**Лабораторная работа 6.**

**Модели динамического программирования.**

**Динамическое программирование** — это вычислительный метод для решения задач определенной структуры. Возникло и сформировалось в 1950-1953 гг. благодаря работам Р. Беллмана над динамическими задачами управления запасами. В упрощенной формулировке динамическое программирование представляет собой направленный последовательный перебор вариантов, который обязательно приводит к глобальному максимуму.

Основные необходимые свойства задач, к которым возможно применить этот принцип:

1. Задача должна допускать интерпретацию как *n*-шаговый процесс принятия решений.
2. Задача должна быть определена для любого числа шагов и иметь структуру, не зависящую от их числа.
3. При рассмотрении *k*-шаговой задачи должно быть задано некоторое множество параметров, описывающих состояние системы, от которых зависят оптимальные значения переменных. Причем это множество не должно изменяться при увеличении числа шагов.
4. Выбор решения (управления) на *k*-м шаге не должен оказывать влияния на предыдущие решения, кроме необходимого пересчета переменных.

Задача о выборе траектории, задача последовательного принятия решения, задача об использовании рабочей силы, задача управления запасами — классические задачи динамического программирования.

**Постановка задачи динамического программирования.**

Постановку задачи динамического программирования рассмотрим на примере инвестирования, связанного с распределением средств между предприятиями. В результате управления инвестициями система последовательно переводится из начального состояния ***S0*** В конечное ***Sn***. Предположим, что управление можно разбить на n шагов и решение принимается последовательно на каждом шаге, а управление представляет собой совокупность n пошаговых управлений. На каждом шаге необходимо определить два типа переменных - переменную состояния системы ***Sk*** переменную управления ***xk*** . Переменная ***Sk*** определяет, в каких состояниях может оказаться система на рассматриваемом *k*-м шаге. В зависимости от состояния ***S*** на этом шаге можно применить некоторые управления, которые характеризуются переменной xk которые удовлетворяют определенным ограничениям и называются допустимыми. Допустим. http://matmetod-popova.narod.ru/theme211/x_.GIF***= (x1, x2, ..., xk, ..., xn )*** - управление, переводящее систему на состояния ***S0*** в состояние ***Sn***, a ***Sk*** - есть состояние системы на *k*-м шаге управления. Тогда последовательность состояний системы можно представить в виде графа, представленного на рис. 2.11.

|  |
| --- |
| http://matmetod-popova.narod.ru/theme211/picture2_11_1.GIF |
| рис. 2.11 |

Применение управляющего воздействия ***xk*** на каждом шаге переводит систему в новое состояние ***S1 (S, xk)*** и приносит некоторый результат ***Wk (S, xk)***. Для каждого возможного состояния на каждом шаге среди всех возможных управлений выбирается оптимальное управление ***x\*k*** , такое, чтобы результат, который достигается за шаги с *k*-го по последний *n*-й, оказался бы оптимальным. Числовая характеристика этого результата называется функцией Беллмана ***Fk (S)*** и зависит от номера шага ***k*** и состояния системы ***S***. Задача динамического программирования формулируется следующим образом: требуется определить такое управление , переводящее систему из начального состояния ***S0*** в конечное состояние ***Sn*** , при котором целевая функция принимает наибольшее (наименьшее) значение ***F(S0,) => extr***.

***Рассмотрим более подробно особенности математической модели динамического программирования:***

1. задача оптимизации формулируется как конечный многошаговый процесс управления;
2. целевая функция (выигрыш) является аддитивной и равна сумме целевых функций каждого шага:  
   http://matmetod-popova.narod.ru/theme211/example_2_11_1.GIF
3. выбор управления ***xk*** на каждом шаге зависит только от состояния системы ***k*** этому шагу ***Sk-1***, и не влияет на предшествующие шаги (нет обратной связи);
4. состояние системы ***Sk*** после каждого шага управления зависит только от предшествующего состояния системы ***Sk-1*** и этого управляющего воздействия ***xh*** (отсутствие последействия) и может быть записано в виде уравнения состояния: ***Sk = fk(Sk-1 , xk), k = 1, n***;
5. на каждом шаге управление ***xk*** зависит от конечного числа управляющих переменных, а состояние системы зависит ***Sk*** – от конечного числа параметров;
6. оптимальное управление представляет собой вектор , определяемый последовательностью оптимальных пошаговых управлений: http://matmetod-popova.narod.ru/theme211/x_.GIF***= (x\*1, x\*2, ..., x\*k, ..., x\*n )***, число которых и определяет количество шагов задачи.

Принцип оптимальности и математическое описание динамического процесса управления.   
В основе метода ДП лежит принцип оптимальности, впервые сформулированный в 1953 г. американским математиком Р.Э.Беллманом: *каково бы ни было состояние системы в результате какого-либо числа шагов, на ближайшем шаге нужно выбирать управление так, чтобы оно в совокупности с оптимальным управлением на всех последующих шагах приводило к оптимальному выигрышу на всех оставшихся шагах, включая выигрыш на данном шаге.* При решении задачи на каждом шаге выбирается управление, которое должно привести к оптимальному выигрышу. Если считать все шаги независимыми, тогда оптимальным управлением будет то управление, которое обеспечит максимальный выигрыш именно на данном шаге. Однако, например, при покупке новой техники взамен устаревшей на ее приобретение затрачиваются определенные средства, поэтому доход от ее эксплуатации в начале может быть небольшой, а в следующие годы новая техника будет приносить больший доход. И наоборот, если принято решение оставить старую технику для получения дохода в текущем году, то в дальнейшем это приведет к значительным убыткам. Этот пример демонстрирует следующий факт: в многошаговых процессах управление на каждом конкретном шаге надо выбирать с учетом его будущих воздействий на весь процесс. Кроме того, при выборе управления на данном шаге следует учитывать возможные варианты состояния предыдущего шага. Например, при определении количества средств, вкладываемых в предприятие в *i*-м году, необходимо знать, сколько средств осталось в наличия к атому году и какой доход получен в предыдущем (*i* - 1)-м году. Таким образом, при выборе шагового управления необходимо учитывать следующие требования:

1. возможные исходы предыдущего шага ***Sk-1*** ;
2. влияние управления ***xk*** на все оставшиеся до конца процесса шаги (***n-k***).

В задачах динамического программирования первое требование учитывают, делая на каждом шаге условные предположения о возможных вариантах окончания предыдущего шага и проводя для каждого из вариантов условную оптимизацию. Выполнение второго требования обеспечивается тем, что в этих задачах условная оптимизация проводится от конца процесса к началу.

**Условная оптимизация**

На первом этапе решения задачи, называемом условной оптимизацией, определяются функция Беллмана и оптимальные управления для всех возможных состояний на каждом шаге, начиная с последнего в соответствии с алгоритмом обратной прогонки. На последнем, *n*-м шаге оптимальное управление - ***х\*n*** определяется функцией Беллмана: ***F(S) = max {Wn (S, xn )}***, в соответствии с которой максимум выбирается из всех возможных значений ***xn*** , причем ***xn € X***.   
Дальнейшие вычисления производятся согласно рекуррентному соотношению, связывающему функцию Беллмана на каждом шаге с этой же функцией, но вычисленной на предыдущем шаге. В общем виде это уравнение имеет вид ***Fn(S) = max {Wn (S, xn ) + Fk+1(Sn(S, xk )}*** ***xk € X***.   
Этот максимум (или минимум) определяется по всем возможным для ***k*** и ***S*** значениям переменной управления ***X***.

**Безусловная оптимизация**

После того, как функция Беллмана и соответствующие оптимальные управления найдены для всех шагов с *n*-го по первый, осуществляется второй этап решения задачи, называемый безусловной оптимизацией. Пользуясь тем, что на первом шаге (***k = 1***) состояние системы известно - это ее начальное состояние ***S0*** , можно найти оптимальный результат за все ***n*** шагов и оптимальное управление на первом шаге ***x1***, которое этот результат доставляет. После применения этого управления система перейдет в другое состояние ***S1(S, x\*1)***, зная которое, можно, пользуясь результатами условной оптимизации, найти оптимальное управление на втором шаге ***x\*2***, и так далее до последнего *n*-го шага. Вычислительную схему динамического программирования можно строить на сетевых моделях, а также по алгоритмам прямой прогонки (от начала) и обратной прогонки (от конца к началу). Рассмотрим примеры решения различных по своей природе задач, содержание которых требует выбора переменных состояния и управления.

**Выбор оптимального маршрута перевозки грузов.**

Математический аппарат ДП, основанный на методология пошаговой оптимизации, может быть использовав при нахождении кратчайших расстояний, например, на географической карте, представленной в виде сети. Решение задачи по определению кратчайших расстояний между пунктами отправления и пунктами получения продукции по существующей транспортной сети является исходным этапом при решении таких экономических задач, как оптимальное прикрепление потребителей за поставщиками, повышение производительности транспорта за счет сокращения непроизводительного пробега и др.

Пусть транспортная сеть состоит из **10** узлов, часть из которых соединены магистралями. На рис.2.12 показана сеть дорог и стоимости перевозки единицы груза между отдельными пунктами сети, которые проставлены у соответствующих ребер. Необходимо определить маршрут доставки груза из пункта **1** в пункт **10**, обеспечивающий наименьшие транспортные расходы. Модель транспортной сети представлена на рис. 2.12.

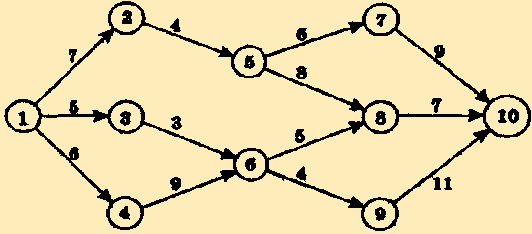


Рис. 2.12

В задаче имеется ограничение - двигаться по изображенным на схеме маршрутам можно только слева на право, т.е. попав, например, в пункт **7**, мы имеем право переместиться только в пункт **10** и не можем возвратиться обратно в **5**-й или **6**-й. Эта особенность транспортной сети дает право отнести каждый из десяти пунктов к одному из поясов. Будем считать, что пункт принадлежит *k*-му поясу, если из него попасть в конечный пункт ровно за ***k*** шагов, т.е. с заездом ровно в (*k* - 1)-й промежуточный пункт. Таким образом, пункты **7, 8** и **9** принадлежат к первому поясу, **5** и **6** - ко второму, **2, 3** и **4** - к третьему и **1** - к четвертому. Тогда на *k*-м шаге будем находить оптимальные маршруты перевозки груза из пунктов *k*-го пояса до конечного пункта. Оптимизацию будем производить с конца процесса, и потому, дойдя до *k*-го шага, неизвестно, в каком из пунктов *k*-го пояса окажется груз, перевозимый из первого пункта.

Введем обозначения:  ***k*** - номер шага (***k = 1, 2,3,4***);   
***i*** - пункт, из которого осуществляются перевозки (***i = 1,2,..., 9***); ***j*** - пункт, в который доставляется груз (***j = 2,3,.., 10***);  ***Сi, j*** - стоимость перевозки груза из пункта ***i*** в пункт ***j***.   
***Fk (i)*** - минимальные затраты на перевозку груза на *k*-м шаге решения задачи из пункта ***i*** до конечного пункта.

Очевидно, что минимум затрат на перевозку груза из пунктов *k*-го пояса до пункта **10** будет зависеть от того, в каком пункте этого пояса мы оказались. Номер ***i*** пункта, принадлежащего *k*-му поясу, будет являться переменной состояния системы на *k*-м шаге. Поскольку оптимизация осуществляется с конца процесса, то, находясь в некотором пункте ***i*** *k*-го пояса, принимается решение о перемещении груза в один из пунктов (*k* – 1)-го пояса, а направление дальнейшего движения известно из предыдущих шагов. Номер ***j*** пункта (*k* - 1)-го пояса будет переменной управления на k-м шаге.

Для первого шага управления (*k* - 1) функция Беллмана представляет собой минимальные затраты на перевозку груза из пунктов **1**-го пояса в конечный пункт, т.е. ***F1(i)*** = ***Сi  10***. Для последующих шагов затраты складываются из двух слагаемых - стоимости перевозки груза ***Сi, j*** из пункта ***i*** *k*-го пояса в пункт ***j*** (*k* - 1)-го пояса и минимально возможных затрат на перевозку из пункта j до конечного пункта, т.е. ***- Fk - 1 (i)***. Таким образом, функциональное уравнение Беллмана будет иметь вид

http://matmetod-popova.narod.ru/theme212/example_2_12_1.GIF

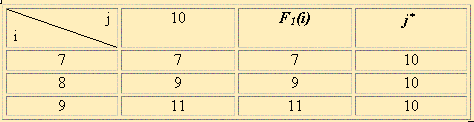
Минимум затрат достигается на некотором значении ***j\****, которое является оптимальным направлением движения из пункта ***i*** в конечный пункт. На четвертом шаге попадаем на **4**-й пояс и состояние системы становится определенным ***i = 1***. Функция ***F4(1)*** представляет собой минимально возможные затраты по перемещению груза из **1**-го пункта в **10**-й. Оптимальный маршрут определяется в результате анализа всех шагов в обратном порядке, а выбор некоторого управления ***j*** на *k*-м шаге приводит к тому, что состояние системы на (*k* - 1)-м шаге становится определенным.

**Пример 2.12.1.**

Решим сформулированную выше задачу, исходные данные которой приведены на рис. 2.12

**I этап.** ***Условная оптимизация.***1-й шаг. ***k = 1***   
***F1(i)*** = ***Сi  10***  
На первом шаге в пункт **10** груз может быть доставлен из пунктов **7,8** или **9**.

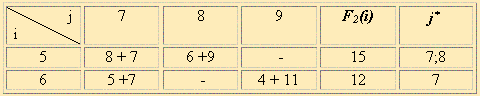
**Таблица. 2.18**



2-й шаг. ***k = 2***   
Функциональное уравнение на втором шаге принимает вид   
***http://matmetod-popova.narod.ru/theme212/example_2_12_2.GIF***

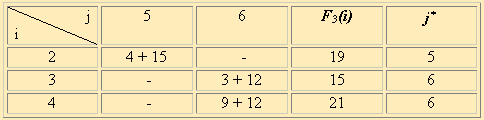
Все возможные перемещения груза на втором шаге и результаты расчета приведены в табл. 2.19

**Таблица 2.19**



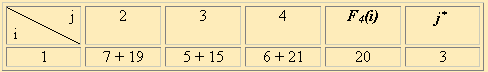
3-й шаг. ***k = 3***.   
http://matmetod-popova.narod.ru/theme212/example_2_12_3.GIF

**Таблица 2.20**



4-й шаг. ***k = 4***.  
http://matmetod-popova.narod.ru/theme212/example_2_12_4.GIF

**Таблица 2.21**



**II этап. *Безусловная оптимизация***

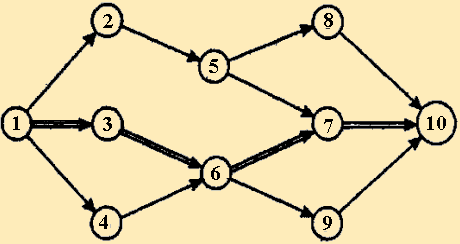


Рис.2.13.

На этапе условной оптимизации получено, что минимальные затраты на перевозку груза из пункта **1** в пункт **10** составляют ***F4(1) = 20***. Данный результат достигается при движении груза из **1**-го пункта в **3**-й. По данным табл. 2.20, из пункта **3** необходимо двигаться в пункт **6**, затем - в пункт **7** (см. табл.2.19) и из него - в конечный пункт (см. табл. 2.18). Таким образом, оптимальный маршрут доставки груза: **1** => **3** => **6** => **7** => **10**. (На рис.2.13 он показан жирными стрелками.)

**РЕШИТЬ ЗАДАЧИ:**

На заданной сети дорог имеется несколько маршрутов по доставке груза из пункта **1** в пункт **11**. стоимость перевозки единицы груза между отдельными пунктами сети проставлены у соответствующий ребер. Необходимо определить оптимальный маршрут доставки груза из пункта **1** в пункт **11**, который обеспечил бы минимальные транспортные расходы.

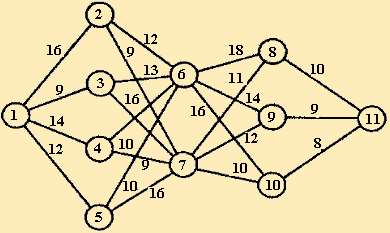


Рис. 2.14