МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

УЧЕБНОЕ ИЗДАНИЕ

ОПТИМИЗАЦИЯ ПРОЕКТНЫХ РЕШЕНИЙ

Министерство образования Республики Беларусь
Учреждение образования «Барановичский государственный университет»

ОПТИМИЗАЦИЯ ПРОЕКТНЫХ РЕШЕНИЙ

Методические рекомендации
по выполнению лабораторных работ

Барановичи
БарГУ
2022

Об издании ***1, 2***

УДК 004.02
ББК 22.19
О62

Составители: О. И. Наранович, О. Д. Кравчук

Рецензенты:
кандидат технических наук, доцент, заведующий кафедрой технологии и оборудования машиностроения учреждения образования «Барановичский государственный университет» *И. А. Богданович*;
инженер-программист ОДО «Тибмебель» *К. В. Козел*

Оптимизация проектных решений : метод. рекомендации по выполнению лаборатор. работ / сост.: О. И. Наранович, О. Д. Кравчук ; М-во образования Респ. Беларусь, Баранович гос. ун-т. — Барановичи : БарГУ, 2022. — 71 с.

Содержит девять лабораторных работ с теоретическим материалом по методам оптимизации и примерами их решения.
Издание предназначено для студентов III и IV курсов инженерного факультета БарГУ.

УДК 004.02
ББК 22.19

Текстовое электронное издание

Системные требования:
IBM PC 486 (рекомендовано Pentium и выше); Windows XP и выше или Linux;
Adobe Acrobat Pro DC; ОЗУ 256 Мб; видеокарта и монитор (1024 × 768); мышь.

Регистрационное свидетельство № 2202230672 от 25.11.2022 г.

‹‹‹ 2 – производственно-технические сведения ›››

- Использованное ПО: Windows 7, Microsoft Office Word 2010;
- техническая подготовка: Adobe Acrobat Pro DC;
- ответственный за выпуск А. Ю. Сидоренко, технический редактор Е. И. Березич; компьютерный набор и верстка С. М. Глушак, корректор Н. Н. Колодко;
- 3,50 Мб;
- подписано к использованию 25.11.2022;
- доступ в локальной сети, Интернете;
- юридическое лицо, разместившее в локальную сеть и Интернет: учреждение образования «Барановичский государственный университет», 225404 г. Барановичи, ул. Войкова, 21. Тел.: 8 (0163) 64 34 77, e-mail: rig@barsu.by .

В начало

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ

Методические рекомендации по выполнению лабораторных работ предназначены для студентов специальностей: 1-40 05 01 Информационные системы и технологии (по направлениям); 6-05 06 11-01 Информационные системы и технологии.

Оптимизация проектных решений — учебная дисциплина, в которой предусматривается формирование у обучаемых базы владения современными информационными технологиями и практическими навыками постановки, моделирования и решения прикладных оптимизационных задач при помощи персонального компьютера.

Задачи учебной дисциплины:

– изучение основных понятий и этапов оптимизационного моделирования и вычислительного эксперимента;

– изучение прямых и итерационных методов решения оптимизационных задач;

– изучение компьютерных САЕ-пакетов, позволяющих применять оптимизационные методы для решения прикладных задач;

– усвоение навыков разработки алгоритмов для решения оптимизационных задач.

В первой лабораторной работе представлены способы решения нелинейных задач в различных пакетах, а также изложены основные аспекты метода прямого перебора по сетке. При выполнении второй работы студенты приобретают навыки решения оптимизационных задач методом Монте-Карло. В третьей лабораторной работе представлен один из методов нелинейного программирования — метод Хука-Дживса. Четвёртая работа содержит информацию о решении задач методом штрафных функций. При выполнении пятой работы обучающиеся знакомятся с задачами линейного дискретного целочисленного программирования. При выполнении шестой лабораторной работы обучающиеся осваивают эволюционные вычисления. В седьмой лабораторной работе обучающимся предлагается получить теоретические и практические навыки решения задач двухпроцессорного обслуживания. Восьмая лабораторная работа содержит информацию о моделях управления запасами. В девятой лабораторной работе представлены способы решения задач инвестирования.

Методические рекомендации содержат краткие теоретические сведения по каждой рассматриваемой теме. Теоретический материал излагается сжато.

Задания методических рекомендаций выполняются каждым студентом индивидуально. Для этого студент получает одно из заданий по указанному преподавателем варианту.

Лабораторная работа считается выполненной после её защиты. Для этого необходимо представить результаты решения и отчёт, оформленный в MS Word, содержащий тему и цель работы, формулировки заданий, тексты программы, результаты тестирования, выводы по работе.

Лабораторная работа 1
МЕТОД ПРЯМОГО ПЕРЕБОРА ПО СЕТКЕ

Цель работы: изучить метод перебора по сетке, продемонстрировать графическое построение и исследование области поиска, выполнить поиск условного экстремума, сравнить полученные результаты в нескольких компьютерных средах.

Краткие теоретические сведения

Оптимизация — это выбор наилучшего варианта из множества возможных. Если критерий выбора известен и вариантов немного, то решение может быть найдено путем перебора и сравнения всех вариантов. Однако часто бывает так, что число возможных вариантов настолько велико, что полный перебор практически невозможен. В таких случаях приходится формулировать задачу на языке математики и применять специальные методы поиска оптимального решения, т. е. методы оптимизации.

Решение оптимизационных задач проводится в три этапа:

1. Построение математической модели задачи (перевод задачи на математический язык или ее формализация).

2. Нахождение оптимального решения одним из математических методов.

3. Анализ полученного оптимального решения и внедрение его в производство.

Методы оптимизации используют для решения задач математического программирования: линейное, нелинейное, динамическое, целочисленное, стохастическое, а также сетевое планирование и теория игр.

Модель общей задачи математического программирования состоит из:

– целевой функции —

$$Z = f\left(x_1, x_2, \ldots, x_n\right);$$

– ограничений —

$$\begin{cases} \varphi_1\left(x_1, x_2, \ldots, x_n\right) \le b_1 \\ \varphi_2\left(x_1, x_2, \ldots, x_n\right) \le b_2 \\ \qquad \ldots \\ \varphi_m\left(x_1, x_2, \ldots, x_n\right) \le b_m \end{cases} \tag{1.1}$$

где f, φ_1, φ_2, …, φ_m — известные функции;

b_j (j = 1, 2, …, m) — заданные значения.

Функция Z выражает в аналитической форме критерий эффективности в зависимости от планируемых параметров и называется целевой функцией, или критерием оптимальности [10].

Ограничения (1.1) называются технологическими, их правые части представляют собой фиксированные объемы имеющихся в распоряжении ресурсов [10].

Решение задачи математического программирования — это набор значений (x_1, x_2, …, x_n), называемый оптимальным планом, при котором выполняются ограничения, а целевая функция принимает оптимальное значение.

В зависимости от вида целевой функции и ограничений задачи математического программирования делятся на две большие группы — линейные и нелинейные [4].

Задача оптимизации целевой функции может являться как задачей максимизации, так и задачей минимизации целевой функции.

Для нелинейной задачи математического программирования с двумя переменными точка A (a; b) называется точкой максимума функции $Z = f$ (a; b), если найдется такая окрестность точки A, что для всех точек A_0 (a_0; b_0) из этой окрестности выполняется неравенство f (a; b) < f_0 (a_0; b_0) (рис. 1.1).

Также точка A (a; b) называется точкой минимума функции $Z = f$ (a; b), если найдется такая окрестность точки A, что для всех точек A_0 (a_0; b_0) из этой окрестности выполняется неравенство f (a; b) < f_0(a_0; b_0) [11] (*см.* рис. 1.1).

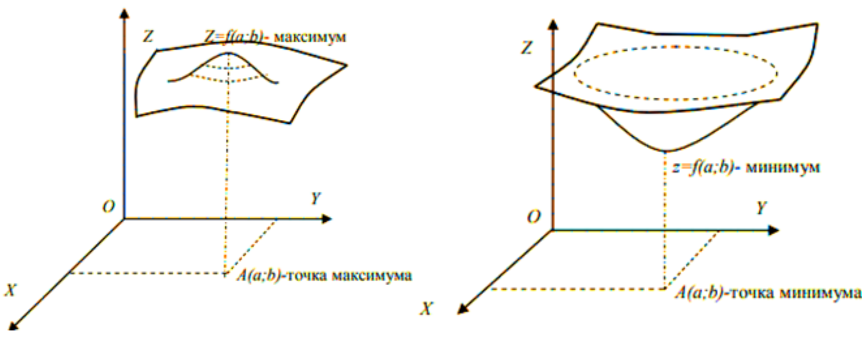

Рисунок 1.1 — Экстремумы функции двух переменных

Поиск экстремума функции.

Пусть задана целевая функция

$$f = x_1^{\ 2} + x_2^{\ 2} + x_1 x_2 - 6x_1. \qquad (1.2)$$

Необходимо найти локальный минимум функции, учитывая ограничения:

$$\begin{cases} x_1 + 2x_2 \geq 1 \\ x_1 + x_2 \leq 2. \end{cases} \qquad (1.3)$$

1. Поиск локального экстремума функции средствами Mathcad.

Для поиска локального минимума в среде Mathcad необходимо:

– задать оптимизируемую функцию;

– задать начальные приближения переменных;

– построить контурный и 3d-графики исследуемой функции;

– задать блок ограничений;

– найти экстремум функции при помощи функций minimize()/maximize();

– найти значение функции при оптимальных параметрах.

Пример поиска минимума функции представлен на рисунке 1.2.

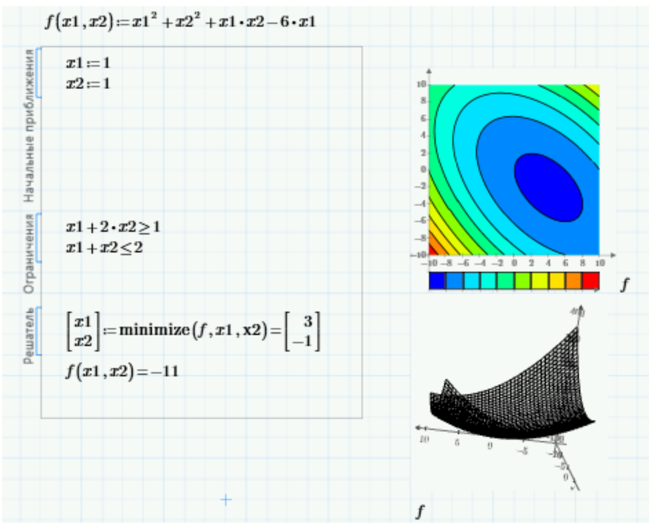

Рисунок 1.2 — Поиск экстремума функции в среде Mathcad

2. Поиск локального экстремума функции средствами MS Excel производится средствами надстройки «Поиск решения» (вкладка «Данные»). Для расчета необходимо ввести ограничения и задать целевую функцию (рис. 1.3).

В поле «Установить целевую ячейку» необходимо указать адрес ячейки, содержащей формулу исходной функции. Далее установить критерий оптимальности (максимум, минимум либо числовое значение). Далее в поле «Изменяя ячейки переменных» указать адреса ячеек, в которых содержатся переменные x_1 и x_2. После этого необходимо задать ограничения в соответствующем окне (рис. 1.4).

	A	B	C
1	x1=		
2	x2=		
3	f=	=B1^2+B2^2+B1*B2-6*B1	
4	ограничения:		
5	=B1+2*B2	>=	1
6	=B1+B2	<=	2

Рисунок 1.3 — Задание целевой функции и ограничений в Excel

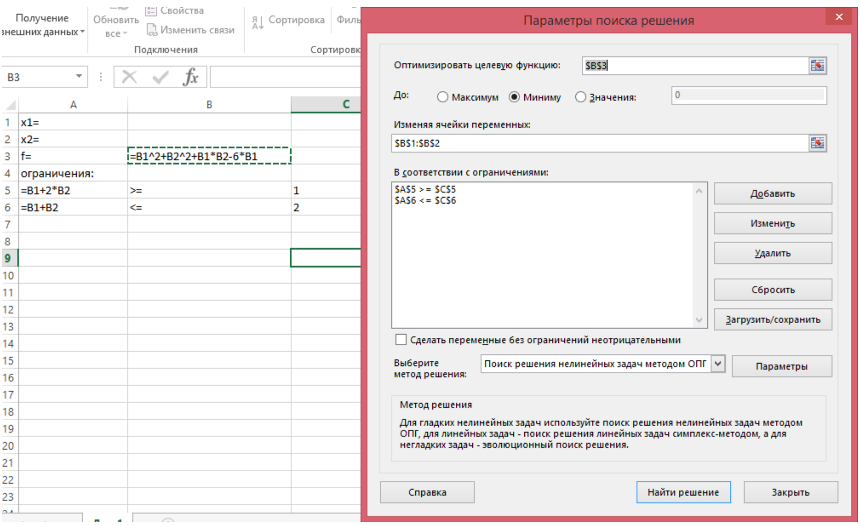

Рисунок 1.4 — Установка параметров поиска решения

	A	B	C
1	x1=	3	
2	x2=	-1	
3	f=	-11	
4	ограничения:		
5	1	>=	1
6	2	<=	2

Рисунок 1.5 — Результат поиска экстремума функции в среде Excel

Результат расчета отображен на рисунке 1.5.

3. Поиск локального экстремума функции средствами MATLAB.

Для поиска локального минимума в среде MATLAB необходимо:

1. Создать *m*-файл, определяющий целевую функцию (рис. 1.6).

```
myfun.m
1     function f = myfun(x)
2  -  f = x(1)^2+x(2)^2+x(1)*x(2)-6*x(1);
3  -  end
```

Рисунок 1.6 — Функция расчета оптимизационной функции в MATLAB

2. Создать скрипт построения графика функции (рис. 1.7).

```
[x1, x2] = meshgrid(-10:0.1:10,-10:0.1:10)
g =x1.^2+x2.^2+x1*x2-6*x1;
surf(x1,x2,g);
```

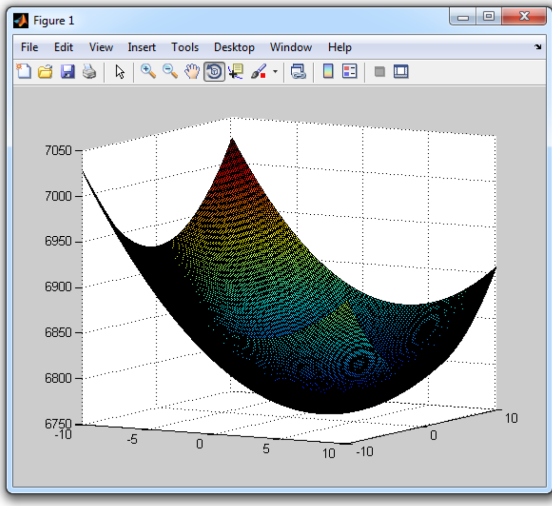

Рисунок 1.7 — Построение графика функции в MATLAB

3. Найти оптимальное значение функции при помощи функции fmincon(). Для этого необходимо записать ограничения в матричной форме $Ax \le b$, где $A = \begin{bmatrix} -1 & -2 \\ 1 & 1 \end{bmatrix}$, $b = \begin{bmatrix} -1 & 2 \end{bmatrix}$.

Далее решение будет иметь вид, представленный на рисунке 1.8.

4. Поиск локального экстремума при помощи метода прямого перебора по сетке.

Алгоритм метода прямого перебора по сетке:

1. Разбить отрезки $[x_{i\min}, x_{i\max}]$, $i = 1, 2, \ldots n$ (где x_i — оптимизируемые параметры) на равные части для построения сетки.
2. Вычислить значения функции в узлах сетки.
3. По полученным значениям функции в узлах сетки методом линейной интерполяции построить линии уровней (линии равных значений).
4. Найти оптимальное (максимальное/минимальное) значение функции из массива полученных значений.

```
>> A = [-1 -2; 1 1];
>> b = [-1; 2];
>> x0 = [1; 1]; % Стартовое значение
>> [x,fval] = fmincon('myfun',x0,A,b)

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

x =

    3.0000
   -1.0000


fval =

  -11.0000
```

Рисунок 1.8 — Оптимальное значение функции в MATLAB

Задания для самостоятельного выполнения

1. Найти экстремум заданной функции согласно варианту (табл. 1.1) в среде Mathcad. Построить график функции и определить интервал, на котором находится экстремум.

2. Найти локальный экстремум функции посредством MS Excel с помощью надстройки «Поиск решения».

3. Найти экстремум функции с помощью среды MATLAB.

4. Разработать приложение поиска экстремума функции двух переменных методом прямого перебора по сетке. Задать область допустимых решений, включив в программу функциональные ограничения в виде неравенств, приведенные в задании. В программном приложении предусмотреть возможность задания области поиска решения, шага сетки и отображения линий уровня оптимизируемой функции.

5. Сравнить полученные результаты в исполняемых средах.

Т а б л и ц а 1.1 — Варианты заданий

Вариант	Функция	Ограничения	Вид экстремума
1	$y = (x_1 - 2)^2 + (x_2 - 1)^2$	$x_1 - 2x_2 \geq 1$ $x_1 + x_2 \leq 3$	Min
2	$y = -x_1^2 - x_2^2 + 2x_1 + 2x_2 + 2$	$x_1 + x_2 \leq 1$ $2x_1 + 2x_2 \geq 0$	Max
3	$y = x_1^2 + x_2^2 - 2{,}4x_1 - 5{,}2x_2$	$2x_1 - 3x_2 \leq 0$ $x_1 + x_2 \geq 1$	Min
4	$y = -0{,}6x_1^2 - 0{,}2x_2^2 + 2x_1 + x_2$	$x_1 + 2x_2 \geq -4$ $x_1 + x_2 \leq -1$	Max
5	$y = 0{,}5x_1^2 + 0{,}5x_2^2 - x_1 - 2x_2 + 5$	$2x_1 + 3x_2 \geq 6$ $x_1 + 4x_2 \leq 10$	Min
6	$y = -x_1^2 - x_2^2 + 2x_1 - x_2$	$x_1 + x_2 \geq 1$ $3x_1 + 3x_2 \leq 5$	Max
7	$y = x_1^2 + x_2^2 - 4x_1 - 2x_2$	$2x_1 + x_2 \geq 4$ $x_1 + 2x_2 \leq 3$	Min
8	$y = -x_1^2 - x_2^2 - x_1 - 3x_2$	$x_1 + 2x_2 \geq 0$ $x_1 + 4x_2 \leq 5$	Max

Окончание табл. 1.1

Вариант	Функция	Ограничения	Вид экстремума
9	$y = 2x_1^2 + 2x_2^2 - 4x_1 - 6x_2$	$-x_1 + 2x_2 \geq 3$ $x_1 + x_2 \leq 3$	Min
10	$y = x_1^2 + x_2^2 - 6x_1 - 10x_2$	$2x_1 + x_2 \leq 10$ $x_1 + x_2 \geq 5$	Min
11	$y = -0{,}6x_1^2 - 0{,}4x_2^2 + 3x_1 + 6x_2$	$-x_1 + x_2 \leq 4$ $x_1 + x_2 \geq 7$	Max
12	$y = x_1^2 + x_2^2 - x_1 - x_2$	$x_1 + x_2 \geq 0$ $x_1 + 2x_2 \leq 0$	Min
13	$y = x_1^2 + x_2^2 + x_1 + 7x_2$	$2x_1 - 2x_2 \geq 0$ $x_1 - x_2 \leq 0$	Min
14	$y = -x_1^2 - x_2^2 + 3x_1 - 6x_2$	$x_1 - x_2 \geq 4$ $4x_1 + x_2 \leq 0$	Max
15	$y = x_1^2 + x_2^2 - 6x_1 - 2x_2 + 45$	$-x_1 + 9x_2 \geq 0$ $x_1 - x_2 \leq 1$	Min

Контрольные вопросы

1. Что собой представляет оптимизация?
2. Какие существуют этапы решения оптимизационных задач?
3. Как выглядит модель общей задачи математического программирования?
4. Чем задачи линейного программирования отличаются от задач нелинейного программирования?
5. Что представляет собой максимум функции?
6. Что представляет собой минимум функции?
7. Опишите алгоритм прямого перебора по сетке.
8. Приведите алгоритм действий при нахождении экстремума функции в среде MS Excel.
9. Приведите порядок действий при нахождении экстремума функции по средствам Mathcad.
10. Приведите порядок действий при нахождении экстремума функции по средствам MATLAB.

Лабораторная работа 2
МЕТОД МОНТЕ-КАРЛО

Цель работы: изучить метод Монте-Карло, продемонстрировать графическое построение и исследование области поиска, выполнить поиск экстремума.

Краткие теоретические сведения

Метод Монте-Карло определяется как способ моделирования случайных величин в целях вычисления характеристик их распределений. Этот метод (как и вся теория вероятностей) возник при попытках людей улучшить шансы в играх. Название метода происходит от названия города Монте-Карло — столицы европейского игорного бизнеса, где играют в рулетку — одно из самых простых устройств для получения случайных чисел.

Метод Монте-Карло оказал и продолжает оказывать существенное влияние на развитие методов вычислительной математики (например, развитие методов численного интегрирования) и при решении многих задач успешно сочетается с другими вычислительными методами и дополняет их. Его применение оправдано в первую очередь в тех задачах, которые допускают теоретико-вероятностное описание. Это объясняется как естественностью получения ответа с некоторой заданной вероятностью, так и существенным упрощением процедуры решения [2].

Общая схема метода Монте-Карло.

1. При использовании методов Монте-Карло значения параметров x_i', $i = 1, 2, \dots n$ (где x_i — оптимизируемые параметры) устанавливаются случайными числами из заданного диапазона $[x_{i\min}, x_{i\max}]$.

2. Генерируется случайное число $r \in (0; 1)$. Задается количество итераций k, вычисляется разность граничных значений заданного диапазона.

3. Вычисляется значение параметра x_i :

$$x_i = x_i' + (x_{i\max} - x_{i\min})r. \qquad (2.1)$$

4. Вычисляется значение оптимизируемой функции в найденных по формуле (2.1) параметрах x_i, учитывая ограничения, и после заключительной итерации находится экстремум из вычисленных значений функций [9].

Метод Монте-Карло относится к числу универсальных, поскольку позволяет решать многоэкстремальные задачи общего вида с отысканием приближенного глобального экстремума. Основной недостаток метода заключается в необходимости проведения большого числа испытаний для получения решения, достаточно близкого к оптимальному [8].

Задания для самостоятельного выполнения

1. Найти экстремум заданной функции методом Монте-Карло по вариантам (*см.* табл. 1.1). Число испытаний $k = 121$ и более. Применить датчик случайных чисел с равномерным распределением.
2. Построить область поиска экстремума и линии уровней оптимизируемой функции с учетом параметрических функциональных ограничений. Нанести на график с линиями уровней вычисляемые точки и выделить экстремум функции.
3. Сравнить результаты решений, полученные методом Монте-Карло и методом прямого перебора по сетке.

Контрольные вопросы

1. Дайте определение понятию «оптимизация».
2. Какие существуют этапы решения оптимизационных задач?
3. Как выглядит модель общей задачи математического программирования?
4. Назовите отличие постановки задач линейного программирования от постановки задач нелинейного программирования.
5. Что представляет собой максимум функции?
6. Что представляет собой минимум функции?
7. Сравните метод Монте-Карло с методом прямого перебора по сетке.

Лабораторная работа 3
МЕТОД ХУКА-ДЖИВСА

Цель работы: изучить метод Хука-Дживса, продемонстрировать графическое построение и исследование области поиска, выполнить поиск экстремума.

Краткие теоретические сведения

Метод Хука-Дживса представляет собой комбинацию исследующего поиска с циклическим изменением переменных и ускоряющего поиска по образцу. Цель исследующего поиска — выявление локального поведения целевой функции и определение направления ее убывания или возрастания. Эта информация используется при поиске по образцу вдоль направления убывания или возрастания целевой функции. Метод Хука-Дживса применяется для решения задачи минимизирования функции без учета ограничений [14].

Исследующий поиск начинается из некоторой начальной точки, называемой старым базисом. В качестве множества направлений поиска выбирается множество координатных направлений. Задается величина шага, которая может быть различной для разных координатных направлений. Фиксируется первое координатное направление, делается шаг в сторону увеличения соответствующей переменной. Если значение исходной функции $f(x)$ в пробной точке меньше (больше) значения функции в исходной точке, то шаг считается удачным. В противном случае из исходной точки делается шаг в противоположном направлении с последующей проверкой поведения функции. Если и в этом случае не происходит уменьшения (увеличения) функции, то происходит уменьшение шага, процедура повторяется. Исследующий поиск по данному направлению заканчивается, когда текущая величина шага становится меньше некоторой величины. После перебора всех координат исследующий поиск завершается, полученная точка называется новым базисом [11].

Модифицированный метод Хука-Дживса.

Модифицированный метод применяется для задач условной оптимизации и предполагает присвоение целевой функции очень большого (маленького) значение там, где ограничения нарушаются.

Алгоритм модифицированного метода Хука-Дживса при поиске условного минимума (максимума) функции:

1. Выбрать начальную базисную точку b_0 и шаг длиной h_j для каждой переменной x_j, где $j = 1, 2, \ldots, n$ — количество переменных.

2. Вычислить значение функции в базисной точке b_0 в целях получения сведений о локальном поведении функции. Эти сведения будут использоваться для нахождения перспективного направления поиска по образцу, с помощью которого можно надеяться достичь большего убывания (возрастания) значения функции. Исследующий поиск в базисной точке b_0 проводится следующим образом:

2.1. Вычисляется значение функции в базисной точке b_0.

2.2. Каждая переменная по очереди изменяется прибавлением длины шага. Нужно проверить, принадлежат ли точки области ограничений. Если нет, то целевой функции присваивается очень большое (маленькое) значение. Если точки принадлежат области ограничений, то вычисляется значение функции в точке $b_0 + h_1e_1$, где e_1 — единичный вектор в направлении оси x. Если это приводит к уменьшению (увеличению) значения функции, то b_0 заменяется на $b_0 + h_1e_1$. В противном случае вычисляется значение функции в точке $b_0 - h_1e_1$, и если ее значение уменьшилось (увеличилось), то b_0 заменяем на $b_0 - h_1e_1$. Если ни один из проделанных шагов не приводит к уменьшению (увеличению) значения функции, то точка b_0 остается неизменной, рассматриваются изменения в направлении оси y, т. е. находится значение функции $b_0 + h_2e_2$ и $b_0 - h_2e_2$. Когда будут рассмотрены все n переменных, то получим базисную точку b_1.

2.3. Если $b_1 = b_0$, т. е. уменьшение (увеличение) функции не было достигнуто, то исследование повторяется вокруг той же базисной точки b_0, но с уменьшенной длиной шага. На практике удовлетворительным является уменьшение шага (шагов) в десять раз от начальной длины.

2.4. Если значение функции в точке b_1 меньше (больше) значения функции в точке b_0, то производится поиск по образцу, т. е. поиск в направлении, заданном минимальным (максимальным) значением функции.

2.5. При поиске по образцу используется информация, полученная в процессе исследования, и минимизация (максимизация) функции завершается поиском в направлении, заданном образцом по пунктам 2.1.—2.4., принимая за базисную точку b_1. Завершить этот процесс, когда длина шага (длины шагов) будет уменьшена до заданного малого значения [10].

Задания для самостоятельного выполнения

1. Разработать программу поиска оптимума функции методом Хука-Дживса для функции по вариантам (*см.* табл. 1.1). Координаты начальной точки поиска должны быть заданы двумя способами: ввод пользователем и случайным образом. Осуществить возможность ввода с клавиатуры шага и точности вычисления переменных.

2. Построить область поиска экстремума и линии уровней оптимизируемой функции с учетом параметрических функциональных ограничений. Нанести на график с линиями уровней стартовую точку, путь поиска оптимума и выделить точку вычисленного экстремума функции.

Контрольные вопросы

1. Дайте определение понятию «оптимизация».
2. Какие существуют этапы решения оптимизационных задач?
3. Как выглядит модель общей задачи математического программирования?
4. Чем задачи линейного программирования отличаются от задач нелинейного программирования?
5. Что представляет собой максимум функции?
6. Что представляет собой минимум функции?
7. Суть метода Хука-Дживса.
8. Что собой представляет модифицированный метод Хука-Дживса?
9. Достоинства и недостатки метода Хука-Дживса.
10. Алгоритм метода Хука-Дживса.
11. Сравните метод Хука-Дживса с методом Монте-Карло.
12. Сравните метод Хука-Дживса с методом прямого перебора по сетке.

Лабораторная работа 4

МЕТОД ШТРАФНЫХ ФУНКЦИЙ

Цель работы: изучить метод штрафных функций, продемонстрировать графическое построение и исследование области поиска, выполнить поиск экстремума.

Краткие теоретические сведения

Методы штрафных функций относятся к группе непрямых методов решения задач нелинейного программирования. Идея метода заключается в преобразовании исходной задачи с ограничениями в последовательность задач безусловной оптимизации. С помощью функций, задающих ограничения, строится так называемый штраф, который добавляется к целевой функции исходной задачи так, что нарушение какого-либо из ограничений становится невыгодным. Вспомогательные функции получаются таким образом, чтобы ограничения в явном виде в задаче оптимизации не фигурировали. Это обеспечивает возможность применения методов безусловной оптимизации. Штрафная функция подбирается так, чтобы она могла препятствовать выходу за пределы области допустимых решений и не имела существенного влияния на результат решения [20].

Использование метода может быть целесообразно по следующим причинам:

1) существуют такие задачи, что некоторые из их ограничений не являются «жесткими», т. е. ограничения, выполнение которых носит не строго обязательный характер;

2) применение штрафных функций позволяет заменить исходную задачу оптимизации со сложной системой ограничений на задачу без них. Это позволяет использовать для численного решения задач методы типа градиентных или покоординатного спуска;

3) на практике могут возникнуть задачи с несовместными системами ограничений. Применение штрафных функций позволяет получить псевдо (обобщённое) решение;

4) метод может быть использован для решения многокритериальных задач [19].

Описание метода штрафных функций.

Методы штрафных функций можно разделить на два класса: параметрические и непараметрические. Параметрические методы характеризуются наличием одного или нескольких параметров, входящих в структуру штрафной функции в качестве весовых коэффициентов [1].

Например, требуется найти минимум функции $f(x)$ при ограничениях: $\varphi_j(x) \geq 0$, $j = 1, 2, \ldots, m$, где m — число ограничений. Введем функцию:

$$D(x) = f(x) + P(x).$$

Здесь $P(x)$ выступает в роли штрафа. Штраф можно выбрать в следующем виде:

$$P(x) = r\sum_{j=1}^{m} \frac{1}{\varphi_j(x)} \quad (4.1)$$

или

$$P(x) = r\sum_{j=1}^{m} \varphi_j^{\,2}(x), \quad (4.2)$$

где r — некоторый положительный параметр [19].

Функция штрафа должна быть равна нулю в допустимой области и на ее границах и стремиться к увеличению при нарушении ограничений, т. е. налагать штраф.

Алгоритм метода штрафных функций.

1. Выбрать начальную точку, штрафной параметр r, коэффициент уменьшения $\beta < 1$ и точность ε.
2. Построить функцию без ограничений, используя штраф.
3. При исходной точке x_k решить задачу любым методом безусловной оптимизации.
4. Если штраф меньше точности, тогда необходимо установить $r_{k+1} = r_k / \beta$. Изменить $k = k + 1$ и перейти к первому шагу $(k + 1)$-й итерации.

Пример преобразования задачи условной оптимизации в задачу безусловной оптимизации с использованием метода штрафных функций.

Пусть требуется найти минимум целевой функции (1.2) с учетом ограничений (1.3). Зададим начальные параметры: $r = 1$, $\beta = 10$, $\varepsilon = 0{,}001$.

Для преобразования необходимо ввести штрафную функцию, которая с учетом ограничений (1.3) будет выглядеть следующим образом.

Для вида штрафа (4.1):

$$P(x)=1\left(\frac{1}{x_1+2x_2-1}+\frac{1}{2-x_1-x_2}\right).$$

Для вида штрафа (4.2):

$$P(x)=1\left((x_1+2x_2-1)^2+\left(2-x_1-x_2\right)^2\right).$$

С учетом штрафа функция будет выглядеть следующим образом:

$$f=x_1^{\,2}+x_2^{\,2}+x_1x_2-6x_1+P(x). \qquad (4.3)$$

Далее необходимо найти минимум функции 4.3 любым методом безусловной оптимизации.

Если штраф меньше точности (0,01 в нашем случае), то решение достигнуто, в противном случае необходимо уменьшить r, r = 1 / 10, переписать штрафную функцию и продолжить решение.

Задания для самостоятельного выполнения

1. Разработать программу поиска экстремума для функции по вариантам (*см.* табл. 1.1), преобразуя задачу условной оптимизации в задачу безусловной оптимизации, используя метод штрафных функций. Дальнейший поиск осуществить методом Хука-Дживса. Продемонстрировать решение задачи с генерацией стартовой точки в недопустимой области решений.

2. Построить область поиска экстремума и линии уровней оптимизируемой функции с учетом параметрических функциональных ограничений. Нанести на график с линиями уровней путь поиска оптимума и точку с экстремумом функции.

Контрольные вопросы

1. Дать определение понятию «оптимизация».
2. Какие существуют этапы решения оптимизационных задач?
3. Как выглядит модель общей задачи математического программирования?
4. Чем задачи линейного программирования отличаются от задач нелинейного программирования?
5. Что представляет собой максимум функции?
6. Что представляет собой минимум функции?
7. В чем разница условного и безусловного экстремума?
8. Сравнение метода штрафных функций с методом Хука-Дживса.
9. Сравнение метода штрафных функций с методом Монте-Карло.
10. Сравнение метода штрафных функций с методом прямого перебора по сетке.
11. Виды штрафов в методе штрафных функций.

Лабораторная работа 5

МЕТОД ВЕТВЕЙ И ГРАНИЦ

Цель работы: овладеть теоретическими и практическими навыками решения задач линейного дискретного целочисленного программирования.

Краткие теоретические сведения

Дискретное программирование сформировалось как самостоятельная и важная часть математического программирования в конце 60-х годов XX века. Основная задача дискретного программирования — выбор наилучшего варианта из конечного, возможно, очень большого их числа. Возникающие при этом экстремальные задачи имеют ряд особенностей, которые не встречаются в таких стандартных задачах математического программирования, как линейные, выпуклые или многокритериальные задачи [21].

Задачи дискретной оптимизации имеют конечное множество допустимых решений, которые теоретически можно перебрать и выбрать наилучшее (дающее минимум или максимум целевой функции). Практически же зачастую это бывает неосуществимо даже для задач небольшой размерности.

В методах неявного перебора делается попытка так организовать перебор, используя свойства рассматриваемой задачи, чтобы отбросить часть допустимых решений. Наибольшее распространение среди схем неявного перебора получил метод ветвей и границ, в основе которого лежит идея последовательного разбиения множества допустимых решений. На каждом шаге метода элементы разбиения (подмножества) подвергаются анализу: содержит данное подмножество оптимальное решение или нет. Если рассматривается задача на минимум, то проверка осуществляется путем сравнения нижней оценки значения целевой функции на данном подмножестве с верхней оценкой функционала. В качестве оценки сверху используется значение целевой функции на некотором допустимом решении. Допустимое решение, дающее наименьшую верхнюю оценку, называют рекордом. Если оценка снизу целевой функции на данном подмножестве не меньше оценки сверху, то рассматриваемое подмножество не содержит решения лучше рекорда и может быть отброшено. Если значение целевой функции на очередном решении меньше рекордного, то происходит смена рекорда. Подмножество решений просмотрено, если установлено, что оно не содержит решения лучше рекорда [16].

Если просмотрены все элементы разбиения, алгоритм завершает работу, а текущий рекорд является оптимальным решением. В противном случае среди непросмотренных элементов разбиения выбирается множество, являющееся в определенном смысле перспективным. Оно подвергается разбиению (ветвлению). Новые подмножества анализируются по описанной выше схеме. Процесс продолжается до тех пор, пока не будут просмотрены все элементы разбиения.

Описание метода ветвей и границ.

Пусть рассматривается задача вида

$$f(x) \xrightarrow[x \in D]{} \min,$$

где $f(x)$ — вещественная функция;

D — конечное множество допустимых решений.

Пусть $d \subseteq D$. Функцию $b(d)$, ставящую в соответствие множеству d разбиение его на подмножества $d_1, ..., d_N$, $N > 1$, будем называть ветвлением.

Вещественная функция $H(d)$ называется нижней границей для d, если:

1) $H(d) \to \min\limits_{x \in D} f(x)$;

2) на одноэлементном множестве $\{x\}$ верно равенство $H(\{x\}) = f(x)$.

Алгоритм, реализующий метод ветвей и границ, состоит из последовательности однотипных шагов. На каждом шаге известен рекорд x_0 и подмножества $t_1, t_2, ..., t_L$ непросмотренных решений. В начале работы алгоритма $L = 1$, $t_1 = D$, x_0 — произвольный элемент множества D или пустое множество (на пустом множестве положим значение функционала равным бесконечности).

На каждом шаге алгоритм начинает работу с проверки элементов разбиения. Пусть проверяется множество t_j. Множество t_j отсекается в одном из двух последовательно проверяемых случаев:

а) если $H(t_j) \geq f(x_0)$;

б) если $H(t_j) < f(x_0)$ и найден такой элемент $y_j \in t_j$, что $f(y_j) = \min\limits_{x \in D} f(x) = H(t_j)$.

Во втором случае происходит смена рекорда $x_0 = y_j$.

Пусть $t_1, t_2, ..., t_M$ ($M \le L$) — неотсеченные множества (будем считать, что отсечены множества с номерами $M + 1, ..., L$).

В случае $M = 0$ алгоритм заканчивает работу, в качестве решения задачи принимается рекорд x_0. При $M \ge 1$ среди множеств $t_1, ..., t_M$ выбирается множество для нового ветвления. Пусть таковым является множество t_1. Тогда осуществляется ветвление $b(t_1) = (d_1, ..., d_N)$, в результате которого получаем список множеств $d_1, ..., d_N, t_2, ..., t_M$. Эти множества нумеруются числами от 1 до L, начинается новый шаг алгоритма.

Описанная последовательность действий является общей схемой метода ветвей и границ для решения задач на минимум. При решении конкретной задачи следует указать способы построения нижней и верхней оценок, метод ветвления, а также правило выбора перспективного множества для разбиения.

При определении «перспективного» элемента разбиения в основном применяются две схемы: одновременного (многостороннего) и одностороннего ветвления. При одновременном ветвлении функция может быть применена к любому элементу разбиения. Часто в качестве такого элемента выбирается подмножество t_k с минимальной нижней границей:

$$H(t_k) = \min_{1 \le i \le L} H(t_i).$$

При одностороннем ветвлении номер разбиваемого подмножества известен заранее. В этом случае, не ограничивая общности, можно считать, что «перспективным» является подмножество t_1. При односторонней схеме ветвления нет необходимости запоминать все элементы разбиения, достаточно иметь информацию о первом элементе разбиения и объединении остальных элементов.

Разбиения множеств решений (ветвление) удобно представлять в виде дерева решений. На рисунках 5.1 и 5.2 приведены примеры одновременной и односторонней схем ветвления соответственно. Каждая вершина дерева соответствует некоторому подмножеству решений. Дуги, исходящие из вершины, означают, что на некотором шаге это подмножество отсечь не удалось, оно было разбито на подмножества. Вершины, в которые входят эти дуги, соответствуют подмножествам, полученным в результате ветвления. Зачеркнутые висячие вершины означают отсеченные подмножества. Незачеркнутые висячие вершины соответствуют непросмотренным множествам, среди которых на следующем шаге алгоритма выбирается подмножество для дальнейшего ветвления.

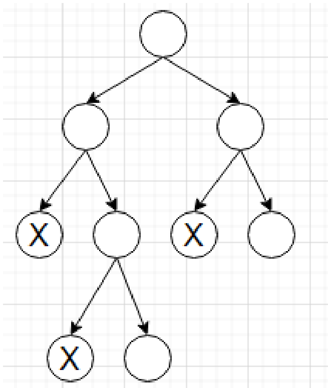

Рисунок 5.1 — Пример одновременной схемы ветвления

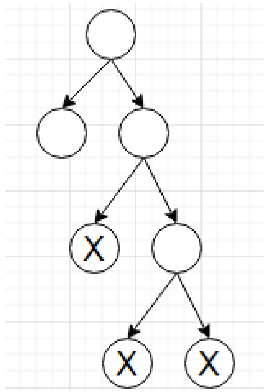

Рисунок 5.2 — Пример односторонней схемы ветвления

В схеме одностороннего ветвления выбирается первая (левая) вершина на нижнем уровне, а для схемы одновременного ветвления такой вершиной может быть любая. Алгоритм заканчивает работу, если зачеркнуты все висячие вершины дерева ветвлений.

Графическое представление метода ветвей и границ иллюстрирует его суть — отсечение ветвей дерева поиска, которое осуществляется на основании сравнения нижней границы и значения функционала на рекорде. Это объясняет название метода [15].

Пример решения задачи методом ветвей и границ.

Рассмотрим задачу для случая обработки четырех партий заготовок, время переналадки технологической линии для которых указано в таблице 5.1. Для обработки на технологической линии поступило четыре партии заготовок. При переходе от обработки одной партии к обработке следующей необходимо выполнять переналадку технологической линии, для обработки всех партий необходимо три такие переналадки. Задача состоит в определении такого порядка запуска партий заготовок на обработку, при котором суммарное время переналадок было бы минимальным.

Т а б л и ц а 5.1 — Заданное время переналадок для заготовок

Партия заготовок / время переналадок	1	2	3	4
1	∞	9	8	4
2	7	∞	4	5
3	5	3	∞	6
4	4	7	1	∞

1-й этап. Нахождение минимума по строкам и редукция строк. Необходимо привести таблицу по строкам, для этого найти минимальное значение в каждой строке (d_i) и выписать его в отдельный столбец. Найденные значения d_i называются константами приведения для строк. Далее необходимо произвести редукцию строк — из каждого элемента в каждой строке вычесть соответствующее ей значение минимума (табл. 5.2).

Т а б л и ц а 5.2 — Нахождение констант приведения для строк и редукция строк

Партия заготовок / время переналадок	1	2	3	4	d_i
1	∞	5	4	0	4
2	3	∞	0	1	4
3	2	0	∞	3	3
4	3	6	0	∞	1

2-й этап. Нахождение минимума по столбцам и редукция столбцов. Необходимо найти минимальные значения в каждом столбце (d_j). Эти минимумы (включая нули) выписать в отдельную строку. Найденные значение d_j называются константами приведения для столбцов. Далее необходимо произвести редукцию столбцов — из каждого элемента в каждом столбце вычесть соответствующее ей значение минимума (табл. 5.3).

Т а б л и ц а 5.3 — Нахождение констант приведения для столбцов и редукция столбцов

Партия заготовок	1	2	3	4
1	∞	5	4	0
2	1	∞	0	1
3	0	0	∞	3
4	1	6	0	∞
d_j	2	0	0	0

3-й этап. Нахождение корневой нижней границы. На этом этапе следует определить корневую локальную нижнюю границу (H_k) длины маршрута. Для этого необходимо суммировать константы приведения d_i и d_j. Формула для вычисления локальной нижней границы в корневой (стартовой) точке решения следующая:

$$H_0 = \sum d_i + \sum d_j.$$

$$H_0 = (4 + 4 + 3 + 1) + (2 + 0 + 0 + 0) = 12 + 2 = 14.$$

Ход решения задачи коммивояжера удобно представить в виде графа, где вершинами будут ключевые решения по включению/невключению в итоговый маршрут тех или иных отрезков пути, а ребра, показывающие ход решения, образуют ветви альтернативных вариантов маршрута. Пока есть только одна вершина — «корень» дерева решения, куда мы вписываем значение локальной нижней границы $H_0 = 14$.

4-й этап. Вычисление оценок нулевых клеток. Для каждой нулевой клетки преобразованной матрицы находим «оценку» (p_{ij}). Ею будет сумма минимума по строке и минимума по столбцу, на пересечении которых находится данная клетка с нулем. При этом сама нулевая клетка, для которой вычисляется оценка, и найденные ранее d_i и d_j не учитываются (но другие нулевые клетки, если они есть в рассматриваемых строке и столбце, учитывать нужно; также учитываются ячейки с ∞). Полученную оценку необходимо записать рядом с нулем, в скобках (табл. 5.4).

Т а б л и ц а 5.4 — Вычисление оценок нулевых клеток

Партия заготовок	1	2	3	4
1	∞	5	4	0 (5)
2	1	∞	0(5)	1
3	0 (1)	0 (5)	∞	3
4	1	6	0 (1)	∞

5-й этап. Выбор нулевой клетки с максимальной оценкой. Это делается для того, чтобы избежать максимального удлинения маршрута, которое появится, если не выбрать такую ячейку. В нашем случае максимальную оценку имеет клетка 1—4.

6-й этап. Редукция матрицы. На этом этапе решение задачи начинает ветвиться. Возникает развилка с парой альтернативных вариантов: ветвь решения, где мы включаем в маршрут выбранный отрезок пути (1—4), и ветвь, где мы его в маршрут не включаем. Для начала рассмотрим первую ветвь решения, предусматривающую включение отрезка в маршрут. Для проверки этого варианта необходимо провести редукцию матрицы. Делается это следующим образом: выбрав клетку с максимальной оценкой, вписываем в нее ∞, затем полностью вычеркиваем соответствующие ей строку и столбец (табл. 5.5).

Т а б л и ц а 5.5 — Вычисление оценок нулевых клеток

Партия заготовок	1	2	3	d_i
2	1	∞	0	0
3	0	0	∞	0
4	1	6	0	0
d_j	0	0	0	

7-й этап. Вычисление нижней границы первой ветви. Затем вычисляется значение локальной нижней границы: $H_k = H_{k-1} + \sum d_i + \sum d_j$.

$H_1 = 14 + (0 + 0 + 0) + (0 + 0 + 0) = 14$. Полученное значение — локальная нижняя граница для данной ветви решения.

8-й этап. Вычисление нижней границы второй ветви. Далее необходимо проверить вторую ветвь — вычислить для нее значение локальной нижней границы. В этом случае не исключим из таблицы отрезок 4—1 и локальная нижняя граница будет равна сумме предыдущей локальной нижней границы и максимальной оценки (т. е. оценки той нулевой клетки, которую мы выбрали на предыдущем шаге): $H_k^* = H_{k-1} + p_{ij}$.

Таким образом, локальная нижняя граница для второй ветви (пометим ее звездочкой) составит: $H_1^* = 14 + 5 = 19$. Данная ветвь решения увеличивает минимально возможную длину итогового маршрута до значения 19.

9-й этап. Построение маршрута. Построенный маршрут будет иметь вид, представленный на рисунке 5.3.

Рисунок 5.3 — Вид дерева после первого этапа

10-й этап. Выбор действия. Если полный маршрут еще не найден, продолжаем решение задачи, если найден — переход к этапу 11.

Если все еще не найдены все отрезки пути, то решение продолжается, здесь возможны три варианта:

1) выбор ветви, включающей рассматриваемый отрезок. В этом случае решение задачи продолжается с пункта 4;

2) выбор ветви, не включающей рассматриваемый отрезок. Такой вариант предусматривает исключение из итогового маршрута искомого отрезка, для чего в соответствующую ему клетку таблицы необходимо поставить ∞, затем ввернуться к пункту 1 и продолжить решение задачи;

3) выбор другой ветви. Здесь необходимо вернуться к соответствующим этой ветви этапу решения и таблице с данными.

11-й этап. Построение полного маршрута и определение его длины. Найдя все отрезки пути, остается только соединить их между собой и рассчитать общую длину пути (стоимость поездки по этому маршруту, затраченное время и т. д.) [15].

В представленной задаче путь равен: 1 — 4 — 3 — 2 — 1. Общая длина пути: $L = \sum C_{ij} = 15$. Дерево будет иметь вид, изображенный на рисунке 5.4.

Рисунок 5.4 — Итоговый вид дерева

Задания для самостоятельного выполнения

Разработать приложение для решения технологической задачи методом ветвей и границ. Отобразить итерационное решение и нарисовать дерево ветвления вычислительного процесса.

Для обработки на технологической линии поступило семь партий заготовок. При переходе от обработки одной партии к обработке следующей необходимо выполнять переналадку технологической линии, для обработки всех партий необходимо шесть таких переналадок. Задача состоит в определении такого порядка запуска партий заготовок на обработку, при котором суммарное время переналадок было бы минимальным.

Вариант 1.

Партия заготовок / время переналадок	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	∞	13	14	6	8	4	12
2	12	∞	12	16	15	7	7
3	11	10	∞	8	8	5	17
4	8	8	5	∞	10	9	8
5	6	8	9	7	∞	4	9
6	7	6	15	14	16	∞	10
7	9	14	8	15	7	12	∞

Вариант 2.

Партия заготовок / время переналадок	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	∞	8	9	7	8	9	6
2	12	∞	11	4	7	4	6
3	3	5	∞	6	7	4	7
4	6	2	3	∞	4	9	9
5	7	4	5	7	∞	8	8
6	6	3	9	5	7	∞	10
7	5	5	8	12	11	11	∞

Вариант 3.

Партия заготовок	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	∞	11	10	17	6	8	9
2	9	∞	14	15	7	8	9
3	7	9	∞	5	8	4	5
4	6	10	7	∞	5	8	6
5	15	12	6	9	∞	7	7
6	7	14	7	8	7	∞	6
7	9	11	7	8	10	5	∞

Вариант 4.

Партия заготовок	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	∞	7	8	9	7	9	6
2	13	∞	8	14	8	8	7
3	14	5	∞	6	7	5	6
4	4	13	4	∞	13	14	5
5	6	11	5	12	∞	17	4
6	17	7	7	11	2	∞	10
7	5	6	6	16	8	9	∞

Вариант 5.

Партия заготовок	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	∞	6	7	8	13	14	8
2	7	∞	5	6	8	9	15
3	13	14	∞	8	9	9	6
4	12	7	4	∞	5	5	7
5	11	6	3	3	∞	14	8
6	14	5	12	14	11	∞	9
7	3	14	11	6	7	8	∞

Вариант 6.

Партия заготовок	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	∞	11	9	11	12	6	7
2	6	∞	12	8	16	13	5
3	15	7	∞	13	7	15	4
4	13	14	8	∞	14	6	4
5	6	12	14	9	∞	5	5
6	8	7	13	12	8	∞	6
7	7	9	8	2	11	7	∞

Вариант 7.

Партия заготовок	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	∞	9	5	6	12	3	9
2	13	∞	14	7	14	4	8
3	14	8	∞	8	15	5	9
4	11	7	13	∞	16	6	12
5	10	6	11	9	∞	7	11
6	12	5	2	9	6	∞	8
7	9	14	13	8	7	9	∞

Вариант 8.

Партия заготовок	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	∞	11	10	12	6	7	8
2	16	∞	5	6	11	8	8
3	11	10	∞	11	5	6	7
4	6	8	13	∞	7	6	8
5	5	7	8	9	∞	5	7
6	14	6	7	14	11	∞	6
7	6	15	6	8	10	7	∞

Вариант 9.

Партия заготовок	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	∞	5	6	11	10	8	9
2	4	∞	7	8	13	10	7
3	5	9	∞	9	8	12	7
4	6	6	7	∞	7	8	8
5	7	9	9	4	∞	6	7
6	8	8	10	11	6	∞	5
7	5	9	7	4	5	6	∞

Вариант 10.

Партия заготовок	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	∞	6	13	5	6	6	7
2	11	∞	12	4	7	8	15
3	12	9	∞	11	10	9	13
4	11	12	3	∞	13	13	7
5	13	13	14	6	∞	15	9
6	14	12	11	10	13	∞	8
7	5	6	7	8	9	7	∞

Вариант 11.

Партия заготовок	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	∞	14	15	6	7	8	14
2	13	∞	5	8	9	8	11
3	5	14	∞	7	8	7	15
4	14	6	6	∞	7	6	11
5	6	7	12	3	∞	8	10
6	5	8	13	4	6	∞	9
7	7	5	12	13	5	4	∞

Вариант 12.

Партия заготовок	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	∞	9	8	7	7	8	9
2	5	∞	6	7	6	5	4
3	8	9	∞	5	12	4	5
4	5	8	7	∞	6	5	4
5	6	8	4	2	∞	11	12
6	13	6	6	6	5	∞	10
7	4	5	8	5	4	9	∞

Вариант 13.

Партия заготовок	∞	13	14	6	8	4	12
1	12	∞	12	16	15	7	7
2	11	10	∞	8	8	5	17
3	8	8	5	∞	10	9	8
4	6	8	9	7	∞	4	9
5	7	6	15	14	16	∞	10
6	9	14	8	15	7	12	∞
7	∞	13	14	6	8	4	12

Вариант 14.

Партия заготовок	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	∞	8	9	7	8	9	6
2	12	∞	11	4	7	4	6
3	3	5	∞	6	7	4	7
4	6	2	3	∞	4	9	9
5	7	4	5	7	∞	8	8
6	6	3	9	5	7	∞	10
7	5	5	8	12	11	11	∞

Вариант 15.

Партия заготовок	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	∞	11	10	17	6	8	9
2	9	∞	14	15	7	8	9
3	7	9	∞	5	8	4	5
4	6	10	7	∞	5	8	6
5	15	12	6	9	∞	7	7
6	7	14	7	8	7	∞	6
7	9	11	7	8	10	5	∞

Контрольные вопросы

1. Постановка задачи дискретного программирования.
2. Методы решения задач дискретного программирования.
3. Сущность метода ветвей и границ.
4. Дать определение нижней границы, точной нижней границы числового множества.
5. Почему сумма констант приведения матрицы является нижней границей множества длин гамильтоновых контуров?
6. Почему в качестве разбивающей принимается дуга нулевой длины?
7. Почему среди нулевых в качестве разбивающей принимается дуга, сумма дополнительных констант приведения для которой максимальна?

Лабораторная работа 6

ЭВОЛЮЦИОННЫЙ АЛГОРИТМ

Цель работы: овладеть теоретическими и практическими навыками решения задач эволюционными алгоритмами.

Краткие теоретические сведения

Эволюционные вычисления — это метод оптимизации, базирующийся на эволюции популяции особей, в процессе которой определяется максимальное значение целевой функции. Этот метод применим для задач оптимизации плохо определенных функций. Основной идеей эволюционных вычислений является использование принципа естественного отбора, заключающегося в том, что наиболее приспособленные особи дают потомство, формирующее следующее поколение. В среднем следующее поколение является более приспособленным к окружающей среде, чем предыдущее.

В эволюционных вычислениях принято выделять подкласс эволюционных алгоритмов. При использовании эволюционных алгоритмов поиск оптимума ведется в пространстве гипотез X. Целевая функция f: $X \rightarrow R$ называется функцией приспособленности, или фитнесс-функцией. Отличительной особенностью эволюционных алгоритмов является использование операции кроссовера, аналогичного скрещиванию в эволюции живых существ. Как правило, операция кроссовера принимает две особи и порождает одну. В дальнейшем рассматриваются только такие операции кроссовера. Особи эволюционного алгоритма принято называть хромосомами [13].

Общая схема эволюционного алгоритма представлена на рисунке 6.1.

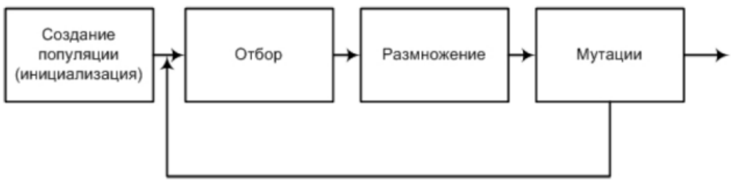

Рисунок 6.1 — Общая схема эволюционного алгоритма

Схема эволюционного алгоритма работает следующим образом:

1) генерируется случайное множество хромосом;

2) пока не выполняется критерий остановки:

– отбираются хромосомы, в наибольшей степени удовлетворяющие условиям задачи, т. е. имеющие большее значение функции f;

– отобранные хромосомы скрещиваются, порождая следующее поколение;

– к некоторым особям потомства применяется мутация [13].

Операции инициализации популяции, отбора, кроссовера и мутации называются генетическими операциями. Конкретные реализации методов представления особей и генетических операций порождают различные разновидности эволюционных алгоритмов.

1. Операция инициализации состоит в создании случайной начальной популяции.

2. Наиболее распространенными методами отбора являются метод рулетки, ранговый метод, элитизм и турнирный метод. Метод рулетки состоит в том, что для каждой особи шанс выжить прямо пропорционален значению ее функции приспособленности.

При использовании рангового метода шанс выжить прямо пропорционален рангу особи — ее порядку в отсортированном по значению фитнесс-функции списку особей.

Метод элитизма заключается в том, что отбрасываются все особи, кроме заданной доли наиболее приспособленных.

При использовании турнирного метода выбираются две случайные особи. После этого с вероятностью p выживает более приспособленная особь, а с вероятностью $1–p$ — менее приспособленная. Процесс повторяется до тех пор, пока не останется заданное количество особей.

3. Кроссовер — это генетический оператор, используемый для объединения генетической информации двух родителей для создания нового потомства.

Виды кроссовера:

1) одноточечный кроссовер — выбирается случайная точка пересечения, хвосты двух ее родителей меняются местами, чтобы получить новые исходные элементы (рис. 6.2);

2) Многоточечный кроссовер — обобщение одноточечного кроссовера, в котором чередующиеся сегменты меняются местами для получения новых выходных данных (рис. 6.3);

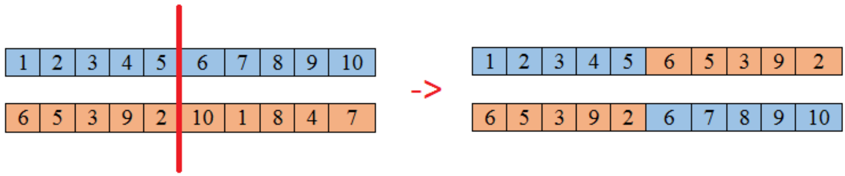

Рисунок 6.2 — Одноточечный кроссовер

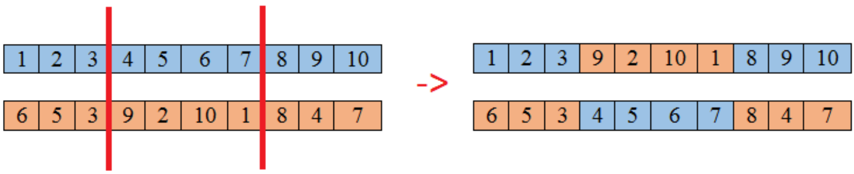

Рисунок 6.3 — Многоточечный кроссовер

Рисунок 6.4 — Однородный кроссовер

3) Однородный кроссовер рассматривает каждый ген отдельно. В однородном кроссовере, как правило, каждая хромосома выбирается из любого родителя с равной вероятностью. Иногда используются другие соотношения смешивания, в результате чего потомство наследует больше генетической информации от одного родителя, чем от другого (рис. 6.4).

4. Наиболее распространен следующий метод мутации — выбирается случайная позиция в строке, символ в этой позиции заменяется случайным образом [12].

Задания для самостоятельного выполнения

1. Разработать алгоритм поиска оптимума функции двух переменных эволюционным алгоритмом для функций по вариантам (*см.* табл. 1.1).
2. Нанести на график с линиями уровней распределение хромосом и экстремум функции.

Контрольные вопросы

1. Что собой представляют эволюционные вычисления?
2. Общая схема эволюционного алгоритма.
3. В чем состоит операция инициализации?
4. Что собой представляет операция отбора?
5. Наиболее распространенные методы отбора.
6. Что собой представляет операция кроссовера?
7. Назовите виды кроссовера.
8. Что собой представляет операция мутации?

Лабораторная работа 7
ЗАДАЧА ДЖОНСОНА

Цель работы: овладеть теоретическими и практическими навыками решения задачи двухпроцессорного обслуживания.

Краткие теоретические сведения

Исследование операций — научный метод выработки количественно обоснованных рекомендаций по принятию решений. Важность количественного фактора в исследовании операций и целенаправленность сформулированных рекомендаций позволяют определить исследование операций как теорию принятия оптимальных решений.

Первоначально исследование операций было связано с решением задач военного содержания, но уже с конца 40-х годов прошлого века оно используется для решения технических, технико-экономических задач, а также задач управления на различных уровнях.

Теория расписаний — это раздел исследования операций, в котором строятся и анализируются математические модели календарного планирования (т. е. упорядочивания во времени) различных целенаправленных действий с учетом целевой функции и различных ограничений [6].

«Эталонной» задачей теории расписаний является проблема составления расписания работы технологической линии, известная в литературе под названием задачи Джонсона, по имени С. М. Джонсона, получившего основные аналитические результаты для простейших ситуаций (вариантов) — частных постановок этой задачи.

Постановка задачи для задачи с двумя станками. Имеется n деталей и два станка. Каждая деталь должна сначала пройти обработку на первом станке, затем — на втором. При этом i-я деталь обрабатывается на первом станке за a_i времени, а на втором — за b_i времени. Каждый станок в каждый момент времени может работать только с одной деталью.

Требуется составить такой порядок подачи деталей на станки, чтобы итоговое время обработки всех деталей было бы минимальным. Эта задача называется иногда задачей двухпроцессорного обслуживания задач, или задачей Джонсона (который в 1954 году предложил алгоритм для её решения).

Алгоритм решения. Можно считать, что порядок обработки деталей на первом и втором станках должен совпадать. В самом деле, так как детали для второго станка становятся доступными только после обработки на первом, а при наличии нескольких доступных для второго станка деталей время их обработки будет равно сумме их b_i независимо от их порядка, выгоднее всего отправлять на второй станок ту деталь, которая раньше других прошла обработку на первом станке [3].

Цель — минимизировать суммарный простой:

$$F(x) = \sum x_i \to \min.$$

Расписание обработки деталей на станках задается перестановкой r натуральных чисел из множества $\{1, 2, ..., n\}$. Если $r = (1, 2, ..., n)$, то $x_1 = a_1$, $x_2 = \max((a_1 + a_2) - (x_1 + b_1), 0)$.

Таким образом, общий вид для x_i выглядит следующим образом:

$$x_i = \max(\sum_{i=1}^{n} a_i - \sum_{i=1}^{n-1} b_i - \sum_{i=1}^{n-1} x_i, 0), \tag{7.1}$$

где x_i — время простоя второго станка непосредственно перед началом обработки детали с номером i, $i = 1, 2, ..., n$, где n — номера деталей,

a_i и b_i — времена обработок детали с номером i на первом и втором станках соответственно.

Пусть r и q — две перестановки, для которых $r = (1, 2, ..., n)$, $q = (1, 2, ..., i - 1, i + 1, i, ..., n)$.

Таким образом, чтобы деталь i шла до детали $i + 1$, достаточно (хотя и не необходимо), чтобы: $F(r) < F(q)$.

$$F(r) = \max\{a_1 + a_2 + ... + a_i - b_1 - b_2 - \ldots - b_{i-1}, a_1 + a_2 + \ldots + a_{i+1} - b_1 - b_2 - \ldots - b_i\} \ F(q) = \max\{a_1 + a_2 + ... + a_{i-1} + a_{i+1} - b_1 - b_2 - \ldots - b_{i-1}, a_1 + a_2 + \ldots + a_{i+1} - b_1 - b_2 - \ldots - b_{i-1} - b_{i+1}\}.$$

Отняв из левой и правой частей неравенства величину

$$t = a_1 + a_2 + ... + a_{i+1} - b_1 - b_2 - b_{i-1}$$

и избавляясь от отрицательных чисел, имеем

$$\min\{a_{i+1}, b_i\} > \min\{a_i, b_{i+1})\}. \tag{7.2}$$

Итак, работа i выполняется раньше работы $i + 1$, если выполняется условие (7.2).

Если минимум достигается на a_i, то деталь с номером i ставится на обработку самой первой, если на b_i, то деталь с номером i ставится на обработку последней [3].

Диаграмма Ганта. Диаграмма Ганта — это инструмент планирования, управления задачами, который придумал американский инженер Генри Гант. Диаграмма имеет вид горизонтальных полос, расположенных между двумя осями: списком задач по вертикали и датами по горизонтали. На диаграмме видны не только сами задачи, но и их последовательность [17].

Пример решения задачи Джонсона.

Постановка задачи. Пусть время обработки пяти деталей на двух машинах задано в таблице 7.1.

Т а б л и ц а 7. 1 — Время обработки двух деталей на пяти станках

Станок / время обработки	a_i	b_i
1	3	6
2	7	2
3	4	7
4	5	3
5	7	4

Диаграмма Ганта для заданного времени обработки отображена на рисунке 7.1. Время простоя выделено красным цветом.

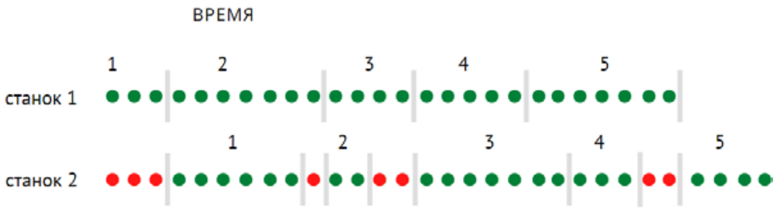

Рисунок 7.1 — Диаграмма Ганта для начального времени обработки

По графику видно, что начальный порядок обработки деталей допускает простои второго станка (суммарное время простоев 8 единиц), длина производственного цикла равна 30 единицам времени.

По алгоритму Джонсона необходимо определить минимальную величину min (a_i, b_i), i = 1, …, n (отмечена * в таблице 7.2). Таким образом, деталь 2 на первом станке обрабатывается последней, т. е. пятой, $i' = 5$.

Т а б л и ц а 7.2 — Первый шаг алгоритма Джонсона

Станок / время обработки	a_i	b_i	i'
1	3	6	
2	7	2 *	5
3	4	7	
4	5	3	
5	7	4	

Среди невычеркнутых элементов необходимо найти min (a_i, b_i), i = 1, …, n. После выбора второй детали минимальное время равно 3, и оно соответствует деталям a_1 и b_4. Можно выбрать любую деталь, поэтому произвольно выбрана a_1, т. е. на первое место помещена деталь 1. Теперь минимальное время соответствует b_4. Следовательно, деталь 4 ставится на предпоследнее место (табл. 7.3).

Т а б л и ц а 7.3 — Второй шаг алгоритма Джонсона

Станок / время обработки	a_i	b_i	i'
1	3 *	6	1
2	7	2	5
3	4	7	
4	5	3 *	4
5	7	4	

Следующая минимальная величина равна 4 (a_3 и b_5). Можно назначить 2-е место на первом станке для детали 3 и 3-е место — для детали 5 (табл. 7.4).

Т а б л и ц а 7.4 — Третий шаг алгоритма Джонсона

Станок / время обработки	a_i	b_i	i'
1	3	6	1
2	7	2	5
3	4 *	7	2
4	5	3	4
5	7	4 *	3

Полученная последовательность обработки деталей на двух станках s^* = (1, 3, 5, 4, 2) будет оптимальной. Эта последовательность представлена диаграммой Ганта на рисунке 7.2.

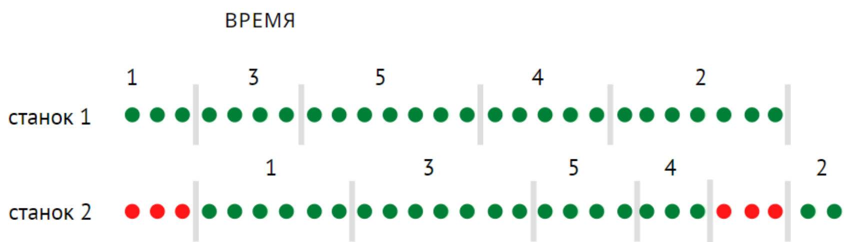

Рисунок 7.2 — Диаграмма Ганта для оптимального времени обработки

Из рисунка видно, что время обработки всех деталей равно 28. Время простоя необходимо определить по формуле (7.1).

Время простоя второй машины при первичном порядке

$$\max(3, 4, 6, 4, 8) = 8.$$

Время простоя при оптимальной перестановке

$$\max(3, 1, 1, 2, 6) = 6.$$

Задания для самостоятельного выполнения

Имеется последовательность обработки на двух машинах 10 различных деталей. Известно время a_i и b_i обработки i-й детали на соответствующих машинах по вариантам (табл. 7.5). Написать программное приложение, которое определит порядок обработки, минимизирующий время простоя второй машины и тем самым сокращающий общее время обработки деталей. Построить диаграмму Ганта для первичного порядка и оптимальной перестановки. Сравнить время простоя и время обработки деталей при первичном порядке обработки и при оптимальной перестановке.

Т а б л и ц а 7.5 — Варианты заданий

Вариант	Время обработки	1	2	3	4	5	6	7	8	9	10
1	a_i	2	4	5	5	4	3	1	2	1	1
	b_i	5	3	4	5	2	1	4	3	1	5
2	a_i	3	1	5	3	3	3	1	5	3	3
	b_i	5	3	4	4	4	4	2	1	1	2
3	a_i	1	5	4	1	3	4	5	1	1	3
	b_i	1	2	1	4	1	1	3	3	5	4
4	a_i	2	1	3	3	4	1	2	3	3	4
	b_i	4	3	3	4	5	3	1	4	4	2
5	a_i	4	5	1	2	3	2	4	4	4	1
	b_i	3	3	3	4	3	1	2	1	1	2
6	a_i	5	2	2	4	4	1	1	5	1	2
	b_i	5	1	5	4	2	3	3	2	4	1
7	a_i	2	5	5	2	3	2	5	4	2	5
	b_i	5	4	1	2	3	3	4	4	2	3
8	a_i	3	5	3	1	2	5	3	3	5	5
	b_i	1	3	2	4	5	1	2	2	2	5
9	a_i	5	4	1	2	1	5	1	3	2	4
	b_i	2	5	3	5	3	3	2	5	4	5
10	a_i	2	4	3	1	4	5	4	3	1	4
	b_i	3	3	1	3	3	5	1	5	2	1
11	a_i	1	2	2	5	3	1	3	4	3	3
	b_i	4	4	2	1	1	5	3	5	2	1
12	a_i	4	4	3	5	5	4	1	3	3	5
	b_i	2	2	5	5	2	3	3	3	5	5
13	a_i	3	5	5	4	1	4	1	3	5	1
	b_i	2	5	2	5	3	5	3	3	1	1

Окончание табл. 7.5

Вариант	Время обработки	1	2	3	4	5	6	7	8	9	10
14	a_i	1	4	5	4	1	4	1	1	4	4
	b_i	2	2	4	3	5	1	3	1	5	5
15	a_i	4	1	2	1	5	1	4	1	5	2
	b_i	4	5	1	5	5	1	4	1	3	4

Контрольные вопросы

1. Что собой представляет научный метод исследования операций?
2. Что изучает теория расписаний?
3. Что собой представляет задача Джонсона?
4. Постановка задачи Джонсона.
5. Алгоритм решения задачи Джонсона.
6. Формула определения времени простоя.
7. Назначение диаграммы Ганта.

Л а б о р а т о р н а я р а б о т а 8

ЗАДАЧА УПРАВЛЕНИЯ ЗАПАСАМИ

Цель работы: овладеть теоретическими и практическими навыками применения методов управления товарными и материальными запасами.

Краткие теоретические сведения

Теория управления запасами разрабатывает методы вычисления величины запасов, обеспечивающих оптимальным путем удовлетворение будущего (не всегда определенного) спроса. При этом под запасом подразумевают товарно-материальные ценности, находящиеся на различных стадиях производственного или торгового процесса, ожидающие потребления. Образование запасов товарно-материальных ценностей на предприятии связано прежде всего с обеспечением нужд производственного или торгового процесса. Запасы позволяют предприятию накапливать ресурсы и тем самым повышать надежность экономического процесса [5].

В основу управления товарными запасами положены различные оптимизационные модели, наработанные экономической наукой и позволяющие не только планировать и контролировать формирование и рациональное использование запасов в торговле, но и минимизировать расходы, связанные с этими процессами. Кроме того, оптимизация процесса управления товарными запасами предполагает и решение вопросов относительно периодичности их пополнения, а также величины заказов [7].

Среди наиболее применяемых в торговле моделей управления запасами можно выделить следующие.

1. *Модель с фиксированным размером заказа* применяется при следующих условиях:

– издержки управления запасами значительные;

– издержки управления запасами можно вычислить;

– при заказе товаров поставщик налагает ограничения на минимальный размер партии, в этом случае легче один раз скорректировать фиксированный размер партии, чем непрерывно регулировать его переменный заказ;

– позволяет справиться с неожиданно большими колебаниями спроса.

В данной модели оптимальный размер заказа для всех комплектующих изделий рассчитывается по формуле Вильсона [6]:

$$q_0 = \sqrt{\frac{2C_1Q}{C_2}}, \tag{8.1}$$

где q_0 — оптимальный размер заказа, шт.;

C_1 — стоимость выполнения одного заказа, руб.;

Q — потребность в товарно-материальных ценностях за определенный период времени (год), шт.;

C_2 — затраты на содержание единицы запаса, руб. / шт.

Интервал времени между заказами для всех комплектующих изделий при условии соблюдения оптимальной партии поставки производится по формуле

$$t = \frac{N_{\text{раб. дн}}}{n},$$

где $N_{\text{раб. дн}}$ — количество рабочих дней в году;

n — количество партий поставок за период (год):

$$n = \frac{Q}{q_0},$$

где Q — потребность в товарно-материальных ценностях за определенный период времени (год), шт.;

q_0 — оптимальный размер заказа, шт.

Порядок расчета параметров системы управления запасами с фиксированным размером заказа представлен в таблице 8.1.

Таблица 8.1 — Порядок расчета параметров системы управления запасами с фиксированным размером заказа

Показатель	Порядок расчета
Потребность, шт.	Данные предприятия
Оптимальный размер заказа, шт.	По формуле (8.1)

Окончание табл. 8.1

Показатель	Порядок расчета
Время поставки, дни (1)	Данные транспортной компании
Возможная задержка поставки, дни (2)	Данные транспортной компании
Ожидаемое дневное потребление, шт. / день (3)	*n* (1) / (число рабочих дней)
Срок расходования заказа, дни (4)	(2) / (5)
Ожидаемое потребление за время поставки, шт. (5)	(3) · (5)
Максимальное потребление за время поставки, шт. (6)	((3) + (4)) · (5)
Страховой запас, шт. (7)	(8) – (7)
Пороговый уровень запаса, шт. (8)	(9) + (7)
Максимальный желательный запас, шт. (9)	(9) + (2)
Срок расходования запаса до порогового уровня, дни, шт. (10)	((11) – (10)) / (5)

График движения запасов модели с фиксированным размером заказа представлен на рисунке 8.1.

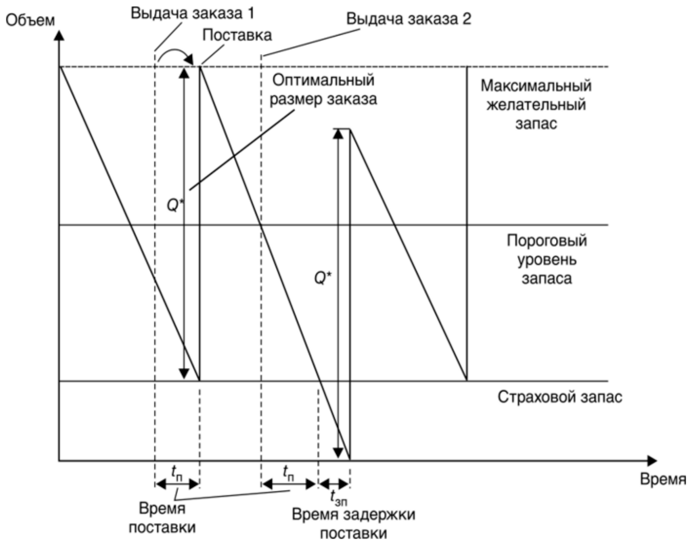

Рисунок 8.1 — График движения запасов модели с фиксированным размером заказа

Пример решения задачи с фиксированным размером заказа.

Рассчитать параметры системы управления заказами с фиксированным размером заказа, если годовая потребность в материалах — 896 шт., число рабочих дней в году — 226 дней, время поставки — 4 дня, возможная задержка поставки — 1 день, стоимость заказа — 100 руб., стоимость единицы изделия — 560 руб., доля затрат на содержание единицы изделия — 20 % от стоимости единицы изделия. Расчетные данные представлены в таблице 8.2.

Таблица 8.2 — Расчет параметров системы управления запасами с фиксированным размером заказа

Показатель	Значение
Потребность, шт.	896
Оптимальный размер заказа, ед.	$\sqrt{\frac{2 \cdot 100 \cdot 896}{112}} = 40$
Время поставки, дни	4
Возможная задержка поставки, дни	1
Ожидаемое дневное потребление, шт. / день	896 / 226 = 4
Срок расходования заказа, дни	40 / 4 = 10
Ожидаемое потребление за время поставки, шт.	4 ·4 = 16
Максимальное потребление за время поставки, шт.	(4 + 1) · 4 = 20
Страховой запас, шт.	20 – 16 = 4
Пороговый уровень запаса, шт.	4 + 16 = 20
Максимальный желательный запас, шт.	4 + 40 = 44
Срок расходования запаса до порогового уровня, дни, шт.	(44 – 20) / 4 = 6

Далее необходимо рассчитать уровни запаса без задержек поставок, с однократной задержкой и с многократными задержками при основных параметрах таблицы 8.2 (табл. 8.3).

Т а б л и ц а 8.3 — Уровни запаса на 30 дней

	Без задержек			С однократной задержкой			С многократными задержками		
День	Запас	Расход	Приход	Запас	Расход	Приход	Запас	Расход	Приход
1	44	4		44	4		44	4	
2	40	4		40	4		40	4	
3	36	4		36	4		36	4	
4	32	4		32	4		32	4	
5	28	4		28	4		28	4	
6	24	4		24	4		24	4	
7	20	4		20	4		20	4	
8	16	4		16	4		16	4	
9	12	4		12	4		12	4	
10	8	4		8	4		8	4	
11	44	4	40	44	4	40	4	4	
12	40	4		40	4		40	4	40
13	36	4		36	4		36	4	
14	32	4		32	4		32	4	
15	28	4		28	4		28	4	
16	24	4		24	4		24	4	
17	20	4		20	4		20	4	
18	16	4		16	4		16	4	
19	12	4		12	4		12	4	
20	8	4		8	4		8	4	
21	44	4	40	4	4		4	4	
22	40	4		40	4	40	0	4	
23	36	4		36	4		36	4	40
24	32	4		32	4		32	4	
25	28	4		28	4		28	4	
26	24	4		24	4		24	4	
27	20	4		20	4		20	4	
28	16	4		16	4		16	4	
29	12	4		12	4		12	4	
30	8	4		8	4		8	4	

График движения запасов представлен на рисунке 8.2.

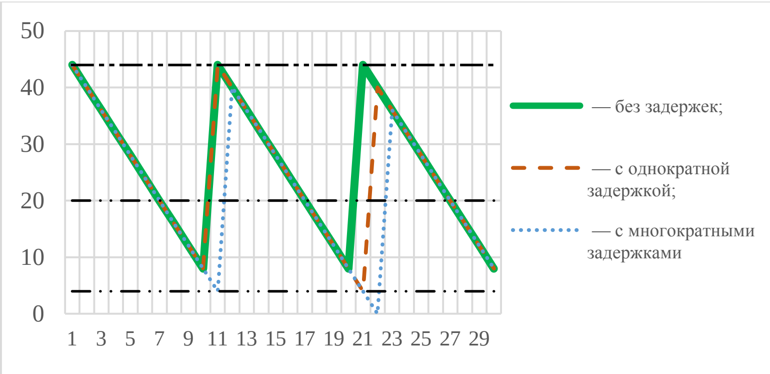

Рисунок 8.2 — График движения запасов фиксированным размером заказа

2. *Модель с фиксированным интервалом между заказами* применяется при следующих условиях:

– издержки управления запасами незначительные, ими можно пренебречь;

– издержки управления запасами вычислить невозможно;

– поставка товаров происходит в установленные сроки, или поставщик налагает ограничения на периодичность и сроки поставки;

– необходимо быстро реагировать на изменение сбыта;

– при управлении достаточно стабильными запасами;

– при предсказуемом спросе в сочетании с низкой стоимостью предметов хранения;

– организация работает в условиях, когда реализация запасов очень неравномерна во времени, подвержена существенным колебаниям и не поддается планированию и прогнозированию;

– отличается простотой управления.

Для расчёта интервала между заказами используется формула

$$t_{\text{мз}} = \frac{NQ}{S}, \quad (8.2)$$

где N — число рабочих дней в плановом периоде, дни;

Q — оптимальный размер заказа, ед.;

S — объем потребности в запасе, ед.

Полученный с помощью формулы (8.2) интервал времени между заказами не является обязательным. Он может быть скорректирован на основе экспертных оценок. Например, при расчетном результате в 4 дня можно использовать интервал в 5 дней, чтобы производить заказы 1 раз в неделю.

Методика управления запасами на основе фиксации интервала между заказами заключается в том, что заказы на пополнение запаса делаются в заранее определенный момент через фиксированные интервалы между заказами в размере, который обеспечивает пополнение запаса до максимально желательного уровня. Размер заказа должен быть равен:

$$Q_i = \text{МЖЗ} - Z_{T_i} + \text{ОП} - Z_{t_i},$$

где Q_i — размер заказа, ед.;

МЖЗ — максимальный желательный запас, ед.;

Z_{T_i} — уровень текущего запаса при выдаче заказа, ед.;

ОП — ожидаемое потребление за время выполнения заказа, единиц;

Z_{t_i} — объем запаса в пути, не полученного к моменту выдачи заказа, единиц [7].

Порядок расчета параметров системы управления запасами с фиксированным интервалом времени между заказами представлен в таблице 8.4.

Таблица 8.4 — Порядок расчета параметров системы управления запасами с фиксированным интервалом времени между заказами

Показатель	Порядок расчета
Потребность, шт. (1)	Данные предприятия
Интервал времени между заказами, дни (2)	По формуле (8.2)
Время поставки, дни (3)	Данные транспортной компании
Возможная задержка поставки, дни (4)	Данные транспортной компании
Ожидаемое дневное потребление, шт. / день (5)	(1) / (число рабочих дней)
Ожидаемое потребление за время поставки, шт. (6)	(3) · (5)
Максимальное потребление за время поставки, шт. (7)	((3) + (4)) · (5)
Страховой запас, шт. (8)	(7) – (6)
Максимальный желательный запас, шт. (9)	(8) + (2) · (5)

График движения запасов модели с фиксированным с фиксированным интервалом времени между заказами представлен на рисунке 8.3.

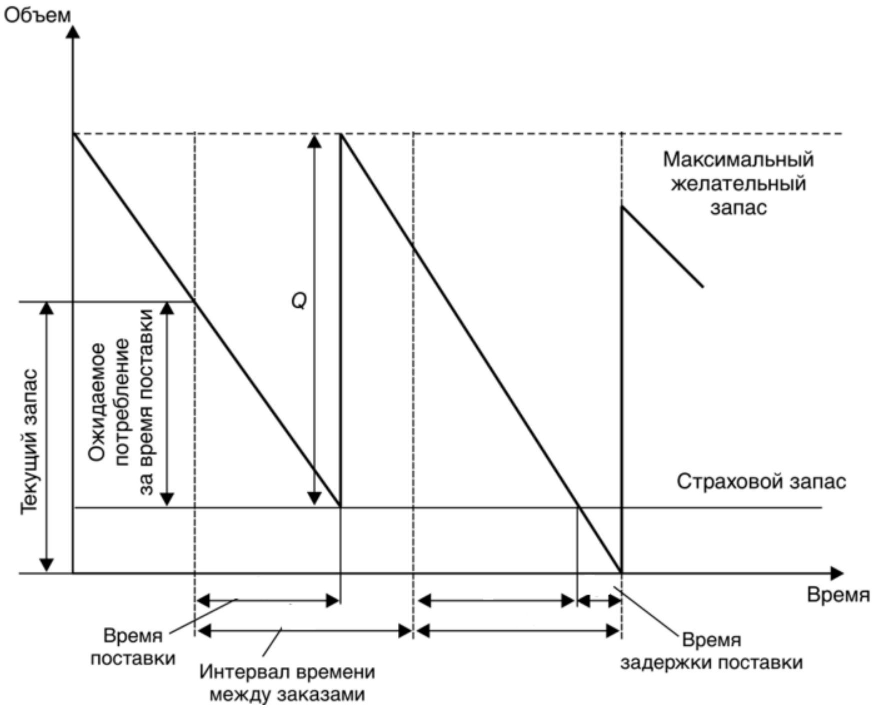

Рисунок 8.3 — График движения запасов модели
с фиксированным интервалом времени между заказами

Пример решения задачи с фиксированным интервалом времени между заказами.

Рассчитать параметры системы управления заказами с фиксированным интервалом времени между заказами, если годовая потребность в материалах составляет 896 шт., число рабочих дней в году — 226 дней, оптимальный размер заказа — 40 шт., время поставки — 4 дня, возможная задержка поставки — 1 день.

Расчетные данные представлены в таблице 8.5.

Т а б л и ц а 8.5 — Расчет параметров системы управления запасами с фиксированным интервалом времени между заказами

Показатель	Порядок расчета
Потребность, шт.	896
Интервал времени между заказами, дни	226 · 40 / 896 = 10
Время поставки, дни	4
Возможная задержка поставки, дни	1
Ожидаемое дневное потребление, шт. / день	896 / 226 = 4
Ожидаемое потребление за время поставки, шт.	4 · 4 = 16
Максимальное потребление за время поставки, шт.	(4 + 1) · 4 = 20
Страховой запас, шт.	20 – 16 = 4
Максимальный желательный запас, шт.	4 + 10 · 4 = 44

Т а б л и ц а 8.6 — Уровни запаса на 30 дней

День	Без задержек				С многократными задержками			
	Запас	Расход	Приход	Заказ	Запас	Расход	Приход	Заказ
1	44	4		44 – 44 + 16 = 16	44	4		44 – 44 + 16 = 16
2	40	4			40	4		
3	36	4			36	4		
4	32	4			30	4		
5	44	4	16		28	4		
6	40	4			40	4	16	
7	36	4			36	4		
8	32	4			32	4		
9	28	4			28	4		
10	24	4			24	4		
11	20	4		44 – 20 + 16 = 40	20	4		44 – 20 + 16 = 40
12	16	4			16	4		
13	12	4			12	4		
14	8	4			8	4		
15	44	4	40		4	4		
16	40	4			40	4	40	
17	36	4			36	4		
18	32	4			32	4		
19	28	4			28	4		
20	24	4			24	4		
21	20	4		44 – 20 + 16 = 40	20	4		44 – 20 + 16 = 40
22	16	4			16	4		

Окончание табл. 8.6

День	Без задержек				С многократными задержками			
	Запас	Расход	Приход	Заказ	Запас	Расход	Приход	Заказ
23	12	4			12	4		
24	8	4			8	4		
25	44	4	40		4	4		
26	40	4			40	4	40	
27	36	4			36	4		
28	32	4			32	4		
29	28	4			28	4		
30	24	4			24	4		

График движения запасов представлен на рисунке 8.4.

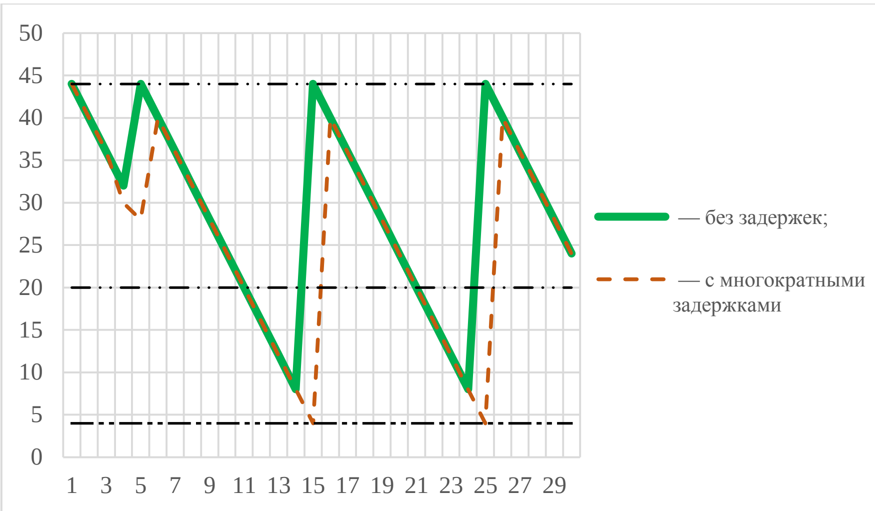

Рисунок 8.4 — График движения запасов с фиксированным интервалом времени

Задания для самостоятельного выполнения

Рассчитать параметры системы управления заказами по вариантам. Построить график движения запасов:

– для модели с фиксированным размером заказа для вариантов: без задержек, с однократной задержкой, с многократными задержками;

– для модели с фиксированным временем между заказами для вариантов: без задержек и с многократными задержками.

График должен отображать движение запасов за 90 дней.

Вариант 1.

Рассчитать параметры системы управления заказами с фиксированным размером заказа, если годовая потребность в материалах — 1 058 шт., число рабочих дней в году — 350 дней, время поставки — 5 дней, возможная задержка поставки — 2 дня, стоимость заказа — 130 руб., стоимость единицы изделия — 520 руб., доля затрат на содержание единицы изделия — 30 % от стоимости единицы изделия.

Вариант 2.

Рассчитать параметры системы управления заказами с фиксированным интервалом времени между заказами, если годовая потребность в материалах составляет 1 004 шт., число рабочих дней в году — 226 дней, оптимальный размер заказа — 50 шт., время поставки — 5 дней, возможная задержка поставки — 1 день.

Вариант 3.

Рассчитать параметры системы управления заказами с фиксированным размером заказа, если годовая потребность в материалах — 1 120 шт., число рабочих дней в году — 350 дней, время поставки — 6 дней, возможная задержка поставки — 2 дня, стоимость заказа — 150 руб., стоимость единицы изделия — 480 руб., доля затрат на содержание единицы изделия — 20 % от стоимости единицы изделия.

Вариант 4.

Рассчитать параметры системы управления заказами с фиксированным интервалом времени между заказами, если годовая потребность в материалах составляет 890 шт., число рабочих дней в году — 350 дней, оптимальный размер заказа — 45 шт., время поставки — 6 дней, возможная задержка поставки — 1 день.

Вариант 5.

Рассчитать параметры системы управления заказами с фиксированным размером заказа, если годовая потребность в материалах — 960 шт., число рабочих дней в году — 350 дней, время поставки — 5 дней, возможная задержка поставки — 1 день, стоимость заказа — 120 руб., стоимость единицы изделия — 520 руб., доля затрат на содержание единицы изделия — 20 % от стоимости единицы изделия.

Вариант 6.

Рассчитать параметры системы управления заказами с фиксированным интервалом времени между заказами, если годовая потребность в материалах составляет 100 шт., число рабочих дней в году — 350 дней, оптимальный размер заказа — 40 шт., время поставки — 7 дней, возможная задержка поставки — 1 день.

Вариант 7.

Рассчитать параметры системы управления заказами с фиксированным размером заказа, если годовая потребность в материалах — 890 шт., число рабочих дней в году — 226 дней, время поставки — 5 дней, возможная задержка поставки — 1 день, стоимость заказа — 110 руб., стоимость единицы изделия — 520 руб., доля затрат на содержание единицы изделия — 30 % от стоимости единицы изделия.

Вариант 8.

Рассчитать параметры системы управления заказами с фиксированным интервалом времени между заказами, если годовая потребность в материалах составляет 1 021 шт., число рабочих дней в году — 226 дней, оптимальный размер заказа — 35 шт., время поставки — 5 дней, возможная задержка поставки — 2 дня.

Вариант 9.

Рассчитать параметры системы управления заказами с фиксированным размером заказа, если годовая потребность в материалах — 1 170 шт., число рабочих дней в году — 365 дней, время поставки — 6 дней, возможная задержка поставки — 2 дня, стоимость заказа — 80 руб., стоимость единицы изделия — 420 руб., доля затрат на содержание единицы изделия — 20 % от стоимости единицы изделия.

Вариант 10.

Рассчитать параметры системы управления заказами с фиксированным интервалом времени между заказами, если годовая потребность в материалах составляет 890 шт., число рабочих дней в году — 226 дней, оптимальный размер заказа — 47 шт., время поставки — 7 дней, возможная задержка поставки — 1 день.

Вариант 11.

Рассчитать параметры системы управления заказами с фиксированным размером заказа, если годовая потребность в материалах — 860 шт., число рабочих дней в году — 350 дней, время поставки — 5 дней, возможная задержка поставки — 2 дня, стоимость заказа — 180 руб., стоимость единицы изделия — 460 руб., доля затрат на содержание единицы изделия — 30 % от стоимости единицы изделия.

Вариант 12.

Рассчитать параметры системы управления заказами с фиксированным интервалом времени между заказами, если годовая потребность в материалах составляет 790 шт., число рабочих дней в году — 226 дней, оптимальный размер заказа — 45 шт., время поставки — 6 дней, возможная задержка поставки — 1 день.

Вариант 13.

Рассчитать параметры системы управления заказами с фиксированным размером заказа, если годовая потребность в материалах — 1 050 шт., число рабочих дней в году — 226 дней, время поставки — 5 дней, возможная задержка поставки — 2 дня, стоимость заказа — 120 руб., стоимость единицы изделия — 530 руб., доля затрат на содержание единицы изделия — 30 % от стоимости единицы изделия.

Вариант 14.

Рассчитать параметры системы управления заказами с фиксированным интервалом времени между заказами, если годовая потребность в материалах составляет 998 шт., число рабочих дней в году — 226 дней, оптимальный размер заказа — 40 шт., время поставки — 6 дней, возможная задержка поставки — 2 дня.

Вариант 15.

Рассчитать параметры системы управления заказами с фиксированным размером заказа, если годовая потребность в материалах — 1 000 шт., число рабочих дней в году — 365 дней, время поставки — 5 дней, возможная задержка поставки — 2 дня, стоимость заказа — 100 руб., стоимость единицы изделия — 460 руб., доля затрат на содержание единицы изделия — 30 % от стоимости единицы изделия.

Контрольные вопросы

1. Что собой представляет теория управления запасами?
2. Модели управления запасами.
3. При каких условиях применяется модель с фиксированным размером заказа?
4. При каких условиях применяется модель с фиксированным интервалом между заказами?
5. В осях каких координат рассматривается движение запаса?
6. Какие главные вопросы решаются при управлении запасами?
7. Почему модель с фиксированным размером заказа и модель с фиксированным интервалом времени между заказами называются основными системами управления запасами?
8. Назовите ключевой параметр модели управления запасами с фиксированным размером заказа.
9. В каких условиях наиболее эффективно применять идеи системы с фиксированным размером заказа?
10. Перечислите исходные данные модели управления запасами с фиксированным размером заказа.
11. Перечислите расчетные параметры модели управления запасами с фиксированным размером заказа.

Лабораторная работа 9

ЗАДАЧА ОПТИМАЛЬНОГО РАСПРЕДЕЛЕНИЯ ИНВЕСТИЦИЙ

Цель работы: овладеть теоретическими и практическими навыками решения задач инвестирования денег в различные отрасли при помощи методов динамического программирования.

Краткие теоретические сведения

Динамическое программирование представляет собой математический аппарат, разработанный для эффективного решения некоторого класса задач математического программирования. Этот класс характеризуется возможностью естественного разбиения всей операции на ряд взаимосвязанных этапов.

Модели динамического программирования могут применяться:

– при разработке правил управления запасами, устанавливающими момент пополнения запасов и размер пополняющего заказа;

– при разработке принципов календарного планирования производства и выравнивания занятости в условиях колеблющегося спроса на продукцию;

– при распределении инвестиций между новыми направлениями их использования;

– при составлении календарных планов текущего и капитального ремонта сложного оборудования и его замены;

– при разработке долгосрочных правил замены выбывающих из эксплуатации основных фондов и т. д.

Проблема распределения инвестиций относится к разряду «вечных»: инвестиции, в отличие от потребностей, всегда ограничены. Их так или иначе приходится распределять на различные нужды постоянно и на всех уровнях [4].

Общая постановка задачи. Инвестор выделяет средства в размере D условных единиц, которые должны быть распределены между предприятиями. Каждое i-е предприятие при инвестировании в него средств x приносит прибыль $\varphi_i(x)$ денежных единиц, $i=\overline{1,m}$. Нужно выбрать оптимальное распределение инвестиций между предприятиями, обеспечивающее максимальную прибыль.

Выигрышем W данной задачи является прибыль, приносимая m-предприятиями.

Построение математической модели.

1. Определение числа шагов. Число шагов n равно числу предприятий, в которые осуществляется инвестирование.

2. Определение состояний системы. Состояние системы на каждом шаге характеризуется количеством средств s, имеющихся в наличии перед данным шагом, $s \le D$.

3. Выбор шаговых управлений. Управлением на i-м шаге x_i, $i = \overline{1, m}$ является количество средств, инвестируемых в i-е предприятие.

4. Функция выигрыша на i-м шаге:

$$\varphi_i(x) \tag{9.1}$$

— это прибыль, которую приносит i-е предприятие при инвестировании в него средств x_i.

$$W = \sum_{j=1}^{m} \varphi_i x_i,$$

следовательно, данная задача может быть решена методом динамического программирования.

5. Определение функции перехода в новое состояние:

$$f_i(s, x_i) = s - x_i э \tag{9.2}$$

Таким образом, если на i-м шаге система находилась в состоянии s, а выбрано управление x, то на i + 1-м шаге система будет находиться в состоянии $s - x$. Другими словами, если в наличии имеются средства в размере s денежных единиц и в i-е предприятие инвестируется x денежных единиц, то для дальнейшего инвестирования остается $s - x$ денежных единиц.

6. Составление функционального уравнения для $i = m$:

$$W_m(s) = \varphi_m(s),$$

$$x_m(s) = s. \tag{9.3}$$

На последнем шаге, перед инвестированием средств в последнее предприятие, условное оптимальное управление соответствует количеству средств, имеющихся в наличии, т. е. сколько средств осталось, столько и надо вложить в последнее предприятие. Условный оптимальный выигрыш равен доходу, приносимому последним предприятием.

7. Составление основного функционального уравнения. При подстановке в формулу (9.3) выражений (9.1) и (9.2) получится следующее функциональное уравнение:

$$W_i(s) = \max\left\{\varphi_i(x) + W_{i+1}(s-x)\right\}.$$

Пусть перед i-м шагом у инвестора остались средства в размере s денежных единиц. Тогда x денежных единиц он может вложить в i-е предприятие, при этом оно принесет доход $\varphi_i(x)$, а оставшиеся $s - x$ денежных единиц — в остальные предприятия с $(i + 1)$ до m-го. Условный оптимальный выигрыш от такого вложения $W_{i+1}(s-x)$. Оптимальным будет то условное управление x, при котором сумма $\varphi_i(x)$ и $W_{i+1}(s-x)$ максимальна [18].

Метод прогонки. Метод прогонки является модификацией метода Гаусса для частного случая разреженных систем — системы уравнений с трехдиагональной матрицей. Такие системы получаются при моделировании некоторых инженерных задач, а также при численном решении краевых задач для дифференциальных уравнений.

Метод прогонки состоит из двух этапов — прямой и обратной прогонки. Прямая прогонка состоит в том, что каждое неизвестное x_i выражается через x_i + 1 с помощью прогоночных коэффициентов.

Обратная прогонка состоит в последовательном вычислении неизвестных x_i. Для этого сначала необходимо найти x_n. Далее, используя формулы и выражения для прогоночных коэффициентов, последовательно необходимо вычислить все неизвестные x_{n-1}, x_{n-2}, ..., x_1.

Пример решения задачи оптимального распределения инвестиций.

Постановка задачи. Необходимо определить оптимальный план расширения производства трех предприятий, если известна их прибыль в год при отсутствии вложений и при инвестировании 1, 2, 3, 4 или 5 денежных единиц. Прибыль предприятий в зависимости от вложений отображена в таблице 9.1.

Т а б л и ц а 9.1 — Распределение прибыли в зависимости от инвестиций

Вложения / Прибыль	1	2	3	4	5
$\varphi_1(x)$	1,5	2	2,5	3	3,6
$\varphi_2(x)$	2	2,1	2,3	3,5	4
$\varphi_3(x)$	1,7	2,4	2,7	3,2	3,5

Решение методом прямой прогонки.

I этап. Условная оптимизация.

1-й шаг: m = 1. Предполагается, что все средства в количестве 5 отданы первому предприятию.

$F_i(L) = \max(\varphi_i)$; $0 \le x_i \le 5$; $x_i = 0, 1, 2, 3, 4, 5$.

Преобразованная матрица отображена в таблице 9.2.

Т а б л и ц а 9.2 — Преобразованная матрица на шаге 1

Вложения	Вложения						
	x_1	0	1	2	3	4	5
x_2	$\varphi_2(x_2)$ / $\varphi_1(x_1)$	0	1,5	2	2,5	3	3,6
0	0	0	1,5	2	2,5	3	3,6
1	2	**2***	**3,5***	**4***	**4,5***	5	0
2	2,1	2,1	3,6	4,1	4,6	0	0
3	2,3	2,3	3,8	4,3	0	0	0
4	3,5	3,5	**5***	0	0	0	0
5	4	4	0	0	0	0	0

Далее на каждой северо-восточной диагонали необходимо найти наибольшее число, которое отмечено в таблице звездочкой, и указать в таблице 9.3 соответствующее значение x_1.

Т а б л и ц а 9.3 — Соответствующие значения для второго предприятия

Прибыль	Вложения				
s_1	1	2	**3**	4	5
$F_2(s_1)$	2	3,5	**4**	4,5	5
x_2	1	1	**1**	1	4

2-й шаг: m = 2. Необходимо определить оптимальную стратегию при распределении денежных средств между предприятиями 2 и 3. При этом рекуррентное соотношение Беллмана имеет вид: $F_i(L) = \max[\varphi_i + F_{i-1}(L - s)]$ и отображено в таблицах 9.4 и 9.5.

Т а б л и ц а 9.4 — Преобразованная матрица на шаге 2

Вложения	Вложения						
	x_2	0	1	2	3	4	5
x_3	$\varphi_3(x_3) / \varphi_2(x_2)$	0	2	3,5	4	4,5	5
0	0	0	0	0	0	0	5
1	1,7	0	0	0	0	6,2	0
2	2,4	0	0	0	**6,4 ***	0	0
3	2,7	0	0	6,2	0	0	0
4	3,2	0	5,2	0	0	0	0
5	3,5	3,5	0	0	0	0	0

Т а б л и ц а 9.5 — Соответствующие значения для третьего предприятия

Прибыль	Вложения					
s_2	0	1	2	3	4	**5**
$F_3(s_2)$						**6,4**
x_3	0	1	2	3	4	**2**

II этап. Безусловная оптимизация.

1-й шаг: m = 3. По данным таблицы 9.5, максимальный доход при распределении денежных средств между предприятиями составляет $s_2 = 5$, $F_3(5) = 6{,}4$. При этом 3-му предприятию нужно выделить $x_3 = 2$ денежные единицы.

2-й шаг: m = 2. Определяется величина оставшихся денежных средств, приходящихся на долю остальных предприятий:

$s_3 = s_2 - x_3 = 5 - 2 = 3.$

По данным таблицы 6.3, максимальный доход при распределении между предприятиями составляет $s_1 = 3$, $F_2(3) = 4$. При этом 2-му предприятию нужно выделить 1 денежную единицу.

3-й шаг: на долю второго предприятия остается 2 денежные единицы. Таким образом, оптимальный план инвестирования предприятия: $x_1 = 2$, $x_2 = 1$, $x_3 = 2$, который обеспечит максимальный доход: $F(5) = \varphi_1(2) + \varphi_2(1) + \varphi_3(2) = 2 + 2 + 2{,}4 = 6{,}4$ денежные единицы.

Решение методом обратной прогонки:

– на этапе условной оптимизации предполагает, что все средства в количестве 5 денежных единиц отданы 3-му предприятию, далее определение оптимальной стратегии идет по убыванию от предприятия 3 к предприятию 1;

– на этапе безусловной оптимизации выделение дохода идет по возрастанию от предприятия 1 к предприятию 3.

Задания для самостоятельного выполнения

Разработать программное приложение, которое позволит определить оптимальный план расширения производства четырех предприятий, если известна их прибыль в год при инвестировании 1, 2, 3, 4 или 5 денежных единиц. Предусмотреть возможность использования методов прямой и обратной прогонки, а также просмотра итерационного решения. Прибыль предприятий в зависимости от вложений отображена в таблице 9.6 по вариантам.

Таблица 9.6 — Варианты заданий

Вариант	Прибыль	Вложения				
		1	2	3	4	5
1	$\varphi_1(x)$	1,6	1,8	1,9	2	2,3
	$\varphi_2(x)$	1,5	1,9	2,3	2,4	2,7
	$\varphi_3(x)$	1,9	2,3	2,5	2,6	2,7
	$\varphi_4(x)$	1,8	1,9	2,1	2,3	2,5
2	$\varphi_1(x)$	1,9	2	2,3	2,6	2,9
	$\varphi_2(x)$	1,7	1,9	2,4	2,8	3
	$\varphi_3(x)$	1,9	2,1	2,5	2,7	2,9
	$\varphi_4(x)$	1,8	2,1	2,5	2,8	3,1
3	$\varphi_1(x)$	2,1	2,3	2,5	2,8	3,2
	$\varphi_2(x)$	2,1	2,5	2,9	3	3,1
	$\varphi_3(x)$	1,9	2,3	2,4	2,8	2,9
	$\varphi_4(x)$	1,7	1,9	2,5	2,9	3,1

Продолжение табл. 9.6

Вариант	Прибыль	Вложения				
		1	2	3	4	5
4	$\varphi_1(x)$	2,3	2,4	2,5	2,7	2,9
	$\varphi_2(x)$	1,8	2,1	2,5	2,9	3,1
	$\varphi_3(x)$	1,9	2,2	2,4	2,7	2,8
	$\varphi_4(x)$	1,7	1,9	2,3	2,5	2,9
5	$\varphi_1(x)$	2,5	2,6	2,7	2,8	2,9
	$\varphi_2(x)$	1,8	1,9	2,3	2,5	2,9
	$\varphi_3(x)$	1,9	2,1	2,5	2,6	2,7
	$\varphi_4(x)$	1,7	1,8	2,5	2,6	2,9
6	$\varphi_1(x)$	1,9	2,5	2,6	2,8	3,1
	$\varphi_2(x)$	1,7	1,9	2,2	2,7	2,8
	$\varphi_3(x)$	1,8	2,2	2,4	2,7	2,9
	$\varphi_4(x)$	1,8	1,9	2,3	2,6	2,7
7	$\varphi_1(x)$	2,2	2,5	2,7	2,9	3,2
	$\varphi_2(x)$	1,8	2,2	2,5	2,7	2,9
	$\varphi_3(x)$	1,9	2,3	2,5	2,8	3,1
	$\varphi_4(x)$	2,2	2,3	2,4	2,6	2,9
8	$\varphi_1(x)$	2,1	2,4	2,5	2,7	3,1
	$\varphi_2(x)$	1,9	2,1	2,4	2,7	2,8
	$\varphi_3(x)$	1,8	2,2	2,3	2,6	3,2
	$\varphi_4(x)$	2,3	2,4	2,5	2,7	2,9
9	$\varphi_1(x)$	1,8	2,1	2,3	2,5	2,8
	$\varphi_2(x)$	2,1	2,2	2,4	2,8	3,1
	$\varphi_3(x)$	1,8	2,1	2,3	2,5	3,1
	$\varphi_4(x)$	2,1	2,3	2,5	2,6	3,1
10	$\varphi_1(x)$	1,7	2,2	2,5	2,7	2,9
	$\varphi_2(x)$	2,2	2,4	2,6	2,9	3,2
	$\varphi_3(x)$	1,9	2,3	2,4	2,6	3,3
	$\varphi_4(x)$	2,2	2,5	2,7	2,9	3,1
11	$\varphi_1(x)$	1,9	2,1	2,4	2,8	3,1
	$\varphi_2(x)$	2,1	2,5	2,6	3,1	3,2
	$\varphi_3(x)$	1,8	2,2	2,5	2,6	3,1
	$\varphi_4(x)$	2,3	2,5	2,8	3,0	3,2
12	$\varphi_1(x)$	1,8	2,2	2,3	2,5	3,2
	$\varphi_2(x)$	2,2	2,4	2,6	2,8	3,1
	$\varphi_3(x)$	1,9	2,1	2,4	2,5	3,2
	$\varphi_4(x)$	2,4	2,5	2,9	3,1	3,3

Окончание табл. 9.6

Вариант	Прибыль	Вложения				
		1	2	3	4	5
13	$\varphi_1(x)$	1,7	2,1	2,5	2,6	3,1
	$\varphi_2(x)$	2,2	2,5	2,7	2,9	3,0
	$\varphi_3(x)$	2,0	2,1	2,3	2,5	3,0
	$\varphi_4(x)$	2,1	2,3	2,7	3,1	3,2
14	$\varphi_1(x)$	1,9	2,2	2,3	2,5	3,0
	$\varphi_2(x)$	2,1	2,3	2,8	2,9	3,1
	$\varphi_3(x)$	2,2	2,5	2,7	2,9	3,0
	$\varphi_4(x)$	2,2	2,3	2,7	3,2	3,3
15	$\varphi_1(x)$	2,0	2,1	2,3	2,6	3,1
	$\varphi_2(x)$	2,2	2,5	2,8	2,9	3,2
	$\varphi_3(x)$	2,3	2,5	2,8	3,0	3,3
	$\varphi_4(x)$	2,3	2,4	2,5	3,1	3,3

Контрольные вопросы

1. Что собой представляет динамическое программирование?
2. В каких сферах могут применяться модели динамического программирования?
3. Опишите общую постановку задачи распределения инвестиций.
4. Как выглядит математическая модель задачи распределения инвестиций?
5. Из каких этапов складывается решение задач методом прогонки?
6. Опишите решение методом прямой прогонки.
7. Опишите решение методом обратной прогонки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Васильев, Ф. П.* Численные методы решения экстремальных задач : учеб. пособие для вузов / П. Ф. Васильев. — 2-е изд., перераб. и доп. — М. : Наука,1988. — 552 с.

2. *Ермаков, С. М.* Метод Монте-Карло и смежные вопросы / С. М. Ермаков. — М. : Наука, 1971. — (Серия «Теория вероятностей и математическая статистика»). — 120 с.

3. *Зайцев, М. Г.* Методы оптимизации управления и принятия решений: примеры, задачи, кейсы : учеб. пособие / М. Г. Зайцев, С. Е. Варюхин. — 5-е изд., испр. и дополн. — М. : Издат. дом Дело, 2017. — 640 с.

4. *Захарченко, Н. С.* Экономико-математические методы : учеб. пособие для вузов по специальности «Экономика и управление на предприятии» / Н. С. Захарченко. — Новочеркасск, 2005. — 64 с.

5. *Кузнецов, Б. Т.* Финансовый менеджмент : учеб. пособие / Б. Т. Кузнецов. — М. : ЮНИТИ-ДАНА, 2005. — 415 с.

6. *Лазарев, А А.* Теория расписаний. Методы и алгоритмы / А. А. Лазарев ; Мин. науки и высш. образования России, Ин-т проблем управления им. В. А. Трапезникова Рос. акад. наук. — М. : ИПУ, 2019. — 407 с.

7. *Леонтьева, Л. С.* Производственный менеджмент : учебник и практикум для приклад. бакалавриата / Л. С. Леонтьева, В. И. Кузнецова. — М. : Изд-во Юрайт, 2015. — 305 с.

8. *Малышев, В. В.* Методы оптимизации по вероятностным критериям : учеб. пособие / В. В. Малышев, К. А. Карп — Москов. авиац. ин-т им. Серго Орджоникидзе. — М. : МАИ, 1994. — 439 с.

9. *Матренин, П. В.* Методы стохастической оптимизации : учеб. пособие / П. В. Матренин, М. Г. Гриф, В. Г. Секаев. — Новосибирск: Изд-во НГТУ, 2016. — 67 с.

10. Менеджмент: методы принятия управленческих решений : учеб. пособие для среднего профес. образования / П. В. Иванов [и др.] ; под ред. П. В. Иванова. — 2-е изд., испр. и доп. — М. : Изд-во Юрайт, 2020. — 276 с.

11. *Мурашко, В. С.* Методы исследования операций в машиностроении : пособие по курсу «Математическое моделирование и методы исследования операций» для студентов специальности 1-53 01 01 «Автоматизация технологических процессов и производств (по направлениям)» дневной формы обучения / В. С. Мурашко. — Гомель : ГГТУ им. П. О. Сухого, 2018. — 115 с.

12. *Николаенко, С. И.* Самообучающиеся системы / С. И. Николаенко, А. Л. Тулупьев. — М. : МЦНМО, 2009. — 288 с.

13. *Пантелеев, А. В.* Методы оптимизации в примерах и задачах : учеб. пособие / А. В. Пантелеев, Т. А. Летова. — 3-е изд., стер. — М. : Высш. шк., 2008. — 544 с.

14. *Пантелеев, А. В*. Методы глобальной оптимизации. Метаэвристические стратегии и алгоритмы / А. В. Пантелеев, Д. В. Метлицкая, Е. А. Алешина. — М. : ВУЗовск. книга, 2013. — 244 с.

15. Решение сложных задач коммивояжера методами функциональных гибридных интеллектуальных систем / А. В. Колесников [и др.] ; Рос. акад. наук, Ин-т проблем информатики. — М. : ИПИ, 2011. — 295 с.

16. *Рубчинский, А. А.* Дискретные математические модели. Начальные понятия и стандартные задачи : учеб. пособие / А. А. Рубчинский. — М. : Директ-Медия, 2014. — 269 с.

17. *Сазерленд, Дж.* Scrum на практике. Высокая продуктивность и результаты — прямо сейчас / Дж. Сазерленд : пер. с англ. К. Пантелеевой ; под науч. ред. М. Вязанкин, Л. Гончаров]. — М. : Манн, Иванов и Фербер, 2021. — 256 с.

18. *Смагин, Б. И.* Экономико-математические методы : учебник для академ. бакалавриата / Б. И. Смагин. — 2-е изд., испр. и доп. — М. : Изд-во Юрайт, 2017. — 272 с.

19. *Смуров, С. И.* Методы оптимизации : метод. указания и задания к практ. занятиям и лаб. работам / С. И. Смуров, Т. В. Сокольская, В. А. Бобкова. — Иваново, 1990. — 72 с.

20. *Трифонов, А. Г.* Постановка задачи оптимизации и численные методы ее решения / А. Г. Трифонов. — М. : Дело, 2003. — 382 с.

21. Цифровая трансформация. Основные понятия и терминология : сб. статей / Нац. акад. наук Беларуси, Объединен. ин-т проблем информатики ; редкол.: А. В. Тузиков (председатель) и др.. — Минск : Беларус. навука, 2020. — 266 с.