Практическое занятие 2 по дисциплине «Методы математического моделирования» на тему: Кратчайшие пути и их приложения

В процессе занятия студент должен:

1. Научиться формулировать задачи в терминах теории графов.
2. Ознакомиться с алгоритмами нахождения кратчайших путей в графах и сетях.
3. Уметь формулировать другие родственные задачи.
4. Выполнить задания в письменной форме.

Пусть $G=(V,E)$ – ориентированный граф, дугам которого приписаны веса $w(e)$, $e=(x,y)\in E$, интерпретируемые как длина (или стоимость) дуги. *Задача о кратчайшем пути* состоит в нахождении кратчайшего пути от заданной начальной вершины $s\in V$ до заданной конечной вершины $t\in V$ при условии, что такой путь существует, т.е. при условии, что вершина t достижима из вершины s.

Веса $w(e)$ могут быть положительными, отрицательными или нулевыми. Единственное ограничение состоит в том, чтобы в G не было циклов с суммарным отрицательным весом. Если такой цикл Φ все же существует и x_i – некоторая его вершина, то, двигаясь от s к x_i, обходя затем Φ достаточно большое число раз и попадая, наконец, в t, получим путь со сколь угодно малым весом. Таким образом, в этом случае кратчайшего пути не существует.

Если, с другой стороны, подобные циклы существуют, то их исключают из рассмотрения и находят кратчайшие гамильтоновы пути (т.е. пути проходящего через все другие вершины графа) из вершины s в концевую вершину t. Задача нахождения кратчайшего гамильтонова пути намного сложнее, чем задача о кратчайшем пути. Далее будем предполагать, что все циклы графа G имеют неотрицательный суммарный вес. Отсюда также вытекает, что рёбра графа G не могут иметь отрицательные веса.

Следующие задачи являются непосредственными обобщениями сформулированной задачи о кратчайшем пути.

1. Для заданной вершины $s\in V$ найти кратчайшие пути (пути минимальной длины) до всех остальных вершин графа.

2. Найти кратчайшие пути между всеми парами вершин.

Почти все методы, позволяющие решить задачу о кратчайшем пути от заданной вершины $s\in V$ до конечной вершины $t\in V$, дают также (в процессе решения) и все кратчайшие пути от s до $v\in V$, т.е. решение задачи 1. Задача 2 может быть решена либо многократным применением алгоритма для задачи 1, когда на каждом шаге в качестве начальной вершины s выбирают различные вершины, либо однократным применением специального алгоритма.

На практике часто требуется найти не только кратчайший путь, но и второй, третий и т.д. кратчайшие пути в графе, которые можно использовать при анализе «чувствительности» задачи о кратчайшем пути.

Рассмотрим несколько примеров задач о кратчайшем пути.

Выбор маршрута. Предположим, что надо проехать из Москвы в Рим, используя магистральные шоссейные дороги, соединяющие различные страны. Как выбрать кратчайший маршрут?

Построим граф, вершины которого соответствуют точкам соединения рассматриваемых дорог. Пусть каждая дуга графа соответствует шоссейной дороге, соединяющей пункты, представленные соответствующими вершинами. Пусть длина дуги определяется длиной (в километрах) соответствующего участка дороги. Теперь задача поиска оптимального маршрута движения между Москвой и Римом сводится к задаче отыскания в построенном графе кратчайшего пути между вершинами, соответствующими Москве и Риму.

Выбор маршрута авиаперелёта. Пассажир обращается за услугами к некоторой авиакомпании. Он желает попасть воздушным путём из Курска в столицу Турции Анкару, проведя в воздухе как можно меньше времени, поскольку полёт на самолёте вызывает у него чувство страха. Какой маршрут в данном случае должна предложить авиакомпания пассажиру?

Для того чтобы ответить на этот вопрос, необходимо построить некий граф. Вершины этого графа представляют аэропорты, через которые может проходить маршрут полёта, а дуги соответствуют полётам между определёнными аэропортами. Длиной каждой дуги следует считать время соответствующего полёта. Теперь авиакомпании необходимо в построенном графе найти кратчайший путь, соединяющий вершины, соответствующие Курску и Анкаре.

Выбор автомобиля. Предположим, что вам необходимо иметь в распоряжении автомобиль, на протяжении ближайших пяти лет до выхода на пенсию. В настоящий момент имеется большой выбор автомобилей, которые можно либо купить, либо взять напрокат. Автомобили имеют различные сроки эксплуатации и разную стоимость. Каким образом сделать выбор в такой ситуации?

Представим моменты времени возможных сделок на ближайший пятилетний период, связанных с покупкой или использованием напрокат автомобиля, вершинами графа. (Для упрощения в качестве моментов времени сделок рассматриваем лишь первые дни каждого месяца.) Изобразим факт приобретения автомобиля дугой, соединяющей вершину, соответствующую моменту покупки или началу использования напрокат, с вершиной, соответствующей моменту окончания срока службы или использования напрокат автомобиля. Пусть длина каждой дуги совпадает со стоимостью соответствующей сделки. Наилучшее сочетание решений, принятых в рассматриваем пятилетнем периоде, должно соответствовать кратчайшему пути в построенном графе между вершиной, представляющей начальный момент, и вершиной, отображающей момент ухода лица, принимающего решения, на пенсию.

Размещение капитала. Мелкий вкладчик решает вопрос о целесообразном размещении своего капитала в следующем году. Он располагает некоторым набором вариантов возможного размещения (может положить деньги на сберегательную книжку, вложить средства в депозитный сертификат, облигации и т.п.). Для упрощения предположим, что вклады могут производиться или изыматься в начале каждого месяца. Поставим в соответствие началу каждого месяца вершину графа. В этом графе каждому вкладу поставлена в соответствие дуга (x, y) в том случае, если вклад был сделан в месяце x, а срок его истекает в месяце y. Заметим, что указанный граф не содержит контуров. Пусть длина каждой дуги равна взятому с обратным знаком доходу от соответствующего вклада. Наилучшая стратегия вложений капитала соответствует кратчайшему пути (в данном случае пути, имеющему максимальное отрицательное значение длины), соединяющему вершину начала и конца рассматриваемого года. Данный пример почти полностью аналогичен выбору автомобиля, за исключением того, что в нём значения длин дуг графа являются не положительными, а отрицательными числами.

Все алгоритмы нахождения кратчайших путей основаны на принципе динамического программирования, который применительно к данной задаче означает, что путь $s,\ldots,x,\ldots,t$ является кратчайшим от вершины s к вершине t тогда и только тогда, когда путь $s,\ldots,x$ является кратчайшим от вершины s к вершине x, а путь $x,\ldots,t$ является кратчайшим от вершины x к вершине t.

Алгоритм Дейкстры применяется для решения задачи о кратчайшем пути при условии, что $w(e)\geq 0$ для всех $e\in E$. Алгоритм основан на присвоении вершинам меток. Первая часть метки $l(u)$ вершины u указывает текущее кратчайшее растояние от вершины s до вершины u, вторая – предшествующую вершину в текущем пути от вершины s до вершины u. Метки могут быть временными или постоянными. Постоянная метка не может изменяться в процессе выполнения алгоритма.

Для вершины $p \in V$ пусть $\Gamma(p) = \{x \in V : (p, x) \in E\}$.

Описание алгоритма Дейкстры

Шаг 1. Положить метку вершины s равной $(0, s)$ и считать эту метку постоянной. Для всех вершин $u \neq s$ положить метки равными (∞, s) и считать эти метки временными. За текущую рассматриваемую вершину с постоянной меткой взять вершину $p = s$.

Шаг 2. Для каждой вершины $u \in \Gamma(p)$ с временной меткой изменить метку в соответствии со следующим выражением: $l(u) = \min\{l(u), l(p) + w(p,u)\}$. При этом если первая часть метки изменилась, то изменить вторую часть метки, положив ее равной p.

Шаг 3. Среди всех вершин с временными метками найти вершину u с минимальной первой частью $l(u)$ метки.

Шаг 4. Считать метку вершины u постоянной и положить $p = u$.

Шаг 5 (а). (При нахождении пути из вершины s в вершину t.) Если $p = t$, то $l(t)$ является длиной кратчайшего пути из s в t. Алгоритм завершает работу. Если $p \neq t$, то перейти к шагу 2.

(б). (При нахождении путей от s ко всем остальным вершинам.) Если все вершины помечены постоянными метками, то эти метки дают длины кратчайших путей. Алгоритм завершает работу. Если некоторые метки являются временными, то перейти к шагу 2.

По завершении алгоритма первые части меток дают искомые кратчайшие расстояния. Сами кратчайшие пути можно получить с помощью рекурсивной процедуры, начиная с вершины t, для которой ищется кратчайший путь от s. Вторая часть метки вершины t указывает вершину u, непосредственно предшествующую ей в кратчайшем пути от s к t. Берем вторую часть метки вершины u и повторяем действия, пока не достигнем вершины s.

Сложность алгоритма Дейкстры равна $O(|V|^2)$.

Найдём кратчайшее расстояние от вершины s до вершины t в сети $G = (V, E)$ на рис.1.

Граф G является неориентированным, поэтому рассматриваем его рёбра как пары противоположно направленных дуг.

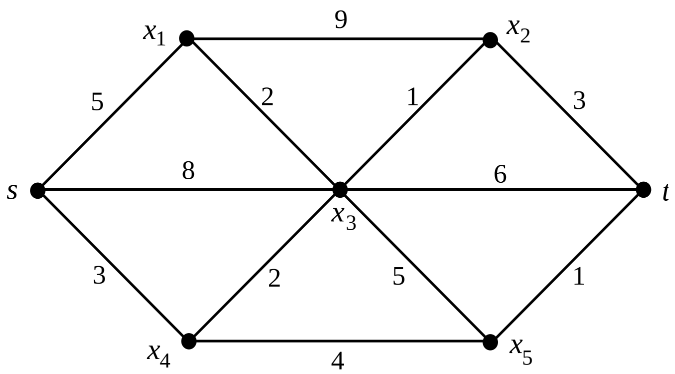


Рис.1. Сеть $G = (V, E)$

Метки вершин в процессе выполнения алгоритма приведены в табл.1.

Таблица 1

s	x_1	x_2	x_3	x_4	x_5	t
$(0, s)*$	(∞, s)	(∞, s)	(∞, s)	(∞, s)	(∞, s)	(∞, s)
	$(5, s)$	(∞, s)	$(8, s)$	$(3, s)*$	(∞, s)	(∞, s)
	$(5, s)*$	(∞, s)	$(5, x_4)$		$(7, x_4)$	(∞, s)
		$(14, x_1)$	$(5, x_4)*$		$(7, x_4)$	(∞, s)
		$(6, x_3)*$			$(7, x_4)$	$(11, x_3)$
					$(7, x_4)*$	$(9, x_2)$
						$(8, x_5)*$

Символом * отмечены постоянные метки.

В разобранном примере все вершины графа получили постоянные метки. Поэтому найдены кратчайшие расстояния от вершины s до всех остальных вершин графа.

Длина кратчайшего пути от вершины s к вершине t равна 8. Сам кратчайший путь от s к t строим по вторым частям меток. Получим путь $s \to x_4 \to x_5 \to t$.

Алгоритм Форда применяется для нахождения путей минимальной длины от вершины $s \in V$ до всех остальных вершин графа. Ограничений на длину дуг не накладывается. Алгоритм позволяет определить наличие контура отрицательной длины в графе.

Пронумеруем вершины исходного графа номерами от 1 до $n = |V|$, причём вершине s присвоим первый номер. Составим $(n \times n)$ -матрицу расстояний $D = (d_{ij})$ для исходной сети, положив:

$$d_{ij} = \begin{cases} 0, \text{ если } i = j; \\ w(x_i, x_j), \text{ если } (x_i, x_j) \in E; \\ \infty, \text{ если } (x_i, x_j) \notin E. \end{cases}$$

Так же как в алгоритме Дейкстры, будем присваивать вершине x_i метку $(l_k(x_i), x_k)$, где $l_k(x_i)$ – кратчайшее расстояние от вершины x_1 до вершины x_i , по всем путям, содержащим не более чем k дуг, x_k – вершина, предшествующая x_i , в кратчайшем пути, содержащим не более чем k дуг.

Описание алгоритма Форда

Шаг 1. Положить $k = 1$, $l_1(x_i) = d_{1i}$, вторую часть метки для всех вершин равной x_1 .

Шаг 2. Найти $l_{k+1}(x_i)=\min\{l_k(x_i), \min\limits_{1\le j\le n}\{l_k(x_j)+d_{ji}\}\}$ для всех i, $i=1,2,\ldots,n$. При $l_{k+1}(x_i)\ne l_k(x_i)$ вторая часть метки вершины x_i полагается равной вершине, на которой достигается $\min\limits_{1\le j\le n}\{l_k(x_j)+d_{ji}\}$.

Шаг 3. Если $l_{k+1}(x_i)\ne l_k(x_i)$ для некоторой вершины x_i и $k+1\le n$, то полагаем $k=k+1$ и возвращаемся к шагу 2.

Если $l_{k+1}(x_i)\ne l_k(x_i)$ для некоторой вершины x_i и $k+1=n+1$, то граф имеет контур отрицательной длины. Задача не имеет решения.

Если $l_{k+1}(x_i)=l_k(x_i)$ для всех i, $i=1,2,\ldots,n$, и $k+1\le n+1$, то метки $l_{k+1}(x_i)$ дают кратчайшие расстояния.

Сами кратчайшие пути строятся так же, как в алгоритме Дейкстры по вторым частям меток. Сложность алгоритма Форда равна $O(\ |V|^3)$.

Пример 1. Найдём кратчайшее расстояние от вершины x_1 до всех остальных вершин сети $G=(V,E)$, представленной на рис.2.

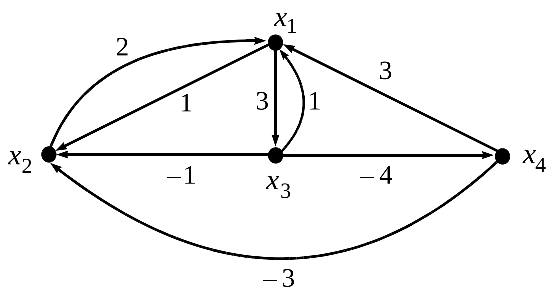


Рис.2. Сеть G

Матрица расстояний для заданной сети имеет вид

$$D=\begin{pmatrix} 0 & 1 & 3 & 3 \\ 2 & 0 & \infty & \infty \\ 1 & -1 & 0 & -4 \\ 3 & -3 & \infty & 0 \end{pmatrix}.$$

Метки вершин в процессе выполнения алгоритма приведены в табл.2.

Таблица 2

Первая итерация	$(0, x_1)$	$(1, x_1)$	$(3, x_1)$	$(3, x_1)$
Вторая итерация	$(0, x_1)$	$(0, x_4)$	$(3, x_1)$	$(-1, x_3)$
Третья итерация	$(0, x_1)$	$(-4, x_4)$	$(3, x_1)$	$(-1, x_3)$
Четвертая итерация	$(-2, x_2)$	$(-4, x_4)$	$(3, x_1)$	$(-1, x_3)$
Пятая итерация	$(-2, x_2)$	$(-4, x_4)$	$(1, x_1)$	$(-1, x_3)$

Метка вершины x_3 изменилась на пятой итерации. Следовательно, граф имеет контур отрицательной длины. Его легко построить по вторым частям меток: $x_3 \to x_4 \to x_2 \to x_1 \to x_3$. Задача не имеет решения. ►

Алгоритм Флойда применяется для нахождения кратчайшего пути для каждой пары вершин сети. Ограничений на длины дуг не накладывается. Алгоритм обнаруживает контур отрицательной длины.

Пронумеруем вершины графа номерами от 1 до $n=|V|$ и построим матрицу расстояний $D=(d_{ij})$ для исходной сети по аналогии с алгоритмом Форда.

Пусть d_{ij}^k – значение кратчайшего расстояния от вершины x_i к вершине x_j по всем путям, содержащим вершины с номерами не более k.

Описание алгоритма Флойда

Шаг 1. Положить $k=0$ и построить $D^0=(d_{ij}^0)$, $T^0=(t_{ij}^0)$, где $d_{ij}^0=d_{ij}$, $t_{ij}^0=t_{ij}$, $i, j=1, 2,\ldots, n$.

Здесь $T^0=(t_{ij}^0)$ – вспомогательная матрица, где $t_{ij}=j$, предназначенная для построения матрицы кратчайших путей.

Шаг 2. Построить матрицы $D^{k+1}=(d_{ij}^{k+1})$, $T^{k+1}=(t_{ij}^{k+1})$ по матрицам $D^k=(d_{ij}^k)$ и $T^k=(t_{ij}^k)$, вычисляя их элементы по формулам:

$$d_{ij}^{k+1}=\begin{cases} d_{ij}^k, \text{ если } i=k \text{ или } j=k; \\ 0, \text{ если } i=j; \\ \max\{d_{ij}^k;\ d_{ik}^k+d_{kj}^k\}, \text{ если } i\ne k,\ j\ne k, \end{cases}$$

$$t_{ij}^{k+1}=\begin{cases} t_{ij}^k, \text{ если } d_{ij}^{k+1}=d_{ij}^k; \\ k, \quad \text{если } d_{ij}^{k+1}\ne d_{ij}^k. \end{cases}.$$

Шаг 3. Если $k+1\le n$ и для некоторого i элемент $d_{ii}^{k+1}<0$, то граф имеет контур отрицательной длины. Задача не имеет решения.

Если $k+1<n$ и $d_{ii}^{k+1}\ge 0$, $i=1, 2,\ldots, n$, то положить $k=k+1$ и перейти к шагу 2.

Если $k+1=n$ и $d_{ii}^{k+1}\ge 0$, $i=1, 2,\ldots, n$, то матрица D^n даёт кратчайшие расстояния между парами вершин.

Кратчайший путь от вершины x_i к вершине x_j строится по элементам матрицы T^n. Элемент $t_{ij}^n=m$ указывает промежуточную вершину x_m в кратчайшем пути от вершины x_i к вершине x_j. Находим t_{im}^n и t_{mj}^n, которые указывают очередные промежуточные вершины. Если $t_{im}^n=m$ $(t_{mj}^n=j)$, то вершина x_m непосредственно следует за вершиной x_i (непосредственно предшествует вершине x_j). Процесс завершаем при получении номера вершины, непосредственно следующей за x_i, и номера вершины, непосредственно предшествующей x_j.

Сложность алгоритма Флойда равна $O(\ |V|^3)$.

Пример 2. Найдём кратчайшие расстояния между каждой парой вершин для сети (рис.3).

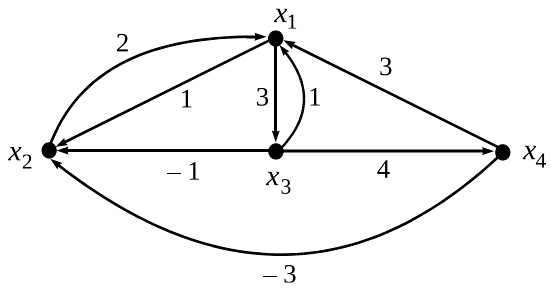


Рис.3. Сеть G

Решение. Согласно алгоритму Флойда получим матрицы:

$$D^0 = \begin{pmatrix} 0 & 1 & 3 & 3 \\ 2 & 0 & \infty & \infty \\ 1 & -1 & 0 & 4 \\ 3 & -3 & \infty & 0 \end{pmatrix}, \qquad T^0 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix};$$

$$D^1 = \begin{pmatrix} 0 & 1 & 3 & 3 \\ 2 & 0 & 5 & 5 \\ 1 & -1 & 0 & 4 \\ 3 & -3 & 6 & 0 \end{pmatrix}, \qquad T^1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 1 & 4 \end{pmatrix};$$

$$D^2 = \begin{pmatrix} 0 & 1 & 3 & 3 \\ 2 & 0 & 5 & 5 \\ 1 & -1 & 0 & 4 \\ -1 & -3 & 2 & 0 \end{pmatrix}, \qquad T^2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 2 & 2 & 2 & 4 \end{pmatrix};$$

$$D^3 = \begin{pmatrix} 0 & 1 & 3 & 3 \\ 2 & 0 & 5 & 5 \\ 1 & -1 & 0 & 4 \\ -1 & -3 & 2 & 0 \end{pmatrix}, \qquad T^3 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 2 & 2 & 2 & 4 \end{pmatrix};$$

$$D^4 = \begin{pmatrix} 0 & 0 & 3 & 3 \\ 2 & 0 & 5 & 5 \\ 1 & -1 & 0 & 4 \\ -1 & -3 & 2 & 0 \end{pmatrix}, \qquad T^4 = \begin{pmatrix} 1 & 4 & 3 & 4 \\ 1 & 2 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 2 & 2 & 2 & 4 \end{pmatrix}.$$

Матрица D^4 даёт значения кратчайших расстояний между вершинами. Например, кратчайшее расстояние между четвёртой и третьей вершинами равно 2. Сам кратчайший путь строим по матрице T^4. Получаем: $x_4 \to x_2 \to x_1 \to x_3$. ►

Сформулируем задачи, для решения которых могут быть использованы алгоритмы Дейкстры, Форда или Флойда.

1. Проектирование системы доставки почты. Крупное учреждение планирует разработать систему внутренней доставки почты, основанную на использовании линий пневматической связи, для распределения корреспонденции между восьмью отделами. Некоторые отделы будут только отсылать поступающую корреспонденцию в другие отделы, не имея при этом возможности получать почту. Все остальные отделы смогут отправлять и получать почту без каких-либо ограничений.

Расположение линий (рис.4) было установлено заранее. Каждый отдел представлен вершиной, а каждая линия – дугой. Ориентированные дуги соответствуют распределительным звеньям, которые могут быть использованы для посылки почты только в указанном направлении. Числа, приписанные дугам, равны расстояниям между соответствующими отделами. Для реализации поставленной цели необходимо найти кратчайшие пути между каждой парой рассматриваемой сети.

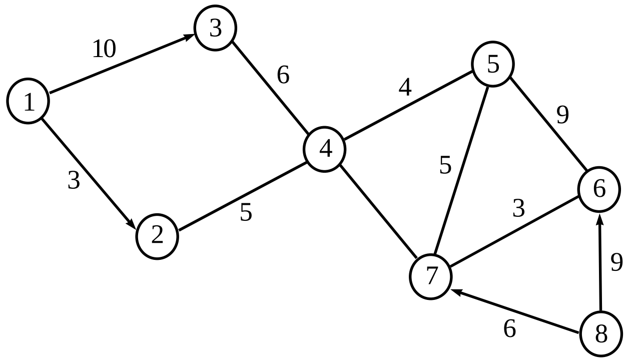


Рис.4. Система доставки почты

2. Задача о покупке автомобиля. При решении вопроса о том, когда следует производить покупку нового автомобиля, необходимо учитывать его стоимость и возрастающие эксплуатационные расходы.

Решим задачу, в которой требуется определить, как часто в течение восьмилетнего периода следует заменять автомобиль, чтобы минимизировать при этом общие затраты. Предполагается, что решение о покупке нового автомобиля может приниматься в начале каждого года, исходя из затрат на его приобретение, эксплуатационных расходов на период, в течение которого автомобиль будет использоваться, и ликвидационной стоимости машины в момент, когда она заменяется на новую. Предположим, что в начальный момент человек не имеет автомобиля и что он собирается покупать новый автомобиль, по крайней мере, каждые четыре года.

Сеть для рассматриваемой задачи изображена на рис.5. Началу каждого года соответствует вершина. Если покупка автомобиля производится в начале i -го года, а в начале j -го года автомобиль заменяется на новый, то данному варианту замены соответствует дуга (i, j). Пусть общие затраты в данном случае равны c_{ij}. Тогда значение параметра дуги (i, j) вычисляется по формуле

$$c_{ij} = P_i + \sum_{k=1}^{j-1} m_k - S_j ,$$

где P_i – стоимость автомобиля в начале i -го года; m_k – эксплуатационные расходы в течение k -го года; S_j – ликвидационная стоимость автомобиля в начале j -го года.

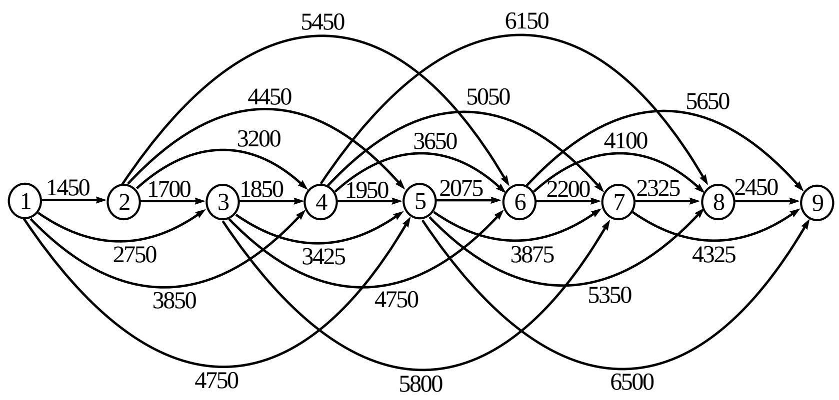


Рис.5. Покупка нового автомобиля

На рис.5 числа, приписанные дугам (i, j), соответствуют планируемым затратам c_{ij}. Вследствие инфляции, возрастания стоимости автомобиля и увеличения расходов на его техническое обслуживание общие затраты в различных периодах одинаковой длины не являются постоянными. Оптимальному решению данной задачи соответствует кратчайший путь из вершины 1 в вершину 9. Эту цепь можно рассматривать как цепь, которая минимизирует стоимость единицы потока из вершины 1 в вершину 9 при условии, что стоимость единицы потока по дуге (i, j) равна c_{ij}. Для минимизации общих затрат покупку автомобиля следует производить в начале первого, пятого и девятого годов. Общие затраты при этом составляют 11 250 долл.

3. Составление маршрута перегона вагонов. Среди общих расходов на транспортировку груза, поступающего в морской порт Владивостока и вывозимого из него, значительную долю составляют затраты на перегон вагонов из одного пункта портового комплекса в другой. В сети (рис.6) вершина 1 соответствует станции прибытия поездов, из которой вагоны прибывающих составов перегоняются к различным пунктам назначения в портовом комплексе, обозначенным остальными вершинами сети. Числа, приписанные дугам, представляют собой затраты на перегон вагонов по соответствующей ветке.

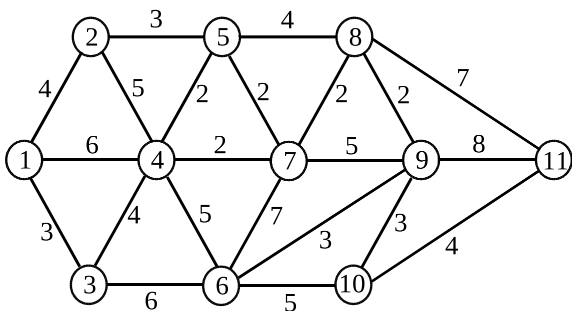


Рис.6. Транспортная сеть

П р и м е р 3. Поезд состоит из 200 вагонов, 100 из которых хопперы, груженные зерном. Хопперы с зерном должны быть доставлены к элеваторам, расположенным в вершинах 3, 6 и 10. После разгрузки они могут быть либо доставлены на заводы, производящие удобрения (вершины 2, 5 и 8), либо перевезены на станцию отправления (вершина 11). Остальные типы вагонов, такие, как крытые товарные вагоны, вагоны-платформы и цистерны, следует доставить в пункты 4, 7 и 9, разгрузить их там и перегнать на станцию отправления.

Решение. Применение алгоритма Флойда транспортной сети (см. рис.6) позволит минимизировать затраты на перегон вагонов из каждого пункта портового комплекса в любой другой пункт. Например, решение данной задачи позволит определить оптимальный путь перевозки хопперов, соединяющий станцию прибытия, элеваторы, заводы, производящие удобрения, и станцию отправки.

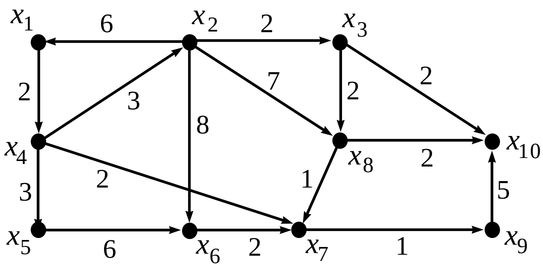
На рис.6 вершина 1 соответствует станции прибытия, вершины 3, 6, 10 – элеваторам, вершины 2, 5, 8 – заводам, производящим удобрения, узлы 4, 7, 9 – другим пунктам портового комплекса, а вершина 11 – станции отправления. ►

4. Задача о k кратчайших путях. При решении некоторых практических задач, связанных с коммуникационными или транспортными сетями, требуется найти не один, а несколько кратчайших путей, расположенных в порядке возрастания их длин. Например, знание нескольких кратчайших путей в сети улиц позволит планировщикам транспортных сетей моделировать потоки движения автомобилей по улицам городов. При составлении маршрута передачи информации по коммуникационной сети, когда некоторые пути временно заняты, могут быть использованы сведения о наилучших из возможных альтернативных вариантов.

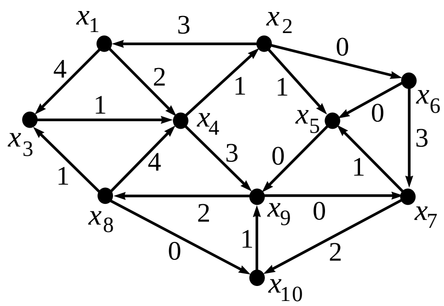
Задачи для самостоятельного решения.

В задачах 1 и 2 найти кратчайший путь от вершины x_1 до вершины x_7.

1.

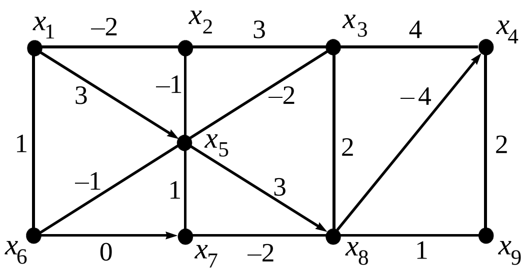


2.

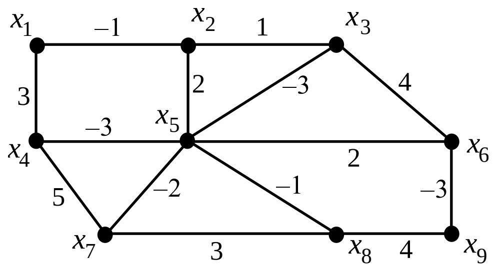


В задачах 3 и 4 найти кратчайшие расстояния от вершины x_1 до всех остальных вершин графа.

3.



4.



5. Определить кратчайшие расстояния между каждой парой вершин для графов со следующими матрицами расстояний:

а) $\begin{pmatrix} 0 & -15 & 15 & \infty & \infty \\ 20 & 0 & 7 & 1 & \infty \\ 8 & \infty & 0 & -10 & -3 \\ \infty & 2 & \infty & 0 & 6 \\ \infty & \infty & 14 & 4 & 0 \end{pmatrix}$; б) $\begin{pmatrix} 0 & 11 & 2 & 8 & 11 & 11 \\ 11 & 0 & 5 & \infty & \infty & 1 \\ \infty & 5 & 0 & \infty & 2 & 1 \\ 2 & \infty & \infty & 0 & 2 & \infty \\ \infty & 9 & \infty & 2 & 0 & 7 \\ \infty & 1 & \infty & \infty & 7 & 0 \end{pmatrix}$;

в) $\begin{pmatrix} 0 & 1 & \infty & 5 & 9 & -3 \\ 1 & 0 & -1 & 7 & 6 & \infty \\ \infty & -1 & 0 & \infty & 7 & 5 \\ 5 & 7 & \infty & 0 & 6 & -2 \\ 9 & 6 & 7 & 6 & 0 & -4 \\ -3 & \infty & 5 & -2 & -4 & 0 \end{pmatrix}$; г) $\begin{pmatrix} 0 & 3 & 2 & 7 & 2 & \infty & \infty \\ 3 & 0 & 1 & -1 & 1 & 7 & \infty \\ 2 & 5 & 0 & 9 & \infty & \infty & -3 \\ 7 & 1 & \infty & 0 & 1 & \infty & 3 \\ 2 & 1 & \infty & 1 & 0 & \infty & \infty \\ \infty & \infty & 2 & \infty & 5 & 0 & 2 \\ \infty & \infty & 3 & \infty & \infty & 2 & 0 \end{pmatrix}$.