Санкт-Петербургский государственный политехнический университет
Институт компьютерных наук и технологий
Кафедра «Компьютерные системы и программные технологии»

К У Р С О В А Я   Р А Б О Т А

Дисциплина: **Методы оптимизации**

Тема: **Формулировка и решение задачи выбора оптимального решения с использованием различных математических моделей**

Выполнил
студент гр.13541/3 <*подпись*> М.А. Олейник

Руководитель
доцент, к.т.н. <*подпись*> А.Г. Сиднев

«___» __________ 2018 г.

Санкт-Петербург

2018

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

ЗАДАНИЕ
НА ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ

студенту группы 13541/3 Олейник Максим Александрович
(номер группы) *(фамилия, имя, отчество)*

1. Тема работы: Формулировка и решение задачи выбора оптимального решения с использованием различных математических моделей

2. Срок сдачи законченного проекта *28.05.2018*

3. Исходные данные к проекту (работе): Содержательные описания задач поиска оптимальных решений и задачи анализа сетевого графа

4. Содержание пояснительной записки (перечень подлежащих разработке вопросов): введение, формализация и решение многокритериальной оптимизационной задачи, поиск оптимальной стратегии принятия решений с использованием марковских моделей, анализ потокового графа, оптимизация сети систем массового обслуживания, заключение, список использованных источников.

Дата получения задания: 5 февраля 2018 г.

Руководитель __________________ Сиднев А. Г.
(подпись) *(инициалы, фамилия)*

Задание принял к исполнению ______________ Олейник М. А.
(подпись студента) *(инициалы, фамилия)*

5 февраля 2018 г
(дата)

Содержание

Введение

Задачи оптимизации встречаются во многих областях науки, техники и экономики. В процессе проектирования ставится обычно задача определения наилучших, в некотором смысле, структуры или значений параметров объектов. Такая задача называется оптимизационной. Если оптимизация связана с расчётом оптимальных значений параметров при заданной структуре объекта, то она называется параметрической оптимизацией. Задача выбора оптимальной структуры является структурной оптимизацией.

Оптимизация – в математике, информатике и исследовании операций задача нахождения экстремума (минимума или максимума) целевой функции в некоторой области конечномерного векторного пространства, ограниченной набором линейных и/или нелинейных равенств и/или неравенств.

Задача многокритериальной оптимизации, задача поиска оптимальной стратегии принятия решений с использованием марковских моделей и задача поиска оптимальных параметров сети систем массового обслуживания непосредственно являются задачами, решив которые можно будет получить информацию о наилучших (оптимальных) значениях параметров системы и тех решениях, которые необходимо принять, чтобы достигнуть этих значений. Задача анализа потокового графа с использованием методики GERT и алгебры потоковых графов также может использоваться для оптимизации процессов.

1 Формализация многокритериальной оптимизационной задачи, методы сведения к однокритериальной, решение с использованием Optimization Toolbox системы Matlab

1.1 Постановка задачи

Фабрика производит два вида красок: первый – для наружных, а второй – для внутренних работ.

Для производства красок используются два ингредиента: A и B. Максимально возможные суточные запасы этих ингредиентов составляют 6 и 8 тонн соответственно. Известны расходы ингредиентов A и B на производство 1 тонны соответствующих красок (табл. 1.1).

Таблица 1.1. Расход ингредиентов.

Ингредиент	Расход, т		Запас, т
	Краска для наружных работ	Краска для внутренних работ	
А	1	2	6
Б	2	1	8

Изучение рынка сбыта показало, что спрос на краску для внутренних работ никогда не превышает 4 тонны в сутки. Оптовая цена одной тонны краски равна 3000 руб. для первого вида и 2000 руб. за краску второго вида. Розничная цена одной тонны краски первого вида равна 5500 руб., а второго вида – 3000 руб.

Какое количество краски каждого вида надо производить в условиях ограниченного количества ингредиентов А и В, чтобы доход от оптовой реализации продукции был максимальным? Какое количество краски каждого вида надо производить, чтобы доход от розничной реализации продукции был максимальным? Какое количество краски каждого вида надо производить, чтобы остаток ингредиента А на складе был минимальным? Примечание. Своеобразное ведение бухучета на фабрике позволяет продать одну и ту же произведенную продукцию дважды (во всяком случае, по документам): по оптовым ценам и по розничным ценам.

Пункты расчетного задания.

1. Осуществить переход от многокритериальной задачи к однокритериальной с использованием следующих подходов:

A. Выделение главного критерия
B. Свертка критериев (аддитивная и мультипликативная)
C. Максимин или минимакс (он же метод максиминной свертки)
D. Метод последовательных уступок
E. fgoalattain
F. Ведение метрики в пространстве критериев

2. Решить задачу стохастического программирования для одной из однокритериальных задач, превратив детерминированное ограничение в вероятностное по схеме

$$\boldsymbol{P\Big(\sum_{j=1}^{n} a_{ij} x_j - b_i \le 0\Big) \ge \alpha_i}$$

Менять α_i в следующем диапазоне $\mathbf{0{,}1 \le \alpha_i \le 0{,}9}$

Считать случайной величиной $\boldsymbol{b_i}$ или элементы $\{\boldsymbol{a_{ij}}\}$ $\boldsymbol{i}$-й строки матрицы $\boldsymbol{A}\{\boldsymbol{a_{ij}}\}$ (по выбору).

Разрешается изменить формулировку исходной задачи, придумать собственную задачу, найти другую аналогичную задачу, которая могла бы быть сформулирована как многокритериальная.

1.2 Математическая модель задачи

Обозначения:

1. N_i – кол-во производимой краски типа i (для наружных (1) и внутренних (2) работ) в сутки.

Целевые функции:

1. $y1 = 3000N_1 + 2000N_2 \rightarrow \max$ – доход от оптовой реализации продукции;
2. $y2 = 5500N_1 + 3000N_2 \rightarrow \max$ – доход от розничной реализации продукции;
3. $y3 = 6 - (N_1 + 2N_2) \rightarrow \min$ – остаток ингредиента А на складе.

Ограничения:

1. $N_1 + 2N_2 \le 6$ – максимально возможные суточные запасы ингредиента A;
2. $2N_1 + N_2 \le 8$ – максимально возможные суточные запасы ингредиента B;
3. $N_2 \le 4$ – спрос на краску для внутренних работ не превышает 4 в сутки;

Перепишем ограничения задачи в матричном виде:

$$\begin{pmatrix}1 & 2\\ 2 & 1\\ 0 & 1\end{pmatrix} * \begin{pmatrix}N_1\\ N_2\end{pmatrix} \le \begin{pmatrix}6\\ 8\\ 4\end{pmatrix};$$

Целевые функции y1 и y2 можно представить в виде:

$neg1 = -3000N_1 - 2000N_2 \rightarrow \min$

neg2 = $-5500N_1 - 3000N_2 \rightarrow \min$

1.3 Поиск оптимального решения для каждого критерия в отдельности

Найдем оптимальные решения для каждого критерия независимо от других. Для этого решим три задачи однокритериальной оптимизации для трёх критериев при тех же ограничениях, что и для задачи многокритериальной оптимизации.

Напишем для этого скрипт:

Листинг 1.1 – Скрипт в Матлаб.

```
% Параметры
stockA = 6; % суточные запасы ингредиента A
stockB = 8; % суточные запасы ингредиента B
optPrice1 = 3000; % оптовая цена краски 1
optPrice2 = 2000; % оптовая цена краски 2
price1 = 5500; % розничная цена краски 1
price2 = 3000; % розничная цена краски 2
supply2 = 4; % спрос на краску для внутренних работ не превышает
n1A = 1; n1B = 2; % расходы ингредиентов на краску 1
n2A = 2; n2B = 1; % расходы ингредиентов на краску 2
% Целевые функции
y1 = @(N) optPrice1*N(1) + optPrice2*N(2); % → max
y2 = @(N) price1*N(1) + price2*N(2); % → max
y3 = @(N) stockA - (N(1) + 2*N(2)); % → min
% Отрицательные целевые функции
neg1 = @(N) -y1(N); % → min
neg2 = @(N) -y2(N); % → min
% Функциональные ограничения
A = [n1A, n1B; n2A, n2B; 0, 1];
b = [stockA; stockB; supply2];
Aeq = [];
beq = [];
% Параметрические ограничения для fmincon
lb = [0; 0]; % нижняя граница
ub = [Inf; Inf]; % верхняя граница
% Поиск оптимального решения для критериев в отдельности
startPoint = lb;
[N1, fval1] = fmincon(neg1, startPoint, A, b, Aeq, beq, lb, ub)
[N2, fval2] = fmincon(neg2, startPoint, A, b, Aeq, beq, lb, ub)
[N3, fval3] = fmincon(y3, startPoint, A, b, Aeq, beq, lb, ub)
```

В результате получили:

Листинг 1.2 – Результат работы скрипта.

```
Local minimum found that satisfies the constraints.
N1 =
```

```
    3.3333
    1.3333
fval1 =
  -1.2667e+04
Local minimum found that satisfies the constraints.
N2 =
    3.3333
    1.3333
fval2 =
  -2.2333e+04
Local minimum found that satisfies the constraints.
N3 =
    1.6013
    2.1994
fval3 =
   1.0000e-05
```

Полученные оптимальные решения:

fval1 = -1.2667e+04; fval2 = -2.2333e+04; fval3 = 1.0e-05.

Получаем оптимальные значения для критериев:

y1 = 12667 руб. – максимальный доход от оптовой продукции;

y2 = 22333 руб. – максимальный доход от розничной продукции;

y3 = 0.00001 т – минимальный остаток ингредиента А.

Рассчитаем для всех полученных N1 и N2 критерии (табл. 1.1):

Таблица 1.1 – Результаты расчетов.

Главный крит.	N1	N2	y1	y2	y3
y1	3,3333	1,3333	12667	22333	0,0001
y2	3,3333	1,3333	12667	22333	0,0001
y3	1,60125	2,19937	9202	15405	0,00001

Здесь и далее будут использоваться абсолютные, округленные значения N1 и N2 при расчетах и предоставлении результатов, это гарантирует, что для одинаковых значений N1 и N2 будут получены одинаковые результаты y1, y2, y3.

1.4 Выделение главного критерия

При этом подходе необходимо выбрать наиболее важный критерий, а на оставшиеся поставить ограничения. Выберем критерий y2 (доход от розничной реализации продукции), поскольку именно от розничной продажи можно получить наибольшую выгоду:

$5500N_1 + 3000N_2 \rightarrow max;$

Границы других критериев мы уже получили путем вычисления оптимального решения задачи для критериев по отдельности:

$-(3000N_1 + 2000N_2) \leq -12667$

$6 - (N_1 + 2N_2) \leq 0.00001$

Для того, чтобы главный критерий оставался главным, снизим значимость найденных границ других критериев на 10%.

$-(3000N_1 + 2000N_2) \leq -11400$

$- (N_1 + 2N_2) \leq -5{,}99999$

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} N_1 \\ N_2 \end{pmatrix} \leq \begin{pmatrix} 6 \\ 8 \\ 4 \end{pmatrix};$$

Данные ограничения и задачу можно переписать в виде:

$$5500N_1 + 3000N_2 \rightarrow max;$$

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 0 & 1 \\ -3000 & -2000 \\ -1 & -2 \end{pmatrix} * \begin{pmatrix} N_1 \\ N_2 \end{pmatrix} \leq \begin{pmatrix} 6 \\ 8 \\ 4 \\ -11400 \\ -5{,}99999 \end{pmatrix};$$

Напишем скрипт для решения этой задачи:

Листинг 1.3 – Скрипт для Матлаб.

```
% Параметры
stockA = 6; % суточные запасы ингредиента A
stockB = 8; % суточные запасы ингредиента B
optPrice1 = 3000; % оптовая цена краски 1
optPrice2 = 2000; % оптовая цена краски 2
price1 = 5500; % розничная цена краски 1
price2 = 3000; % розничная цена краски 2
supply2 = 4; % спрос на краску для внутренних работ не превышает
n1A = 1; n1B = 2; % расходы ингредиентов на краску 1
n2A = 2; n2B = 1; % расходы ингредиентов на краску 2
% Целевые функции
y1 = @(N) optPrice1*N(1) + optPrice2*N(2); % → max
y2 = @(N) price1*N(1) + price2*N(2); % → max
y3 = @(N) stockA - (N(1) + 2*N(2)); % → min
```

```
% Отрицательные целевые функции
neg1 = @(N) -y1(N); % → min
neg2 = @(N) -y2(N); % → min
% Функциональные ограничения
A = [n1A, n1B; n2A, n2B; 0, 1; -3000, -2000; -1, -2];
b = [stockA; stockB; supply2; -11400; -5.99999];
Aeq = [];
beq = [];
% Параметрические ограничения для fmincon
lb = [0; 0]; % нижняя граница
ub = [Inf; Inf]; % верхняя граница
% Выделение главного критерия
startPoint = lb;
[N, fval] = fmincon(neg2, startPoint, A, b, Aeq, beq, lb, ub)
y1(N)
y2(N)
y3(N)
```

В результате получаем:

Листинг 1.4 – Результат расчетов.

```
Local minimum found that satisfies the constraints.
N =
    3.3333
    1.3333
fval =
  -2.2333e+04
ans =
   1.2667e+04
ans =
   2.2333e+04
ans =
   1.0000e-04
```

Были также посчитаны значения критериев для найденного решения. Способ выделения главного критерия позволил получить следующие ответы:

N_1 = 3.3333 т – кол-во краски 1;
N_2 = 1.3333 т – кол-во краски 2;
y1 = 12667 руб. – выручка от оптовых продаж;
y2 = 22333 руб. – выручка от розничных продаж;
y3 = 0.0001 т – остаток ингредиента А.

Давайте проанализируем полученный результат. Фабрике необходимо производить 3.3333 тонны краски первого типа и 1.3333 тонны краски второго типа, чтобы остаток ингредиента А на складе был минимальным (0.0001 кг), а прибыли от оптовых и розничных продаж были максимальными (12667 руб. и 22333 руб. соответственно) (табл. 2.2).

Таблица 1.2 – Результаты расчетов.

N1	N2	y1	y2	y3
3,3333	1,3333	12667	22333	0,0001

1.5 Свёртка критериев

1.5.1 Аддитивная свертка критериев

В случае использования аддитивных преобразований свёртка выглядит следующим образом:

$$C(a) = \sum_{i=1}^{m} w_i C_i(a)$$

здесь $w_1 + w_2 + \cdots + w_m = 1$.

Также можно использовать формулу с нормированными критериями:

$$C(a) = \sum_{i=1}^{m} w_i C_i^n(a)$$

здесь $C_i^n(a) = {C_i(a)}\big/{C_i^*}$, где C_i^* - оптимальное решение задачи по каждому критерию в отдельности.

Выполнить нормировку критериев и сделаем их значения соизмеримыми, а единицы измерения – безразмерными: вычислим оптимальные решения задачи для каждого критерия в отдельности, далее в fmincon передадим сумму нормированных значений (1-ый критерий делится на fval1, 2-ой на fval2, а 3-ий на fval3 - на оптимальные решения задачи для каждого критерия в отдельности) Где значения fval1, fval2, fval3?, каждое из которых умножено на определенный весовой коэффициент (0,3 для 1-го критерия, 0,4 для 2-го и 0,3 для 3-го, считаем критерий выручки розничной продажи немного важнее остальных).

Значения критериев fval1, fval2, fval3 получены в пункте 2.1:

fval1 = -12667

fval2 = -22333

fval3 = 0.00001

$$0.3 * neg1^{н} + 0.4 * neg2^{н} + 0.3 * y3^{н} \rightarrow \min;$$

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 0 & 1 \\ -3000 & -2000 \\ -1 & -2 \end{pmatrix} * \begin{pmatrix} N_1 \\ N_2 \end{pmatrix} \le \begin{pmatrix} 6 \\ 8 \\ 4 \\ -11400 \\ -5{,}99999 \end{pmatrix};$$

Скрипт задачи:

Листинг 1.5 – Скрипт для Матлаб.

```
% Параметры
stockA = 6; % суточные запасы ингредиента A
stockB = 8; % суточные запасы ингредиента B
optPrice1 = 3000; % оптовая цена краски 1
optPrice2 = 2000; % оптовая цена краски 2
price1 = 5500; % розничная цена краски 1
price2 = 3000; % розничная цена краски 2
supply2 = 4; % спрос на краску для внутренних работ не превышает
n1A = 1; n1B = 2; % расходы ингредиентов на краску 1
n2A = 2; n2B = 1; % расходы ингредиентов на краску 2
% Целевые функции
y1 = @(N) optPrice1*N(1) + optPrice2*N(2);
y2 = @(N) price1*N(1) + price2*N(2);
y3 = @(N) stockA - (N(1) + 2*N(2));
% Отрицательные целевые функции
neg1 = @(N) -y1(N);
neg2 = @(N) -y2(N);
% Функциональные ограничения
A = [n1A, n1B; n2A, n2B; 0, 1];
b = [stockA; stockB; supply2];
Aeq = [];
beq = [];
% Параметрические ограничения для fmincon
lb = [0; 0]; % нижняя граница
ub = [Inf; Inf]; % верхняя граница
% Поиск оптимального решения для критериев в отдельности
startPoint = lb;
[N1, fval1] = fmincon(neg1, startPoint, A, b, Aeq, beq, lb, ub) ;
[N2, fval2] = fmincon(neg2, startPoint, A, b, Aeq, beq, lb, ub);
[N3, fval3] = fmincon(y3, startPoint, A, b, Aeq, beq, lb, ub);
% Нормировка критериев
neg1_norm = @(N) neg1(N)/abs(fval1);
neg2_norm = @(N) neg2(N)/abs(fval2);
y3_norm   = @(N) y3(N)/abs(fval3);
% Аддитивная свертка
f = @(N) 0.3*neg1_norm(N) + 0.4*neg2_norm(N) + 0.3*y3_norm(N);
A = [n1A, n1B; n2A, n2B; 0, 1; -3000, -2000; -1, -2];
b = [stockA; stockB; supply2; -11400; -5.99999];
[Nf, f_opt] = fmincon(f, startPoint, A, b, Aeq, beq, lb, ub)
y1(Nf)
y2(Nf)
```

```
y3(Nf)
```

Результаты выполнения:

Листинг 1.6 – Результаты расчетов.

```
Local minimum found that satisfies the constraints.
Nf =
    3.3333
    1.3333
f_opt =
   -0.7000
ans =
   1.2667e+04
ans =
   2.2333e+04
ans =
   1.0000e-4
```

Метод аддитивной свертки дал решение (табл. 1.3):

N_1 = 3.3333 т – кол-во краски 1; N_2 = 1.3334 т – кол-во краски 2;

y1 = 12667 руб. – выручка от оптовых продаж;

y2 = 22333 руб. – выручка от розничных продаж;

y3 = 0.0001 т – остаток ингредиента А.

Таблица 1.3 – Результаты расчетов.

N1	N2	y1	y2	y3
3,3333	1,3333	12667	22333	0,0001

1.5.2 Мультипликативная свертка критериев

В случае использования мультипликативной свёртки обобщённый критерий формируется следующим образом:

$$\boldsymbol{F(x) = \prod_{i=1}^{r} f_i[x]^{\lambda_i}},$$

где $\boldsymbol{\lambda_i}$ – некоторые вещественные числа.

Мультипликативный критерий иногда представляется в виде отношения произведений частных критериев (выходных параметров) [2]:

$$F(X) = \frac{\prod_{i=1}^{m_1} F_i^{+}(X)}{\prod_{j=1}^{m_2} F_j^{-}(X)}, \quad m_1+m_2=m;$$

Откуда эта формула?

где в числителе перемножаются все выходные параметры, требующие максимизации и имеющие ограничения $F_i^+(X) \geq TT_i$, а в знаменателе – все выходные параметры, требующие минимизации и имеющие ограничения $F_i^-(X) \leq TT_i$., где TT_i – значение технического требования, предъявленного к i– му критерию[1].

Произведение критериев y1 и y2 будет находится в числителе, а критерий y3 будет в знаменателе. Этот «супер-критерий» необходимо максимизировать.

$$\mathrm{Fm} = \frac{y1 * y2}{y3} \rightarrow \max;$$

Задача будет выглядеть так:

$$(-\mathrm{Fm}) \rightarrow \min;$$

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} N_1 \\ N_2 \end{pmatrix} \leq \begin{pmatrix} 6 \\ 8 \\ 4 \end{pmatrix};$$

Напишем скрипт для решения задачи:

Листинг 1.7 – Скрипт для Матлаб.

```
% Параметры
stockA = 6; % суточные запасы ингредиента A
stockB = 8; % суточные запасы ингредиента B
optPrice1 = 3000; % оптовая цена краски 1
optPrice2 = 2000; % оптовая цена краски 2
price1 = 5500; % розничная цена краски 1
price2 = 3000; % розничная цена краски 2
supply2 = 4; % спрос на краску для внутренних работ не превышает
n1A = 1; n1B = 2; % расходы ингредиентов на краску 1
n2A = 2; n2B = 1; % расходы ингредиентов на краску 2
% Целевые функции
y1 = @(N) optPrice1*N(1) + optPrice2*N(2); % ? max
y2 = @(N) price1*N(1) + price2*N(2); % ? max
y3 = @(N) stockA - (N(1) + 2*N(2));
% Функциональные ограничения
A = [n1A, n1B; n2A, n2B; 0, 1];
b = [stockA; stockB; supply2];
Aeq = [];
beq = [];
% Параметрические ограничения для fmincon
lb = [0; 0]; % нижняя граница
ub = [Inf; Inf]; % верхняя граница
startPoint = lb;
```

```
% Мультипликативная свертка
f = @(N) y1(N)*y2(N)/y3(N);
f1 = @(N) -f(N);
[Nf, f_opt] = fmincon(f1, startPoint, A, b, Aeq, beq, lb, ub)
y1(Nf)
y2(Nf)
y3(Nf)
```

Результат выполнения:

Листинг 1.8 – Результаты расчетов.

```
Local minimum possible. Constraints satisfied.
Nf =
    3.3204
    1.3398
f_opt =
  -1.3135e+13
ans =
   1.2641e+04
ans =
   2.2282e+04
ans =
   4.0000e-05
```

Метод мультипликативной свертки дал решение (табл. 1.4):

N_1 = 3.32044 т – кол-во краски 1; N_2 = 1. 33976 т – кол-во краски 2;

y1 = 12641 руб. – выручка от оптовых продаж;

y2 = 22282 руб. – выручка от розничных продаж;

y3 = 0.00005 т – остаток ингредиента А.

По результатам расчета по двум метода можно сказать, что метод аддитивной свертки привел к максимизации оптовой и розничной прибыли до оптимального уровня, в то время как метод мультипликативной свёртки получил более компромиссное решение, которое имеет третий критерий ближе к оптимуму.

Таблица 1.4 – Результаты расчетов.

N1	N2	y1	y2	y3
3.32044	1. 33976	12641	22282	0.00005

1.6 Минимакс

Минимаксный критерий используется в случаях, когда требуется в любых условиях избежать повышенного риска. Предпочтение отдается решению, для которого максимальные потери при различных вариантах обстановки окажутся минимальными. Формализованное выражение:

$$\min_{x} \max_{\{F_i\}} \{F_i(x)\}$$

Некорректное использование метода максимин. Исправлено.

Так как оцениваемые показатели разновелики, необходимо нормировать критерии. Критерии нормированы, так же, как и в предыдущем пункте.

Значения критериев fval1, fval2, fval3 получены в пункте 2.1:

fval1 = 12667

fval2 = 22333

fval3 = 0.00001

Целевые функции примут вид:

$$f1 = \left((3000N_1 + 2000N_2)/12667\right)^{-1}$$

$$f2 = \left((5500N_1 + 3000N_2)/22333\right)^{-1}$$

$$f3 = \left(6 - (N_1 + 2N_2)\right)/0.00001$$

Тогда решение задачи минимаксного критерия будет выглядеть так:

$$max\{f1; f2; f3\} \rightarrow \text{min};$$

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} N_1 \\ N_2 \end{pmatrix} \le \begin{pmatrix} 6 \\ 8 \\ 4 \end{pmatrix};$$

Скрипт решения задачи:

Листинг 1.9 – Скрипт для Матлаб.

```
% Целевые функции
y1 = @(N) optPrice1*N(1) + optPrice2*N(2); % ? max
y2 = @(N) price1*N(1) + price2*N(2); % ? max
y3 = @(N) stockA - (N(1) + 2*N(2)); % ? min
% Функциональные ограничения
A = [n1A, n1B; n2A, n2B; 0, 1];
b = [stockA; stockB; supply2];
Aeq = [];
beq = [];
% Параметрические ограничения для fminimax
lb = [1; 1]; % нижняя граница
```

```
ub = [Inf; Inf]; % верхняя граница
startPoint = lb;
% Оптимальные результаты
fval1 = 12667;
fval2 = 22333;
fval3 = 0.00001;
% Функции для минимакса
f1 = @(N) 1/((3000*N(1) + 2000*N(2))/12667);
f2 = @(N) 1/((5500*N(1) + 3000*N(2))/22333);
f3 = @(N) (6 - (N(1) + 2*N(2)))/0.00001;
% Минимакс
f = @(N) [f1(N), f2(N), f3(N)];
A = [n1A, n1B; n2A, n2B; 0, 1];
b = [stockA; stockB; supply2];
[Nf, f_opt] = fminimax(f, startPoint, A, b, Aeq, beq, lb, ub)
y1(Nf)
y2(Nf)
y3(Nf)
```

Результат решения:

Листинг 1.10 – Результаты расчетов.

```
Nf =
    3.3333
    1.3333
f_opt =
   1.0000    1.0000         0
ans =
   1.2667e+04
ans =
   2.2333e+04
ans =
   1.0000e-04
```

Метод минимакс дал решение (табл. 1.5):

N_1 = 3.3333 т – кол-во краски 1; N_2 = 1.3333 т – кол-во краски 2;

y1 = 12667 руб. – выручка от оптовых продаж;

y2 = 22333 руб. – выручка от розничных продаж;

y3 = 0.0001 т – остаток ингредиента А.

Таблица 1.5 – Результаты расчетов.

N1	N2	y1	y2	y3
3,3333	1,3333	12667	22333	0,0001

1.7 Метод последовательных уступков

Для метода последовательных уступок необходимо отсортировать критерии по степени важности: neg2, neg1, y3. Кроме того, нужно выбрать значение уступки, на которую предприятие готово пойти – 10%.

В этом случае, получаем такую последовательность действий:

1. Найти минимум neg2 при исходных ограничениях – эта задача была решена в п. 2.1.
2. Найти минимум neg1 в исходной системе ограничений, дополненной неравенством: $neg2 \leq neg2^{min} + \Delta_2$, где $\Delta_2 = -0.1neg2^{min} > 0$. Это -20100.
3. Найти минимум y3 в исходной системе ограничений, дополненной неравенствами: $neg2 \leq neg2^{min} + \Delta_2$, где $\Delta_2 = -0.1neg2^{min} > 0$ и $neg1 \leq neg1^{min} + \Delta_1$, где $\Delta_1 = -0.1neg1^{min} > 0$. Это -11400.

Задача принимает вид:

$$6 - (N_1 + 2N_2) \rightarrow \min$$

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \\ -5500 & -3000 \\ -3000 & -2000 \end{pmatrix} * \begin{pmatrix} N_1 \\ N_2 \end{pmatrix} \leq \begin{pmatrix} 6 \\ 8 \\ -20100 \\ -11400 \end{pmatrix};$$

Напишем скрипт для решения этой задачи:

Листинг 1.11 – Скрипт для Матлаб.

```
% Целевые функции
y1 = @(N) optPrice1*N(1) + optPrice2*N(2); % ? max
y2 = @(N) price1*N(1) + price2*N(2); % ? max
y3 = @(N) stockA - (N(1) + 2*N(2)); % ? min
% Отрицательные целевые функции
neg1 = @(N) -y1(N); % ? min
neg2 = @(N) -y2(N); % ? min
% Функциональные ограничения
A = [n1A, n1B; n2A, n2B; 0, 1];
b = [stockA; stockB; supply2];
Aeq = [];
beq = [];
% Параметрические ограничения для fmincon
lb = [0; 0]; % нижняя граница
ub = [Inf; Inf]; % верхняя граница
startPoint = lb;
% Поиск оптимального решения для критерия neg2
[N2, fval2] = fmincon(neg2, startPoint, A, b, Aeq, beq, lb, ub)
% Дополнение функциональных ограничений
A = [n1A, n1B; n2A, n2B; 0, 1;    -price1, -price2];
```

```
b = [stockA;   stockB;   supply2; (0.9*fval2)];
% Поиск оптимального решения для критерия neg1
[N1, fval1] = fmincon(neg1, startPoint, A, b, Aeq, beq, lb, ub)
% Дополнение функциональных ограничений
A = [n1A, n1B; n2A, n2B; 0, 1;    -price1, -price2; -optPrice1, -
optPrice2];
b = [stockA;   stockB;   supply2; (0.9*fval2);      (0.9*fval1)];
% Поиск оптимального решения для критерия y3
[N3, fval3] = fmincon(y3, startPoint, A, b, Aeq, beq, lb, ub)
y1(N3)
y2(N3)
y3(N3)
```

Результат выполнения:

Листинг 1.12 – Результаты вычисления.

```
N2 =
    3.3333
    1.3333
fval2 =
  -2.2333e+04
N1 =
    3.3333
    1.3333
fval1 =
  -1.2667e+04
N3 =
    3.1044
    1.4478
fval3 =
   4.0000e-07
ans =
   2.0000e-05
```

Метод последовательных уступок дал решение (табл. 1.6):

N_1 = 3.1044 т – кол-во краски 1; N_2 = 1.44779 т – кол-во краски 2;

y1 = 12209 руб. – выручка от оптовых продаж;

y2 = 21418 руб. – выручка от розничных продаж;

y3 = 0.00002 т – остаток ингредиента А.

Можно заметить, что прибыль стала меньше, однако потребление ингредиента А выросло и его меньше осталось в итоге на складе.

Таблица 1.6 – Результаты расчетов.

N1	N2	y1	y2	y3
3,1044	1,44779	12209	21418	0,00002

1.8 Метод достижения цели

fgoalattain решает задачу достижения цели, которая является одной из формулировок задач для векторной оптимизации.

Синтаксис:

$x = fgoalattain(fun, x_0, goal, weight)$, где

fun – целевая функция,

x_0 – начальные значения,

$goal$ – целевые значения,

$weight$ – веса.

Тогда задача достижения цели, решаемая функцией fgoalattain может быть сформулирована следующим образом:

$$\min_{x,\gamma} \gamma, \text{ где } \begin{cases} F(x) - w\gamma \le goal \\ G(x) \le b \end{cases},$$

где γ – скалярный критерий, называемый attain factor; F(x) = (f1(x), f2(x), … , fr(x)) – вектор критериев оптимизации; goal = (f1*, f2*, … , fr*) – вектор целей, которые могут быть как достижимыми, так и недостижимыми; w – вектор весовых коэффициентов, которые рекомендуется принимать равными |goal| для того, чтобы обеспечить ту же самую долю не- или передостижения активных целей; и G(x) ≤ b – система ограничений.

Так как цели в векторе goal могут быть недостижимы без ущерба для решения задачи, пусть goal = (fval1, fval2, fval3), w = (|fval1|, | fval2|, | fval3|).

Значения критериев fval1, fval2, fval3 получены в пункте 2.1:

fval1 = -12667

fval2 = -22333

fval3 = 0.00001

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} N_1 \\ N_2 \end{pmatrix} \le \begin{pmatrix} 6 \\ 8 \\ 4 \end{pmatrix};$$

goal = (-12667, -22333, 0.00001), w = (12667, 22333, 0.00001).

Тогда скрипт для решения задачи будет выглядеть следующим образом:

Листинг 1.13 – Скрипт для Матлаб.

```
% Целевые функции
y1 = @(N) optPrice1*N(1) + optPrice2*N(2); % ? max
y2 = @(N) price1*N(1) + price2*N(2); % ? max
y3 = @(N) stockA - (N(1) + 2*N(2)); % ? min
% Отрицательные целевые функции
neg1 = @(N) -y1(N); % ? min
neg2 = @(N) -y2(N); % ? min
% Функциональные ограничения
A = [n1A, n1B; n2A, n2B; 0, 1];
b = [stockA; stockB; supply2];
Aeq = [];
beq = [];
% Параметрические ограничения для fmincon
lb = [0; 0]; % нижняя граница
ub = [Inf; Inf]; % верхняя граница
startPoint = lb;
% Оптимальные результаты
fval1 = -12667;
fval2 = -22333;
fval3 = 0.00001;
% Метод достижения цели
f = @(N) [neg1(N), neg2(N), y3(N)];
goal = [fval1, fval2, fval3];
w = abs(goal);
startPoint = [10, 10];
[Nf, f_opt, af] = fgoalattain(f, startPoint, goal, w, A, b, Aeq, 
beq, lb, ub)
y1(Nf)
y2(Nf)
y3(Nf)
```

Результат выполнения:

Листинг 1.14 – Результат решения.

```
Nf =
    3.3333    1.3333
f_opt =
   1.0e+04 *
   -1.2667   -2.2333    0.0000
af =
   2.6320e-05
ans =
   1.2667e+04
ans =
   1.0000e-04
```

Метод последовательных уступок дал решение (табл. 1.7):

N_1 = 3.3333 т – кол-во краски 1; N_2 = 1.3333 т – кол-во краски 2;

y1 = 12667 руб. – выручка от оптовых продаж;

y2 = 22333 руб. – выручка от розничных продаж;

y3 = 0.0001 т – остаток ингредиента А.

Судя по attainfactor, цели почти достигнуты.

Таблица 1.7 – Результаты решения.

N1	N2	y1	y2	y3
3,3333	1,3333	12667	22333	0,0001

1.9 Введение метрики в пространстве критериев

Рассматривается m-мерное пространство (где m число локальных критериев), в котором априори выбирается вектор, отображающий "идеальное" решение (или, что тоже самое, "идеальная" точка, координатами которой являются "идеальные" значения (например, минимальные или максимальные значения) локальных критериев). В этом пространстве вводится некоторая метрика, с целью вычисления расстояния между вектором, отображающим рассматриваемое решения, и "идеальным". В качестве наилучшего выбирается такое решение, векторная оценка которого наиболее близка к "идеальной" точке. Недостатками метода являются произвол при выборе идеальной точки и введение метрики [2].

Определим обобщенный критерий следующим образом. Положим $a = (a_1, a_2, \dots, a_m)$, где $a_i = \min F_i(x)$. Точка a называется *идеальной*.

a = (-12667, -22333, 0.00001)

Смысл названия связан с тем, что такие точки оптимальны сразу по всем критериям – получить меньшее значение ни по одному критерию невозможно. Как правило, точка $a \in Y_D$. Зададим для всех точек $Y \in Y_D$ функцию, являющуюся евклидовым расстоянием между точками Y и a

$$\rho(y,a) = \left[\sum_{i=1}^{m} (a_i - y_i)^2 \right]^{\frac{1}{2}}.$$

За целевую функцию (обобщённый критерий) берут выражение

$$f(X) = \sum_{i=1}^{m} \lambda_i \left[a_i - F_i(X) \right]^2,$$

где λi – весовые коэффициенты.

Таким образом, задача оптимизации формулируется следующим образом [2]

$$\min \sum_{i=1}^{m} \lambda_i [a_i - F_i(X)]^2$$

С учётом нормировки и при равнозначности критериев весовые коэффициенты можно опустить, тогда формула принимает вид:

$$f = \sum_{i=1}^{m} \left(\frac{a_i - F_i(X)}{F_i^0} \right)^2 = \sum_{i=1}^{m} \left(1 - \frac{F_i(X)}{F_i^0} \right)^2$$

так как $a_i = F_i^0$.

Значения критериев fval1, fval2, fval3 получены в пункте 1.3:

fval1 = -12667; fval2 = -22333; fval3 = 0.00001

Тогда за целевую функцию (обобщенный критерий), с учётом необходимости нормировки, можно взять выражение:

$$f = \left(1 - \frac{neg1}{fval1}\right)^2 + \left(1 - \frac{neg2}{fval2}\right)^2 + \left(1 - \frac{y3}{fval3}\right)^2 \rightarrow \min$$

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} N_1 \\ N_2 \end{pmatrix} \le \begin{pmatrix} 6 \\ 8 \\ 4 \end{pmatrix};$$

Напишем скрипт для решения этой задачи:

Листинг 1.15 – Скрипт для Матлаб.

```
% Целевые функции
y1 = @(N) optPrice1*N(1) + optPrice2*N(2); % ? max
y2 = @(N) price1*N(1) + price2*N(2); % ? max
y3 = @(N) stockA - (N(1) + 2*N(2)); % ? min
% Отрицательные целевые функции
neg1 = @(N) -y1(N); % ? min
neg2 = @(N) -y2(N); % ? min
% Функциональные ограничения
A = [n1A, n1B; n2A, n2B; 0, 1];
b = [stockA; stockB; supply2];
Aeq = [];
beq = [];
% Параметрические ограничения для fmincon
lb = [1; 1]; % нижняя граница
ub = [Inf; Inf]; % верхняя граница
startPoint = lb;
% Оптимальные результаты
fval1 = -12667;
fval2 = -22333;
fval3 = 0.00001;
```

```
% Введение метрики в пространстве критериев
f = @(N) (1-neg1(N)/fval1)^2 + (1-neg2(N)/fval2)^2 + (1-
y3(N)/fval3)^2;
[Nf, f_opt] = fmincon(f, startPoint, A, b, Aeq, beq, lb, ub)
Nf = [3.0001; 1.4999];
y1(Nf)
y2(Nf)
y3(Nf)
```

Результат решения:

Листинг 1.16 – Скрипт для Матлаб.

```
Nf =
    3.0001
    1.4999
f_opt =
    0.0064
ans =
   1.2000e+04
ans =
   2.1000e+04
ans =
   1.0000e-04
```

Метод введения метрики в пространстве критериев дал решение (табл. 1.8):

N_1 = 3.0001 т – кол-во краски 1; N_2 = 1.4999 т – кол-во краски 2;

y1 = 12000 руб. – выручка от оптовых продаж;

y2 = 21000 руб. – выручка от розничных продаж;

y3 = 0.0001 т – остаток ингредиента А.

Таблица 1.8 – Результат решения.

N1	N2	y1	y2	y3
3,0001	1,4999	12000	21000	0,0001

1.10 Оценка Парето-оптимальности полученных решений

Давайте составим таблицу (табл. 1.9), включающую все полученные результаты и определим те, которые входят в множество Парето. Пусть имеются два решения X_1 и X_2. Говорят, что решение X_1 лучше (предпочтительнее, эффективнее, доминирует) решения X_2, если $F_i(X_1) <= F_i(X_2)$ для всех $i=1,m$, и хотя бы для одного j-го критерия выполняется строгое неравенство $F_i(X_1) < F_i(X_2)$.

Таблица 1.9 – Сводная таблица результатов.

	Метод	y1, руб	y2, руб	y3, т	N1, т	N2, т
1	Выделение главного критерия	12667	22333	0,0001	3,3333	1,3333

2	Аддитивная свертка	12667	22333	0,0001	3,3333	1,3333
3	Мультипликативная свертка	12641	22282	0.00005	3,32044	1,33976
4	Минимакс	12667	22333	0,0001	3,3333	1,3333
5	Метод последовательных уступок	12209	21418	0.00002	3,1044	1,44779
6	Метод достижения цели (fgoalattain)	12667	22333	0,0001	3,3333	1,3333
7	Введение метрики в пространстве критериев	12000	21000	0,0001	3,0001	1,4999

Можно увидеть, что методы 1, 2, 4, 6 доминируют над остальными методами по критериям y1 и y2, а метод 5 доминирует над остальными по критерию y3. Они входят в множество Парето.

1.11 Решение задачи стохастического программирования

Решим задачу стохастического программирования для однокритериальной задачи, полученной вследствие применения метода аддитивной свёртки, превратив детерминированное ограничение в вероятностное по схеме

$$\boldsymbol{P}\left(\sum_{j=1}^{n} \boldsymbol{a}_{ij}\boldsymbol{x}_j - \boldsymbol{b}_i \leq \boldsymbol{0}\right) \geq \boldsymbol{\alpha}_i$$

Менять $\boldsymbol{\alpha}_i$ будем в следующем диапазоне $\boldsymbol{0{,}1 \leq \alpha_i \leq 0{,}9}$

Будем считать случайной величиной $\boldsymbol{b}_i$.

По условию, $N_2 \leq$ supply2 – спрос на краску для внутренних работ не превышает 4 в сутки. Так как спрос может колебаться, давайте преобразуем это ограничение в вероятностное. Пусть спрос на краску второго типа является независимой нормально распределенной случайной величиной:

$supply2 = N(\overline{supply2}, \sigma 2)$,

где $\overline{supply2} = 4$ т, $\sigma = 0{,}6$ т – параметры распределения случайной величины. Другие ограничения и критерии не зависят от $supply2$. Тогда третье ограничение системы преобразуется в вероятностное, и можно рассматривать его в виде выражения:

$P(N_2 \leq supply2) \geq \alpha$,

где $0 \leq \alpha \leq 1$ – доверительная вероятность.

Тогда исходное вероятностное ограничение равно детерминированному линейному **[1]**:

$N_2 \le M(supply2) - K_\alpha\sqrt{D(supply2)},$

где

$$M(supply2) = \overline{supply2} = 4$$

$$D(supply2) = \sigma^2 = 0{,}36$$

Детерминированное линейное ограничение выглядит как:

$$N_2 \le 4 - K_\alpha\sqrt{0{,}36}$$

$$N_2 \le 4 - K_\alpha * 0{,}6$$

$$N_2 - 4 + K_\alpha * 0{,}6 \le 0$$

Целевые функции:

$$f1 = \left((3000N_1 + 2000N_2)/12667\right)^{-1}$$

$$f2 = \left((5500N_1 + 3000N_2)/22333\right)^{-1}$$

$$f3 = \left(6 - (N_1 + 2N_2)\right)/0.00001$$

Итоговая задача:

$$0.3*f1 + 0.4*f2 + 0.3*f3 \to \min;$$

$$N_2 - 4 + K_\alpha * 0{,}6 \le 0$$

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \\ -1 & -2 \\ -3000 & -2000 \end{pmatrix} * \begin{pmatrix} N_1 \\ N_2 \end{pmatrix} \le \begin{pmatrix} 6 \\ 8 \\ -5{,}99999 \\ -11400 \end{pmatrix};$$

Решим набор таких задач для α = 0.1, 0.2, … , 0.9:

Листинг 1.17 – Скрипт для Матлаб.

```
% Параметры
stockA = 6; % суточные запасы ингредиента A
stockB = 8; % суточные запасы ингредиента B
optPrice1 = 3000; % оптовая цена краски 1
optPrice2 = 2000; % оптовая цена краски 2
price1 = 5500; % розничная цена краски 1
price2 = 3000; % розничная цена краски 2
supply2 = 4; % спрос на краску для внутренних работ не превышает
n1A = 1; n1B = 2; % расходы ингредиентов на краску 1
n2A = 2; n2B = 1; % расходы ингредиентов на краску 2
% Целевые функции
y1 = @(N) optPrice1*N(1) + optPrice2*N(2); % ? max
```

```
y2 = @(N) price1*N(1) + price2*N(2); % ? max
y3 = @(N) stockA - (N(1) + 2*N(2)); % ? min
% Функциональные ограничения
Aeq = []; beq = [];
% Параметрические ограничения для fmincon
lb = [0; 0]; % нижняя граница
ub = [Inf; Inf]; % верхняя граница
startPoint = lb;
f1 = @(N) 1/((3000*N(1) + 2000*N(2))/12667);
f2 = @(N) 1/((5500*N(1) + 3000*N(2))/22333);
f3 = @(N) (6 - (N(1) + 2*N(2)))/0.00001;
% Аддитивная свертка
f = @(N) 0.3*f1(N) + 0.4*f2(N) + 0.3*f3(N);
options = optimoptions('fmincon','Display','none'); % опции
оптимизатора
% Детерминированная задача
Ad = [n1A, n1B; n2A, n2B; 0, 1; -3000, -2000; -1, -2];
bd = [stockA;   stockB; 4; -11400; -5.99999];
[N3, fval3] = fmincon(f, startPoint, Ad, bd, Aeq, beq, lb, ub);
% Параметры для задачи стохастического программирования
alpha = 0.1:0.1:0.9;
Ka = icdf('Normal', alpha, 0, 1); % Квантили распределения N(0,1)
global K;
A = [n1A, n1B; n2A, n2B; -3000, -2000; -1, -2];
b = [stockA;   stockB;   -11400;     -5.99999];
% Решение стохастической задачи
N = zeros(2,9);
for i = 1:9
    K = Ka(i);
    [N(:,i), ~] = fmincon(f, startPoint, A, b, Aeq, beq, lb, ub,
@nonline, options);
    [c, ceq] = nonline(N(:,i));
end
function [ c, ceq ] = nonline( N )
    global K;
    c = N(2) - 4 + K * 0.6;
    ceq = [];
end
```

Результат работы скрипта:

Листинг 1.17 – Результаты расчетов.

```
alpha = 0.1, N1 = 3.3333, N2 = 1.3333, y1 = 12667, y2 = 22333, y3 = 0.0001
alpha = 0.2, N1 = 3.3333, N2 = 1.3333, y1 = 12667, y2 = 22333, y3 = 0.0001
alpha = 0.3, N1 = 3.3333, N2 = 1.3333, y1 = 12667, y2 = 22333, y3 = 0.0001
alpha = 0.4, N1 = 3.3333, N2 = 1.3333, y1 = 12667, y2 = 22333, y3 = 0.0001
alpha = 0.5, N1 = 3.3333, N2 = 1.3333, y1 = 12667, y2 = 22333, y3 = 0.0001
alpha = 0.6, N1 = 3.3333, N2 = 1.3333, y1 = 12667, y2 = 22333, y3 = 0.0001
alpha = 0.7, N1 = 3.3333, N2 = 1.3333, y1 = 12667, y2 = 22333, y3 = 0.0001
alpha = 0.8, N1 = 3.3333, N2 = 1.3333, y1 = 12667, y2 = 22333, y3 = 0.0001
alpha = 0.9, N1 = 3.3333, N2 = 1.3333, y1 = 12667, y2 = 22333, y3 = 0.0001
             N1 = 3.3333, N2 = 1.3333, y1 = 12667, y2 = 22333, y3 = 0.0001
```

Результат решения задачи (табл. 1.10):

Таблица 1.10 – Результаты решения задачи.

α	N1	N2	y1	y2	y3
0,1	3.3333	1.3333	12667	22333	0.0001
0,2	3.3333	1.3333	12667	22333	0.0001
0,3	3.3333	1.3333	12667	22333	0.0001
0,4	3.3333	1.3333	12667	22333	0.0001
0,5	3.3333	1.3333	12667	22333	0.0001
0,6	3.3333	1.3333	12667	22333	0.0001
0,7	3.3333	1.3333	12667	22333	0.0001
0,8	3.3333	1.3333	12667	22333	0.0001
0,9	3.3333	1.3333	12667	22333	0.0001
Детермин.	3.3333	1.3333	12667	22333	0.0001

Почему получился такой результат? У нас есть вероятностное ограничение:

$$N_2 - 4 + K_\alpha * 0{,}6 \le 0$$

При крайних значениях α = [0,1; 0,9] α-квантиль распределения принимает значения Kα = [-1,281; 1,281], ограничения в таком случае принимают вид:

N2(α=0,1) ≤ 4,77;   N2(α=0,9) ≤ 3,23

Таким образом, эти ограничения не являются активными для решения задачи (N1 = 3.3333; N2 = 1.3333) при доверительных вероятностях α = [0,1 - 0,9]. Иными словами, данное ограничение никогда не станет активным и не будет влиять на результат решения задачи.

2 Поиск оптимальной стратегии принятия решений с использованием марковских моделей

2.1 Постановка задачи

Решить задачу методом линейного программирования при бесконечном горизонте.

Ежедневно утром производится проверка дорогостоящей машины с целью выявления, находится ли она в исправном состоянии, требует мелкого ремонта или нуждается в серьезном ремонте. Обозначим эти состояния 0, 1, 2 соответственно. Если машина находится в совершенно исправном состоянии, то вероятность того, что она останется в таком же состоянии на начало следующего дня, равна p(0 | 0); вероятность того, что потребуется мелкий ремонт, равна p(1 | 0) и вероятность того, что возникает необходимость серьезного ремонта, равна p(2 | 0). В случае, когда машина требует ремонта, фирма может прибегнуть к услугам двух ремонтных фирм, одна из которых (фирма F, гарантирующая качество ремонта) взимает плату M за мелкий ремонт и плату R за крупный. Вторая (фирма T, не гарантирующая качества ремонта) взимает соответственно плату m и r, где m < M и r < R. Легко себе представить, что качество работ, производимых фирмой F, выше, чем у фирмы T, что отражается значением вероятности полностью исправного состояния машины на начало следующего за ремонтом дня. Пусть решение d = 1 определяет выбор фирмы F и решение d = 2 — выбор фирмы T. Обозначим через p(j\i, d) вероятность перехода машины в состояние j на следующем отрезке (j = 0, 1, 2) при условии, что она находится в состоянии i на текущем отрезке (i = 1, 2) и принимается решение d (d = 1, 2).

Примем, a = 1 и предположим, что

p (0 | 0) = 0,6, (машина осталась исправной)

p (1 | 0) = 0,3, (машина требует мелкого ремонта)

p (2 | 0) = 0,1, (машина требует крупного ремонта)

p (0 | 1,1) = 0,9, M = 14 (F выполнила мелкий ремонт)

p (1 | 1,1) = 0,1, (F выполняет мелкий ремонт, оплаты нет)

p (2 | 1,1) = 0, (невозможное событие – после ремонта поломка усугубилась)

p (0 | 1,2) = 0,7, m = 12, (T выполнила мелкий ремонт)

p (1 | 1,2) = 0,2, (T выполняет мелкий ремонт, оплаты нет)

p (2 | 1,2) = 0,1, (T выполняет мелкий ремонт, поломка усугубляется, оплаты нет)

p (0 | 2,1) = 0,6, R = 21 (F выполнила крупный ремонт)

p (1 | 2,1) = 0,3, R – M = 7 (F выполнила частично крупный ремонт, но машина еще требует мелкого ремонта)

p (2 | 2,1) = 0,1, (F выполняет крупный ремонт, оплаты нет)

p (0 | 2,2) = 0,5, r = 19 (T выполнила крупный ремонт)

p (1 | 2,2) = 0,4, r – m = 7 (T выполнила частично крупный ремонт, но машина еще требует мелкого ремонта)

p (2 | 2,2) = 0,1, (T выполняет крупный ремонт, оплаты нет).

2.2 Ход работы

Имеется три состояния машины:

1 – исправна;

2 – нужен мелкий ремонт;

3 – нужен крупный ремонт;

и три решения:

X1 – ничего не делать;

X2 – выбрать качественную фирму F;

X3 – выбрать дешёвую фирму T.

Основные матрицы переходных вероятностей (P_1, P_2, P_3) и затрат (R_1, R_2, R_3) имеют вид:

$$P_1 = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ - & - & - \\ - & - & - \end{pmatrix} \quad P_2 = \begin{pmatrix} - & - & - \\ 0.9 & 0.1 & 0 \\ 0.6 & 0.3 & 0.1 \end{pmatrix} \quad P_3 = \begin{pmatrix} - & - & - \\ 0.7 & 0.2 & 0.1 \\ 0.5 & 0.4 & 0.1 \end{pmatrix}$$

$$R_1 = \begin{pmatrix} 0 & 0 & 0 \\ - & - & - \\ - & - & - \end{pmatrix} \quad R_2 = \begin{pmatrix} - & - & - \\ 14 & 0 & 0 \\ 21 & 7 & 0 \end{pmatrix} \quad R_3 = \begin{pmatrix} - & - & - \\ 12 & 0 & 0 \\ 19 & 7 & 0 \end{pmatrix}$$

Все стационарные стратегии приведены в таблице ниже:

Таблица 2.1 – Стационарные стратегии.

Состояние машины	**Исправна**	**Легкая поломка**	**Серьёзная поломка**
№ стратегии	Используемая фирма		
1	-	F	T
2	-	F	F
3	-	T	T
4	-	T	F

2.3 Метод линейного программирования

Пусть q_j^i – условная вероятность того, что будет принято допустимое решение X_i

Для любых $j = \overline{1, m}$ и $i = 1, M$ полагаем

$$\omega_{ji} = \Pi_j q_j^i,$$

где ω_{ji} — вероятность пребывания системы S в состоянии S_j при принятии решения $X_i \in G$. При этом для любого $j = \overline{1, m}$

$$\sum_{i=1}^{M} \omega_{ji} = \sum_{i=1}^{M} \Pi_j q_j^i = \Pi_j \sum_{i=1}^{M} q_j^i = \Pi_j.$$

Таким образом,

$$q_j^i = \frac{\omega_{ji}}{\sum\limits_{i=1}^{M} \omega_{ji}},$$

и ограничение

$$\sum_{j=1}^{m} \Pi_j = 1$$

эквивалентно ограничению

$$\sum_{j=1}^{m} \sum_{i=1}^{M} \omega_{ji} = 1.$$

Скалярная функция e может быть представлена в виде

$$e = \sum_{j=1}^{m} \sum_{i=1}^{M} \nu_j^i \omega_{ji},$$

а исходная задача может быть сформулирована в виде задачи линейного программирования с переменными ω_{ji}:

$$\begin{cases} \sum\limits_{j=1}^{m} \sum\limits_{i=1}^{M} \nu_j(X_i) \omega_{ji} \to \max; \\ \sum\limits_{i=1}^{M} \omega_{ji} - \sum\limits_{k=1}^{m} \sum\limits_{i=1}^{M} p_{kj}(X_i) \omega_{ki} = 0, \quad j = \overline{1, m}, \\ \sum\limits_{j=1}^{m} \sum\limits_{i=1}^{M} \omega_{ji} = 1, \\ \omega_{ji} \geqslant 0, j = \overline{1, m}, \ i = \overline{1, M}. \end{cases}$$

Оптимальное решение гарантирует выполнение равенства $q_j^i = 1$ для фиксированного i при любом j.

$$v_i^k = \sum_{j=1}^{m} p_{ij}^k r_{ij}^k,$$

Результаты вычисления v_i^k (табл. 2.2):

Таблица 2.2 – Результаты вычисления v_i^k.

k	v_1	v_2	v_3
1	0	-	-
2	-	12,6	14,7
3	-	8,4	12,3

Сформулируем решаемую задачу в виде задачи линейного программирования:

$$\begin{cases} 0 * w_{11} + 12.6 * w_{22} + 8.4 * w_{23} + 14.7 * w_{32} + 12.3 w_{33} \to min \\ (1 - 0.6) w_{11} - 0.9 w_{22} - 0.7 w_{23} - 0.6 w_{32} - 0.5 w_{33} = 0 \\ -0.3 w_{11} + (1 - 0.1) w_{22} + (1 - 0.2) w_{23} - 0.3 w_{32} - 0.4 w_{33} = 0 \\ -0.1 w_{11} - 0 * w_{22} - 0.1 w_{23} + (1 - 0.1) w_{32} + (1 - 0.1) w_{33} = 0 \\ w_{11} + w_{22} + w_{23} + w_{32} + w_{33} = 1, \ \ w_{ji} \geq 0, \quad j = 1,2,3, \qquad i = 1,2,3 \end{cases}$$

Для решения этой задачи напишем скрипт в Матлабе:

Листинг 2.1 – Код на Matlab.

```
%Начальные условия
x0=[0;0;0;0;0];
A=[-1 0  0 0 0;
    0 -1 0 0 0;
    0 0 -1 0 0;
    0 0 0 -1 0;
    0 0 0 0 -1];
b=[0;0;0;0;0];
beq=[0;0;0;1];
%Задача линейного программирования в виде матрицы
Aeq=[0.4 -0.9 -0.7 -0.6 -0.5;
     -0.3 0.9 0.8 -0.3 -0.4;
     -0.1 -0 -0.1 0.9 0.9;
     1 1 1 1 1];
%Максимизирование функции
[x, fval] = fmincon(inline('0*x(1)+12.6*x(2)+8.4*x(3)+14.7*x(4)+12.3*x(5)'), x0,
A, b, Aeq, beq)
%Расчитываем вероятности
q(1,1) = x(1)/(x(1));
q(2,2) = x(2)/(x(2)+x(3));
q(3,2) = x(3)/(x(2)+x(3));
q(2,3) = x(4)/(x(4)+x(5));
q(3,3) = x(5)/(x(4)+x(5));
```

Получены следующие результаты:

Таблица 2.3 – Результаты расчетов.

k	q_1	q_2	q_3
1	1	0	0

2	0	0	0
3	0	1	1

Получается, что при бесконечном горизонте лучше всего всегда чинить машину у фирмы T, которая выполняет свою работу дешевле: $q_2^3 = q_3^3 = 1$. Если машина не сломана, существует единственное решение – ничего не делать: $q_1^1 = 1$.

Оптимальной является стратегия τ* = (X_1 X_3 X_3).

3 Поиск оптимальных параметров сети систем массового обслуживания

3.1 Постановка задачи

1. Задание по оптимизации многоканальной сети Джексона (алгоритм 1)

Задача 1

$$\min \sum_{j=1}^{n} m_j$$

$$L(m) = \sum_{j=1}^{n} L_j \le L_r$$

Дано:

$$\left\{ n - \textit{число узлов}, \quad Q = \left\{ q_{ij} \right\}_{\substack{i=\overline{0,n} \\ j=\overline{0,n}}}, \vec{\mu} = (\mu_1, \mu_2, \ . \ . \ . \ , \ \mu_n) \right\}$$

В предположении наличия установившегося режима в ССМО подобрать λ_0 так, чтобы $\min\limits_{(j)} \left\{ \overline{n_{обсл\, j}} = \dfrac{\lambda_j}{\mu_j} \right\} \ge 2$.

Граничное значение L_r числа заявок в сети также следует подобрать самостоятельно.

№ вар	$Q = \left\{ q_{ij} \right\}_{\substack{i=\overline{0,n} \\ j=\overline{0,n}}}$	$\vec{\mu}$

Задача 1											
115	0	0,1	0,3	0,2	0,4			2	2	3	3
	0,5	0	0,1	0,1	0,3						
	0,3	0	0	0,5	0,2						
	0,2	0,1	0,7	0	0						
	0,4	0,3	0,2	0,1	0						

3.2 Ход работы

Подберём λ_0. Необходимо сначала найти частоты $\{\alpha_i\}$, потом найти $\min\left\{\frac{\alpha_i}{\mu_i}\right\} = \frac{\alpha_s}{\mu_s}$ и далее выбрать $\lambda_0 \le \frac{2\mu_s}{\alpha_s}$.

Частоты $\{\alpha_i\}$ находятся из системы уравнений:

$$\begin{cases} \alpha_j = \sum_{i=0}^{n} \alpha_i q_{ij}, \qquad j = \overline{0,n}, \qquad n = 4 \\ \alpha_0 = 1 \end{cases}$$

или

$$\begin{cases} \alpha_0 = 1 \\ \alpha_1 = \alpha_0 * 0{,}1 + \alpha_1 * 0 + \alpha_2 * 0 + \alpha_3 * 0{,}1 + \alpha_4 * 0{,}3 \\ \alpha_2 = \alpha_0 * 0{,}3 + \alpha_1 * 0{,}1 + \alpha_2 * 0 + \alpha_3 * 0{,}7 + \alpha_4 * 0{,}2 \\ \alpha_3 = \alpha_0 * 0{,}2 + \alpha_1 * 0{,}1 + \alpha_2 * 0{,}5 + \alpha_3 * 0 + \alpha_4 * 0{,}1 \\ \alpha_4 = \alpha_0 * 0{,}4 + \alpha_1 * 0{,}3 + \alpha_2 * 0{,}2 + \alpha_3 * 0 + \alpha_4 * 0 \end{cases}$$

Для решения данной системы уравнений напишем скрипт на Matlab:

Листинг 3.1 – Код на Matlab.

```
A= [1    0   0  0  0;
    0.1 -1   0 0.1 0.3;
    0.3 0.1 -1 0.7 0.2;
    0.2 0.1 0.5 -1 0.1;
    0.4 0.3 0.2 0 -1  ];
b= [1;  0;  0;  0;  0];
alpha_j=A\b
```

Листинг 3.2 – Результат.

```
alpha_j =
    1.0000
    0.4083
    1.0914
    0.8606
```

```
    0.7408
```

Далее найдём $\min\left\{\frac{\alpha_i}{\mu_i}\right\} = \frac{\alpha_s}{\mu_s}$:

Листинг 3.3 – Код на Matlab.

```
mu_j= [2; 2; 3; 3];
alpha_j_from_1=alpha_j(2:5);
c= alpha_j_from_1 ./ mu_j;
min(c)
```

Листинг 3.2 – Результат.

```
c =
    0.2041
    0.5457
    0.2869
    0.2469

ans =
    0.2041
```

Самый ненагруженный узел – j=1. Выберем значение $\lambda_0 \leq \frac{2\mu_s}{\alpha_s}$:

Листинг 3.3 – Код на Matlab.

```
lambda_0_max= 2*2/0.4083
```

Листинг 3.4 – Результат.

```
lambda_0_max =
    9.7967
```

В таком случае примем λ_0 = 9.

Граничное значение L_r числа заявок в сети также следует подобрать самостоятельно.

Минимальное значение числа заявок в сети (при сколь угодно большом, даже бесконечном числе каналов в узле) равно $\sum_{j=1}^{n}\frac{\lambda_j}{\mu_j}$. Таким образом, нижняя граница L_r определена: $L_{r\,\min} = \sum_{j=1}^{n}\frac{\lambda_j}{\mu_j}$.

Для начала, найдем интенсивности входного потока λ_j . Из формулы

$$\alpha_j = \frac{\lambda_j}{\lambda_0}, \quad j = \overline{1,\ M}$$

выразим интенсивность входного потока:

$$\lambda_j = \alpha_j * \lambda_0$$

Высчитаем это в матлабе:

Листинг 3.5 – Код на Matlab.

```
lambda_0= 9
lambda_j_from_1= alpha_j_from_1 * lambda_0
```

Листинг 3.6 – Результат.

```
lambda_j_from_1 =
    3.6746
    9.8227
    7.7455
    6.6669
```

Нижнюю границу L_r посчитаем по формуле: $L_{r\,\min} = \sum_{j=1}^{n} \frac{\lambda_j}{\mu_j}$.

Листинг 3.7 – Код на Matlab.

```
Lr_min= sum(lambda_j_from_1 ./ mu_j)
```

Листинг 3.8 – Результат.

```
Lr_min =
   11.5528
```

Верхняя граница $L_{r\,\max}$ также может быть рассчитана, если выбрать начальное распределение числа каналов по узлам в соответствии простейшим правилом: $m_j = \left[\frac{\lambda_0 \alpha_j}{\mu_j}\right] + 1, \quad j = \overline{1,n}$, обеспечивающим установившийся режим в ССМО, где $[\cdot]$ — целая часть числа.

Листинг 3.9 – Расчет m_j^0.

```
for j=1:4
    m_j_0(j)= fix(lambda_j_from_1(j) / mu_j(j)) + 1;
end
```

```
m_j_0
```

Листинг 3.10 – Результат.

```
m_j_0 =
     2     5     3     3
```

Для расчета верхней границы воспользуемся следующими формулами:

$$L_j(m_j, \mu_j, \lambda_j) = \frac{\dfrac{\lambda_j}{m_j\mu_j}\left(\dfrac{\lambda_j}{\mu_j}\right)^{m_j} P_0}{m_j!\left(1 - \dfrac{\lambda_j}{m_j\mu_j}\right)^2} + \frac{\lambda_j}{\mu_j},$$

где

$$P_0 = \left[1 + \sum_{i=1}^{m_j} \frac{\left(\dfrac{\lambda_j}{\mu_j}\right)^i}{i!} + \frac{\left(\dfrac{\lambda_j}{\mu_j}\right)^{m_j+1}}{m_j!\left(m_j - \dfrac{\lambda_j}{m_j\mu_j}\right)}\right]^{-1};$$

Реализуем эти формулы как функции в Матлабе:

Листинг 3.11 – Код на Matlab.

```
function [ L_j ] = Lj( m_j, mu_j, lambda_j )
    d1= (lambda_j/(m_j*mu_j)) * ((lambda_j/mu_j)^m_j) * P0(m_j,mu_j,lambda_j);
    d2= factorial(m_j) * ((1 - lambda_j/(m_j*mu_j))^2);
    s2= lambda_j / mu_j;

    L_j=d1/d2 + s2;
end

function [ P_0 ] = P0( m_j, mu_j, lambda_j )
    sum= 0;

    for i=1:m_j
        s1= ((lambda_j / mu_j)^i) / factorial(i);
        d1= (lambda_j / mu_j)^(m_j+1);
        d2= factorial(m_j) * (m_j - lambda_j/(m_j*mu_j));
        s2= d1 / d2;
        s = s1 + s2;

        sum= sum + s;
    end
```

```
    P_0= (1 + sum)^(-1);
end
```

Высчитаем верхнюю границу:

Листинг 3.12 – Код на Matlab.

```
for j=1:4
    L_j(j)= Lj(m_j_0(j), mu_j(j), lambda_j_from_1(j));
end
L_j
sum(L_j)
```

Листинг 3.13 – Результат.

```
L_j =
   24.6760  326.6030    8.8817    3.7831

ans =
  363.9438
```

Необходимо выбрать $L_{r\,\min} < L_r < L_{r\,\max}$, пусть $L_r = 15$.

Теперь для каждого узла формируются два показателя его реакции на единичное увеличение числа каналов: приращение стоимости узла и уменьшение числа заявок в узле.

$$\Delta F_j(m_j+1) = F_j(m_j+1) - F_j(m_j) \geq 0\,;$$

$$\Delta L_j(m_j+1) = L_j(m_j+1) - L_j(m_j) < 0\,.$$

Напишем соответствующие функции на Матлабе:

Листинг 3.14 – Код на Matlab.

```
function [ deltaL_j ] = deltaLj( m_j, mu_j, lambda_j )
    deltaL_j= Lj(m_j, mu_j, lambda_j) - Lj(m_j - 1, mu_j, lambda_j);
end

function [ F_j ] = Fj( m_j )
    F_j= m_j;
end

function [ deltaF_j ] = deltaFj( m_j )
    deltaF_j= Fj(m_j) - Fj(m_j - 1);
end
```

Для каждого j-го узла рассчитывается индекс приоритета на добавление канала по следующей формуле:

$$PI_j(m_j) = \frac{\Delta F_j(m_j+1)}{-v_j \Delta L_j(m_j+1)} .$$

За каждую находящуюся в сети заявку комплексу приходится платить штраф. Если последний равен 1 (v=1), то величина суммарного штрафа равна среднему числу заявок в сети.

Напишем соответствующую функцию:

Листинг 3.15 – Код на Matlab.

```
function [ PI_j ] = PIj( m_j, mu_j, lambda_j, v_j )
    PI_j= deltaFj(m_j + 1) / (-v_j * deltaLj(m_j + 1, mu_j, lambda_j));
end
```

Далее проведём оптимизацию сети по следующему алгоритму:

Шаг 1. Найти $\{m_j^0\},\ j=\overline{1,M}$, идти к шагу 2.

Шаг 2. Если $L(\{m_j^0\}) \le L_T$, вектор $\{m_j^0\},\ j=\overline{1,M}$, есть оптимальное решение.

Идти к шагу 9, иначе идти к шагу 3.

Шаг 3. Найти $PI(m)=\{PI_j(m_j)\},\ j=\overline{1,M}$, идти к шагу 4.

Шаг 4. Найти $PI_{j^*} = \min\limits_{(j)} PI_j(m_j)$, идти к шагу 5.

Шаг 5. $m_{j^*} = m_{j^*} + 1$, идти к шагу 6.

Шаг 6. Пересчитать $L(j^*), F_j(m_{j^*})$, $PI_{j^*}(m_{j^*})$, идти к шагу 7.

Шаг 7. Коррекция $m, L(m), F(m), PI(m)$, идти к шагу 8.

Шаг 8. Если $L(m) \le L_T$, то вектор $m=\{m_j\},\ j=\overline{1,M}$, есть оптимальное решение.

Идти к шагу 9, иначе идти к шагу 4.

Шаг 9. ОСТАНОВ.

Напишем скрипт, реализующий данный алгоритм:

Листинг 3.16 – Код на Matlab.

```
L_r= 15;
v_j= 1; % штраф равен 1
m_j= m_j_0; % начинаем с минимального количества каналов на узлах
L_j= L_j_0; % начинаем с L_j при m_j_0

% считаем стартовые значения PI_j
for j=1:4
    PI_j(j)= PIj(m_j(j), mu_j(j), lambda_j_from_1(j), v_j);
end

while 1
    % находим индекс минимального элемента PI_j
    j = find( PI_j == min(PI_j) )

    % у узла с минимальным приоритетом увеличиваем на 1 кол-во каналов
    m_j(j)= m_j(j) + 1;

    % пересчитываем PI_j(j)
    PI_j(j)= PIj(m_j(j), mu_j(j), lambda_j_from_1(j), v_j);

    % пересчитываем L_j(j)
    L_j(j)= Lj(m_j(j), mu_j(j), lambda_j_from_1(j));


    if (sum(L_j) <= L_r)
        break
    end
end
```

Листинг 3.17 – Результат.

```
m_j_0 =
     2     5     3     3


j =
     2
j =
     1
j =
     3
j =
     2


PI_j =
    2.3574    2.8115    2.4338    0.7665
```

```
m_j =
     3     7     4     3

L_j =
    2.3678    5.4681    3.1303    3.7831
```

Оптимальное решение было получено за 5 итераций. Представим полученные результаты в виде таблиц (табл. 3.1, табл. 3.2, табл. 3.3).

Таблица 3.1 – Изменение кол-ва каналов.

№	m1	m2	m3	m4
1	2	5	3	3
2	2	6	3	3
3	3	6	3	3
4	3	6	4	3
5	3	7	4	3

Таблица 3.2 – Изменение приоритета на добавление канала.

№	PI1	PI2	PI3	PI4
1	0,045	0,003	0,174	0,766
2	0,045	0,593	0,174	0,766
3	2,357	0,593	0,174	0,766
4	2,357	0,593	2,434	0,766
5	2,357	2,811	2,434	0,766

Таблица 3.3 – Изменение среднего числа заявок на узле.

№	L1	L2	L3	L4	L
1	24,68	326,60	8,88	3,78	363,94
2	24,68	7,15	8,88	3,78	44,49
3	2,37	7,15	8,88	3,78	22,19
4	2,37	7,15	3,13	3,78	16,44
5	2,37	5,47	3,13	3,78	14,75

В данной работе была произведена оптимизация ССМО путём добавления дополнительных каналов в узлы сети для уменьшения очереди.

Начальные значения:

$$m = [2; 5; 3; 3]$$

$$L_j = [24{,}68; 326{,}6; 8{,}88; 3{,}78]$$

$$L = 363{,}94$$

После оптимизации:

$$m = [3; 7; 4; 3]$$

$$L_j = [2{,}37; 5{,}47; 3{,}13; 3{,}78]$$

$$L = 14{,}75$$

Оптимизация путём добавления дополнительных каналов в узлы показала значительное уменьшение очередей в сети.

4 Решение задачи анализа потокового графа с использованием методики GERT и алгебры потоковых графов

4.1 Постановка задачи

Дано:

1. Граф GERT-сети:

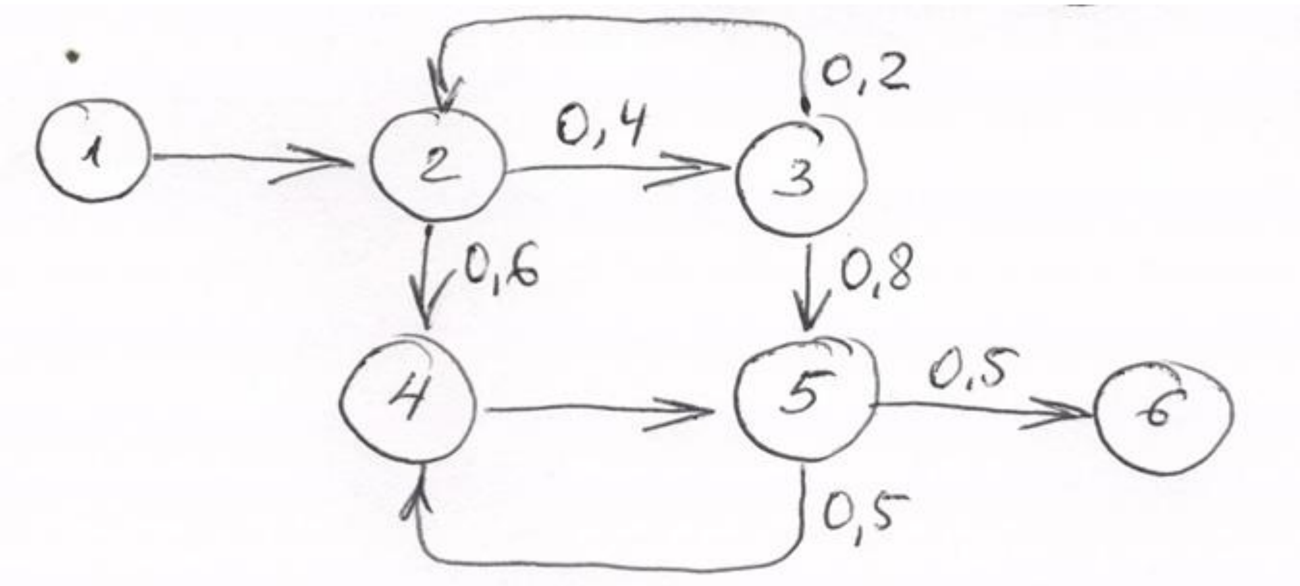

Рисунок 4.1 – Граф GERT-сети.

2. Каждой дуге-работе поставлены в соответствие следующие данные:

- закон распределения времени выполнения работы (будем считать его нормальным);
- параметры закона распределения (математическое ожидание и дисперсия);
- вероятность выполнения работы, показанная на графе.

Часть 1

Используя методику GERT, изложенную в книге «Методы анализа сетей», найти:

1. Вероятность выхода в завершающий узел графа (для всех вариантов узел 6);

2. Производящую функцию длительности процесса от начального узла до завершающего узла;

3. Математическое ожидание длительности процесса от начального узла до завершающего узла;

4. Дисперсию ожидание длительности процесса от начального узла до завершающего узла.

В отчете перечислить все петли всех порядков, обнаруженные на графе, выписать уравнение Мейсона, получить решение для $\boldsymbol{W_E(s)}$ и найти требуемые параметры. Примерно так, как это сделано в примере на стр. 403-409 книги Филипса и Гарсиа «Методы анализа сетей».

Часть 2

Повторить пункты задания 2, 3, 4 используя методику анализа потокового графа, основанную на обработке матрицы передач (Branch Transmittance Matrix).

Для выполнения задания рекомендуется пользоваться следующими источниками:

1. Филипс и Гарсиа «Методы анализа сетей»;

2. Презентация GERT_&_Flowgraph_Algebra.pdf;

3. Ren_The Methodology of Flowgraph.pdf.

4.2 Ход работы

Замкнём сеть дугой (из узла 6 в узел 1 (рис. 4.2)), чтобы определить эквивалентную W-функцию для анализируемой GERT-сети.

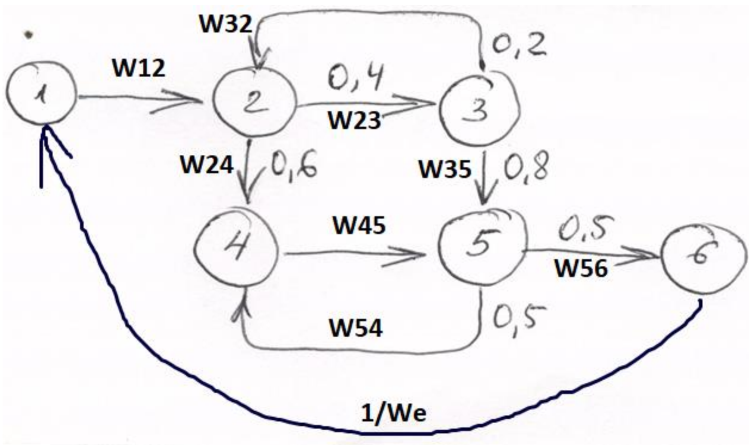

Рисунок 4.2 – Замкнутая GERT-цепь.

Опишем данные сети в таблице 4.1.

Таблица 4.1 – Данные сети.

Ветвь	Вес ребра	Мат. Ожидание	Дисперсия	W-функция
(1,2)	1	44	25	$\exp(44s + 12{,}5s^2)$
(2,3)	0,4	24	25	$0{,}4 * \exp(24s + 12{,}5s^2)$
(2,4)	0,6	40	25	$0{,}6 * \exp(40s + 12{,}5s^2)$
(3,2)	0,2	34	25	$0{,}2 * \exp(34s + 12{,}5s^2)$
(3,5)	0,8	15	9	$0{,}8 * \exp(15s + 4{,}5s^2)$
(4,5)	1	32	25	$\exp(32s + 12{,}5s^2)$
(5,4)	0,5	11	4	$0{,}5 * \exp(11s + 2s^2)$
(5,6)	0,5	24	25	$0{,}5 * \exp(24s + 12{,}5s^2)$

Петли первого порядка:

- $W_{23}W_{32}$
- $W_{45}W_{54}$
- $W_{12}W_{23}W_{35}W_{56}(1/W_E)$
- $W_{12}W_{24}W_{45}W_{56}(1/W_E)$

Петли второго порядка:

- $W_{23}W_{32}$ и $W_{45}W_{54}$

Часть 1

Напишем уравнение Мейсона:

$$H = 1 - W_{23}W_{32}$$
$$-W_{45}W_{54}$$
$$-W_{12}W_{23}W_{35}W_{56}\frac{1}{W_E}$$
$$-W_{12}W_{24}W_{45}W_{56}\frac{1}{W_E}$$
$$+W_{23}W_{32}W_{45}W_{54} = 0$$

Выразим W_E (s):

$$1 - W_{23}W_{32} - W_{45}W_{54} + W_{23}W_{32}W_{45}W_{54}$$
$$= (W_{12}W_{23}W_{35}W_{56} + W_{12}W_{24}W_{45}W_{56}) * \frac{1}{W_E}$$

$$W_E(s) = \frac{W_{12}W_{23}W_{35}W_{56} + W_{12}W_{24}W_{45}W_{56}}{1 - W_{23}W_{32} - W_{45}W_{54} + W_{23}W_{32}W_{45}W_{54}}$$

Вычислим математическое ожидание и дисперсию: $M_E(s) = 1$ при $s = 0$

Поскольку $W_E(s) = p_E M_E(s)$, то $p_E = W_E(0)$, откуда следует, что

$$M_E(s) = \frac{W_E(s)}{p_E} = \frac{W_E(s)}{W_E(0)}$$

Вычисляя первую и вторую производные по s функции $M_E(s)$, и, полагая s=0, находим математическое ожидание:

$$\mu_{1E} = \frac{\partial M_E(s)}{\partial s} | s = 0$$

и дисперсию: $\sigma^2 = \mu_{2E} - [\mu_{1E}]^2$.

Вероятность выхода в завершающий узел графа: $p_E = W_E(0)$.

Напишем скрипт в среде Matlab, позволяющий вычислить $W_E(0)$, математическое ожидание и дисперсию.

Листинг 4.1 – Скрипт Matlab.

```
M12=44;
M23=24;
M24=40;
M32=34;
M35=15;
M45=32;
M54=11;
M56=24;

D12=25;
```

```
D23=25;
D24=25;
D32=25;
D35=9;
D45=25;
D54=4;
D56=25;

syms s;

W12=     exp(M12*s+D12/2*s^2);
W23= 0.4*exp(M23*s+D23/2*s^2);
W24= 0.6*exp(M24*s+D24/2*s^2);
W32= 0.2*exp(M32*s+D32/2*s^2);
W35= 0.8*exp(M35*s+D35/2*s^2);
W45=     exp(M45*s+D45/2*s^2);
W54= 0.5*exp(M54*s+D54/2*s^2);
W56= 0.5*exp(M56*s+D56/2*s^2);

We = (W12*W23*W35*W56 + W12*W24*W45*W56) / (1 - W23*W32 - W45*W54 + W23*W32*W45*W54);

We = simplify(We)
We0 = subs(We, 's', 0)  % We(0)

Me = We/We0;

m1 = diff(Me, 's');
m1 = subs(m1, 's', 0)

m2 = diff(Me, 's',2);
m2=subs(m2, 's', 0)

D = m2 - (m1)^2
```

Результат:

```
We =
-(15*exp(10*s*(5*s + 14)) + 8*exp(s*(42*s + 107)))/(4*exp(s*(25*s + 58)) + 25*exp((s*(29*s + 86))/2) - 2*exp((s*(79*s + 202))/2) - 50)

We0 =
1

m1 =
4061/23

m2 =
18814440/529

D =
2322719/529
```

Был получен следующий результат (табл. 4.2):

Таблица 4.2 – Результаты.

Вероятность выхода в завершающий узел графа	Математическое ожидание	Дисперсия времени выхода процесса в завершающий узел графа
$p = W_E$	μ_{1E}	D_E
1	176,57	4390,77

4.3 Метод анализа потокового графа, оснсованный на обработке матрицы передач

Определяем матрицу Q:

$$Q =$$

0	q12	0	0	0	0
0	0	q23	q24	0	0
0	q32	0	0	q35	0
0	0	0	0	q45	0
0	0	0	q54	0	q56
w61	0	0	0	0	0

Определяем матрицу коэффициентов $A = I_6 - Q^T$:

$$A =$$

1	0	0	0	0	-w61
-q12	1	-q32	0	0	0
0	-q23	1	0	0	0
0	-q24	0	1	-q54	0
0	0	-q35	-q45	1	0
0	0	0	0	-q56	1

Далее находим $det(A)$; $\frac{\partial det(A)}{\partial w_{61}}$; $det(A|w_{61} = 0)$.

$W_E(s)$ высчитывается по формуле:

$$W_E(s) = -\frac{\frac{\partial det(A)}{\partial w_{61}}}{det(A|w_{61} = 0)}$$

Напишем скрипт в среде Matlab.

Листинг 4.2 – Скрипт Matlab.

```
clear all;
```

```
M12=44;
M23=24;
M24=40;
M32=34;
M35=15;
M45=32;
M54=11;
M56=24;

D12=25;
D23=25;
D24=25;
D32=25;
D35=9;
D45=25;
D54=4;
D56=25;

syms q12;
syms q23;
syms q24;
syms q32;
syms q35;
syms q45;
syms q54;
syms q56;
syms w61;
syms s;

Q= [0   q12 0   0   0   0;
    0   0   q23 q24 0   0;
    0   q32 0   0   q35 0;
    0   0   0   0   q45 0;
    0   0   0   q54 0   q56;
    w61 0   0   0   0   0];

A= eye(size(Q,1)) - transpose(Q)

det_A= det(A);
det_dw= diff(det_A,w61);
det2_A= subs(det_A,w61,0);
We= -det_dw/det2_A

We= subs(We,q12,     exp(M12*s+D12/2*s^2));
We= subs(We,q23, 0.4*exp(M23*s+D23/2*s^2));
We= subs(We,q24, 0.6*exp(M24*s+D24/2*s^2));
We= subs(We,q32, 0.2*exp(M32*s+D32/2*s^2));
We= subs(We,q35, 0.8*exp(M35*s+D35/2*s^2));
We= subs(We,q45,     exp(M45*s+D45/2*s^2));
We= subs(We,q54, 0.5*exp(M54*s+D54/2*s^2));
We= subs(We,q56, 0.5*exp(M56*s+D56/2*s^2));
We= simplify(We)
We0= subs(We, 's', 0);
```

```
Me= We/We0;
m1= diff(Me, 's');
m1= subs(m1, 's', 0)
m2= diff(Me, 's',2);
m2=subs(m2, 's', 0)
D= m2 - (m1)^2
```

Результат:

```
A =
[    1,    0,    0,    0,    0, -w61]
[ -q12,    1, -q32,    0,    0,    0]
[    0, -q23,    1,    0,    0,    0]
[    0, -q24,    0,    1, -q54,    0]
[    0,    0, -q35, -q45,    1,    0]
[    0,    0,    0,    0, -q56,    1]

We =
-(q12*q23*q35*q56 + q12*q24*q45*q56)/(q23*q32 + q45*q54 - q23*q32*q45*q54 - 1)

We0 =
1

m1 =
4061/23

m2 =
18814440/529

D =
2322719/529
```

Был получен следующий результат (табл. 4.3):

Таблица 4.3 – Результаты.

Вероятность выхода в завершающий узел графа	Математическое ожидание	Дисперсия времени выхода процесса в завершающий узел графа
$p = W_E$	μ_{1E}	D_E
1	176,57	4390,77

Данный результат полностью совпадает с результатом, полученным в первой части. Это говорит о правильности обоих методов.

Заключение

В результате выполнения данной курсовой работы были получены навыки решения задач оптимизации в различных ситуациях при заданных ограничениях. Часто в математической модели требуется найти наибольшее или наименьшее значение некоторой функции на некотором множестве, то есть решить задачу оптимизации. Методов решения задач оптимизации достаточно много. В данной курсовой работе рассмотрены основные методы решения задач оптимизации и принятия решений.

Решение задач многокритериальной оптимизации, поиска оптимальной стратегии принятия решений с использованием марковских моделей и поиска оптимальных параметров сети систем массового обслуживания позволяет получить информацию о наилучших (оптимальных) значениях параметров системы и тех решениях, которые необходимо принять, чтобы достигнуть этих значений. Задача анализа потокового графа с использованием методики GERT и алгебры потоковых графов также может использоваться для оптимизации процессов.

Развитие таких разделов математики как математическое программирование, теория игр, теория массового обслуживания способствует постоянному усовершенствованию способов применения различных методов поиска оптимальных значений параметров систем и наилучших решений. В данной курсовой работе рассмотрены способы применения задач оптимизации в жизни на реальных примерах, это позволяет сказать, что область использования методов оптимизации невероятна велика.

Список использованных источников

[1] Таха Х. А. Введение в исследование операций: Пер. с англ. 7-е изд. М.: Издательский дом «Вильямс», 2005.

[2] Горбунов В.М. Теория принятия решений: учебное пособие. Томск: Изд-во Томск. политех. ун-та, 2010.

[3] Сиднев А. Г. Системный анализ. Часть 1 [Электронный ресурс] // Интранет портал ИИТУ СПбПУ. 2018. URL: http://intranet.ftk.spbstu.ru/docinfo.php? InfoFtkDocumentID=1386947 (дата обращения: 02.03.2018)