**Лекция 9 Основные понятия динамического программирования: шаговое управление, управление операцией в целом, оптимальное управление, выигрыш на данном шаге, выигрыш за всю операцию, аддитивный критерий, мультипликативный критерий**

***Динамическое программирование*** — это способ решения сложных задач путём разбиения их на более простые подзадачи.

Ключевая идея в динамическом программировании достаточно проста. Как правило, чтобы решить поставленную задачу, требуется решить отдельные части задачи (подзадачи), после чего объединить решения подзадач в одно общее решение. Часто многие из этих подзадач одинаковы. Подход динамического программирования состоит в том, чтобы решить каждую подзадачу только один раз, сократив тем самым количество вычислений.

Для определения сущности динамического программирования представим себе некоторую операцию *О*, состоящую из ряда последовательных "шагов" или этапов, например, деятельность отрасли промышленности в течение *m*хозяйственных лет. Выигрыш (эффективность операции) Z за всю операцию складывается из выигрышей на отдельных шагах:



где – выигрыш на i-м шаге.

Если *Z* обладает таким свойством, то его называют ***аддитивным критерием***.

Операция *О* является управляемым процессом, то есть мы можем выбирать какие-то параметры, которые влияют на его ход и исход, причем на каждом шаге выбирается решение, от которого зависит выигрыш и на данном шаге, и выигрыш за операцию в целом. Эти решения называются ***шаговыми***.

Совокупность всех шаговых решений является ***управлением операцией*** ***в целом.*** Обозначим его буквой *х*, а шаговые управления – буквами *х1*, *х2, ... , хm:*

*х=х(х1, х2, ... , хm).*

Требуется найти такое управление *х*, при котором выигрыш *Z* обращается в максимум:



Управление *х\**, при котором этот максимум достигается, называется ***оптимальным управлением***. Оно состоит из совокупности оптимальных шаговых управлений: *х\*=х\*(х1\*, х2\*, ... , хm\*).*

Максимальный выигрыш, который достигается при этом управлении, обозначим следующим образом:



где *Х*– множество допустимых (возможных) управлений.

Самый простой способ решения задачи – полный перебор всех вариантов. Когда количество вариантов невелико, этот способ вполне приемлем. Однако на практике задачи с небольшим числом вариантов встречаются весьма редко, поэтому полный перебор, как правило, неприемлем из-за чрезмерных затрат вычислительных ресурсов. Поэтому в таких случаях на помощь приходит динамическое программирование.

В идее динамического программирования есть принципиальная тонкость: каждый шаг оптимизируется не сам по себе, а с "оглядкой на будущее", на последствия принимаемого "шагового" решения. Оно должно обеспечить максимальный выигрыш не на данном конкретном шаге, а на всей совокупности шагов, входящих в операцию.

***Метод динамического програмирования может применяться только для определенного класса задач.*** Эти задачи должны удовлетворять таким требованиям:

* Задача оптимизации интерпретируется как *n*-шаговый процесс управления.
* Целевая функция равна сумме целевых функций каждого шага.
* Выбор управления на *k*-м шаге зависит только от состояния системы к этому шагу и не влияет на предшествующие шаги (нет обратной связи).
* Состояние системы *sk* после *k*-го шага управления зависит только от предшествующего состояния *sk-1* и управления *x*k (отсутствие последействия).
* На каждом шаге управление *xk* зависит от конечного числа управляющих переменных, а состояние *sk*– от конечного числа параметров.

В основе решения всех задач динамического программирования лежит ***"принцип оптимальности" Беллмана***, который выглядит следующим образом: *каково бы ни было состояние системы s 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 данный*.

Этот принцип впервые был сформулирован Р. Беллманом в 1953 г.

Принцип оптимальности утверждает, что для любого процесса без обратной связи оптимальное управление таково, что оно является оптимальным для любого подпроцесса по отношению к исходному состоянию этого подпроцесса. Поэтому решение на каждом шаге оказывается наилучшим с точки зрения управления в целом.

Модели динамического программирования могут применяться, например, при разработке правил управления запасами, устанавливающими момент пополнения запасов и размер пополняющего заказа; при разработке принципов календарного планирования производства и выравнивания занятости в условиях колеблющегося спроса на продукцию; при распределении дефицитных капиталовложений между возможными новыми направлениями их использования; при составлении календарных планов текущего и капитального ремонта сложного оборудования и его замены; при разработке долгосрочных правил замены выбывающих из эксплуатации основных фондов и т.д.

**Суть метода динамического программирования.**

В основу метода динамического программирования положен *принцип оптимальности*, сформулированный в 1957 г. американским математиком Ричардом Беллманом: «Оптимальное поведение обладает тем свойством, что каковы бы ни были первоначальные состояние и решение в начальный момент времени, последующие решения должны составлять оптимальное поведение относительно состояния, получающегося в результате первого решения».

Физическая сущность принципа оптимальности заключается в том, что ошибка выбора решения в данный момент не может быть исправлена в будущем.

Рассматривается следующая общая задача. Имеется некоторая физическая система, в которой происходит какой-то процесс, состоящий из *n* шагов. Эффективность процесса характеризуется некоторым показателем *W*, который называют *выигрышем*. Пусть общий выигрыш *W* за все *n* шагов процесса складывается из выигрышей на отдельных шагах

 (2.1)

где *wi* — выигрыш на *i*-м шаге. Если *W* обладает таким свойством, то его называют *аддитивным критерием*.

Процесс, о котором идет речь, представляет собой управляемый процесс, т.е. имеется возможность выбирать какие-то параметры, влияющие на его ход и исход, причем на каждом шаге выбирается какое-то решение, от которого зависит выигрыш на данном шаге. Это решение называется *шаговым управлением*. Совокупность всех шаговых управлений представляет собой управление процессом в целом. Обозначим его буквой *U*, а шаговые управления — буквами  . Тогда



Шаговые управления  в общем случае не числа, а, как правило, векторы, функции и т.п.

В модели динамического программирования процесс на каждом шаге находится в одном из состояний *s* множества состояний *S*. Считается, что всякому состоянию сопоставлены некоторые шаговые управления. Эти управления таковы, что управление, выбранное в данном состоянии при любой предыстории процесса, определяет полностью следующее состояние процесса. Обычно выделены два особых состояния: *s*0 — начальное и *sw* — конечное.

Итак, пусть каждому состоянию  поставлено множество допустимых шаговых управлений  , и каждому шаговому управлению  , соответствует  — состояние, в которое процесс попадает из *si* в результате использования шагового управления *u*. Пусть процесс находится в начальном состоянии *s*0. Выбор  переводит процесс в состояние *s*1 = σ(*s*0,*u*1), выбор  — в состояние *s*2 = σ(*s*1,*u*2) и т.д. В результате получается траектория процесса, которая состоит из последовательности пар



и заканчивается конечным состоянием. Для единообразия можно считать, что  включает только одно состояние  , оставляющее процесс в том же конечном состоянии. Следует отметить, что множества допустимых состояний и управлений

 (2.2)

 (2.3)

конечны и *Us* для различных *s* не пересекаются.

В общем виде задача динамического программирования формулируется следующим образом: найти такую траекторию процесса, при которой выигрыш (2.1)будет максимальным.

То управление, при котором достигается максимальный выигрыш, называется *оптимальным управлением*. Оно состоит из совокупности шаговых управлений

 (2.4)

Тот максимальный выигрыш, который достигается при этом управлении обозначим *Wmax*:

*Wmax* = max*U*{*W*(*u*)}. (2.5)

Рассмотрим на примере задачи о рюкзаке, что понимается под шагом, состоянием, управлением и выигрышем.

Загрузку рюкзака можно представить себе как процесс, состоящий из *n*шагов. На каждом шаге требуется ответить на вопрос: взять данный предмет в рюкзак, или нет? Таким образом, шаг процесса — присваивание переменной *xi* значения 1 или 0.

Теперь определим состояния. Очевидно, что текущее состояние процесса характеризует остаточная грузоподъёмность рюкзака — вес, который остался в нашем распоряжении до конца (до полной укладки рюкзака). Следовательно, под состоянием перед *i*-м шагом понимается величина

 (2.6)

при этом *s*0 является начальным состоянием, которому соответствует величина *b* — исходная грузоподъемность рюкзака.

Управление на *i*-м шаге означает присваивание двоичной переменной *xi*значения 0 или 1. Значит, на каждом шаге имеем всего два управления. Причем допустимость управления *ui*, устанавливающего *xi* = 1, определяется условием

 (2.7)

Далее везде вместо переменных  будем использовать соответствующие управления  . Тогда формулы (2.6), (2.7) примут следующий вид:

 (2.8)

 (2.9)

Шаговый выигрыш можно определить как  . Поэтому

 (2.10)

Требуется найти оптимальное управление  , при котором величина выигрыша (2.10) обращается в максимум.[3]

3. Пример решения задачи методом динамического программирования.

Задание. Инвестор выделяет средства в размере 5 тыс. ден. ед., которые должны быть распределены между тремя предприятиями.

Требуется, используя принцип оптимальности Беллмана, построить план распределения инвестиций между предприятиями, обеспечивающий наибольшую общую прибыль, если каждое предприятие при инвестировании в него средств x тыс. ден. ед. приносит прибыль p;(x) тыс. ден. ед. (i=1, 2 и 3) по следующим данным:

|  |  |
| --- | --- |
| Инвестирование средств (тыс. ден. ед.) | Прибыль (тыс. ден. ед.) |
| x | pi(x) | p2 (x) | p3(x) |
|  | 3,22 | 3,33 | 4,27 |
|  | 3,57 | 4,87 | 7,64 |
|  | 4,12 | 5,26 | 10,25 |
|  |  | 7,34 | 15,93 |
|  | 4,85 | 9,49 | 16,12 |

|  |
| --- |
|   |

Решение. Составим математическую модель задачи.

1.Число шагов равно 3.

2.Пусть s - количество средств, имеющихся в наличии перед данным шагом, и характеризующих состояние системы на каждом шаге.

3. Управление на i-ом шаге (i=1,2,3) выберем xi - количество средств, инвестируемых в i- ое предприятие.

4. Выигрыш pi(xi) на i-ом шаге - это прибыль, которую приносит i-ое предприятие при инвестировании в него средств xi. Если через выигрыш в целом обозначить общую прибыль W, то W=p1(x1)+ p2(x2)+ p3(x3).

5. Если в наличии имеются средства в количестве s тыс. ден. ед. и в i-ое предприятие инвестируется x тыс. ден. ед, то для дальнейшего инвестирования остается (s-x) тыс. ден. ед. Таким образом, если на i-ом шаге система находилась в состоянии s и выбрано управление x, то на (i+1)-ом шаге система будет находится в состоянии (s-x), и, следовательно, функция перехода в новое состояние имеет вид: fi(s, x) = s-x.

6.На последнем (i=3) шаге оптимальное управление соответствует количеству средств, имеющихся в наличии, а выигрыш равен доходу, приносимым последним предприятием: x3(s)=s, W3(s)=p3(s).

7.Согласно принципу оптимальности Беллмана, управление на каждом шаге нужно выбирать так, чтобы оптимальной была сумма выигрышей на всех оставшихся до конца процесса шагах, включая выигрыш на данном шаге. Основное функциональное уравнение примет вид

W2 (s) = max{*p*2 (x) *+ W3(s -* x)}

x< *s*

Проведем пошаговую оптимизацию, по результатам которой заполним таблицу.

|  |  |  |  |
| --- | --- | --- | --- |
| s | i=3 | i=2 | i=1 |
|   | x3(s) | W3(s) | x2(s) | W2(s) | xi(s) | Wi(s) |
|  |  | 4,27 |  | 4,27 |   |   |
|  |  | 7,64 |  | 7,64 |   |   |
|  |  | 10,25 |  | 10,97 |   |   |
|  |  | 15,93 |  | 15,93 |   |   |
|  |  | 16,12 |  | 19,26 |  | 19,26 |

|  |
| --- |
|   |

В первой колонке таблицы записываются возможные состояния системы, в верхней строке - номера шагов с оптимальным управлением и выигрышем на каждом шаге, начиная с последнего. Так как для последнего шага i=3 функциональное уравнение имеет вид x3(s)=s, W3(s)=p3(s), то две колонки таблицы, соответствующие i=3, заполняются автоматически по таблице исходных данных.

На шаге i=2 основное функциональное уравнение имеет вид

W2 (s) = max{p2 (x) + W3(s - x)}

x≤ s

Поэтому для проведения оптимизации на этом шаге заполним таблицу для различных состояний s при шаге i=3.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| s | x | s-x | p2(x) | W3(s-x) | p2(x)+W3(s-x) | W2(s) |
|  |  |  |  | 4,27 | 4,27 | 4,27 |
|  |  | 3,33 |  | 3,33 |  |  |
|  |  |  |  | 7,64 | 7,64 | 7,64 |
|  |  | 3,33 | 4,27 | 7,6 |  |  |
|  |  | 4,87 |  | 4,87 |  |  |
|  |  |  |  | 10,25 | 10,25 | 10,97 |
|  |  | 3,33 | 7,64 | 10,97 |  |  |
|  |  | 4,87 | 4,27 | 9,14 |  |  |
|  |  | 5,26 |  | 5,26 |  |  |
|  |  |  |  | 15,93 | 15,93 | 15,93 |
|  |  | 3,33 | 10,25 | 13,58 |  |  |
|  |  | 4,87 | 7,64 | 12,51 |  |  |
|  |  | 5,26 | 4,27 | 9,53 |  |  |
|  |  | 7,34 |  | 7,34 |  |  |
|  |  |  |  | 16,12 | 16,12 | 19,26 |
|  |  | 3,33 | 15,93 | 19,26 |  |  |
|  |  | 4,87 | 10,25 | 15,12 |  |  |
|  |  | 5,26 | 7,64 | 12,9 |  |  |
|  |  | 7,34 | 4,27 | 11,61 |  |  |
|  |  | 9,49 |  | 9,49 |  |  |

|  |
| --- |
|   |

На шаге i=1 основное функциональное уравнение имеет вид

Wx( s) = max{ px( x) + W2( s - x)}

x ≤ s

а состояние системы перед первым шагом s=5, поэтому для проведения оптимизации на этом шаге заполним таблицу.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| s | x | s-x | pi(x) | W2(s-x) | pi(x)+W2(s-x) | Wi(s) |
|  |  |  |  | 19,26 | 19,26 | 19,26 |
|  |  | 3,22 | 15,93 | 19,15 |  |  |
|  |  | 3,57 | 10,97 | 14,54 |  |  |
|  |  | 4,12 | 7,64 | 11,76 |  |  |
|  |  |  | 4,27 | 8,27 |  |  |
|  |  | 4,85 |  | 4,85 |  |  |

Видно, что наибольшее значение выигрыша составляет 19,26. При этом оптимальное управление на первом шаге составляет x1(s1)=0 (s1=5), на втором шаге x2(s2) =1 (s2=s1-x1=5) и на третьем шаге x3(s3) =4 (s3=s2-x2=4).

Это означает, что (0, 1, 4) - оптимальный план распределения инвестиций между предприятиями.

Таким образом, для получения наибольшей общей прибыли в размере 19,26 тыс. ден. ед., необходимо вложить 1 тыс. ден. ед. во второе предприятие и 4 тыс. ден. ед. в третье предприятие.[4]

Список используемых источников

1. Беллман Р., Динамическое программирование, пер. с англ., М., 1960

2. Болтянский В. Г.,Математические методы оптимального управления, М., 1966

*Пример.*Для развития трех предприятий выделено 5 млн руб. Известна эффективность капитальных вложений в каждое предприятие, заданная функцией полезности  (*i* = 1, 2, 3). Составить оптимальный план распределения средств между предприятиями, предположив, что оно проводится в целых числах (0, 1, 2, 3, 4 и 5 млн руб.).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *x* | 0 | 1 | 2 | 3 | 4 | 5 |
| http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image026.gif | 0 | 4,1 | 4,5 | 5,1 | 6,7 | 7,0 |
| http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image028.gif | 0 | 4,0 | 5,0 | 5,5 | 6,0 | 8,0 |
| http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image030.gif | 0 | 3,1 | 4,7 | 5,3 | 5,9 | 6,5 |

 Исходные данные задачи приведены в таблице. Имеем три управляющие переменные , ,  и четыре параметра состояния , , , .

Уравнениями состояния служат равенства:

; ; ; .

Данный процесс является трехшаговым. Так как на последнем шаге процесс завершается и прибыль на «последующих» шагах отсутствует, то и

.

Запишем уравнение Беллмана для последнего шага:



и для всех предыдущих шагов:

, 

Уравнение Беллмана для любого шага:

.

Расчеты располагаем в двух таблицах – основной, в которой помещаем результаты условной оптимизации, и вспомогательной, в которой определяем  и выполняем условную оптимизацию.

В основной таблице входом является параметр  (0, 1, 2, 3, 4, 5). Условную оптимизацию начнем с расчета третьего шага.

Таблица 16.1 (основная)

|  |  |  |  |
| --- | --- | --- | --- |
| http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image066.gif | 3-й шаг | 2-й шаг | 1-й шаг |
| http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image071.gif | http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image073.gif | http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image075.gif | http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image077.gif | http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image079.gif | http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image081.gif |
| 012345 | 03,14,75,35,96,5 | 012345 | 04,07,18,79,710,3 | 011122 | 04,18,111,212,813,8 | 011111 |

 Таблица 16.2 (вспомогательная)

|  |  |  |
| --- | --- | --- |
| k = 2, 1 | 2-й шаг | 1-й шаг |
| http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image083.gif | http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image085.gif | http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image087.gif | http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image089.gif | http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image071.gif | http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image092.gif | http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image094.gif | http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image075.gif | http://www.math.mrsu.ru/text/courses/method/prost_ekon_zad.files/image097.gif |
| 1 | 01 | 10 | 04,0 | 3,10 | 0 + 3,1 = 3,1**4,0 + 0 = 4,0** | 04,1 | 4,00 | 4,0**4,1** |
|  2  | 012 | 210 | 04,05,0 | 4,73,10 | 0 + 4,7 = 4,7**4,0 + 3,1 = 7,1**5,0 + 0 = 5,0 | 04,14,5 | 7,14,00 | 7,1**8,1**4,5 |
|   3  | 0123 | 3210 | 04,05,05,5 | 5,34,73,10 | 0 + 5,3 = 5,3**4,0 + 4,7 = 8,7**5,0 + 3,1 = 8,15,5 + 0 = 5,5 | 04,14,55,1 | 8,77,14,00 | 8,7**11,2**8,55,1 |
|   4  | 01234 | 43210 | 04,05,05,56,0 | 5,95,34,73,10 | 0 + 5,9 = 5,94,0 + 5,3 = 8,3**5,0 + 4,7 = 9,7**5,5 + 3,1 = 8,66,0 + 0 = 6,0 | 04,14,55,16,7 | 9,78,77,14,00 | 9,7**12,8**11,69,16,7 |
|    5 | 012345 | 543210 | 04,05,05,56,08,0 | 6,55,95,34,73,10 | 0 + 6,5 = 6,54,0 + 5,9 = 9,9**5,0 + 5,3 = 10,3**5,5 + 4,7 = 10,26,0 + 3,1 = 9,18,0 + 0 = 8,0 | 04,14,55,16,77,0 | 10,39,78,77,14,00 | 10,3**13,8**13,212,210,77,0 |

 Перейдем к безусловной оптимизации. Из первого (последнего по порядку действий) шага условной оптимизации получаем  – максимальный доход. Здесь же получаем , т. е. первому предприятию следует выделить 1 млн. руб.

При  из уравнения состояния находим: .

В пятом столбце таблицы 1 находим . Вычисляем .

Из третьего столбца таблицы 1 получаем .

Итак, , , ;

4,1 +5,0 + 4,7 = 13,8.

Приведем решение задачи с использованием **алгоритма прямой прогонки**.

1. Предположим, что все средства отданы первому предприятию. Тогда можно записать:

.

В этом случае максимальная прибыль будет получена от вложения всех 5 млн руб. в это предприятие () и составит 7,0 млн руб.

2. Определим оптимальную стратегию при распределении средств между первым и вторым предприятиями. При этом

.

Очевидно, что .

;  .

;  .

;  .

;  .

;  .

В этом случае максимальная прибыль будет получена от вложения 1 млн. руб. во второе предприятие () и 4 млн руб. ( 4) в первое предприятие и составит

 млн руб.

3. Определим оптимальную стратегию при распределении денежных средств между третьим и первыми двумя предприятиями.

.

На третьем, последнем, шаге достаточно найти .

;  .

Таким образом, максимальный доход при распределении 5 млн руб. между тремя предприятиями составит  млн руб. При этом третьему предприятию нужно выделить 2 млн руб. ().

Тогда на долю первых двух предприятий остается

 млн руб.

Из второго шага находим млн руб. Эта прибыль достигается, если второму предприятию выделить 2 млн руб. ().

Тогда на долю первого предприятия остается

 млн руб. ().

Таким образом, оптимальный план распределения капиталовложений представлен как Х\* = (1, 2 ,2).

При таком варианте распределения средств будет получен максимальный доход:

4,1 + 5,0 + 4,7 = 13,8 (млн руб.).

Ответ:  Х\* = (1, 2 ,2), 13,8.

 **Задача 2 (Задача календарного планирования трудовых ресурсов).**Предпринимателю необходимо составить план регулирования численности рабочих на последующие пять недель. Он оценивает минимальные потребности в рабочей силе  на каждую из пяти недель следующим образом: 5, 7, 8, 4 и 6 рабочих для *i*= 1, 2, 3, 4 и 5 соответственно. Предприниматель имеет возможность регулировать количество имеющихся в наличии рабочих путем найма и увольнения. Пусть  – количество рабочих, имеющихся в наличии на *j*-й неделе.  Определим    как  величину  убытков,  связанных с тем, что  превышает заданное значение , а  –  как  величину  накладных  расходов  по найму  новых  рабочих .  Необходимо  составить  оптимальный  план  регулирования  численности  рабочих для 5-недельного периода планирования при условии, что исходное количество  рабочих, имеющихся в наличии к началу первой недели, составляет пять человек. Опишем задачу в виде модели динамического программирования.

Этап*j* ставится в соответствие *j*-й неделе. Состояние на этапе *j* выражает количество рабочих, имеющихся к концу этапа *j*– 1. Вариантырешения  описываются количеством рабочих, имеющихся на этапе *j*. Обозначим через  минимальную величину расходов, осуществленных в течение периодов времени (недель) , при заданном . Рекуррентное соотношение записывается в следующем виде:



,  *j* = 4, 3, 2, 1.

**Задача 3 (Задача о загрузке, или задача о рюкзаке (ранце)).** Самолет загружается предметами *n* различных типов. Каждый предмет типа *i*имеет вес  и стоимость . Максимальная грузоподъемность самолета равна *W*.  Требуется определить максимальную стоимость груза, вес которого не должен превышать максимальной грузоподъемности самолета.

Обозначим количество предметов типа *i* через . Составим математическую модель задачи

при ограничениях



        Каждый из трех основных элементов модели динамического программирования определяется следующим образом.

Этап *j* ставится в соответствие типу *j*, . Состояние  на этапе *j* выражает суммарный вес предметов, решения о погрузке которых приняты на этапах ; при этом  и  при . Вариантырешения   на  этапе  *j* описываются  количеством предметов типа *j*. Значение  заключено в пределах от нуля до . Пусть  – максимальная суммарная стоимость предметов, решения о погрузке которых приняты на этапах  при заданном состоянии . Рекуррентное соотношение (для процедуры обратной прогонки) имеет следующий вид:





ЗАДАНИЕ. Для двух предприятий выделено *a* единиц средств. Как распределить все

средства в течение 4 лет, чтобы доход был наибольшим, если известно, что доход от *x*

единиц средств, вложенных в первое предприятие, равен *f* (*x*) 1 , а доход от *y* единиц

средств, вложенных во второе предприятие, равен *f* (*y*) 2 . Остаток средств к концу года

составляет *g* (*x*) 1 для первого предприятия и *g* (*y*) 2 для второго предприятия. Задачу

решить методом динамического программирования.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| а | *f* (*x*) 1 | *g* (*x*) 1 | *f* (*y*) 2 . | *g* (*y*) 2 |
| 1000 | 3х | 0,1х | 2у | 0,5у |

РЕШЕНИЕ. Процесс распределения средств разобьем на 4 этапа – по соответствующим

годам.

Обозначим *k k k a* = *x* + *y* - средства, которые распределяются на *k* –ом шаге как

сумма средств по предприятиям.

Суммарный доход от обоих предприятий на *k* –ом шаге: