МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра «Прикладная математика и информатика»
(наименование)

01.04.02 Прикладная математика и информатика
(код и наименование направления подготовки)

Математическое моделирование
(направленность (профиль))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

на тему «Исследование метода динамического программирования для решения задач оптимальной переработки сырья»

Обучающийся И.О. Коновалова
(Инициалы Фамилия) (личная подпись)

Научный руководитель к.ф.-м.н. О.В. Лелонд
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Оглавление

Введение

Актуальность и научная значимость исследования. В современном мире одной из важнейших научно-технических задач является оптимизация производств, выпускающих ту или иную продукцию. Для повышения эффективности, конкурентоспособности и оптимальности производственных процессов, необходимо разрабатывать технологические режимы, сводящие к минимуму потери и простои производственных линий. Научный подход к планированию и управлению производством, расчет всех этапов жизненного цикла продукции позволяет снизить экономические затраты на производство, обеспечить высокую доходность и рентабельность производства, увеличить конкурентоспособность конечного продукта, а также расширить ассортимент продукции, производимой на предприятии. Несмотря на очевидную важность указанной задачи, в научной литературе редко можно встретить модели оптимизации затрат и разработки алгоритмов принятия управленческих решений, рассчитанные для предприятий по переработке сырья.

Классическим способом теории управления дискретными процессами, применяемым в экономико-математическом моделировании, является метод динамического программирования. Применение данного способа подразумевает решение математических задач по планированию, возникающих в связи с процессами многоэтапного выбора. Многоэтапные процессы разбиваются на последовательные операции, при этом результат предыдущих операций можно использовать для управления ходом будущих операций.

Однако применение метода динамического программирования связано с определенными трудностями, заключающимися в сложности вычислительных процессов. Для повышения производительности и эффективности расчетов целесообразно использовать современные компьютерные способы вычислений и программные продукты.

Проблемы оптимизации затрат на переработку сырья, учитывающие различные ограничения, в научной литературе рассматривается недостаточно полно. Существуют различные методы и подходы к оптимизации производств, но в этом многообразии нет определенной модели, перспективной для предприятия по переработке сырья. Поэтому представляет интерес исследование метода динамического программирования для решения задач оптимальной переработки сырья. В этой связи необходимо исследование математических моделей и численных методов решения проблемы оптимальной переработки сырья, нахождение эффективного метода решения задачи оптимальной переработки сырья.

Исследованию отдельных аспектов проблем оптимизации и процессов принятия решений с использованием метода динамического программирования посвящены диссертационные работы и монографии, среди которых можно выделить работы Архипкиной А.И. [2], Баландина Д. В. [4], Баширзаде Л.И., Алиева Г. С. [5], Боровкова А.А., Савельева В. П. [9], Гавриловской С.П. [14], Гаджиева А.А., Сулеймановой О.Ш. [15], Геворгян М.В., Киракосян Г.Т. [16], Карпова Д.А., Струченкова В.И. [21], Нефедова Д.Г. [26], Рудычева А.А., Лева О.В. [33], Чернышева С.И. [37].

В научной литературе проблема применения метода динамического программирования для решения производственных задач является предметом исследования для таких авторов, как Акулич И.Л. [1], Беккенбах Э.Ф. [6], Беллман Р. [7], [8], Вентцель Е.С. [10], [11], Габасов Р., Кириллова Ф.М. [13], Грызина Н.Ю., Мастяева И.Н., Семенихина О.Н. [17], Калихман И.Л., Войтенко М.А. [20]. Труды этих заслуженных математиков составили теоретическую и практическую основу проведенного исследования.

Объектом исследования является производственный процесс переработки сырья.

Предметом исследования являются математические модели и численные методы решения задач оптимальной переработки сырья.

Цель исследования состоит в решении проблемы оптимизации затрат на переработку сырья, с учетом различных ограничений и особенностей производства.

Гипотеза исследования состоит в предположении, что применение метода динамического программирования существенно повышает эффективность управления производством по переработке сырья, снижает экономические затраты, обеспечивает конкурентоспособность и рентабельность производства. Однако применение метода динамического программирования связано с определенными трудностями, заключающимися в сложности вычислительных процессов. Для повышения производительности и эффективности расчетов целесообразна разработка программного обеспечения, реализующего решение данной задачи.

На основе обозначенной проблемы и сформулированной гипотезы, предлагается следующая тема магистерской диссертации: «Исследование метода динамического программирования для решения задач оптимальной переработки сырья».

Задачи исследования. Для достижения поставленной цели необходимо решить следующие задачи:

– изучение сущности метода динамического программирования;
– выявление особенностей метода динамического программирования;
– постановка задачи по оптимальному вложению инвестиций в модернизацию производства по переработке сырья;
– разработка математической модели задачи по оптимальному вложению инвестиций в модернизацию производства по переработке сырья;
– решение задачи по оптимальному вложению инвестиций в модернизацию производства по переработке сырья различными способами и методом динамического программирования;
– анализ требований к разрабатываемому программному обеспечению;
– анализ языков программирования, выбор наилучшего варианта;

– разработка кода программного обеспечения;
– анализ средств создания графического интерфейса пользователя, выбор наилучшего варианта. Разработка графического интерфейса пользователя программного обеспечения;
– тестирование работоспособности разработанного программного обеспечения.

Теоретической основой проводимого исследования являются научные труды российских и зарубежных ученых, разработки и исследования в сфере динамического программирования, математического моделирования, а также исследования по оптимальной переработке сырья.

В работе были применены следующие программные средства: программное обеспечение для работы с электронными таблицами Microsoft Excel, пакет математических вычислений SciLab, библиотека для создания графического интерфейса пользователя Tkinter. Для написания программного кода был использован язык программирования Python.

Методы исследования, применяемые во время написания работы: математическое моделирование, анализ и синтез модели, методы математической статистики, табличные и графические методы для предоставления результатов, методы цифровой обработки информации.

Научная новизна исследования заключается в разработке математической модели, перспективной для предприятия по переработке сырья, и проектировании программного обеспечения, с помощью которого возможно решение проблемы оптимизации затрат на переработку сырья, с учетом различных ограничений и особенностей производства.

Практическая значимость исследования заключается в возможности применения разработанного программного обеспечения на предприятии для решения вопросов по оптимальному распределению ресурсов. Это позволит обеспечить повышение эффективности принимаемых инвестиционных решений и снижение финансовых рисков.

Глава 1 Метод динамического программирования

1.1 Сущность метода динамического программирования

Динамическое программирование − один из самых мощных современных методов оптимизации. Специалисты различных профилей занимаются рациональным принятием решений, выбором оптимальных вариантов и оптимальным выполнением управленческих задач.

Метод динамического программирования занимает особое место в методе оптимизации. Этот метод очень привлекателен простотой и ясностью его основного принципа − принципа оптимальности. Сфера применения принципа оптимальности очень широка, а спектр задач, которые могут быть реализованы, еще не полностью определен. Динамическое программирование с самого начала выступает в качестве инструмента для практических решений проблем оптимизации.

Теория динамического программирования возникла из ряда технических и экономических проблем, включая наиболее эффективное использование оборудования и наиболее выгодную политику закупок. Новые и сложные математические задачи этого класса возникли в связи с многоэтапным процессом отбора. Рассматриваемая задача называется задачей планирования. Термин «динамическое планирование» был предложен, чтобы подчеркнуть роль времени и наличие в нем ряда вариантов. При правильной интерпретации эта динамическая перспектива способствует глубокому математическому пониманию проблемы.

Многошаговый процесс состоит из цикла операций, которые могут использовать результаты предыдущих операций, чтобы увидеть ход будущих операций. Одновременно можно выделить два вида операций: детерминированные операции, в которых результат полностью определен, и стохастические операции, в которых результат не определен, но может быть предсказан с использованием определенного распределения вероятностей.

Чтобы ознакомиться с этими математическими методами, необходимыми для решения сложной и уникальной новой математической задачи, рассмотрим простой пример, называемый «проблемой узких мест». Эта работа является серьезной в области промышленного производства. Она заключается в использовании промышленных комплексов для производства единого продукта.

Прежде чем приступить к математическим исследованиям, обобщим характерные особенности, присущие всем задачам динамического планирования:

– в любой момент времени t, состояние процесса определяется набором из нескольких параметров. Задачи, связанные с производственным процессом, требуют наличия производственного оборудования и инвентаря продукции, а также времени на производство;
– процесс отбора состоит в преобразовании этих параметров в параметры с другими числовыми значениями. В рассматриваемом примере процесс отбора состоит в перераспределении имеющихся запасов сырья и производственных мощностей различных секторов путем распределения более или менее основных ресурсов для каждой отрасли;
– предыдущие действия, выполняемые с системы не важны, при определении будущих действий. Данное качество можно получить, введя дополнительные параметры в описание процесса. Это поможет только в том случае, если необходимых параметров немного. Если имеется временная задержка, то целесообразнее использование функций вместо параметров.

Во всех рассматриваемых процессах последовательно осуществляются выборы наилучших стратегий. Самая благоприятная стратегия с учетом ранее определенных критериев, называется оптимальной стратегией.

Главное качество оптимальной стратегии: независимо от начального состояния и начального выбора, любой оставшийся выбор должен составлять оптимальную стратегию для ситуаций, возникающих в результате первоначального выбора [14], [35].

1.2 Применение метода динамического программирования в решении прикладных задач

1.2.1 Задачи на узкие места

Эти задачи появляются при изучении производственных процессов, связанных с различными промышленными комплексами. В задаче на узкие места уровень активности регулируется ресурсами, поставляемыми в минимальных количествах.

Успешная формулировка для этого типа задач является ключом к успеху. Все математические формулировки задачи обязательно являются приближенными, поэтому очень важно установить подход, позволяющий наиболее эффективно применять математический анализ.

Чтобы доказать, что решение действительно получено, применяется линейность задачи и создается двойственная ей задача. Двойственные задачи служат важным теоретическим инструментом, а также являются методом проверки.

Рассмотрим простую модель разветвленного на три отрасли производственного процесса, в котором основными отраслями служат автомобильная, сталелитейная и инструментальная промышленности.

Эти отрасли промышленности используются для производства как можно большего количества автомобилей за определенный период времени T.

Для любого заданного времени t, изначально принимающего дискретные значения 0, 1, 2, …, T, определяем:

- $x_1(t)$ – количество автомобилей, произведенных до времени T;
- $x_2(t)$ – производственная мощность автосборочного производства;
- $x_3(t)$ – запас материала;
- $x_4(t)$ – производственная мощность металлургического производства;

– $x_5(t)$ – запас инструментов и оснастки;

– $x_6(t)$ – производственная мощность станкостроительного производства.

Озвучим основные правила взаимодействия производств:

– для увеличения производительности автосборочного, металлургического и станкостроительного производств необходимы материалы и инструменты;

– для автосборочного производства необходимы производственные мощности и материалы;

– для металлургического производства необходимы только мощности по производству стали;

– для станкостроительного производства необходимы производственные мощности оснастки и материалы.

Предположим, что производство прямо пропорционально производственной мощности, содержащей достаточное количество сырья, и прямо пропорционально количеству выделяемого сырья с достаточным количеством соответствующих производственных мощностей.

Каждый единичный интервал начинается с распределения ресурсов для производства автомобилей, стали и инструментов с целью пополнения запасов автомобилей, стали и инструментов.

Введем обозначения:

– $z_i(t)$ – объём ресурсов стали, распределенный в момент t для увеличения $x_i(t)$;

– $w_i(t)$ – объём ресурсов инструментов, распределенный в момент t для увеличения $x_i(t)$.

Применяя правила, описанные ранее, можно записать:

$$z_3(t) = 0,\ w_1 = w_3 = w_5 = 0.$$

Таким образом, в зависимости от x_i, z_i и w_i в момент t, выводится формула для x_i в момент t+1:

$$\left.\begin{array}{c}
x_1(t+1) = x_1(t) + min[\gamma_1 x_2(t), \alpha_1 z_1(t), \beta_1 w_1(t)],\\
x_2(t+1) = x_2(t) + min[\alpha_2 z_2(t), \beta_2 w_2(t)],\\
x_3(t+1) = x_3(t) - z_1(t) - z_2(t) - z_4(t) - z_5(t) - z_6(t) + \gamma_3 x_4(t),\\
x_4(t+1) = x_4(t) + min[\alpha_4 z_4(t), \beta_4 w_4(t)],\\
x_5(t+1) = x_5(t) - w_2(t) - w_4(t) - w_6(t) + min[\gamma_5 x_6(t), \alpha_5 z_5(t)],\\
x_6(t+1) = x_6(t) + min[\alpha_6 z_6(t), \beta_6 w_6(t)].
\end{array}\right\} \qquad (1)$$

где α_i, β_i и γ_i – постоянные.

Ограничения, применяемые к z_i и w_i:

$$z_i \geq 0,$$

$$w_i \geq 0,$$

$$z_1 + z_2 + z_4 + z_5 + z_6 \leq x_3,$$

$$w_2 + w_4 + w_6 \leq x_5.$$

Из них следуют дополнительные ограничения:

$$\alpha_1 z_1 = \beta_1 w_1 \leq \gamma_1 x_2,$$

$$\alpha_2 z_2 = \beta_2 w_2,$$

$$\alpha_4 z_4 = \beta_4 w_4,$$

$$\alpha_5 z_5 = \gamma_5 x_6,$$

$$\alpha_6 z_6 = \beta_6 w_6.$$

Применив полученные выше ограничения в уравнениях (1), можно полностью исключить w_i и вывести следующие формулы:

$$\left.\begin{array}{c} x_1(t+1) = x_1(t) + \alpha_1 z_1(t), \\ x_2(t+1) = x_2(t) + \alpha_2 z_2(t), \\ x_3(t+1) = x_3(t) - z_1(t) - z_2(t) - z_4(t) - z_5(t) - z_6(t) + \gamma_3 x_4(t), \\ x_4(t+1) = x_4(t) + \alpha_4 z_4(t), \\ x_5(t+1) = x_5(t) - \varepsilon_2 z_2(t) - \varepsilon_4 z_4(t) - \varepsilon_6 z_6(t) + \alpha_5 z_5(t), \qquad \varepsilon_i = \dfrac{\alpha_i}{\beta_i}, \\ x_6(t+1) = x_6(t) + \alpha_6 z_6(t). \end{array}\right\} \quad (2)$$

Здесь для каждого t ограничения $z_i(t)$ становится следующим:

$$z_i \geq 0,$$

$$z_1 + z_2 + z_4 + z_5 + z_6 \leq x_3, \quad (3)$$

$$\varepsilon_2 z_2 + \varepsilon_4 z_4 + \varepsilon_6 z_6 \leq x_5, \quad (4)$$

$$z_1 \leq \mu_1 x_2,$$

$$z_5 = \mu_5 x_6, \qquad \mu_i = \frac{\gamma_i}{\alpha_i}.$$

Необходимо выбрать $z_i(t)$ для t = 0, 1, 2, …, T-1, учитывая выведенные выше ограничения таким образом, чтобы $x_1(T)$ был максимизирован.

В данной формулировке проблема находится в области линейного программирования и решается при любом наборе ограничений с помощью различных итеративных процессов.

Однако самый грубый расчет количества переменных показывает, что даже эта несложная трехотраслевая задача приводит к большому количеству независимых переменных z_i, на которые накладываются еще большее количество ограничений. Для определения оптимальной стратегии придется решать вычислительную задачу огромного объема.

Для нахождения оптимальной стратегии, которая не зависит от численных значений, необходимо кардинально изменить задачу.

1.2.1.1 Непрерывный вариант

В непрерывном варианте задачи распределение $z_i(t)$ во временном интервале [t, t+1] заменяется распределением $z_i(t)\Delta t$ во временном интервале [t, t+Δt]. Ключевым значением $z_i(t)$ является плотность дисперсии. Когда $\Delta t \to 0$, уравнения (2) принимают следующий вид:

$$\left.\begin{array}{c} \dot{x_1}(t) = \alpha_1 z_1(t), \\ \dot{x_2}(t) = \alpha_2 z_2(t), \\ \dot{x_3}(t) = -z_1(t) - z_2(t) - z_4(t) - z_5(t) - z_6(t) + \gamma_3 x_4(t), \\ \dot{x_4}(t) = \alpha_4 z_4(t), \\ \dot{x_5}(t) = -\varepsilon_2 z_2(t) - \varepsilon_4 z_4(t) - \varepsilon_6 z_6(t) + \alpha_5 z_5(t), \\ \dot{x_6}(t) = \alpha_6 z_6(t). \end{array}\right\}$$

Ограничения, применяемые к z_i, включают:

$$z_i \geq 0,$$

$$z_1 + z_2 + z_4 + z_5 + z_6 \leq \infty, \tag{5}$$

$$\varepsilon_2 z_2 + \varepsilon_4 z_4 + \varepsilon_6 z_6 \leq \infty, \tag{6}$$

$$z_1 \leq \mu_1 x_2,$$

$$z_5 \leq \mu_5 x_6.$$

Становится очевидным, что ограничения, представленные неравенствами (5) и (6), переходят в неравенства

$$x_3 \geq 0, \quad x_5 \geq 0. \tag{7}$$

Стоит отметить, что в частном варианте неравенств (7) вида

$$x_3 = 0, \quad x_5 \geq 0,$$

требуется ввести дополнительное ограничение:

$$z_1 + z_2 + z_4 + z_5 + z_6 \leq \gamma_3 x_4.$$

Соответственно, при

$$x_3 \geq 0, \quad x_5 = 0,$$

требуется ввести дополнительное ограничение:

$$\varepsilon_2 z_2 + \varepsilon_4 z_4 + \varepsilon_6 z_6 \leq \alpha_5 z_5.$$

1.2.1.2 Дополнительные пояснения обозначений

Для лучшего понимания теории, установим некоторые векторно-матричные обозначения. Пусть x(t) и z(t) являются n-мерными векторами-столбцами:

$$x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ . \\ . \\ . \\ x_n(t) \end{pmatrix}, \qquad z(t) = \begin{pmatrix} z_1(t) \\ z_2(t) \\ . \\ . \\ . \\ z_n(t) \end{pmatrix}.$$

Символы A_i и B_j представляют собой матрицу $m\ x\ n$.

Неотрицательный вектор x(t) означает, что все его элементы неотрицательны, то есть вектор $x_1 \geq 0$ можно записать как $x \geq 0$. Неравенство $x \geq y$ эквивалентно неравенству $x - y \geq 0$.

В этих обозначениях предыдущие уравнения и ограничения выглядят следующим образом:

$$\frac{dx}{dt} = A_1 x + A_2 z, \quad x(0) = c,$$

$$z \geq 0,$$

$$B_1 z \leq B_2 x.$$

Чтобы выразить функцию более понятным способом, установим скалярное произведение векторов:

$$(x, y) = \sum_{i=1}^{n} x_i y_i.$$

Выразим $x_1(T)$:

$$x_1(T) = (x(T), \alpha),$$

где вектор α имеет первую компоненту, равную 1, а последующие компоненты равные 0.

Далее отметим,

$$sup(x(T), \alpha) = f(c, T), \quad (8)$$

где f – скалярная функция вектора c и время T, т.е.:

$$f(c, T) = f(c_1, c_2, \dots, c_n; T).$$

Следует обратить внимание, что в выражении (8) максимальное значение может быть не достигнуто, поэтому применяется верхний предел (sup).

1.2.1.3 Основное функциональное уравнение

При написании выражения (8) учитывалось, что оптимальное значение x_1(T) является функцией начального состояния и продолжительности процесса.

Следует также отметить, что последующие состояния оптимального решения продолжение в любом конечном диапазоне, например [S, T], должно быть оптимальной стратегией для процесса T-S с периодом времени, начальным состоянием которого является c(S).

Базовое функциональное уравнение для указанной особенности:

$$f(c, S+T) = \max_{[0,S]} f[c(S), T], \tag{9}$$

где $\max\limits_{[0,S]}$ – максимизация для всех z(t), 0 ≤ t ≤ S, удовлетворяющих ограничениям.

Для доступности понимания предположим, что на данный момент максимум достигнут [1], [33].

1.2.1.4 Бесконечно малые аналогии

Чтобы получить максимальный результат от выражения (9), необходимо представить S бесконечно малым. Полученное нелинейное уравнение в частных производных эквивалентно выражению (9). Для этого необходимо принять некоторые предположения. Предположим, что c(S) = x(S) имеет кусочно-непрерывные производные, а f имеет те же свойства относительно компонент векторов c и T.

Если S = 0 является точкой непрерывности производных, то

$$\left.\begin{aligned} x(S) = c + [A_1 c + A_2 z(0)]S + o(S), \\ f(c, S+T) = f(c,T) + S\frac{df}{dT} + o(S), \\ f[c(S), T] = f\{c + [A_1 c + A_2 z(0)]S, T\} = \\ = f(c,T) + S\left(A_1 c + A_2 z(0), \frac{df}{dc}\right) + o(S), \end{aligned}\right\}, \tag{10}$$

$$\frac{df}{dc} = \begin{pmatrix} \dfrac{df}{dc_1} \\ \cdot \\ \cdot \\ \cdot \\ \dfrac{df}{dc_n} \end{pmatrix}.$$

Если подставить уравнение (10) в уравнение (9) и принять s → 0, то можно вывести дифференциальную форму уравнения (9):

$$\frac{df}{dT} = \max_{z(0)} \left[\left(A_1 c + A_2 z(0), \frac{df}{dc} \right) \right].$$

Здесь максимальное значение задается через n-мерную область z, определяемую ограничениями, возникающими при T=0.

Если в диапазоне [0, T] найдено решение для всех начальных состояний, то применяя бесконечно малую аналогию, можно найти решение в диапазоне [0, T+ΔT] [37].

1.2.1.5 Двойственная задача

Добавим к базовому рассматриваемому выражению

$$\frac{dx}{dt} = Az, \quad x(0) = c$$

дополнительные переменные и применить следующие ограничения

$$z \geq 0, \quad Bz \leq x.$$

Учитывая, что

$$x = c + \int_0^t Az\, dt,$$

ограничение можно записать следующим образом:

$$Bz + \int_0^t Cz\, dt \leq c\ ,$$

$$C = -A.$$

Задача максимизации (x(T), α) эквивалентна задаче максимизации уравнения:

$$\int_0^T (Az, \alpha)dt = \int_0^T (z, \acute{\alpha})dt.$$

Исходная задача может быть задана следующим образом: найти максимум $\int_0^T (z, \acute{\alpha})dt$ для всех z, удовлетворяющих ограничению

$$z \geq 0, \qquad Bz + \int_0^t Cz\, dt \leq c. \tag{11}$$

Примем неотрицательный вектор w(t), одинаковой размерности с c. Используя неравенство (11), выведем:

$$\int_0^T \left(w, Bz + \int_0^t Cz\, dt_1 \right) dt \leq \int_0^T (w, c)dt.$$

Введем B\` и C', выражающие транспонированные матрицы B и C соответственно:

$$(Bz, w) = \left(z, \acute{B}w\right).$$

Следующее равенство достигается путем частичной интеграции:

$$\int_0^T \left(w, \int_0^t Cz\, dt_1 \right) dt = \int_0^T \left(\int_t^T \acute{C}w\, dt_1, z \right) dt.$$

Объединив равенства, описанные выше, можно увидеть:

$$\int_0^T \left(w, Bz + \int_0^t Cz\, dt_1 \right) dt = \int_0^T \left(\acute{B}w + \int_t^T \acute{C}w\, dt_1, z \right) dt.$$

Пусть можно найти неотрицательный вектор w = w(t), удовлетворяющий неравенству:

$$\acute{B}w + \int_t^T \acute{C}w\, dt_1 \geq \acute{\alpha}. \tag{12}$$

В таком случае можно вывести набор уравнений и неравенств:

$$\begin{gathered} \int_0^T (w, c) dt \geq \int_t^T \left(w, Bz + \int_0^t Cz\, dt_1 \right) dt = \\ = \int_0^T \left(\acute{B}w + \int_t^T \acute{C}w\, dt, z \right) dt \geq \int_0^T (\alpha, z) dt. \end{gathered} \tag{13}$$

Следовательно,

$$min \int_0^T (w, c) dt \geq \max_z \int_0^T (z, \acute{\alpha}) dt. \tag{14}$$

Здесь берутся минимальные и максимальные значения для всех w и z, удовлетворяющие уравнениям (11) и (12). Следует обратить внимание, что если z имеет максимальное значение, это вовсе не значит, что w имеет минимальное значение.

Если максимумы или минимумы неравенства (14) равны, тогда применяется следующее зависимости:

$$\left.\begin{aligned} w_i = 0,\ \text{если}\ c_i > \left(Bz + \int_0^t Cz\,dt\right)_i, \\ z_i = 0,\ \text{если}\ \acute{\alpha}_j < \left(\acute{B}w + \int_t^T \acute{C}w\,dt\right)_j. \end{aligned}\right\} \tag{15}$$

Следует раскрыть важный факт, что любая неотрицательная пара z и w, удовлетворяющая выражению (15) и начальным ограничениям, обеспечивает решение проблемы максимальных и минимальных значений.

Данное утверждение доказывается следующим образом: если выполняется неравенство (15), то все зависимости в уравнении (13) преобразуются в равенства.

Если допустить, что некоторый вектор z удовлетворяет всем ограничениям, а также выполнимо следующее неравенство:

$$\int_0^T (z, \acute{\alpha})\,dt < \int_0^T (\bar{z}, \acute{\alpha})dt.$$

В итоге, для w, которое соответствует вектору z, верно выражение:

$$\int_0^T (\bar{z}, \acute{\alpha})dt \le \int_0^T \left(\bar{z}, \acute{B}w + \int_t^T \acute{A}w\, dt_1 \right) dt = \int_0^T \left(B\bar{z} + \int_0^t A\bar{z}\, dt_1, w \right) dt \le$$

$$\le \int_0^T (c, w)dt = \int_0^T (z, \acute{\alpha})dt.$$

В результате описана систематическая процедура проверки ожидаемого решения. Для z необходимо найти w, используя выражение (15), чтобы увидеть, удовлетворяет ли w заданному ограничению [26].

1.2.1.6 Более распространенные задачи

Многие математические задачи сводятся к поиску максимального значения интеграла

$$J(z) = \int_0^T F\left(x_1, x_{2,} \dots, x_n;\ z_1, z_{2,} \dots, z_m\right) dt,$$

когда x_i сопряжены с z следующими равенствами и ограничениями:

$$\frac{dx_i}{dt} = G_i(x, z),$$
$$i = 1, 2, \dots, n,$$
$$R_k(x, z) \le 0,$$
$$k = 1, 2, \dots, K.$$

Данные задачи могут быть решены методом функциональных уравнений. Это возможно, потому как определение собственных значений в задаче Штурма-Лиувилля

$$\begin{cases} u''(t) + \sigma\varphi(t)u(t) = 0, \\ u(0) = u(1) = 0 \end{cases}$$

эквивалентно определению относительного минимума интеграла $\int_0^1 (u'(t))^2\, dt$ при $\int_0^1 \varphi u^2(t)\, dt$ и эта проблема может быть решена аналогичным образом [23].

1.2.1.7 Задачи регулирования

Особую техническую и экономическую важность имеют задачи, основанные на регулировании систем в заданных временных интервалах. Для поддержания физической или экономической динамической системы в определенном состоянии или с незначительным отклонением от минимальной стоимости. Данная стоимость состоит из двух составляющих: первая составляющая определяется стоимостью отклонений системы от необходимого состояния, вторая составляющая – определяется стоимостью регулирующего инструмента.

Допустим предположение, что состояние системы представлено вектором x(t), удовлетворяющим линейному дифференциальному уравнению

$$\frac{dx}{dt} = Ax + f(t),$$
$$x(0) = c,$$

где A – константа;

f(t) – описывает управляющий вектор.

Введем обозначения:

- y(t) – состояние, к которому стремится система;
- F(x–y) – функционал, устанавливающий стоимость отклонения состояния x от y;
- G(f) – функционал, устанавливающий стоимость регулирующего инструмента.

Формула нахождения общей стоимости будет выглядеть следующим образом:

$$J(f, c) = F(x - y) + G(f).$$

При этом необходимо подобрать f таким образом, чтобы общая стоимость получилась минимальной [20], [43].

1.2.2 Задачи на оптимальное распределение ресурсов

1.2.2.1 Задача о закупках

В ситуациях, связанных с закупками, очень часто встречается проблема закупки оборудования для реализации конкретной работы. Обычно эта задача не проста и состоит из нескольких задач разного характера, так что оборудование, идеально подходящее для одной задачи, может оказаться посредственным для другой и совершенно бесполезным для третьей. Чтобы избежать подобной ситуации, часто приходится приобретать разные виды оборудования и комбинировать их для выполнения работы.

Общее описание задачи. Для того, чтобы выполнять определенную работу в течение N лет, в начале каждого года необходимо заказывать определенную единицу оборудования. Изначально имеются инвестиции x, которые делятся на две части:

- y – используется для покупки оборудования типа A;
- (x–y) – используется для покупки оборудования типа B.

Инвестиции y, потраченные на покупку оборудования типа A, принесут g(y) часов полезного труда. Инвестиции x–y, потраченные на покупку оборудования типа B, принесут h(x–y) часов полезного труда. В конце года оборудование типов A и B может быть сдано на металлолом, а на вырученные средства будет закуплено новое оборудование. В этом случае за оборудование типа A будет получено ay денежных единиц, где $0 < a < 1$. За оборудование типа B будет получено $b(x - y)$ денежных единиц, где $0 < b < 1$. Эта процедура производится ежегодно на протяжении N лет.

Необходимо определить оптимальное распределение инвестиций по годам, позволяющее получить максимальное количество часов полезного труда оборудования за период N лет.

Классическая формулировка задачи. Пусть $y_1, y_2, \dots, y_N$ – инвестиции, вложенные в покупку оборудования типа А в начале первого, второго, ..., N-го годов. Количество рабочих часов, отработанных на оборудовании за N лет находится по формуле:

$$J(y_1, y_2, \dots, y_N) = g(y_1) + h(x_1 - y_1) + g(y_2) + h(x_2 - y_2) + \cdots$$
$$\dots + g(y_N) + h(x_N - y_N),$$
$$0 \le y_i \le x_i, \tag{16}$$

$$\left.\begin{array}{c} x_1 = x, \\ x_2 = ay_1 + b(x_1 - y_1), \\ . \\ . \\ . \\ x_N = ay_{N-1} + b(x_{N-1} - y_{N-1}). \end{array}\right\} \tag{17}$$

Задача о закупках, описанная выше, эквивалентна задаче нахождения максимумов функции $J(y_1, y_2, \dots, y_N)$ в N-мерной области, определяемой по формуле (16). Дифференциальное и интегральное исчисления подходят для решения этой задачи, но их следует использовать с осторожностью, так как некоторые точки максимума могут находиться на границах области.

Если N невелико, то найти решение легко находится. Однако при N, выбранным случайным образом, например N = 10, задача анализа области N-мерного пространства, определяемой равенствами (17), становится очень время затратной.

Следует также отметить, что такой метод предоставляет слишком много информации: при заданных x и N, нет потребности знать $y_2, y_3, \dots, y_N$. Необходимо только y_1 в течение второго года. Поэтому, при определении y_1 как функции от x_1 и N, задача будет решена [3].

Постановка задачи с точки зрения динамического программирования. Учитывая функции g и h, не зависящих от времени, можно понять, что общее количество часов полезного труда, достигнутое за N лет работы оборудования, при оптимальной стратегии расходования средств, является функцией только от x (инвестированной суммы денег) и N (количества лет в промежутке времени). Количество часов полезного труда $f_N(x)$ за промежуток времени N при инвестициях x при применении оптимальной стратегии определяется следующим образом:

$$f_1(x) = \max_{0 \le y \le x} [g(y) + h(x-y)]. \tag{18}$$

Определим главное рекуррентное уравнение между $f_{N+1}(x)$ и $f_N(x)$. Пусть на первом шаге (N + 1)-шагового процесса сумма x разделяется на y и x–y. Число полезных рабочих часов, полученных за (N + 1) лет, будет равно $g(y) + h(x-y)$. Эта сумма складывается из часов, полученных в первый год и часов за оставшийся временной отрезок N при начальной сумме $ay + b(x-y)$.

Сумма $ay + b(x-y)$ должна использоваться оптимально в течение оставшихся N шагов, независимо от выбора начального y. Поэтому число часов, полученных в течение последних N шагов, будет равно $f_N[ay + b(x-y)]$. Следовательно, общая отдача при начальном разбиении на y и x–y будет равна

$$R_{N+1}(x,y) = g(y) + h(x-y) f_N[ay + b(x-y)].$$

Чтобы сделать общую отдачу максимальной, надо найти y, при котором $R_{N+1}(x,y)$ будет максимальным. Это подводит к следующей формуле:

$$\begin{aligned} f_{N+1}(x) &= \max_{0 \le y \le x} R_{N+1}(x,y) = \\ &= \max_{0 \le y \le x} \{g(y) + h(x-y) + f_N[ay + b(x-y)]\} \end{aligned} \tag{19}$$

для N = 1, 2, ...[13], [22].

1.2.2.2 Бесконечная аппроксимация

В задачах, где будущее играет незначительную роль, продуктивным способом нахождения решения является предположение, что N бесконечно. Эта математическая конструкция значительно упрощает задачу. При этом последовательность $\{f_N(x)\}$ заменяется следующей функцией:

$$f(x) = \lim_{N\to\infty} f_N(x).$$

Тогда вместо выражения (19) можно вывести следующее уравнение:

$$f(x) = \max_{0\le y\le x}\{g(y) + h(x-y) + f[ay + b(x-y)]\}. \quad (20)$$

После преобразования процесса в бесконечноэтапный, возникают новые проблемы, которых не было в конечном процессе. Теперь необходимо учитывать существование и единственность решений уравнения (20).

Очевидно, что если к решению уравнения (20) добавить любую константу, то будет получено новое решение. Чтобы зафиксировать функцию, заданную в исходной задаче, необходимо добавить условие $f(0) = 0$ [2], [7], [17].

1.2.2.3 Теорема существования и единственности

Для подробного описания вероятных способов решения уравнения (20), необходимо исследовать теорему существования и единственности.

Рассмотрим следующее уравнение:

$$f(x) = \max_{R}\{a(x_1, x_2, \dots, x_N) + f[b(x_1, x_2, \dots, x_N)]\}, \quad (21)$$

где область $R = R(x)$ определена соотношениями

$$x_k \ge 0,\ \ x = \sum_{k=1}^{N} x_k.$$

Если:

– функция $a(x_1, x_2, \dots, x_N)$ непрерывна и неотрицательна на R(x) для $0 \le x \le x_0$, причем $a(0, 0, \dots, 0) = 0$;

– функция $b(x_1, x_2, \dots, x_N)$ непрерывна и неотрицательна на R, причем на R(x_0)

$$b(x_1, x_2, \dots, x_N) \le c \sum_{k=1}^{N} x_k = cx, \quad 0 < c < 1;$$

– $\sum_{l=0}^{\infty} h(c^l x_0) < \infty$,

$$h(x) = \max_{0 \le y \le x} \left[\max_{R(y)} a(x_1, x_2, \dots, x_N) \right],$$

то для $0 \le x \le x_0$ существует непрерывное решение уравнения (21), для которого $f(0) = 0$.

Пусть $f(0)$ – значение, получаемое при выборе

$$x_1 = x, \qquad x_2 = x_3 = \cdots = x_n = 0,$$

где n = 2, 3, ... Тогда

$$f_0(x) = a(x) + a[b(x)] + \cdots,$$

где *a(x) = a(x, 0, ... , 0), b(x) = b(x, 0, ..., 0).*

Ряд справа мажорируется рядом

$$\sum_{l=0}^{\infty} h(c^l x_0)$$

и поэтому равномерно сходится.

Допустим

$$f_{n+1}(x) = \max_{R(x)}\{a(x_1, x_2, \dots, x_N) + f_n[b(x_1, x_2, \dots, x_N)]\}. \qquad (22)$$

Из определения f_0 вытекает, что $f_1 \geq 0$ и, следовательно, $f_{n+1} \geq f_n$. Пусть

$$M_n(x) = \max_{0 \leq y \leq x} f_n(y).$$

Тогда из выражения (22) имеем

$$M_{n+1}(x) \leq \mathrm{h(x)} + M_n(cx),$$

откуда

$$M_n(x) \leq \sum_{l=0}^{\infty} h(c^l x).$$

Поэтому $f_n(x)$ сходится к функции $f(x)$ для всех $x \in [0, x_0]$. Видно, что $f(x)$ удовлетворяет уравнению

$$f(x) = sup\{a(x_1, x_2, \dots, x_N) + f[b(x_1, x_2, \dots, x_N)]\} \qquad (23)$$

Чтобы определить наличие непрерывного решения и на основании этого заменить верхнюю границу sup максимумом в выражении (23), действия должны быть другими.

Пусть $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_N)$ – точка в R(x), в которой достигается максимум. Тогда

$$f_{n+1}(x) = \{a(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_N) + f_n[b(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_N)]\} \geq \\ \geq \{a(\bar{y}_1, \bar{y}_2, \dots, \bar{y}_N) + f_n[b(\bar{y}_1, \bar{y}_2, \dots, \bar{y}_N)]\}, \quad (24)$$

где $(\bar{y}_1, \bar{y}_2, \dots, \bar{y}_N)$ – любая другая точка в R(x).

Если за $(\bar{y}_1, \bar{y}_2, \dots, \bar{y}_N)$ принять точку, в которой выражение для n–1 принимает максимальное значение, то можно получить дополнительное условие:

$$f_n(x) = \{a(\bar{y}_1, \bar{y}_2, \dots, \bar{y}_N) + f_{n-1}[b(\bar{y}_1, \bar{y}_2, \dots, \bar{y}_N)]\} \geq \\ \geq \{a(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_N) + f_{n-1}[b(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_N)]\}. \quad (25)$$

Из выражений (24) и (25) следует неравенство:

$$|f_{n+1}(x) - f_n(x)| \leq max \begin{cases} |f_n[b(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_N)] - f_{n-1}[b(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_N)]| \\ |f_n[b(\bar{y}_1, \bar{y}_2, \dots, \bar{y}_N)] - f_{n-1}[b(\bar{y}_1, \bar{y}_2, \dots, \bar{y}_N)]| \end{cases}. \quad (26)$$

Допустим предположение

$$M_0(x) = \max_{0 \leq y \leq x} f_0(y),$$

$$M_{n+1}(x) = \max_{0 \leq y \leq x} |f_{n+1}(y) - f_n(y)|, \quad n \geq 0.$$

При предположении $b(x_1, x_2, \dots, x_N) \leq cx$ для всех $(x_1, x_2, \dots, x_N) \in R(x)$, из выражения (26) для n > 0 получаем

$$M_{n+1}(x) \leq M_n(\text{c}x).$$

Следовательно,

$$M_n(x) \leq M_0(\text{c}^n x) = h(\text{c}^n x).$$

По предположению, если ряд

$$\sum_{n=0}^{\infty} h(c^n x_0)$$

сходится, то ряд

$$\sum_{n=0}^{\infty} [f_{n+1}(x) - f_n(x)]$$

сходится равномерно для $0 \le x \le x_0$. Следовательно,

$$f(x) = \lim_{n \to \infty} f_n(x)$$

является непрерывной функцией на [0, x_0]. Поэтому *f(x)* – это решение уравнения (21).

Поскольку *M(x)* непрерывна и *M(0) = 0*, получаем *M(x) ≤ 0.* Это означает, что *M(x)* равна нулю [6], [11].

1.2.2.4 Аналитические результаты

Рассмотрим уравнение (20). Если g(x) и h(x) – строго выпуклые функции от x, то оптимальная стратегия требует, чтобы либо *y = 0*, либо *y = x*.

Ситуация усложняется, если функции g и h вогнуты обе. Если

$$g(0) = h(0) = 0;$$
$$g'(x) \ge 0, \qquad h'(x) \ge 0 \text{ для } x \ge 0;$$
$$g''(x) \le 0, \qquad h''(x) \le 0 \text{ для } x \ge 0.$$

Подвергнем обсуждению следующие выражения:

$$f_1(x) = \max_{0 \le y \le x} [g(y) + h(x-y)],$$

$$f_{n+1}(x) = \max_{0\le y\le x}\{g(y) + h(x-y) + f_n[ay + b(x-y)]\},$$
$$n = 1, 2, \dots .$$

Для каждого n существует единственное $y_n = y_n(x)$, позволяющее получить максимальное значение. Если b < a, то $y_1 \le y_2 \le y_3 \le \cdots$; если b > a, то выполняются обратные неравенства. Если b < a и $y_n(x) = x$ для некоторого n, то $y_m(x) = x$ для $m \ge n$.

Однако, если g и h выпуклы и либо *y = 0*, либо *y = x*, сложно определить, какая величина y верна. Поэтому следующий результат полезен для приближенных вычислений:

$$\mathrm{F(x)} = \max_{0\le y\le x}\left[cx^d + F(ax),\ \ ex^f + F(bx)\right]$$

определяется выражением

$$y = \begin{cases} x \text{ при } 0 \le x \le x_0, \\ 0 \text{ при } x_0 \le x, \end{cases}$$

$$x_0 = \left[\frac{\dfrac{c}{(1-a^d)}}{\dfrac{e}{(1-b^d)}}\right]^{\frac{1}{(f-d)}}.$$

1.2.2.5 Методика вычислений

При создании математической модели физической задачи возникает конфликт между необходимостью точно описать реальные физические явления и упрощением математической модели для легкости анализа. Поэтому чем больше точность реальности, тем меньше вероятность того, что задача может быть решена только аналитическими методами.

Поэтому очевидно, что использование математики влечет за собой две задачи:

– найти единую методологию для решения небольших задач;
– обеспечить обширную базу для приближенного решения крупномасштабных задач, включая инструменты для повышения точности приближения ценой дополнительных затрат времени и сил.

Отсюда следует вывод, что главная цель изучения простой модели со свойствами, характерными исходнику, – найти отправную точку для дальнейших исследований более сложных моделей.

При исследовании задач динамического программирования, используются методы аппроксимации, которые не применимы при исследовании уравнений функционального анализа. Это связано с двойственностью, которая существует между решением функционального уравнения и решением задач оптимизации.

Возвращаясь к выражению (20), мы видим, что для заданной функции $f(x)$ можно найти все максимизирующие y. И наоборот, если мы знаем, что y является функцией от x, мы можем найти $f(x)$ используя метод итерации. Следовательно, функция определяют все оптимальные стратегии, а оптимальные стратегии определяют функция [5], [10], [24], [42].

Выводы по главе 1

В главе 1 были проведены теоретические исследования поставленной темы, рассмотрена общая сущность метода динамического программирования, описаны задачи и способы их решения с применением принципов динамического программирования.

Глава 2 Постановка задачи оптимального распределения ресурсов

2.1 Общая постановка задачи оптимального распределения ресурсов

Исследуем метод динамического программирования для решения задач оптимального распределения инвестиций в модернизацию производства по сбору и переработке вторичного сырья.

Рассмотрим задачу по оптимальному вложению инвестиций в модернизацию производства по переработке сырья.

Наше предприятие имеет четыре производственные площадки, которые оснащены различным оборудованием и имеют различные технические возможности по переработке вторичного сырья. Производительность производственных площадок зависит от многих факторов, в числе которых:

– характеристики имеющегося на площадке оборудования (режимы работы, объемы загрузки, время на обслуживание);
– режим работы персонала;
– наличие промышленных шредеров для измельчения бумажного сырья;
– характеристики автомобильных весов, позволяющих взвешивать автомобили определенной длины;
– объемы складов или мест складирования для вторсырья;
– объемы складов или мест складирования для переработанной продукции.

Компания планирует модернизацию своих производственных площадок и рассматривает проекты по приемке и переработке 3 видов вторичного сырья:

– макулатура – книги, картон, бумага, бумажная и картонная упаковка, шпагат;
– полимеры – пластмассовая тара, пластмассовые изделия;

– пленка – стрейчпленка, полиэтиленовая упаковка, пакеты.

В реализацию проектов планируется вложить 10 млн. руб. Необходимо выбрать такой проект по переработке вторичного сырья для каждой производственной площадки компании, чтобы обеспечить максимальную прибыль в годовом выражении. При этом на каждую производственную площадку можно завозить только один вид сырья.

Оформим исходные данные в виде таблицы 1.

Таблица 1 – Исходные данные

Проект (вид сырья)	Производственная площадка №1		Производственная площадка №2		Производственная площадка №3		Производственная площадка №4	
	I_1	P_1	I_2	P_2	I_3	P_3	I_4	P_4
x_1 – макулатура	2	3	2	4	1	3	1	1
x_2 – полимеры	3	6	4	7	2	5	3	4
x_3 – пленка	–	–	–	–	5	7	–	–

2.2 Построение математической модели поставленной задачи

Введем обозначения:

– j – производственная площадка;

– x_j – номер проекта, который будет реализовываться на производственной площадке j;

– I_j – (от англ. invest) стоимость модернизации производственной площадки j, при реализации проекта x_j, млн. руб;

– P_j – годовая прибыль (от англ. profit), полученная в результате модернизации производственной площадки j, млн. руб/год;

– n – количество производственных площадок (в нашей задаче n = 4);

– C_j – (от англ. capital) инвестиции, выделенные на модернизацию производственных площадок j, j+1, ..., n;

– $F_1(C_1)$ – максимальная годовая прибыль, которая будет получена от реализации проектов $x_1, x_2, ..., x_n$ при заданном объеме инвестиций C_1, млн. руб.

Составим математическую формулировку поставленной задачи:

$$\left.\begin{array}{c} F_1(\mathrm{C}_1) = \max\limits_{x_1,\dots,x_n}\left\{\sum\limits_{j=1}^{n} \mathrm{P_j}(\mathrm{x_j})\right\}, \\ \sum\limits_{j=1}^{n} \mathrm{I_j}(\mathrm{x_j}) \le \mathrm{C}_1\,, \\ \mathrm{j} = 1, \dots, \mathrm{n} \end{array}\right\}$$

Задача сводится к нахождению функции $F_1(\mathrm{C}_1) = \max\limits_{x_1,\dots,x_n}\left\{\sum_{j=1}^{n} \mathrm{P_j}(\mathrm{x_j})\right\}$ при ограничении $\sum_{j=1}^{n} \mathrm{I_j}(\mathrm{x_j}) \le \mathrm{C}_1$ и при $\mathrm{j} = 1, \dots, \mathrm{n}$.

Выберем фиксированное значение $\mathrm{x}_1 = \tilde{\mathrm{x}}_1$, а также введем новое обозначение:

$\tilde{F}_1(\mathrm{C}_1)$ – наибольшая годовая прибыль, при вложении инвестиций C_1 и при фиксированном значении $\mathrm{x}_1 = \tilde{\mathrm{x}}_1$.

Расчет начнем с $\tilde{\mathrm{x}}_1 = 1$:

$$\left.\begin{array}{c} \tilde{F}_1(\mathrm{C}_1) = \mathrm{P}_1(\tilde{\mathrm{x}}_1) + \max\limits_{x_2,\dots x_n}\left\{\sum\limits_{j=2}^{n} \mathrm{P_j}(\mathrm{x_j})\right\}, \\ \mathrm{I}_1(\tilde{\mathrm{x}}_1) + \sum\limits_{j=2}^{n} \mathrm{I_j}(\mathrm{x_j}) \le \mathrm{C}_1, \\ \mathrm{j} = 2, \dots, \mathrm{n} \end{array}\right\}$$

Сократим запись:

$$\tilde{F}_1(\mathrm{C}_1) = \mathrm{P}_1(\tilde{\mathrm{x}}_1) + F_2(\mathrm{C}_1 - \mathrm{I}_1(\tilde{\mathrm{x}}_1)),$$

где $F_2(\mathrm{C}_1 - \mathrm{I}_1(\tilde{\mathrm{x}}_1))$ – это годовая прибыль от вложения $\mathrm{C}_1 - \mathrm{I}_1(\tilde{\mathrm{x}}_1)$

млн. руб. в модернизацию предприятий $2, \dots, \mathrm{n}$.

Переберем все возможные значения $\tilde{\mathrm{x}}_1$.

$$F_1(\mathrm{C}_1) = \max_{\tilde{\mathrm{x}}_1}\{\mathrm{P}_1(\tilde{\mathrm{x}}_1) + F_2(\mathrm{C}_1 - \mathrm{I}_1(\tilde{\mathrm{x}}_1))\}.$$

Не будем писать тильду в обозначении переменной x_1.

$C_2 = C_1 - I_1(x_1)$ – количество денежных средств, инвестированных в производственные площадки 2, …, n.

Так как $C_2 \geq 0$, то $I_1(x_1) \leq C_1$. В итоге получаем формулу для вычисления $F_1(\mathrm{C}_1)$:

$$F_1(\mathrm{C}_1) = \max_{x_1 | I_1(x_1) \leq C_1}\{\mathrm{P}_1(x_1) + F_2(\mathrm{C}_1 - \mathrm{I}_1(x_1))\}.$$

Аналогично выведем формулы для последующих вычислений:

$$F_2(\mathrm{C}_2) = \max_{x_2 | I_2(x_2) \leq C_2}\left\{\mathrm{P}_2(x_2) + F_3\left(\mathrm{C}_2 - \mathrm{I}_2(x_2)\right)\right\},$$

$$\mathrm{C}_2 \in \{0, \dots, 8\},$$

где $F_3(C_3)$ – доходы для (n – 2) производственных площадок;

$C_3 = C_2 - I_2(x_2)$ – количество денежных средств, инвестированных в производственные площадки 3, …, n.

$$F_3(\mathrm{C}_3) = \max_{x_3 | I_3(x_3) \leq C_3}\left\{\mathrm{P}_3(x_3) + F_4\left(\mathrm{C}_3 - \mathrm{I}_3(x_3)\right)\right\},$$

$$F_4(\mathrm{C}_4) = \max_{x_4 | I_4(x_4) \leq C_4}\{\mathrm{P}_4(x_4)\}.$$

Общий вид рекуррентного уравнения Беллмана для процедуры обратной прогонки:

$$F_{n+1}(C_{n+1}) = 0,$$

$$F_j(\mathrm{C_j}) = \max_{x_j | I_j(x_j) \le C_j} \left\{ \mathrm{P_j}(x_j) + F_{j+1}\left(\mathrm{C_j} - \mathrm{I_j}(x_j)\right) \right\},$$

$$j = n, n-1, \dots, 1.$$

Рекуррентное уравнение Беллмана для процедуры обратной прогонки для нашей задачи будет иметь вид [9], [29]:

$$F_5(C_5) = 0,$$

$$F_j(\mathrm{C_j}) = \max_{x_j | I_j(x_j) \le C_j} \left\{ \mathrm{P_j}(x_j) + F_{j+1}\left(\mathrm{C_j} - \mathrm{I_j}(x_j)\right) \right\},$$

$$j = 4, 3, 2, 1.$$

2.3 Решение задачи оптимального распределения ресурсов с применением метода динамического программирования

Применяя метод динамического программирования, поставленную задачу можно решить следующими способами:

- способ полного перебора;
- графический способ;
- табличный способ;
- решение при помощи программного обеспечения для математических вычислений;
- реализация при помощи программного обеспечения [28].

2.3.1 Решение задачи способом полного перебора

Способ полного перебора самый простой, но трудоемкий, если считать вручную. Рассматриваемая задача имеет $2 \times 2 \times 3 \times 2 = 24$ варианта возможных решений. Не все варианты приемлемы, так как могут требовать вложения больших инвестиций, чем планируется вложить по условию задачи.

Для расчетов используем программное обеспечение Microsoft Excel. Создадим две таблицы. Таблица 2 содержит исходные данные, в таблице 3 происходит расчет методом полного перебора с использованием формул.

Таблица 2 – Исходные данные, введенные в документ Microsoft Excel

	A	B	C	D	E	F	G	H	I
1	Проект	Производственная площадка №1		Производственная площадка №2		Производственная площадка №3		Производственная площадка №4	
2		I_1	P_1	I_2	P_2	I_3	P_3	I_4	P_4
3	макулатура	2	3	2	4	1	3	1	1
4	полимеры	3	6	4	7	2	5	3	4
5	пленка	0	0	0	0	5	7	0	0

Таблица 3 – Расчет методом полного перебора в документе Microsoft Excel

	A	B	C	D	E	F	G
7	x_1	x_2	x_3	x_4	Затраты $\sum I_j(x_j)$	Прибыль $\sum P_j(x_j)$	Вариант приемлем?
8	1	1	1	1	6	11	Да
9	1	1	1	2	8	14	Да
10	1	1	2	1	7	13	Да
11	1	1	2	2	9	16	Да
12	1	1	3	1	10	15	Да
13	1	1	3	2	12	18	Нет
14	1	2	1	1	8	14	Да
15	1	2	1	2	10	17	Да
16	1	2	2	1	9	16	Да
17	1	2	2	2	11	19	Нет
18	1	2	3	1	12	18	Нет
19	1	2	3	2	14	21	Нет
20	2	1	1	1	7	14	Да
21	2	1	1	2	9	17	Да
22	2	1	2	1	8	16	Да
23	2	1	2	2	10	19	Да
24	2	1	3	1	11	18	Нет
25	2	1	3	2	13	21	Нет
26	2	2	1	1	9	17	Да
27	2	2	1	2	11	20	Нет
28	2	2	2	1	10	19	Да
29	2	2	2	2	12	22	Нет
30	2	2	3	1	13	21	Нет
31	2	2	3	2	15	24	Нет

Формула для вычисления затрат $\sum I_j(x_j)$ в столбце E:
=(ЕСЛИ(B8=1;B3;B4))+(ЕСЛИ(C8=1;D3;D4))+
+(ЕСЛИ(D8=1;F3;ЕСЛИ(D8=2;F4;F5)))+(ЕСЛИ(E8=1;H3;H4)).

Формула для вычисления прибыли $\sum P_j(x_j)$ в столбце F:
=(ЕСЛИ(B8=1;C3;C4))+(ЕСЛИ(C8=1;E3;E4))+
+(ЕСЛИ(D8=1;G3;ЕСЛИ(D8=2;G4;G5)))+(ЕСЛИ(E8=1;I3;I4)).

Формула для определения приемлемости варианта в столбце G:
=ЕСЛИ(F8<=10;"Да";"Нет") [39].

Из приемлемых вариантов определим комбинации проектов, при реализации которых оптимальным образом будут вложены имеющиеся у предприятия 10 млн. руб.

Таким образом, после вычисления задачи способом полного перебора, были найдены два оптимальных решения:

– $x_{(1)}$ = (2, 1, 2, 2), F(10) = 19,

– $x_{(2)}$ = (2, 2, 2, 1), F(10) = 19.

2.3.2 Решение задачи графическим способом

Расчет начинается с определения диапазонов допустимых стоимостей модернизаций для производственных площадок.

Для производственной площадки №4, проект требующий наименьших вложений стоит 1 млн. руб., поэтому $C_4 \geq 1$. Для площадок №1, №2 и №3 также должны быть предусмотрены самые малобюджетные проекты, поэтому верхняя граница C_4 не 10 млн. руб., а меньше на $I_1(1) + I_2(1) + I_3(1) = 2 + 2 + 1 = 5$ млн. руб. Таким образом, в модернизацию производственной площадки 4 можно вложить денежные средства в диапазоне $C_4 \in 1 : 5$.

На производственных площадках №3 и №4 нижняя граница вложений на модернизацию равна $C_3 = I_3(1) + I_4(1) = 1 + 1 = 2$ млн. руб. Верхняя граница вложений равна $10 - I_1(1) - I_2(1) = 10 - 2 - 2 = 6.$ Таким образом, в модернизацию производственных площадок №3 и №4 можно вложить денежные средства в диапазоне $C_3 \in 2 : 6$.

На производственных площадках №2, №3 и №4 нижняя граница вложений на модернизацию равна $С_2 = I_2(1) + I_3(1) + I_4(1) = 2 + 1 + 1 = 4$ млн. руб. Верхняя граница вложений равна $10 - I_1(1) = 10 - 2 = 8$. Таким образом, в модернизацию производственных площадок №2, №3 и №4 можно вложить денежные средства в диапазоне $C_2 \in 4 : 8$.

Для производственной площадки №1 будет применена величина $С_1 = 10$ млн. руб.

Рассчитанные данные внесены в таблицу 4.

Таблица 4 – Вычисление диапазонов допустимых стоимостей модернизаций $С_j$

Проект	Производственная площадка №1		Производственная площадка №2		Производственная площадка №3		Производственная площадка №4	
	I_1	P_1	I_2	P_2	I_3	P_3	I_4	P_4
$x_j = 1$	2	3	2	4	1	3	1	1
$x_j = 2$	3	6	4	7	2	5	3	4
$x_j = 3$	–	–	–	–	5	7	–	–
	$С_1 = 10$							
			$C_2 \in 4 : 8$					
					$C_3 \in 2 : 6$			
							$C_4 \in 1 : 5$	

Начнем с решения задачи для производственной площадки №4. Затем будем решать задачу для производственных площадок №3 и №4, затем для площадок №2, №3 и №4. Завершающим шагом будет решение задачи для площадок №1, №2, №3, №4.

Результаты вычислений будут заноситься на графы следующим образом. В вершины графа вписаны инвестиции $С_j$, выделенные на модернизацию производственных площадок из диапазонов, рассчитанных в таблице 4. Первое число над вершиной – это максимально возможный доход от освоения инвестиций (записанных в вершине). Второе число над вершиной – номер проекта, который необходимо реализовать для получения максимально возможного дохода. Вершины, для которых верно равенство

$C_j - I_j(x_j) = C_{j+1}$, соединяются дугами. Длины дуг равны доходам от реализации соответствующих проектов.

Шаг 4. На шаге 4 решается задача для производственной площадки №4. Инвестиции для модернизации площадки №4 распределяются следующим образом: если средств достаточно для реализации дорогого проекта, т.е. $C_4 \geq 3$, то реализуется проект 2. Если средств недостаточно для реализации проекта 2, т.е. $C_4 < 3$, то реализуется проект 1. Вершина $C_5 = 0$ несуществующая, так как производственная площадка №5 в задаче не предусмотрена. На граф данная вершина нанесена для более логичного объяснения процедуры решения задачи. Все вершины C_4 соединяются дугой с вершиной $C_5 = 0$. В соответствии с выбранным проектом, дуги будет иметь длину $P_4(1) = 1$ или $P_4(2) = 4$. Доход от реализации проектов также будет равен $P_4(1) = 1$ млн. руб. или $P_4(2) = 4$ млн. руб., в зависимости от выбранного проекта.

Шаг 4 решения задачи графическим способом показан на рисунке 1.

Шаг 3. На шаге 3 решается задача для производственных площадок №3 и №4.

Рассмотрим вершину $C_3 = 6$. Необходимо найти оптимальный вариант освоения 6 млн. руб., выделенных для модернизации производственных площадок №3 и №4. При реализации проекта 1 на площадке №3, для площадки №4 останется $C_3 - I_3(1) = 6 - 1 = 5$ млн. руб. Соединяются дугой вершины $C_3 = 6$ и $C_4 = 1$, при этом дуга будет иметь длину $P_3(1) = 3$. Доход от реализации проектов будет равен $3 + 4 = 7$ млн. руб.

При реализации проекта 2 на площадке №3, для площадки №4 останется $C_3 - I_3(2) = 6 - 2 = 4$ млн. руб. Соединяются дугой вершины $C_3 = 6$ и $C_4 = 4$, при этом дуга будет иметь длину $P_3(2) = 5$. Доход от реализации проектов будет равен $5 + 4 = 9$ млн. руб.

При реализации проекта 3 на площадке №3, для площадки №4 останется $C_3 - I_3(3) = 6 - 5 = 1$ млн. руб. Соединяются дугой вершины $C_3 = 6$ и $C_4 = 1$, при этом дуга будет иметь длину $P_3(3) = 7$. Доход от реализации проектов будет равен 7 + 1 = 8 млн. руб.

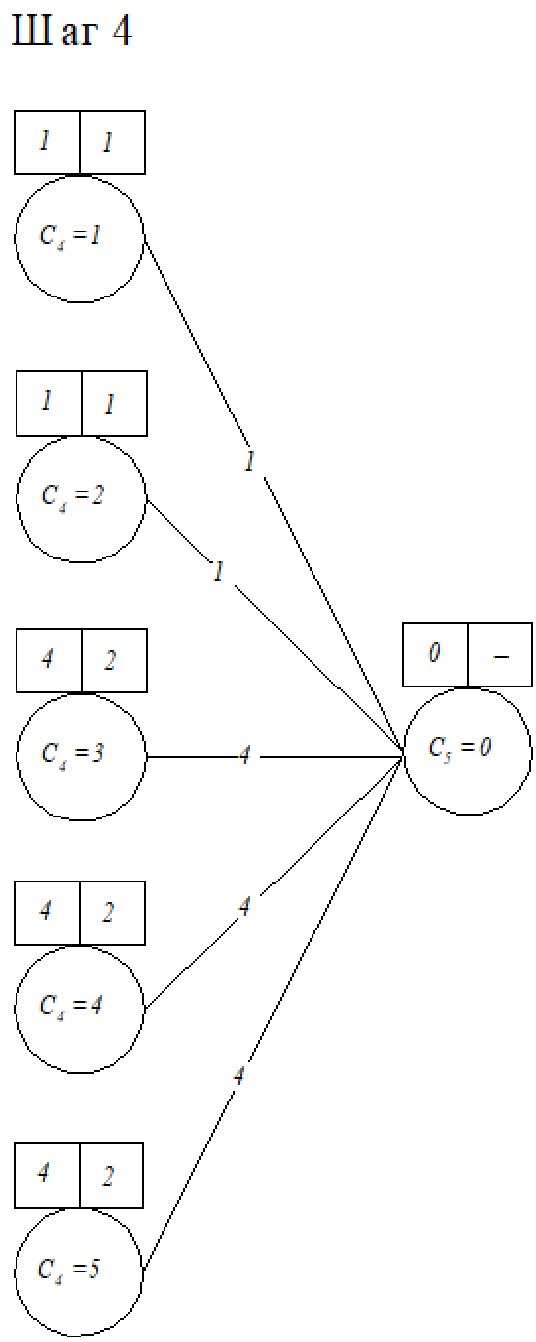


Рисунок 1 – Шаг 4 решения задачи графическим способом

Из расчетов видно, что максимальный доход 9 млн. руб. при вложении $C_3 = 6$ млн. руб. может быть получен от реализации проекта 2. Цифры 9 и 2 записываются над вершиной $C_3 = 6$.

Расчет для остальных вершин шага 3 происходит аналогично.

Шаг 3 решения задачи графическим способом показан на рисунке 2.

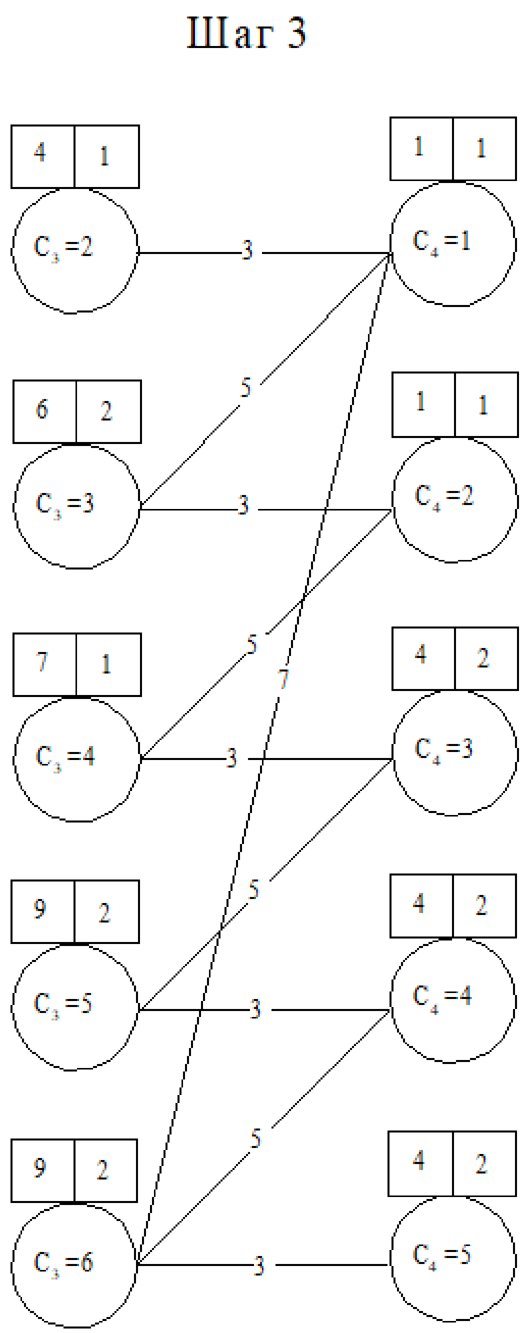


Рисунок 2 – Шаг 3 решения задачи графическим способом

Шаг 2. На шаге 2 решается задача для производственных площадок №2, №3 и №4. Расчет ведется аналогично расчету на шаге 3.

Шаг 2 решения задачи графическим способом показан на рисунке 3.

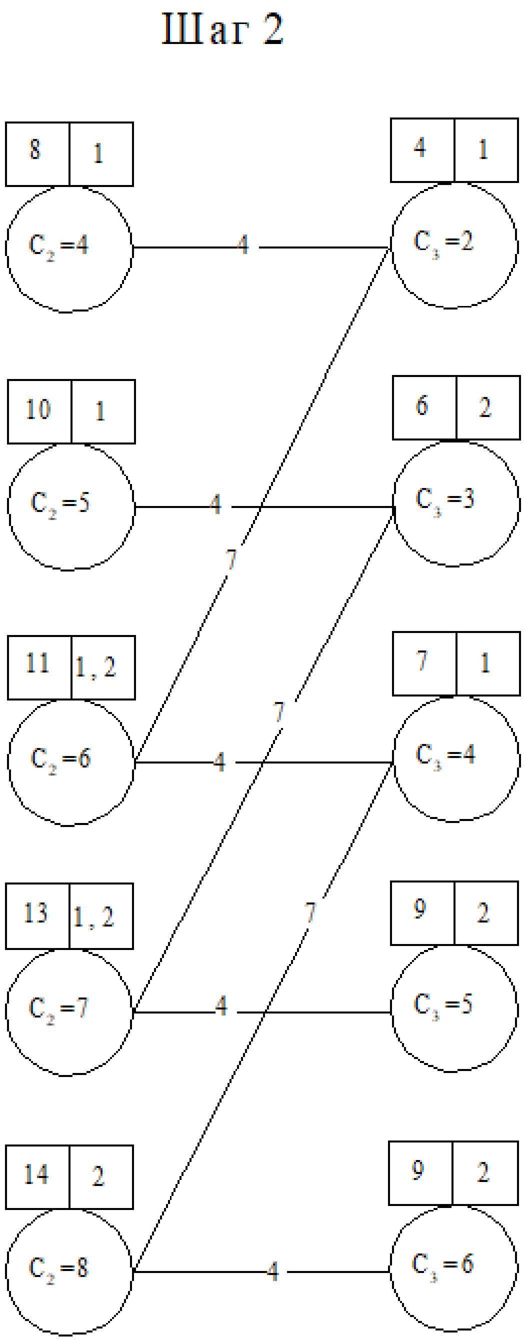


Рисунок 3 – Шаг 2 решения задачи графическим способом

Шаг 1. На шаге 1 решается задача для производственных площадок №1, №2, №3 и №4.

$C_1 - C_2 = 10 - 8 = 2$ млн. руб. Если средств достаточно для реализации дорогого проекта, т.е. $C_1 \geq 3$, то реализуется проект 2. Если средств недостаточно для реализации проекта 2, т.е. $C_1 < 3$, то реализуется проект 1. Все вершины C_2 соединяются дугой с вершиной $C_1 = 10$. В соответствии с выбранным проектом, дуги будет иметь длину $P_1(1) = 3$ или $P_1(2) = 6$. Из расчетов видно, что максимальный доход 19 млн. руб. при вложении $C_1 = 10$ млн. руб. может быть получен от реализации проекта 2 на площадке №1.

Шаг 1 решения задачи графическим способом показан на рисунке 4.

На рисунке 5 показан граф, получившийся в результате решения задачи графическим способом.

Определим комбинации проектов, при реализации которых оптимальным образом будут вложены имеющиеся у предприятия 10 млн. руб.

Таким образом, после вычисления задачи графическим способом, были найдены два оптимальных решения:

– $x_{(1)} = (2, 1, 2, 2)$, $F(10) = 19$,

– $x_{(2)} = (2, 2, 2, 1)$, $F(10) = 19$.

Эти оптимальные решения совпадают с решениями, найденными ранее способом полного перебора [8], [15], [21].

Шаг 1

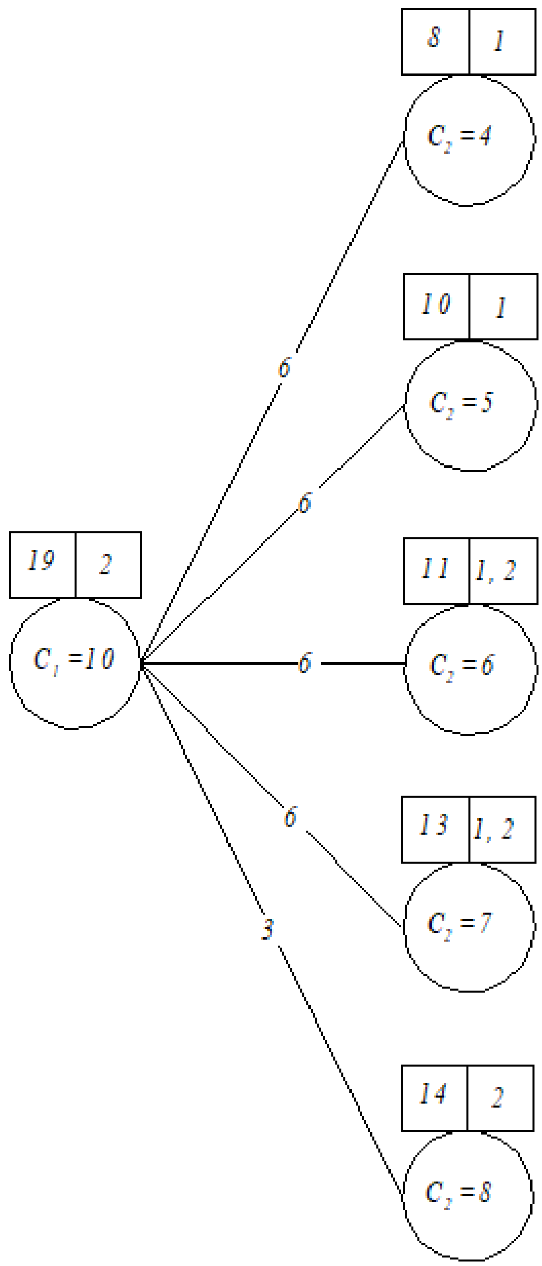


Рисунок 4 – Шаг 1 решения задачи графическим способом

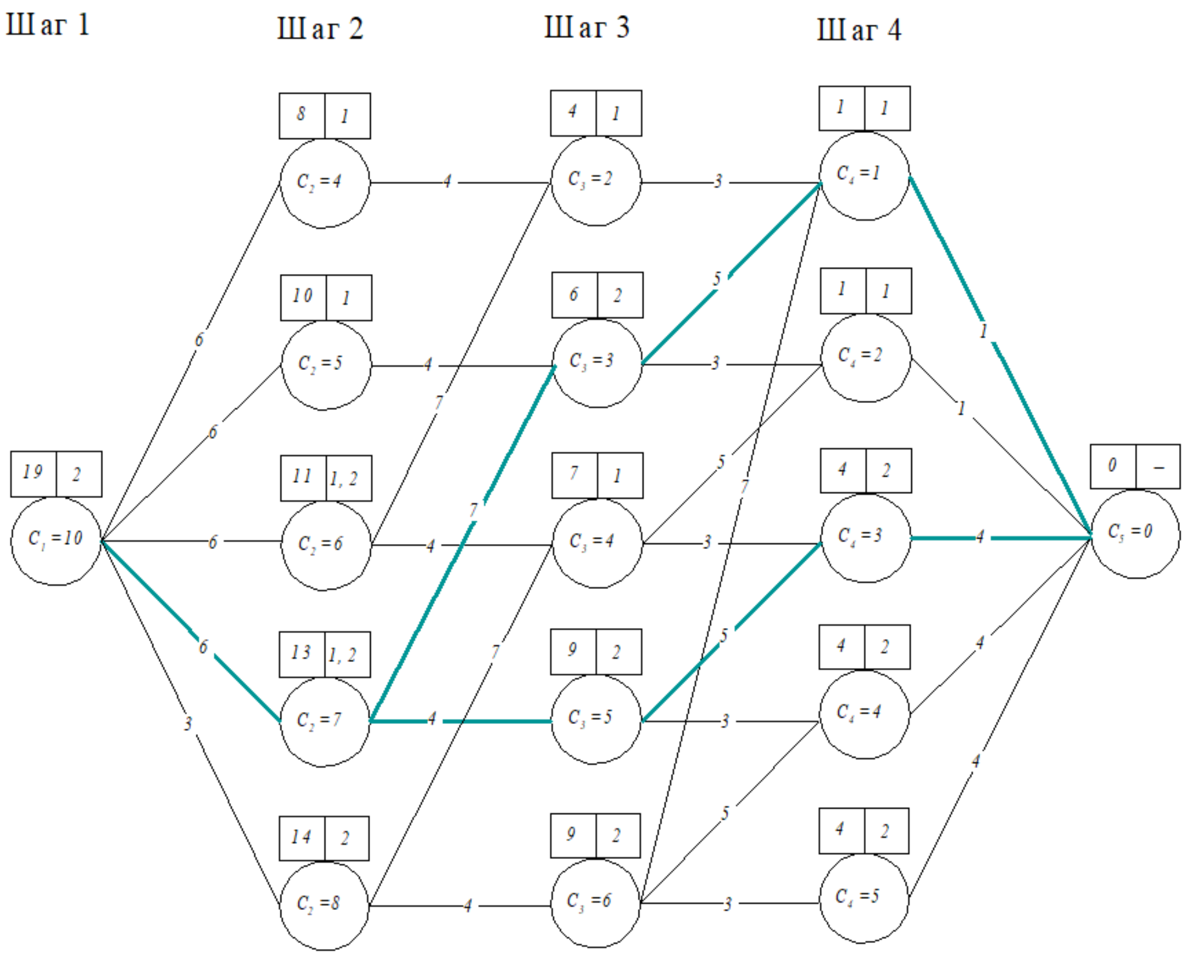


Рисунок 5 – Решение задачи графическим способом

2.3.3 Решение задачи табличным способом

Рекуррентное уравнение Беллмана для процедуры обратной прогонки для рассматриваемой задачи имеет вид:

$$F_5(C_5) = 0,$$

$$F_j(\mathrm{C_j}) = \max_{x_j | I_j(x_j) \le C_j} \left\{ \mathrm{P_j}(x_j) + F_{j+1}\left(\mathrm{C_j} - \mathrm{I_j}(x_j)\right) \right\},$$

$$j = 4, 3, 2, 1.$$

Начнем с решения задачи для производственной площадки №4. Затем будем решать задачу для производственных площадок №3 и №4, затем для площадок №2, №3 и №4. Завершающим шагом будет решение задачи для площадок №1, №2, №3, №4.

Расчет будем вестись вычислением значения функции $P_j(x_j) + F_{j+1}\left(C_j - I_j(x_j)\right)$ переменных C_j и x_j. Функция рассчитывается для каждого возможного значения состояния C_j и управления x_j. Если вычисление для некоторых пар (C_j, x_j) невозможно, то в качестве значения функции ставится прочерк.

Шаг 4. На шаге 4 решается задача для производственной площадки №4. Результаты вычислений внесены в таблицу 5.

Таблица 5 – Шаг 4. Производственная площадка №4

$F_4(C_4) = \max\limits_{x_4=1:2 \mid I_4(x_4) \le C_4} \{P_4(x_4)\}$				
C_4	$P_4(x_4)$		Оптимальное решение	
	$x_4 = 1$	$x_4 = 2$	$F_4(C_4)$	x_4^*
1	1	–	**1**	**1**
2	1	–	1	1
3	1	4	**4**	**2**
4	1	4	4	2
5	1	4	4	2
6	1	4	4	2
7	1	4	4	2
8	1	4	4	2
9	1	4	4	2
10	1	4	4	2

В первый столбец C_4 таблицы 5 внесены все возможные варианты инвестиций в модернизацию производственной площадки №4: $C_4 \in 1:10$. Два следующих столбца $x_4 = 1$ и $x_4 = 2$ соответствуют двум проектам.

Для реализации проекта $x_4 = 1$ на производственной площадке №4 необходимо вложить $I_4 = 1$ млн. руб. При этом будет получен доход $P_4 = 1$ млн. руб. Весь столбец $x_4 = 1$ заполняется цифрой 1.

Для реализации проекта $x_4 = 2$ на производственной площадке №4 необходимо вложить $I_4 = 3$ млн. руб. Варианты вложения инвестиций $C_4 = 1$

млн. руб. и $C_4 = 2$ млн. руб. для проекта $x_4 = 2$ недопустимы, поэтому ставится прочерк. При вложении инвестиций $C_4 \geq 3$ млн. руб., будет получен доход $P_4 = 4$ млн. руб. Во всех строках столбца $x_4 = 2$, удовлетворяющих условию $C_4 \geq 3$, проставляется цифра 4.

В последних двух столбцах таблицы 5 анализируется оптимальное решение. В столбце $F_4(C_4)$ проставляется максимальное число из столбцов $P_4(x_4)$, а в столбце x_4^* – соответствующий ему номер проекта.

Шаг 3. На шаге 3 решается задача для производственных площадок №3 и №4. Результаты вычислений внесены в таблицу 6.

Таблица 6 – Шаг 3. Производственные площадки №3 и №4

$F_3(C_3) = \max\limits_{x_3=1:3 \mid I_3(x_3) \leq C_3} \{P_3(x_3) + F_4(C_3 - I_3(x_3))\}$					
C_3	$P_3(x_3) + F_4(C_3 - I_3(x_3))$			Оптимальное решение	
	$x_3 = 1$	$x_3 = 2$	$x_3 = 3$	$F_3(C_3)$	x_3^*
1	–	–	–	–	–
2	3 + 1 = 4	–	–	4	1
3	3 + 1 = 4	5 + 1 = 6	–	**6**	**2**
4	3 + 4 = 7	5 + 1 = 6	–	7	1
5	3 + 4 = 7	5 + 4 = 9	–	**9**	**2**
6	3 + 4 = 7	5 + 4 = 9	7 + 1 = 8	9	2
7	3 + 4 = 7	5 + 4 = 9	7 + 1 = 8	9	2
8	3 + 4 = 7	5 + 4 = 9	7 + 4 = 11	11	3
9	3 + 4 = 7	5 + 4 = 9	7 + 4 = 11	11	3
10	3 + 4 = 7	5 + 4 = 9	7 + 4 = 11	11	3

В первый столбец C_3 таблицы 6 внесены все возможные варианты инвестиций в модернизацию производственных площадок №3 и №4: $C_3 \in 1{:}10$. Три следующих столбца $x_3 = 1$, $x_3 = 2$ и $x_3 = 3$ соответствуют трем проектам.

Столбцы $x_3 = 1$, $x_3 = 2$ и $x_3 = 3$ заполняются путем вычисления функции $P_3(x_3) + F_4(C_3 - I_3(x_3))$. Значение $P_3(x_3)$ берётся из исходных данных для производственной площадки №3 (таблица 1, столбец P_3), а значение

$F_4(C_3 - I_3(x_3))$ – из таблицы 5, столбец $F_4(C_4)$. Если инвестиций C_3 недостаточно для реализации проектов, ставится прочерк.

В последних двух столбцах таблицы 6 анализируется оптимальное решение. В столбце $F_3(C_3)$ проставляется максимальное число из столбцов $P_3(x_3) + F_4(C_3 - I_3(x_3))$, а в столбце x_3^* – соответствующий ему номер проекта.

Шаг 2. На шаге 2 решается задача для производственных площадок №2, №3 и №4. Результаты вычислений внесены в таблицу 7.

Таблица 7 – Шаг 2. Производственные площадки №2, №3 и №4

$F_2(C_2) = \max\limits_{x_2=1:3 \mid I_2(x_2) \le C_2} \{P_2(x_2) + F_3(C_2 - I_2(x_2))\}$				
C_2	$P_2(x_2) + F_3(C_2 - I_2(x_2))$		Оптимальное решение	
	$x_2 = 1$	$x_2 = 2$	$F_2(C_2)$	x_2^*
1	–	–	–	–
2	–	–	–	–
3	–	–	–	–
4	4 + 4 = 8	–	8	1
5	4 + 6 = 10	–	10	1
6	4 + 7 = 11	7 + 4 = 11	11	1, 2
7	4 + 9 = 13	7 + 6 = 13	**13**	**1, 2**
8	4 + 9 = 13	7 + 7 = 14	14	2
9	4 + 9 = 13	7 + 9 = 16	16	2
10	4 + 11 = 15	7 + 9 = 16	16	2

В первый столбец C_2 таблицы 7 внесены все возможные варианты инвестиций в модернизацию производственных площадок №2, №3 и №4: $C_2 \in 1:10$. Два следующих столбца $x_2 = 1$ и $x_2 = 2$ соответствуют двум проектам.

Столбцы $x_2 = 1$ и $x_2 = 2$ заполняются путем вычисления функции $P_2(x_2) + F_3(C_2 - I_2(x_2))$. Значение $P_2(x_2)$ берётся из исходных данных для производственной площадки №2 (таблица 1, столбец P_2), а значение $F_3(C_2 - I_2(x_2))$ – из таблицы 6, столбец $F_3(C_3)$. Если инвестиций C_2 недостаточно для реализации проектов, ставится прочерк.

В последних двух столбцах таблицы 7 анализируется оптимальное решение. В столбце $F_2(C_2)$ проставляется максимальное число из столбцов $P_2(x_2) + F_3\big(C_2 - I_2(x_2)\big)$, а в столбце x_2^* – соответствующий ему номер проекта.

Шаг 1. На шаге 1 решается задача для производственных площадок №1, №2, №3 и №4. Результаты вычислений внесены в таблицу 8.

Таблица 8 – Шаг 1. Производственные площадки №1, №2, №3 и №4

$F_1(C_1) = \max\limits_{x_1=1:2 \mid I_1(x_1) \le C_1} \{P_1(x_1) + F_2(C_1 - I_1(x_1))\}$				
C_1	$P_1(x_1) + F_2(C_1 - I_1(x_1))$		Оптимальное решение	
	$x_1 = 1$	$x_1 = 2$	$F_1(C_1)$	x_1^*
1	–	–	–	–
2	–	–	–	–
3	–	–	–	–
4	–	–	–	–
5	–	–	–	–
6	3 + 8 = 11	–	11	1
7	3 + 10 = 13	6 + 8 = 14	14	2
8	3 + 11 = 14	6 + 10 = 16	16	2
9	3 + 13 = 16	6 + 11 = 17	17	2
10	3 + 14 = 17	6 + 13 = 19	**19**	**2**

В первый столбец C_1 таблицы 8 внесены все возможные варианты инвестиций в модернизацию производственных площадок №1, №2, №3 и №4: $C_1 \in 1:10$. Два следующих столбца $x_1 = 1$ и $x_1 = 2$ соответствуют двум проектам.

Столбцы $x_1 = 1$ и $x_1 = 2$ заполняются путем вычисления функции $P_1(x_1) + F_2(C_1 - I_1(x_1))$. Значение $P_1(x_1)$ берётся из исходных данных для производственной площадки №1 (таблица 1, столбец P_1), а значение $F_2(C_1 - I_1(x_1))$ – из таблицы 7, столбец $F_2(C_2)$. Если инвестиций C_1 недостаточно для реализации проектов, ставится прочерк.

В последних двух столбцах таблицы 8 анализируется оптимальное решение. В столбце $F_1(C_1)$ проставляется максимальное число из столбцов

$P_1(x_1) + F_2(C_1 - I_1(x_1))$, а в столбце x_1^* – соответствующий ему номер проекта.

Когда все таблицы заполнены, необходимо приступить к поиску оптимального решения. Из таблицы 8 видно, что максимально возможный доход составляет 19 млн. руб. Для его достижения на производственной площадке №1 необходимо выбрать проект 2.

Значит для модернизации производственных площадок №2, №3 и №4 останется 19 – P_1 = 19 – 6 = 13 млн. руб. Из таблицы 7 видно, что для получения максимально возможного дохода 13 млн. руб. на производственной площадке №2 необходимо выбрать проект 1 или 2. Так как оба этих варианта являются оптимальными, просчитаем каждый из них.

Вариант 1. На производственной площадке №2 выбран проект 1. Значит для модернизации производственных площадок №3 и №4 останется 13 – P_2 = 13 – – 4 = 9 млн. руб. Из таблицы 6 видно, что для получения максимально возможного дохода 9 млн. руб. на производственной площадке №3 необходимо выбрать проект 2.

Вариант 2. На производственной площадке №2 выбран проект 2. Значит для модернизации производственных площадок №3 и №4 останется 13 – P_2 = 13 – – 7 = 6 млн. руб. Из таблицы 6 видно, что для получения максимально возможного дохода 6 млн. руб. на производственной площадке №3 необходимо выбрать проект 2.

Так как оба этих варианта являются оптимальными, просчитаем каждый из них.

Вариант 1. Для модернизации производственной площадки №4 останется: 9 – P_3 = 9 – 5 = 4 млн. руб. Из таблицы 5 видно, что для получения максимально возможного дохода 4 млн. руб. на производственной площадке №1 необходимо выбрать проект 2.

Вариант 2. Для модернизации производственной площадки №4 останется: 6 – P_3 = 6 – 5 = 1 млн. руб. Из таблицы 5 видно, что для получения максимально возможного дохода 1 млн. руб. на производственной площадке №1 необходимо выбрать проект 1.

Составим комбинации проектов, при реализации которых оптимальным образом будут вложены имеющиеся у предприятия 10 млн. руб.

Таким образом, после вычисления задачи табличным способом, были найдены два оптимальных решения:

– $x_{(1)} = (2, 1, 2, 2)$, $F(10) = 19$,

– $x_{(2)} = (2, 2, 2, 1)$, $F(10) = 19$.

Эти оптимальные решения совпадают с решениями, найденными ранее способом полного перебора и графическим способом [12], [30], [41].

2.3.4 Решение задачи при помощи программного обеспечения для математических вычислений

Для решения задачи рассматривались следующие пакеты математических вычислений:

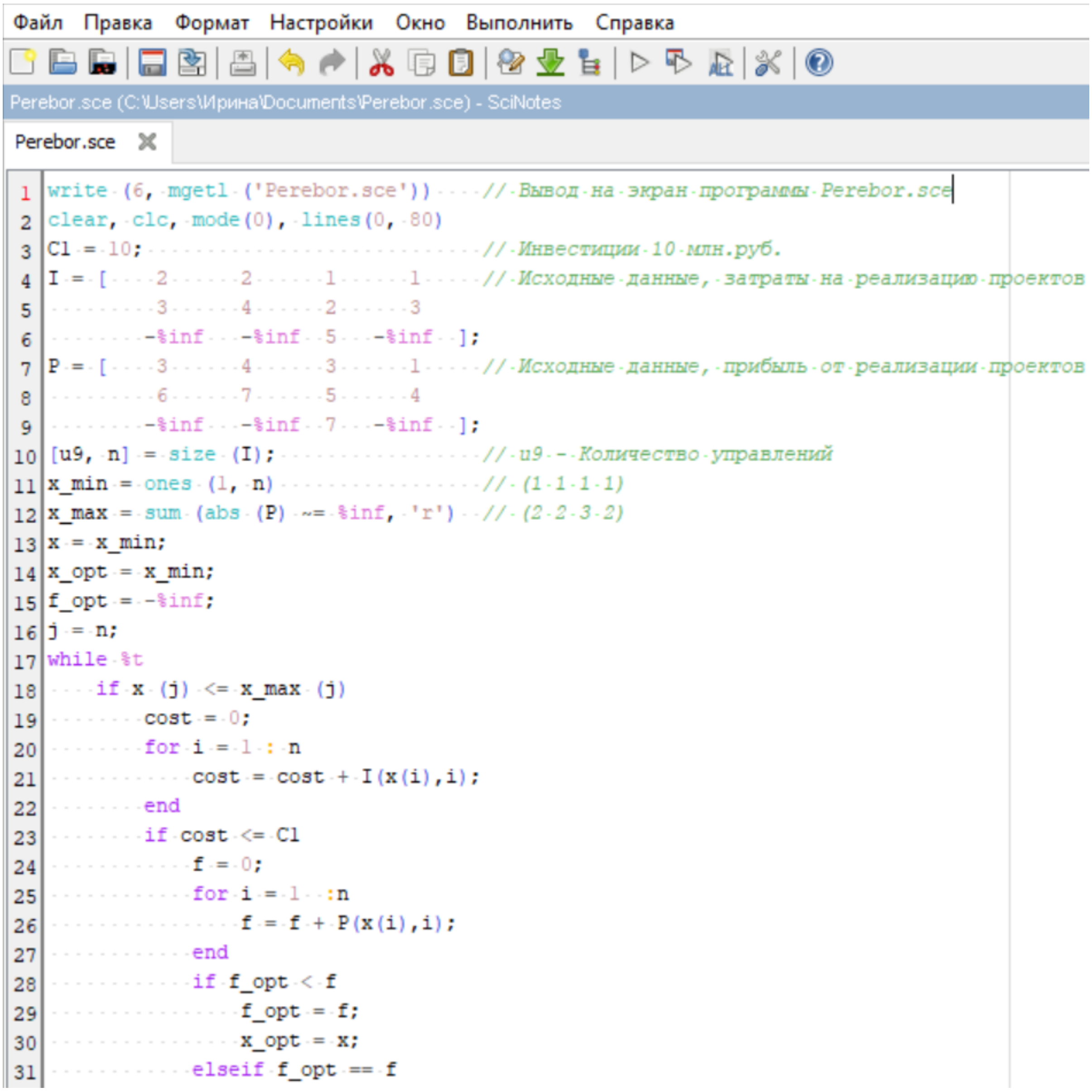
- Mathematica,
- Maple,
- Mathlab,
- Mathcad,
- Scilab.

После оценки ключевых особенностей перечисленных математических пакетов, для проведения расчетов было решено использовать программное обеспечение с открытым исходным кодом для численных вычислений Scilab. На решение повлияли следующие положительные отличия данной среды программирования:

- бесплатность и свободность распространения. Русифицированную версию программного обеспечения можно бесплатно скачать на официальном сайте по ссылке https://www.scilab.org/download/scilab-2024.0.0.;
- небольшой размер дистрибутива;

– возможность запуска в консоли без использования графического интерфейса.

Для решения задачи была использована версия Scilab 2024.0.0 для Windows. Программа написана в блокноте SciNotes, а результат выведен в командном окне. Комментарии выделены парным символом слэш. Код программы представлен на рисунках 6 и 7, ответ представлен на рисунке 8.




```
write (6, mgetl ('Perebor.sce'))    // Вывод на экран программы Perebor.sce
clear, clc, mode(0), lines(0, 80)
C1 = 10;                            // Инвестиции 10 млн.руб.
I = [    2      2      1      1     // Исходные данные, затраты на реализацию проектов
         3      4      2      3
       -%inf  -%inf    5   -%inf ];
P = [    3      4      3      1     // Исходные данные, прибыль от реализации проектов
         6      7      5      4
       -%inf  -%inf    7   -%inf ];
[u9, n] = size (I);                 // u9 - Количество управлений
x_min = ones (1, n)                 // (1 1 1 1)
x_max = sum (abs (P) ~= %inf, 'r')  // (2 2 3 2)
x = x_min;
x_opt = x_min;
f_opt = -%inf;
j = n;
while %t
    if x (j) <= x_max (j)
        cost = 0;
        for i = 1 : n
            cost = cost + I(x(i),i);
        end
        if cost <= C1
            f = 0;
            for i = 1 :n
                f = f + P(x(i),i);
            end
            if f_opt < f
                f_opt = f;
                x_opt = x;
            elseif f_opt == f
```

Рисунок 6 – Код программы, часть 1

```
                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-8
            break
        end
    end // if
    x (j) = x (j) + 1;
end // while 1
write (6, [' *****Ответ***** '])
write (6, ' f_opt = ' + string (f_opt))
x_opt
exec 'Perebor.sce' // Запуск программы Perebor.sce на выполнение.
```

Рисунок 7 – Код программы, часть 2

```
x_min  =
   1.   1.   1.   1.
x_max  =
   2.   2.   3.   2.
*****Ответ*****
f_opt = 19
x_opt  =
   2.   1.   2.   2.
   2.   2.   2.   1.
```

Рисунок 8 – Ответ, выведенный в командном окне

Таким образом, при помощи программы, написанной в пакете математических вычислений SciLab, были найдены два оптимальных решения:

- $x_{(1)} = (2, 1, 2, 2)$, $F(10) = 19$,
- $x_{(2)} = (2, 2, 2, 1)$, $F(10) = 19$.

Эти оптимальные решения совпадают с решениями, найденными ранее способом полного перебора, графическим способом, табличным способом [27], [31], [32], [34].

Выводы по главе 2

В главе 2 была поставлена задача по оптимальному вложению инвестиций в модернизацию производства по переработке сырья. Для рассматриваемой задачи была сформулирована математическая модель.

Поставленная задача была решена следующими способами с применением метода динамического программирования:

- способом полного перебора,
- графическим способом,
- табличным способом,
- при помощи программного обеспечения для математических вычислений Scilab версии 2024.0.0.

Ответы задачи, вычисленные различными способами, совпали. Стоит отметить, что применение метода динамического программирования значительно ускоряет процесс решения задачи, позволяет экономить время выполнения, избегать повторного вычисления одних и тех же подзадач.

Глава 3 Разработка программного обеспечения для решения задачи оптимального распределения ресурсов

3.1 Выбор языка программирования

При выборе языка программирования учитывалось множество факторов:

- навыки разработчика при работе с языками программирования;
- требования к производительности языка;
- требования к безопасности языка;
- наличие библиотек, фреймворков и инструментов;
- наличие документации и справочных материалов, примеров кода;
- возможность создавать кросс-платформенные приложения, которые можно запускать в различных операционных системах;
- сложность решаемой задачи и тип планируемого решения задачи.

Наиболее подходящим языком программирования для решения поставленной задачи оказался Python. Это универсальный и относительно легкий язык, имеющий обширный диапазон применения, большое количество библиотек, хорошую скорость работы и простой синтаксис.

3.2 Описание графического интерфейса пользователя

Для создания графического интерфейса пользователя была выбрана библиотека Tkinter. Это стандартная библиотека Python, которая предоставляет собой набор виджетов и инструментов для создания интерактивных и диалоговых окон.

Преимущества Tkinter:

- не требует дополнительной установки;

– позволяет быстро создавать приложения с простым графическим интерфейсом на Python;
– позволяет создавать кросс-платформенные приложения, которые можно запускать в различных операционных системах [16], [40].

На рисунке 9 представлена схема работы с библиотекой Tkinter.

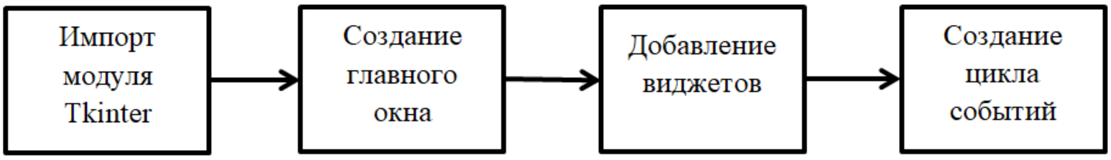


Рисунок 9 – Схема работы с библиотекой Tkinter

Сначала импортируем Tkinter. Виджет messagebox позволяет создать отдельное окно с информацией:

```
import tkinter as tk
from tkinter import messagebox
```

Создадим главное окно приложения и назначаем ему заголовок «Оптимальные проекты»:

```
root = tk.Tk()
root.title("Оптимальные проекты")
```

Создадим виджет Frame для позиционирования элементов. В окне приложения необходимо разместить следующие элементы: поля ввода исходных данных и одну кнопку для запуска расчета. Применим метод grid для размещения элементов по условной сетке. Регулировать расположение элементов будем следующими параметрами:

– column – номер столбца;

– row – номер строки;

– columnspan – количество столбцов, которые будет занимать элемент;

– padx – отступ по горизонтали;

– pady – отступ по вертикали.

Поля ввода исходных данных создадим при помощи виджетов Entry (поле для ввода информации) и Label (текстовые надписи).

```
frame_projects = tk.Frame(root)
frame_projects.grid(row=0, column=0, padx=10, pady=10)

label_projects = tk.Label(frame_projects, text="Введите стоимость модернизации (cost) и годовую прибыль (profit) для каждого проекта:")
label_projects.grid(row=0, column=0, columnspan=4, pady=(0, 5))

entries_cost = []
for i in range(3):
    row_entries = []
    for j in range(4):
        entry = tk.Entry(frame_projects, width=5)
        entry.grid(row=i+1, column=j, padx=5, pady=5)
        row_entries.append(entry)
    entries_cost.append(row_entries)

label_cost = tk.Label(frame_projects, text="Cost")
label_cost.grid(row=4, column=0, pady=(0, 5))

entries_profit = []
for i in range(3):
    row_entries = []
    for j in range(4):
```

```
            entry = tk.Entry(frame_projects, width=5)
            entry.grid(row=i+5, column=j, padx=5, pady=5)
            row_entries.append(entry)
        entries_profit.append(row_entries)

    label_profit = tk.Label(frame_projects, text="Profit")
    label_profit.grid(row=8, column=0, pady=(0, 5))

    frame_budget = tk.Frame(root)
    frame_budget.grid(row=1, column=0, padx=10, pady=10)

    label_budget = tk.Label(frame_budget, text="Введите бюджет:")
    label_budget.grid(row=0, column=0)
    entry_budget = tk.Entry(frame_budget, width=10)
    entry_budget.grid(row=0, column=1)
```

Создадим кнопку для запуска расчета:

```
    button_calculate = tk.Button(root, text="Рассчитать", command=calculate)
    button_calculate.grid(row=2, column=0, pady=(10, 0))

    text_output = tk.Text(root, width=40, height=10)
    text_output.grid(row=3, column=0, padx=10, pady=10)
```

3.3 Разработка программного кода

Для генерации всех возможных комбинаций проектов, при реализации которых оптимальным образом будут вложены имеющиеся у предприятия 10 млн. руб., применим itertools.product. Благодаря этому методу формируется декартово произведение, в результате которого на выходе будут получены кортежи вида (1, 2, 3, 4) с элементами из каждого списка:

from itertools import product

Поиск оптимального решения:

```
def find_optimal_solutions(projects, budget):
    num_projects = len(projects)
    num_locations = len(projects[0])
```

Вычислим все возможные комбинации проектов для каждой площадки:

all_combinations = list(product(range(num_projects), repeat= num_ locations))

Введем ограничение: рассматривать только те комбинации, которые соответствуют бюджету:

valid_combinations = [comb for comb in all_combinations if sum(projects[comb[i]][i][0] for i in range(num_locations)) == budget]

Найдем максимальную прибыль среди всех валидных комбинаций:

```
max_profit = 0
optimal_solutions = []
for comb in valid_combinations:
    profit = sum(projects[comb[i]][i][1] for i in range(num_locations))
    if profit > max_profit:
    max_profit = profit
        optimal_solutions = [tuple(p + 1 for p in comb)]
    elif profit == max_profit:
        optimal_solutions.append(tuple(p + 1 for p in comb))

    return optimal_solutions, max_profit
```

Опишем функцию:

```
def calculate():
```

```
    projects = []
    for i in range(3):
        row = []
        for j in range(4):
            cost = int(entries_cost[i][j].get())
            profit = int(entries_profit[i][j].get())
            row.append((cost, profit))
        projects.append(row)
```

Определим и обработаем исключения:

```
try:
    budget = int(entry_budget.get())
    optimal_solutions, max_profit = find_optimal_solutions(projects, budget)
```

Очистим вывод перед добавлением новых результатов:

```
text_output.delete(1.0, tk.END)

text_output.insert(tk.END, "Оптимальные проекты для каждой площадки:\n")
for solution in optimal_solutions:
    text_output.insert(tk.END, solution)
    text_output.insert(tk.END, "\n")
text_output.insert(tk.END, "Максимальная прибыль: {}\n".format(max_profit))
```

Запустим приложение:

root.mainloop() [4], [18], [19], [25], [36]

3.4 Тестирование и отладка программного обеспечения

Для проверки работоспособности разработанного программного обеспечения был применен метод функционального тестирования. Главная задача этого метода – удостовериться, что программное обеспечение работает в соответствии с заданными функциональными требованиями и не содержит ошибок. Обнаруженные ошибки должны быть исправлены и затем должно быть проведено повторное тестирование.

Для лучшего понимания целей и задач функционального тестирования, сравним его с нефункциональным тестированием. На рисунке 10 представлено сравнение этих двух методов [38].

	Функциональное тестирование – проверяет, обеспечивает ли система желаемый результат	Нефункциональное тестирование – проверяет производительность, безопасность, масштабируемость
Метод	Обычно выполняется по методу “черного ящика”, при котором тестировщики проверяют только входы и выходы, а не внутреннюю структуру системы.	Обычно выполняется по методу “белого ящика”. В этом случае тестировщик получает информацию о внутреннем устройстве системы для создания соответствующих тест-кейсов.
Что исследуется	Удовлетворяют ли выходные данные системы заданным спецификациям или требованиям.	Производительность системы, стабильность, безопасность, удобство использования и т.д.
Входные данные	Бизнес-требования, спецификации клиента.	Скорость, пропускная способность, масштабируемость и т.д.
Примеры	– модульное тестирование; – тестирование API; – регрессионное тестирование (может быть как функциональным, так и нефункциональным).	– тестирование безопасности; – тестирование производительности; – нагрузочное тестирование; – стресс-тестирование.

Рисунок 10 – Разница между функциональным тестированием и нефункциональным тестированием

При запуске программного обеспечения перед пользователем появляется автономное диалоговое окно. В строке заголовка видим название приложения – «Оптимальные проекты». Диалоговое окно содержит следующие элементы:

– поля ввода стоимости модернизации трех проектов (invest) для каждого предприятия;
– поля ввода годовой прибыли от реализации проектов (profit) для каждого предприятия;
– поле ввода бюджета, выделенного на модернизацию всех площадок;
– кнопку «Рассчитать», при нажатии на которую происходит запуск алгоритма расчета.

На рисунке 11 представлен графический интерфейс пользователя.

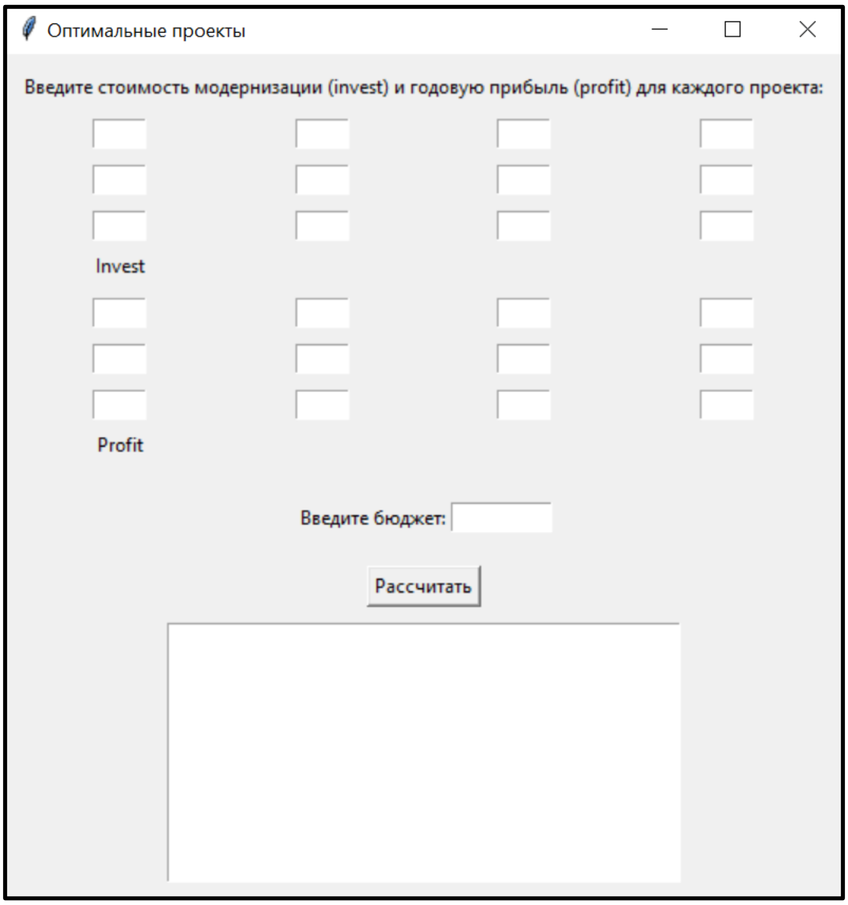


Рисунок 11 – Графический интерфейс пользователя

Далее пользователю предлагается ввести исходные данные задачи. На рисунке 12 представлен ввод исходных данных.

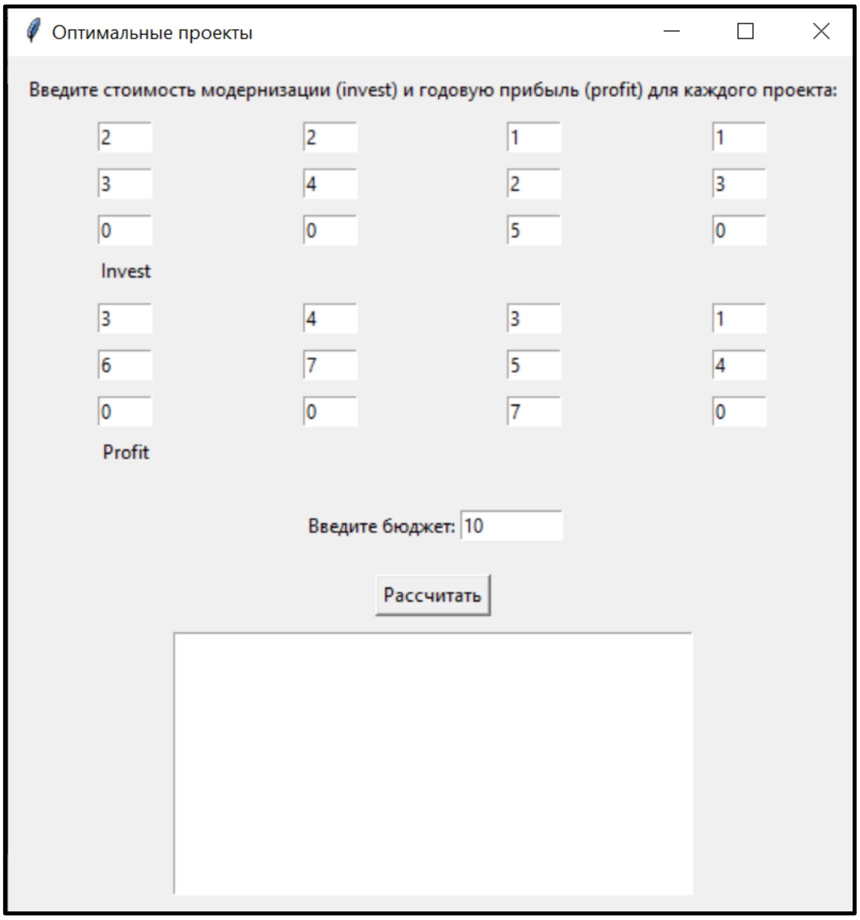


Рисунок 12 – Ввод исходных данных

Алгоритм программы построен таким образом, что программа не начнет расчет, пока не будут введены все исходные данные. В этом случае появляется всплывающее окно, сообщающее об ошибке. На рисунке 13 проиллюстрирован случай, когда пользователь не ввел значение в поле бюджета.

На рисунке 14 показан результат выполнения программы. Ответ представлен в виде кортежей (1, 2, 3, 4).

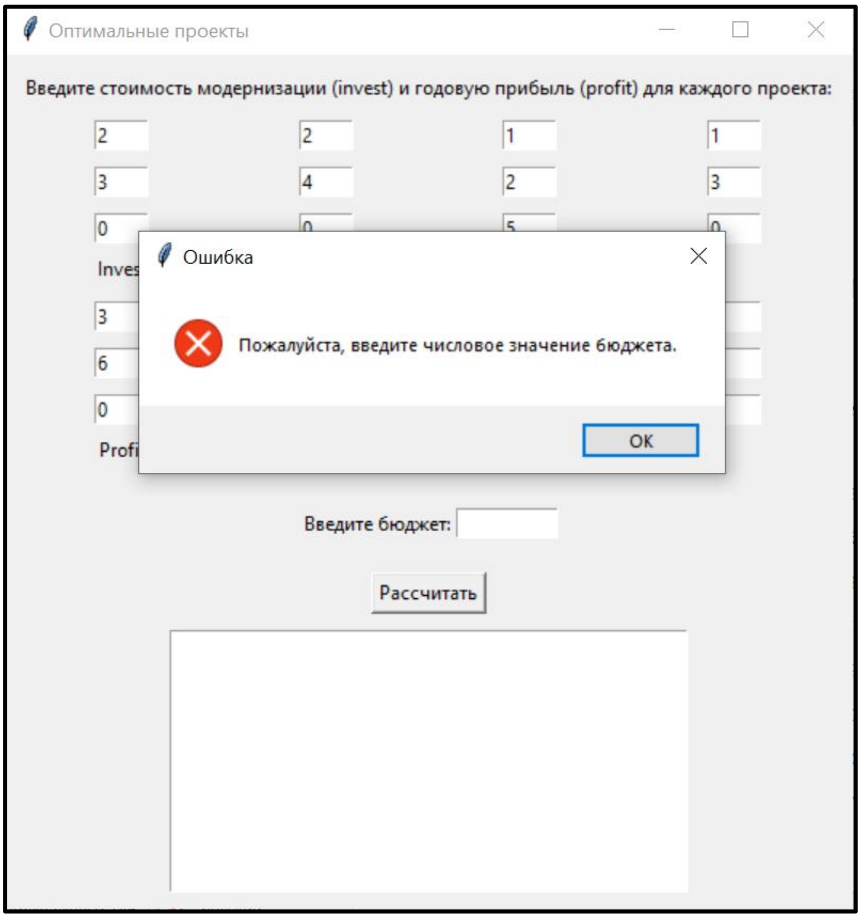


Рисунок 13 – Предупреждение об ошибке

Таким образом, при помощи программного обеспечения, разработанного для решения задачи оптимального распределения ресурсов на языке программирования Python, были найдены два оптимальных решения:

– x(1) = (2, 1, 2, 2), F(10) = 19,

– x(2) = (2, 2, 2, 1), F(10) = 19.

Эти оптимальные решения совпадают с решениями, найденными ранее способом полного перебора, графическим способом, табличным способом и при помощи программы, написанной в пакете математических вычислений SciLab.

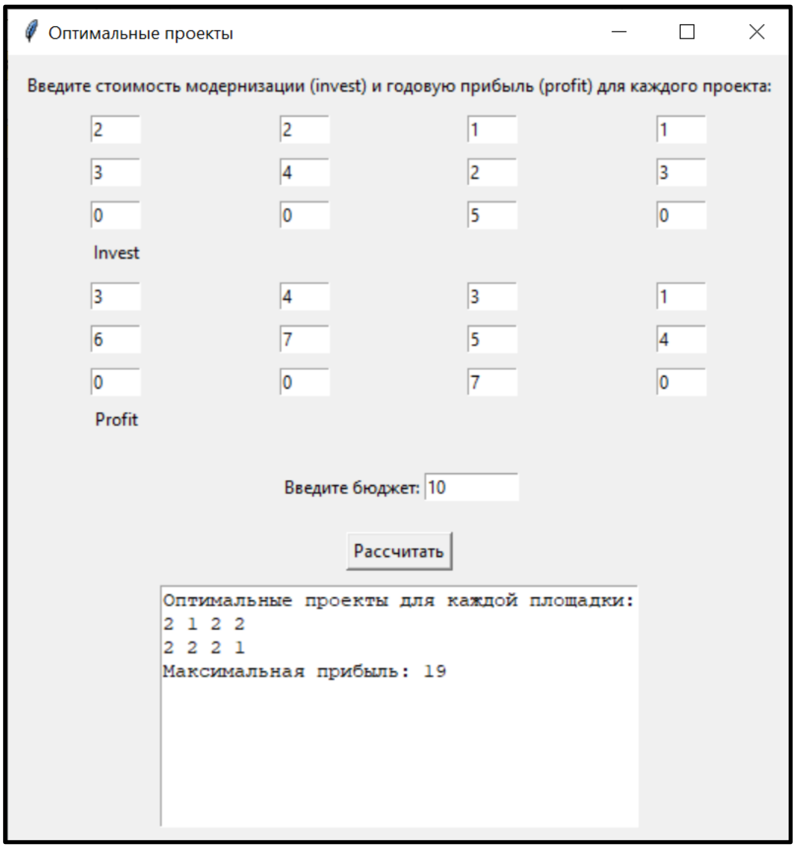


Рисунок 14 – Результат выполнения программы

Выводы по главе 3

В главе 3 разработано и описано программное обеспечение для решения задачи оптимального распределения ресурсов. Рассмотрены требования к разрабатываемому программному обеспечению. Проанализированы языки программирования, среди которых выбран Python. На его основе разработан и описан код программного обеспечения.

Проанализированы средства создания графического интерфейса пользователя, среди которых выбрана библиотека Tkinter. На ее основе разработан и описан графический интерфейс пользователя. С целью проверки работоспособности разработанного программного обеспечения был применен и описан метод функционального тестирования.

Решение задачи с применением предложенного программного обеспечения имеет ряд преимуществ над решением задачи вручную, такие как быстрота и надежность расчетов, возможность легкой корректировки исходных данных, защита от ошибок.

Заключение

В результате комплекса проведенных исследований по изучению применения метода динамического программирования при решении задач оптимальной переработки сырья, были получены следующие основные результаты и выводы:

- полученные результаты исследования применения метода динамического программирования на производственном предприятии в вопросах оптимального распределения инвестиций, позволяют провести оценку их влияния на эффективность принятия управленческих решений;
- применение метода динамического программирования существенно повышает эффективность управления производством по переработке сырья, снижает экономические затраты, обеспечивает конкурентоспособность и рентабельность производства;
- применение метода динамического программирования связано с определенными трудностями, заключающимися в сложности вычислительных процессов;
- результатом выполнения работы стала разработка математической модели, перспективной для внедрения на предприятии по переработке сырья, и проектирование программного обеспечения, с помощью которого возможно решение проблемы оптимизации затрат на переработку сырья, с учетом различных ограничений и особенностей производства;
- применение разработанного программного обеспечения на предприятии для решения вопросов по оптимальному распределению ресурсов позволит обеспечить повышение эффективности принимаемых инвестиционных решений, повышение доходности и снижение финансовых рисков.

Цель и задачи, поставленные на начальном этапе выполнения работы, достигнуты и решены в полном объеме. Подтверждена гипотеза исследования, состоящая в предположении, что применение метода динамического программирования существенно повышает эффективность управления производством по переработке сырья, снижает экономические затраты, обеспечивает конкурентоспособность и рентабельность производства. Однако применение метода динамического программирования связано с определенными трудностями, заключающимися в сложности вычислительных процессов.

Проведенные исследования позволяют предупреждать и минимизировать финансовые риски при планировании модернизации производственных площадок. Возможно использование полученных результатов исследований в других направлениях, где применим метод динамического программирования.

Предложение по продолжению работы заключается в модернизации программного обеспечения. В результате может быть получен конкурентоспособный продукт с возможностью коммерциализации и дальнейшего продвижения.

Список используемой 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. Архипкина А. И. Метод динамического программирования как инструмент поддержки принятия решения при планировании инвестиционного проекта // Наука и образование: новое время. 2018. № 3.

3. Ашманов С. А., Карманов В. Г. Динамическое программирование // Математическая энциклопедия. М.: Советская энциклопедия. 1979. Т. 2. С. 154–155.

4. Баландин Д. В. Программный модуль для построения оптимального графика переработки сырья // Современные информационные технологии и ИТ-образование. 2021. Т. 17, № 2. С. 442-452.

5. Баширзаде Л. И., Алиев Г. С. Применение динамического программирования для моделирования процессов принятия решений // Научный журнал «Архивариус». 2022. Т. 7. № 3 (66). С. 51-55.

6. Беккенбах Э. Ф. Современная математика для инженеров. М.: Иностранная литература, 1958. 489 с.

7. Беллман Р. Динамическое программирование. М.: Иностранная литература, 1960. 395 с.

8. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. М.: Наука, 1965. 460 с.

9. Боровков А. А., Савельев В. П. Оптимизация ритмичности производства с двумя типами сырья // Математическое моделирование. Оптимальное управление. Вестник Нижегородского университета им. Н.И. Лобачевского. 2013. № 4 (1). С. 216–218.

10. Вентцель Е. С. Исследование операций. Задачи, принципы, методология: учеб. пособие для студ. вузов. М.: Высшая школа, 2001. 208 с.

11. Вентцель Е. С. Элементы динамического программирования. М.: Наука, 1964. 176 с.

12. Визгунов Н. П. Динамическое программирование в экономических задачах с применением системы SciLab : учебно-методическое пособие. Н.Новгород: ННГУ, 2011. 60 с.

13. Габасов Р., Кириллова Ф. М. Основы динамического программирования. Минск: Изд-во БГУ им. В. И. Ленина, 1975. 264 с.

14. Гавриловская С. П. Принятие управленческих решений с использованием метода динамического программирования // Математические методы в технике и технологиях: сб. трудов XXII Междунар. науч. конф. Псков, 2009. Т. 7.

15. Гаджиев А. А., Сулейманова О. Ш. Задача оптимального распределения ресурсов и два подхода к её решению // Вестник Дагестанского государственного технического университета. Технические науки. 2010. № 16. С. 53-60.

16. Геворгян М. В., Киракосян Г. Т. Динамическая математическая модель и программное обеспечение оптимальной переработки техногенного ресурса // Автоматизация и системы управления. Изв. НАН РА и ГИУА. Сер. ТН. 2005. Т. LVIII, № 3. С. 593-601.

17. Грызина Н. Ю., Мастяева И. Н., Семенихина О. Н. Математические методы исследования операций в экономике: учебно-методический комплекс. М.: Изд. центр ЕАОИ, 2009. 196 с.

18. Документация Python [Электронный ресурс]: URL: https://digitology.tech/docs/python_3/ (дата обращения: 25.01.2024).

19. Доусон М. Программируем на Python. СПб.: Питер, 2012. 432 с.

20. Калихман И. Л., Войтенко М. А. Динамическое программирование в примерах и задачах: учеб. пособие. М.: Высш. Школа, 1979. 125 с.

21. Карпов Д. А., Струченков В. И. Динамическое программирование в прикладных задачах специального вида // Прикладная информатика. 2020. Том 15. № 3 (87). С. 46–59.

22. Коган Д. И. Динамическое программирование и дискретная многокритериальная оптимизация: учеб. пособие. Нижний Новгород: Изд-во ННГУ, 2004. 121 с.

23. Кремер Н. Ш., Путко Б. А., Тришин И. М., Фридман М. Н. Исследование операций в экономике: учеб. пособие для вузов. М.: Банки и биржи, ЮНИТИ, 1997. 407 с.

24. Лаборатория системного анализа. Исследование операций и оптимизация. Общие принципы и методология [Электронный ресурс]: URL: https://system-laboratory.ru/assets/systems_analysis.pdf (дата обращения: 05.01.2023).

25. Лутц М. Изучаем Python, 4-е издание. Пер. с англ. СПб.: Символ-Плюс, 2011. 1280 с.

26. Нефедов Д. Г. Математические модели и методы решения задач оптимального размещения элементов распределенной производственной структуры: диссертация ... кандидата наук. Челябинск, 2015. 120 с.

27. Официальное руководство по работе с пакетом SCILAB на английском языке [Электронный ресурс]: URL: https://www.scilab.org/sites/default/files/Scilab_beginners.pdf (дата обращения: 14.01.2024).

28. Пелих А. С., Терехов Л. Л., Терехова Л. А. Экономико-математические методы и модели в управлении производством. Ростов н/Д: Феникс, 2005. 248 с. (Высшее образование)

29. Пигнастый О. М. Обзор моделей управляемых производственных процессов поточных линий производственных систем // Научные ведомости. Серия Экономика. Информатика. 2015. № 7 (204). № 34/1. С. 137-152.

30. Постановка задачи оптимизации и численные методы ее решения [Электронный ресурс]: URL: https://hub.exponenta.ru/post/postanovka-zadachi-optimizatsii-i-chislennye-metody-ee-resheniya356#up/ (дата обращения: 05.10.2023).

31. Программное обеспечение с открытым исходным кодом для численных вычислений [Электронный ресурс] : URL: https://www.scilab.org/download/scilab-2024.0.0/ (дата обращения: 14.01.2024).

32. Решение задач динамического программирования сетевыми методами [Электронный ресурс]: URL: http://nauteh-journal.ru/files/700b5a07-a4e3-49ad-8db8-398b163221b8/ (дата обращения: 14.12.2023).

33. Рудычев А. А., Лева О. В. Динамическое программирование как средство моделирования производственной системы // Научные исследования, наносистемы и ресурсосберегающие технологии в стройиндустрии: сб. докл. Междунар. науч.-практ. конф. Белгород: Изд-во БГТУ им. В.Г. Шухова, 2007. Ч. 10.

34. Руководство по работе с пакетом SCILAB [Электронный ресурс]: URL: https://moodle.kstu.ru/pluginfile.php/308603/mod_resource/content/1/Scilab.pdf (дата обращения: 14.01.2024).

35. Рыкова И. Н. Моделирование банковского портфеля при размещении ресурсов на фондовом рынке // Финансовая аналитика: проблемы и решения. 2008. № 10(10). С. 44-54.

36. Седжвик Р., Уэйн К., Дондеро Р. Программирование на языке Python: учебный курс.: Пер. с англ. СПб: ООО “Альфа-книга”, 2017. 736 с.

37. Чернышев С. И. Об использовании метода динамического программирования Р. Беллмана в задачах экономического содержания // Бизнесинформ. 2013. № 6. С. 110-119.

38. Andersson J. A. E., Gillis J., Horn G. et al. CasADi – A software framework for nonlinear optimization and optimal control // Mathematical Programming Computation. 2019. Vol. 11, no. 1. Pp. 1–36.

39. Ben-Tal A., Ghaoui L.E., Nemirovski A. Robust optimization. Princeton and Oxford, Princeton University Press, 2009. 564 p.

40. Houska B., Ferreau H. J., Diehl M. ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization // Optimal Control Applications and Methods. 2011. Vol. 32, no. 3. Pp. 298–312.

41. Monteiro Renato, Adler Ilan. Interior path following primal-dual algorithms: Part II: Convex quadratic programming // Mathematical Programming. 1989. Vol. 44. Pp. 43–66.

42. Rothberg E. An Evolutionary Algorithm for Polishing Mixed Integer Programming Solutions // INFORMS Journal on Computing. 2007. 11. Vol. 19. Pp. 534–541.

43. Trentelman H., Stoorvogel A., Hautus M. Control Theory for Linear Systems. University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science, 2002. 403 pp.