Министерство общего и профессионального образования
Российской Федерации

Нижегородский государственный университет им.
Н.И.Лобачевского

Экономический факультет
Кафедра экономической информатики

Динамическое программирование в экономических задачах с применением системы MATLAB

Нижний Новгород, 2006

УДК 338.23:519.876

Динамическое программирование в экономических задачах с применением системы MATLAB / Н.П.Визгунов. — Н.Новгород: ННГУ, 2006.

Методическая разработка предназначена для студентов экономического факультета дневного, вечернего и заочного обучения. На примерах показано, как решать экономические задачи, которые сводятся к задачам динамического программирования. Задачи решаются не только вручную, но и предлагаются соответствующие программы в системе MATLAB. Эти программы позволяют решать реальные экономические задачи, содержащие многие тысячи переменных. Программы являются достаточно простыми, поэтому они могут быть легко адаптированы и для решения аналогичных экономических задач. Большинство задач снабжено ответами и решениями.

Составители: доцент Н.П.Визгунов

Рецензенты: зав. кафедрой ГМУ экономического ф-та,
профессор, к.э.н. Ю.А. Лебедев.

Нижегородский государственный университет им.
Н.И.Лобачевского
2006

Содержание

Список таблиц

Список иллюстраций

1 Динамическое программирование

Динамическое программирование - это вычислительный метод для решения задач определенной структуры. Динамическое программирование возникло и сформировалось в 1950–1953 гг. благодаря работам Ричарда Беллмана и его сотрудников. Беллман (Bellman) Ричард Эрнест (1920-84) – американский математик. Основные труды по вычислительной математике и теории оптимального управления. Разработал метод динамического программирования. Задачи управления запасами были первыми задачами, которые решались этим методом.

2 Задача распределения инвестиций

Задачей распределения инвестиций будем называть следующую задачу оптимального планирования.

Задача 1 (распределения инвестиций) *Совет директоров фирмы изучает предложения по модернизации четырех предприятий. Для этих целей выделено 5 миллионов долларов. Для каждого предприятия j разработано несколько альтернативных проектов. Каждый из проектов характеризуется суммарными затратами c_j и будущими доходами R_j. На каждом предприятии можно реализовать только по одному проекту. Соответствующие данные приведены в таблице 1.*

Необходимо выбрать такие проекты для каждого предприятия, чтобы фирма получила максимальный годовой доход.

Обратим внимание, что все числа в таблице – целые. Затраты c_j измеряются в миллионах долларов, доходы R_j – в миллионах долларов в год. Через x_j обозначен номер проекта, который выбирается на j–ом предприятии. В нашей конкретной задаче проект с номером 1 пустой, расширения предприятия при этом не предполагается.

Проект	Пр-тие 1		Пр-тие 2		Пр-тие 3		Пр-тие 4	
	c_1	R_1	c_2	R_2	c_3	R_3	c_4	R_4
$x_j = 1$	0	0	$\underline{0}$	$\underline{0}$	0	0	0	0
$x_j = 2$	$\underline{1}$	$\underline{3}$	3	5	1	4	$\underline{2}$	$\underline{3}$
$x_j = 3$	–	–	5	9	$\underline{2}$	$\underline{6}$	–	–

Таблица 1. Задача распределения 5 миллионов долларов

Самый простой способ решения задачи — использовать полный перебор.

x_1	x_2	x_3	x_4	Затраты $\sum c_j(x_j)$	Доходы $\sum R_j(x_j)$	План допустим ?
1	1	1	1	0	0	Да
1	1	1	2	2	3	Да
1	1	2	1	1	4	Да
1	1	2	2	1+2=3	4+3=7	Да
1	1	3	1	2	6	Да
1	1	3	2	2+2=4	6+3=9	Да
...	...	...	...	...	...	...
1	3	3	2	0+5+2+2=9	Нет	Нет
2	1	1	1	1+0+0+0=1	3+0+0+0=3	Да
2	1	1	2	1+0+0+2=3	3+0+0+3=6	Да
...	...	...	...	...	...	...
2	3	3	2	1+5+2+2=10	Нет	Нет

Таблица 2. Полный перебор

Задача имеет $2 * 3 * 3 * 2 = 36$ возможных решений, причем некоторые из них не являются допустимыми, так как для реализации такого плана потребуется денег больше, чем 5 миллионов долларов.

Перейдем теперь к математической формулировке задачи распределения инвестиций.

$$\left.\begin{array}{r} f_1(y_1) = \max\limits_{x_1,\ldots,x_n} \left\{\sum_{j=1}^n R_j(x_j)\right\}, \\ \sum_{j=1}^n c_j(x_j) \le y_1, \\ x_j - \text{целые},\ j = 1,\ldots,n. \end{array}\right\} \quad (1)$$

В нашей задаче $y_1 = 5$ миллионов долларов, $n = 4$ – количество предприятий. Будем в дальнейшем использовать следующие обозначения.

y_j – количество денег, выделенных для расширения предприятий $j, j+1, \ldots, n$.

x_j – номер проекта, выбранного на предприятии j.

$c_j(x_j)$– затраты (от слова *cost*) на j-ом предприятии, когда выбран проект с номером x_j. Измеряется в миллионах долларов.

$R_j(x_j)$ – годовой доход (*result, revenue*), который будет получен от реализации проекта x_j. Измеряется в миллионах долларов в год.

$f_1(y_1)$ – максимальный годовой доход, который будет получен от реализации проектов $x_1, x_2, \ldots, x_n$ при заданном объеме инвестиций y_1 миллионов долларов.

Мы пользуемся довольно сложными обозначениями, но их надо просто один раз запомнить. В дальнейшем будем пользоваться только этими обозначениями и новых постараемся не вводить. В задаче 1 выберем фиксированное значение $x_1 = \tilde{x}_1$. Например, сначала $\tilde{x}_1 = 1$.

$$\left.\begin{array}{r} \tilde{f}_1(y_1) = R_1(\tilde{x}_1) + \max\limits_{x_2,\ldots,x_n} \left\{\sum_{j=2}^n R_j(x_j)\right\}, \\ c_1(\tilde{x}_1) + \sum_{j=2}^n c_j(x_j) \le y_1, \\ x_j - \text{целые},\ j = 2,\ldots,n. \end{array}\right\} \quad (2)$$

Можно записать более компактно

$$\tilde{f}_1(y_1) = R_1(\tilde{x}_1) + f_2(y_1 - c_1(\tilde{x}_1)), \tag{3}$$

где $\tilde{f}_1(y_1)$ – наибольший годовой доход при инвестициях y_1 и при фиксированном значении $\tilde{x}_1 = 1$. В нашей конкретной задаче выбираем максимум из первой половины таблицы 1, то есть не из всех 36 значений, а только из первых 18 строчек этой таблицы. $f_2(y_1 - c_1(\tilde{x}_1))$ – это годовой доход от расширения предприятий $2, \ldots, n$, если инвестировано денег $y_1 - c_1(\tilde{x}_1)$ миллионов долларов.

Остается перебрать все возможные значения $\tilde{x}_1$. В нашем конкретном примере осталось посмотреть значения доходов для второй половины таблицы 1, где значение $\tilde{x}_1 = 2$. В результате получим полный перебор, то есть найдем оптимальное значение $f_1(y_1)$.

$$f_1(y_1) = \max_{\tilde{x}_1} \{R_1(\tilde{x}_1) + f_2(y_1 - c_1(\tilde{x}_1))\}. \tag{4}$$

Тильду в обозначении переменной x_1 можно не писать, $y_2 = y_1 - c_1(x_1)$ – количество денег, инвестированных в предприятия $2, \ldots, n$. Из $y_2 \geq 0$ следует $c_1(x_1) \leq y_1$.

Окончательно получаем

$$f_1(y_1) = \max_{x_1 | c_1(x_1) \leq y_1} \{R_1(x_1) + f_2(y_1 - c_1(x_1))\}, \tag{5}$$

О чем говорит выведенная формула? Если мы умеем решать задачу распределения инвестиций для $(n-1)$ предприятия $2, \ldots, n$ для всех возможных значений инвестиций $y_2 = 0, 1, \ldots, 5$, то задача распределения $y_1 = 5$ миллионов долларов легко решается. Но вычисление $f_2(y_2)$ также можно осуществить по аналогичной формуле, используя значения доходов $f_3(y_3)$ для $(n-2)$ предприятий, где $y_3 = y_2 - c_2(x_2)$ – деньги для предприятий $3, \ldots, n$. Формула выводится по известной уже схеме:

$$\begin{aligned} f_2(y_2) = \max_{x_2 | c_2(x_2) \leq y_2} \{R_2(x_2) + f_3(y_2 - c_2(x_2))\}&, \\ y_2 \in \{0, \ldots, 5\}&. \end{aligned} \tag{6}$$

По той же схеме выводятся формулы для вычислений $f_3(y_3), \ldots, f_n(y_n)$.

Осталось записать формулу в общем случае, для вычисления максимального дохода от модернизации предприятий $j, j+1, \ldots, n$:

$$\left. \begin{aligned} & f_{n+1}(y_{n+1}) = 0, \\ & f_j(y_j) = \max_{x_j | c_j(x_j) \leq y_j} \{R_j(x_j) + f_{j+1}(y_j - c_j(x_j))\}, \\ & \qquad\qquad\qquad\qquad\qquad j = n, n-1, \ldots, 1. \end{aligned} \right\} \tag{7}$$

Эта зависимость носит название *уравнение Беллмана для процедуры обратной прогонки.* Обратная прогонка, потому что сначала надо решить задачу для последнего предприятия n, затем для предприятий $n-1, n$, затем для $n-2, n-1, n$ и так далее, на последнем шаге надо вычислить $f_1(y_1)$ для всех n предприятий $1, 2, \ldots, n$.

Аргументы функций x_j и y_j в динамическом программировании имеют свои названия.

- x_j – переменная или *управление* у Беллмана,
- y_j – параметр, *состояние* в терминологии Беллмана – количество денег для предприятий $j, \ldots, n$ в нашей задаче о распределении инвестиций.

Обозначения еще раз иллюстрируются для отрезка $j, j+1, \ldots, n$ на рисунке 1.

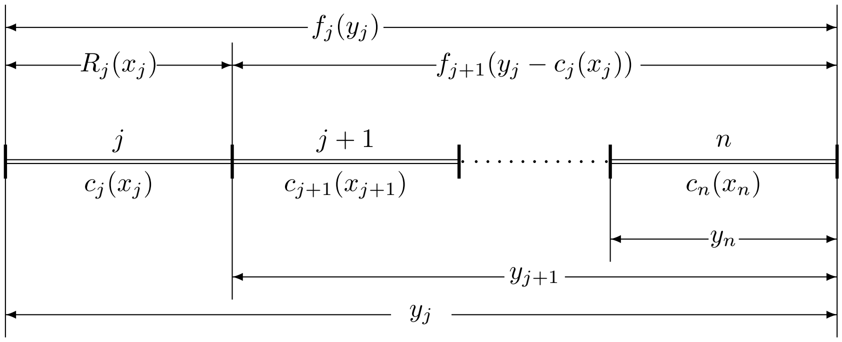

Рис. 1. Пояснение обозначений для отрезка $j, \ldots, n$ в задаче распределения инвестиций.

2.1 Решение задачи распределения инвестиций с помощью таблиц.

Рекуррентное уравнение Беллмана для процедуры обратной прогонки записывается следующим образом для нашей конкретной задачи:

$$f_5(y_5) = 0,$$
$$f_j(y_j) = \max_{x_j | c_j(x_j) \le y_j} \{R_j(x_j) + f_{j+1}(y_j - c_j(x_j))\}, \; j = 4, 3, 2, 1.$$

Сначала надо решить задачу для четвертого предприятия, затем для третьего и четвертого предприятия, затем для 2, 3 и 4 предприятия и, наконец, для предприятий 1, 2 , 3 и 4. В таблицах 3 – 6 приводятся результаты расчетов. Результаты получены с помощью таблиц, в которые помещены значения функции $\{R_j(x_j) + f_{j+1}(y_j - c_j(x_j))\}$ двух переменных y_j и x_j. Функция вычислена для каждого возможного значения состояния y_j и управления x_j. Если вычисление функции для некоторых пар (y_j, x_j) невозможно, то в качестве значения функции ставим черточку.

Этап 4. Предприятие 4.

$$f_4(y_4) = \max_{x_4 = 1:2 | c_4(x_4) \le y_4} \{R_4(x_4)\}.$$

	$R_4(x_4)$		Оптимальное решение	
y_4	$x_4 = 1$	$x_4 = 2$	$f_4(y_4)$	x_4^*
0	0	–	0	1
1	0	–	0	1
$\underline{2}$	0	3	$\underline{3}$	$\underline{2}$
3	0	3	3	2
4	0	3	3	2
5	0	3	3	2

Таблица 3. Задача об инвестициях, этап 4

2.1.1 Заполнение таблицы этапа 4.

Подробно опишем построение таблицы 3, соответствующей этапу 4. В таблице вычислены значения функции $\{R_4(x_4)\}$ для 6 возможных состояний $y_4 \in \{0, \ldots, 5\}$ (строки таблицы) и 2 возможных управлений $x_4 \in \{1, 2\}$ (столбцы таблицы). Из исходной таблицы 1, расположенной на стр. 3, видно, что $R_4(1) = 0$. (это самое правое верхнее число исходной таблицы). В таблице 3 весь столбец $x_4 = 1$ заполнен этими нулями.

В столбце $x_4 = 2$ стоят тройки везде, кроме 2 первых строчек, где стоят черточки. Проект номер 2 на предприятии 4 стоит 2 миллиона долларов ($c_4(2) = 2$), поэтому для его реализации денег должно быть не менее 2 миллионов, и варианты с $y_4 = 0$ и $y_4 = 1$ недопустимы. Тройка $R_4(2) = 3$ взята из исходной таблицы 1 – это доход от реализации 2-го проекта на 4 предприятии. Если денег на реализацию этого проекта хватает $y_4 \geq 2$, то получим доход в три миллиона долларов в год.

Последние 2 столбца таблицы 3 этапа 4 дают оптимальное решение. В каждой строчке таблицы $\{R_4(x_4)\}$ надо найти максимальное число и поместить его в столбец $f_4(y_4)$. В последний столбец x_4^* помещаем номер того столбца (1 или 2), который найденному максимуму соответствует. Например, в строке $y_4 = 0$ в таблице $\{R_4(x_4)\}$ стоит 0 и черточка. Ноль находится в столбце $x_4 = 1$, поэтому последние два числа в строке $y_4 = 0$ — ноль и единица ($f_4(0) = 0$ и $x_4^*(0) = 1$).

Этап 3. Предприятия 3 и 4.

$$f_3(y_3) = \max_{x_3=1:3|c_3(x_3)\leq y_3} \{R_3(x_3) + f_4(y_3 - c_3(x_3))\}.$$

	$R_3(x_3) + f_4(y_3 - c_3(x_3))$			Оптимальное решение	
y_3	$x_3 = 1$	$x_3 = 2$	$x_3 = 3$	$f_3(y_3)$	x_3^*
0	0+0=0	–	–	0	1
1	0+0=0	4+0=4	–	4	2
2	0+3=3	4+0=4	6+0=6	6	3
3	0+3=3	4+3=7	6+0=6	7	2
<u>4</u>	0+3=3	4+3=7	6+3=9	<u>9</u>	<u>3</u>
5	0+3=3	4+3=7	6+3=9	9	3

Таблица 4. Задача об инвестициях, этап 3

2.1.2 Заполнение таблицы этапа 3 и последующих.

Заполнение таблицы 4 опишем подробно. В первом столбце перечисляются все возможные состояния $y_3 \in 0, \ldots, 5$. Последние столбцы 5 и 6 содержат оптимальное решение $f_3(y_3)$ и x_3^*. Содержательно оптимальное решение говорит об максимальном доходе, который можно получить при оптимальном расходе y_3 миллионов долларов, выделенных для предприятий 3 и 4. Оптимальное управление x_3^* – номер выбранного проекта для реализации на 3 предприятии, который позволит получить этот максимальный доход.

Три столбца, соответствующие управлениям $x_3 = 1, 2$ и 3, заполняются довольно просто. Надо для каждой допустимой пары аргументов (y_3, x_3) вычислить значение функции $\{R_3(x_3) + f_4(y_3 - c_3(x_3))\}$. Формула для вычисления этой функции записана сверху в таблице 4 без фигурных скобок.

Первое слагаемое формулы $\{R_3(x_3) + \ldots\}$ в столбце $x_3 = 1$ равно нулю ($R_3(1) = 0$), в столбце $x_3 = 2$ первое слагаемое равно 4 ($R_3(2) = 4$), и в столбце ($x_3 = 3$) $R_3(3) = 6$. Числа $R_3(x_3)$ берутся из исходной таблицы 1. Это столбец R_3 для предприятия 3 – доходы, которые будут получены на этом предприятии от реализации проектов 1, 2 или 3.

Обратим внимание, что столбцы $x_3 = 2$ и $x_3 = 3$ нашей таблицы начинаются с черточек, такие сочетания (y_3, x_3) невозможны. Объяснения этому дадим чуть позже.

Второе слагаемое формулы $\{\ldots + f_4(y_3 - c_3(x_3))\}$ берется из предыдущей таблицы этапа 4. Вспомним, что $y_3 - c_3(x_3) = y_4$. Если из денег y_3, выделенных на предприятия 3 и 4 вычесть $c_3(x_3)$ миллионов долларов – стоимость проекта x_3 для предприятия 3, то в результате останется y_4 миллионов долларов для предприятия 4. Столбец $x_3 = 1$ соответствует пустому проекту, поэтому $y_3 = y_4$ и в качестве второго слагаемого переписывается оптимальный доход $f_4(y_4)$ из предыдущей таблицы.

Заполняем второй столбец $x_3 = 2$ таблицы 4. Второй столбец соответствует проекту с характеристиками $(c_3, R_3) = (1, 4)$. Проект с номером 2 стоит $c_3(2) = 1$ миллион долларов, поэтому из $y_3 - c_3(x_3) \geq 0$ следует $y_3 \geq 1$. Из этих соображений столбец $x_3 = 2$ начинается с черточки. Если у нас нет миллиона, то проект стоимостью 1 миллион реализовать нельзя. Ниже черточки в качестве второго слагаемого переписывается начало столбца $f_4(y_4)$ из предыдущей таблицы.

Столбец $x_3 = 3$ вычисляется для проекта с характеристиками $(c_3, R_3) = (2, 6)$. Проект стоит 2 миллиона долларов и дает доход 6 миллионов в год. По этой причине 2 первых элемента столбца – черточки. Для допустимых вариантов при $y_3 \geq 2$ первое слагаемое в этом столбце равно 6. Второе слагаемое – начало столбца $f_4(y_4)$ из таблицы 3 этапа 4.

Оптимальное решение в последних 2 столбцах считается традиционно. В каждой строчке y_3 в столбцах $x_3 = 1, 2$ и 3 находим максимум и записываем его в $f_3(y_3)$. В x_3^* записываем номер того столбца, где этот максимум находится. Например, для последней строки таблицы $y_3 = 5 \quad \max\{3, 7, 9\} = 9$. Это число записывается в качестве $f_3(5)$ в предпоследнем столбце таблицы, а в столбце $x_3^*(5)$ записываем 3, так как 9 находится в столбце $x_3 = 3$.

Этап 2. Предприятия 2, 3 и 4.

$$f_2(y_2) = \max_{x_2=1:3|c_2(x_2)\leq y_2} \{R_2(x_2) + f_3(y_2 - c_2(x_2))\}.$$

	$R_2(x_2) + f_3(y_2 - c_2(x_2))$			Оптимальное решение	
y_2	$x_2 = 1$	$x_2 = 2$	$x_2 = 3$	$f_2(y_2)$	x_2^*
0	0+0=0	–	–	0	1
1	0+4=4	–	–	4	1
2	0+6=6	–	–	6	1
3	0+7=7	5+0=5	–	7	1
<u>4</u>	0+9=9	5+4=9	–	<u>9</u>	<u>1</u>, 2
5	0+9=9	5+6=11	0+9=9	11	2

Таблица 5. Задача об инвестициях, этап 2

Этап 1. Предприятия 1, 2, 3 и 4.

$$f_1(y_1) = \max_{x_1=1:2|c_1(x_1)\leq y_1} \{R_1(x_1) + f_2(y_1 - c_1(x_1))\}.$$

	$R_1(x_1) + f_2(y_1 - c_1(x_1))$		Оптимальное решение	
y_1	$x_1 = 1$	$x_1 = 2$	$f_1(y_1)$	x_1^*
0	0+0=0	–	0	1
1	0+4=4	3+0=3	4	1
2	0+6=6	3+4=7	7	2
3	0+7=7	3+6=9	9	2
4	0+9=9	3+7=10	10	2
<u>5</u>	0+11=11	3+9=12	<u>12</u>	<u>2</u>

Таблица 6. Задача об инвестициях, этап 1

Остальные таблицы для этапов $j = 2$ и $j = 1$ заполняются аналогично. Записывается уравнение Беллмана для конкретного j, заполняется таблица для всех допустимых состояний y_j и управлений x_j. После этого ищется максимум по строкам таблицы и находится оптимальное решение, которое располагается в 2 последних столбцах любой таблицы.

2.1.3 Получение оптимального решения в задаче распределения инвестиций.

Когда все 4 таблицы заполнены и получены для каждого j оптимальное решение, надо записать оптимальное решение всей задачи.

Из последней таблицы 6 этапа 1 видно, что получаемый максимальный доход составляет $f_1(5) = 12$ миллионов долларов в год. Для этого нужно на 1 предприятии применить 2 проект $x_1^* = 2$. Характеристики второго проекта $(c_1(2), R_1(2)) = (1, 3)$. Эти числа в столбцах c_1 и R_1 подчеркнуты в исходной таблице 1. На предприятия 2, 3 и 4 остается денег $y_2 = y_1 - c_1(2) = 5 - 1 = 4$ миллиона долларов.

Переходим к таблице этапа 2 и подчеркиваем в строке $y_2 = 4$ числа $f_2(4) = 9$ и $x_2^*(4) = 1$. Из-за того, что в качестве $x_2^*(4)$ стоит не только 1 (только что подчеркнутая), но еще и число 2, можем утверждать, что оптимальных решений в нашей задаче больше одного. $x_2^* = 1$ соответствует проекту с характеристиками $(c_2(1), R_2(1)) = (0, 0)$. Эти числа подчеркиваются в исходной таблице 1. $y_3 = y_2 - c_2(1) = 4 - 0 = 4$, поэтому на предприятия 3 и 4 переходят все 4 миллиона долларов.

Из таблицы этапа 3 выясняем, как эти $y_3 = 4$ миллиона долларов могут быть оптимально использованы. Подчеркиваем $f_3(4) = 9$ и $x_3^*(4) = 3$. Третий проект третьего предприятия $x_3^* = 3$ имеет характеристики $(c_3, R_3) = (2, 6)$. Подчеркиваем эти числа в исходной таблице 1. $y_4 = y_3 - c_3(3) = 4 - 2 = 2$.

Как потратить оптимально $y_4 = 2$ миллиона долларов видно из таблицы этапа 4. Для этого надо применить $x_1^* = 2$ проект на 4 предприятии. Подчеркиваем в исходной таблице $c_4(2) = 2$ и $R_4(2) = 3$ и записываем окончательный ответ.

Для того, чтобы оптимальным образом потратить 5 миллионов долларов на 4 предприятиях, надо выбирать проекты

$$x^* = (2, 1, 3, 2).$$

При этом будет получен максимальный ежегодный доход $f_1(5) = 12$ миллионов долларов в год.

Полезно проверить наши вычисления. Общая стоимость всех проектов $\sum_j c_j(x_j^*) = 1 + 0 + 2 + 2 = 5$ миллионов долларов. Ежегодный доход действительно равняется 12: $f_1(5) = \sum_j R_j(x_j^*) = 3+0+6+3 = 12$ миллионов долларов в год.

В качестве упражнения попробуйте получить все оптимальные решения нашей задачи. Подсказка. Всего в задаче 2 оптимальных решения. Второе решение $x_{(2)}^* = (2, 2, 2, 1)$.

2.2 Графическое решение задачи распределения инвестиций.

Графическое решение задачи полностью повторяет решение, полученное с помощью таблиц. Каждая вершина графа соответствует некоторому известному состоянию системы, каждой дуге поставлен в соответствие определенный проект. Считаем, что дуги имеют направление справа налево. Сначала над вершиной $y_5 = 0$ ставим 0 и $*$. Выполняем этап 4. Для каждого состояния $y_4 \in 0 : 5$ вычисляем пару чисел $f_4(y_4)$ и $x_4^*(y_4)$. Эти два числа – максимальный доход, получаемый на 4 предприятии и номер проекта, который следует для этого применять, записываем над вершиной y_4. Понятно, что над вершинами $y_j \in 0 : 5$ стоят те же самые числа, которые стояли в двух правых столбцах таблицы 3, расположенной на странице 5.

3 Задача распределения инвестиций без пустых проектов.

Существенной особенностью предыдущей задачи является наличие в таблице 1 первой строки, состоящей из одних нулей. Предполагается, что можно модернизацию любого предприятия не проводить вообще. В реальных задачах это не так. Раз уж предприятие мы собираемся модернизировать, то хотя бы самый дешевый способ модернизации мы должны запланировать. Следующая задача как раз такого типа.

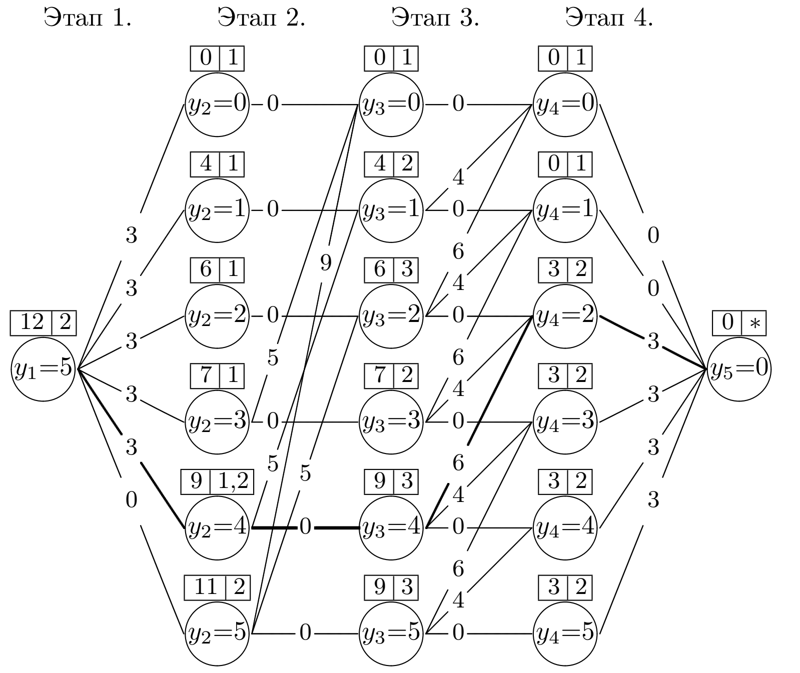

Рис. 2. Графическое решение задачи о распределении инвестиций.

Проект	Пр-тие 1		Пр-тие 2		Пр-тие 3		Пр-тие 4	
	c_1	R_1	c_2	R_2	c_3	R_3	c_4	R_4
$x_j = 1$	$\underline{1}$	$\underline{1}$	$\underline{2}$	$\underline{2}$	1	1	$\underline{1}$	$\underline{3}$
$x_j = 2$	3	4	4	6	2	3	3	5
$x_j = 3$	—	—	—	—	$\underline{4}$	$\underline{7}$	—	—

Таблица 7. Инвестиции 8 миллионов долларов без пустых проектов

Рекуррентное уравнение Беллмана для процедуры обратной прогонки то же самое

$$f_5(y_5) = 0,$$
$$f_j(y_j) = \max_{x_j | c_j(x_j) \leq y_j} \{R_j(x_j) + f_{j+1}(y_j - c_j(x_j))\},\ j = 4, 3, 2, 1.$$

Сначала надо решить задачу для четвертого предприятия, затем для третьего и четвертого предприятия, затем для 2, 3 и 4 предприятия и, наконец, для предприятий 1, 2 , 3 и 4. В таблицах $8 - 11$ приводятся результаты расчетов.

Единственная разница с предыдущим примером – более узкие рамки для изменения состояний. Для предприятия 4 самый дешевый проект стоит 1 миллион долларов, поэтому $y_4 \geq 1$. Для предприятий 1, 2 и 3 также надо предусмотреть хотя бы самые дешевые проекты, поэтому верхняя граница y_4 не 8 миллионов, а на $c_1(1) + c_2(1) + c_3(1) = 1 + 2 + 1 = 4$ миллиона долларов меньше. Таким образом, на 4 предприятии можно тратить деньги в диапазоне $y_4 \in 1 : 4$. В остальном ничего нового в построении таблицы этапа 4 нет.

Таблицу этапа 3 строим для состояний $y_3 \in 2 : 5$. Действительно, для предприятий 3 и 4 надо предусмотреть хотя бы самые дешевые проекты $c_3(1) + c_4(1) = 2$. Для предприятий 1 и 2 самые дешевые проекты стоят $c_1(1) + c_2(1) = 1 + 2 = 3$ миллиона долларов, поэтому $y_3 \leq (8 - 3)$.

Таблицу этапа 2 строим для состояний $y_2 \in 4 : 7$. Для предприятий 2, 3 и 4 самые дешевые проекты стоят $c_2(1) + c_3(1) + c_4(1) = 4$. Для предприятия 1 проект 1 стоит $c_1(1) = 1$ миллион долларов, поэтому $y_2 \leq 8 - 1$.

В последней таблице можно вообще находить только одну строч-

Этап 4. Предприятие 4.

$$f_4(y_4) = \max_{x_4=1:2|c_4(x_4)\le y_4} \{R_4(x_4)\}.$$

y_4	$R_4(x_4)$		Оптимальное решение	
	$x_4 = 1$	$x_4 = 2$	$f_4(y_4)$	x_4^*
$\underline{1}$	3	—	$\underline{3}$	$\underline{1}$
2	3	—	3	1
3	3	5	5	2
4	3	5	5	2

Таблица 8. Инвестиции без пустых проектов, этап 4

Этап 3. Предприятия 3 и 4.

$$f_3(y_3) = \max_{x_3=1:3|c_3(x_3)\le y_3} \{R_3(x_3) + f_4(y_3 - c_3(x_3))\}.$$

y_3	$R_3(x_3) + f_4(y_3 - c_3(x_3))$			Оптимальное решение	
	$x_3 = 1$	$x_3 = 2$	$x_3 = 3$	$f_3(y_3)$	x_3^*
2	1+3=4	—	—	4	1
3	1+3=4	3+3=6	—	6	2
4	1+5=6	3+3=6	—	6	1,2
$\underline{5}$	1+5=6	3+5=8	7+3=10	10	$\underline{3}$

Таблица 9. Инвестиции без пустых проектов, этап 3

Этап 2. Предприятия 2, 3 и 4.

$$f_2(y_2) = \max_{x_2=1:3|c_2(x_2)\le y_2} \{R_2(x_2) + f_3(y_2 - c_2(x_2))\}.$$

y_2	$R_2(x_2) + f_3(y_2 - c_2(x_2))$		Оптимальное решение	
	$x_2 = 1$	$x_2 = 2$	$f_2(y_2)$	x_2^*
4	2+ 4=6	—	6	1
5	2+ 6=8	—	8	1
6	2+ 6=8	6+4=10	10	2
$\underline{7}$	2+10=12	6+6=12	12	$\underline{1}$, 2

Таблица 10. Инвестиции без пустых проектов, этап 2

Этап 1. Предприятия 1, 2, 3 и 4.

$$f_1(y_1) = \max_{x_1=1:2|c_1(x_1)\le y_1} \{R_1(x_1) + f_2(y_1 - c_1(x_1))\}.$$

y_1	$R_1(x_1) + f_2(y_1 - c_1(x_1))$		Оптимальное решение	
	$x_1 = 1$	$x_1 = 2$	$f_1(y_1)$	x_1^*
5	1+ 6= 7	—	7	1
6	1+ 8= 9	—	9	1
7	1+10=11	4+6=10	11	1
$\underline{8}$	1+12=13	4+8=12	$\underline{13}$	$\underline{1}$

Таблица 11. Инвестиции без пустых проектов, этап 1

ку, соответствующую состоянию $y_1 = 8$. Но можно для проведения дальнейшего анализа найти еще несколько строчек. Самая верхняя соответствует затратам, необходимым для реализации варианта $x = (1, 1, 1, 1)$. Для этого необходимо затратить $\sum_j c_j(1) = 1{+}2{+}1{+}1$ $= 5$ миллионов долларов.

Итак, запишем один из ответов нашей задачи. Для того, чтобы оптимальным образом потратить 8 миллионов долларов на 4 предприятиях, надо выбирать проекты

$$x^* = (1, 1, 3, 1).$$

$$f_1(8) = 13$$

миллионов долларов в год при этом составит максимальный годовой доход.

Задание. Найти все оптимальные решения задачи.

3.1 Задача распределения инвестиций на компьютере.

Решаем задачу динамического программирования с помощью пакета MATLAB. Программы написаны таким образом, чтобы их можно было запускать на любой версии MATLAB, начиная с «досовской» версии 3.5g. Эта версия помещается на одну дискетку 1.44 mb, да еще и остается место для любимого бесплатного текстового редактора Aditor ver.2.10b. Последняя версия MATLAB, на которой проверялись все программы этого пособия — версия 7.01.

Единственное исключение — MATLAB ver.6.0. Эта версия не понимает строчную букву «я» даже в комментариях. Приходится заменять ее прописной буквой «Я». Это ошибка данной версии, в последующем она была исправлена, поэтому некоторые программы настроены на версию 6, но большинство моих программ в ней работать откажутся.

Наша первая программа называется InvestD.m. Чтобы напечатать текст программы на экране, надо набрать type InvestD, чтобы запустить на выполнение, набирается текст InvestD. Из протокола сессии работы с пакетом MATLAB хорошо видно, как это все выполняется. Если неясен результат работы какого-либо оператора, надо набрать слово help и добавить имя этого оператора. MATLAB просветит Вас на английском языке, что это за оператор.

Следующий текст был получен с помощью «дневника» MATLAB, который записал весь текстовый вывод на экран в файл InvestD.rtf. Для этого набирался в ответ на приглашени MATLAB >> следующий текст:

```
>> diary InvestD.rtf
>> type InvestD
>> InvestD
>> diary off
```

Теперь изучаем, что из себя представляет текст файла InvestD.rtf:

```
type InvestD

% Программа InvestD.m  Лабораторная номер 1
% ==== 5 апреля 2006 года ==== Визгунов Н.П.

format compact; clc
y1 = 8
C = [   1    2    1    1
        3    4    2    3
      -inf -inf   4  -inf ]
R = [   1    2    1    3
        4    6    3    5
      -inf -inf   7  -inf ]

[u9,n] = size(C);   % u9 - Количество управлений
s9 = y1 + 1;        % s9 - Количество состояний

%// Допустимые состояния на этапе j меняются в пределах
%// от y_min(j) до y_max(j) включительно
```

```
y_min = zeros(1, n);
y_min(n) = C(1, n);
for j = n -1 : -1 : 1
    y_min(j) = y_min(j +1) + C(1,j);
end

y_max = zeros(1, n);
y_max(1) = y1;
for j = 2 : n
    y_max(j) = y_max(j -1) - C(1, j -1);
end

%// Теперь можно вычислять таблицы для ДП

X = -inf * ones(s9, n+1); F = X;
F(:,n+1) = zeros(s9, 1)

for j = n : -1 :1
    Fsu = -inf * ones(s9, u9);
    for s = 1 : s9          % Yj = s -1
        for u = 1 : u9      % Xj = u
            s1 = s - C(u,j);
            if 1 <= s1 & s1 <= s9
                % Уравнение Ричарда Беллмана
                Fsu(s, u) = R(u, j) + F(s1, j+1);
            end
        end
    end
    [fs, index] = max(Fsu');
    F(:,j) = fs';
    X(:,j) = index';

    disp('------------------------------------')
    disp('     y  F(:,j) X(:,j)       Fsu     ')
    disp('------------------------------------')

    %disp([(0 : y1)'  F(:,j)  X(:,j)  Fsu ])

    % печатаем только нужные строчки таблицы j,
    % за столбцами не следим:

    Temp = [(0 : y1)', fs', index', Fsu];
    disp(Temp((y_min(j) : y_max(j))+1, :));

end

% Печать результатов
% =================
x_opt = zeros(1, n);
s = s9;
for j = 1 : n
    x_opt(j) = X(s, j);
    s = s - C(x_opt(j), j);
end
f_opt = F(s9, 1), x_opt

InvestD
y1 =
     8
C =
     1     2     1     1
     3     4     2     3
  -Inf  -Inf     4  -Inf
R =
```

```
     1     2     1     3
     4     6     3     5
  -Inf  -Inf     7  -Inf
F =
  -Inf  -Inf  -Inf  -Inf     0
  -Inf  -Inf  -Inf  -Inf     0
  -Inf  -Inf  -Inf  -Inf     0
  -Inf  -Inf  -Inf  -Inf     0
  -Inf  -Inf  -Inf  -Inf     0
  -Inf  -Inf  -Inf  -Inf     0
  -Inf  -Inf  -Inf  -Inf     0
  -Inf  -Inf  -Inf  -Inf     0
  -Inf  -Inf  -Inf  -Inf     0
-----------------------------------
     y  F(:,j) X(:,j)      Fsu
-----------------------------------
     1     3     1     3  -Inf  -Inf
     2     3     1     3  -Inf  -Inf
     3     5     2     3     5  -Inf
     4     5     2     3     5  -Inf
-----------------------------------
     y  F(:,j) X(:,j)      Fsu
-----------------------------------
     2     4     1     4  -Inf  -Inf
     3     6     2     4     6  -Inf
     4     6     1     6     6  -Inf
     5    10     3     6     8    10
-----------------------------------
     y  F(:,j) X(:,j)      Fsu
-----------------------------------
     4     6     1     6  -Inf  -Inf
     5     8     1     8  -Inf  -Inf
     6    10     2     8    10  -Inf
     7    12     1    12    12  -Inf
-----------------------------------
     y  F(:,j) X(:,j)      Fsu
-----------------------------------
     5     7     1     7  -Inf  -Inf
     6     9     1     9  -Inf  -Inf
     7    11     1    11    10  -Inf
     8    13     1    13    12  -Inf
f_opt =
    13
x_opt =
     1     1     3     1
diary off
```

Как мы видим, решение, которые выдает наша программа, почти полностью совпадает с результатом, который получен вручную. Хотя результат и получен в «старинном стиле», в командном окне, но таблицы компьютер вычисляет верно и делает это достаточно простая программа. Самое заметное отличие от табличек, которые мы только что рисовали — два последних столбца с оптимальными решениями сдвинуты на второе и третье место. Такие таблицы оказались удобнее для решения задач на доске. Кроме того, алгоритм получения оптимального решения x^* становится более понятным.

Единственный существенный недостаток нашей программы — она дает один ответ. Чтобы получить все ответы, предложенную программу динамического программирования пришлось бы сильно усложнить.

Для получения всех оптимальных решений, нам проще написать другую программу, а именно, программу полного перебора. Она очень хорошо справляется с небольшими задачами и проверяет наше умение решать задачи методом динамического программирования. К тому же, теперь мы всегда будем знать все оптимальные решения.

```
type InvestP

% Программа InvestP.m  Лабораторная номер 1
% ==== 5 апреля 2006 года ==== Визгунов Н.П.

format compact; clc
y1 = 8
C = [   1    2    1    1
        3    4    2    3
      -inf -inf   4  -inf ]
R = [   1    2    1    3
        4    6    3    5
      -inf -inf   7  -inf ]

[u9,n] = size(C);       % u9 - число возможных управлений

x_min = ones(1,n)       % [ 1  1  1  1 ]
x_max = sum(finite(R))  % [ 2  3  3  2 ]

x     = x_min;   x_opt = x_min;
f_opt = -inf;

j = n;
while 1
    if  x(j) <= x_max(j)
        cost = 0;
        for i = 1 :n
            cost = cost + C(x(i),i);
        end
        if cost <= y1
            f = 0;
            for i = 1:n
               f = f + R(x(i),i);
            end
            if  f_opt < f
                % Для подавления печати промежуточных данных
                % поставить в конце оператора ;
                f_opt = f
                x_opt = x
            elseif f_opt == f
                f_opt
                x_opt = [x_opt; x ]
            end
        end % Cost
        j = n;
    else    % x(j) > x_max(j)
        x(j) = x_min(j);
        j = j - 1;
        if j <= 1e-5
            % Решение получено - выйти из цикла
            break
        end
    end % if
    x(j) = x(j) + 1;
end % while 1
disp('     ******')
disp('     Ответ:')
disp('     ******')
f_opt, x_opt

InvestP
```

```
y1 =
     8
C =
     1     2     1     1
     3     4     2     3
  -Inf  -Inf     4  -Inf
R =
     1     2     1     3
     4     6     3     5
  -Inf  -Inf     7  -Inf
x_min =
     1     1     1     1
x_max =
     2     2     3     2
f_opt =
     7
x_opt =
     1     1     1     1
f_opt =
     9
x_opt =
     1     1     1     2
f_opt =
     9
x_opt =
     1     1     1     2
     1     1     2     1
f_opt =
    11
x_opt =
     1     1     2     2
f_opt =
    13
x_opt =
     1     1     3     1
f_opt =
    13
x_opt =
     1     1     3     1
     1     2     2     1
     ******
     Ответ:
     ******
f_opt =
    13
x_opt =
     1     1     3     1
     1     2     2     1
diary off
```

Из распечатки видно, что у рассмотренной задачи о распределении инвестиций есть еще одно оптимальное решение

$$x^* = (1, 2, 2, 1).$$

Так как решение оптимальное, то доход также равен $f_1(8) = 13$ миллионов долларов в год.

4 Задача о загрузке(о рюкзаке или о ранце)

Задача 2 (о рюкзаке) *Самолет загружается предметами n различных типов. Каждый предмет типа j дает доход c_j тысяч дол-*

ларов и весит a_j тонн. Грузоподъемность самолета – b тонн.

Выбрать предметы, погрузка которых позволит получить максимальный доход без превышения грузоподъемности самолета.

Сначала рассмотрим задачу в общей постановке.

$$
\begin{aligned}
f_1(y_1) = \max_{x_1,\dots,x_n} \quad & \left\{ \sum_{j=1}^{n} c_j x_j \right\}, \\
& \sum_{j=1}^{n} a_j x_j \le y_1, \\
& x_j \ge 0 - \text{целые},\ j \in 1:n, \\
& y_1 = b.
\end{aligned}
$$

Легко вывести рекуррентные уравнения Беллмана для процедуры обратной прогонки.

$$
\begin{aligned}
f_n(y_n) &= \max_{x_n=0,1,\dots,[b/a_n]} \quad \{c_n x_n\}, \\
f_j(y_j) &= \max_{x_j=0,1,\dots,[b/a_j]} \quad \{c_j x_j + f_{j+1}(y_j - a_j x_j)\}, \\
& \qquad j = n-1, \dots, 2, 1
\end{aligned}
$$

Здесь квадратные скобки в выражении $[b/a_j]$ используются для обозначения целой части числа b/a_j, стоящего в этих скобках.

Другие обозначения:

c_j – доход, получаемый от перевозки одного предмета типа j,

a_j – вес этого предмета,

x_j – количество предметов j-го типа (управление),

y_j – часть грузоподъемности самолета, выделенная для предметов $j, j+1, \dots, n$ (состояние),

$f_j(y_j)$ – максимальный доход от погрузки предметов $j, j+1, \dots, n$, если в самолете выделено y_j тонн под эти предметы.

Решим следующую конкретную задачу с помощью этого рекуррентного уравнения.

$$
\begin{aligned}
f_1(y_1) = \max_{x_1,x_2,x_3} \quad & \{65x_1 + 80x_2 + 30x_3\}, \\
& 2x_1 + 3x_2 + 1x_3 \le y_1, \\
& x_j - \text{целые},\ y_1 = 5.
\end{aligned}
$$

В таблицах 12 – 14 выполнены все вычисления, необходимые для получения оптимального решения.

Этап 3. Предметы 3 типа.

$$f_3(y_3) = \max_{x_3}\{30x_3\},\ \max x_3 = [5/1] = 5$$

	$30x_3$						Оптимальное решение	
y_3	x_3=0	x_3=1	x_3=2	x_3=3	x_3=4	x_3=5	$f_3(y_3)$	x_3^*
0	0	–	–	–	–	–	0	0
<u>1</u>	0	30	–	–	–	–	<u>30</u>	<u>1</u>
2	0	30	60	–	–	–	60	2
3	0	30	60	90	–	–	90	3
4	0	30	60	90	120	–	120	4
5	0	30	60	90	120	150	150	5

Таблица 12. Задача о рюкзаке, этап 3

При заданном $y_1 = 5$ оптимальным решением является $x^* = (2, 0, 1)$, а суммарная стоимость груза равна 160.

Заметим, что на этапе 1 достаточно построить только одну строку таблицы для значения $y_1 = 5$. Однако, располагая полной таблицей, можно провести анализ чувствительности решения, то есть посмотреть уменьшение целевой функции при уменьшении грузоподъемности самолета. Вычислительная схема динамического программирования автоматически обеспечивает проведение анализа модели на чувствительность.

Этап 2. Предметы 2 и 3 типа.

$$f_2(y_2) = \max_{x_2}\{80x_2 + f_3(y_2 - 3x_2)\}, \max x_2 = [5/3] = 1$$

	$80x_2 + f_3(y_2 - 3x_2)$		Оптимальное решение	
y_2	$x_2 = 0$	$x_2 = 1$	$f_2(y_2)$	x_2^*
0	0+ 0= 0	–	0	0
<u>1</u>	0+ 30=30	–	<u>30</u>	<u>0</u>
2	0+ 60=60	–	60	0
3	0+ 90=90	80+0 = 80	90	0
4	0+120=120	80+30=110	120	0
5	0+150=150	80+60=140	150	0

Таблица 13. Задача о рюкзаке, этап 2

Этап 1. Предметы 1, 2 и 3 типа.

$$f_1(y_1) = \max_{x_1}\{65x_1 + f_2(y_1 - 2x_1)\}, \max x_1 = [5/2] = 2$$

	$65x_1 + f_2(y_1 - 2x_1)$			Оптимальное решение	
y_1	$x_1 = 0$	$x_1 = 1$	$x_1 = 2$	$f_1(y_1)$	x_1^*
0	0+ 0= 0	–	–	0	0
1	0+ 30= 30	–	–	30	0
2	0+ 60= 60	65+ 0= 65	–	65	1
3	0+ 90= 90	65+30= 95	–	95	1
4	0+120=120	65+60=125	130+ 0=130	130	2
<u>5</u>	0+150=150	65+90=155	130+30=160	<u>160</u>	<u>2</u>

Таблица 14. Задача о рюкзаке, этап 1

Легко решить задачу о рюкзаке и графически. Ничего нового в здесь нет, поэтому предлагается сделать это самостоятельно.

4.1 Задача о рюкзаке на компьютере

Задача о рюкзаке является, с точки зрения математиков, частным случаем задачи об инвестициях. Поэтому самое простое — из векторов a и c надо вычислить матрицы результатов C и R, что и делается в следующей программе.

```
type RjukCR

% function RjukCR(c, a, b )
% программа RjukCR.m  Лабор. 2.  Вычисляем C и R !!!
% =====5 апреля 2006 года=====Визгунов Н.П.=========

format compact, clc

c = [ 65  80  30 ] % доход от перевозки одного предмета
a = [  2   3   1 ] % вес этого предмета
b = 5, y1 = b;     % грузоподъемность самолета

n  = length(a);
s9 = y1 + 1;                 % s9 - Кол-во состояний
u9 = 1 + floor(b / min(a)); % u9 - Кол-во управлений

% Очевидный способ вычисления C и R

C = -inf * ones(u9, n);
R = C;
for u = 1 : u9      % (u-1) -это количество предметов
    for j = 1 : n
```

```
        temp = a(j) * (u -1);   % j  -номер предмета
        if temp <= b   % если предметы помещаются в самолет
            C(u, j) = temp;
            R(u, j) = c(j) * (u -1);
        end
    end
end
C;
R;  % для печати убрать ;

% Вычисления C и R в духе MatLab

C = (0 : u9 -1)' * a;
indexC = find(C > y1);
C(indexC) = -inf * ones(1, length(indexC))
R = (0 : u9 -1)' * c;
R(indexC) = -inf * ones(1, length(indexC))

% Теперь почти задача о распределении инвестиций

X = -inf * ones(s9, n+1);
F = X;
F(:, n +1) = zeros(s9, 1);

for j = n : -1 : 1
    Fsu = -inf * ones(s9, u9);
    for s = 1 : s9          % Yj = s -1
        for u = 1 : u9      % Xj = u
            s1 = s - C(u,j);
            if 1 <= s1 & s1 <= s9
                Fsu(s, u) = R(u, j) + F(s1, j +1);
            end
        end
    end
    [fs, ind] = max(Fsu');
    F(:,j) = fs';
    X(:,j) = ind';
    line50 = '-';  line50 = line50(ones(1,54));

    disp(line50)
    disp(['     y  F(:,j) X(:,j)',  blanks(15),  'Fsu'])
    disp(line50)

    disp([(0 : u9-1)',  F(:,j),  X(:,j)-1,  Fsu])
end

% печать результатов
% =================
x_opt = zeros(1, n);
s = s9;
for j = 1 : n
    x_opt(j) = X(s, j);
    s = s - C(x_opt(j), j);
end

% здесь придется вычесть 1, чтобы найти
% количество предметов, а не номер столбца
x_opt = x_opt - 1

f_opt = F(s9, 1)

RjukCR
c =
    65    80    30
```

```
a =
     2     3     1
b =
     5
C =
     0     0     0
     2     3     1
     4  -Inf     2
  -Inf  -Inf     3
  -Inf  -Inf     4
  -Inf  -Inf     5
R =
     0     0     0
    65    80    30
   130  -Inf    60
  -Inf  -Inf    90
  -Inf  -Inf   120
  -Inf  -Inf   150
-------------------------------------------------------
     y  F(:,j) X(:,j)               Fsu
-------------------------------------------------------
     0     0     0     0  -Inf  -Inf  -Inf  -Inf  -Inf
     1    30     1     0    30  -Inf  -Inf  -Inf  -Inf
     2    60     2     0    30    60  -Inf  -Inf  -Inf
     3    90     3     0    30    60    90  -Inf  -Inf
     4   120     4     0    30    60    90   120  -Inf
     5   150     5     0    30    60    90   120   150
-------------------------------------------------------
     y  F(:,j) X(:,j)               Fsu
-------------------------------------------------------
     0     0     0     0  -Inf  -Inf  -Inf  -Inf  -Inf
     1    30     0    30  -Inf  -Inf  -Inf  -Inf  -Inf
     2    60     0    60  -Inf  -Inf  -Inf  -Inf  -Inf
     3    90     0    90    80  -Inf  -Inf  -Inf  -Inf
     4   120     0   120   110  -Inf  -Inf  -Inf  -Inf
     5   150     0   150   140  -Inf  -Inf  -Inf  -Inf
-------------------------------------------------------
     y  F(:,j) X(:,j)               Fsu
-------------------------------------------------------
     0     0     0     0  -Inf  -Inf  -Inf  -Inf  -Inf
     1    30     0    30  -Inf  -Inf  -Inf  -Inf  -Inf
     2    65     1    60    65  -Inf  -Inf  -Inf  -Inf
     3    95     1    90    95  -Inf  -Inf  -Inf  -Inf
     4   130     2   120   125   130  -Inf  -Inf  -Inf
     5   160     2   150   155   160  -Inf  -Inf  -Inf
x_opt =
     2     0     1
f_opt =
   160
diary off
```

Но можно совершить дополнительные усилия и написать более компактную программу, которая решает задачу о рюкзаке быстрее и позволяет решать более объемные задачи.

```
type Rjuk

% function Rjuk(c, a, b )
% программа Rjuk.m  Без вычислений C и R !!!
% =====5 апреля 2006 года=====Визгунов Н.П.

format compact, clc

c = [ 65  80  30 ]  % доход от перевозки одного предмета
```

```
a = [  2   3   1 ]  % вес этого предмета
b = 5, y1 = b;      % грузоподъемность самолета

n  = length(a);
s9 = y1 + 1;                % s9 - Кол-во состояний
u9 = 1 + floor(b / min(a)); % u9 - Кол-во управлений

X = -inf * ones(s9, n+1);
F = X;
F(:, n +1) = zeros(s9, 1);

for j = n : -1 : 1
    Fsu = -inf * ones(s9, u9);
    for s = 1 : s9                  % Yj = s -1
        for u = 1 : u9              % Xj = u
            s1 = s - a(j)*(u -1);   % - C(u,j) было
            if 1 <= s1 & s1 <= s9
                Fsu(s, u) = c(j)*(u-1) + F(s1, j +1);
            end
        end
    end
    [fs, ind] = max(Fsu');
    F(:,j) = fs';
    X(:,j) = ind' -1;
    line50 = '-';  line50 = line50(ones(1, 54));

    disp(line50)    % напечатать 50 минусов
    disp(['     y  F(:,j) X(:,j)',  blanks(15),  'Fsu'])
    disp(line50)

    disp([(0 : u9-1)',  F(:,j),  X(:,j),  Fsu])
end

% печать результатов
% ==================
x_opt = zeros(1, n);
s = s9;
for j = 1 : n
    x_opt(j) = X(s, j);
    s = s - a(j) * x_opt(j);
end
x_opt
f_opt = F(s9, 1)

Rjuk
c =
    65    80    30
a =
     2     3     1
b =
     5
------------------------------------------------------
     y  F(:,j) X(:,j)               Fsu
------------------------------------------------------
     0     0     0     0  -Inf  -Inf  -Inf  -Inf  -Inf
     1    30     1     0    30  -Inf  -Inf  -Inf  -Inf
     2    60     2     0    30    60  -Inf  -Inf  -Inf
     3    90     3     0    30    60    90  -Inf  -Inf
     4   120     4     0    30    60    90   120  -Inf
     5   150     5     0    30    60    90   120   150
------------------------------------------------------
     y  F(:,j) X(:,j)               Fsu
------------------------------------------------------
     0     0     0     0  -Inf  -Inf  -Inf  -Inf  -Inf
```

```
     1    30     0    30  -Inf  -Inf  -Inf  -Inf  -Inf
     2    60     0    60  -Inf  -Inf  -Inf  -Inf  -Inf
     3    90     0    90    80  -Inf  -Inf  -Inf  -Inf
     4   120     0   120   110  -Inf  -Inf  -Inf  -Inf
     5   150     0   150   140  -Inf  -Inf  -Inf  -Inf
------------------------------------------------------
     y  F(:,j) X(:,j)               Fsu
------------------------------------------------------
     0     0     0     0  -Inf  -Inf  -Inf  -Inf  -Inf
     1    30     0    30  -Inf  -Inf  -Inf  -Inf  -Inf
     2    65     1    60    65  -Inf  -Inf  -Inf  -Inf
     3    95     1    90    95  -Inf  -Inf  -Inf  -Inf
     4   130     2   120   125   130  -Inf  -Inf  -Inf
     5   160     2   150   155   160  -Inf  -Inf  -Inf
x_opt =
     2     0     1
f_opt =
   160
diary off
```

Для получения всех решений полезно решить нашу тестовую задачу полным перебором. Правда, решений в этой задаче немного, всего одно. Поэтому для демонстрации программы полного перебора полезно взять другой пример, который дает много решений. Этих решений будет целых пять!

```
type RjukP

% function RjukP(c, a, b)
% программа RjukP.m  Лабораторная номер 2
% =====5 апреля 2006 года=====Визгунов Н.П.

format compact, clc

% c = [ 65  80  30 ]
% a = [  2   3   1 ]

b = 5,  y1 = b;

% Здесь будет целых 5 ответов !!
c = [45    30    15]
a = [ 3     2     1]

n  = length(a);
s9 = y1 + 1;                  % s9 - Кол-во состояний
u9 = 1 + floor(b / min(a));
% u9 = 1 + floor(max(b ./ a));

x_min = zeros(1, n);   %  [0  0  0 ]
x = x_min;
x_max = floor(b ./ a); %  [2  1  5 ]

f_opt = -inf;
j = n;
while 1
    if  x(j) <= x_max(j)
        if a * x' <= b  % если предметы помещаются в самолет
            f = c * x';
            if f_opt < f
                disp('f_opt < f   <<<<<<<<<<<<<<<<<<<<')
                f_opt = f
                x_opt = x
                disp(' ')
            elseif  f == f_opt  % elseif - одно слово !
```

```
                disp('f_opt == f ====================')
                f_opt
                x_opt = [x_opt; x]
                disp(' ')
            end
        end
        j = n;
    else    % x(j) > x_max(j)
        x(j) = x_min(j);
        j = j - 1;
        % закончить бесконечный цикл
        if j <= 1e-5 , break,  end
    end % if
    x(j) = x(j) + 1;
end % while 1

disp('ответ:')
disp('======')
x_opt, f_opt
```

```
RjukP
b =
     5
c =
    45    30    15
a =
     3     2     1
f_opt < f   <<<<<<<<<<<<<<<<<<<
f_opt =
     0
x_opt =
     0     0     0

f_opt < f   <<<<<<<<<<<<<<<<<<<
f_opt =
    15
x_opt =
     0     0     1

f_opt < f   <<<<<<<<<<<<<<<<<<<
f_opt =
    30
x_opt =
     0     0     2

f_opt < f   <<<<<<<<<<<<<<<<<<<
f_opt =
    45
x_opt =
     0     0     3

f_opt < f   <<<<<<<<<<<<<<<<<<<
f_opt =
    60
x_opt =
     0     0     4

f_opt < f   <<<<<<<<<<<<<<<<<<<
f_opt =
    75
x_opt =
     0     0     5

f_opt == f ===================
```

```
f_opt =
    75
x_opt =
     0     0     5
     0     1     3

f_opt == f ===================
f_opt =
    75
x_opt =
     0     0     5
     0     1     3
     0     2     1

f_opt == f ===================
f_opt =
    75
x_opt =
     0     0     5
     0     1     3
     0     2     1
     1     0     2

f_opt == f ===================
f_opt =
    75
x_opt =
     0     0     5
     0     1     3
     0     2     1
     1     0     2
     1     1     0

ответ:
======
x_opt =
     0     0     5
     0     1     3
     0     2     1
     1     0     2
     1     1     0
f_opt =
    75
diary off
```

5 Задача о надежности

Задача 3 (о надежности) *Конструируется электронный прибор, состоящий из трех основных компонентов. Все компоненты соединены последовательно, поэтому выход из строя одной из них приводит к отказу всего прибора. Надежность прибора можно повысить путем дублирования каждой компоненты. Каждая компонента может состоять из одного, двух или трех блоков. Общая стоимость прибора не должна превышать 10 тысяч долларов. В таблице 15 приведены данные о стоимости* $c_j(x_j)$ *и надежности* $R_j(x_j)$ j*-ой компоненты, состоящей из* x_j *блоков.*

Требуется определить количество блоков x_j *в компоненте* j*, при котором надежность прибора максимальна, а стоимость не превышает заданной величины.*

Сначала рассмотрим задачу в общей постановке.

x_j	$j=1$		$j=2$		$j=3$	
	C_1	R_1	C_2	R_2	C_3	R_3
1	1	0.6	$\underline{3}$	$\underline{0.7}$	2	0.5
2	$\underline{2}$	$\underline{0.8}$	5	0.8	4	0.7
3	3	0.9	6	0.9	$\underline{5}$	$\underline{0.9}$

Таблица 15. Данные о стоимости и надежности каждой компоненты прибора.

$$
\begin{aligned}
f_1(y_1) = \max_{x_1,\dots,x_n} \quad & \left\{ \prod_{j=1}^{n} R_j(x_j) \right\}, \\
& \sum_{j=1}^{n} c_j(x_j) \le y_1, \\
& x_j \ge 1 \text{ – целые}, \ j = 1:n.
\end{aligned}
$$

Легко вывести рекуррентные уравнения Беллмана для процедуры обратной прогонки.

$$
\begin{aligned}
f_n(y_n) &= \max_{x_n | c_n(x_n) \le y_n} \{R_n(x_n)\}, \\
f_j(y_j) &= \max_{x_j | c_j(x_j) \le y_j} \{R_j(x_j) * f_{j+1}(y_j - c_j(x_j))\}, \\
& j = n-1, n-2, \dots 1.
\end{aligned}
$$

Этап 3. Третья компонента прибора.

$$f_3(y_3) = \max_{x_3} \{R_3(X_3)\}.$$

	$R_3(x_3)$			Оптимальное решение	
y_3	$x_3 = 1$	$x_3 = 2$	$x_3 = 3$	$f_3(y_3)$	x_3^*
2	0.5	–	–	0.5	1
3	0.5	–	–	0.5	1
4	0.5	0.7	–	0.7	2
$\underline{5}$	0.5	0.7	0.9	$\underline{0.9}$	$\underline{3}$
6	0.5	0.7	0.9	0.9	3

Таблица 16. Задача о надежности, этап 3

Этап 2. Вторая и третья компоненты прибора.

$$f_2(y_2) = \max_{x_2 = 1:3} \{R_2(x_2) * f_3(y_2 - c_2(x_2))\},$$

	$R_2(x_2) * f_3(y_2 - c_2(x_2))$			Оптимальное решение	
y_2	$x_2 = 1$	$x_2 = 2$	$x_2 = 3$	$f_2(y_2)$	x_2^*
5	.7*.5=.35	–	–	.35	1
6	.7*.5=.35	–	–	.35	1
7	.7*.7=.49	.8*.5=.40	–	.49	1
$\underline{8}$	.7*.9=.63	.8*.5=.40	.9*.5=.45	$\underline{.63}$	$\underline{1}$
9	.7*.9=.63	.8*.7=.56	.9*.5=.45	.63	1

Таблица 17. Задача о надежности, этап 2

Как видим на этом примере, методом динамического программирования можно решать задачи не только с аддитивной целевой функцией, но и с мультипликативной. В остальном решение задачи о надежности ничем не отличается от задачи о распределении инвестиций без пустых проектов.

5.1 Задача о надежности на компьютере

Здесь в одной распечатке представлены программа для вычисления методом динамического программирования и программа для полного перебора. Для «красоты» дробные числа в таблицах представлены в виде целых чисел. Например, максимальная надежность

Этап 1. Первая, вторая и третья компоненты прибора.

$$f_1(y_1) = \max_{x_1=1:3}\{R_1(x_1) * f_2(y_1 - c_1(x_1))\},$$

	$R_1(x_1) * f_2(y_1 - c_1(x_1))$			Оптим. решение	
y_1	$x_1 = 0$	$x_1 = 1$	$x_1 = 2$	$f_1(y_1)$	x_1^*
6	.6*.35=.210	–	–	.210	1
7	.6*.35=.210	.8*.35=.280	–	.280	2
8	.6*.49=.294	.8*.35=.280	.9*.35=.315	.315	3
9	.6*.63=.378	.8*.49=.392	.9*.35=.315	.392	2
10	.6*.63=.378	.8*.63=.504	.9*.49=.441	.504	2

Таблица 18. Задача о надежности, этап 1

нашей системы составит 0.504, а программа печатает в таблице просто 504, без нуля и десятичной точки.

```
type Nad

% программа Nad.m  Лаборатор. номер 3
% =====5 апреля 2006 г.=====Визгунов Н.П.

format compact, clc

y1 = 10
C = [  1    3     2
       2    5     4
       3    6     5  ]
R = [  0.6  0.7   0.5
       0.8  0.8   0.7
       0.9  0.9   0.9]

[u9,n] = size(C);   % u9 - Макс. количество управлений
s9 = y1 + 1;        % s9 - Макс. количество состоЯний
R = round(R * 10);

y_min = zeros(1, n+1);
for j = n : -1 : 1
    y_min(j) = y_min(j +1) + C(1,j);
end
y_max = zeros(1, n);
y_max(1) = y1;
for j = 2 : n
    y_max(j) = y_max(j -1) - C(1, j -1);
end
x_max = sum(finite(R)); % isfinite(R) в ver. 5.0

s8 = max(y_max - y_min(1 :n) +1);
X = -inf * ones(s8, n +1);
F = X;
F(:, n +1) = ones(s8, 1);

for j = n : -1 :1

    s7 = y_max(j) - y_min(j) + 1;
    u7 = x_max(j);
    Fsu = -inf * ones(s7, u7);
    for s = 1 : s7      % было Yj = s -1
        for u = 1 : u7  % было    Xj = u
            Yj = y_min(j) + s - 1;
            s1 = (Yj - C(u,j)) - y_min(j +1) + 1;
            if 1 <= s1 & s1 <= s7
                Fsu(s, u) = R(u, j) * F(s1, j+1);
```

```
            end
        end % for u
    end % for s
    [fs, ind] = max(Fsu');
    F(1: s7, j) = fs';
    X(1: s7, j) = ind';

    Tire = setstr(ones(1, (u7 +3) * 6) * '-');
    Js = int2str(j);
    disp(' ')
    disp(['Этап ' int2str(j)])
    disp(Tire)
    disp(['    Yj  F(:,j) X(:,j)', blanks(u7*3 -2), 'Xj' ])
    disp(Tire)
    disp([(y_min(j) : y_max(j))',  fs', ind', Fsu ])
    disp(' ')
end

% печать результатов
% ==================

x_opt = zeros(1, n);
Yj = y1;
for j = 1 : n
    s = Yj - y_min(j) + 1;
    x_opt(j) = X(s,j);
    Yj = Yj - C(x_opt(j),j);
end
x_opt, f_opt=F(s8, 1)

Nad
y1 =
    10
C =
     1     3     2
     2     5     4
     3     6     5
R =
    0.6000    0.7000    0.5000
    0.8000    0.8000    0.7000
    0.9000    0.9000    0.9000

Этап 3
------------------------------------
    Yj  F(:,j) X(:,j)       Xj
------------------------------------
     2     5     1     5  -Inf  -Inf
     3     5     1     5  -Inf  -Inf
     4     7     2     5     7  -Inf
     5     9     3     5     7     9
     6     9     3     5     7     9


Этап 2
------------------------------------
    Yj  F(:,j) X(:,j)       Xj
------------------------------------
     5    35     1    35  -Inf  -Inf
     6    35     1    35  -Inf  -Inf
     7    49     1    49    40  -Inf
     8    63     1    63    40    45
     9    63     1    63    56    45
```

```
Этап 1
------------------------------------
    Yj  F(:,j) X(:,j)       Xj
------------------------------------
     6   210     1   210  -Inf  -Inf
     7   280     2   210   280  -Inf
     8   315     3   294   280   315
     9   392     2   378   392   315
    10   504     2   378   504   441

x_opt =
     2     1     3
f_opt =
   504
type NadP

% Программа NadP.m
% ===================
format compact
y1 = 10
C = [  1    3     2
       2    5     4
       3    6     5  ]
R = [  0.6  0.7   0.5
       0.8  0.8   0.7
       0.9  0.9   0.9]
[u9,n] = size(C); % u9 - число возможных управлений
R = round(R * 10);

x_min = ones(1,n);
x_max = sum(finite(R));
x = x_min;

f_opt = -inf;
x_opt = zeros(1,n);

j = n;
while 1
    if  x(j) <= x_max(j)
        cost = 0;
        for i = 1:n
            cost = cost + C(x(i),i);
        end
        if cost <= y1
            f = 1;
            for i = 1 :n
               f = f * R(x(i), i);
            end
            if  f_opt < f
                f_opt = f
                x_opt = x
            elseif  f_opt == f
                f_opt
                x_opt = [x_opt; x ]
            end
        end % cost
        j = n;
    else    % x(j) > x_max(j)
        x(j) = x_min(j);
        j = j - 1;
        if j <= 0.001
            break
        end
    end % if
```

```
    x(j) = x(j) + 1;
end % while 1

disp('===== Ответ: =====')
disp(sprintf('f_opt = 0.%-6.0f', f_opt))
%disp(['f_opt = 0.', int2str(f_opt)])
x_opt

NadP
y1 =
    10
C =
     1     3     2
     2     5     4
     3     6     5
R =
    0.6000    0.7000    0.5000
    0.8000    0.8000    0.7000
    0.9000    0.9000    0.9000
f_opt =
   210
x_opt =
     1     1     1
f_opt =
   294
x_opt =
     1     1     2
f_opt =
   378
x_opt =
     1     1     3
f_opt =
   392
x_opt =
     2     1     2
f_opt =
   504
x_opt =
     2     1     3
===== Ответ: =====
f_opt = 0.504
x_opt =
     2     1     3
diary off
```

6 Задача календарного планирования трудовых ресурсов

Задача 4 *Предпринимателю необходимо составить план регулирования численности рабочих на последующие пять месяцев – январь, февраль, март, апрель и май. Он оценивает следующим образом* b_j *– минимальные потребности в рабочей силе:*
$(b_1, \ldots, b_j, \ldots, b_n) = (5, 7, 8, 4, 6)$. *Другими словами, в январе ему потребуется не меньше 5 человек, в феврале 7 человек или больше и т.д.*

Предприниматель может принимать новых людей на работу и увольнять их. Пусть y_j *– количество рабочих, которые работают в месяце* j. *Очевидно, должно выполняться* $y_j \geq b_j$. *При лишних рабочих* $y_j > b_j$ *предприниматель терпит убытки, которые вычисляются по формуле*

$$C_л(y_j - b_j) = 3(y_j - b_j), \; j \in 1:5.$$

Расходы, связанные с наймом новых рабочих, вычисляются по другой формуле:

$$C_H(y_j - y_{j-1}) = \begin{cases} 4 + 2(y_j - y_{j-1}), & \text{при } y_j > y_{j-1}, \\ 0 & \text{иначе.} \end{cases}$$

Необходимо составить оптимальный план численности рабочих на 5 месяцев, при условии, что в конце декабря работало 5 человек.

Рекуррентное соотношение записывается в виде:

$$\begin{aligned} f_5(y_4) &= \min_{y_5 = b_5} \{C_Л(y_5 - b_5) + C_Н(y_5 - y_4)\}, \\ f_j(y_{j-1}) &= \min_{y_j \geq b_j} \{C_Л(y_j - b_j) + C_Н(y_j - y_{j-1}) + f_{j+1}(y_j)\}, \\ & j = 4, 3, 2, 1. \end{aligned}$$

Перед тем, как приступить к вычислениям, надо определить границы изменения переменных $y_1, \ldots, y_5$. Легко понять, что для получения оптимального решения достаточно рассмотреть значения $y_1 \in 5 : 8$, $y_2 \in 7 : 8$, $y_3 = 8$, $y_4 \in 4 : 6$, $y_5 = 6$. Дело в том, что больше 8 человек не потребуется никогда, поэтому $y_j \leq 8$. Более жесткое ограничение $y_4 \leq 6$ тоже совершенно понятно. y_4 - это количество рабочих в 4 месяце, в апреле. В мае нам потребуется ровно 6 человек, поэтому больше 6 рабочих нет необходимости оплачивать и в предыдущем месяце, в апреле. Дешевле лишних людей уволить в начале апреля, так как мы увольняем в нашей задаче рабочих без выходного пособия.

Решим задачу с помощью таблиц и рекуррентного уравнения Беллмана.

Этап 5. Май.

y_4	$b_5 = 6$ $C_Л(y_5 - 6) + C_Н(y_5 - y_4)$	Оптимальное решение	
	$y_5 = 6$	$f_5(y_4)$	y_5^*
4	0+8=8	8	6
5	0+6=6	6	6
6	0+0=0	0	6

Таблица 19. Календарное планирование, этап 5

Этап 4. Апрель, май.

y_3	$b_4 = 4$ $C_Л(y_4 - 4) + C_Н(y_4 - y_3) + f_5(y_4)$			Оптимальное решение	
	$y_4 = 4$	$y_4 = 5$	$y_4 = 6$	$f_4(y_3)$	y_4^*
8	0+0+8=8	3+0+6=9	6+0+0=6	6	6

Таблица 20. Календарное планирование, этап 4

Этап 3. Март, апрель, май.

y_2	$b_3 = 8$ $C_Л(y_3 - 8) + C_Н(y_3 - y_2) + f_4(y_3)$	Оптимальное решение	
	$y_3 = 8$	$f_3(y_2)$	y_3^*
7	0+6+6=12	12	8
8	0+0+6= 6	6	8

Таблица 21. Календарное планирование, этап 3

Графическое решение достаточно сложное, поэтому это решение представлено на рисунке 3.

Также изобразим рисунок, поясняющий исходные данные и ответ.

Еще раз подчеркнем, что ответом задачи является:

$$y^* = (5, 8, 8, 6, 6)$$

$$f_1(5) = 19.$$

Этап 2. Февраль, март, апрель, май.

	$b_2=7$ $C_{\text{Л}}(y_2-7)+C_{\text{Н}}(y_2-y_1)+f_3(y_2)$		Оптимальное решение	
y_1	$y_2=7$	$y_2=8$	$f_2(y_1)$	y_2^*
5	0+8+12=20	3+10+6=19	19	8
6	0+6+12=18	3+ 8+6=17	17	8
7	0+0+12=12	3+ 6+6=15	12	7
8	0+0+12=12	3+ 0+6= 9	9	8

Таблица 22. Календарное планирование, этап 2

Этап 1. Январь, февраль, март, апрель, май.

	$b_1=5$ $C_{\text{Л}}(y_1-5)+C_{\text{Н}}(y_1-y_0)+f_2(y_1)$				Оптимальное решение	
y_0	$y_1=5$	$y_1=6$	$y_1=7$	$y_1=8$	$f_2(y_1)$	y_2^*
5	0+0+ +19=19	3+6+ +17=26	6+8+ +12=26	9+10+ +9=28	19	5

Таблица 23. Календарное планирование, этап 1

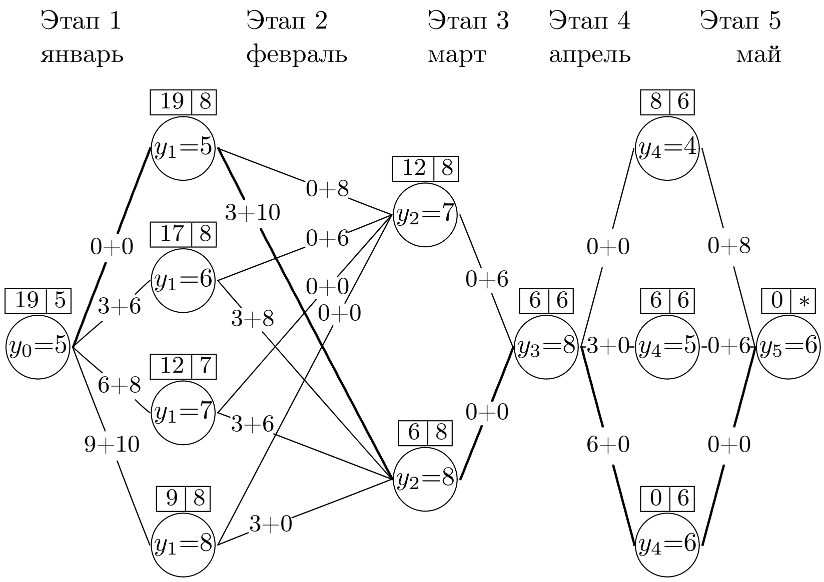

Рис. 3. Задача о календарного планирования трудовых ресурсов.

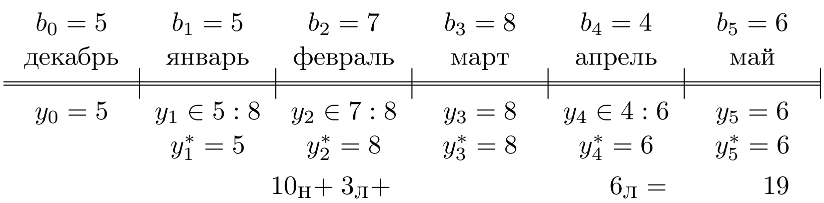

Рис. 4. Графическая иллюстрация исходных данных и ответа в задаче о календарном планировании трудовых ресурсов.

6.1 Календарное планирование на компьютере

В распечатке есть главная программа `TrudMain.m`, содержащая исходные данные. Главная программа обращается к двум подпрограммам (`function`), которые размещены в отдельных файлах `TrudD.m` и `TrudP.m` и решают задачу методом динамического программирования и методом полного перебора. Это стандартный способ организации большой программы, принятый в большинстве языков программирования. В этой работе способ этот применяется впервые, потому что переменные из подпрограмм не видны в MATLAB, и у студентов возникает много проблем при отладке.

```
type TrudMain

format compact, clc

y_max = [8 8 8 6 6 ];
b     = [5 7 8 4 6 ];
y0 = 5;

TrudD(y0,y_max,b);
TrudP(y0,y_max,b);

type TrudD

% программа TrudD.m Динамическое программирование
%==========Визгунов Н.П. 7 апреля 2006===========

function [] = TrudD(y0,y_max,b)

n  = length(b);
s9 = max(y_max -b +1);

F = inf * ones(s9, n +1);
Y = F;
F(:, n+1) = zeros(s9, 1);

for j = n : -1 : 1
    s7 = 1;
    if j >= 2,  s7 = y_max(j -1) - b(j -1) +1; end
    u7 = y_max(j) - b(j) +1;
    Fsu = inf * ones(s7, u7);
    for s = 1 : s7
        Yj_1 = y0;
        if j >= 2,   Yj_1 = s + b(j -1) -1; end
        for u = 1 : u7
            Yj = u + b(j) -1;
            Cn = 0;
            if Yj > Yj_1, Cn = 4 + 2* (Yj - Yj_1); end
            Fsu(s,u) = 3 * (Yj - b(j)) + Cn + F(u, j +1);
        end % for u
    end % for s

    % обработка временной таблицы Fsu

    if u7 == 1 % это вектор - столбец высотой s7
        fs = Fsu';
        index = ones(1,s7);
    else
        [fs, index] = min(Fsu');
    end
    F(1: s7, j) = fs';
    Y(1: s7, j) = index';

    % печать полученных таблиц

    Tire = setstr(ones(1, (u7 +3) *6 +3) * '=');
%   Tire = char(ones(1, (u7 +3) *6 +3) * '=');
%   в версии 5.x лучше заменить setstr() на char()
    Kepka = ['    Y' int2str(j-1) ' '];
    for u = 1 : u7
        Kepka = [Kepka, '  Y', int2str(j), ...
```

```
                         '=', int2str(b(j) +u -1)];
    end
    Kepka = [Kepka ' F(:,j) Y(:,j)'];

    disp(Tire)
    disp(Kepka)
    disp(Tire)
%   можно напечатать более простую шапку одним оператором
%   disp(['Y(j-1)', blanks(u7*3), 'Fsu ', ...
%                    blanks(u7*3 -3), 'F(:,j) Y(:,j)'])

    if j >= 2  % из-за b(0) надо разделять случаи
        disp([(b(j-1) : b(j -1) +s7 -1)', ...
                        Fsu,  fs', index'+b(j)-1])
    else
        disp([y0,  Fsu,  fs',  index'+b(j)-1])
    end
    disp(' ') %
end

% печать результатов
% ==================
y_opt = zeros(1, n);
s = 1;
for j = 1 : n
    s = Y(s,j);
    y_opt(j) = s;
end % for j
disp('**********ответ ДП************')
f_opt = F(1, 1)
y_opt = y_opt +b -1
disp('******************************')

type TrudP

%   программа KaleP.m Визгунов Н.П.
% =======5 апреля 2006г.==========

function [] = TrudP(y0,y_max,b)
disp('==============полный перебор==============')

n  = length(b);
y_max = [y0 y_max];
b     = [y0     b];
n = n +1;
f_opt = inf;
y_opt = zeros(1, n +1);

y = b;
j = n;
while 1
    if y(j) <= y_max(j)

        f = 0;
        for j = 2 : n
            Cn = 0;
            if y(j) > y(j -1)
                Cn = 4 + 2 *(y(j) - y(j -1));
            end
            f = f + 3 *(y(j) -b(j)) +Cn;
        end % for j
        if  f_opt > f % задача решается на min
            f_opt = f
            y_opt = y(2 :n)
```

```
        elseif f_opt == f
            f_opt
            y_opt = [y_opt; y(2:n)]
        end % elseif

        j = n;
    else    % y(j) > y_max(j)
        y(j) = b(j);
        j = j -1;
        if j <= 0.0001
            break
        end % if
    end % if
    y(j) = y(j) + 1;
end % while 1

disp('====='), disp('ответ'), disp('=====')
f_opt, y0, b = b(2:n), y_opt

TrudMain
===========================
    Y4   Y5=6 F(:,j) Y(:,j)
===========================
     4     8     8     6
     5     6     6     6
     6     0     0     6

=======================================
    Y3   Y4=4  Y4=5  Y4=6 F(:,j) Y(:,j)
=======================================
     8     8     9     6     6     6

===========================
    Y2   Y3=8 F(:,j) Y(:,j)
===========================
     7    12    12     8
     8     6     6     8

=================================
    Y1   Y2=7  Y2=8 F(:,j) Y(:,j)
=================================
     5    20    19    19     8
     6    18    17    17     8
     7    12    15    12     7
     8    12     9     9     8

=============================================
    Y0   Y1=5  Y1=6  Y1=7  Y1=8 F(:,j) Y(:,j)
=============================================
     5    19    26    26    28    19     5

**********ответ ДП************
f_opt =
    19
y_opt =
     5     8     8     6     6
******************************
==============полный перебор==============
f_opt =
    22
y_opt =
     5     7     8     4     6
f_opt =
    20
```

```
y_opt =
     5     7     8     6     6
f_opt =
    19
y_opt =
     5     8     8     6     6
=====
ответ
=====
f_opt =
    19
y0 =
     5
b =
     5     7     8     4     6
y_opt =
     5     8     8     6     6
diary off
```

7 Задача о дилижансах

Задача 5 *Путешественник собирается пересечь нашу страну с востока на запад и добраться из Владивостока в далекий Калининград. Перед ним лежит карта, на которой указаны все маршруты. Для каждого маршрута указана цена билета. Выбрать самый дешевый маршрут, соединяющий Владивосток и Калининград.*

Карта, которая лежит перед нашим путешественником, приведена на рисунке 5. Карта немножко своеобразная, потому что любой маршрут, соединяющий город Владивосток ($y_5 = 10$) и город Калининград ($y_1 = 1$), состоит из 4 участков. Никаких прямых рейсов нет. Решить графически такую задачу не составляет никакого труда, это и сделано на рисунке. Решать задачу надо сначала для городов 7, 8 и 9 обычным способом. Это будет этап 4. Далее решается задача динамического программирования для городов 5 и 6 (этап 3). Затем надо выполнить для городов 2, 3 и 4(этап 2). Последним выполняется этап 1. Он соответствует самому западному городу Калиниграду, который имеет номер $y_1 = 1$ на рисунке.

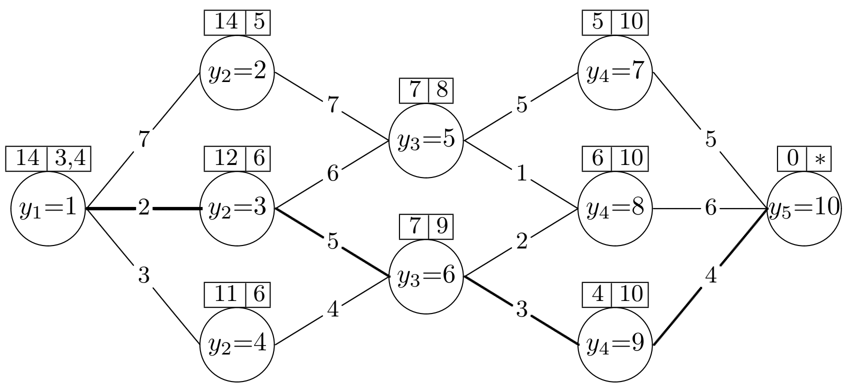

Рис. 5. Задача о дилижансах.

Мы еще не написали ни одной формулы, а уже решили задачу. Ответ сейчас можно написать так. Минимальные расходы равны 14 денежных единиц. Что такое денежная единица, уточнять не будем, так как при езде на дилижансах денежная единица может быть и не совсем традиционной. Один из оптимальных маршрутов – это набор городов $(y_1^*, \ldots, y_5^*) = (1, 3, 6, 9, 10)$.

Эту же задачу можно решить и с помощью таблиц. В этом случае без формул уже не обойтись.

Уравнение Беллмана будет выглядеть следующим образом:

$$f_j(y_j) = \min_{\substack{y_{j+1} | \text{маршрут} \\ (y_j, y_{j+1}) - \text{существует}}} \{R(y_j, y_{j+1}) + f_{j+1}(y_{j+1})\}.$$

Здесь нет традиционных переменных x_j, хотя нет никаких сомнений, что мы только что решили задачу динамического программирования. Состояния здесь – это y_j; $f_j(y_j)$ – минимальные расходы на переезд в соответствующий город этапа j. Что же здесь является управлением? Да, конечно же, переменная y_{j+1}, по которой мы ищем минимум.

Сейчас решим задачу о дилижансах с помощью таблиц. Как и раньше, по строчкам будут располагаться все возможные состояния, по столбцам — перечисляются все значения управлений.

Этап 4:

$$f_4(y_4) = \min_{y_5=10} \{R(y_4, y_5) + f_5(y_5)\}.$$

	$R(y_4, y_5) + 0$	Оптимальное решение	
y_4	$y_5 = 10$	$f_4(y_4)$	y_5^*
7	5	5	10
8	6	6	10
$\underline{9}$	4	$\underline{4}$	$\underline{10}$

Таблица 24. Задача о дилижансах, этап 4

Этап 3:

$$f_3(y_3) = \min_{y_4 \in \{7,8,9\}} \{R(y_3, y_4) + f_4(y_4)\}.$$

	$R(y_3, y_4) + f_4(y_4)$			Оптимальное решение	
y_3	$y_4 = 7$	$y_4 = 8$	$y_4 = 9$	$f_3(y_3)$	y_4^*
5	5+5=10	1+6=7	—	7	8
$\underline{6}$	—	2+6=8	3+4=7	$\underline{7}$	$\underline{9}$

Таблица 25. Задача о дилижансах, этап 3

Этап 2:

$$f_2(y_2) = \min_{y_3 \in \{5,6\}} \{R(y_2, y_3) + f_3(y_3)\}.$$

	$R(y_2, y_3) + f_3(y_3)$		Оптимальное решение	
y_2	$y_3 = 5$	$y_3 = 6$	$f_2(y_2)$	y_3^*
2	7+7=14	—	14	5
$\underline{3}$	6+7=13	5+7=12	$\underline{12}$	$\underline{6}$
4	—	4+7=11	11	6

Таблица 26. Задача о дилижансах, этап 2

Этап 1:

$$f_1(y_1) = \min_{y_2 \in \{2,3,4\}} \{R(y_1, y_2) + f_2(y_2)\}.$$

	$R(y_1, y_2) + f_2(y_2)$			Оптимальное решение	
y_1	$y_2 = 2$	$y_2 = 3$	$y_2 = 4$	$f_1(y_1)$	y_2^*
$\underline{1}$	7+14=21	2+12=14	3+11=14	14	$\underline{3}$,4

Таблица 27. Задача о дилижансах, этап 1

8 Управление запасами

Задача 6 (Управление запасами.) *Завод может производить ежемесячно до 4 единиц некоторой продукции. Затраты на производство x_j единиц продукции в месяце j указаны в следующей таблице 28. Каждый месяц завод должен отгрузить 2 единицы продукции своим потребителям. Продукцию можно хранить на складах. Затраты на хранение 1 единицы продукции составляют 1 миллион рублей. Оплата за хранение производится в конце месяца j. Склады могут вместить до 4 единиц продукции. Учитывая, что в начале на складах было 2 единицы продукции, определить оптимальный план производства на 4 месяца.*

x_j	0	1	2	3	4	штук
$C(x_j)$	0	7	9	11	13	млн. руб.

Таблица 28. Затраты на производство x_j единиц продукции.

Введем обозначения для переменных и констант задачи.

n – количество месяцев планового отрезка,

x_j – выпуск продукции в месяце j, $j = 1, \dots, n$ (управление),

y_j – запасы в конце месяца j, или в начале месяца $j+1$ (состояние),

d_j – спрос в месяце j, заданная величина.

Переменные x_j и y_j связаны следующим балансовым соотношением

$$y_j = y_{i-1} + x_j - d_j, \quad j = 1, \dots, n$$

— запасы y_j в конце текущего j-го месяца равны запасам в предыдущем месяце y_{j-1} плюс x_j произведенных в текущем месяце j изделий и минус d_j проданных изделий.

В числовом примере, который мы будет использовать, $n = 4$. Мы будет планировать на январь, февраль, март и апрель. $d_j = 2, \ j = 1, \dots, n, \quad x_j, y_j \in 0:4$

Математическая модель задачи выглядит следующим образом. Найти оптимальный план производства $x_1, \dots, x_n$, минимизирующий затраты

$$\left.\begin{array}{l}
f_1(y_0) = \min\limits_{x_1,\dots,x_n} \left\{ \sum\limits_{j=1}^{n} C(x_j) + 1(y_{j-1} + x_j - 2) \right\} \\
\text{при ограничениях} \\
\begin{array}{lllllll}
y_0 & +x_1 & -2 & & & & = y_1, \\
 & y_1 & +x_2 & -2 & & & = y_2, \\
 & & \cdots\cdots & & & & \\
 & & y_{j-1} & +x_j & -2 & & = y_j, \\
 & & & & \cdots\cdots & & \\
 & & & & y_{n-1} & +x_n - 2 & = y_n,
\end{array} \\
x_j, y_j \in 0:4 \text{ для любого } j, \\
y_0 = 2, \ y_n = 0.
\end{array}\right\} \qquad (8)$$

Для решения этой задачи следует записать уравнение Беллмана. Для этого надо сложить затраты в месяце j и затраты на временном отрезке $j+1, \dots, n$, а затем минимизировать суммарные затраты выбором x_j:

$$f_j(y_{j-1}) = \min_{x_j}\{C(x_j) + 1 \cdot (y_{j-1} + x_j - 2) + f_{j+1}(y_{j-1} + x_j - 2)\}.$$

Справедливость уравнения Беллмана легко доказывается, несмотря на то, что в исходной задаче 8 довольно много ограничений.

Когда строго выводят уравнение Беллмана из задачи 8, то математики говорят об этом факте следующей фразой: « *доказано,* *что принцип оптимальности Беллмана справедлив»*. Мы проводили уже один раз такое доказательство в самой первой задаче – в задаче о распределении инвестиций. Не будем повторять эти рассуждения, так как они достаточно простые.

Начинаем решение задачи о запасах с последнего месяца — апреля.

Этап 4: апрель.

$$f_4(y_3) = \min_{x_4 \in 0:4} \{C(x_4) + 1 \cdot (y_3 + x_4 - 2)\}.$$

y_3	Оптимальное решение $f_4(y_3)$	x_4^*
<u>0</u>	<u>9</u>	<u>2</u>
1	7	1
2	0	0

Таблица 29. Управление запасами, этап 4

В конце апреля запасы на складе должны быть равны нулю $y_3 + x_4 - 2 = y_4 = 0$, отсюда $y_3 + x_4 = 2$ и все возможные значения y_3 и x_4 в таблице 29 перечислены. Таких пар всего три. Для каждого возможного значения (y_3, x_4) вычислена целевая функция по совсем простой формуле $C(x_4)$.

Этап 3: март, апрель.

$$f_3(y_2) = \min_{x_3 \in 0:4} \{C(x_3) + 1 \cdot (y_2 + x_3 - 2) + f_4(y_2 + x_3 - 2)\}.$$

	$C(x_3) + 1 \cdot (y_2 + x_3 - 2) + f_4(y_2 + x_3 - 2)$					Оптим. решение	
y_2	$x_3 = 0$	$x_3 = 1$	$x_3 = 2$	$x_3 = 3$	$x_3 = 4$	$f_3(y_2)$	x_3^*
0	—	—	9+0+9 =18	11+1+7 =19	13+2+0 =15	15	4
1	—	7+0+9 =16	9+1+7 =17	11+2+0 =13	—	13	3
<u>2</u>	0+0+9 =9	7+1+7 =15	9+2+0 =11	—	—	<u>9</u>	<u>0</u>
3	0+1+7 =8	7+2+0 =9	—	—	—	8	0
4	0+2+0 =2	—	—	—	—	2	0

Таблица 30. Управление запасами, этап 3

Для этапа 3 (март, апрель) из $y_3 \in 0:4,\ \ y_3 = y_2 + x_3 - 2$ следует

$$0 \le y_2 + x_3 - 2 \le 4.$$

Левое ограничение $0 + 2 \le y_2 + x_3$ дает запрещенные варианты в левой верхней части таблицы 30, правое ограничение $y_2 + x_3 \le 4 + 2$ соответствует черточкам в правой нижней части таблицы. В этой таблице уже перечислены все пять возможных значений состояния $y_2 \in 0:4$.

Для этапа 2 «северо-западные» и «юго-восточные» углы матрицы заполняются черточками из-за ограничения $0 \le y_2 = y_1 + x_2 - 2 \le 4$.

8.1 Вычисление оптимального решения

Находим оптимальное решение, начиная с последнего этапа 1. $y_0 = 2$ из условий задачи, поэтому из таблицы 32 следует, что $x_1^* = 0,\ \ y_1 = y_0 + x_1^* - 2 = 0$. В таблице этапа 2 в строке $y_1 = 0$ видим два оптимальных решения. Рассмотрим сначала решение $x_2^* = 4$. Это решение подчеркнуто. Из уравнения $y_2 = y_1 + x_2^* - 2$ находим $y_2 =$

Этап 2: февраль, март, апрель.

$$f_2(y_1) = \min_{x_2 \in 0:4} \{C(x_2) + 1 \cdot (y_1 + x_2 - 2) + f_3(y_1 + x_2 - 2)\}.$$

	$C(x_2) + 1 \cdot (y_1 + x_2 - 2) + f_3(y_1 + x_2 - 2)$					Оптим. решение	
y_1	$x_2 = 0$	$x_2 = 1$	$x_2 = 2$	$x_2 = 3$	$x_2 = 4$	$f_2(y_1)$	x_2^*
$\underline{0}$	—	—	9+0+15 =24	11+1+13 =25	13+2+9 =24	$\underline{24}$	2, $\underline{4}$
1	—	7+0+15 =22	9+1+13 =23	11+2+9 =22	13+3+8 =24	22	1,3
2	0+0+15 =15	7+1+13 =21	9+2+9 =20	11+3+8 =22	13+4+2 =19	15	0
3	0+1+13 =14	7+2+9 =18	9+3+8 =20	11+4+2 =17	—	14	0
4	0+2+ 9 =11	7+3+8 =18	9+4+2 =15	—	—	11	0

Таблица 31. Управление запасами, этап 2

Этап 1: январь,февраль, март, апрель.

$$f_1(y_0) = \min_{x_1 \in 0:4} \{C(x_1) + 1 \cdot (y_0 + x_1 - 2) + f_2(y_0 + x_1 - 2)\}.$$

	$C(x_1) + 1 \cdot (y_0 + x_1 - 2) + f_2(y_0 + x_1 - 2)$					Оптим. решение	
y_0	$x_1 = 0$	$x_1 = 1$	$x_1 = 2$	$x_1 = 3$	$x_1 = 4$	$f_1(y_0)$	x_1^*
0	—	—	9+0+24 =33	11+1+22 =34	13+2+15 =30	30	4
1	—	7+0+24 =31	9+1+22 =32	11+2+15 =28	13+3+14 =30	28	3
$\underline{2}$	0+0+24 =24	7+1+22 =30	9+2+15 =26	11+3+14 =28	13+4+11 =28	$\underline{24}$	$\underline{0}$
3	0+1+22 =23	7+2+15 =24	9+3+14 =26	11+4+11 =26	—	23	0
4	0+2+15 =17	7+3+14 =24	9+4+11 =24	—	—	17	0

Таблица 32. Управление запасами, этап 1

$0+4-2 = 2$. Переходим к таблице этапа 3 к строчке $y_2 = 2$. Понятно, что соответствующая компонента решения $x_3^* = 0$ и в таблице этапа 4 надо смотреть на строку $y_3 = y_2 + 0 - 2 = 0$. Эта строка $y_2 = 0$ дает последнюю компоненту решения $x_4^* = 2$.

Теперь можно записать окончательный ответ.

$$x^* = (0, 4, 0, 2), \quad f_1(2) = 24.$$

В феврале надо произвести 4 единицы продукции и в апреле 2 единицы, при этом за четыре месяца будет затрачено 24 миллиона рублей. С меньшими затратами нельзя выполнить намеченную программу. Этот оптимальный план не единственный. Другой оптимальный план получить самостоятельно.

8.2 Управление запасами на компьютере

В методе полного перебора полезно напечатать в качестве ответа не только x^* — выпуск продукции в каждом месяце, но и запасы в конце каждого месяца y^*. Например, из компьютерного решения видно, что одним из ответов является следующий:

$$f_1(2) = 24$$
$$x^* = (0, 4, 0, 2)$$
$$y^* = (0, 2, 0, 0)$$

```
type ZapasD

%   программа ZapasD.m Динамическое программирование
%   =====Визгунов Н.П. 5 апреля 2006г.============

format compact, clc
% x:  1  2  3  4  5
c1 = [0  7  9 11 13]; % Затраты на пр-во x-1 ед. продукции
x9 = length(c1) - 1;  % Максимум производства в штуках
y0 = 2;               % Запасы на складе в конце декабря
d = 2;                % Отгрузка потребителям
y9 = 4;               % Максимум запасов на складе в штуках
n = 4;                % Количество месяцев в плановом периоде

s9 = y9 + 1;          % Макс. количество состояний
u9 = x9 + 1;          % Макс. количество управлений

F = inf * ones(y9 +1, n);
X = F;

% Начальные вычисления для этапа n.

for s = 1 : s9
    u = d -s +2;
    if u >= 1
        F(s, n) = c1(u);
        X(s, n) = u;
    end
end % for s

% Печать для этапа n

Tire = setstr(ones(1, 3*6 +2) * '=');
Kepka = ['    Y', int2str(n -1), '  F(:,j) Y(:,j)'];
s7 = sum(finite(F(:, n)));
disp(Tire)
disp(Kepka)
disp(Tire)
disp([(0: s7 -1)', F(1: s7, n), X(1: s7, n)-1])
disp(' ')

% Произвольный j-ый этап.

Fsu = inf * ones(s9, u9);
for j = n -1 : -1 : 1
    for s = 1 : s9          % s = Yj_1 +1
        for u = 1 : u9      % u = Xj +1
            s1 = (s -1) + (u -1) -d +1;
            if  1  <= s1  & s1 <= n  +1
                Fsu(s, u) = c1(u) + (s1 -1) + F(s1, j+1);
            end
        end % for u
    end % for s
    [fs, index] = min(Fsu');

    % Обработка таблиц Fsu     F(состояние, управление)
    F(:, j) = fs';
    X(:, j) = index';

    % Печать таблиц
    Tire = setstr(ones(1, (u9 +3) *6 + 3) *'=');
%    Kepka = ['  Y',int2str(j -1), '  '];
%    for u = 1 : u9
%        Kepka = [Kepka,'  X', int2str(j), ...
```

```
%                            '=', int2str(u -1)];
%    end
%    Kepka = [Kepka, ' F(:,j) Y(:,j)'];

    Kepka = ['    Y', int2str(j -1), '  '];
    Frmt=['X', int2str(j), '=%-3.0f'];

    % Kepka = [Kepka, sprintf(Frmt,  (0 : u9 -1))];
    % не работает в MatLab 3.5g, приходится расписывать:
    for i = 0: u9 -1
        Kepka = [Kepka,  sprintf(Frmt,i)];
    end
    Kepka = [Kepka, 'F(:,j) Y(:,j)'];

    disp(Tire)
    disp(Kepka)
    disp(Tire)
    disp([(0: s9 -1)', Fsu, F(:,j), X(:,j)-1 ])
    disp(' ')
end % for j

% печать результатов
% ==================
x_opt = zeros(1, n);
s = y0 + 1;
for j = 1 : n
   x_opt(j) = X(s,j);
   s = (s -1) +X(s,j) -d ;
end % for j
x_opt = x_opt -1
f_opt = F(y0 +1, 1)

ZapasD
====================
    Y3  F(:,j) Y(:,j)
====================
     0     9     2
     1     7     1
     2     0     0

===================================================
    Y2  X3=0  X3=1  X3=2  X3=3  X3=4  F(:,j) Y(:,j)
===================================================
     0   Inf   Inf    18    19    15    15     4
     1   Inf    16    17    13   Inf    13     3
     2     9    15    11   Inf   Inf     9     0
     3     8     9   Inf   Inf   Inf     8     0
     4     2   Inf   Inf   Inf   Inf     2     0

===================================================
    Y1  X2=0  X2=1  X2=2  X2=3  X2=4  F(:,j) Y(:,j)
===================================================
     0   Inf   Inf    24    25    24    24     2
     1   Inf    22    23    22    24    22     1
     2    15    21    20    22    19    15     0
     3    14    18    20    17   Inf    14     0
     4    11    18    15   Inf   Inf    11     0

===================================================
    Y0  X1=0  X1=1  X1=2  X1=3  X1=4  F(:,j) Y(:,j)
===================================================
     0   Inf   Inf    33    34    30    30     4
     1   Inf    31    32    28    30    28     3
     2    24    30    26    28    28    24     0
```

```
     3    23    24    26    26   Inf    23     0
     4    17    24    24   Inf   Inf    17     0

x_opt =
     0     2     4     0
f_opt =
    24
type ZapasP

%      Программа ZapasP.m
% =====Визгунов Н.П. 5 апреля 2006г.=============

format compact, clc
% x:  1  2  3  4  5
c1 = [0  7  9 11 13]; % Затраты на пр-во x-1 ед. продукции
x9 = length(c1) - 1;  % Максимум производства в штуках
y0 = 2;               % Запасы на складе в конце декабрЯ
y9 = 4;               % Максимум запасов на складе в штуках
k = 1;                % Плата за единицу хранениЯ на складе

d = 2;                % Отгрузка потребителЯм
n = 4;                % Количество месЯцев в плановом периоде

s9 = y9 + 1;          % Макс. количество состоЯний
u9 = x9 + 1;          % Макс. количество управлений

x = zeros(1, n);
f_opt = inf;
x_opt = x;

j = n;
while 1
    if  x(j) <= x9

        y(1) = y0 + x(1) - d;
        for j = 2 : n
            y(j) = y(j-1) + x(j) - d;
        end % for

        if  0 <= min(y) & max(y) <= y9

            f = 0;
            for j = 1:n
                f = f + c1(x(j) +1) + k * y(j);
            end
            if  f_opt > f
                f_opt = f
                x_opt = x
                y_opt = y;
            elseif f_opt == f
                f_opt = f
                x_opt = [x_opt; x ]
                y_opt = [y_opt; y ];
            end
        end % if

        j = n;
    else
        x(j) = 0;
        j = j - 1;
        if j <= 0.0001
            break
        end % if
    end % if
```

```
    x(j) = x(j) + 1;
end

disp('========= Ответ: =========')
f_opt, x_opt, y_opt

ZapasP
f_opt =
    27
x_opt =
     0     2     2     2
f_opt =
    24
x_opt =
     0     2     4     0
f_opt =
    24
x_opt =
     0     2     4     0
     0     4     0     2
========= Ответ: =========
f_opt =
    24
x_opt =
     0     2     4     0
     0     4     0     2
y_opt =
     0     0     2     0
     0     2     0     0
diary off
```

9 Замена оборудования.

Задача 7 *К началу текущей пятилетки на предприятии установлено новое оборудование. Производительность этого оборудования и затраты на содержание и ремонт зависят от времени. В таблице 33 показаны эти зависимости. Затраты на приобретение и установку нового оборудования составляют 40 тысяч рублей, старое оборудование списывается. Составить план замены оборудования на пять лет, максимизирующий общую прибыль.*

	Время y_j, в течение которого используется оборудование (лет)				
$y_j =$	0	1	2	3	4
Годовой выпуск продукции $R(y_j)$ (тысяч рублей)	80	75	65	60	60
Ежегодные затраты $C(y_j)$ на содержание и ремонт оборудования (тысяч руб.)	20	25	30	35	45

Таблица 33. Задача о замене оборудования

Введем следующие обозначения.

x_j – управление. Переменная x_j может принимать два значения, в зависимости от того, будем покупать новое оборудование или нет.

y_j – возраст оборудования к началу j–го года (или к конце $j-1$ года)

$f_j(y_j)$ – максимальный доход на временном отрезке $j, \ldots, n$, при возрасте оборудования y_j лет в начале j–го года.

$C_{\text{Н}}$ – начальные затраты на приобретение и установку нового оборудования (цена нового оборудования).

Составляем уравнение Беллмана.

$$f_j(y_j) = \max_{x_j \in \{\text{С,Н}\}} \begin{cases} R(y_j) - C(y_j) + f_{j+1}(y_j + 1) & \text{при } x_j = \text{стар.}, \\ R(0) - C(0) - C_{\text{Н}} + f_{j+1}(1) & \text{при } x_j = \text{нов.} \end{cases}$$

Решение начинается с последнего 5 года пятилетки. Пусть это будет 1975 год. Так как в 1971 году оборудование было новым, то к началу 1975 года y_5 может принимать значения 1, 2, 3 или 4.

Этап 5:

$$f_5(y_5) = \max_{x_5 \in \{\text{С,Н}\}} \begin{cases} R(y_5) - C(y_5) & \text{при } x_5 = \text{стар.}, \\ R(0) - C(0) - C_{\text{Н}} = 20 & \text{при } x_5 = \text{нов.} \end{cases}$$

	$R(y_5) - C(y_5)$	$R(0) - C(0) - C_{\text{Н}}$ $=20$	Оптимальное решение	
y_5	x_5 = старое	x_5 = новое	$f_5(y_5)$	x_5^*
1	75-25 = 50	80-20-40 = 20	50	с
2	65-30 = 35	80-20-40 = 20	35	с
3	60-35 = 25	80-20-40 = 20	25	с
4	60-45 = 15	80-20-40 = 20	20	н

Таблица 34. Замена оборудования, этап 5

Этап 4:

$$f_4(y_4) = \max_{x_4 \in \{\text{С,Н}\}} \begin{cases} R(y_4) - C(y_4) + f_5(y_4 + 1) & \text{при } x_4 = \text{стар.}, \\ R(0) - C(0) - C_{\text{Н}} + f_5(1) & \text{при } x_4 = \text{нов.} \end{cases}$$

	$R(y_4) - C(y_4)+$ $+f_5(y_4+1)$	$R(0) - C(0) - C_{\text{Н}}+$ $+f_5(1)$	Оптимальное решение	
y_4	x_4 = старое	x_4 = новое	$f_4(y_4)$	x_4^*
1	75-25+35 = 85	20+50 = 70	85	с
2	65-30+25 = 60	20+50 = 70	70	н
3	60-35+20 = 45	20+50 = 70	70	н

Таблица 35. Замена оборудования, этап 4

Теперь приступим к этапу 3. В начале 1973 года возраст оборудования может принимать значения 1 и 2.

Этап 3:

$$f_3(y_3) = \max_{x_3 \in \{\text{С,Н}\}} \begin{cases} R(y_3) - C(y_3) + f_4(y_3 + 1) & \text{при } x_3 = \text{стар.}, \\ R(0) - C(0) - C_{\text{Н}} + f_4(1) & \text{при } x_3 = \text{нов.} \end{cases}$$

	$R(y_3) - C(y_3)+$ $+f_4(y_3+1)$	$R(0) - C(0) - C_{\text{Н}}+$ $+f_4(1)$	Оптимальное решение	
y_3	x_3 = старое	x_3 = новое	$f_3(y_3)$	x_3^*
1	75-25+70 = 120	20+85 = 105	120	с
2	65-30+70 = 105	20+85 = 105	105	с,н

Таблица 36. Замена оборудования, этап 3

Теперь приступим к последнему этапу 2. В начале 1972 года возраст оборудования может принимать значения только одно значение - 1 год.

Осталось из таблиц найти оптимальное решение. В начале 1971 года оборудование было новым. Из последней таблицы этапа 2 следует, что в начале 1972 года надо оборудование не менять, оставить старое. В конце 1972 года этому оборудованию будет 2 года, поэтому в таблице этапа 3 надо выбрать строку, соответствующую $y_3 = 2$. В этой строке записано, что оборудование можно оставить прежнее, а

Этап 2:

$$f_2(y_2) = \max_{x_2 \in \{\text{с},\text{н}\}} \begin{cases} R(y_2) - C(y_2) + f_3(y_2+1) & \text{при } x_2 = \text{стар.}, \\ R(0) - C(0) - C_{\text{Н}} + f_3(1) & \text{при } x_2 = \text{нов.} \end{cases}$$

	$R(y_2)-C(y_2)+$ $+f_3(y_2+1)$	$R(0)-C(0)-C_{\text{Н}}+$ $+f_3(1)$	Оптимальное решение	
y_2	x_2 = старое	x_2 = новое	$f_2(y_2)$	x_2^*
1	75-25+105 = 155	20+120 = 140	155	с

Таблица 37. Замена оборудования, этап 2

можно и поменять, доход от этого не изменится. Заменим оборудование, купим новое. До конца пятилетки будет пользоваться этим оборудованием. В конце 1975 года списываем его. Оптимальное решение

$$f_2(1) = 155, \quad (x_2^*, x_3^*, x_4^*, x_5^*) = (\text{с, н, с, с.})$$

Лучше учесть и доходы от нового оборудования в 1971 году. Они равны $R(0) - c(0) = 60$. Окончательно получаем оптимальное решение в более привычном виде:

$$f_1(0) = 215, \quad x^* = (x_1^*, \ldots, x_5^*) = (\text{н, с, н, с, с.})$$

На рисунке 6 изображено это оптимальное решение в более наглядном виде.

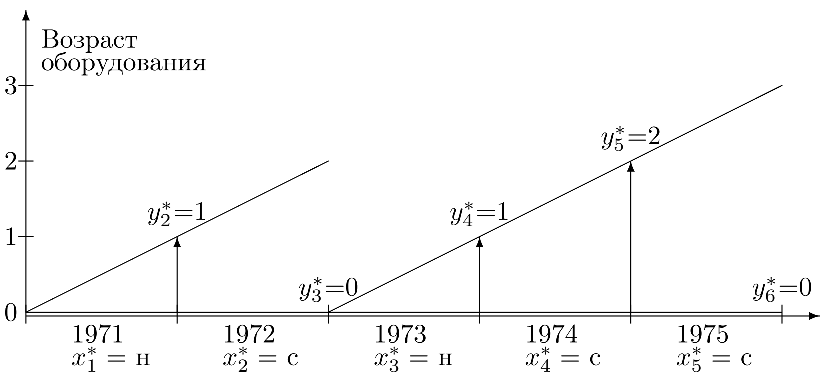

Рис. 6. Ответ в графическом виде для задачи о замене оборудования.

9.1 Замена оборудования на компьютере

Как обычно, приводятся программы для решения задачи методом динамического программирования и полным перебором. Из компьютерного решения видно, что у задачи есть еще одно решение

$$f_1(0) = 215, \quad x^* = (x_1^*, \ldots, x_5^*) = (\text{н, с, с, н, с.}).$$

Другими словами, можно оборудование поменять не только в начале 1973 года, но и в конце его.

```
type ZamenaD

% function [] = Zamena(r1,c1,Cn,y1,n)
% программа Zamena.m Динамическое программирование
% ==========Визгунов Н.П. 5 апреля 2006г.=========

format compact
% s :  1  2  3  4  5
r1 = [80 75 65 60 60]; % Доход от авто возраста s -1 лет
c1 = [20 25 30 35 45]; % Затраты на него
Cn = 40;               % Цена нового авто
```

```
y1 = 0;                 % В начале 1971 года авто - новый
n = 5;                  % Кол-во месЯцев в плановом периоде
if y1 + length(c1) > n
    error('не хватает данных')
end

Rnew = r1(1) - c1(1) - Cn;  % Прибыль от нового автомобилЯ
Fsu  = inf * ones(n ,2);
X = inf * ones(n ,n +1);
F = X;
F(:, n +1) = zeros(n, 1);
Tire = setstr(ones(1, 5* 6) * '-');
for j = n : -1 : 2
    for Yj = 1 : j -1   % s = Yj
        for Xj = 0 : 1  %   u = Xj +1
            if Xj == 0      % стараЯ машина
                Fsu(Yj,1) = r1(Yj +1) - c1(Yj +1) + ...
                                         F(Yj +1, j +1);
            else            % новаЯ Xj == 1
                Fsu(Yj,2) = Rnew + F(1, j +1);
            end % if
        end % for Xj
    end % for Yj
    [fs, index] = max(Fsu');
    F(1 :j -1, j) = fs(   1 : j -1)';
    X(1 :j -1, j) = index(1 : j -1)' -1;

    % Печать таблиц
    Sj = int2str(j);
    Kepka = ['    Y',Sj, '  X',Sj, '=ст. X', Sj, ...
                              '=нов.',' F    X* '];
    disp(Tire); disp(Kepka); disp(Tire);
    disp([(1 : j -1)', Fsu(1: j -1, :), fs(1: j -1)', ...
                                 index(1: j -1)'-1 ])
end % for j

% печать результатов
% ==================
x_opt = ones(n, 6);
Yj =1;
for j = 1 : n
    if X(Yj,j) == 0  % старый авто
        Yj = Yj +1;
        x_opt(j,:) = 'ста.  ';
    else                % новый авто
        Yj = 1;
        x_opt(j,:) = 'нов.  ';
    end % if
end % for

f_opt = F(1,2) + r1(1) - c1(1)
disp('x_opt =')
temp = x_opt';
disp(setstr(temp(:)'))

ZamenaD
------------------------------
    Y5  X5=ст. X5=нов. F    X*
------------------------------
     1     50     20     50     0
     2     35     20     35     0
     3     25     20     25     0
     4     15     20     20     1
------------------------------
```

```
    Y4  X4=ст. X4=нов. F    X*
------------------------------
     1    85    70    85     0
     2    60    70    70     1
     3    45    70    70     1
------------------------------
    Y3  X3=ст. X3=нов. F    X*
------------------------------
     1   120   105   120     0
     2   105   105   105     0
------------------------------
    Y2  X2=ст. X2=нов. F    X*
------------------------------
     1   155   140   155     0
f_opt =
   215
x_opt =
нов.  ста.  ста.  нов.  ста.
type ZamenaP

%     программа ZamenaP.m
% ====Визгунов Н.П. 5 апреля 2006г.======

format compact
% s :  1  2  3  4  5
r1 = [80 75 65 60 60]; % Доход от авто возраста s -1 лет
c1 = [20 25 30 35 45]; % Затраты на него
Cn = 40;               % Цена нового авто
y1 = 0;                % В начале 1971 года авто - новый

n = 5;                 % Кол-во месЯцев в плановом периоде
if y1 + length(c1) > n
    error('не хватает данных')
end
%Rnew = r1(1) - c1(1) - Cn;% Прибыль от нового автомобилЯ

f_opt = -inf;
x_opt = zeros(1, n +1);

x_max = ones(1, n);
x_min = zeros(1, n);

x = zeros(1, n);
j = n;
while 1
    if x(j) <= x_max(j)

       for j = 1 : n
            if  1 == x(j)
                y(j) = 0;
            elseif 0 == x(j) & j >= 2
                y(j) = y(j-1) +1;
            elseif 0 == x(1)
                y(1) = y1;
            end
        end % j

        f = Cn;
        for j = 1 : n
            if  0 == x(j) & y(j) >= 1
                f = f + r1(y(j) +1) - c1(y(j) +1);
            elseif 1 == x(j)
                f = f + r1(1) - c1(1) - Cn;
            end
```

```
        end

        if  f_opt < f
            f_opt = f
            x_opt = x
        elseif f_opt == f
            f_opt
            x_opt = [x_opt; x]
        end % elseif

        j = n;
    else    % x(j) > x_max(j)
        x(j) = x_min(j);
        j = j -1;
        if j <= 0.0001
            break
        end % if
    end % if
    x(j) = x(j) + 1;
end % while 1

disp('ответ')
f_opt, x_opt


ZamenaP
f_opt =
   165
x_opt =
     0     0     0     0     0
f_opt =
   170
x_opt =
     0     0     0     0     1
f_opt =
   195
x_opt =
     0     0     0     1     0
f_opt =
   195
x_opt =
     0     0     0     1     0
     0     0     1     0     0
f_opt =
   215
x_opt =
     1     0     0     1     0
f_opt =
   215
x_opt =
     1     0     0     1     0
     1     0     1     0     0
ответ
f_opt =
   215
x_opt =
     1     0     0     1     0
     1     0     1     0     0
diary off
```

Список литературы

[1] Акулич И.Л. *Математическое программирование в примерах и задачах: Учеб. пособие для студентов эконом. спец. вузов.* — М.: Высшая школа, 1986. — 319 с.

[2] Ануфриев И.Е. *Самоучитель MatLab 5.3/6.x.* — СПб.: БХВ-Петербург, 2002. — 736 с.

[3] Зайченко Ю.П. *Исследование операций: Учеб. пособие для студентов вузов.— 2-е изд. перераб. и доп.* — Киев.: Вища школа. Головное изд-во, 1979. — 392 с.

[4] *Исследование операций в экономике: Учебное пособие для вузов/ Н.Ш.Кремер, Б.А.Путко, И.М.Тришин, М.Н.Фридман; Под ред. проф. Н.Ш.Кремера.* — М.: Банки и биржи, ЮНИТИ, 1997. — 407 с.

[5] Потемкин В.Г. *Система MatLab. Справочное пособие* — М.: ДИАЛОГ–МИФИ, 1997. — 350 с.

[6] Таха Х. *Введение в исследование операций: В 2-х книгах. Кн.1. Пер. с англ.* — М.: Мир, 1985. — 479 с.

[7] Таха, Хэмди, А. *Введение в исследование операций, 6-е издание: Пер. с англ.* — М.: Издательский дом «Вильямс» , 2001. — 912 с.