графе

Булева матрица смежности  $\|S\|_{n \times n}$  графа  $G=(V, \Gamma)$  без параллельных дуг показывает, какие существуют пути длины 1 (например, рис. 2.1), при этом  $|V| = n$ .

Применим к матрице  $\|S\|$  операции булевых умножения и сложения. Пусть

$$\|S\|^2 = \|S\| \wedge \|S\|,$$

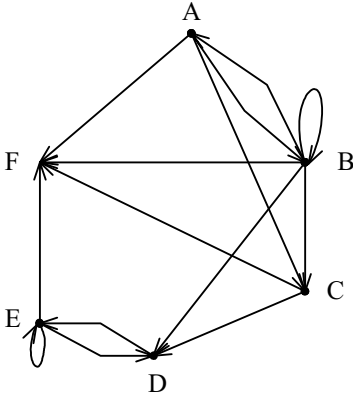
$$\|S\|^4 = \|S\|^2 \wedge \|S\|^2,$$

.....

тогда

$$\|S\|_{ij}^{2^r} = \|S\|_{i1}^{2^{r-1}} \wedge \|S\|_{1j}^{2^{r-1}} \vee \|S\|_{i2}^{2^{r-1}} \wedge \|S\|_{2j}^{2^{r-1}} \vee \dots \vee \|S\|_{in}^{2^{r-1}} \wedge \|S\|_{nj}^{2^{r-1}},$$

где  $i, j=1(1)n$  (изменение от 1 с шагом 1 до  $n$ );  $\wedge$  – булево умножение;  $\vee$  – булево сложение;  $r$  – номер новой булевой матрицы.



$\|S\|_{6 \times 6}$ :

	A	B	C	D	E	F
A	0	1	1	0	0	1
B	1	1	1	1	0	1
C	0	0	0	1	0	1
D	0	0	0	0	1	0
E	0	0	0	1	1	1
F	0	0	0	0	0	0

Рис. 2.1. Граф и его булева матрица смежности

Умножение продолжается до тех пор, пока не сравняются две последние матрицы  $\|S\|^{2^r} = \|S\|^{2^{r-1}}$ . Тогда ненулевые элементы матрицы  $\|S\|^{2^{r-1}}$  покажут, какие вершины в графе связаны какими-либо путями.

### Пример

На рис.2.1 изображен граф и его матрица смежности  $\|S\|$ , а на рис.2.2 представлены матрицы  $\|S\|^2$ ,  $\|S\|^4$ ,  $\|S\|^8$ .

$\|S\|^2$ :

	A	B	C	D	E	F
A	1	1	1	1	0	1
B	1	1	1	1	1	1
C	0	0	0	0	1	0
D	0	0	0	1	1	1
E	0	0	0	1	1	1
F	0	0	0	0	0	0

$\|S\|^4$ :

	A	B	C	D	E	F
A	1	1	1	1	1	1
B	1	1	1	1	1	1
C	0	0	0	1	1	1
D	0	0	0	1	1	1
E	0	0	0	1	1	1
F	0	0	0	0	0	0

$\|S\|^8$ :

	A	B	C	D	E	F
A	1	1	1	1	1	1
B	1	1	1	1	1	1
C	0	0	0	1	1	1
D	0	0	0	1	1	1
E	0	0	0	1	1	1
F	0	0	0	0	0	0

Рис. 2.2. Булевы матрицы существования путей между вершинами графа

Матрица  $\|S\|^2$  показывает, какие существуют пути длины 2 или меньше. Поскольку  $\|S\|^4 = \|S\|^8$ , постольку  $\|S\|^4$  показывает, какие вершины в графе связаны путями.

**2.1.2. Описание машинного алгоритма задачи о существовании путей в графе**

Возможный вариант алгоритма задачи определения существования путей в графе может включать следующие шаги.

1. Ввести булеву матрицу смежности (S) и ее размер (n).
2. Скопировать матрицу S в рабочую матрицу R (R=S), сбросить счетчик новых матриц (r=0).
3. С помощью процедуры булевого умножения матриц вычислить новую матрицу  $R1=R \wedge S$ , вывести ее и увеличить счетчик  $r = r+1$ .
4. В циклах по строкам (i) и столбцам (j) матрицы поэлементно сравнивать  $R1_{ij}$  и  $R_{ij}$ . Если  $R1_{ij} \neq R_{ij}$ , то скопировать  $R=R1$  и перейти к шагу 3, иначе вывести счетчик длин путей  $L = 2^{r-1}$  как результат решения задачи.

**2.2. Пересчет путей в орграфе**

Постановка задачи. В отличие от предыдущей задачи требуется вычислить количество путей конкретной длины (как число дуг) между каждой парой вершин графа.

**2.2.1. Метод решения задачи пересчета путей**

Пересчет различных путей длины r осуществляется получением степеней исходной матрицы смежности графа ( $\|S\|$ ) с помощью операции линейной алгебры обычного умножения двух матриц:

- $\|S\|$  – дает число путей длины 1,
- $\|S\|^2 = \|S\| \times \|S\|$  дает число путей длины 2.
- $\|S\|^3 = \|S\|^2 \times \|S\|$  дает число путей длины 3,

.....

$\|S\|^L = \|S\|^{L-1} \times \|S\|$  – дает число путей длины L для каждой пары вершин  $(v_i, v_j)$ , где

$$S_{ij}^k = S_{i1}^{k-1} \cdot S_{1j} + S_{i2}^{k-1} \cdot S_{2j} + \dots + S_{in}^{k-1} \cdot S_{nj} = \sum_{m=1}^n S_{im}^{k-1} \cdot S_{mj}, k \geq 2.$$

Значение  $L$  определяется в предыдущей задаче существования путей в графе как результирующее значение степени матрицы, после которой начинается повторение путей. В предыдущем примере  $L=4$ .

**П р и м е р**

Для графа (рис.2.1) матрицы (рис.2.3) дают число путей длин соответственно  $k=1,2,3,4$ . Например, имеется пять путей длины 3 от В к D (см.  $\|S\|^3$ ), девять путей длины 4 от В к D (см.  $\|S\|^4$ ).

**2.2.2. Описание машинного алгоритма пересчета путей в графе**

Алгоритм пересчета длин путей можно описать следующим образом:

1. Ввести матрицу смежности ( $S$ ), ее размер ( $n$ ) и конечную длину пути ( $L$ ), полученную в предыдущей задаче.
1. Скопировать  $S$  в рабочую матрицу  $R$ .
2. В цикле  $k=2(1)L$  использовать процедуру умножения двух матриц ( $R1=R \times S$ ), выводить матрицу  $R1$  как результат и копировать  $R1$  в  $R$ .

$\|S\|:$

	A	B	C	D	E	F
A	0	1	1	0	0	1
B	1	1	1	1	0	1
C	0	0	0	1	0	1
D	0	0	0	0	1	0
E	0	0	0	1	1	1
F	0	0	0	0	0	0

$\|S\|^2:$

	A	B	C	D	E	F
A	1	1	1	2	0	2
B	1	2	2	2	1	3
C	0	0	0	0	1	0
D	0	0	0	1	1	1
E	0	0	0	1	2	1
F	0	0	0	0	0	0

$\|S\|^3:$

	A	B	C	D	E	F
A	1	2	2	2	2	3
B	2	3	3	5	3	6
C	0	0	0	1	1	1
D	0	0	0	1	2	1
E	0	0	0	2	3	2
F	0	0	0	0	0	0

$\|S\|^4:$

	A	B	C	D	E	F
A	2	3	3	6	4	7
B	3	5	5	9	8	11
C	0	0	0	1	2	1
D	0	0	0	2	3	2
E	0	0	0	3	5	3
F	0	0	0	0	0	0

**Рис. 2.3. Матрицы пересчета длин путей между вершинами графа**

## 2.3. Путь с наименьшим числом дуг. Поиск транзитивного замыкания вершин

Постановка задачи. Пусть задан произвольный граф  $G=(V,U)$ . Требуется построить такой путь, соединяющий две заданные вершины  $v$  и  $w$ , который содержит наименьшее число дуг (это путь минимальной длины, если считать, что длины всех дуг одинаковы, т. е. элементарный путь).

### 2.3.1. Метод решения задачи поиска пути с наименьшим числом дуг

Рассмотрим алгоритм этой задачи, который проиллюстрирован на рис. 2.4. Алгоритм можно разделить на два этапа: прямой ход – присваивание вершинам графа меток; обратный ход – определение минимального пути от конечной вершины до исходной.

*Прямой ход:*

1. Присвоить вершине  $v$  метку 0.
2. Если  $x_i \in \Gamma v$  и  $x_i \neq v$ , то присвоить каждой такой вершине метку 1, (т. е. это вершины-концы дуг, исходящих из вершины  $v$ , кроме самой вершины  $v$ , при этом исключаются петли).
3. Пусть  $X(m)$  ( $m=0,1,2,\dots$ ) – множество вершин, имеющих метку  $m$ . Вершинам  $X(m+1) = \{x \in \Gamma X(m), x \notin X(k) \text{ при } k \leq m\}$  (т. е. вершинам-концам дуг, исходящих из вершин множества  $X(m)$ , кроме вершин, уже получивших метки ранее, – тем самым исключаются контуры) присвоить метки  $m+1$ .
4. Процесс присвоения вершинам меток прекратить, как только вершина  $w$  получит метку  $n$  ( $w \in X(n)$ ).

*Обратный ход:*

5. Рассмотреть вершины  $x_{i_1}, x_{i_2}, \dots, x_{i_n}$  такие, что

$$x_{i_1} \in X(n-1), w \in \Gamma x_{i_1},$$

$$x_{i_2} \in X(n-2), x_{i_1} \in \Gamma x_{i_2},$$

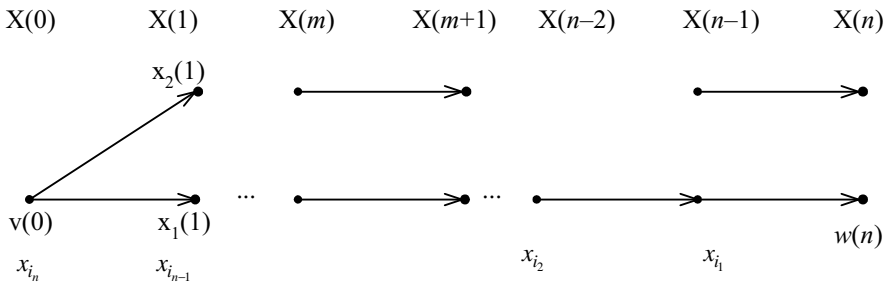
$$\dots\dots\dots$$

$$x_{i_n} \in X(0), x_{i_{n-1}} \in \Gamma x_{i_n}.$$

Путь  $\mu = (v = x_{i_n}, x_{i_{n-1}}, \dots, x_{i_2}, x_{i_1}, w)$  дает решение задачи.

*Замечание.* Если на некотором шаге невозможно присвоение метки  $m+1$  вершинам, поскольку множество  $\Gamma X(m)$  пусто и вершина  $w$  не получила метки, то это означает, что в графе  $G$  не существует пути, соединяющего вершину  $v$  с вершиной  $w$ .





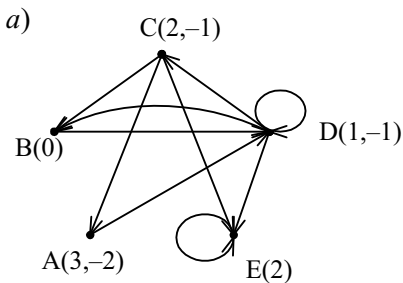
**Рис. 2.4.** Процесс поиска пути между вершинами  $v$  и  $w$

Рассмотренный алгоритм описывает процесс нахождения транзитивного замыкания  $\hat{\Gamma}^v$ , отображающего дерево кратчайших путей, исходящих от любой вершины  $v$  как от корня дерева.

Аналогично может быть описан процесс нахождения обратного транзитивного замыкания  $\hat{\Gamma}^{-v}$ , т. е. множества вершин, из которых можно попасть в  $v$  по кратчайшему пути.

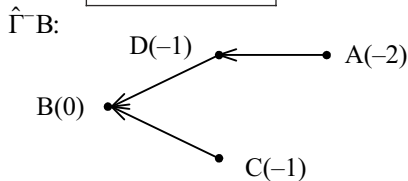
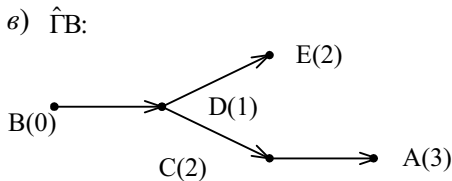
**П р и м е р**

Для графа  $G$  на рис. 2.5,*a* показаны метки вершин, начиная с вершины  $B$ , положительные для транзитивного замыкания  $\hat{\Gamma}^B$  и отрицательные для обратного транзитивного замыкания  $\hat{\Gamma}^{-B}$ .



b)  $\|S\|$ :

0	0	0	1	0	A	3				
0	0	0	1	0	B	0				
1	1	0	0	1	C	2				
0	1	1	1	1	D	1				
0	0	0	0	1	E	2				
					A	B	C	D	E	$\hat{\Gamma}^B$



**Рис. 2.5.** Пути в графе с наименьшим числом дуг для вершины  $B$ :  
*a* – граф с метками  $\hat{\Gamma}^B$  и  $\hat{\Gamma}^{-B}$ ; *b* – матрица смежности графа  $S$  и замыканий  $\hat{\Gamma}^B$  и  $\hat{\Gamma}^{-B}$ ; *в* – графы замыканий вершины  $B$

Принципы построения машинного алгоритма поиска транзитивного и обратного транзитивного замыканий рассмотрим на примере получения  $\hat{\Gamma}V$  и  $\hat{\Gamma}^{-1}V$  на основе анализа матрицы смежности графа (рис. 2.5,б).

Справа от матрицы смежности  $S$  находится вектор-столбец для  $\hat{\Gamma}V$ , а внизу – вектор-строка для  $\hat{\Gamma}^{-1}V$ . Рассмотрим порядок заполнения столбца  $\hat{\Gamma}V$ .

1. В позицию  $B$  столбца  $\hat{\Gamma}V$  записываем ноль (начальная вершина).

2. В строке  $B$  матрицы  $S$  единица находится в позициях, соответствующих вершинам, в которые есть путь длины 1, поэтому в позицию  $D$  столбца  $\hat{\Gamma}V$  записываем 1.

3. Строка  $D$  матрицы содержит 1 в столбцах, соответствующих вершинам, которые отстоят в графе от вершины  $D$  на одну дугу, а значит, кратчайший путь до них от вершины  $B$  равен 2 дугам.

Если в клетках столбца  $\hat{\Gamma}V$ , соответствующих единичным позициям столбцов в строке  $D$  матрицы, уже находятся числа, то они не меняются ( $B, D$ ), а в пустые клетки  $C$  и  $E$  записываем 2. Тем самым исключаются петли (например, в вершине  $D$ ) и контуры (например, дуга  $(D,B)$ ), и продолжается рост дерева кратчайших путей от вершины  $B$ .

4. Повторяем п. 3 для вершин  $C, E$  и других, увеличивая длину пути на 1, до тех пор, пока это возможно.

Аналогично действуем для получения  $\hat{\Gamma}^{-1}V$ , но при этом необходимо транспонировать матрицу смежности (поменять строки и столбцы местами) и использовать рассмотренный алгоритм или выполнять анализ исходной матрицы по столбцам. Числа, полученные в строке  $\hat{\Gamma}^{-1}V$ , дают минимальные длины путей (числа дуг) из тех вершин, из которых можно попасть в вершину  $B$ , например из вершины  $E$  нельзя попасть в  $B$  никаким путем.

### 2.3.2. Описание алгоритма поиска транзитивного замыкания

Машинный алгоритм должен обеспечивать поиск транзитивного замыкания (вектор  $Z$ ) для любой начальной вершины ( $K$ ) графа. Возможный вариант алгоритма может включать следующие основные шаги.

1. Ввод матрицы смежности  $S$ , ее размера ( $n$ ) и номера начальной вершины ( $k$ ).

2. Установка начального значения вектора  $Z$  в цикле ( $i=1(1)n, Z_i = -1$ ).

3. Установка  $Z_k=0$  (начальная вершина),  $L=0$  (счетчик длины пути).

4. Цикл анализа необработанных вершин:

5. {  $C=0$  ; (счетчик обработанных вершин)
6. цикл  $i=1(1)n$  (заполнение вектора  $Z$ )
  - { если ( $Z_i \neq L$ ) (вершина не удалена на длину  $L$ , то)
  - {  $C=C+1$ ; ( подсчет таких вершин )
  - продолжение цикла  $i$ ;
  - }
- иначе
7. цикл  $j=1(1)n$  (по позициям строки  $i$  матрицы  $S$ )
  - {если ( $S_{ij} = 1$  и  $Z_j < 0$ ) (вершина  $j$  входит в замыкание и еще не включена,то)
  - $Z_j = L+1$ ; (занесение длины пути до вершины  $j$  в  $Z_j$ )
8. } (конец цикла  $j$ )
9. } (конец цикла  $i$ )
10.  $L=L+1$ ; (изменение счетчика длины пути )
- 11.} пока ( $C < n$ ); (не все вершины обработаны, продолжать цикл анализа)
12. Вывод результата:  $Z$  (вектор транзитивного замыкания).

### 3. ЗАДАЧИ ОБ ОПТИМАЛЬНЫХ РАССТОЯНИЯХ

В предыдущей задаче мы считали, что длины всех дуг графа  $G(V,U)$  были равны, например  $L(u)=1$ . Рассмотрим теперь случай, когда каждой дуге  $u$  поставлено в соответствие число  $L(u)>0$ , называемое *длиной дуги*. В зависимости от конкретного приложения, это число может быть мерой физического расстояния, времени, стоимости или другого важного параметра.

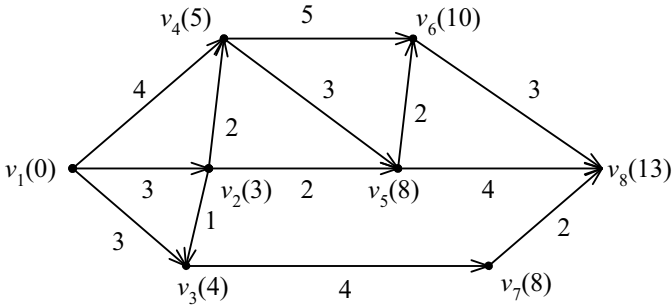
Рассмотрим общую постановку задачи об оптимальных расстояниях на примере задачи сетевого планирования и управления (СПУ) операциями, называемых также задачами типа ПЕРТ (PERT).

Орграф является естественным средством для описания и анализа сложных проектов, требующих выполнения большого числа взаимосвязанных операций (работ). Проектом может быть, например, процесс разработки, построения и проверки некоторого изделия, в том числе и с использованием САПР (системы автоматизированного проектирования) или процесс проектирования и строительства здания, включая этапы получения и подготовки места строительства. В общем случае предположим, что рассматривается некоторый проект  $P$  и множество всех операций, связанных с выполнением проекта, можно разделить на отдельные непересекающиеся операции  $P\{a_1, a_2, \dots, a_n\}$ .

Хотя некоторые операции могут быть независимы друг от друга, в общем случае между ними существует зависимость по времени, например операция  $a_i$  должна закончиться, чтобы могла начаться операция  $a_j$ . Если заданы все такие зависимости, то их удобно представить в виде орграфа, как показано на рис. 2.6. Каждая дуга графа соответствует одной операции, а каждая вершина, называемая событием, соответствует некоторому моменту времени.

Для выполнения операции  $a_i$  выделяется некоторое время  $t(a_i)$ , что отражено числами на дугах графа. Длина (т. е. сумма временных интервалов) любого пути из  $v_1$  в  $v_i$  соответствует нижней границе времени, измеряемого от начала проекта до наступления события  $v_i$ , после

которого могут быть начаты операции, имеющие  $v_i$  в качестве начальной вершины.



**Рис. 3.1. Граф выполнения операций проекта**

При расчетах каждой вершине удобно поставить в соответствие число (время) следующим образом:

$$T(v_1)=0, T(v_i)=\max \{t(\mu)\} \quad (i \neq 1),$$

где  $t(\mu)$  означает длину пути и максимум определяется по всем путям из  $v_1$  в  $v_i$ .

*Длиной пути  $\mu$*  называется сумма длин дуг, входящих в  $\mu$ :

$$L(\mu) = \sum_{u \in \mu} L(u).$$

Длиннейший по времени путь от начального события  $v_1$  до конечного события  $v_n$  называется *критическим путем*, который может быть не единственным, например для графа на рис. 3.1 есть два критических пути (по вершинам графа):

$$\mu_1 = (v_1, v_2, v_4, v_5, v_6, v_8), \quad L(\mu_1) = 13,$$

$$\mu_2 = (v_1, v_2, v_4, v_6, v_8), \quad L(\mu_2) = 13.$$

Длина критического пути соответствует кратчайшему времени, за которое могут быть выполнены все операции проекта, а значит, и весь проект в целом.

Заметим, что по своей природе граф, представляющий процесс выполнения операций, является *ациклическим*. Наличие цикла создало бы невозможную ситуацию, когда ни одна из операций цикла не могла бы начаться первой, поскольку любая из них зависит от другой операции.

В дальнейшем мы будем пользоваться терминологией, соответствующей расстояниям, хотя всегда нужно помнить и о других интерпретациях. Отметим две важнейшие характеристики длины, существенные для дальнейшего рассмотрения.

1. Длина должна быть аддитивна (длина совокупности дуг равна сумме длин отдельных дуг).

2. Длина должна использоваться в качестве целевой функции, которую необходимо оптимизировать (минимизировать или максимизировать) при ограничениях на допустимые множества дуг.

Для графа с контурами (циклического) можно говорить только о задаче отыскания минимальных путей графа, а в ациклическом графе можно искать как минимальный, так и максимальный путь между вершинами графа.

### 3.1. Поиск минимальных путей в графе

Постановка задачи. Для произвольного орграфа  $G(V, U)$ , где для  $\forall u \in U$  сопоставлено число  $L(u) > 0$  (длина дуги), необходимо найти путь  $\mu$  минимальной длины, соединяющий вершину  $v_0$  с вершиной  $v_n$ .

#### 3.1.1. Метод решения задачи поиска минимального пути

Существует несколько методов решения этой задачи, среди которых рассмотрим алгоритм Форда, описываемый следующими правилами, которые можно разделить на два этапа: прямой ход – вычисление и назначение меток вершинам графа; обратный ход – определение минимального пути.

*Прямой ход.*

1. Назначение номеров вершинам. Переименовать вершины графа  $G$  так, чтобы исходная вершина получила номер 0, если это необходимо.

2. Присвоение начальных меток. Присвоить каждой вершине  $v_i$  метку  $\lambda_i$  так, чтобы  $\lambda_0 = 0$ ,  $\lambda_i = \infty$  при  $i > 0$  (под символом  $\infty$  здесь следует понимать достаточно большое положительное значение).

3. Уменьшение меток. Найти такую дугу  $(v_i, v_j)$ , для которой  $\lambda_j - \lambda_i > L(v_i, v_j)$ . У вершины  $v_j$  заменить метку  $\lambda_j$  на новую метку  $\lambda_j = \lambda_i + L(v_i, v_j)$ , если  $\lambda_j > \lambda_j$ .

4. Пересчет меток. Применять правило 3 до тех пор, пока для каждой дуги  $(v_i, v_j)$  не станет справедливым неравенство  $\lambda_j - \lambda_i \geq L(v_i, v_j)$ .

Обратный ход.

5. В множестве  $\Gamma^{-1}v_n$  найти такую вершину  $v_{p_1}$ , что

$$\lambda_n = \lambda_{p_1} + L(v_{p_1}, v_n) \text{ или } \lambda_n - \lambda_{p_1} = L(v_{p_1}, v_n).$$

Аналогично, в множестве  $\Gamma^{-1}v_{p_1}$  найти такую вершину  $v_{p_2}$ , чтобы было справедливо равенство

$$\lambda_{p_1} = \lambda_{p_2} + L(v_{p_2}, v_{p_1}) \text{ или } \lambda_{p_1} - \lambda_{p_2} = L(v_{p_2}, v_{p_1}) \text{ и т. д.}$$

После некоторого числа шагов вершина  $v_{p_k}$  совпадет с вершиной  $v_0$ .

Путь  $\mu = (v_0 = v_{p_k}, v_{p_{k-1}}, \dots, v_{p_2}, v_{p_1}, v_n)$  – минимальный и его длина  $L(\mu) = \lambda_n$ .

### Пример

Проиллюстрируем работу алгоритма поиска минимального пути для графа на рис. 3.2.

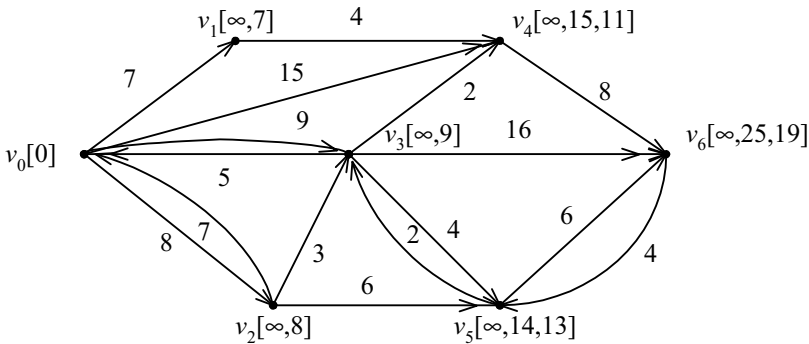


Рис. 3.2. Орграф с длинами дуг и метками вершин

Этап 1 (прямой ход) – вычисление новых меток при вершинах.

Приписываем вершине  $v_0$  метку 0, а остальным вершинам – метку  $\infty$ . Находим новые метки, которые показаны в квадратных скобках при вершинах графа:

1.  $L(v_0, v_1)=7, \lambda'_1=0+7 < \lambda_1=\infty$ , остается  $\lambda_1=7$ ;
2.  $L(v_0, v_2)=8, \lambda'_2=0+8 < \lambda_2=\infty$ , остается  $\lambda_2=8$ ;
3.  $L(v_0, v_3)=9, \lambda'_3=0+9 < \lambda_3=\infty$ , остается  $\lambda_3=9$ ;
4.  $L(v_0, v_4)=15, \lambda'_4=0+15 < \lambda_4=\infty$ ;
5.  $L(v_1, v_4)=4, \lambda'_4=7+4=11 < \lambda_4=15$ , остается  $\lambda_4=11$ ;
6.  $L(v_2, v_3)=3, \lambda'_3=8+3=11 > \lambda_3=9$ ;

7.  $L(v_2, v_5)=6, \lambda'_5=8+6=14 < \lambda_5=\infty,$
8.  $L(v_3, v_4)=2, \lambda'_4=9+2=11 = \lambda_4=11,$
9.  $L(v_3, v_5)=4, \lambda'_5=9+4=13 < \lambda_5=14,$  остается  $\lambda_5=13;$
10.  $L(v_3, v_6)=16, \lambda'_6=9+16=25 < \lambda_6=\infty,$
11.  $L(v_4, v_6)=8, \lambda'_6=11+8=19 < \lambda_6=25,$
12.  $L(v_5, v_3)=2, \lambda'_3=13+2=15 > \lambda_3=9,$
13.  $L(v_5, v_6)=6, \lambda'_6=13+6=19,$  остается  $\lambda_6=19.$

Этап 2 (обратный ход) – определение минимального пути.

Из вершины  $v_6$  находим:

$$1. \lambda_6 - L(v_4, v_6) = 19 - 8 = 11 = \lambda_4,$$

$$\lambda_4 - L(v_1, v_4) = 11 - 4 = 7 = \lambda_1,$$

$$\lambda_1 - L(v_0, v_1) = 7 - 7 = 0 = \lambda_0,$$

значит, путь  $\mu = (v_0, v_1, v_4, v_6)$  – минимальный.

$$2. \lambda_6 - L(v_3, v_6) = 19 - 16 = 3 < \lambda_3 = 9,$$

значит, дуга  $(v_3, v_6)$  не входит в минимальный путь.

$$3. \lambda_6 - L(v_5, v_6) = 19 - 6 = 13 = \lambda_5,$$

$$\lambda_5 - L(v_3, v_5) = 13 - 4 = 9 = \lambda_3,$$

$$\lambda_3 - L(v_0, v_3) = 9 - 9 = 0 = \lambda_0,$$

значит, путь  $\mu = (v_0, v_3, v_5, v_6)$  – минимальный.

4. Так же можно доказать, что путь  $\mu = (v_0, v_3, v_4, v_6)$  – минимальный.

### 3.1.2. Обоснование алгоритма Форда

Докажем, что после конечного числа применений правила 3 для каждой дуги графа станет справедливым неравенство

$$\lambda_j - \lambda_i \leq L(v_i, v_j) \text{ (правило 4).}$$

На любом шаге изменения метка  $\lambda_j > 0$  при  $j \neq 0$  (метка  $\lambda_0 = 0$ ), что можно доказать по индукции.

1. Действительно, при первом применении правила 3 будет изменена метка  $\lambda_k$  у одной из вершин  $v_k$ , смежных с вершиной  $v_0$  ( $v_k \in \Gamma v_0$ ). Эта вершина  $v_k$  получит новую метку  $\lambda_k = L(v_0, v_k) > 0$ , например  $v_4[15]$ .

Предположим, что после применения правила 3  $m$  раз ( $m > 1$ ), станет справедливым утверждение, что  $\lambda_0 = 0, \lambda_i > 0$  для  $i > 0$ . На  $m+1$ -м шаге какая-то вершина  $v_j$  получит новую метку  $\lambda_j = \lambda_i + L(v_i, v_j)$ . Но в силу предположения индукции  $\lambda_i \geq 0$ , кроме того,  $L(v_i, v_j) > 0$ , поэтому  $\lambda_j > 0$ .



2. Нетрудно заметить, что при каждом изменении метка вершины графа уменьшается на положительную величину, не меньшую, чем минимальная разность длин путей графа.

Из этих двух утверждений следует, что метка любой вершины графа может изменяться лишь конечное число раз. Так как вершин конечное множество, то правило 3 может применяться лишь конечное число раз.

Рассмотрим обратный этап и покажем, что, применяя правило 5, мы можем найти минимальные пути.

Действительно, так как в результате применения правила 3 метка  $\lambda_n$  монотонно уменьшается, то мы придем к такой вершине  $v_{p_1}$  (это начало дуги, с помощью которой в последний раз уменьшилась метка  $\lambda_n$ ), что

$$\lambda_n - \lambda_{p_1} = L(v_{p_1}, v_n), \text{ например } \lambda_6 - \lambda_4 = L(v_4, v_6).$$

По этой же причине найдется вершина  $v_{p_2}$ , для которой соблюдается соответствие

$$\lambda_{p_1} - \lambda_{p_2} = L(v_{p_2}, v_{p_1}) \text{ и т. д.}$$

По условию задачи длина дуги графа  $L(v_i, v_j) > 0$ , поэтому метки  $\lambda_n, \lambda_{p_1}, \lambda_{p_2}, \dots$  образуют строго убывающую последовательность неотрицательных чисел, отличающихся друг от друга на величину, большую или равную длине кратчайшей дуги графа. Следовательно, при некотором  $k$  получим  $\lambda_{p_k} = 0, v_{p_k} = v_0$ . (Вершина  $v_0$  выделена тем, что ей в силу правила 1 с самого начала присвоена метка  $\lambda_0 = 0$  и в формировании этой метки не участвуют дуги графа).

Покажем, что  $\mu = (v_0 = v_{p_k}, v_{p_{k-1}}, \dots, v_{p_2}, v_{p_1}, v_n)$  – минимальный путь со значением длины пути  $L(\mu) = \lambda_n$ , т. е. значение  $L(v_0, v_{i_1}, v_{i_2}, \dots, v_{i_m}, v_n)$  любого другого пути между  $v_0$  и  $v_n$  не меньше  $\lambda_n$ .

Имеем неравенства (по правилу 4):

$$\begin{aligned} \lambda_{i_1} - \lambda_0 &\leq L(v_0, v_{i_1}), \\ &+ \\ \lambda_{i_2} - \lambda_{i_1} &\leq L(v_{i_1}, v_{i_2}), \\ &\dots\dots\dots, \\ &+ \\ \lambda_n - \lambda_{i_m} &\leq L(v_{i_m}, v_n). \end{aligned}$$

Складывая эти неравенства, получим (при  $\lambda_0 = 0$ )

$$\lambda_n \leq L(v_0, v_{i_1}, v_{i_2}, \dots, v_{i_m}, v_n).$$

В то же время по построению пути  $\mu$  имеем  $L(\mu) = \lambda_n$ , откуда следует

$$L(\mu) \leq L(v_0, v_{i_1}, v_{i_2}, \dots, v_{i_m}, v_n),$$

т. е.  $\mu = (v_0 = v_{p_k}, v_{p_{k-1}}, \dots, v_{p_2}, v_{p_1}, v_n)$  – минимальный путь.

Как показано в примере, минимальный путь в графе может быть не один.

### 3.1.3. Построение машинного алгоритма поиска минимальных путей в графе

Исходными данными являются матрица смежности графа для  $n$  вершин ( $\|S\|_{n \times n}$ ) и матрица длин дуг графа ( $\|L\|_{n \times n}$ ), например для графа на рис. 3.2 эти матрицы имеют вид, приведенный на рис. 3.3.

$\|S\|$ :

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

$\|L\|$ :

	0	1	2	3	4	5	6
0	0	7	8	9	15	0	0
1	0	0	0	0	4	0	0
2	7	0	0	3	0	6	0
3	5	0	0	0	2	4	16
4	0	0	0	0	0	0	8
5	0	0	0	2	0	0	6
6	0	0	0	0	0	4	0

Рис. 3.3. Матрицы смежности и длин дуг графа

Матрица длин дуг графа по сути включает всю информацию о графе, что делает матрицу смежности излишней для данной задачи. Для построения алгоритма необходимы вектор меток вершин (M) и матрица результатов, содержащая вершины кратчайших путей ( $\|K\|_{n \times n}$ ).

Построение машинного алгоритма основано на анализе матрицы длин дуг графа (L), постепенном заполнении вектора меток вершин (M) новыми метками и получении матрицы кратчайших путей (K).

В результате вектор меток (M) отражает минимальные расстояния между начальной вершиной  $v_0$  и остальными вершинами. Матрица кратчайших путей (K) отражает информацию о подграфе минимальных путей исходного графа между начальной вершиной  $v_0$  и конечной вершиной  $v_n$ .

### 3.1.4. Описание машинного алгоритма Форда

Поскольку задача может иметь несколько вариантов решения, то возможный алгоритм может включать следующие основные шаги.

1. Ввод матрицы длин дуг графа ( $L$ ) и ее размера ( $n$ ).
2. Установка начальных значений: 1) вектора меток:  $M_0=0$ ;  $i=1(1)n-1$ ,  $M_i=\max$ ; где  $\max$  – максимально допустимое число данного типа; 2) обнуление вспомогательного вектора вершин, входящих в критические пути,  $V$  и матрицы кратчайших путей  $K$ .

*Прямой ход*

(Реализация правил 3 и 4 алгоритма Форда – циклы поиска смежных вершин графа по строкам  $i$  и столбцам  $j$  и обновление меток вершин):

3. Цикл  $i=0(1)n-1$  (по строкам матрицы  $L$ )
4. Цикл  $j=0(1)n-1$  (по столбцам матрицы  $L$ )  
 если ( $L_{ij}>0$  и  $M_j - M_i > L_{ij}$ ) (есть метка, требующая изменения, то)  
 {  $MN=M_i + L_{ij}$ ; (вычисление новой метки ( $MN$ ) вершины  $j$ )  
 если ( $MN < M_j$ ) (новая метка меньше старой, то)  
 $M_j=MN$ ; (изменение метки вершины  $j$ )  
 }

*Обратный ход*

(Реализация правила 5 алгоритма Форда – поиск вершин кратчайших путей):

5.  $V_{n-1}=1$ ; (конечная вершина кратчайшего пути)
6. Цикл  $j=n-1(-1)0$  (обратный цикл по вершинам, начиная с последней)  
 если ( $V_j=1$ ) (вершина входит в кратчайший путь, то)
7. Цикл  $i=0(1)n-1$  (поиск предшествующих вершин  $i$ , смежных с вершиной  $j$ )  
 {если ( $L_{ij}>0$  и  $M_j - M_i = L_{ij}$ ) (разность меток смежных вершин равна длине дуги между ними, то)  
 {  $K_{ij}=L_{ij}$ ; (заполнение матрицы кратчайших путей)  
 $V_i=1$ ; (включение вершины  $i$  в вектор  $V$ )  
 }  
 }

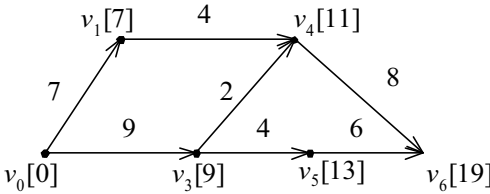
8. Вывод матрицы смежности  $K$  ( подграфа кратчайших путей) и вектора меток вершин  $M$  (длин кратчайших путей от вершины  $V_0$ ).

По матрице  $K$  необходимо построить для исходного графа подграф кратчайших путей от вершины  $v_0$  к  $v_n$ .

а) М:

Вершины	0	1	2	3	4	5	6
Метки	0	7	8	9	11	13	19

б)



в) ||K||:

	0	1	2	3	4	5	6
0	0	7	0	9	0	0	0
1	0	0	0	0	4	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	2	4	0
4	0	0	0	0	0	0	8
5	0	0	0	0	0	0	6
6	0	0	0	0	0	0	0

**Рис. 3.4.** Результаты поиска минимальных путей в графе: *a* – вектор меток М (длин кратчайших путей от вершины  $v_0$  до остальных); *б* – подграф кратчайших путей от вершины  $v_0$  до  $v_6$ ; *в* – матрица длин путей (К) подграфа

На рис. 3.4 представлены результаты обработки матрицы длин дуг L (рис. 3.3) исходного графа на рис. 3.2 в соответствии с алгоритмом Форда.

### 3.2. Поиск максимального пути в графе без контуров

Алгоритм Форда может быть использован для отыскания максимального пути (или критического пути в задаче сетевого планирования и управления) в ациклическом графе. Достаточно положить начальные метки вершин  $\lambda_i=0, i=0, 1, 2, \dots, n$ , а затем заменять  $\lambda_j$  на  $\lambda'_j=\lambda_i+L(v_i, v_j)$ , если  $\lambda'_j > \lambda_j$ , до тех пор, пока возможно увеличивать  $\lambda_j$ .

## 4. ЗАДАЧА ВЫБОРА ПРЕДПОЧТИТЕЛЬНЫХ ВАРИАНТОВ ИССЛЕДУЕМОЙ СИСТЕМЫ

### 4.1. Общая постановка задачи

При проектировании сложной системы, как правило, исследуется несколько вариантов системы, при этом возникают проблемы как при оценке эффективности системы, так и при выборе наиболее эффективного, а значит, и предпочтительного варианта. Под эффективностью системы понимается мера соответствия системы своему назначению, т. е. степень приспособленности ее для решения поставленных перед ней задач, которая оценивается заданной целевой функцией  $f$  (функцией критерия эффективности).

Трудность состоит в том, что эффективность сложной системы не может быть оценена одним показателем, даже универсальным, поскольку она является функцией большого количества факторов и критериев эффективности всех подсистем, входящих в рассматриваемую систему. Эффективность сложной системы необходимо оценивать с различных точек зрения с помощью частных показателей качества ее функционирования, в общем случае противоречивых, например габариты, вес, пропускная способность, надежность, сложность алгоритмов функционирования, стоимость производства и эксплуатации и т. п. Таким образом, критерий эффективности сложной системы является векторным

$$F = \{f_n\}, \quad n = \overline{1, N},$$

где  $f_n$  – частные показатели качества (критерии эффективности) функционирования сложной системы.

Подобная проблема возникает и при экспертном анализе выбираемых объектов, когда объекты, например квартира, оценивается несколькими специалистами (экспертами) по нескольким показателям (метраж, удобства, стоимость, удаленность от транспорта или центра города и др.). При этом оценки экспертов могут отличаться весьма существенно.

Возникает общая проблема оценки  $M$  вариантов исследуемой системы (объектов) по  $N$  критериям и выбора наиболее предпочтительных из них (желательно выбрать один вариант), с точки зрения всех критериев. Таким образом, эту проблему выбора можно разбить на два этапа.

На первом этапе варианты системы исследуются по множеству критериев  $F = \{f_n\}$ ,  $n = \overline{1, N}$ . Результаты исследования (оценивания) всех  $M$  вариантов сводятся в матрицу оценок  $\|\mathbf{P}\|_{M \times N}$ , отображающую статически устойчивые или детерминированные оценки показателей качества  $M$  вариантов по  $N$  критериям. В нашем последующем рассмотрении задачи мы будем считать эти данные  $(M, N, P)$  как исходные.

Второй этап заключается в последовательной обработке матрицы оценок  $\mathbf{P}$ . Необходимо из совокупности индивидуальных оценок вариантов по заданному набору критериев выработать групповое решение о порядке предпочтения альтернативных вариантов, т. е. должна быть решена проблема группового выбора. Формализация и решение этой задачи показаны далее.

## 4.2. Получение сравнимых оценок

Пусть  $p_{mn}$  – оценка  $m$ -го варианта по критерию  $f_n$ . Оценки  $p_{mn}$ ,  $m = \overline{1, M}$ , являются абсолютными, они получены в “естественной” для каждого критерия шкале и расположены в некотором диапазоне  $[p_{\min}(n), p_{\max}(n)]$ . При этом для одних критериев  $p_{\max}$  является наилучшей из полученных оценок, для других наихудшей, т. е. такие оценки по разным критериям являются несравнимыми.

Первый шаг обработки матрицы оценок  $\mathbf{P}$  заключается в преобразовании элементов матрицы с целью получения сравнимых оценок путем приведения их к единому масштабу и началу отсчета. Эта задача может быть решена различными методами, например ранжированием оценок. Ранжирование представляется процедурой вычисления оценок в ранговой шкале, которую можно описать следующим образом.

Обозначим  $p_1(n)$ ,  $p_x(n)$  – соответственно лучшую и худшую оценки по критерию  $f_n \in F$ ,  $n = \overline{1, N}$ . Оценки по критерию  $f_n$  для всех вариантов находятся в диапазоне

$$p_x(n) \leq p_{mn} \leq p_1(n), \quad m = \overline{1, M}.$$

Построение шкалы оценок  $S_n$  заключается в задании числа градаций  $D$  (рангов, делений) шкалы и определении цены деления шкалы

$$d_n = (p_1(n) - p_x(n)) / D.$$

Близкие оценки (с точностью до цены деления) могут попасть в один ранг шкалы  $S_n$ , т. е. эти оценки эквивалентны в данной шкале  $S_n$ , значит,  $D$  определяет количество классов эквивалентностей в матрице оценок.

Из исходной матрицы оценок  $\mathbf{P}$  абсолютные оценки  $p_{mn}$  преобразуются в ранжированные оценки  $r_{mn}$  следующим образом:

для  $p_1(n)$  полагаем  $r_{mn} = D$ ,

для остальных вычисляем по формуле  $r_{mn} = [(p_{mn} - p_x(n)) / d_n + 1]$ , где  $[x]$  – ближайшее целое число, меньшее или равное  $x$ , т. е. целая часть  $x$ . Оценки  $r_{mn}$  соответствуют рангу, т. е. более предпочтительному объекту соответствует более высокий ранг.

Преобразованная таким образом матрица ранжированных оценок  $\|\mathbf{R}\|_{M \times N}$  содержит сравнимые по различным критериям оценки, поскольку ранжированные оценки теряют “абсолютный” вес и знак, специфичность, определяемую ценой деления шкалы данного критерия, сохраняя относительный вес, т. е. положение в ранговой шкале.

### 4.3. Получение групповых оценок

Второй шаг обработки матрицы оценок  $\mathbf{P}$  состоит в выработке группового предпочтения вариантов на основе индивидуальных ранжированных оценок матрицы  $\mathbf{R}$ , полученной в результате согласования общей шкалы оценок.

Простейший способ получения групповой оценки состоит в том, что каждому критерию  $f_n$  ставится в соответствие число  $l_n$ , называемое “весом” критерия  $f_n$ . Совокупность  $\{l_n\}$ ,  $n = \overline{1, N}$ , упорядочивает множество  $F = \{f_n\}$  по степени “важности” критериев. Групповая оценка определяется как вектор  $\mathbf{G} = \mathbf{R}_{M \times N} \times \mathbf{L}$ , где  $\mathbf{L} = \{l_n\}$ ,  $n = \overline{1, N}$ , а элементы вектора  $\mathbf{G}$  определяются выражением

$$g_m = \sum_{n=1}^N r_{mn} l_n,$$

отсюда можно получить усредненные значения

$$\bar{g}_m = g_m / N = \left( \sum_{n=1}^N r_{mn} l_n \right) / N, \quad m = \overline{1, M},$$

где  $M$  – количество вариантов.

Варианты объектов сравнения упорядочиваются в соответствии со значением величин  $g_m$  или  $\bar{g}_m$ , при этом вариант с максимальным значением  $g_m$  ( $\bar{g}_m$ ) является наиболее предпочтительным. Если образуется множество неразличимых по  $g_m$  вариантов, то дальнейшее упорядочение осуществляется с помощью лексиграфического порядка. Принимается, что вариант  $v_1$  предпочтительнее варианта  $v_2$ , если оценка  $v_1$  по самому важному показателю выше, чем оценка  $v_2$ , при равенстве этих оценок варианты сравниваются по показателю, следующему по важности и т. д.

### Пример

Рассмотрим шаги обработки матрицы оценок  $P_{M \times N}$  для следующих исходных данных: количество объектов сравнения (вариантов)  $M=4$ , количество критериев (функций предпочтения)  $N=6$ , матрица оценок  $P_{4 \times 6}$ :

	1	2	3	4	5	6
1	2,5	102	0,1	2	-3,5	4
2	-3	57	0,9	1	0	0
3	5,7	32	0,4	10	-5	2
4	17	52	1,1	0	5	3

*Первый шаг* обработки  $P$  – получение относительных оценок. Диапазоны оценок по критериям  $f_n \rightarrow [p_x(n), p_1(n)]$  и цены делений шкал  $S_n$  при числе рангов  $D=10$ :

$$\begin{aligned} f_1 &\rightarrow [-3, 17], d_1 = 2, \\ f_2 &\rightarrow [102, 32], d_2 = -7, \\ f_3 &\rightarrow [0, 1; 1, 1], d_3 = 0, 1, \\ f_4 &\rightarrow [10, 0], d_4 = -1, \\ f_5 &\rightarrow [-5, 5], d_5 = 1, \\ f_6 &\rightarrow [0, 4], d_6 = 0, 4. \end{aligned}$$



Ранжированная матрица оценок  $R_{4 \times 6}$ :

	1	2	3	4	5	6
1	3	1	1	9	20	10
2	1	7	9	10	6	1
3	5	10	4	1	1	6
4	10	7	10	10	10	8

*Второй шаг* – получение групповых оценок.

$f_n$	1	2	3	4	5	6
Веса критериев: $l_n$	3	2	2	3	1	1

Групповые оценки вариантов:

$$g_1 = 3 \cdot 3 + 2 \cdot 1 + 2 \cdot 1 + 3 \cdot 9 + 1 \cdot 2 + 1 \cdot 10 = 52, \quad g_1 = 8,7,$$

$$g_2 = 3 \cdot 1 + 2 \cdot 7 + 2 \cdot 9 + 3 \cdot 10 + 1 \cdot 6 + 1 \cdot 1 = 72, \quad g_2 = 12,$$

$$g_3 = 3 \cdot 5 + 2 \cdot 10 + 2 \cdot 4 + 3 \cdot 1 + 1 \cdot 1 + 1 \cdot 6 = 53, \quad g_3 = 8,7,$$

$$g_4 = 3 \cdot 10 + 2 \cdot 7 + 2 \cdot 10 + 3 \cdot 10 + 1 \cdot 10 + 1 \cdot 8 = 112, \quad g_4 = 18,67.$$

Исходя из усредненных групповых оценок вариант 4 можно считать предпочтительным из рассматриваемых вариантов сравнения. Однако по поводу усредненных групповых оценок можно сказать то же, что и о средней температуре больных в больнице или относительно средних оценок экспертов. Эти оценки нивелируют различия в функциях предпочтения, они отражают то общее, что есть в индивидуальных оценках, и сглаживают более тонкие отношения между различными критериями.

Задачу выбора предпочтительных вариантов можно решить другим способом – производя анализ предпочтений среди альтернативных вариантов путем их попарного сравнения. Сформулируем эту задачу выбора как задачу теории графов.

#### 4.4. Формализация задачи выбора предпочтительных объектов

Пусть  $V$  – множество объектов (вариантов) сравнения, причем  $|V| = M$  (количество объектов), а  $F$  – множество критериев (функций предпочтения), по которым можно оценивать эти объекты, и  $|F| = N$  (количество функций). Величины  $p_n$ , называемые оценками, отражают различие между элементами  $V$  по критерию  $f_n \in F$ ,  $n = \overline{1, N}$ , т. е. представляют собой отображение  $p_n$ :

$V \rightarrow f_n$ , которое позволяет выполнить сравнение по критерию  $f_n$ . Обозначим через  $F_n$  множество оценок, которые можно получить, исследуя элементы  $V$  с точки зрения критерия  $f_n$ . При анализе по всем критериям каждому элементу  $v \in V$  ставится в соответствие последовательность из  $N$  оценок, взятых из множеств  $F_1, \dots, F_N$ , т. е. с каждым элементом множества  $V$  сопоставляется элемент декартова произведения  $F' = F_1 \times F_2 \times \dots \times F_N$ .

Ставится задача выполнить сравнение объектов множества  $V$  с помощью многомерного состояния  $F' = F_1 \times F_2 \times \dots \times F_N$  с целью выбора одного или нескольких объектов, являющихся наилучшими с точки зрения множества критериев  $F$ . Представим решение задачи попарного сравнения объектов на языке теории графов.

Множество всех упорядоченных пар  $v, v' \in V$  обозначается  $V \times V$ . Подмножество  $B \subseteq V \times V$  называется бинарным отношением, т. е.  $(v, v') \in B$ . Выше мы рассмотрели соответствие между бинарным отношением и орграфом, при этом вершины графа соответствуют объектам сравнения  $v_i \in V$ , а дуги отражают бинарное отношение между объектами.

Напомним, что если  $(v, v') \in B$ , то в графе проводится дуга из  $v$  в  $v'$ , при этом  $v$  – начало, а  $v'$  – конец дуги, т. е. дуга определяется парой  $(v, v')$ . Последовательность дуг  $(v_i, v_{i+1}), i=1, 2, \dots, k-1$ , называется *путем*. Конечный путь, у которого  $v_1=v_k$ , образует контур. Дуга  $(v, v)$  называется *петлей*.

Как известно, бинарное отношение называется рефлексивным, если  $(v, v) \in B$  (т. е. граф с петлями); антирефлексивным, если  $(v, v) \notin B$  (т. е. граф без петель); симметричным, если  $(v, v') \in B \leftrightarrow (v', v) \in B$  (т. е. для каждой пары смежных вершин есть встречные дуги); антисимметричным, если  $(v, v') \in B$  и  $(v', v) \in B \rightarrow v=v'$  (т. е. вершины со встречными дугами рассматриваются как одна вершина); асимметричным, если  $(v, v') \in B \rightarrow (v', v) \notin B$  (т. е. граф без встречных дуг между парами вершин); транзитивным, если  $(v, v') \in B$  и  $(v', v'') \in B \rightarrow (v, v'') \in B$  (т. е. если любые три последовательные вершины связаны двумя последовательными дугами, то существует дуга между первой и третьей вершинами); линейным, если  $(v, v') \in B$  или  $(v', v) \in B$  (т. е. любые две вершины связаны хотя бы одной дугой).

Рассмотрим теперь множество критериев  $F$ . Каждый частный критерий  $f_n \in F$  задает направление возрастания предпочтения объектов, в соответствии с увеличением их оценок, определяемых в количественной или ранговой шкале, т. е. можно задать бинарное отношение предпоч-

тения (превосходства)  $P_f$ , которое связывает пару объектов множества  $V$ :  $(v, v') \in P_f$ , если оценка  $p(v) \leq p(v')$ .

Для каждого частного критерия  $f_n$  можно построить граф  $G_n = (V, U_n)$  (здесь  $U_n$  – множество всех дуг графа  $G_n$ ), в котором проводится дуга  $(v, v') \in U_n$ , если оценка  $p_n(v) < p_n(v')$ , т. е. стрелка дуги показывает на предпочтительный объект и проводятся встречные дуги  $(v, v')$ ,  $(v', v) \in U_n$ , если  $p_n(v) = p_n(v')$ . Граф  $G_n = (V, U_n)$  будет транзитивным и рефлексивным, но не будет асимметричным, так как между двумя вершинами могут быть встречные дуги.

Транзитивное и рефлексивное отношение  $P_f$  называется квазипорядком, а при условии, что любые два объекта сравнимы по  $P_f$ :  $(v, v') \in P_f$  или  $(v', v) \in P_f$ , оно называется линейным, или полным квазипорядком.

Введем отношение неразличимости  $N_f$  для объектов с  $p(v) = p(v')$ :  $(v, v') \in N_f$ . Это транзитивное, рефлексивное и симметричное отношение, называемое *эквивалентностью*, отражается на графе  $G_n$  контурами и порождает классы (подмножества) эквивалентных объектов  $K_i \subseteq V$ . Между классами эквивалентностей существует отношение строгого предпочтения:  $(v(K), v(K')) \subseteq P$ , если  $p(v(K)) < p(v(K'))$ . Эти отношения, называемые квазисериями, рефлексивны и транзитивны. Квазисерию  $P$ , для которой  $N_f$  является отношением равенства, т. е.  $(v, v') \in N_f \rightarrow v = v'$  называют серией. Серии соответствуют строгому упорядочению объектов.

Полный квазипорядок является полным порядком, или серией, тогда и только тогда, когда он антисимметричен, т. е. когда классы эквивалентностей сведены к одному объекту:

$$(v, v') \in N_f \text{ и } (v', v) \in N_f \rightarrow v = v'.$$

Таким образом, граф  $G_n$  воспроизводит отношение полного квази-порядка на множестве объектов  $V$  на основании критерия  $f_n$ .

Указав рядом с вершинами графа значения оценок объектов, полученных в количественной или ранговой шкале, можно отразить на графе всю информацию по сравнению объектов.

Совместив одноименные вершины графов  $G_n$ ,  $n = \overline{1, N}$ , получим мультиграф  $G_V$ , отражающий информацию для сравнения множества объектов  $V$  по множеству критериев  $F$ .

#### Пример

На основе ранжированной матрицы оценок  $R_{4 \times 6}$ , полученной в предыдущем примере для четырех объектов сравнения, на рис. 4.1 построены

частные графы  $G_n$  для критериев  $f_n$  ( $n=\overline{1,6}$ ) по четырем вершинам (объектам –  $v_1, v_2, v_3, v_4$ ), где числа при вершинах – это значения ранжированных оценок объектов сравнения. После совмещения вершин и наложения дуг получим мультиграф  $G_V$  (рис. 4.2,а), где числа при дугах показывают количество параллельных дуг данного направления.

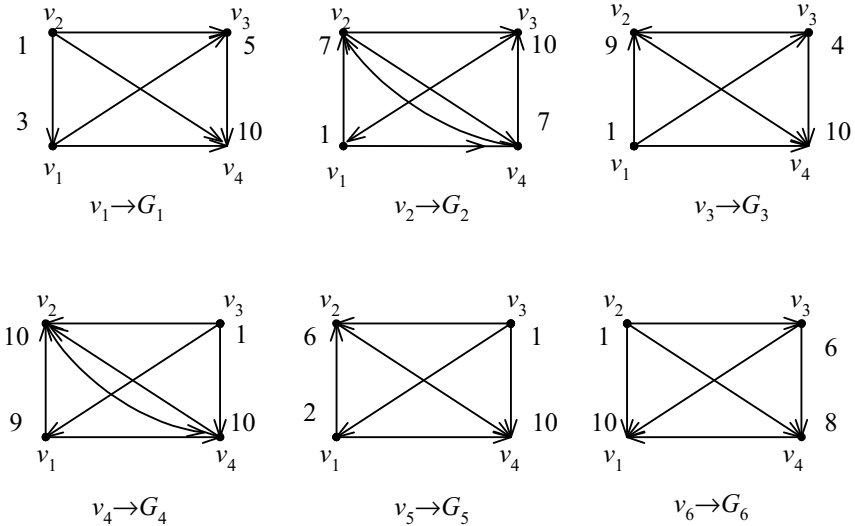


Рис.4.1. Частные графы  $G_n$

В мультиграфе  $G_V$  нарушается транзитивность отношения неразличимости, вследствие необходимости сравнивать по нескольким критериям. Необходимо ввести новое отношение, которое трансформировало бы мультиграф  $G_V$  в граф  $G=(V,U)$ , воспроизводящий квазипорядок на множестве  $V$  с учетом оценок по всем критериям.

Преобразование  $G_V \rightarrow G=(V,U)$  сводится к установлению правила, которое позволяет заменить множество дуг между вершинами  $v$  и  $v'$  в  $G_V$  к дуге  $(v,v') \in U$  или  $(v',v) \in U$  в графе  $G$ .

Самое простое правило преобразования  $G_V \rightarrow G$  сводится к принципу голосования (большинства), по которому в графе  $G$  проводится дуга  $(v,v')$ , если в графе  $G_V$  количество дуг  $(v,v')$  превышает число дуг  $(v',v)$ , а в случае равенства проводятся встречные дуги  $(v,v')$  и  $(v',v)$ , при этом веса дуг полагаются одинаковыми, т. е. игнорируется

количество параллельных дуг, важность критериев и величины оценок по разным критериям. Для учета последних параметров используются более сложные правила и алгоритмы преобразования  $G_V \rightarrow G=(V,U)$ .

Используя принцип голосования (большинства) как отношение превосходства, из графа  $G_V$  (рис. 4.2,а) можно легко получить граф  $G=(V,U)$  (рис. 4.2,б), который может быть не транзитивным, несмотря на то, что все исходные графы  $G_V$  транзитивны.

Граф  $G$  отражается булевой матрицей смежности  $B_{M \times M}$  (рис. 4.2,в), в которой элемент  $b_{ij}=1$ , если дуга  $(v,v') \in U$  (стрелка идет от  $v$  к  $v'$ ), т. е. вершина  $v'$  превосходит (более предпочтительна)  $v$ , в противном случае  $b_{ij}=0$ .

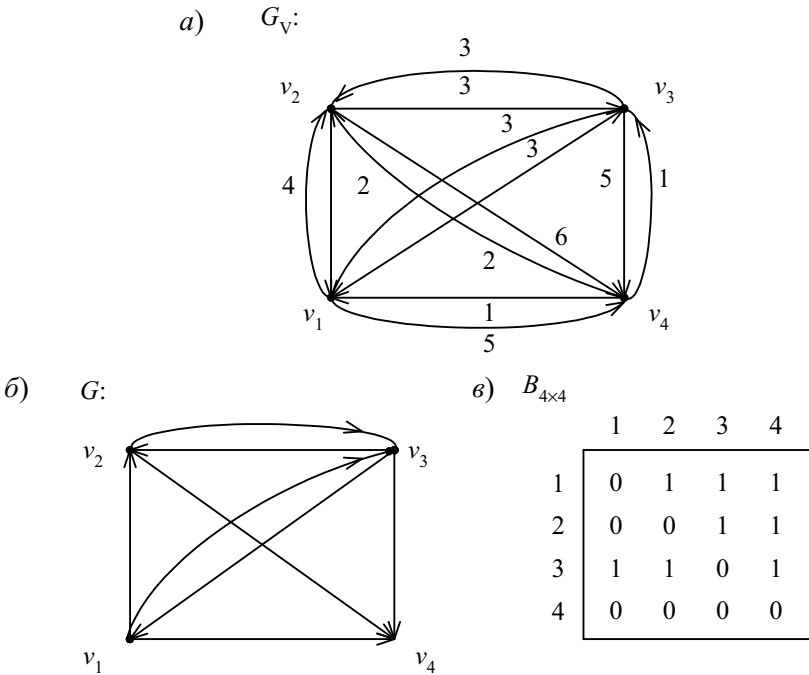


Рис. 4.2. Преобразование мультиграфа  $G_V \rightarrow G$ : а – мультиграф  $G_V$ ; б – граф  $G=(V, U)$ ; в – матрица смежности графа  $G$

Таким образом, в результате двух шагов обработки исходной матрицы оценок  $P_{M \times N}$  получена булева матрица  $B_{M \times M}$ , отражающая граф  $G$  превосходства (предпочтения) вершин (объектов сравнения).

## 4.5. Методы и алгоритмы выбора предпочтительных объектов

### 4.5.1. Метод и алгоритм поиска контуров в графе

Рассмотрим граф  $G$  как пару  $G = (V, \Gamma)$ , где  $\Gamma$  представляет многозначное отображение множества вершин  $V$  в себя, например, как было показано выше  $\Gamma_v$ , обозначает множество вершин-концов дуг, имеющих началом вершину  $v$ ;  $\hat{\Gamma}^+ v$  и  $\hat{\Gamma}^- v$  представляют транзитивное и обратное транзитивное замыкание для вершины  $v$ .

Отсутствие транзитивности в графе превосходства объектов сравнения  $G$  свидетельствуют о наличии контуров, объединяющих подмножества объектов, неразличимых по информации, содержащейся в графе, в соответствии с отношением неразличимости  $N_f$  для объектов с одинаковыми оценками  $p(v) = p(v') : (v, v') \in N_f$ . Это отношение  $N_f$  является также отношением эквивалентности (транзитивным, рефлексивным и симметричным).

Пусть  $K$  является контуром в графе  $G$ , тогда объекты  $v \in K$  можно рассматривать как эквивалентные, т. е. контуры образуют классы эквивалентностей  $(K_1, \dots, K_m)$  и представляются в графе как транзитивные, рефлексивные, симметричные подграфы, определенные выше (см. п. 1.4.8) как сильно связанные графы. Таким образом множество объектов  $V$  (вершин графа  $G$ ) можно разложить на классы  $K_1, \dots, K_m$ , каждый из которых есть контур, т. е. сильно связный граф.

Максимальным сильно связным подграфом  $G'$  графа  $G = (V, \Gamma)$  называется такой подграф, для которого не существует сильно связного графа  $G''$ , строго содержащего  $G'$ .

Легко показать, что подмножество классов  $\{K_i\}$ ,  $i = \overline{1, m}$ , упорядочиваются отношением  $R'$ : “существует путь из класса  $K_r$  в  $K_s$ ”. Отношение  $R'$  рефлексивно и транзитивно. Оно также асимметрично (т. е. в графе нет встречных дуг), так как если бы существовали пути как из  $K_r$  в  $K_s$ , так и из  $K_s$  в  $K_r$ , то  $K_r$  и  $K_s$  следовало бы объединить в один класс, что противоречит их максимальности.

Первый шаг упорядочения графа превосходства  $G = (V, \Gamma)$  состоит в нахождении контуров графа и рассмотрении их как отдельных объектов.

Метод поиска контуров состоит в разложении графа на классы эквивалентностей:

если  $K(v_i)$  – класс, содержащий вершину  $v_i$ , то

$$K(v_i) = \hat{\Gamma}v_i \cap \hat{\Gamma}^{-1}v_i,$$

т. е. контур для некоторой вершины  $v_i$  есть подмножество вершин, образуемое пересечением множеств транзитивного и обратного транзитивного замыканий для вершины  $v_i$ . В частности, класс  $K(v_i)$  может состоять и из одной вершины  $v_i$ .

Алгоритм метода включает следующие основные шаги.

1. Для обрабатываемых вершин выбираем начальную вершину  $v_i$  и определяем  $\hat{\Gamma}v_i$ ,  $\hat{\Gamma}^{-1}v_i$ ,  $K(v_i)$ .

2. Удаляем из дальнейшего анализа вершины из  $K(v_i)$  путем подавления их связей, для чего в матрице смежности необходимо обнулить строки и столбцы для вершин из класса  $K(v_i)$ .

3. Если есть еще необработанные вершины, то перейти на шаг 1.

#### Пример

Для графа  $G=(V,U)$ , где  $|V|=11$  (рис. 4.3) ниже представлен процесс получения контуров от начальных вершин. Результирующий вектор  $\mathbf{V}$  должен содержать номера контуров, при вершинах их образующих.

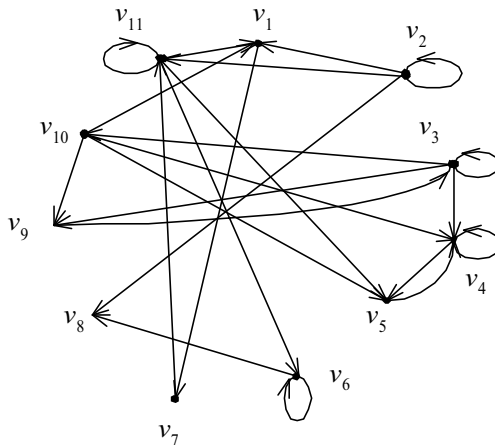


Рис. 4.3. Орграф G

Этап 1. Обработка исходной матрицы смежности  $S$ . Начальная вершина  $v_1$  отмечена значением 0 в векторах замыканий.

$\ S\ :$	1	2	3	4	5	6	7	8	9	10	11	$\hat{\Gamma}v_1$
1	0	0	0	0	0	0	1	0	0	0	0	0
2	1	1	0	0	0	0	0	1	0	0	1	-1
3	0	0	1	1	0	0	0	0	1	1	0	-1
4	0	0	0	1	1	0	0	0	0	0	0	4
5	0	0	0	1	0	0	0	0	0	0	0	3
6	0	0	0	0	0	1	0	1	0	0	0	3
7	0	0	0	0	0	0	0	0	0	0	1	1
8	0	0	0	0	0	0	0	0	0	0	0	4
9	0	0	1	0	0	0	0	0	0	0	0	-1
10	1	0	0	1	1	0	0	0	1	0	0	-1
11	1	0	0	0	1	1	0	0	0	0	1	2

$$\hat{\Gamma}^{-1}v_1 \quad \boxed{0 \quad 1 \quad 2 \quad -1 \quad -1 \quad -1 \quad 2 \quad -1 \quad 3 \quad 1 \quad 1}$$

Значением  $-1$  отмечены вершины, не вошедшие в замыкания. Остальные вершины входят в прямое и (или) обратное транзитивные замыкания от начальной вершины  $v_1$  в соответствии с алгоритмом п. 2.3.2:

$$\hat{\Gamma}v_1 = \{v_1, v_4, v_5, v_6, v_7, v_8, v_{11}\} \text{ (транзитивное замыкание - } Z\text{)}.$$

$$\hat{\Gamma}^{-1}v_1 = \{v_1, v_2, v_3, v_7, v_9, v_{10}, v_{11}\} \text{ (обратное транзитивное замыкание - } ZO\text{)}.$$

Вершины, вошедшие в оба замыкания, образуют контур  $K_1$  :

$$K_1(v_1) = \hat{\Gamma}v_1 \cap \hat{\Gamma}^{-1}v_1 = \{v_1, v_7, v_{11}\}.$$

В векторе  $\mathbf{V}$  в позициях вершин с номерами 1, 7, 11 заносится номер контура  $K_1$ , т. е. 1 :

Номера вершин графа:

$$\boxed{1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \quad 11}$$

Вектор номеров контуров  $\mathbf{V}$  :

$$\boxed{1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1}$$

В матрице смежности  $S$  обнуляются строки и столбцы с номерами 1, 7, 11, тем самым вершины с этими номерами исключаются из дальнейшего анализа. Если в векторе  $\mathbf{V}$  есть необработанные вершины (значение номера контура = 0), то продолжаютя этапы поиска контуров в графе.

Этап 2. Обработка матрицы  $S$  после подавления связей вершин  $\{v_1, v_7, v_{11}\}$ . Начальная вершина  $v_2 = 0$ .



$\ S\ :$	1	2	3	4	5	6	7	8	9	10	11	$\hat{\Gamma}v_1$
1	0	0	0	0	0	0	0	0	0	0	0	-1
2	0	1	0	0	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0	-1
4	0	0	0	1	1	0	0	0	0	0	0	-1
5	0	0	0	1	0	0	0	0	0	0	0	-1
6	0	0	0	0	0	1	0	1	0	0	0	-1
7	0	0	0	0	0	0	0	0	0	0	0	-1
8	0	0	0	0	0	0	0	0	0	0	0	1
9	0	0	1	0	0	0	0	0	0	0	0	-1
10	0	0	0	1	1	0	0	0	1	0	0	-1
11	0	0	0	0	0	0	0	0	0	0	0	-1

$$\hat{\Gamma}^{-1}v_1 \begin{bmatrix} -1 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

$\hat{\Gamma}v_1 = \{v_2, v_8\}$  (транзитивное замыкание – Z).

$\hat{\Gamma}^{-1}v_1 = \{v_2\}$  (обратное транзитивное замыкание – ZO).

Вершины, вошедшие в оба замыкания, образуют контур  $K_2$  :

$K_2(v_2) = \hat{\Gamma}v_2 \cap \hat{\Gamma}^{-1}v_2 = \{v_2\}$ , т. е. контур включает только одну вершину.

ну.

В векторе  $V$  в позиции вершины с номером 2 заносится номер контура  $K_2$ , т. е. 2 :

Номера вершин графа:

1	2	3	4	5	6	7	8	9	10	11
---	---	---	---	---	---	---	---	---	----	----

Вектор номеров контуров  $V$  :

1	2	0	0	0	0	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---

Аналогично получаются следующие классы эквивалентностей (рекомендуется проверить самостоятельно):

$K_3(v_3) = \{v_3, v_9, v_{10}\}$ ;  $K_4(v_4) = \{v_4, v_5\}$ ;  $K_5(v_6) = \{v_6\}$ ;  $K_6(v_8) = \{v_8\}$ .

В результате полной обработки матрицы смежности  $S$  можно получить вектор  $V$ , содержащий номера контуров в позициях вершин, включенных в данный контур:

Номера вершин графа:

1	2	3	4	5	6	7	8	9	10	11
---	---	---	---	---	---	---	---	---	----	----

Вектор номеров контуров  $V$ :

1	2	3	4	4	5	1	6	3	3	1
---	---	---	---	---	---	---	---	---	---	---

Рассмотренный выше процесс обработки матрицы смежности  $S$  и получения вектора контуров  $V$  можно представить в виде машинного алгоритма.

### 4.5.2. Описание машинного алгоритма поиска контуров в графе

Возможный вариант машинного алгоритма может включать следующие основные шаги.

1. Ввод матрицы смежности орграфа (S) и ее размера (n).
2. Обнуление V – вектора вершин как необработанных и NK-счетчика контуров.

3. Цикл  $i=1(1)n$  (по вершинам вектора V)
  - { если ( $V_i = 0$ ) (вершина не обработана)
    - { получение по подпрограмме для вершины  $V_i$  вектора транзитивного замыкания (Z);
    - транспонирование матрицы S (ST);
    - получение вектора обратного транзитивного замыкания (ZO);
    - $NK=NK + 1$ ; (получение номера контура)
4. цикл  $j=i(1)n$  (для определения вершин контура)
  - {если ( $Z_j \geq 0$  и  $ZO_j \geq 0$ ) (найдена вершина контура)
    - {  $V_j = NK$ ; (занесение номера контура в вектор V )
5. цикл  $m=1(1)n$  (подавление связей вершин контура)
  - {  $S_{mj} = 0$ ;  $S_{jm} = 0$ ; (обнуление строки и столбца  $j$  в S)
    - }
  - }
  - }
  - }
6. Вывод результатов: V, NK (вектор контуров и их количество).

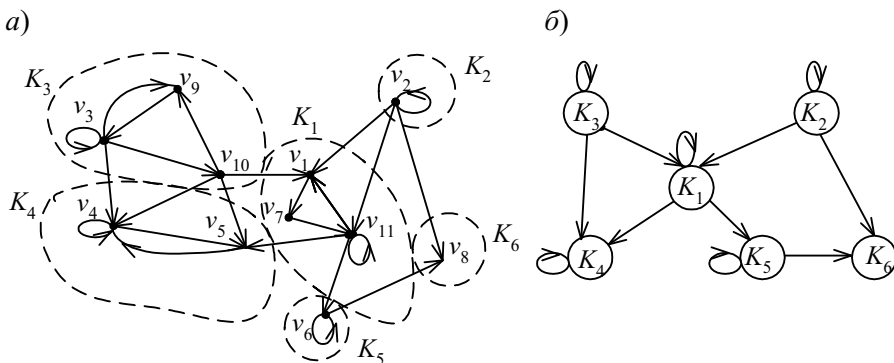
На основе полученного вектора контуров V можно изобразить орграф G с выделенными классами эквивалентностей ( контурами).

### 4.5.3. Построение графа контуров

Рис. 4.4,а показывает, как можно упорядочить классы эквивалентных объектов, образующих контуры, для графа на рис. 4.3. На рис. 4.4,б показан граф самих контуров как отдельных объектов, после удаления кратных дуг между контурами. Подобное представление контура в виде одного объекта называется *стягиванием* контура.

Граф контуров, полученный на основе отношения  $R'$ : “существует путь из класса  $K_r$  в  $K_s$ ”, является транзитивным, рефлексивным, асимметричным (не имеет встречных дуг), значит, он отражает частичный порядок, т. е. является квазисерией, а поскольку в силу отношения неразличимости

(эквивалентности) элементов контура  $((v,v') \in N_f$  и  $(v,v') \in N_f \rightarrow v=v')$  они сведены (стянуты) к одному объекту ( $K$ ), постольку граф контуров является также антисимметричным, т. е. отражает полный порядок, или серию.



**Рис. 4.4. Граф, разложенный на максимально сильно связанные подграфы (контуры): а – граф классов эквивалентных объектов; б – граф контуров**

Граф классов эквивалентностей является ациклическим, а для выявления в нем порядка, который не выглядит очевидным, существуют различные методы упорядочения вершин, которые будут рассмотрены ниже.

#### 4.6. Разбиение ациклического графа на уровни

Постановка задачи. В этой задаче осуществляется упорядочение вершин по уровням последовательностей дуг в ациклическом графе.

##### 4.6.1. Порядковая функция графа

Рассмотрим граф без контуров  $G=(V,\Gamma)$  и определим его подмножества  $N_0, N_1, \dots, N_r$  следующим образом:

$N_0 = \{ v_i \mid v_i \in V, \hat{\Gamma}^{-1}v_i = \emptyset \}$ , где символ  $\mid$  читается как “если”, т. е.  $N_0$  – это подмножество вершин  $V$ , если им не предшествует ни одна вершина (нет входящих дуг) ;

$N_1 = \{ v_i \mid v_i \in V - N_0, \hat{\Gamma}^{-1}v_i \subset N_0 \}$ , т. е.  $N_1$  – подмножество из оставшихся вершин-концов дуг, причем вершины-начала дуг, находятся в предшествующем множестве  $N_0$  ;

$N_2 = \{ v_i \mid v_i \in V - (N_0 \cup N_1), \hat{\Gamma}^{-1}v_i \subset N_0 \cup N_1 \}$ , т. е.  $N_2$  – подмножество из оставшихся вершин-концов дуг, причем вершины-начала дуг могут находиться в любых предыдущих множествах; наконец,

$N_r = \{ v_i \mid v_i \in V - \bigcup_{k=0}^{r-1} N_k, \hat{\Gamma}^{-1}v_i \subset \bigcup_{k=0}^{r-1} N_k \}$ , где  $r$  – такое наименьшее число, что  $\Gamma N_r = \emptyset$ , т. е. из вершин этого подмножества не исходит ни одна дуга (это висячие вершины).

Подмножества  $N_k, k=0, 1, 2, \dots, r$  образуют разбиение  $V$  и упорядочены отношением следования ( $\rightarrow$ ):

$$N_k \rightarrow N_{k'} \Leftrightarrow k < k'.$$

Порядковая функция графа без контуров ( $O(v)$ ) определяется равенством

$$v_i \in N_k \Rightarrow O(v_i) = k.$$

Другими словами, предлагается разбить множество вершин графа без контуров на непересекающиеся подмножества, упорядоченные так, что если вершина принадлежит подмножеству с номером  $k$ , то следующая за ней вершина входит в подмножество с номером, большим  $k$ .

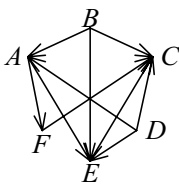
Подмножества этого разбиения называются уровнями. Они упорядочивают последовательность путей в графе.

### Пример

Алгоритм нахождения уровней ациклического графа и его порядковой функции рассмотрим на примере графа на рис 4.5,а.

Представим граф его матрицей смежности  $S$  и рассмотрим основные шаги алгоритма нахождения уровней графа (см.рис. 4.5,б).

а)



б)  $\|S\|$ :

A	0	0	0	0	1	1
B	1	0	1	0	1	0
C	0	0	0	0	0	0
D	1	0	1	0	1	0
E	0	0	1	0	0	0
F	0	0	1	0	0	0
	A	B	C	D	E	F

	A	B	C	D	E	F	
$C_0$	2	0	4	0	3	1	$N_0 = \{B, D\}$
$C_1$	0	-1	2	-1	1	1	$N_1 = \{A\}$
$C_2$	-1	-1	2	-1	0	0	$N_2 = \{E, F\}$
$C_3$	-1	-1	0	-1	-1	-1	$N_3 = \{C\}$

**Рис. 4.5. Выделение уровней ациклического графа:**  
а – исходный граф  $G$ ; б – матрица смежности и уровни графа

*Шаг 1.* Образует строку  $C_0$ , записывая для каждой вершины количество предшествующих ей вершин, для чего необходимо просуммировать значения в столбцах матрицы графа. В множество  $N_0$  войдут вершины  $B$  и  $D$ , которым не предшествует ни одна вершина (нули в соответствующих позициях строки  $C_0$ ).

*Шаг 2.* Заполняем строку  $C_1$  следующим образом : 1) в позициях строки  $C_1$ , в которых был 0 в строке  $C_0$  ( $B, D$ ), запишем  $-1$  (вершины обработаны, т. е. попали на свой уровень);

2) исключаем вершины предыдущего уровня  $N_0$  путем обнуления строк  $B$  и  $D$  матрицы  $S$ ;

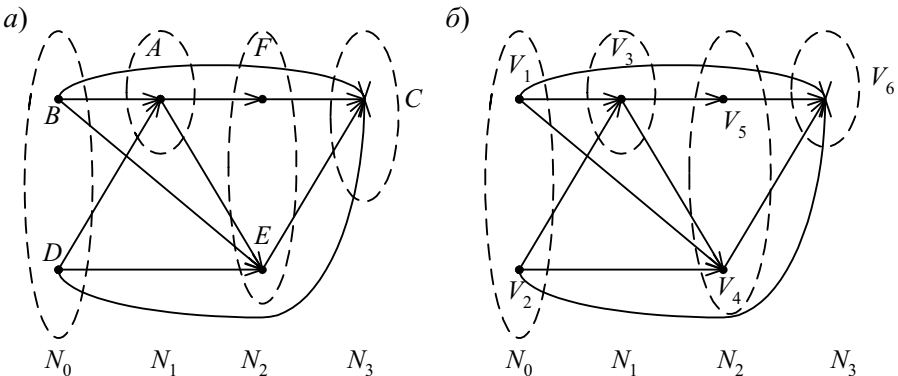
3) суммируем столбцы, кроме  $B$  и  $D$ . Вершины, которым соответствуют нули в строке  $C_1$ , составляют уровень  $N_1$ , в примере это вершина  $A$ .

*Шаг  $i$ .* Аналогично шагу 2 заполняем строку  $C_i$  и записываем уровни  $N_i$  до тех пор, пока возможно.

*Замечание.* Если граф содержит контур, то обязательно появится строка  $C_i$  без нулей. Отсюда следует, что описанный алгоритм дает возможность установить наличие контуров в графе.

На рис. 4.6,*a* показан исходный граф (рис.4.5,*a*), разбитый на уровни. Каждой вершине  $v_i$  этого графа соответствует некоторое  $N_k$ , т. е. некоторое  $k \in \{0,1,2,3\}$ . Эта порядковая функция  $v_i \rightarrow k$  задается таблицей:

Вершина	A	B	C	D	E	F
Уровень ( $k$ )	1	0	3	0	2	2



**Рис. 4.6. Граф, разбитый на уровни:**  
*a* – исходный граф; *b* – граф с новыми вершинами

Порядковая функция позволяет перенумеровать вершины так, что вершины уровня  $N_i$  имеют номера меньше, чем вершины уровня  $N_{i+1}$  (рис. 4.6,б.). Порядковая функция играет важную роль во многих комбинаторных задачах, например позволяет решить задачу выбора предпочтительных объектов.

#### 4.6.2. Описание машинного алгоритма разбиения графа на уровни

Возможный вариант алгоритма включает следующие шаги.

1. Ввод матрицы смежности ( $S$ ) графа и ее размера ( $n$ );
2.  $k=0$ ; (начальный номер уровня разбиения графа)  
 $v=0$ ; (счетчик обработанных вершин)
3. Цикл  $j=1(1)n$   
 $C[j]=0$ ; (обнуление вектора  $C$ )
4. Начало цикла по  $v$  (цикл по обрабатываемым вершинам)  
 $\{ k0=0$ ; (счетчик нулевых столбцов в цикле обработки вершин)
5. Цикл  $j=1(1)n$  (заполнение векторов  $C$  и  $N$ )
6.  $\{$  если  $(C[j] \neq -1)$  (вершина  $j$  не обработана )  
 $\{$   $sum=0$ ; (начало счета "1" для столбца)  
7. цикл  $i=1(1)n$   
 $sum = sum+S[i,j]$ ; (суммирование столбца  $j$ )  
8. если  $(sum \neq 0)$  (столбец  $j$  не пустой)  
 $C[j]=sum$ ; (заполнение  $C_j$ )  
9. иначе  
 $\{ C[j] = -1$  ; (вершина обработана)  
 $N[j]=k$  ; (запись уровня вершины)  
 $v = v+1$  ; (подсчет обработанных вершин)  
 $k0=k0+1$ ; (подсчет нулевых столбцов)  
 $\}$   
 $\}$  (конец цикла по  $j$ )
10. если  $(k0 \neq 0)$  (есть нулевые столбцы)
11.  $\{$  цикл  $i=1(1)n$  (исключение в  $S$  строк обработанных вершин)
12.  $\{$  если  $(N[i]=k)$  (вершина  $i$  попала на уровень  $k$ )
13. цикл  $j=1(1)n S[i,j]=0$ ; (обнуление строки  $i$ )  
 $\}$  (конец цикла по  $i$ )
14.  $k=k+1$ ; (получение нового номера уровня)

15. } (если нет нулевых столбцов)  
 иначе { вывод сообщения (“В графе есть контуры”);  
 переход на конец алгоритма – п.18 ;  
 }
16. } пока ( $v \leq n$ ); (условие продолжения цикла по  $v$  – п. 4)
17. Вывод  $N$ ; (вектор номеров уровней вершин)
18. Конец алгоритма.

#### 4.6.3. Порядковая функция классов эквивалентностей графа. Выделение предпочтительных объектов сравнения

Как показано в п. 4.6.1, граф без контуров можно упорядочить по уровням в соответствии с порядковой функцией. Рассмотрим классы эквивалентностей графа (контуры, максимально сильно связанные подмножества вершин). Как мы видели в п. 4.5.3, эти классы упорядочены и граф классов эквивалентностей (рис. 4.4,б) не имеет контуров, поэтому можно определить его уровни. Например, после применения алгоритма разбиения графа на уровни на рис. 4.7 показаны уровни для графа классов эквивалентностей (рис. 4.4,б).

Исходный граф на рис. 4.3 был построен на основе отношения предпочтения (превосходства – см. п. 4.4)  $P_f: (v, v') \in P_f$ , т. е.  $v'$  превосходит  $v$ , а граф контуров (рис. 4.4,б) был построен на основе отношения  $R'$ : “есть путь из  $K_r$  в  $K_s$ ”, значит, висячие вершины (не имеющие превосходящих вершин) графа на рис. 4.7 (т. е. классы  $K_4 = \{v_4, v_5\}$  и  $K_6 = \{v_8\}$ ) содержат вершины-объекты сравнения, которые являются предпочтительными в данном анализе и несравнимы между собой по информации, содержащейся в графе. Таким образом, задача выбора предпочтительных объектов из множества сравниваемых между собой по нескольким критериям решена, но это не единственный метод.

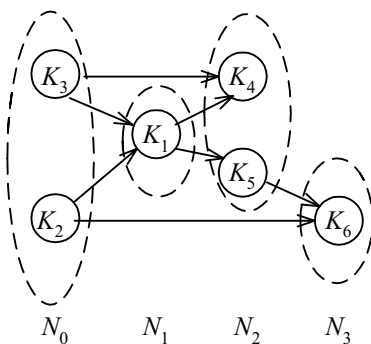


Рис. 4.7. Уровни графа классов эквивалентностей

## 4.7. Выделение ядер графа

В теории графов возможно использовать методы отыскания ядер графа  $G$ , состоящих из подмножества несравнимых между собой объектов, как основу для выделения наиболее предпочтительных объектов. Рассмотрим понятия и подходы, необходимые для решения этой задачи.

### 4.7.1. Внутренне устойчивое подмножество

Пусть задан граф  $G=(V,\Gamma)$ ; подмножество  $S \subset V$  называется внутренне устойчивым, если  $S \cap \Gamma S = \emptyset$ , т. е. никакие две вершины  $S$  не смежны (не связаны дугой).

Из определения следует, что вершина, имеющая петлю  $(v,v)$ , не может принадлежать внутренне устойчивому подмножеству, поэтому далее будут рассматриваться только графы без петель. Для графа с петлями можно рассматривать соответствующий граф без петель.

#### Пример

Рассмотрим граф на рис. 4.8. Подмножества  $S_1=\{A,D\}$ ,  $S_2=\{B,E\}$ ,  $S_3=\{A,C,D\}$  внутренне устойчивы. Проверим это, например, для  $S_1$ :

$$\Gamma A = \{B,E\}, \quad \Gamma D = \{E\},$$

$$\Gamma S_1 = \Gamma A \cup \Gamma D = \{B,E\},$$

$$S_1 \cap \Gamma S_1 = \{A,D\} \cap \{B,E\} = \emptyset.$$

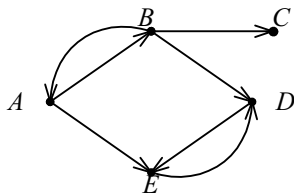


Рис. 4.8. Орграф без петель

Максимальное внутренне устойчивое подмножество. Это внутренне устойчивое подмножество, не являющееся собственным подмножеством никакого другого внутренне устойчивого подмножества. Например, в предыдущем примере  $S_1 \subset S_3$ , где  $S_3$  – максимальное внутренне устойчивое подмножество. Граф может обладать несколькими максимальными внутренне устойчивыми подмножествами.

### 4.7.2. Метод поиска семейства максимальных внутренне устойчивых подмножеств (метод Магу)

Этот метод использует свойства булевых уравнений. Пусть  $S$  – некоторое внутренне устойчивое подмножество. Любой вершине графа  $v_i \in V$  поставим в соответствие булеву переменную  $x_i$ , полагая, что, если  $v_i \notin S$ , то  $x_i = 0$  или  $\bar{x}_i = 1$ . Введем переменную  $\alpha_{ij}$ :

для  $j \neq i$ , если  $v_j \in \Gamma v_i$  (т. е. вершины смежные,  $v_j$  – конец дуги), то  $\alpha_{ij} = 1$ ,



если  $v_j \notin \Gamma v_i$  (т. е. вершины несмежные), то  $\alpha_{ij}=0$ ,  
 что соответствует матрице смежности.

Для любой пары вершин  $v_i$  и  $v_j$  (с учетом  $S \cap \Gamma S = \emptyset$ ) справедливо утверждение :

$$(i \neq j ; v_i \in \Gamma v_j \text{ или } v_j \in \Gamma v_i) \Rightarrow (v_i \notin S \text{ или } v_j \notin S),$$

(здесь *или* означает операцию неисключающее “или”), т. е. только один из концов дуги может входить в  $S$ , либо оба не входят. Это можно записать так для вершин, не входящих в  $S$  (при  $i \neq j$ ):

$$\begin{aligned} & (\overline{\alpha_{ij} \vee \alpha_{ji}}) \vee \bar{x}_i \vee \bar{x}_j = 1, \\ \text{или } & (\bar{\alpha}_{ij} \bar{\alpha}_{ji}) \vee \bar{x}_i \vee \bar{x}_j = 1. \end{aligned}$$

Выполнив булево умножение по всем вершинам графа, приходим к уравнению

$$\Phi_S = \prod_i \prod_{j \neq i} (\bar{\alpha}_{ij} \bar{\alpha}_{ji} \vee \bar{x}_i \vee \bar{x}_j = 1),$$

здесь  $\vee$  – знак дизъюнкции;  $\prod$  – знак булева умножения (каждый член учитывается один раз).

Имея в виду, что  $S \cap \Gamma^{-1} S = \emptyset$ , т. е. множество вершин из внутренне устойчивого подмножества и вне его не пересекаются, можно упростить это уравнение, заменив выражение  $\bar{\alpha}_{ij} \bar{\alpha}_{ji}$  на  $\bar{\alpha}_{ij}$

$$\begin{aligned} \Phi_S &= \prod_i \prod_{j \neq i} (\bar{\alpha}_{ij} \vee \bar{x}_i \vee \bar{x}_j) = 1, \\ \text{или } \Phi_S &= \prod_i \left( \bar{x}_i \vee \left( \prod_{j \neq i} (\bar{\alpha}_{ij} \vee \bar{x}_j) \right) \right) = 1. \end{aligned} \quad (4.1)$$

Для дальнейшего упрощения уравнения необходимо раскрыть скобки и привести подобные члены с учетом операции поглощения :  $x \vee x \cdot y = x$  и операции приведения подобных членов  $x \cdot x = x$ .

В результате упрощения получим, что для каждого члена уравнения  $\Phi_S$  совокупность всех вершин, соответствующих переменным, не представленным в нем, образует максимальное внутренне устойчивое подмножество графа.

### Пример

Рассмотрим булеву матрицу (рис.4.9) орграфа представленного на рис. 4.8.

	A	B	C	D	E
A	0	1	0	0	1
B	1	0	1	1	0
C	0	0	0	0	0
D	0	0	0	0	1
E	0	0	0	1	0

**Рис. 4.9. Булева матрица орграфа**

Для элементов матрицы  $\alpha_{ij} = 1$  получаем члены уравнения вида

$$\bar{\alpha}_{AB} \vee \bar{a} \vee \bar{b} = 0 \vee \bar{a} \vee \bar{b} = \bar{a} \vee \bar{b},$$

а для элементов матрицы  $\alpha_{ij} = 0$  :

$$\bar{\alpha}_{AC} \vee \bar{a} \vee \bar{c} = 1 \vee \bar{a} \vee \bar{c} = 1.$$

Поскольку члены уравнения вида  $(\bar{a} \vee \bar{b})$  и  $(\bar{b} \vee \bar{a})$  подобны, то уравнение  $\Phi_S$  с разными членами имеет вид

$$\Phi_S = (\bar{a} \vee \bar{b})(\bar{a} \vee \bar{e})(\bar{b} \vee \bar{c})(\bar{b} \vee \bar{d})(\bar{d} \vee \bar{e}) = 1.$$

Упрощая, получаем

$$\Phi_S = (\bar{a} \vee \bar{b}\bar{e})(\bar{b} \vee \bar{c}\bar{d})(\bar{d} \vee \bar{e}) = 1,$$

и, наконец,  $\Phi_S = \bar{a}\bar{b}\bar{d} \vee \bar{a}\bar{c}\bar{d} \vee \bar{b}\bar{e} = 1.$

Таким образом, граф на рис. 4.8 обладает тремя максимальными внутренне устойчивыми подмножествами :  $\{C,E\}$ ,  $\{B,E\}$ ,  $\{A,C,D\}$ .

#### 4.7.3. Внешне устойчивое подмножество

Пусть задан граф  $G=(V,\Gamma)$  ; подмножество  $T \subset V$  называется внешне устойчивым, если  $(\forall v \notin T) T \cap \Gamma v \neq \emptyset$ , т. е. вершина  $v$ , не принадлежащая  $T$ , связана по крайней мере с одной вершиной из  $T$  дугой, начало которой лежит в  $V-T$ . Это можно записать также в следующем виде:

$$(\forall v \in V) (\{v\} \cup \Gamma v) \cap T \neq \emptyset. \tag{4.2}$$

#### Пример

Для графа на рис. 4.8 подмножество  $\{C,D,E\}$  внешне устойчивое, что легко проверяется :  $T=\{C,D,E\}$ ,  $\Gamma A=\{B,E\}$ ,  $\Gamma B=\{A,C,D\}$ ,

$$T \cap \Gamma A = \{C,D,E\} \cap \{B,E\} = \{E\} \neq \emptyset,$$

$$T \cap \Gamma B = \{C,D,E\} \cap \{A,C,D\} = \{C,D\} \neq \emptyset.$$

Очевидно, что если  $T \subset T' \subset V$ , то  $T'$  внешне устойчивое подмножество. Любая висячая вершина  $v$  ( $\Gamma v = \emptyset$ , т. е. из нее не исходит дуга) принадлежит каждому внешне устойчивому подмножеству.

Минимальное внешне устойчивое подмножество не содержит строго никакого другого внешне устойчивого подмножества, например для графа на рис. 4.8 подмножество  $\{C, E\}$  внешне устойчивое и минимальное. Граф может обладать несколькими минимальными внешне устойчивыми подмножествами.

#### 4.7.4. Метод поиска семейства минимальных внешне устойчивых подмножеств (метод Магу)

Из условия (4.2) следует, что подмножество  $T$  должно содержать вместе с  $v_i$  по крайней мере одну из вершин  $\Gamma v_i$ . Следовательно, справедливо условие

$$(\forall v_i \in V) (v_i \in T \text{ или } (\exists v_j)(v_j \in T \text{ и } v_j \in \Gamma v_i)).$$

Пусть  $x_i = 1$ , если  $v_i \in T$  и  $\alpha_{ij} = 1$  ( $\alpha_{ij}$  определено выше), тогда

$$\prod_i \left( x_i \vee \left( \sum_i \alpha_{ij} x_j \right) \right) = 1,$$

Так как  $(\forall v_i) \left( x_i \vee \left( \sum_i \alpha_{ij} x_j \right) \right) = \sum_j \alpha_{ij} x_j$ ,

то 
$$\Phi_T = \prod_i \sum_j \alpha_{ij} x_j = 1. \tag{4.3}$$

При разложении  $\Phi_T$  после операций поглощения  $(x \vee x \cdot y = x)$  и приведения подобных членов каждый член этого разложения дает минимальное внешне устойчивое подмножество. Действительно, такой член не содержит переменных с отрицанием, и поэтому из множества вершин  $v_i$ , соответствующих переменным  $x_i$ , встречающихся в этом члене, нельзя удалить ни одну.

#### Пример

Рассмотрим граф (рис. 4.8) и его булеву матрицу (рис. 4.9).

Так, для 1-й строки матрицы можно получить выражение

$$\alpha_{AA} a \vee \alpha_{AB} b \vee \alpha_{AE} e = a \vee b \vee e.$$

Аналогичные выражения можно получить для остальных строк. В силу (4.3)

$$\Phi_T = (a \vee b \vee e) (b \vee a \vee c \vee d) c (d \vee e) = 1.$$

Упрощая, имеем

$$\Phi_T = (a \vee b \vee e) c (d \vee e) = 1,$$

или  $\Phi_T = acd \vee bcd \vee ce = 1.$

Таким образом, граф обладает тремя минимальными внешне устойчивыми подмножествами :

$$\{A,C,D\}, \{B,C,D\}, \{C,E\}.$$

#### 4.7.5. Ядра графа

Пусть задан граф  $G=(V,\Gamma)$ . Подмножество  $N \subset V$  называется ядром графа  $G$ , если  $N$  одновременно внутренне и внешне устойчивое подмножество, т. е.

$$(\forall v_i \in N) N \cap \Gamma v_i = \emptyset, \text{ (т. е. вершины } N \text{ не смежны);}$$

$$(\forall v_j \notin N) N \cap \Gamma v_j \neq \emptyset, \text{ (т. е. в } N \text{ входят концы дуг, начала которых вне } N).$$

Отсюда следует, что ядро содержит всякую вершину  $v_i$  с  $\Gamma v_i = \emptyset$  (т. е. висячую вершину) и не содержит вершин с петлями. Очевидно, что  $\emptyset$  не есть ядро графа.

Граф может обладать несколькими ядрами или вообще не иметь ядра.

#### 4.7.6. Поиск ядер графа (метод Магу)

Полагаем  $x_i = 1$ , если  $v_i \in N$  и рассмотрим уравнения  $\Phi_S = 1$  и  $\Phi_T = 1$ . Так как эти равенства должны выполняться, то

$$\Phi_N = \Phi_S \cdot \Phi_T = 1,$$

т. е. 
$$\left( \prod_i \left( \bar{x}_i \vee \prod_{j \neq i} (\bar{\alpha}_{ij} \vee \bar{x}_j) \right) \right) \left( \prod_i \sum_j \alpha_{ij} x_j \right) = 1,$$

или 
$$\prod_i \left( \bar{x}_i \sum_j \alpha_{ij} x_j \vee \bar{x}_i \prod_{j \neq i} (\bar{\alpha}_{ij} \vee \bar{x}_j) \right) = 1. \tag{3.4}$$

В результате решения уравнения  $\Phi_N$  получим ядра графа.

#### Пример

Для графа на рис. 4.8. согласно (3.4) по булевой матрице на рис.4.9 находим члены уравнения  $\Phi_N$  (замечая, что  $x\bar{x} = 0, x\bar{x} = x, x \vee x = x$ ) :

$$\bar{a} (a \vee b \vee e) \vee a (\bar{b} \bar{e}) = \bar{a} (b \vee e) \vee a (\bar{b} \bar{e}),$$

$$\bar{b} (a \vee b \vee c \vee d) \vee b (\bar{a} \bar{c} \bar{d}) = \bar{b} (a \vee c \vee d) \vee b (\bar{a} \bar{c} \bar{d}),$$

$$\bar{d} (d \vee e) \vee d \bar{e} = \bar{d} e \vee d \bar{e},$$

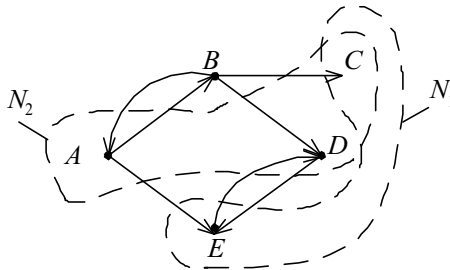
$$\bar{e} (d \vee e) \vee e \bar{d} = d \bar{e} \vee \bar{d} e.$$

$$\begin{aligned} \Phi_N &= (\bar{a} b \vee \bar{a} e \vee a \bar{b} \bar{e}) (\bar{b} a \vee \bar{b} c \vee \bar{b} d) c (\bar{d} e \vee d \bar{e}) = \\ &= (\bar{a} b \vee \bar{a} e \vee a \bar{b} \bar{e}) (\bar{b} c) (\bar{d} e \vee d \bar{e}) = \\ &= (\bar{a} \bar{b} c e \vee a \bar{b} c \bar{e}) (\bar{d} e \vee d \bar{e}) = \bar{a} \bar{b} c \bar{d} e \vee a \bar{b} c d \bar{e} = 1, \end{aligned}$$

т. е.  $\Phi_N = \bar{a} \bar{b} c \bar{d} e \vee a \bar{b} c d \bar{e} = 1$ .

Таким образом, граф обладает двумя ядрами:  $N_1 = \{C, E\}$  и  $N_2 = \{A, C, D\}$ , которые выделены на рис. 4.10.

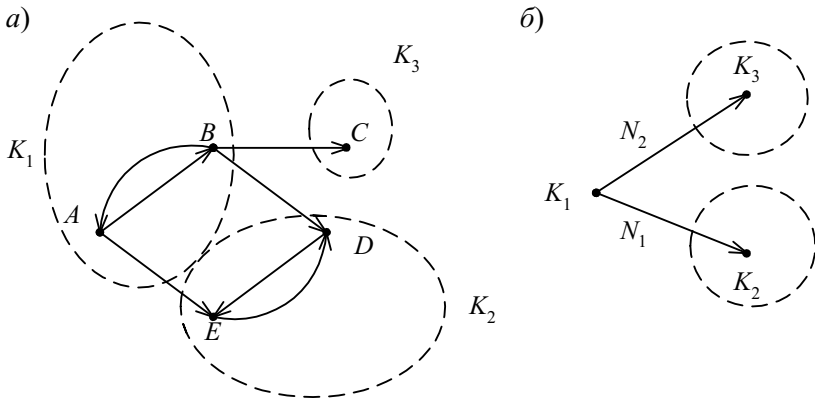
Из рис. 4.10 видно, что между вершинами ядра произвольного графа может нарушаться отношение превосходства:  $(v, v') \in U$ , если  $(v, v') \in P_f$ , т. е. если  $v'$  превосходит  $v$ , то существует дуга  $(v, v') \in U$ . Это происходит вследствие присутствия контуров в графе, например вершины контура  $\{E, D\}$ , принадлежат разным ядрам, а также из-за нарушения отношения транзитивности между вершинами ядер, например между вершинами  $A$  и  $C$  есть путь  $(A, B, C)$ . Поэтому поиск предпочтительных объектов с использованием ядер графа можно разбить на два этапа.



**Рис. 4.10. Граф с выделенными ядрами**

1 этап. В исходном графе осуществляется поиск классов эквивалентностей вершин (т. е. контуров) в соответствии с алгоритмом п. 4.5.2 и трансформация исходного графа в граф классов эквивалентностей, который является графом без контуров, т. е. ациклическим.

2 этап. В теории графов доказана следующая теорема: *граф без контуров всегда обладает ядром*. На основе этой теоремы осуществляется поиск ядер ациклического графа и выделение среди них



**Рис. 4.11. Поиск предпочтительных объектов:**  
**а** – выделение классов эквивалентностей вершин;  
**б** – выделение ядер графа

чих вершин. Этот процесс поиска предпочтительных объектов проиллюстрирован на рис. 4.11.

#### 4.7.7. Выделение предпочтительных объектов в ядрах графа

Рассмотрим способ, который вводит в процесс поиска ядер графа отношение превосходства, что позволяет объединить в один алгоритм этапы выделения контуров в графе и поиска предпочтительных объектов в ядрах графа путем стягивания контуров.

Пусть  $P_f$  отношение превосходства :  $(v, v') \in P_f \rightarrow (v, v') \in U$  в графе  $G=(V, U)$ . В дуге  $(v, v')$  вершину-конец дуги ( $v'$ ) будем называть оставляемой (превосходящей, предпочтительной), а вершину-начало дуги – исключаемой.

Пусть  $P$  – множество оставляемых вершин  $v'$ , а  $V-P$  – множество исключаемых вершин  $v$ . Множество  $P$  должно удовлетворять следующим двум свойствам:

1) внутренней устойчивости:

$\forall v \in P$  и  $\forall v' \in P \rightarrow (v, v') \notin U$ , т. е. никакой объект из множества  $P$  не превосходится другим объектом из  $P$ ;

2) внешней устойчивости :

$\forall v \in V-P \exists v' \in P \rightarrow (v, v') \in U$ , т. е. для любой из исключаемых вершин  $v$  существует среди оставляемых по крайней мере одна вершина  $v'$ , которая превосходит  $v$ .

Подмножество вершин, удовлетворяющих обоим свойствам, составляют ядра графа с учетом отношения предпочтения  $P_f$ .

Пусть  $e \in K$  (где  $K$  – контур) есть элемент (объект), рассматриваемый как эквивалентный другим объектам из контура  $K$  в графе  $G$ . Положим, что  $v \in V-K$  превосходит  $e$ , если существует  $e' \in K$  такой, что  $v$  превосходит  $e'$ , т. е.  $v$  превосходит  $K$  в целом, такое преобразование в теории графов называется *стягиванием контура  $K$* .

На основе введенного отношения превосходства разработан алгоритм, который позволяет вести поиск ядер графа совместно со стягиванием встречающихся контуров. Основные этапы алгоритма :

1. Решение задачи поиска контуров в графе и получение вектора  $V$  с номерами контуров.

2. В циклах просмотра контуров и вершин вне контура определение вершин вне контура, превосходящих хотя бы одну вершину контура, и исключение такого контура из вектора  $V$ .

Оставшиеся в векторе  $V$  вершины и контуры составляют ядра графа. Если ядер графа больше одного, то они несравнимы по информации, содержащейся в графе, поэтому необходим дополнительный анализ объектов сравнения.

#### 4.7.8. Описание машинного алгоритма поиска ядер графа

Возможный вариант алгоритма включает следующие шаги.

1. Ввод булевой матрицы смежности графа ( $S$ ) и ее размера ( $n$ ).  
 2. Вызов подпрограммы поиска контуров CONTUR ( $S, n, V, nk$ ),  
 где  $S, n$  – входные данные, а результаты:  $V$  – вектор номеров контуров в графе,  $nk$  – количество контуров.

3. Цикл  $k=1(1)nk$  (по номерам контуров)
4. { цикл  $i=1(1)n$  (выделение вершин контура)
5. { если  $(V_i=k)$  (вершина  $V_i$  входит в контур  $k$ )
6. { цикл  $j=1(1)n$  (выделение вершин вне контура)
7. { если  $(V_i \neq V_j \wedge S_{ij}=1)$  (вершина вне контура  $V_j$  превосходит вершину из контура  $V_i$ )
8. { цикл  $m=1(1)n$  (исключение вершин контура)
9. { если  $(V_m=k)$  (это вершина контура )  
 $V_m=0$ ; (исключение номера контура из вектора  $V$ )
10. } (конец цикла по  $m$ )

11.    переход на продолжение цикла по  $k$  ;
12.    }
13.    }    (конец цикла по  $j$ )
14.    }
15.    }    (конец цикла по  $i$ )
16.    }    (конец цикла по  $k$ )
17. Вывод вектора  $V$  ( номера контуров, образующих ядра графа).

### Библиографический список

1. *Кофман А.* Введение в прикладную комбинаторику. М.: Наука, 1975. 480 с.
2. *Белов В. В. и др.* Теория графов. М.: Высшая школа, 1976. 392 с.
3. *Басакер Р., Саати Т.* Конечные графы и сети. М.: Наука, 1974. 386 с.
4. *Прокушев Л. А.* Дискретная математика – 2: Методические указания по алгоритмизации задач теории графов/ ГУАП. СПб., 1998. 24 с.



## ОГЛАВЛЕНИЕ

Введение .....	3
1. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ И ПОНЯТИЯ .....	5
1.1. Элементы теории множеств .....	5
1.2. Основные понятия для графов .....	8
1.3. Основные понятия для неориентированных графов .....	10
1.4. Основные понятия для ориентированных графов .....	23
1.5. Реализация алгоритмов теории графов на ЭВМ .....	34
2. ЗАДАЧИ О ПУТЯХ В ОРГРАФЕ .....	36
2.1. Существование путей в орграфе .....	36
2.2. Пересчет путей в орграфе .....	38
2.3. Путь с наименьшим числом дуг. Поиск транзитивного замыкания вершин .....	40
3. ЗАДАЧИ ОБ ОПТИМАЛЬНЫХ РАССТОЯНИЯХ .....	44
3.1. Поиск минимальных путей в графе .....	46
3.2. Поиск максимального пути в графе без контуров .....	52
4. ЗАДАЧА ВЫБОРА ПРЕДПОЧТИТЕЛЬНЫХ ВАРИАНТОВ ИССЛЕДУЕМОЙ СИСТЕМЫ .....	53
4.1. Общая постановка задачи .....	53
4.2. Получение сравнимых оценок .....	54
4.3. Получение групповых оценок .....	55
4.4. Формализация задачи выбора предпочтительных объектов ....	57
4.5. Методы и алгоритмы выбора предпочтительных объектов .....	62
4.6. Разбиение ациклического графа на уровни .....	67
4.7. Выделение ядер графа .....	72
Библиографический список .....	80

Учебное издание

**Прокушев Лев Антонович**

**ДИСКРЕТНАЯ МАТЕМАТИКА**  
**Основы теории графов и алгоритмизации задач**

Учебное пособие

Редактор *Г. Д. Бакастова*

Компьютерная верстка *Ю. С. Бардуковой, А. Н. Колешко*

---

Лицензия ЛР № 020341 от 07.05.97. Сдано в набор 26.09.00. Подписано к печати 20.12.00.  
Формат 60×84 1/16. Бумага тип. №3. Печать офсетная. Усл. печ. л. 2,9. Усл. кр.-отг. 3,0.  
Уч. -изд. л. 3,15. Тираж 100 экз. Заказ №

---

Редакционно-издательский отдел  
Лаборатория компьютерно-издательских технологий  
Отдел оперативной полиграфии  
СПбГУАП

190000, Санкт-Петербург, ул. Б. Морская, 67