

Лабораторная работа №2.

Динамическое программирование

Цель работы

Изучить теорию и методы решения задач динамического программирования; приобрести навыки решения задач динамического программирования на ЭВМ.

Методические указания

Динамическое программирование представляет собой математический метод для нахождения оптимальных решений многошаговых (многоэтапных) задач оптимизации.

1. Многошаговые процессы в динамических задачах Некоторые задачи математического программирования обладают специфическими особенностями, которые позволяют свести их решение к рассмотрению множества более простых «подзадач». В результате вопрос о глобальной оптимизации целевой функции сводится к поэтапной оптимизации некоторых промежуточных целевых функций.

Пусть, например, на период времени, состоящий из m лет, планируется деятельность группы промышленных предприятий. В начале планируемого периода на развитие предприятий выделяются основные средства Q_0 , которые необходимо распределить между предприятиями. В процессе функционирования предприятий выделенные им средства частично расходуются. Однако каждое из этих предприятий за определенный период времени (хозяйственный год) получает прибыль, зависящую от объема вложенных средств. В начале каждого года имеющиеся средства могут перераспределяться между предприятиями. Требуется определить, сколько средств надо выделить каждому предприятию в начале каждого года, чтобы суммарный доход от всей группы предприятий за весь период времени был максимальным.

Эта задача является многошаговой. Шагом управления здесь будет хозяйственный год. Управление процессом состоит в перераспределении средств в начале каждого хозяйственного года.

Обычно методами динамического программирования оптимизируют работу управляемых систем, эффект которой оценивается аддитивной целевой функцией. Аддитивной называется функция многих переменных $f(x_1, x_2, \dots, x_n)$ вида

$$f(x_1, x_2, \dots, x_n) = \sum_{j=1}^n f_j(x_j), \quad (1)$$

где каждая функция f_j зависит только от одной переменной x_j . Слагаемые аддитивной целевой функции соответствуют эффекту решений, принимаемых на отдельных этапах управляемого процесса.

Отметим, что есть многошаговые задачи, естественным образом распадающиеся на отдельные этапы, но имеются и задачи, в которых разбиение приходится вводить искусственно.

1. Принцип оптимальности и рекуррентное соотношение

Динамическое программирование как научное направление возникло и сформировалось в 1951–1953 гг. благодаря работам Р. Беллмана и его сотрудников. Метод динамического программирования позволяет одну задачу со многими переменными заменить рядом последовательно решаемых задач с меньшим числом переменных. Процесс решения задачи разбивается на шаги. При этом если задано начальное состояние управляемой системы, то нумерация шагов осуществляется от конца к началу, а если конечное, то – от начала к концу.

Основным принципом, на котором базируется оптимизация многошагового процесса и особенности вычислительного метода динамического программирования, является принцип оптимальности Р. Беллмана.

Приведем его формулировку: оптимальное поведение обладает тем свойством, что каковы бы ни были начальное состояние и начальное решение, последующие решения должны составлять оптимальное поведение относительно состояния, полученного в результате начального решения.

Этот принцип можно сформулировать и по-другому: оптимальное поведение в многошаговом процессе обладает тем свойством, что какими бы ни было решение, принятое на последнем шаге, и состояние процесса перед последним шагом, предыдущие решения должны составлять оптимальное относительно этого состояния поведение.

Принцип оптимальности имеет конструктивный характер и непосредственно указывает процедуру нахождения оптимального решения. Математически он записывается выражением вида

$$f_{n-l}(S_l) = \text{optimum}_{U_{l+1}} [R_{l+1}(S_l, U_{l+1}) + f_{n-l-1}(S_{l+1})], \quad (2)$$

$l = 0, 1, \dots, n-1$, где $U_l = (u_l^{(1)}; \dots; u_l^{(m)})$ - решение (управление), выбранное на l -м шаге; $S_l = (s_l^{(1)}; \dots; s_l^{(m)})$ - состояние системы на l -м шаге; R_l - непосредственный эффект, достигаемый на l -м шаге; f_{n-l} - оптимальное значение эффекта, достигаемого за $n-l$ шагов; n - количество шагов (этапов). «Optimum» в выражении (2) означает максимум или минимум в зависимости от условия задачи. Формула (2) носит название уравнения Беллмана или рекуррентного соотношения. Процесс вычисления f_{n-l} , $l = 0, \dots, n-1$, осуществляется при естественном начальном условии $f_0(S_n) = 0$, которое означает, что за пределами конечного состояния системы эффект равен нулю.

Сформулированный принцип и уравнение Беллмана носят общий характер и применяются не только в задачах дискретной оптимизации. Метод динамического программирования широко используется для решения многих экономических задач, связанных с планированием производственных программ, оптимальным распределением ресурсов (денежных средств, рабочей силы, сырья и т.д.).

Поскольку многие особенности реализации метода динамического программирования определяются конкретными задачами, не имеет смысла подробно описывать вычислительный алгоритм в общем случае. Поясним метод на примере одной из наиболее характерных задач - задаче о кратчайшем маршруте.

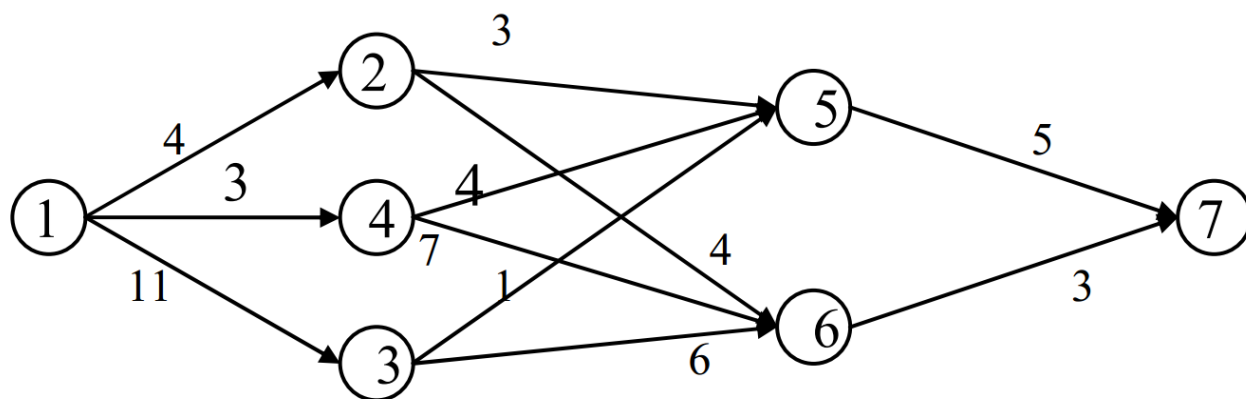


Рис. 1: Граф транспортной сети

Пусть требуется перевезти груз из города в город. Сеть дорог, связывающих эти города, изображена в виде графа на рис. (1). Вершинам графа поставлены в соответствие города, а дугам - транспортные магистрали. Стоимость перевозки груза из города s ($s = 1, \dots, 6$) в город j ($j = 2, \dots, 7$) проставлена над соответствующими дугами графа. Необходимо найти маршрут, связывающий города 1 и 7, для которого суммарные затраты на перевозку груза были бы наименьшими.

Для решения задачи разобьем все множество вершин (городов) на подмножества. В первое подмножество включим исходную вершину 1. Во второе - вершины, в которые входят дуги, выходящие из вершины 1. В третье - вершины, в которые входят дуги, выходящие из вершин второго подмножества. Таким образом, продолжая разбиение дальше, получим четыре подмножества: 1, 2, 3, 4, 5, 6, 7. Очевидно, что любой маршрут из города 1 в город 7 содержит ровно три дуги, каждая из которых связывает вершины, принадлежащие соответствующим подмножествам. Следовательно, процесс решения задачи (нахождения оптимального маршрута) разбивается на три этапа. На первом этапе принимается решение о том, через какой город, принадлежащий второму подмножеству, везти груз из города 1. На втором этапе необходимо определить, через какой город третьего подмножества везти груз из некоторого города, принадлежащего второму подмножеству. На последнем (третьем) этапе формируется оптимальный маршрут.

Перенумеруем этапы от конечной вершины графа к начальной и введем обозначения: n - номер шага ($n = 1, 2, 3$); $f_n(s)$ - минимальные затраты на перевозку груза от города s до конечного города, если до конечного города осталось n шагов; $j_n(s)$ - номер города, через который надо ехать из города s , чтобы затраты были минимальными $f_n(s)$; $c_{s,j}$ - стоимость перевозки груза из города s в город j .

Здесь все обозначения несут важную смысловую нагрузку: f - это целевая функция, s - состояние системы (номер города), индекс n - динамическая информация о том, что из города s до конечного города осталось n шагов.

Предположим, что груз доставлен в город 7, следовательно, число оставшихся шагов равно нулю ($n = 0$) и $f_n(s) = f_0(7) = 0$, так как из города 7 груз везти не надо.

Рассмотрим последний шаг ($n = 1$) и вычислим для него значение функции. Очевидно, что в город 7 груз может быть доставлен из города 5 или из города 6. Вычислим затраты на перевозку для этих состояний:

$$f_1(5) = c_{5,7} + f_0(7) = 5 + 0 = 5, s = 5, j_1(5) = 7; \quad (3)$$

$$f_1(6) = c_{6,7} + f_0(7) = 3 + 0 = 3, s = 3, j_1(6) = 7. \quad (4)$$

Чтобы произвести расчет для $n = 2$, выдвинем гипотезы о месте нахождения груза: 1-я гипотеза – груз находится в городе 2; 2-я гипотеза – груз находится в городе 3; 3-я гипотеза – груз находится в городе 4.

Из города 2 в город 7 можно перевезти груз или через город 5, или через город 6. Поэтому оптимальный маршрут из города 2 найдем по выражению

$$f_2(2) = \min[c_{2,5} + f_1(5), c_{2,6} + f_1(6)] = \min[3 + 5, 4 + 3] = 7. \quad (5)$$

Здесь $s = 2$ и $j_2(2) = 6$, т. е. условно-оптимальный маршрут проходит через город 6. Аналогично для $s = 3$ и $s = 4$:

$$f_2(3) = \min[c_{3,5} + f_1(5), c_{3,6} + f_1(6)] = 6, j_2(3) = 5; \quad (6)$$

$$f_2(4) = \min[c_{4,5} + f_1(5), c_{4,6} + f_1(6)] = 9, j_2(4) = 5. \quad (7)$$

Вычисления для третьего шага ($n = 3$) показывают, что

$$f_3(1) = \min[c_{1,2} + f_2(2), c_{1,3} + f_2(3), c_{1,4} + f_2(4)] = 11, \quad (8)$$

$j_3(1) = 2$, т.е. минимальные затраты на перевозку груза $f_3(1) = 11$ и оптимальный маршрут проходит через город 2. Далее из вычислений $f_2(2)$ следует, что оптимальный маршрут проходит через город 6, так как $j_2(2) = 6$. Наконец, из города 6 груз доставляется в конечный город 7 (место назначения). Таким образом, мы определили оптимальный маршрут $x^* = (1-2-6-7)$, затраты на перевозку груза по которому составляют $f_3(1) = 4 + 4 + 3 = 11$.

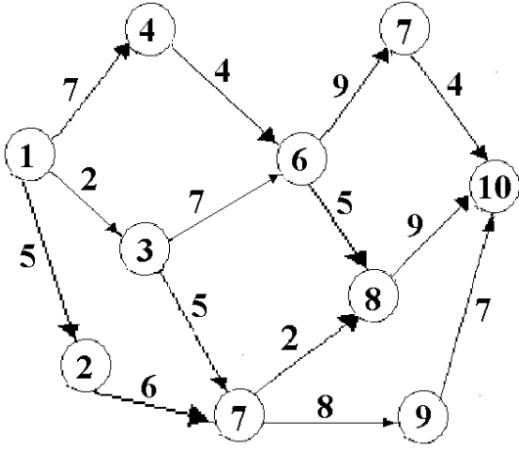
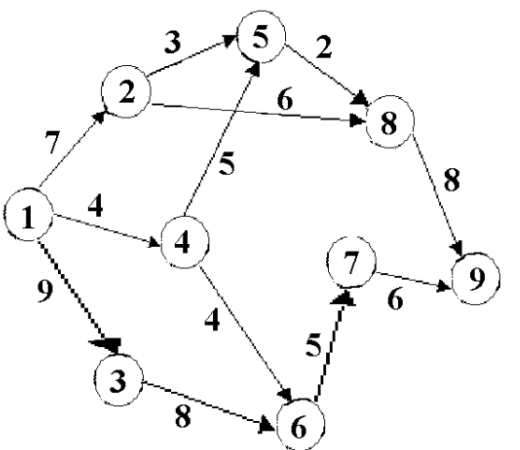
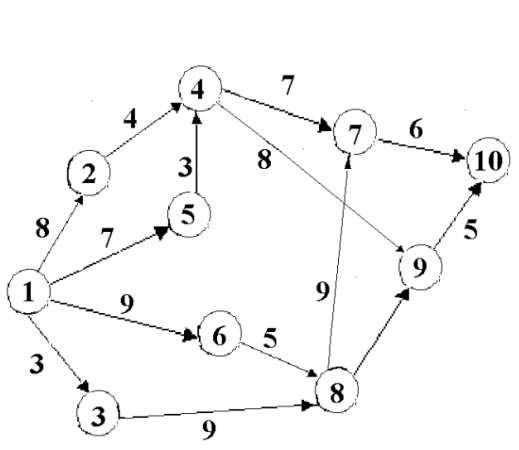
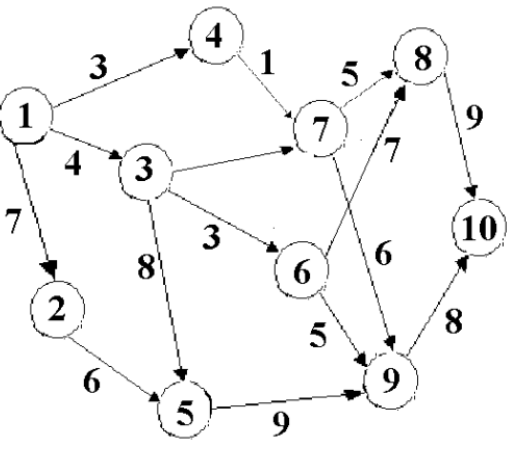
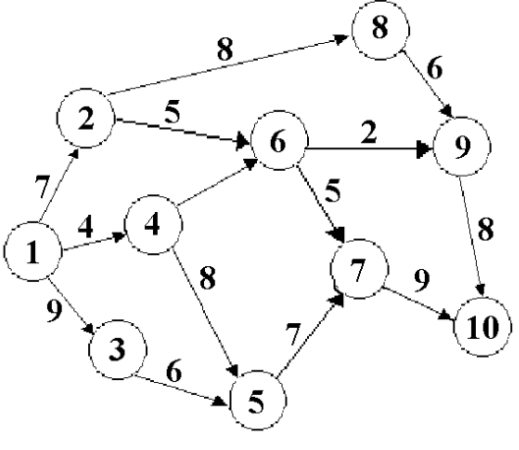
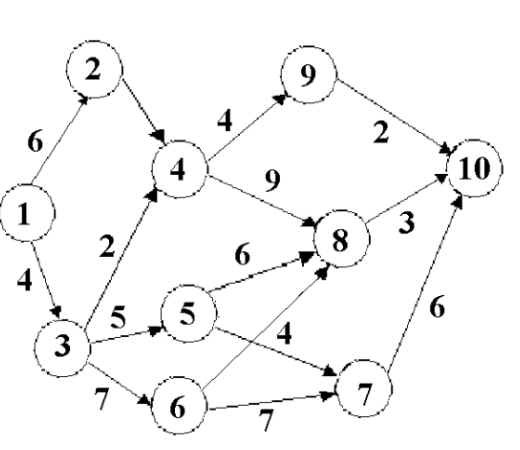
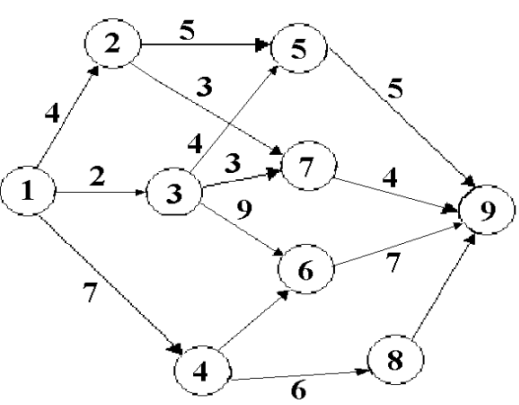
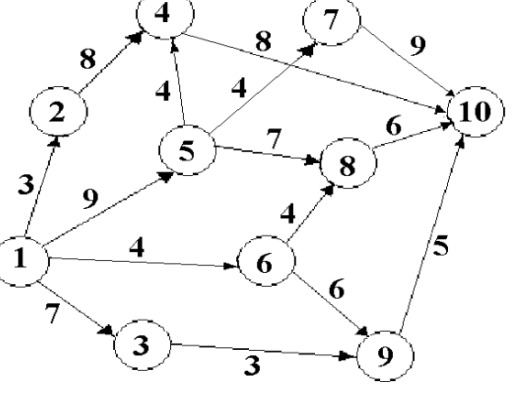
Следует отметить, что метод динамического программирования применим только для нахождения кратчайшего пути на связных графах, где любой маршрут состоит из одного и того же числа дуг, как, например, на рассмотренном графе. Для графов более общей структуры используются соответствующие алгоритмы теории графов.

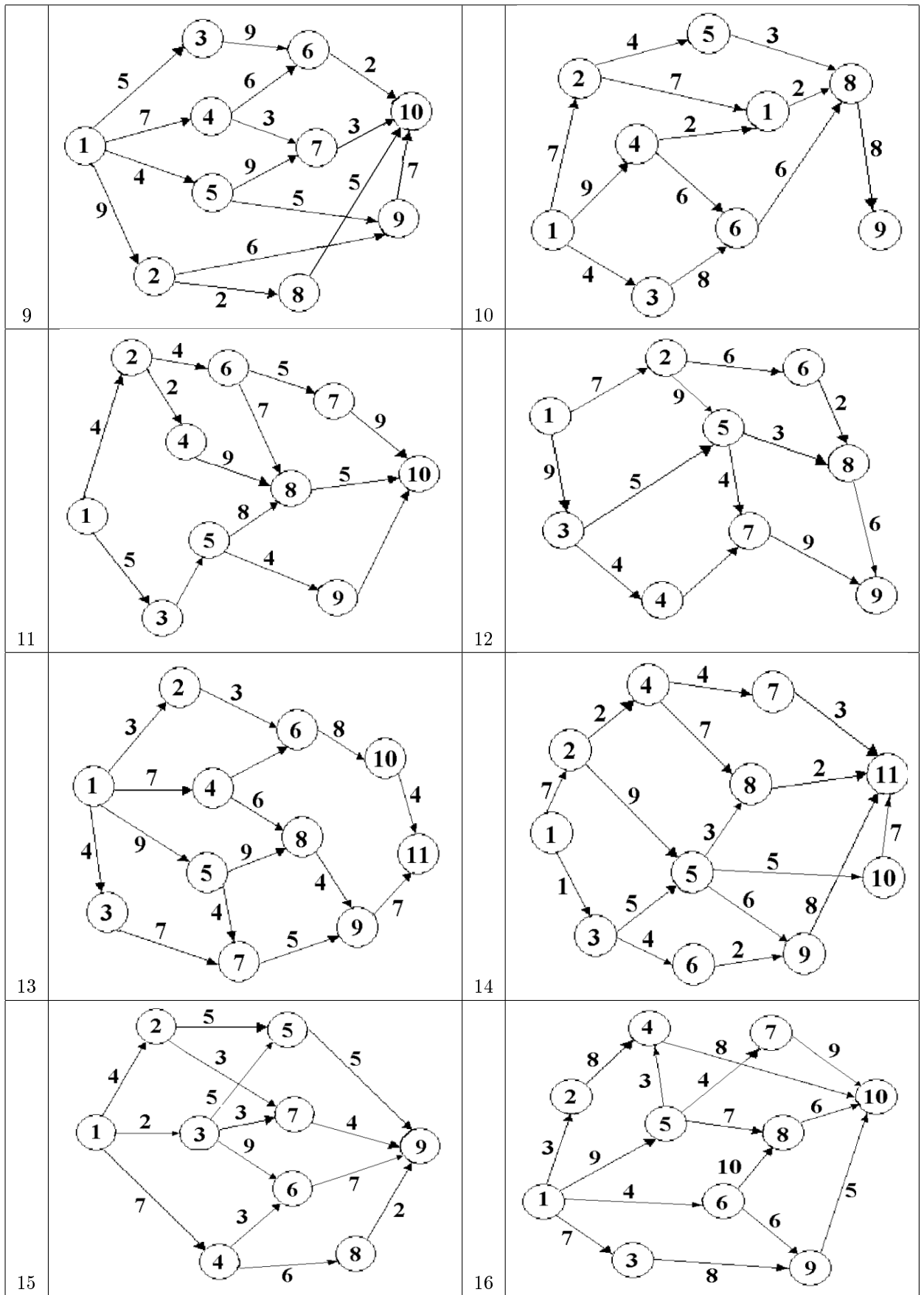
Порядок выполнения работы

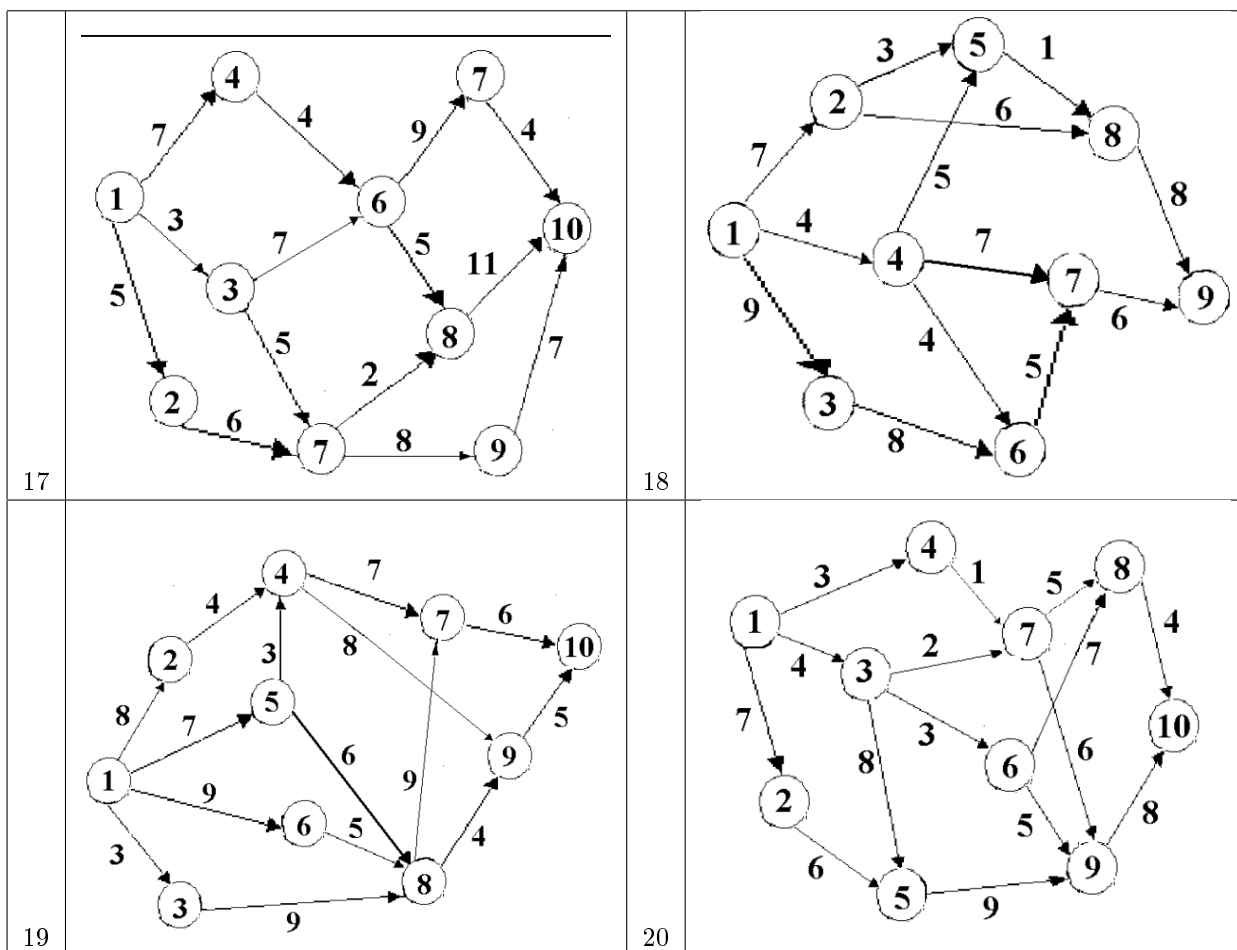
1. Изучить основные принципы динамического программирования.
2. Изучить порядок эксплуатации программных средств.
3. Ввести данные в ЭВМ и получить решение задачи.
4. Оформить отчет, в который включить: вариант задания, результаты расчета на ЭВМ, результаты ручного расчета (должно совпасть с программным решением), анализ результатов и выводы.

Варианты заданий

Найти кратчайший путь из первой вершины в последнюю по счету в заданном графе:

1		2	
3		4	
5		6	
7		8	





Контрольные вопросы

1. Для каких оптимизационных задач применяется метод динамического программирования?
2. В чем заключается суть метода динамического программирования?
3. Сформулируйте принцип оптимальности Беллмана.
4. Что является целевой функцией в задаче о кратчайшем маршруте?
5. Какой параметр определяет состояние системы на каждом шаге?
6. Нахождение экстремума методом множителей Лагранжа (решение практической задачи)
7. Ручной расчет задачи о кратчайшем маршруте.

Список литературы

- [1] Лесин, В.В. Основы методов оптимизации. [Электронный ресурс] / В.В. Лесин, Ю.П. Лисовец. — Электрон. дан. — СПб. : Лань, 2016. — 344 с. — Режим доступа: <http://e.lanbook.com/book/86017>
- [2] Пантелеев, А.В. Методы оптимизации в примерах и задачах. [Электронный ресурс] / А.В. Пантелеев, Т.А. Легова. — Электрон. дан. — СПб. : Лань, 2015. — 512 с. — Режим доступа: <http://e.lanbook.com/book/67460>