Министерство науки и высшего образования Российской Федерации

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

Кафедра информационных систем и вычислительной техники

СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

Методические указания к практическим занятиям для студентов магистратуры направления 09.04.02

САНКТ-ПЕТЕРБУРГ
2021

УДК 681.3.06 (073)

СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ: Методические указания к практическим занятиям / Санкт-Петербургский горный университет. Сост.: *И.В. Иванова*. СПб, 2021. 32 с.

Методические указания предназначены для выполнения практических занятий по дисциплине «Системы поддержки принятия решений». В состав методических указаний входят теоретический материал, описание методологии решения задач, принцип формирования варианта задания (согласно шифру), приведен пример выполнения задания, список рекомендуемой литературы. Цель – ознакомить с методами получения результата при решении сложных задач принятия решений и реализации возможности принятия рациональных решений, дать практические навыки выбора конкретной альтернативы проектного решения.

Предназначены для студентов магистратуры направления 09.04.02 «Информационные системы и технологии», направленность «Информационные системы и технологии».

Научный редактор доц. *Е.Б. Мазаков*

Рецензент: канд. техн. наук, проф. *А.М. Заяц (*Санкт-Петербургский государственный Лесотехнический университет им. С.М. Кирова)

ОБЩИЕ ПОЛОЖЕНИЯ

Изучение материала по каждой лекции дисциплины завершается выполнением практической работы. Практические занятия позволяют закрепить полученные на лекциях знания и выработать необходимые для решения задач навыки работы с изученными алгоритмами. Цель – расширить и углубить эти навыки, научить студента самостоятельно применять полученные знания для решения конкретных задач.

ЛЕКЦИЯ ПЕРВАЯ С ПРИМЕРОМ ВЫПОЛНЕНИЯ ЗАДАНИЯ

1.1 Применение метода динамического программирования для решения детерминированных задач теории принятия решений

1.1.1 Основные понятия и определения

Динамическое программирование – это совокупность процедур, используемых для оптимизации многошаговых процессов принятия решений.

Определение 1. Многошаговый процесс принятия решений – это схема принятия последовательных решений, направленных на достижение единой цели.

На основе концепции динамического программирования в работах Вальда, Гиршика, Массе были решены частные задачи. Беллманом был сформулирован принцип оптимальности и обосновано функциональное уравнение. Им разработаны схемы принятия решений на основе принципа оптимальности для многочисленных оптимизационных задач, возникающих в экономике, технике и социальных исследованиях. Концептуально динамическое программирование применяется для анализа систем, которые характеризуются следующими признаками:

– процесс функционирования системы включает последовательные этапы – текущий этап имеет номер i, конечный этап – номер m;

– на i-м шаге управление U_i, переводит систему из состояния S_{i-1}, достигнутого на i–1 -м этапе, в состоянии S_i;

– для анализируемой системы выполняется принцип отсутствия последействия, состоящий в том, что состояние S_i, зависит только от состояния для предыдущего этапа S_{i-1} управления U_i, но не зависит от состояний $S_{i-2}, S_{i-3}, \ldots, S_1$ и управлений $U_{i-1}, U_{i-2}, \ldots, U_1$. Известна функция локального дохода $w_i(S_i, U_i)$, определяющая значение прибыли, получаемой при применении на i-м этапе управления U_i, в случае нахождения системы в состоянии S_i.

Доход от функционирования системы за m-этапов равен сумме локальных доходов, полученных на каждом этапе (рис. 1).

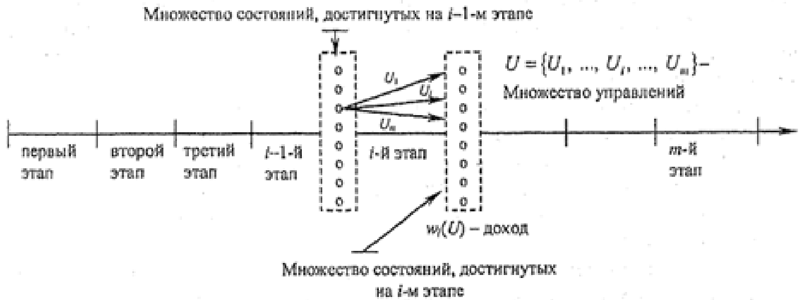

Рис. 1. Многошаговые процессы принятия решений

Необходимо найти такой вектор управлений $U = \{U_1, \ldots, U_i, \ldots, U_m\}$, который обеспечивает максимизацию суммарного дохода: $W = \sum_{i=1}^{m} w_i(U) \to \max$.

Понятие *этапа, состояния, управления, дохода* зависит от предметной ориентации сложной системы.

Для организационно-экономических систем:

– в качестве *этапов* могут выступать интервалы планирования – год, квартал;

– *состояние* – характеризуется наличием свободных финансовых средств;

– *управление* – представляет возможные варианты инвестирования этих средств;

– *локальный доход* от i-го этапа - это прибыль, получаемая на i-м этапе.

Для дискретной системы принятие решения, к которой относятся процедуры решения в задаче коммивояжера:

– *этапы* – это переход из одного города в другой;

– *состояние* – город, в котором находится коммивояжер, и перечень, тех городов, которые ему необходимо посетить;

– *управление* – выбор следующего для посещения города из множества не посещенных городов;

– *локальных доход* – расстояние между соседними городами;

– *суммарный доход* – длина замкнутого пути, проходящего через все города.

Для технических движущихся объектов:

– *состояние* – это совокупность таких параметров, как скорость, геометрические координаты, ускорение и др.;

– *управление* – это изменение координат, скорости;

– *локальный доход* – либо время перехода между состояниями для соседних этапов, либо расход топлива;

– *суммарный доход* – *время* перемещения между начальным и конечным состояниями.

Для решения такого класса задач могут использоваться два подхода.

Первый подход предполагает, что одновременно находятся компоненты вектора управление для всех этапов решения. Такой подход глобальной оптимизации приводит к существенному возрастанию трудоемкости получения решения. ***Второй подход*** базируется на нахождении оптимального управления последовательно для каждого из этапов. Очевидно, если находить оптимальное управление для первого этапа, не учитывая дальнейшее функционирование системы и, самое главное, не учитывая влияние решении первого этапа на последующие решения, то локальное оптимальное решение может существенно отличаться от глобального оптимального решения для этого этапа. Следовательно, необходимо выбрать такой этап и такую последовательность их рассмотрения, при которой бы оптимальное управление на таком этапе не оказывало влияния на последующие этапы.

Тогда ***классическая идея динамического программирования***, основанная на алгоритме обратной прогонки, состоит в том, что в качестве этапа, для которого на первом шаге находится локальное оптимальное управление, рассматривается последний этап принятия решений. Очевидно, что решения, принятые на этом этапе не оказывают влияния на последующий этап, т.к. этот этап является последним. Однако при таком подходе является неизвестным состояние, в котором будет находиться система перед началом выполнения последнего этапа. Поэтому локальное оптимальное управление необходимо найти для всех возможных состояний, в которых может находиться система, перед выполнением последнего этапа. После этого осуществляется переход к оптимизации управления на предпоследнем этапе. Оптимальное управление на этом этапе находится с учетом того, что уже известно оптимальное управление на последнем этапе. Таким образом, для каждого состояния предпоследнего этапа находится локальное оптимальное управление. Такая процедура повторяется для всех последующих этапов: n–2. n–3,..., i,…,2, вплоть до первого этапа, т.е. оптимальное решение для первого этапа определяется с учетом ранее найденных оптимальных решений для последующих этапов.

1.1.2 Современная трактовка динамического программирования

Принцип решения локальной задачи, начиная с последнего этапа, был предложен Беллманом. Современная трактовка динамического программирования предполагает *применение алгоритмов обратной и прямой прогонки.* Отличие этих алгоритмов целесообразно проиллюстрировать на примере решения задачи поиска кратчайшего пути в ориентированном графе. Для ориентированного графа известными являются вершины и дуги, над которыми проставлены времена перехода между вершинами (рис.2).

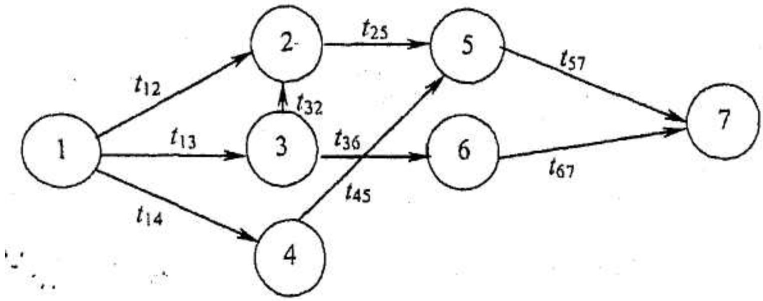

Рис.2. Пример ориентированного графа

Наличие дуги (i, j) предполагает процедуру перемещения от узла i к узлу j. Конечная последовательность узлов $(i_1, i_2, \ldots, i_n)$ таких, что (i_k, i_{k+1}) является направленной дугой для $k = 1, 2, 3, \ldots, n-1$ называется путем в графе. Путь, у которого начальная и конечная вершины совпадают $i_1 = i_n$, называется циклом. Сеть является циклической, если она содержит один или более циклов. В противном случае сеть называется ациклической. Длина пути в графе представляет собой сумму длин, входящих в него дуг.

Для вычисления кратчайшей длины пути в графе от начальной вершины 1 до текущей вершины j применяется следующее соотношение

$$f_j = \min\left\{f_i + t_{ij}\right\}, \quad j = \overline{2, m},$$

где $f_j, f_i -$ кратчайшие пути в графе от вершины 1 соответственно к вершинам i и j; t_{ij} – длина дуги, соединяющей i-ю и j-ю вершины.

В соответствии с *алгоритмом обратной прогонки* рассматриваются дуги для тех узлов, кратчайшие пути до которых еще неизвестны. Реализация алгоритма включает следующие шаги:

Шаг 1. Для каждого узла вводится переменная v_j, которая предназначена для фиксации функций длины пути от вершины номер 1 до вершины с номером j. Окончательное значение переменной v_j совпадает со значением переменной f_j. Начальные значения переменной v_j определяются следующим образом: $v_1 = 0, v_k = \infty, k = \overline{2,m},$ значение переменной *j* полагается равной 2.

Шаг 2. Для каждой дуги (i, j), входящей в узел *j*, необходимо вычислить $v_j = \min\{v_j, v_i + t_{ij}\}$.

Шаг 3. Проверить условие $j = m$. Если да, то $f_j = v_j$, и выйти из алгоритма. В противном случае $j = j + 1$, и перейти на выполнение шага 2.

Алгоритм прямой прогонки основан на том, что рассматриваются дуги для тех узлов, кратчайшие пути для которых известны. Этот алгоритм включает выполнение следующих шагов:

Шаг 1. Задаются начальные значения переменной $v_k, k = \overline{1,m}, \; v_1 = 0, v_k = \infty, k = \overline{2,m}.$

Начальное значение переменной $i = 1$, т.е. вначале только для вершины с номером 1 определена длина кратчайшего пути $v_1 = f_1 = 0$.

Шаг 2. Для каждой дуги (i, j), исходящей из узла *i*, необходимо вычислить $v_j = \min\{v_j, v_i + t_{ij}\}$.

Шаг 3. Проверить выполнение условия $i = m - 1$. Если да, то $f_j = v_j$, и необходимо выйти из алгоритма. В противном случае $i = i + 1$, и следует перейти к реализации шага 2. Необходимо отметить, что *алгоритм обратной прогонки* предполагает возможность начала его выполнения в любом конце сети и движения в обратном направлении к другому концу сети.

1.1.3 Принцип Беллмана и общая схема решения функционального уравнения

Принцип Беллмана. Каково бы ни было состояние системы на i-м этапе, управление должно быть выбрано из условия, что и на последующих этапах управление будет оптимальным.

Первоначальная формулировка принципа оптимальности, предложенная Р. Беллманом: *"Оптимальное поведение обладает тем свойством, что каковы бы не были первоначальное состояние и решение в начальный момент, последующие решения должны составлять оптимальное поведение относительно состояния, получающегося в результате первого решения"*

Для сетевых задач принцип оптимальности может быть представлен следующим образом: *подпутъ кратчайшего пути сам является кратчайшим путем.*

Один из вариантов современной трактовки представляется следующим образом: *допущенную ошибку на начальных этапах нельзя исправить на последующих этапах принятия решений.*

Общая схема решения задачи на основе принципа Беллмана предполагает ***формальное определение таких компонент*** как этап, состояние, управление, оператор перехода, локальный доход на i-м этапе, условный оптимальный доход для i-го этапа.

Этапы: при решении задачи должны быть перечислены и определены этапы принятия решения. Необходимо помнить, что количество этапов конечно. Качественное определение этапов зависит от природы управляемой системы, и они связаны либо с временной, либо с логической последовательностью принятия решений.

Состояние – это такой набор параметров, который однозначно характеризует поведение системы. Для движущихся систем состояние – это значения координат, значения 1,2, ..., n-х производных. Для экономических систем – это имеющие свободные средства, для задачи коммивояжера – это город, в котором находится коммивояжер, и перечень городов, которые ему необходимо посетить, и т. д. Таким образом, общее требование к состоянию – это сохранение в нем информации о предыстории процесса, или иначе состояние должно быть описано с той степенью подробности, которая позволяет провести оценку текущих альтернативных решений. Состояние в общем случае обозначается как S.

Управление U_i – это целенаправленное воздействие на управляемую систему на i-м этапе принятия решения.

Условное оптимальное управление $U_i(S)$ – это наилучшая стратегия для каждого из состояний. При этом система переходит из одного состояния в другое.

Оператор перехода – предназначен для установления связи между состояниями для соседних этапов принятия решений, $S' = \varphi(S, U_i)$ – оператор перехода, задающий переход из состояния S в состояние S' при использовании на i-м этапе стратегии U_i.

Локальный доход на i-м этапе $w_i(S, U_i)$ – это величина прибыли, получаемая от функционирования системы на i-м этапе, при условии, что она находилась в состоянии S, и была выбрана стратегия U_i.

Условный оптимальный доход $w_i(S)$ – это суммарный оптимальный доход полученный от функционирования системы на i-м, i+1-м, …, m-м этапах, при условии, что на i-м этапе система находилась в состоянии S. Тогда функциональное уравнение Беллмана представляется следующим образом: условный оптимальный доход на i-м этапе для состояния S – это максимум из суммы локального дохода на i-м этапе и условного оптимального дохода для состояния S', вычисленного для следующего i +1-го этапа:

$$W_i(S) = \max_{U_i} \left\{ w_i(S, U_i) + W_{i+1}\left(\varphi(S, U_i)\right) \right\}.$$

В соответствии с алгоритмом обратной прогонки решение функционального уравнения начинается с последнего этапа принятия решений, т.е. i полагается равным m. Тогда

$$W_m(S) = \max_{U_m} \left\{ w_m(S, U_m) \right\}.$$

В результате решения этого уравнения находится условное оптимальное управление $U_m(S)$ для каждого состояния S и условный оптимальный доход $W_m(S)$. Найденные значения используются для решения функционального уравнения Беллмана при $i = m - 1$:

$$W_{m-1}(S) = \max_{U_{m-1}} \left\{ w_{m-1}(S, U_{m-1}) + W_m\left(\varphi(S, U_{m-1})\right) \right\}.$$

В результате решения этого уравнения для состояния S на m–1-м этапе находятся $U_{m-1}(S)$ и $W_{m-1}(S)$.

Таким образом, последовательное решение уравнения Беллмана позволяет найти пары: $U_i(S), W_i(S), i=\overline{1,m}.$ Затем задается начальное состояние S_0 , и для этого состояния определяется оптимальное управление $U_1(S_0)$. После этого находится оптимальное состояние $S_1 = \varphi(S_0, U_1(S_0))$. Для найденного состояния S_1 определяется оптимальная стратегия $U_2(S_1)$, а затем с помощью оператора перехода вычисляется оптимальное состояние S_2 и т.д.

Таким образом, формируется цепочка:

$$S_0 \to U_1(S_0) \to S_1 = \varphi(S_0, U_1(S_0)) \to U_2(S_1) \to$$
$$\to S_2 = \varphi(S_1, U_2(S_1)) \to U_3(S_2) \to \ldots U_m(S_{m-1})$$

Эта цепочка содержит оптимальные последовательности состояний и стратегий.

В задаче динамического программирования при использовании схемы прямой прогонки функциональное уравнение Беллмана имеет следующий вид:

$$W_i(S) = \max_{U_i} \left\{ w_i(S, U_i) + W_{i-1}\left(\varphi^{-1}(S, U_i)\right) \right\},$$

где $\varphi^{-1}(S, U_i)$– оператор обратного перехода, устанавливающий состояние для предыдущего этапа.

1.1.4 Решение задачи коммивояжера методом динамического программирования

Применение метода динамического программирования для задачи коммивояжера впервые было предложено одновременно и независимо Р. Беллманом и М. Хельдом, а также Р. Карпом в 1962 году.

Постановка задачи. Задано множество городов $A = \{A_1, \ldots, A_i, \ldots, A_n\}$, матрица расстояний $C = \| c_{ij} \|$. Необходимо найти такую перестановку городов $\sigma_n = \langle i_1, i_2, \ldots, i_s, \ldots, i_n \rangle$, для которой длина маршрута $c_{i_1 i_2} + c_{i_2 i_3} + \ldots + c_{i_{n-1} i_n} + c_{i_n i_1}$ будет минимальной.

Шаг 1. Решение задачи начинается с последнего этапа. В качестве конечного города фиксируется город A_n и предполагается, что коммивояжер может находиться в одном из городов $A_1, \ldots, A_{n-1}$. В качестве состояния для этого этапа выбирается конструкция $A_i\{0\}$, где A_i указывает город, в котором находится коммивояжер на последнем этапе принятия решения, а наличие нуля в фигурных скобках соответствует отсутствию промежуточных городов для посещения. Последний этап принятия решений связан с числом городов соотношением $m = n-1$ (рис. 3).

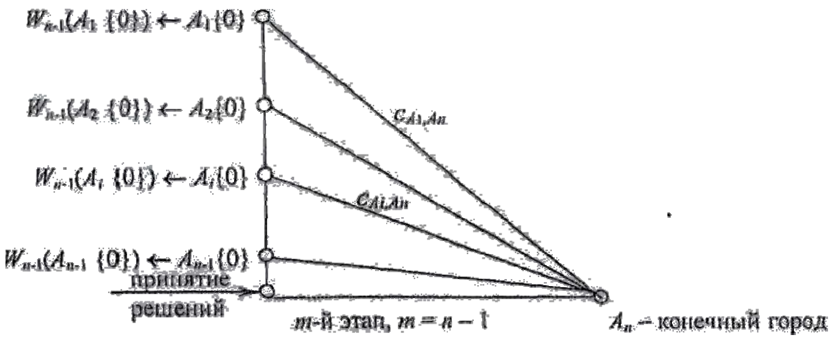

Рис. 3. Графическое представление m-го этапа, $m = n-1$

Общее количество состояний равно $n-1$. Для каждого состояния $A_i\{0\}$ вводится функция условного оптимального дохода, которая равна расстоянию от города A_i, в котором находится коммивояжер, до конечного города A_n, т.е. $W_{n-1}(A_i\{0\}) = c_{A_1 A_n}$.

Для каждого состояния существует единственное управление, предполагающее переход из A_i -го города в A_n -й город.

Шаг 2. Соответствует $m-1$-му этапу принятия решения, т.е. $m-1=n-2$. На этом этапе коммивояжер находится в городе A_i, ему необходимо посетить один промежуточный город, а затем из промежуточного города переехать в конечный город A_n. В качестве состояния вводится следующая конструкция: $A_i\{A_{U_1}\}$, где A_i – номер города, в котором находится коммивояжер на $m-1$-м этапе, а A_{U_1} – промежуточный город, который ему необходимо посетить.

Очевидно, что $A_{U_1} \in A \setminus (A_i \vee A_n)$, где A – множество всех городов (рис. 4).

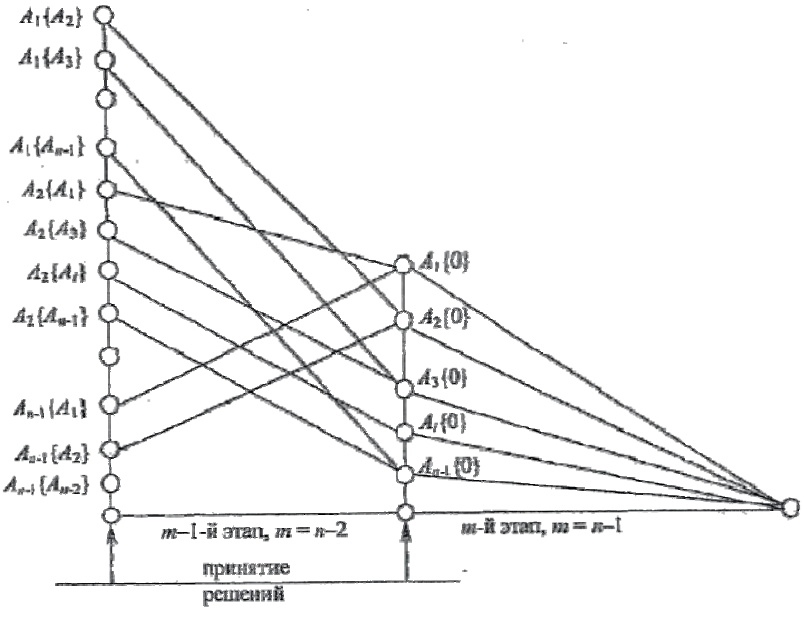

Рис. 4. Графическое представление $m-1$-го этапа, $m-1=n-2$.

Для каждого состояния $A_i\{A_{U_1}\}$ вычисляется функция условного оптимального дохода

$$W_{n-2}(A_i\left\{A_{U_1}\right\})=c_{A_i A_{U_1}}+W_{n-1}(A_{U_1}\left\{0\right\}).$$

Эта функция равна длине маршрута между городами A_i и A_n с обязательным посещением промежуточного города A_{U_i}.

Необходимо отметить, что на рассматриваемом этапе для каждого из состояний отсутствуют альтернативные варианты управления и, следовательно, между городами A_i и A_n с посещением промежуточного города A_{U_i} существует единственный маршрут. В частности, для состояния $A_2\{A_3\}$ возможно только одно управление – это переезд из города A_2 в город A_3. Общее число состояний на этом этапе равно $(n-1)^2$.

Шаг 3. Соответствует ситуации, в которой коммивояжер находится в одном из городов A_i, ему необходимо посетить два промежуточных города и после этого переехать в конечный город A_n.

Состояние представляется в виде следующей конструкции: $A_i\{A_{U_1}, A_{U_2}\}$.

Для компонент состояния справедливо следующее условие: $A_{U_1} \neq A_{U_2} \neq A_i \neq A_n$, где A_i – город, в котором находится коммивояжер на рассматриваемом этапе принятия решений; A_{U_1}, A_{U_2} – промежуточные города, которые ему необходимо посетить. Необходимо отметить, что для идентификации состояния последовательность городов, указанная в фигурных скобках, является безразличной. Следовательно, состояния $A_1\{A_2, A_3\}$ и $A_1\{A_3, A_2\}$ являются одинаковыми. В каждом из состояний $A_i\{A_{U_1}, A_{U_2}\}$ существует два варианта принятия решений: *первый вариант* предполагает переезд из города A_i в город A_{U_1} с дальнейшим посещением городов A_{U_2} и A_n; *второй вариант* основан на переезде из города A_i в город A_{U_2} с дальнейшим посещением городов A_{U_2} и A_n (рис.5). Из этих двух альтернатив необходимо выбрать такой вариант управления, для которого функция условного оптимального дохода будет минимальной, т.е.

$$W_{n-3}(A_i\{A_{U1}, A_{U2}\}) = \min_{A_i \to A_{U1}, A_i \to A_{U2}} (c_{A_i A_{U_1}} + W_{n-2}(A_{U1}\{A_{U2}\}),$$

$$c_{A_i A_{U_2}} + W_{n-2}(A_{U2}\{A_{U_1}\})).$$

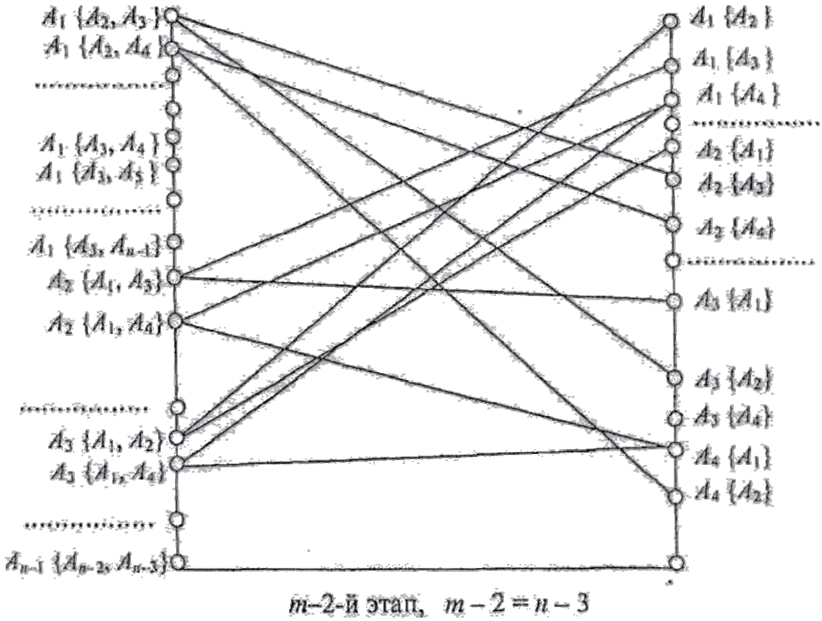

Рис. 5. Выбор управления на $m-2$-м шаге

Функция условного оптимального дохода $W_{n-3}(A_i\left\{A_{U_1}, A_{U_2}\right\})$ представляет собой длину кратчайшего маршрута между городами A_i и A_n при условии обязательного посещения двух промежуточных городов: A_{U_1} и A_{U_2} при этом последовательность посещения определяется только условием минимизации длины маршрута.

Шаг k. Этап принятия решений, рассматриваемый на этом шаге, характеризуется тем, что коммивояжер, находится в одном из городов A_i, ему необходимо посетить $k-1$ промежуточных городов и затем переехать в последний город A_n. Состояние задается конструкцией: $A_i\left\{A_{U_1}, A_{U_2}, \ldots, A_{U_{k-1}}\right\}$, где A_i – город, в котором находится коммивояжер на рассматриваемом этапе, а $A_{U_1}, A_{U_2}, \ldots, A_{U_{k-1}}$ – промежуточные города, которые не совпадают между собой и не совпадают с городами A_i и A_n, т.е $A_{U_1} \neq A_{U_2} \neq \ldots \neq A_{U_{k-1}} \neq A_i \neq A_n$. В каждом из состояний $A_i\left\{A_{U_1}, A_{U_2}, \ldots, A_{U_{k-1}}\right\}$, существует $k-1$ вариант принятия решений: *первый вариант* предполагает переезд из A_i-го города в A_{U_1} -й город с последующим посещением $A_{U_2}, A_{U_3}, \ldots, A_{U_{k-1}}$ -го городов; *второй вариант* содержит переезд из A_i-го города в A_{U_2} -й город с последующим посещением $A_{U_1}, A_{U_3}, \ldots, A_{U_{k-1}}$ -го городов; $k-1$ *-й вариант* включает переезд в $A_{U_{k-1}}$ -й город с последующим посещением $A_{U_1}, A_{U_2}, \ldots, A_{U_{k-2}}$ -го городов.

Функциональное уравнение для этого этапа представляется следующим образом:

$$
\begin{aligned}
& W_{n-k}\left(A_i\left\{A_{U_1}, A_{U_2}, \ldots, A_{U_{k-1}}\right\}\right)= \\
& =\min \{c_{A_i A_{U_1}}+W_{n-k+1}(A_{U_1}\{A_{U_2}, A_{U_3}, \ldots, A_{U_{k-1}}\}), \\
& c_{A_i A_{U_2}}+W_{n-k+1}(A_{U_2}\{A_{U_1}, A_{U_3}, \ldots, A_{U_{k-1}}\})), \\
& c_{A_i A_{U_3}}+W_{n-k+1}(A_{U_3}\{A_{U_1}, A_{U_2}, A_{U_4}, \ldots, A_{U_{k-1}}\}), \ldots, \\
& c_{A_i A_{U_{k-1}}}+W_{n-k+1}(A_{U_{k-1}}\{A_{U_1}, A_{U_2}, \ldots, A_{U_{k-2}}\})\}.
\end{aligned}
$$

Шаг n. Это последний шаг алгоритма, соответствует первому этапу с начала принятия решения коммивояжером и характеризуется тем, что коммивояжер находится в начальном фиксированном городе A_n, ему необходимо посетить все остальные города $A_1, A_2, \ldots, A_{n-1}$ и вернуться в город A_n (рис.6).

Функциональное уравнение Беллмана имеет следующий вид:

$$W_1\left(A_n\left\{A_1, A_2, A_3, \ldots, A_{n-1}\right\}\right) == \min\{c_{A_nA_1} + W_2(A_1\{A_2, A_3, \ldots, A_{n-1}\}),$$
$$c_{A_nA_2} + W_2(A_2\{A_1, A_3, \ldots, A_{n-1}\}, \ldots, c_{A_nA_{n-1}} + W_2(A_{n-1}\{A_1, A_2, \ldots, A_{n-2}\})\}.$$

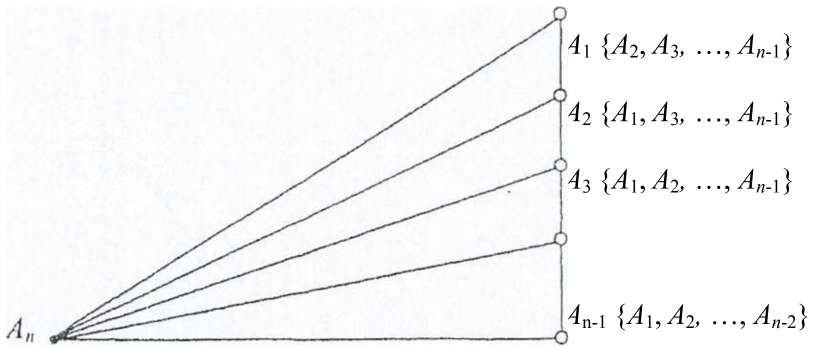

Рис.6. Выбор управления на первом этапе принятия решений

Решение уравнения позволяет вычислить длину оптимального маршрута и найти соответствующее ему оптимальное управление, т.е. определить оптимальный переезд из города A_n в следующий город. Пусть это будет город A_k , а оптимальное решение – $A_k\{A_1, A_2, \ldots, A_{k-1}, A_{k+1} \ldots, A_{n-1}\}$. На основе ранее полученного решения функционального уравнения находится следующий город, входящий в оптимальный маршрут, т.е. фиксируется город A_r в который переезжает коммивояжер из города A_k. Аналогично определяются остальные города, входящие в оптимальный маршрут.

Пример. Методом динамического программирования найти оптимальное решение для задачи коммивояжера. Матрица расстояний имеет следующий вид:

$$C = \begin{vmatrix} \infty & 8 & 8 & 5 & 9 & 2 & 13 \\ 9 & \infty & 1 & 6 & 7 & 3 & 7 \\ 11 & 4 & \infty & 4 & 2 & 9 & 3 \\ 6 & 3 & 2 & \infty & 2 & 8 & 15 \\ 2 & 2 & 8 & 4 & \infty & 3 & 8 \\ 1 & 3 & 16 & 5 & 3 & \infty & 5 \\ 2 & 4 & 12 & 5 & 13 & 8 & \infty \end{vmatrix}.$$

Шаг 1. Решение задачи производится в соответствии с алгоритмом обратной прогонки. Рассматривается последний этап принятия решений. В качестве конечного города фиксируется город A_7. Предполагается, что коммивояжер может находиться в одном из следующих городов: $A_1, A_2, A_3, A_4, A_5, A_6$. Состояние определяется конструкцией $A_i\{0\}$. Тогда $W_7(A_i\{0\}) = c_{A_1 A_6}$:

$$W_7(A_1\{0\}) = 13, W_7(A_2\{0\}) = 7, W_7(A_3\{0\}) = 3,$$

$$W_7(A_4\{0\}) = 15, W_7(A_5\{0\}) = 8, W_7(A_6\{0\}) = 5.$$

Шаг 2. Анализируется предпоследний этап принятия решений. Предполагается, что коммивояжер находится в одном из городов A_i, $i = \overline{1,6}$, , ему необходимо посетить один промежуточный город и затем переехать в конечный город A_7. Состояние задается следующей конструкцией: $A_i\left\{A_{u_1}\right\}, i = \overline{1,6}; u_1 = \{1, 2, 3, 4, 5, 6\}; A_i \neq A_{u_1} \neq A_7.$ Тогда уравнение Беллмана имеет вид $W_6\left(A_i\left\{A_{u_1}\right\}\right) = c_{iu_1} + W_7\left(A_{u_1}\{0\}\right).$ Значения функции $W_6\left(A_i\left\{A_{u_1}\right\}\right)$ определяются следующим образом:

$$W_6(A_1\{A_2\}) = c_{12} + W_7(A_2\{0\}) = 15;$$

$$W_6(A_1\{A_3\}) = c_{13} + W_7(A_3\{0\}) = 11;$$

$$W_6(A_1\{A_4\}) = c_{14} + W_7(A_4\{0\}) = 20;$$

$$W_6(A_1\{A_5\}) = c_{15} + W_7(A_5\{0\}) = 17;$$
$$W_6(A_1\{A_6\}) = c_{16} + W_7(A_6\{0\}) = 7;$$
$$W_6(A_2\{A_1\}) = c_{21} + W_7(A_1\{0\}) = 22;$$
$$W_6(A_2\{A_3\}) = c_{23} + W_7(A_3\{0\}) = 4;$$
$$W_6(A_2\{A_4\}) = c_{24} + W_7(A_4\{0\}) = 21;$$
$$W_6(A_2\{A_5\}) = c_{25} + W_7(A_5\{0\}) = 15;$$
$$W_6(A_2\{A_6\}) = c_{26} + W_7(A_6\{0\}) = 8;$$
$$W_6(A_3\{A_1\}) = c_{31} + W_7(A_1\{0\}) = 24;$$
$$W_6(A_3\{A_2\}) = c_{32} + W_7(A_2\{0\}) = 11;$$
$$W_6(A_3\{A_4\}) = c_{34} + W_7(A_4\{0\}) = 19;$$
$$W_6(A_3\{A_5\}) = c_{35} + W_7(A_5\{0\}) = 10;$$
$$W_6(A_3\{A_6\}) = c_{36} + W_7(A_6\{0\}) = 14;$$
$$W_6(A_4\{A_1\}) = c_{41} + W_7(A_1\{0\}) = 19;$$
$$W_6(A_4\{A_2\}) = c_{42} + W_7(A_2\{0\}) = 10;$$
$$W_6(A_4\{A_3\}) = c_{43} + W_7(A_3\{0\}) = 5;$$
$$W_6(A_4\{A_5\}) = c_{45} + W_7(A_5\{0\}) = 10;$$
$$W_6(A_4\{A_6\}) = c_{46} + W_7(A_6\{0\}) = 13;$$
$$W_6(A_5\{A_1\}) = c_{51} + W_7(A_1\{0\}) = 15;$$
$$W_6(A_5\{A_2\}) = c_{52} + W_7(A_2\{0\}) = 9;$$
$$W_6(A_5\{A_3\}) = c_{53} + W_7(A_3\{0\}) = 11;$$
$$W_6(A_5\{A_4\}) = c_{54} + W_7(A_4\{0\}) = 19;$$
$$W_6(A_5\{A_6\}) = c_{56} + W_7(A_6\{0\}) = 8;$$
$$W_6(A_6\{A_1\}) = c_{61} + W_7(A_1\{0\}) = 14;$$
$$W_6(A_6\{A_2\}) = c_{62} + W_7(A_2\{0\}) = 10;$$
$$W_6(A_6\{A_3\}) = c_{63} + W_7(A_3\{0\}) = 19;$$
$$W_6(A_6\{A_4\}) = c_{64} + W_7(A_4\{0\}) = 20;$$
$$W_6(A_6\{A_5\}) = c_{65} + W_7(A_5\{0\}) = 22.$$

Шаг 3. Предполагается, что коммивояжер находится в одном из городов A_i, $i = \overline{1,6}$, , ему необходимо посетить два промежуточных города: A_{u_1} и A_{u_2}, и затем переехать в конечный город A_7 Состояние определяется следующим образом: $A_i\left\{A_{u_1}A_{u_2}\right\}$, $A_i \neq A_{u_1} \neq A_{u_2} \neq A_7$. Уравнение Беллмана имеет вид:

$$W_5\left(A_i\left\{A_{u_1}, A_{u_2}\right\}\right) = \min\left\{c_{A_i A_{u1}} + W_6\left(A_{u_2}\left\{A_{u_1}\right\}\right)\right\}.$$

Для наилучших значений функции Беллмана применяется подчеркивание:

$$W_5(A_1\{A_2, A_3\}) = \min\,(\underline{c_{12} + W_6(A_2\{A_3\})}, c_{13} + W_6(A_2\{A_3\})) =$$
$$= \min\left(12{,}19\right) = 12;$$

$$W_5(A_1\{A_2, A_4\}) = \min\,(c_{12} + W_6(A_2\{A_4\}), \underline{c_{14} + W_6(A_4\{A_2\})}) =$$
$$= \min\left(29{,}15\right) = 15;$$

$$W_5(A_1\{A_2, A_5\}) = \min\,(c_{12} + W_6(A_2\{A_5\}), \underline{c_{15} + W_6(A_5\{A_2\})}) =$$
$$= \min\left(23{,}18\right) = 18;$$

$$W_5(A_1\{A_2, A_6\}) = \min\,(c_{12} + W_6(A_2\{A_6\}), \underline{c_{16} + W_6(A_6\{A_2\})}) =$$
$$= \min\left(16{,}12\right) = 12;$$

$$W_5(A_1\{A_2, A_4\}) = \min\,(c_{13} + W_6(A_3\{A_4\}), c_{14} + \underline{W_6(A_4\{A_3\})}) =$$
$$= \min\left(27{,}10\right) = 10;$$

$$W_5(A_1\{A_3, A_5\}) = \min\,(c_{13} + W_6(A_3\{A_5\}), \underline{c_{15} + W_6(A_5\{A_3\})}) =$$
$$= \min\left(18{,}20\right) = 18;$$

$$W_5(A_1\{A_3,A_6\}) = \min\,(c_{13} + W_6(A_3\{A_6\}), \underline{c_{16} + W_6(A_6\{A_3\})}) =$$
$$= \min(22,21) = 21;$$

$$W_5(A_1\{A_4,A_5\}) = \min\,(\underline{c_{14} + W_6(A_4\{A_5\})}, c_{15} + W_6(A_5\{A_4\})) =$$
$$= \min(15,28) = 15;$$

$$W_5(A_1\{A_4,A_6\}) = \min\,(\underline{c_{14} + W_6(A_4\{A_6\})}, c_{16} + W_6(A_6\{A_4\})) =$$
$$= \min(18,22) = 18;$$

$$W_5(A_1\{A_5,A_6\}) = \min\,(c_{15} + W_6(A_5\{A_6\}), \underline{c_{16} + W_6(A_6\{A_5\})}) =$$
$$= \min(17,13) = 13;$$

$$W_5(A_2\{A_1,A_3\}) = \min\,(\underline{c_{21} + W_6(A_1\{A_3\})}, c_{23} + W_6(A_3\{A_1\})) =$$
$$= \min(20,25) = 20;$$

$$W_5(A_2\{A_1,A_4\}) = \min\,(c_{21} + W_6(A_1\{A_4\}), \underline{c_{24} + W_6(A_4\{A_1\})}) =$$
$$= \min(29,25) = 25;$$

$$W_5(A_2\{A_1,A_5\}) = \min\,(c_{21} + W_6(A_1\{A_5\}), \underline{c_{25} + W_6(A_5\{A_1\})}) =$$
$$= \min(26,22) = 22;$$

$$W_5(A_2\{A_1,A_6\}) = \min\,(\underline{c_{21} + W_6(A_1\{A_6\})}, c_{26} + W_6(A_6\{A_1\})) =$$
$$= \min(16,17) = 16;$$

$$W_5(A_2\{A_3,A_4\}) = \min\,(c_{23} + W_6(A_3\{A_4\}), \underline{c_{24} + W_6(A_4\{A_3\})}) =$$
$$= \min(20,11) = 11;$$

$$W_5(A_2\{A_3,A_5\}) = \min\,(\underline{c_{23} + W_6(A_3\{A_5\})}, c_{25} + W_6(A_5\{A_3\})) =$$
$$= \min(11,18) = 11;$$

$$W_5(A_2\{A_3,A_6\}) = \min\,(\underline{c_{23} + W_6(A_3\{A_6\})}, c_{26} + W_6(A_6\{A_3\})) =$$
$$= \min(15,22) = 15;$$

$$W_5(A_2\{A_4,A_5\}) = \min\,(\underline{c_{24}+W_6(A_4\{A_5\})},\, c_{12}+W_6(A_5\{A_4\})) =$$
$$= \min\left(16,26\right) = 16;$$

$$W_5(A_2\{A_4,A_6\}) = \min\,(\underline{c_{24}+W_6(A_4\{A_6\})},\, c_{26}+W_6(A_6\{A_4\})) =$$
$$= \min\left(19,23\right) = 19;$$

$$W_5(A_2\{A_5,A_6\}) = \min\,(c_{25}+W_6(A_5\{A_6\}),\, \underline{c_{26}+W_6(A_6\{A_5\})}) =$$
$$= \min\left(15,14\right) = 14;$$

$$W_5(A_3\{A_1,A_2\}) = \min\,(\underline{c_{31}+W_6(A_1\{A_2\})},\, \underline{c_{32}+W_6(A_2\{A_1\})}) =$$
$$= \min\left(26,26\right) = 26;$$

$$W_5(A_3\{A_1,A_4\}) = \min\,(c_{31}+W_6(A_1\{A_4\}),\, \underline{c_{34}+W_6(A_4\{A_1\})}) =$$
$$= \min\left(31,23\right) = 23;$$

$$W_5(A_3\{A_1,A_5\}) = \min\,(c_{31}+W_6(A_1\{A_5\}),\, \underline{c_{35}+W_6(A_5\{A_1\})}) =$$
$$= \min\left(28,17\right) = 17;$$

$$W_5(A_3\{A_1,A_6\}) = \min\,(\underline{c_{31}+W_6(A_1\{A_6\})},\, c_{36}+W_6(A_6\{A_1\})) =$$
$$= \min\left(18,22\right) = 18;$$

$$W_5(A_3\{A_2,A_4\}) = \min\,(c_{32}+W_6(A_2\{A_4\}),\, \underline{c_{34}+W_6(A_4\{A_2\})}) =$$
$$= \min\left(25,14\right) = 14;$$

$$W_5(A_3\{A_2,A_5\}) = \min\,(c_{32}+W_6(A_2\{A_5\}),\, \underline{c_{35}+W_6(A_5\{A_2\})}) =$$
$$= \min\left(19,11\right) = 11;$$

$$W_5(A_3\{A_2,A_6\}) = \min\,(\underline{c_{32}+W_6(A_2\{A_6\})},\, c_{36}+W_6(A_6\{A_2\})) =$$
$$= \min\left(12,19\right) = 12;$$

$$W_5(A_3\{A_4,A_5\}) = \min\,(\underline{c_{34}+W_6(A_4\{A_5\})},\, c_{35}+W_6(A_5\{A_4\})) =$$
$$= \min\left(14,21\right) = 14;$$

$$W_5(A_3\{A_4, A_6\}) = \min\,(\underline{c_{34} + W_6(A_4\{A_6\}}), c_{36} + W_6(A_6\{A_4\})) =$$
$$= \min\left(17, 29\right) = 17;$$

$$W_5(A_3\{A_5, A_6\}) = \min\,(c_{35} + W_6(A_5\{A_6\}), c_{36} + W_6(A_6\{A_5\})) =$$
$$= \min\left(10, 20\right) = 10;$$

$$W_5(A_4\{A_1, A_2\}) = \min\,(c_{41} + W_6(A_1\{A_2\}), c_{42} + W_6(A_2\{A_1\})) =$$
$$= \min\left(21, 25\right) = 21;$$

$$W_5(A_4\{A_1, A_3\}) = \min\,(c_{41} + W_6(A_1\{A_3\}), c_{43} + W_6(A_3\{A_1\})) =$$
$$= \min\left(17, 26\right) = 17;$$

$$W_5(A_4\{A_1, A_5\}) = \min\,(c_{41} + W_6(A_1\{A_5\}), c_{45} + W_6(A_5\{A_1\})) =$$
$$= \min\left(23,17\right) = 17;$$

$$W_5(A_4\{A_1, A_6\}) = \min\,(c_{41} + W_6(A_1\{A_6\}), c_{46} + W_6(A_6\{A_1\})) =$$
$$= \min\left(13, 22\right) = 13;$$

$$W_5(A_4\{A_2, A_3\}) = \min\,(c_{42} + W_6(A_2\{A_3\}), c_{43} + W_6(A_3\{A_2\})) =$$
$$= \min\left(7,13\right) = 7;$$

$$W_5(A_4\{A_2, A_5\}) = \min\,(c_{42} + W_6(A_2\{A_5\}), c_{45} + W_6(A_5\{A_2\})) =$$
$$= \min\left(18,11\right) = 11;$$

$$W_5(A_4\{A_2, A_6\}) = \min\,(c_{42} + W_6(A_2\{A_6\}), c_{46} + W_6(A_6\{A_2\})) =$$
$$= \min\left(11,18\right) = 11;$$

$$W_5(A_4\{A_3, A_5\}) = \min(c_{43} + W_6(A_3\{A_5\}), c_{45} + W_6(A_4\{A_5\})) =$$
$$= \min\left(12,13\right) = 12;$$

$$W_5(A_4\{A_3,A_6\}) = \min\,(c_{43}+W_6(A_6\{A_3\}),\,c_{46}+W_6(A_6\{A_3\})) =$$
$$= \min\left(16,27\right) = 16;$$

$$W_5(A_4\{A_5,A_6\}) = \min\,(c_{45}+W_6(A_5\{A_6\}),\,c_{46}+W_6(A_6\{A_5\})) =$$
$$= \min\left(10,19\right) = 10;$$

$$W_5(A_5\{A_1,A_2\}) = \min\,(c_{51}+W_6(A_1\{A_2\}),\,c_{52}+W_6(A_2\{A_1\})) =$$
$$= \min\left(17,24\right) = 17;$$

$$W_5(A_5\{A_1,A_3\}) = \min\,(c_{51}+W_6(A_1\{A_3\}),\,c_{53}+W_6(A_3\{A_1\})) =$$
$$= \min\left(13,32\right) = 13;$$

$$W_5(A_5\{A_1,A_4\}) = \min\,(c_{51}+W_6(A_1\{A_4\}),\,c_{54}+W_6(A_4\{A_1\})) =$$
$$= \min\left(22,23\right) = 22;$$

$$W_5(A_5\{A_1,A_6\}) = \min(c_{51}+W_6(A_1\{A_6\}),\,c_{56}+W_6(A_6\{A_1\})) =$$
$$= \min\left(9,17\right) = 9;$$

$$W_5(A_5\{A_2,A_3\}) = \min(c_{52}+W_6(A_2\{A_3\}),\,c_{53}+W_6(A_3\{A_2\})) =$$
$$= \min\left(6,19\right) = 6;$$

$$W_5(A_5\{A_2,A_4\}) = \min(c_{52}+W_6(A_2\{A_4\}),\,c_{54}+W_6(A_4\{A_2\})) =$$
$$= \min\left(23,14\right) = 14;$$

$$W_5(A_5\{A_2,A_6\}) = \min(c_{52}+W_6(A_2\{A_6\}),\,c_{56}+W_6(A_6\{A_2\})) =$$
$$= \min\left(10,13\right) = 10;$$

$$W_5(A_5\{A_3,A_4\}) = \min(c_{53}+W_6(A_3\{A_4\}),\,c_{54}+W_6(A_4\{A_3\})) =$$
$$= \min\left(27,9\right) = 9;$$

$$W_5(A_5\{A_3,A_6\})=\min(c_{53}+W_6(A_3\{A_6\}), c_{56}+W_6(A_6\{A_3\}))=$$
$$=\min\left(22,22\right)=22;$$

$$W_5(A_5\{A_4,A_6\})=\min(c_{54}+W_6(A_4\{A_6\}), c_{56}+W_6(A_6\{A_4\}))=$$
$$=\min\left(17,23\right)=17;$$

$$W_5(A_6\{A_1,A_2\})=\min(c_{61}+W_6(A_1\{A_2\}), c_{62}+W_6(A_2\{A_1\}))=$$
$$=\min\left(16,25\right)=16;$$

$$W_5(A_6\{A_1,A_3\})=\min(c_{61}+W_6(A_1\{A_3\}), c_{63}+W_6(A_3\{A_1\}))=$$
$$=\min\left(12,40\right)=12;$$

$$W_5(A_6\{A_1,A_4\})=\min(c_{61}+W_6(A_1\{A_4\}), c_{64}+W_6(A_4\{A_1\}))=$$
$$=\min\left(21,24\right)=21;$$

$$W_5(A_6\{A_1,A_5\})=\min(c_{61}+W_6(A_1\{A_5\}), c_{65}+W_6(A_5\{A_1\}))=$$
$$=\min\left(18,18\right)=18;$$

$$W_5(A_6\{A_2,A_3\})=\min(c_{62}+W_6(A_2\{A_3\}), c_{63}+W_6(A_3\{A_2\}))=$$
$$=\min\left(7,27\right)=7;$$

$$W_5(A_6\{A_2,A_4\})=\min(c_{62}+W_6(A_2\{A_4\}), c_{64}+W_6(A_4\{A_2\}))=$$
$$=\min\left(24,15\right)=15;$$

$$W_5(A_6\{A_2,A_5\})=\min(c_{62}+W_6(A_2\{A_5\}), c_{65}+W_6(A_5\{A_2\}))=$$
$$=\min\left(18,12\right)=12;$$

$$W_5(A_6\{A_3,A_4\})=\min(c_{63}+W_6(A_3\{A_4\}), c_{64}+W_6(A_4\{A_3\}))=$$
$$=\min\left(35,10\right)=10;$$

$$W_5(A_6\{A_3, A_5\}) = \min(c_{63} + W_6(A_3\{A_5\}), c_{65} + W_6(A_5\{A_3\})) =$$
$$= \min\left(26{,}14\right) = 14;$$

$$W_5(A_6\{A_4, A_5\}) = \min(c_{64} + W_6(A_4\{A_5\}), c_{65} + W_6(A_5\{A_4\})) =$$
$$= \min\left(15{,}22\right) = 15.$$

Ввиду громоздкости дальнейших вычислений, приводится выполнение только предпоследнего и последнего шагов алгоритма.

Шаг 6. Рассматривается второй этап принятия решений. Предполагается, что коммивояжер находится в городе A_i, ему необходимо посетить пять промежуточных городов A_{u1}, A_{u2}, A_{u3}, A_{u4}, A_{u5} и переехать в последний город A_7, т.е. $A_i \neq A_{u1} \neq A_{u2} \neq A_{u3} \neq A_{u4} \neq A_{u5} \neq A_7$. В качестве состояния выбирается следующая конструкция $A_i\{A_{u1}, A_{u2}, A_{u3}, A_{u4}, A_{u5}\}$. Уравнение Беллмана имеет следующий вид:

$$W_2(A_i\{A_{u1}, A_{u2}, A_{u3}, A_{u4}, A_{u5}\}) = \min\,(c_{AiAu1} + W_3(A_{u1}\{A_{u2}, A_{u3}, A_{u4}, A_{u5}\}),$$
$$c_{AiAu2} + W_3(A_{u2}\{A_{u1}, A_{u3}, A_{u4}, A_{u5}\}), c_{AiAu3} + W_3(A_{u3}\{A_{u1}, A_{u2}, A_{u4}, A_{u5}\}),$$
$$c_{AiAu4} + W_3(A_{u4}\{A_{u1}, A_{u2}, A_{u3}, A_{u5}\}), c_{AiAu5} + W_3(A_{u5}\{A_{12}, A_{u2}, A_{u3}, A_{u4}\})).$$

$$W_2(A_1\{A_2, A_3, A_4, A_5, A_6\}) = \min(c_{12} + W_3(A_2\{A_3, A_4, A_5, A_6\}),$$
$$c_{13} + W_3(A_3\{A_2, A_4, A_5, A_6\}), c_{14} + W_3(A_4\{A_2, A_3, A_5, A_6\}), c_{15} +$$
$$+W_3(A_5\{A_2, A_3, A_4, A_6\}), c_{16} + \underline{W_3(A_6\{A_2, A_3, A_4, A_5\}})) =$$
$$= \min\,(23{,}24{,}17{,}24{,}15) = 15;$$

$$W_2(A_2\{A_1, A_3, A_4, A_5, A_6\}) = \min\,(c_{12} + W_3(A_2\{A_3, A_4, A_5, A_6\}),$$
$$c_{23} + W_3(A_3\{A_1, A_4, A_5, A_6\}), c_{24} + W_3(A_4\{A_1, A_3, A_5, A_6\}), c_{25} +$$
$$+W_3(A_5\{A_1, A_3, A_4, A_6\}), c_{26} + W_3(A_6\{A_1, A_3, A_4, A_5\})) =$$
$$= \min\,(23{,}16{,}19{,}21{,}18) = 16;$$

$$W_2(A_3\{A_1,A_2,A_4,A_5,A_6\}) = \min\,(c_{31}+W_3(A_1\{A_2,A_4,A_5,A_6\}),$$
$$c_{32}+W_3(A_2\{A_1,A_4,A_5,A_6\}),\, c_{34}+W_3(A_4\{A_1,A_2,A_5,A_6\}), c_{35}+$$
$$+W_3(A_5\{A_1,A_2,A_4,A_6\}),\, c_{36}+W_3(A_6\{A_1,A_2,A_4,A_5\})) =$$
$$= \min\,(29,21,20,20,26) = 20;$$

$$W_2(A_4\{A_1,A_2,A_3,A_5,A_6\}) = \min\,(c_{41}+W_3(A_1\{A_2,A_3,A_5,A_6\}),$$
$$c_{42}+W_3(A_2\{A_1,A_3,A_5,A_6\}),\, c_{43}+W_3(A_3\{A_1,A_2,A_5,A_6\}), c_{45}+$$
$$+W_3(A_5\{A_1,A_2,A_3,A_6\}),\, c_{46}+W_3(A_6\{A_1,A_2,A_3,A_5\})) =$$
$$= \min\,(17,15,18,13,24) = 13;$$

$$W_2(A_5\{A_1,A_2,A_3,A_4,A_6\}) = \min\,(c_{51}+W_3(A_1\{A_2,A_3,A_4,A_6\}),$$
$$c_{52}+W_3(A_2\{A_1,A_3,A_4,A_6\}),\, c_{53}+W_3(A_3\{A_1,A_2,A_4,A_6\}), c_{54}+$$
$$+W_3(A_4\{A_1,A_2,A_3,A_6\}),\, c_{56}+W_3(A_6\{A_1,A_2,A_3,A_4\})) =$$
$$= \min\,(16,16,30,19,16) = 16;$$

$$W_2(A_6\{A_1,A_2,A_3,A_4,A_5\}) = \min\,(c_{61}+W_3(A_1\{A_2,A_3,A_4,A_5\}),$$
$$c_{62}+W_3(A_2\{A_1,A_3,A_4,A_5\}),\, c_{63}+W_3(A_3\{A_1,A_2,A_4,A_5\}), c_{64}+$$
$$+W_3(A_4\{A_1,A_2,A_3,A_5\}),\, c_{65}+W_3(A_5\{A_1,A_2,A_3,A_4\})) =$$
$$= \min\,(14,22,35,21,17) = 14.$$

Шаг 7. Рассматривается первый этап принятия решений. Коммивояжер находится в фиксированном городе A_7, ему необходимо посетить остальные города $A_1, A_2, A_3, A_4, A_5, A_6$ и вернуться в город A_7. Состояние определяется, как $A_7\{A_1, A_2, A_3, A_4, A_5, A_6\}$. Уравнение Беллмана имеет следующий вид:

$$W_1(A_7\{A_1,A_2,A_3,A_4,A_5,A_6\}) = \min\,(c_{71}+W_2(A_1\{A_2,A_3,A_4,A_5,A_6\}),$$
$$c_{72}+W_2(A_2\{A_1,A_3,A_4,A_5,A_6\}),\, c_{73}+W_2(A_3\{A_1,A_2,A_4,A_5,A_6\}),$$
$$c_{74}+W_2(A_4\{A_1,A_2,A_3,A_5,A_6\}),\, c_{75}+W_2(A_5\{A_1,A_2,A_3,A_4,A_6\}),$$
$$c_{76}+W_2(A_6\{A_1,A_2,A_3,A_4,A_5\})) = \min\left(17,20,32,18,29,22\right) = 17.$$

Таким образом, длина оптимального маршрута равна 17. Для нахождения городов, входящих в этот маршрут, рассматриваемую задачу необходимо решить в обратном порядке. Оптимальная последовательность состояний следующая:

$$A_7\{A_1, A_2, A_3, A_4, A_5, A_6\} \to A_6\{A_2, A_3, A_4, A_5\}$$

$$A_4\{A_2, A_3, A_5\} \to A_5\{A_2, A_3\} \to A_2\{A_3\} \to A_3\{0\}.$$

Следовательно, *оптимальный маршрут*

$A_7 \to A_1 \to A_6 \to A_4 \to A_5 \to A_2 \to A_3 \to A_7$.

ПРАКТИЧЕСКИЕ ЗАДАНИЯ ПО ДИСЦИПЛИНЕ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

Вариант выбирается согласно трем последним цифрам шифра k_3 , k_2 и k_1 – последние $k_3k_2k_1$ цифры номера зачетной книжки студента (если цифра шифра 0, то +1).

Lec_1 (1) ЗАДАНИЕ I Методом динамического программирования найти оптимальное решение для задачи коммивояжера. Известно: множество городов $A = \{A_1, A_2, A_3, A_4, A_5, A_6\}$ и матрица расстояний:

$$C = \begin{Vmatrix} \infty & 5 & 9 & 6 & 3 & 5 \\ k_1 & \infty & 8 & 8 & 5 & 9 \\ 6 & 9 & \infty & 1 & 6 & 7 \\ 7 & k_2 & 4 & \infty & 4 & 2 \\ 4 & 6 & 3 & 2 & \infty & 2 \\ 5 & 2 & 2 & k_3 & 1 & \infty \end{Vmatrix}.$$

Lec_2 ЗАДАНИЕ II Руководство корпорации решило провести реконструкцию пяти заводов. Общий объем инвестиций, равный 500 единиц, необходимо распределить между заводами так, чтобы добиться максимальной суммарной прибыли. Величина прибыли $q_i(v_j)$, получаемой от инвестирования v_j средств в i-й завод корпорации, задается табл 1.

Таблица 1

Величина прибыли

Инвестиции v_j	Заводы				
	1	2	3	4	5
0	0	0	0	0	0
100	40	50	30	60	50
200	50	$70+k_1$	55	75	60
300	65	85	$70+k_2$	95	80
400	75	95	95	110	$100+k_3$
500	85	110	120	125	130

Lec_3 ЗАДАНИЕ III Методом динамического программирования решить минимаксную задачу распределения программных модулей между процессорами для следующих исходных данных: количество модулей $m = 6$; общее количество процессоров $D_0 = 15$; матрица времени выполнения программных модулей в зависимости от количества выделяемых процессоров

$$T = \begin{Vmatrix} 12 & 14 & 0 & 0 & 0 & 0 \\ 7 & 10 & 9 & 0 & 0 & 0 \\ 2 & 7 & 5 & 9 & 0 & 0 \\ 0 & 4 & 2 & 6 & 8 & 0 \\ 0 & 2 & 0 & 4 & 6 & 7 \\ 0 & 0 & 0 & k_1 & 4 & 4 \\ 0 & 0 & 0 & 0 & k_2 & 2 \\ 0 & 0 & 0 & 0 & 0 & k_3 \end{Vmatrix}$$

Lec_4 ЗАДАНИЕ IV Методом динамического программирования определить оптимальный план производства предприятия для следующих исходных данных: количество интервалов планирования N=4*;* величина спроса на продукцию постоянна для всех этапов планирования: $d_1 = d_2 = d_3 = d_4 = 6$; затраты на производство и хранение продукции $C_n(x_n i_{n-1})$ =12+ k_2 x_n+1·i_{n-1}; затраты на формирование начального запаса $C_0\left(i_0\right)=\left(k_1+k_3\right)i_0$; ограничение на производственные мощности x_{max}=6; ограничение на предельный уровень запасов i_{max}= 4.

Lec_5 ЗАДАНИЕ V Методом динамического программирования необходимо найти оптимальное решение задачи о ранце для следующих исходных данных: количество предметов n = 12;
D=D_0 = 95
ценность предметов c = (5, k_1, 12, 14, k_2, 6, 5,10, k_3, 8, 9, 4);
вес предметов d = (20, 15, 24, 28, 15, 8, 9,10, 12, 16, 21,17).

Lec_6 ЗАДАНИЕ VI Методом динамического программирования найти оптимальное решение для задачи трех станков для следующих исходных данных: количество деталей n=5; матрица трудоемкостей:

Таблица 2

Матрица трудоемкостей

Станки	Номера деталей				
	1	2	3	4	5
a_i	8	9	5	4	6
b_i	7	4	9	3+k_2	2+k_3
c_i	9	5	3+k_1	6	5

РЕКОМЕНДАТЕЛЬНЫЙ БИБЛИОГРАФИЧЕСКИЙ СПИСОК

Основная литература

1. *Лисьев, Г.А.* Технологии поддержки принятия решений: учебное пособие / Г.А. Лисьев, И.В. Попова. – 3-е изд., стереотип. - Москва: Флинта, 2017. – 133 с. http://biblioclub.ru/index.php?page=book&id=103806

2. *Мендель, А.В.* Модели принятия решений: учебное пособие / А.В. Мендель. – Москва: Юнити-Дана, 2015. – 463 с. http://biblioclub.ru/index.php?page=book&id=115173.

3. *Ехлаков, Ю.П.* Теоретические основы автоматизированного управления / Ю.П. Ехлаков. – Томск: Томский государственный университет систем управления и радиоэлектроники, 2001. – 338 с. http://biblioclub.ru/index.php?page=book&id=208590 .

4. *Горелик, В.А.* Теория принятия решений: учебное пособие для магистрантов / В.А. Горелик; Министерство образования и науки Российской Федерации, Московский педагогический государственный университет. – Москва: МПГУ, 2016. – 152 с. http://biblioclub.ru/index.php?page=book&id=472093.

Дополнительная литература

5. *Соловьев, Н.* Основы теории принятия решений для программистов: учебное пособие / Н. Соловьев, Е. Чернопрудова, Д.А. Лесовой; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Оренбургский государственный университет». – Оренбург: ОГУ, 2012. – 187 с. http://biblioclub.ru/index.php?page=book&id=270301.

СОДЕРЖАНИЕ

СИСТЕМЫ ПОДДЕРЖКИ
ПРИНЯТИЯ РЕШЕНИЙ

Методические указания к практическим занятиям
для студентов магистратуры направления 09.04.02

Сост. *И.В. Иванова*

Печатается с оригинал-макета, подготовленного кафедрой
информационных систем и вычислительной техники

Ответственный за выпуск *И.В. Иванова*

Лицензия ИД № 06517 от 09.01.2002

Подписано к печати 07.09.2021 . Формат 60×84/16.
Усл. печ. л. 1,9. Усл.кр.-отт. 1,9. Уч.-изд.л. 1,6. Тираж 50 экз. Заказ 798.

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