Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

КУРСОВАЯ РАБОТА

Дисциплина: **Методы оптимизации**

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

Выполнила студентка гр.13541/3 Д.Н. Мельникова
Руководитель, к.т.н., доц. А.Г. Сиднев
28 мая 2018 г.

Санкт-Петербург
2018

Оглавление

Введение

В данной работе рассматриваются следующие задачи:

1. Формализация многокритериальной оптимизационной задачи, методы сведения к однокритериальной, решение с использованием Optimization Toolbox системы среды Matlab
2. Поиск оптимальной стратегии принятия решений с использованием марковских моделей
3. Решение задачи анализа потокового графа с использованием методики GERT и алгебры потоковых графов
4. Поиск оптимальных параметров сети массового обслуживания

Глава 1. Формализация многокритериальной задачи, методы сведения к однокритериальной, решение с использованием Optimization Toolbox среды Matlab

1.1 Постановка задачи

Задача 4.

Автозавод производит 2 модели автомобилей I и II. Цена модели I – 10000$; цена модели II – 15000$. Стоимость материалов, необходимых для производства 7000$ и 11000$ соответственно.

Каждый автомобиль собирается роботами и работниками завода. Для сборки автомобиля модели I необходимо 5 часов работы робота и 1 час работы сборщиков. Для производства автомобиля модели II необходимо 4 часа работы робота и 2 часа работы сборщиков. Известно, что робот не может работать более 120 часов в неделю, а сборщики – более 48 часов. Один час эксплуатации робота обходится в 50$, а час работы сборщиков 100$.

Кроме того известно, что заводу еще не удавалось продать за неделю более 30 автомобилей модели I и более 15 модели II. В то же время спрос не опускался ниже 20 и 11 автомобилей моделей I и II соответственно.

Критерии.

Какое количество автомобилей каждой модели нужно производить, чтобы в условиях обеспечения выпуска автомобилей в пределах упомянутого выше спроса:

1) выручка была максимальной?

2) стоимость используемых для производства автомобилей материалов была минимальной?

3) затраты на производство (использование робота и зарплата сборщиков) были минимальны?

4) время производства было минимальным?

Пункты расчетного задания:

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

- Выделение главного критерия
- Свертка критериев (аддитивная и мультипликативная)
- Максимин или минимакс (максиминная свертка)
- Метод последовательных уступок
- fgoalattain
- Введение метрики в пространстве критериев

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

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

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

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

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

1.2 Построение математической модели

Расчетное время – 1 неделя (в условии все данные относятся к одной неделе). Введем независимые переменные:

x_1 – число автомобилей модели I;

x_2 – число автомобилей модели II.

Формируем критерии:

1) выручка ($) $f_1 = 10000 \cdot x_1 + 15000 \cdot x_2$;

2) стоимость материалов ($) $f_2 = 7000 \cdot x_1 + 11000 \cdot x_2$;

3) затраты на производство ($)

$f_3 = 50 \cdot (5 \cdot x_1 + 4 \cdot x_2) + 100 \cdot (1 \cdot x_1 + 2 \cdot x_2) = 350 \cdot x_1 + 400 \cdot x_2$;

4) время производства (ч)

$f_4 = (5 \cdot x_1 + 4 \cdot x_2) + (1 \cdot x_1 + 2 \cdot x_2) = 6 \cdot x_1 + 8 \cdot x_2$.

Формируем ограничения:

по числу автомобилей – здесь возможно формулирование верхних ограничений как по максимальным продажам за неделю, так и по гарантированному спросу

$x_1 \geq 20$ (спрос); $x_1 \leq 30$ (продажи);

$x_2 \geq 11$ (спрос); $x_2 \leq 15$ (продажи);

по времени работы робота (ч)

$5 \cdot x_1 + 4 \cdot x_2 \leq 120$;

по времени работы сборщиков (ч)

$1 \cdot x_1 + 2 \cdot x_2 \leq 48$.

Все переменные и критерии неотрицательны.

Математическая модель поставленной задачи может быть записана следующим образом:

$$y_1 = 10000 \cdot x_1 + 15000 \cdot x_2 \to max$$

$$y_2 = 7000 \cdot x_1 + 11000 \cdot x_2 \to min$$

$$y_3 = 350 \cdot x_1 + 400 \cdot x_2 \to min$$

$$y_4 = 6 \cdot x_1 + 8 \cdot x_2 \to min$$

С учетом требований пакета MATLAB к постановке задач оптимизации, необходимо представить целевые функции как поиск минимумов, а ограничения записать в виде $g_i(x) \leq b_i$. В итоге получим:

$$z_1 = -10000 \cdot x_1 - 15000 \cdot x_2 \rightarrow min$$

$$y_2 = 7000 \cdot x_1 + 11000 \cdot x_2 \rightarrow min$$

$$y_3 = 350 \cdot x_1 + 400 \cdot x_2 \rightarrow min$$

$$y_4 = 6 \cdot x_1 + 8 \cdot x_2 \rightarrow min$$

$-x_1 \le -20$;

$x_1 \le 30$;

$-x_2 \le -11$;

$x_2 \le 15$;

$5 \cdot x_1 + 4 \cdot x_2 \le 120$;

$1 \cdot x_1 + 2 \cdot x_2 \le 48$.

Заметим, что неравенства в системе являются линейными, и тогда можно переписать систему ограничений в матричном виде:

$$z_1 = -10000 \cdot x_1 - 15000 \cdot x_2 \rightarrow min$$

$$y_2 = 7000 \cdot x_1 + 11000 \cdot x_2 \rightarrow min$$

$$y_3 = 350 \cdot x_1 + 400 \cdot x_2 \rightarrow min$$

$$y_4 = 6 \cdot x_1 + 8 \cdot x_2 \rightarrow min$$

$$\begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \\ 5 & 4 \\ 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le \begin{bmatrix} -20 \\ 30 \\ -11 \\ 15 \\ 120 \\ 48 \end{bmatrix}$$

1.3 Поиск оптимумов частных критериев

Найдем оптимумы каждой из целевых функций независимо от других. Для этого необходимо решить четыре задачи однокритериальной оптимизации при тех же ограничениях, что имеют место для задачи многокритериальной оптимизации, поставленной выше.

Для решения в пакете MATLAB были исполнены скрипты z1.m…z4.m.

Скрипт z1.m:

```
A=[-1 0;1 0;0 -1;0 1;5 4;1 2];%матрица коэффициентов ограничений
b=[-20;30;-11;15;120;48];     %вектор свободных членов ограничений
z1==[-10000 -15000];                    %целевая функция
lb=[20;11];                              %нижняя граница – нули
 [x,z1_opt]=linprog(z1,A,b,[],[],lb)  %решение задачи, [](пустая матрица) стоит на местах, где должны быть коэффициенты и свободные члены  ограничений-равенств – их в задаче нет
y1=-z1_opt                              %расчет критериев
y2=[7000 11000]*x
y3=[350 400]*x
y4=[6 8]*x
```

Результат выполнения скрипта следующий:

```
>> Exiting: One or more of the residuals, duality gap, or total relative error
        has grown 100000 times greater than its minimum value so far:
        the primal appears to be infeasible (and the dual unbounded).
        (The dual residual < TolFun=1.00e-008.)

x =
    0.0010
   12.3633

z1_opt =
 -1.8546e+005

y1 =
  1.8546e+005

y2 =
  1.3600e+005

y3 =
  4.9457e+003

y4 =
   98.9124
```

Таким образом, по критерию z_1 получено

$z_{1\ min} = -185460$; ($y_{1\ max} = 185460$)

при плане выпуска моделей I и II соответственно:

$x_1 = 12{,}36$; $x_2 = 0$;

при этом не все ограничения выполнены. Очевидно, ограничения на минимальное количество продукции

$-x_1 \leq -20$; $-x_2 \leq -11$;

и ограничения на время

$5 \cdot x_1 + 4 \cdot x_2 \leq 120$;

$1 \cdot x_1 + 2 \cdot x_2 \leq 48$.

не могут быть выполнены одновременно. Требуется удалить одну пару ограничений. Логично удалить ограничения на минимальное количество продукции, т.к. ограничения на время заданы существующим производством.

Скорректируем модель:

$$z_1 = -10000 \cdot x_1 - 15000 \cdot x_2 \rightarrow min$$

$$y_2 = 7000 \cdot x_1 + 11000 \cdot x_2 \rightarrow min$$

$$y_3 = 350 \cdot x_1 + 400 \cdot x_2 \rightarrow min$$

$$y_4 = 6 \cdot x_1 + 8 \cdot x_2 \rightarrow min$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 5 & 4 \\ 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 30 \\ 15 \\ 120 \\ 48 \end{bmatrix}$$

Скрипт z1.m (исправлен):

```
A=[1 0;0 1;5 4;1 2];
b=[30;15;120;48];
z1=[-10000 -15000];
lb=[0;0];
[x,z1_opt]=linprog(z1,A,b,[],[],lb)
y1=-z1_opt
y2=[7000 11000]*x
y3=[350 400]*x
y4=[6 8]*x
```

Результат:

```
>> Optimization terminated successfully.
x =
   12.0000
   15.0000
z1_opt =
 -3.4500e+005
y1 =
  3.4500e+005
y2 =
  2.4900e+005
y3 =
```

```
  1.0200e+004
y4 =
  192.0000
```

Оптимизация	y_1	y_2	y_3	y_4	x_1	x_2
Критерий z_1	345000	249000	10200	192	12	15

Скрипт z2.m:

```
A=[1 0;0 1;5 4;1 2];
b=[30;15;120;48];
y2=[7000 11000];
lb=[0;0];
[x,y2_opt]=linprog(y2,A,b,[],[],lb)

y1=[10000 15000]*x
y3=[350 400]*x
y4=[6 8]*x
```

Результат выполнения скрипта следующий:

```
>> Optimization terminated successfully.
x =
  1.0e-016 *
    0.0909
    0.2509
y2_opt =
  3.3964e-013
y1 =
  4.6728e-013
y3 =
  1.3219e-014
y4 =
  2.5528e-016
```

Таким образом, по критерию y_2 получен ожидаемый результат

$y_{2\ min} = 0$

при плане выпуска моделей I и II соответственно:

$x_1 = 0$; $x_2 = 0$

Оптимизация	y_1	y_2	y_3	y_4	x_1	x_2
Критерий y_2	0	0	0	0	0	0

Скрипт z3.m:

```
A=[1 0;0 1;5 4;1 2];
b=[30;15;120;48];
y3=[350 400];
lb=[0;0];
[x,y3_opt]=linprog(y3,A,b,[],[],lb)

y1=[10000 15000]*x
y2=[7000 11000]*x
y4=[6 8]*x
```

Результат выполнения скрипта следующий:

```
>> Optimization terminated successfully.
x =
  1.0e-015 *
    0.2152
    0.4282
y3_opt =
  2.4660e-013
y1 =
  8.5751e-012
y2 =
  6.2167e-012
y4 =
  4.7169e-015
```

Оптимизация	y_1	y_2	y_3	y_4	x_1	x_2
Критерий y_3	0	0	0	0	0	0

Скрипт z4.m:

```
A=[1 0;0 1;5 4;1 2];
b=[30;15;120;48];
y4=[6 8];
lb=[0;0];
[x,y4_opt]=linprog(y4,A,b,[],[],lb)

y1=[10000 15000]*x
y2=[7000 11000]*x
y3=[350 400]*x
```

Результат выполнения скрипта следующий:

```
>> Optimization terminated successfully.
x =
  1.0e-009 *
```

```
   0.2641
   0.4207
y4_opt =
  4.9501e-009
y1 =
  8.9513e-006
y2 =
  6.4763e-006
y3 =
  2.6071e-007
```

Оптимизация	y_1	y_2	y_3	y_4	x_1	x_2
Критерий y_4	0	0	0	0	0	0

1.4 Переход от многокритериальной задачи к однокритериальной

1.4.1. Выделение главного критерия

Как правило, главный критерий выделить достаточно сложно, и эта задача часто не имеет единственного решения. В данном случае переход от исходной многокритериальной задачи к однокритериальной методом выделения главного критерия может быть осуществлен, например, с использованием условия, что «выручка составляет не меньше 80% от максимальной». Тогда главным можно выбрать второй критерий – затраты на материалы. Дополнительное ограничение имеет вид:

$$z_1 = -10000 \cdot x_1 - 15000 \cdot x_2 \leq -(0{,}8 \cdot 345000) = -276000$$

Математическая модель

$$y_2 = 7000 \cdot x_1 + 11000 \cdot x_2 \rightarrow min$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 5 & 4 \\ 1 & 2 \\ -10000 & -15000 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 30 \\ 15 \\ 120 \\ 48 \\ -276000 \end{bmatrix}$$

Скрипт z21.m:

```
A=[1 0;0 1;5 4;1 2;-10000 -15000];
b=[30;15;120;48;-276000];
```

```
y2=[7000 11000];
lb=[0;0];
[x,y2_opt]=linprog(y2,A,b,[],[],lb)

y1=[10000 15000]*x
y3=[350 400]*x
y4=[6 8]*x
```

Результат выполнения

```
>> Optimization terminated successfully.
x =
   19.8857
    5.1429
y2_opt =
  1.9577e+005
y1 =
  2.7600e+005
y3 =
  9.0171e+003
y4 =
  160.4571
```

Вывод: для оптимизации по главному критерию – минимизации стоимости используемых для производства автомобилей материалов – следует изготовить моделей I – 19,89 шт., моделей II – 5,14 шт., при этом:

Оптимизация	y_1	y_2	y_3	y_4	x_1	x_2
Главный критерий	276000	195770	9017	160,46	19,89	5,14

1.4.2 Свертка критериев

Аддитивная свертка критериев

Для использования метода аддитивной свертки необходимо выполнить нормировку критериев с тем, чтобы сделать их значения соизмеримыми, а единицы измерения – безразмерными. Выполним нормировку следующим образом по результатам оптимизации по 1-му критерию:

$$\bar{z}_1 = \frac{z_1}{|z_{1min}|} = \frac{-10 \cdot x_1 - 15 \cdot x_2}{345}$$

$$\bar{z}_2 = \frac{y_2}{|z_{2min}|} = \frac{7 \cdot x_1 + 11 \cdot x_2}{249}$$

$$\bar{z}_3 = \frac{y_3}{|z_{3min}|} = \frac{7 \cdot x_1 + 8 \cdot x_2}{204}$$

$$\bar{z}_4 = \frac{y_4}{|z_{3min}|} = \frac{6 \cdot x_1 + 8 \cdot x_2}{192}$$

Аддитивная свертка производится суммированием критериев

$$f_{add} = \sum \lambda_i \cdot f_i = \lambda_1 \cdot f_1 + \lambda_2 \cdot f_2 + \lambda_3 \cdot f_3 + \lambda_4 \cdot f_4$$

$$\sum \lambda_i = 1$$

Примем $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0{,}25$, считая нормированные критерии одинаково важными. Тогда, учитывая ограничение по минимальной выручке, получим следующую задачу однокритериальной оптимизации:

$$f = 0{,}25 \cdot \left(\begin{gathered} \frac{-10 \cdot x_1 - 15 \cdot x_2}{345} + \frac{7 \cdot x_1 + 11 \cdot x_2}{249} + \\ + \frac{7 \cdot x_1 + 8 \cdot x_2}{204} + \frac{6 \cdot x_1 + 8 \cdot x_2}{192} \end{gathered} \right) =$$

$$= 0{,}02269 \cdot x_1 + 0{,}02040 \cdot x_2 \to min$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 5 & 4 \\ 1 & 2 \\ -10000 & -15000 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 30 \\ 15 \\ 120 \\ 48 \\ -276000 \end{bmatrix}$$

Скрипт z22.m:

```
A=[1 0;0 1;5 4;1 2;-10000 -15000];
b=[30;15;120;48;-276000];
f=[0.02269;0.02040];
lb=[0;0];
[x,f_opt]=linprog(f,A,b,[],[],lb)

y1=[10000 15000]*x
y2=[7000 11000]*x
y3=[350 400]*x
y4=[6 8]*x
```

Результаты:

```
>> Optimization terminated successfully.
x =
    5.1000
   15.0000
f_opt =
    0.4217
y1 =
  2.7600e+005
y2 =
  2.0070e+005
y3 =
  7.7850e+003
y4 =
  150.6000
```

Таким образом, метод аддитивной свертки позволил получить решение:

$x_1 = 5{,}1$; $x_2 = 15$;

Оптимизация	y_1	y_2	y_3	y_4	x_1	x_2
Аддитивная свертка	276000	200700	7785	150,6	5,1	15

Мультипликативная свертка критериев

Формула мультипликативной свертки

$$F = \prod f_i^{\lambda_i} = f_1^{\lambda_1} \cdot f_2^{\lambda_2}$$

Примем $\lambda_1 = \lambda_2 = \lambda_2 = \lambda_2 = 1$, считая критерии одинаково важными. Тогда их нормировка не требуется. Так как одна целевая функция неположительна, и требует минимизации, то следует использовать альтернативный вид мультипликативного критерия

$$F = \frac{1}{-z_1} \cdot z_2 \cdot z_3 \cdot z_4 \to min$$

$$z_1 = -10000 \cdot x_1 - 15000 \cdot x_2$$

$$y_2 = 7000 \cdot x_1 + 11000 \cdot x_2$$

$$y_3 = 350 \cdot x_1 + 400 \cdot x_2$$

$$y_4 = 6 \cdot x_1 + 8 \cdot x_2$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 5 & 4 \\ 1 & 2 \\ -10000 & -15000 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le \begin{bmatrix} 30 \\ 15 \\ 120 \\ 48 \\ -276000 \end{bmatrix}$$

Оформим критерий отдельным m-файлом myfun.m:

```
function F=myfun(x)
z1=[-10000 -15000]*x;
z2=[7000 10000]*x;
z3=[350 400]*x;
z4=[6 8]*x;
F=-z2*z3*z4/z1;
```

Скрипт z23.m составим с учетом нелинейности мультипликативного критерия:

```
A=[1 0;0 1;5 4;1 2;-10000 -15000];
b=[30;15;120;48;-276000];
lb=[1;1];
ub=[inf;inf];
[x,F_opt]=fmincon(@myfun,lb,A,b,[],[],lb,ub)

y1=[10000 15000]*x
y2=[7000 11000]*x
y3=[350 400]*x
y4=[6 8]*x
```

Результаты:

```
Optimization terminated successfully:
 First-order optimality measure less than options.TolFun and
  maximum constraint violation is less than options.TolCon
Active Constraints:
     4
     7

x =
    5.1000
   15.0000
F_opt =
  7.8884e+005
y1 =
```

```
     276000
y2 =
     200700
y3 =
       7785
y4 =
  150.6000
```

Таким образом, метод мультипликативной свертки позволил получить решение:

$x_1 = 5,1$; $x_2 = 15$;

Оптимизация	y_1	y_2	y_3	y_4	x_1	x_2
Мультипликативная свертка	276000	200700	7785	150,6	5,1	15

Вывод: и аддитивный, и мультипликативный критерии дали одинаковые планы выпуска.

1.4.3 Максимин или минимакс

Минимизируется максимальный из набора нормированных критериев. Задача формулируется следующим образом:

$$max\{\bar{z}_1; \bar{z}_2; \bar{z}_3; \bar{z}_4\} \to min$$

$$\bar{z}_1 = \frac{1}{\frac{y_1}{|z_{1min}|}} = \frac{345}{10 \cdot x_1 + 15 \cdot x_2}$$

$$\bar{z}_2 = \frac{y_2}{|z_{2min}|} = \frac{7 \cdot x_1 + 11 \cdot x_2}{249}$$

$$\bar{z}_3 = \frac{y_3}{|z_{3min}|} = \frac{7 \cdot x_1 + 8 \cdot x_2}{204}$$

$$\bar{z}_4 = \frac{y_4}{|z_{3min}|} = \frac{6 \cdot x_1 + 8 \cdot x_2}{192}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 5 & 4 \\ 1 & 2 \\ -10000 & -15000 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le \begin{bmatrix} 30 \\ 15 \\ 120 \\ 48 \\ -276000 \end{bmatrix}$$

Формируем функцию критериев, оформляем ее отдельным m-файлом

```
function f=fmimax(x)
function f=min_max(x)
f(1)=345/([10 15]*x);
f(2)=([7 11]*x)/249;
f(3)=([7 8]*x)/204;
f(4)=([6 8]*x)/192;
```

Скрипт z24.m:

```
A=[1 0;0 1;5 4;1 2;-10000 -15000];
b=[30;15;120;48;-276000];
lb=[1;1];
ub=[inf;inf];
[x,F_opt]=fminimax(@min_max,lb,A,b,[],[],lb,ub)

y1=[10000 15000]*x
y2=[7000 11000]*x
y3=[350 400]*x
y4=[6 8]*x
```

Результат:

```
>> Optimization terminated successfully:
Search direction less than 2*options.TolX and  maximum constraint violation is less than
options.TolCon
Active Constraints:
     8
x =
    8.7240
   12.5840
F_opt =
    1.2500    0.8012    0.7928    0.7970
y1 =
  2.7600e+005
y2 =
  1.9949e+005
y3 =
  8.0870e+003
y4 =
  153.0160
```

Таким образом, метод минимаксной свертки позволил получить решение:

$x_1 = 8{,}72$; $x_2 = 12{,}58$;

Оптимизация	y_1	y_2	y_3	y_4	x_1	x_2
Минимакс	276000	199490	8087	153.0	8,72	12,58

Вывод: минимаксный критерий в данном случае дал результат, отличающийся от ранее полученных.

1.4.4 Метод последовательных уступок

Для того чтобы осуществить переход к задаче однокритериальной оптимизации методом последовательных уступок, необходимо ранжировать критерии по степени важности. Пусть z_1 важнее z_2; z_2 важнее z_3; z_3 важнее z_4. Кроме того, известно значение уступки Δ, на которую предприятие готово пойти при оптимизации каждого критерия (значение Δ может быть различным для различных критериев, здесь примем Δ = 0,2 или 20% для всех критериев).

Таким образом, получаем четыре задачи однокритериальной оптимизации:

1) Найти минимум z_1 при исходных ограничениях:

$$z_1 = -10000 \cdot x_1 - 15000 \cdot x_2 \to min$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 5 & 4 \\ 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le \begin{bmatrix} 30 \\ 15 \\ 120 \\ 48 \end{bmatrix}$$

Эта задача была решена в п. 1.1 при поиске оптимумов частных критериев.

2) Найти минимум z_2 в исходной системе ограничений, дополненной неравенством: $z_1 \le z_{1\min} + \Delta$, где $\Delta = -0.2 \cdot z_{1\min}$ (выручка должна составлять не

меньше 0% от максимальной, то величина уступки Δ_1 по модулю равна 20% от оптимального значения критерия z_1).

Тогда получаем следующую задачу однокритериальной оптимизации:

$$y_2 = 7000 \cdot x_1 + 11000 \cdot x_2 \to min$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 5 & 4 \\ 1 & 2 \\ -10000 & -15000 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 30 \\ 15 \\ 120 \\ 48 \\ -276000 \end{bmatrix}$$

Эта задача также решена ранее методом выбора главного критерия в п.2.1; $y_{2\,min} = 195770$.

Новое ограничение

$y_2 \leq y_{2min} + 0.2 \cdot y_{2min} = 234900$

Задача оптимизации третьего критерия:

$$y_3 = 350 \cdot x_1 + 400 \cdot x_2 \to min$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 5 & 4 \\ 1 & 2 \\ -10000 & -15000 \\ 7000 & 10000 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 30 \\ 15 \\ 120 \\ 48 \\ -276000 \\ 234900 \end{bmatrix}$$

Скрипт z25.m:

```
A=[1 0;0 1;5 4;1 2;-10000 -15000;7000 10000];
b=[30;15;120;48;-276000;234900];
f=[350;400];
lb=[0;0];
[x,f_opt]=linprog(f,A,b,[],[],lb)

y1=[10000 15000]*x
y2=[7000 11000]*x
y3=[350 400]*x
y4=[6 8]*x
```

Результат

```
>> Optimization terminated successfully.
x =
    5.1000
   15.0000
f_opt =
```

```
  7.7850e+003
y1 =
  2.7600e+005
y2 =
  2.0070e+005
y3 =
  7.7850e+003
y4 =
  150.6000
```
$y_{3\,min}$ = 7785.

Новое ограничение

$y_3 \le y_{3min} + 0.2 \cdot y_{3min} = 9342$

Задача оптимизации четвертого критерия:

$$y_4 = 6 \cdot x_1 + 8 \cdot x_2 \to min$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 5 & 4 \\ 1 & 2 \\ -10000 & -15000 \\ 7000 & 10000 \\ 350 & 400 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le \begin{bmatrix} 30 \\ 15 \\ 120 \\ 48 \\ -276000 \\ 234900 \\ 9342 \end{bmatrix}$$

Скрипт z251.m:

```
A=[1 0;0 1;5 4;1 2;-10000 -15000;7000 10000;350 400];
b=[30;15;120;48;-276000;234900;9342];
f=[350;400];
lb=[0;0];
[x,f_opt]=linprog(f,A,b,[],[],lb)

y1=[10000 15000]*x
y2=[7000 11000]*x
y3=[350 400]*x
y4=[6 8]*x
```

Результат

```
>> Optimization terminated successfully.
x =
    5.1000
   15.0000
f_opt =
  7.7850e+003
y1 =
  2.7600e+005
```

```
y2 =
  2.0070e+005
y3 =
  7.7850e+003
y4 =
  150.6000
```

Таким образом, метод последовательных уступок позволил получить решение:

x_1 = 5,1; x_2 = 15;

Оптимизация	y_1	y_2	y_3	y_4	x_1	x_2
Последовательные уступки	276000	200700	7785	150,6	5,1	15

1.4.5 Метод достижения цели fgoalattain

В Matlab есть специальная функция, реализующая этот метод. Постановка задачи здесь – это набор оптимальных критериев – цель минимизации, которую нужно достигнуть, и ограничения. Например, попытаемся достигнуть той же цели, что и в п.2.4, считая её оптимальной:

$$z_1 = -10000 \cdot x_1 - 15000 \cdot x_2 \rightarrow -276000$$

$$y_2 = 7000 \cdot x_1 + 11000 \cdot x_2 \rightarrow 200700$$

$$y_3 = 350 \cdot x_1 + 400 \cdot x_2 \rightarrow 7785$$

$$y_4 = 6 \cdot x_1 + 8 \cdot x_2 \rightarrow 150$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 5 & 4 \\ 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le \begin{bmatrix} 30 \\ 15 \\ 120 \\ 48 \end{bmatrix}$$

Оформим функцию критериев m-файлом:

```
function f=fgoal(x)
f(1)=[-10000 -15000]*x;
f(2)=[7000 11000]*x;
f(3)=[350 400]*x;
f(4)=[6 8]*x;
Скрипт z26.m:
```

```
A=[1 0;0 1;5 4;1 2];
b=[30;15;120;48];
x0=[5;5];                                  %начальное значение
goal=[-276000;200700;7785;150];   %вектор целей
weight=abs(goal);                          % вектор весов ЦФ
lb=[0;0];
ub=[inf;inf];
[x,fval,attainfactor]=fgoalattain('fgoal',x0,goal,weight,A,b,[],[],lb,ub)

y1=[10000 15000]*x
y2=[7000 11000]*x
y3=[350 400]*x
y4=[6 8]*x
```

Результат:

```
>> Optimization terminated successfully:
 Search direction less than 2*options.TolX and   maximum constraint violation is less than
options.TolCon
Active Constraints:
     4
     7
    10
x =
    5.0475
   15.0000
fval =
  1.0e+005 *
   -2.7548    2.0033    0.0777    0.0015
attainfactor =
    0.0019
y1 =
  2.7548e+005
y2 =
  2.0033e+005
y3 =
  7.7666e+003
y4 =
  150.2852
```

Таким образом, метод достижения цели позволил получить решение:

$x_1 = 5{,}1$; $x_2 = 15$;

Оптимизация	y_1	y_2	y_3	y_4	x_1	x_2
Достижение цели	275480	200330	7767	150,3	5,0	15
% к заданному	99,81	99,82	99,76	100,19	-	-

Действительно, все критерии практически равны своим оптимальным значениям, что соответствует тому факту, что критерии были приняты одинаково важными.

1.4.6 Ведение метрики в пространстве критериев

Введем в в пространстве критериев евклидову метрику

$$d = \sqrt{\sum \left(1 - \frac{f_i}{|f_{i\,min}|}\right)^2}$$

где $f_{i\,min}$ – оптимальное значение каждого критерия (например, результаты п.2.4). Для ускорения вычислений опустим корень и модифицируем метрику

$$f = \sum \left(1 - \frac{f_i}{|f_{i\,min}|}\right)^2$$

Эта метрика и будет целевой функцией, которую нужно минимизировать. Задача формулируется следующим образом:

$$f = \left(1 - \frac{10000 \cdot x_1 + 15000 \cdot x_2}{276000}\right)^2 + \left(1 - \frac{7000 \cdot x_1 + 11000 \cdot x_2}{200700}\right)^2 +$$

$$+ \left(1 - \frac{350 \cdot x_1 + 400 \cdot x_2}{7785}\right)^2 + \left(1 - \frac{6 \cdot x_1 + 8 \cdot x_2}{150}\right)^2 \to min$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 5 & 4 \\ 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le \begin{bmatrix} 30 \\ 15 \\ 120 \\ 48 \end{bmatrix}$$

или

$$f = \left(1 - \frac{10 \cdot x_1 + 15 \cdot x_2}{276}\right)^2 + \left(1 - \frac{7 \cdot x_1 + 11 \cdot x_2}{200{,}7}\right)^2 +$$

$$+ \left(1 - \frac{350 \cdot x_1 + 400 \cdot x_2}{7785}\right)^2 + \left(1 - \frac{6 \cdot x_1 + 8 \cdot x_2}{150}\right)^2 \to min$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 5 & 4 \\ 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le \begin{bmatrix} 30 \\ 15 \\ 120 \\ 48 \end{bmatrix}$$

Скрипт z27.m:

```
A=[1 0;0 1;5 4;1 2];
b=[30;15;120;48];
f=inline('(1-([10 15]*x)/276)^2+(1-([7 11]*x)/200.7)^2+(1-([350 400]*x)/7785)^2+(1-([6
8]*x)/150)^2');
lb=[1;1];
ub=[inf;inf];
[x,f_opt]=fmincon(f,lb,A,b,[],[],lb,ub)

y1=[10000 15000]*x
y2=[7000 11000]*x
y3=[350 400]*x
y4=[6 8]*x
```

Результаты:

```
Optimization terminated successfully:
 Magnitude of directional derivative in search direction
  less than 2*options.TolFun and maximum constraint violation
  is less than options.TolCon
 No Active Constraints

x =
    9.0764
   12.0726
f_opt =
    0.0016
y1 =
  2.7185e+005
y2 =
  1.9633e+005
y3 =
  8.0058e+003
y4 =
  151.0396
```

Метод введения метрики позволил получить решение:

$x_1 = 9{,}1$; $x_2 = 12{,}1$;

Оптимизация	y_1	y_2	y_3	y_4	x_1	x_2
По метрике	271850	196330	8006	151,0	9,1	12,1

1.5 Оценка Парето-оптимальности полученных решений

Для того чтобы уменьшить количество альтернатив, среди которых лицо, принимающее решение (ЛПР), должно сделать выбор, можно выделить множество Парето среди всех полученных решений (то есть найти множество недоминируемых решений). Для этого сведем их в табл. 1.

Таблица 1

Метод перехода к однокритериальной задаче	y_1	y_2	y_3	y_4	x_1	x_2
А. Выделение главного критерия	276000	195770	9017	160,46	19,89	5,14
Б1. Аддитивная свертка	276000	200700	7785	150,6	5,1	15
Б2. Мультипликативная свертка	276000	200700	7785	150,6	5,1	15
В. Минимакс	276000	199490	8087	153,0	8,72	12,58
Г. Метод последовательных уступок	276000	200700	7785	150,6	5,1	15
Д. Метод достижения цели	275480	200330	7767	150,3	5,0	15
Е. Введение метрики в пространстве критериев	271850	196330	8006	151,0	9,1	12,1

Считая, что критерии в таблице расположены по важности, вычеркиваем сначала строки с наименьшим y_1; затем с наибольшей суммой затрат на производство ($y_2 + y_3$), затем с наибольшим временем y_4.

Таким образом, Парето-оптимальным является только решение А.

1.6 Решение задачи стохастического программирования

Рассмотрим задачу стохастического программирования на основе задачи однокритериальной оптимизации, которая была получена из исходной методом введения метрики в пространстве критериев:

$$f = \left(1 - \frac{10 \cdot x_1 + 15 \cdot x_2}{276}\right)^2 + \left(1 - \frac{7 \cdot x_1 + 11 \cdot x_2}{200{,}7}\right)^2 +$$

$$+ \left(1 - \frac{350 \cdot x_1 + 400 \cdot x_2}{7785}\right)^2 + \left(1 - \frac{6 \cdot x_1 + 8 \cdot x_2}{150}\right)^2 \to min$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 5 & 4 \\ 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 30 \\ 15 \\ 120 \\ 48 \end{bmatrix}$$

Пусть временя работы робота является нормально распределенной случайной величиной, а именно:

$$b_3 = b_3(M, \sigma)$$

где M = 120 часов – математическое ожидание верхнего предела затрат времени на изготовление деталей;

σ = 30 часов – среднеквадратическое отклонение, здесь принято произвольно, определяется опытным путем.

Будем считать, что робот может с некоторой вероятностью работать и дольше 120 часов в неделю. Тогда третье ограничение системы преобразуется в вероятностное, и можно рассматривать его в виде выражения:

$$P\left(5 \cdot x_1 + 4 \cdot x_2 \leq b_3(120{,}30)_{b_3 \geq 120}\right) \geq \alpha$$

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

Решим эту задачу стохастического программирования при

α = 0.1, 0.2, … , 0.9.

Для этого заменим вероятностное ограничение детерминированным

$$5 \cdot x_1 + 4 \cdot x_2 \leq h_3(\alpha)$$

где h_3 – корень уравнения

$P\left(b_3(120{,}30)_{b_3 \geq 120}\right) \geq \alpha$ (*)

Для решения этого уравнения используем функцию Matlab (пакет «статистика»)

norminv(PLeft, MU, SIGMA)

возвращающую корень уравнения (*).

Скрипт z28.m:

```
A=[1 0;0 1;5 4;1 2];
b=[30;15;120;48];
f=inline('(1-([10 15]*x)/276)^2+(1-([7 11]*x)/200.7)^2+(1-([350 400]*x)/7785)^2+(1-([6
8]*x)/150)^2');
lb=[1;1];
ub=[inf;inf];
%determ – детерминираванная оптимизация
[x,f_opt]=fmincon(f,lb,A,b,[],[],lb,ub);

ydet1=[10000 15000]*x;
ydet2=[7000 11000]*x;
ydet3=[350 400]*x;
ydet4=[6 8]*x;

%stoh – стохастическая оптимизация
alfa=0.1:0.1:0.9;
bst=b;
yst=zeros(numel(alfa),7); %матрица выходных данных

for i=1:numel(alfa)
bst(3)=b(3)-norminv(alfa(i)/2,0,30); %решение (*)

xst=fmincon(f,lb,A,bst,[],[],lb,ub);
yst(i,1)=xst(1);
yst(i,2)=xst(2);
yst(i,3)=[10000 15000]*x;
yst(i,4)=[7000 11000]*xst;
yst(i,5)=[350 400]*xst;
yst(i,6)=[6 8]*xst;
yst(i,7)=bst(3);
end
%цикл вывода матрицы выходных данных
for i=1:numel(alfa)
fprintf('alfa=%.1f,x1=%4.2f,x2=%4.2f,y1=%6.0f,y2=%6.0f,y3=%6.1f,y4=%6.1f,b3=%6.2f\n',...
   alfa(i),yst(i,1),yst(i,2),yst(i,3),yst(i,4),yst(i,5),yst(i,6),yst(i,7))
end

fprintf('determ:\n')
fprintf('x1=%5.2f,x2=%5.2f,y1=%6.0f,y2=%6.0f,y3=%6.1f,y4=%6.2f\n',x,ydet1,ydet2,ydet3,yde
t4)
```

Результат работы:

```
alfa=0.1,x1=9.08,x2=12.07,y1=271854,y2=196334,y3=8005.8,y4= 151.0,b3=169.35
alfa=0.2,x1=9.08,x2=12.07,y1=271854,y2=196334,y3=8005.8,y4= 151.0,b3=158.45
alfa=0.3,x1=9.08,x2=12.07,y1=271854,y2=196334,y3=8005.8,y4= 151.0,b3=151.09
alfa=0.4,x1=9.08,x2=12.07,y1=271854,y2=196334,y3=8005.8,y4= 151.0,b3=145.25
alfa=0.5,x1=9.08,x2=12.07,y1=271854,y2=196334,y3=8005.8,y4= 151.0,b3=140.23
alfa=0.6,x1=9.08,x2=12.07,y1=271854,y2=196334,y3=8005.8,y4= 151.0,b3=135.73
alfa=0.7,x1=9.08,x2=12.07,y1=271854,y2=196334,y3=8005.8,y4= 151.0,b3=131.56
alfa=0.8,x1=9.08,x2=12.07,y1=271854,y2=196334,y3=8005.8,y4= 151.0,b3=127.60
alfa=0.9,x1=9.08,x2=12.07,y1=271854,y2=196334,y3=8005.8,y4= 151.0,b3=123.77
determ:
x1= 9.08,x2=12.07,y1=271854,y2=196334,y3=8005.8,y4=151.04
```

Вывод: случайный характер изменения ограничения b_3 в сторону увеличения никак не влияет на оптимальный план производства.

Исследование влияния этого фактора в сторону уменьшения путем замены строки

bst(3)=b(3)-norminv(alfa(i)/2,0,30);

на строку

bst(3)=b(3)+norminv(alfa(i)/2,0,30);

дало следующие результаты:

```
alfa=0.1,x1=2.13,x2=15.00,y1=271854,y2=179916,y3=6745.8,y4= 132.8,b3= 70.65
alfa=0.2,x1=4.31,x2=15.00,y1=271854,y2=195175,y3=7508.7,y4= 145.9,b3= 81.55
alfa=0.3,x1=5.07,x2=15.00,y1=271854,y2=200521,y3=7776.1,y4= 150.4,b3= 88.91
alfa=0.4,x1=9.08,x2=12.07,y1=271854,y2=196334,y3=8005.8,y4= 151.0,b3= 94.75
alfa=0.5,x1=9.08,x2=12.07,y1=271854,y2=196334,y3=8005.8,y4= 151.0,b3= 99.77
alfa=0.6,x1=9.08,x2=12.07,y1=271854,y2=196334,y3=8005.8,y4= 151.0,b3=104.27
alfa=0.7,x1=9.08,x2=12.07,y1=271854,y2=196334,y3=8005.8,y4= 151.0,b3=108.44
alfa=0.8,x1=9.08,x2=12.07,y1=271854,y2=196334,y3=8005.8,y4= 151.0,b3=112.40
alfa=0.9,x1=9.08,x2=12.07,y1=271854,y2=196334,y3=8005.8,y4= 151.0,b3=116.23
determ:
x1= 9.08,x2=12.07,y1=271854,y2=196334,y3=8005.8,y4=151.04
```

Глава 2. Поиск оптимальной стратегии принятия решения с использованием марковских моделей

2.1 Постановка задачи

Задача 15:

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 т и г, где т < M и г < 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, (невозможное событие)

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).

Найти оптимальную стратегию и минимальные затраты на отрезке (использовать алгоритм итераций по стратегии). Показать соответствующее оптимальное решение двойственной задачи линейного программирования.

2.2. Решение

Сформулируем задачу линейного программирования:

$$\begin{cases} \max\limits_{\tau}\left\{e = \sum_{j=1}^{m} \Pi_j \left(\sum_{i=1}^{M} q_j^i V_j^i\right)\right\}, \text{где} \\ \text{M} - \text{число доп. решений } X_i, i = \overline{1, M} \\ \Pi_j = \sum_{k=1}^{m} \Pi_k \, p_{kj}(X_i),\ j = \overline{1, m} \\ \sum_{j=1}^{m} \Pi_j = 1 \\ \sum_{i=1}^{M} q_j^i = 1 \\ \begin{cases} \Pi_j \geq 0,\ j = \overline{1, m} \\ q_j^i \geq 0,\ j = \overline{1, m},\ i = \overline{1, M} \end{cases} \end{cases}$$

Трансформируем ее следующим образом:

$$\max_{(X_i)} \sum_{j=1}^{m} \sum_{i=1}^{M} \omega_{ji} V_j(X_i)$$

$$\begin{cases} \sum_{j=1}^{M} \omega_{ji} - \sum_{k=1}^{m} \sum_{i=1}^{M} p_{kj}(X_i)\omega_{ki} = 0\,, j = \overline{1,m} \\ \sum_{j=1}^{m} \sum_{i=1}^{M} \omega_{ji} = 1 \\ \omega_{ji} \geq 0,\ \ j = \overline{1,m}; i = \overline{1,M} \end{cases}$$

Матрицы переходных вероятностей для каждого из трех состояний автомобиля имеют вид:

$$P_1 = \begin{pmatrix} 0,6 & 0,3 & 0,1 \\ 0,9 & 0,1 & 0 \\ 0,6 & 0,3 & 0,1 \end{pmatrix}$$

$P_2 = \begin{pmatrix} 0,6 & 0,3 & 0,1 \\ 0,7 & 0,2 & 0,1 \\ 0,5 & 0,4 & 0,1 \end{pmatrix}$, где P_1 – матрица перехода для компании F, P_2 – матрица перехода для компании T

Матрицы расходов для компаний F и T соответственно:

$$R_1 = \begin{pmatrix} 0 & 0 & 0 \\ -14 & -14 & -14 \\ -21 & -21 & -21 \end{pmatrix}$$

$$R_2 = \begin{pmatrix} 0 & 0 & 0 \\ -12 & -12 & -12 \\ -19 & -19 & -19 \end{pmatrix}$$

Найдем значения ожидаемых доходов:

$$V_1(X_1) = 0.6 * 0 + 0.3 * 0 + 0.1 * 0 = 0$$

$$V_2(X_1) = 0.9 * (-14) + 0.1 * (-14) + 0 * (-14) = -14$$

$$V_3(X_1) = 0.6 * (-21) + 0.3 * (-21) + 0.1 * (-21) = -21$$

$$V_1(X_2) = 0.6 * 0 + 0.3 * 0 + 0.1 * 0 = 0$$

$$V_2(X_2) = 0.7 * (-12) + 0.2 * (-12) + 0.1 * (-12) = -12$$

$$V_3(X_2) = 0.5 * (-19) + 0.4 * (-19) + 0.1 * (-19) = -19$$

Тогда задача линейного программирования для текущей задачи выглядит следующим образом:

$$\begin{cases} 0 * \omega_{11} - 0 * \omega_{12} - 14\omega_{21} - 12\omega_{22} - 21\omega_{31} - 19\omega_{32} \to max; \\ (1-0.6)\omega_{11} + (1-0.6)\omega_{12} - 0.9\omega_{21} - 0.7\omega_{22} - 0.6\omega_{31} - 0.5\omega_{32} = 0 \\ -0.3\omega_{11} - 0.3\omega_{12} + (1-0.1)\omega_{21} + (1-0.2)\omega_{22} - 0.3\omega_{31} - 0.4\omega_{32} = 0 \\ -0.1\omega_{11} - 0.1\omega_{12} - 0.1\omega_{22} + (1-0.1)\omega_{31} + (1-0.1)\omega_{32} = 0 \\ \sum_{j=1}^{3}\sum_{i=1}^{2} \omega_{ji} = 1, \quad \omega_{ji} \geq 0, \quad j = 1,2,3, \ i = 1,2. \end{cases}$$

, что преобразуется следующим образом:

$$\begin{cases} 14\omega_{21} + 12\omega_{22} + 21\omega_{31} + 19\omega_{32} \to min; \\ 0.4\omega_{11} + 0.4\omega_{12} - 0.9\omega_{21} - 0.7\omega_{22} - 0.6\omega_{31} - 0.5\omega_{32} = 0 \\ -0.3\omega_{11} - 0.3\omega_{12} + 0.9\omega_{21} + 0.8\omega_{22} - 0.3\omega_{31} - 0.4\omega_{32} = 0 \\ -0.1\omega_{11} - 0.1\omega_{12} - 0.1\omega_{22} + 0.9\omega_{31} + 0.9\omega_{32} = 0 \\ \sum_{j=1}^{3}\sum_{i=1}^{2} \omega_{ji} = 1, \quad \omega_{ji} \geq 0, \quad j = 1,2,3, \ i = 1,2. \end{cases}$$

В таком виде задача может быть решена в среде Matlab с использованием следующего скрипта:

```
F = @(x) 14*x(3)+12*x(4)+21*x(5)+19*x(6);
[W,Fmin] = fmincon(F,[0;0;0;0;0;0],[],[],[],[],[],[],@nonlcon)

function [c,ceq] = nonlcon(x)
ceq(1) = 0.4*x(1)+0.4*x(2)-0.9*x(3)-0.7*x(4)-0.6*x(5)-0.5*x(6);
ceq(2) = -0.3*x(1)-0.3*x(2)+0.9*x(3)+0.8*x(4)-0.3*x(5)-0.4*x(6);
ceq(3) = -0.1*x(1)-0.1*x(2)-0.1*x(4)+0.9*x(5)+0.9*x(6);
ceq(4) = x(1)+x(2)+x(3)+x(4)+x(5)+x(6)-1;
c(1)=-x(1);
c(2)=-x(2);
c(3)=-x(3);
c(4)=-x(4);
c(5)=-x(5);
c(6)=-x(6);
end
```

Результат:

```
W =
0.3414
0.3286
0.2538
0.0015
0.0124
```

```
0.0623

Fmin =
5.0139
```

Таким образом, получаем $\begin{cases} q_2^1 = q_3^2 = 1 \\ q_1^1 = q_1^2 = 0.5 \end{cases}$, что логично, так как если машина исправна, то ремонт ей не нужен, а значит, не имеет значения, в какую мастерскую обращаться.

Тогда в оптимальном решении первый элемент может быть любым, а значит имеется 2 стратегии:

$\tau_1^* = (X_1\ X_1\ X_2)^T$

и

$\tau_2^* = (X_2\ X_1\ X_2)^T$.

То есть:

- Если машина исправна, выбор мастерской не имеет значения
- При мелком ремонте следует выбрать мастерскую F
- При крупном ремонте следует обратиться в мастерскую Т

При обеих стратегиях минимальные затраты равны 5.

2.3 Вывод

Для исходной задачи было получено оптимальное решение путем применения метода линейного программирования. Так как в случае отсутствия поломки автомобиля не имеет значение, в какую мастерскую обращаться, в результате получилось 2 оптимальных решения.

Глава 3. Решение задачи анализа потокового графа с использованием методики GERT и алгебры потоковых графов

3.1 Постановка задачи

Дано:

1. Граф GERT-сети

2. каждой дуге-работе («ij») поставлены в соответствие следующие данные:

А) Закон распределения времени выполнения работы. Будем считать его нормальным

Б) параметры закона распределения (математическое ожидание Mи дисперсия D)

В) вероятность выполнения работы, показанная на графе.

Найти:

Часть 1

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

Часть 2

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

Исходный граф системы представлен на рисунке 1 (вариант №38).

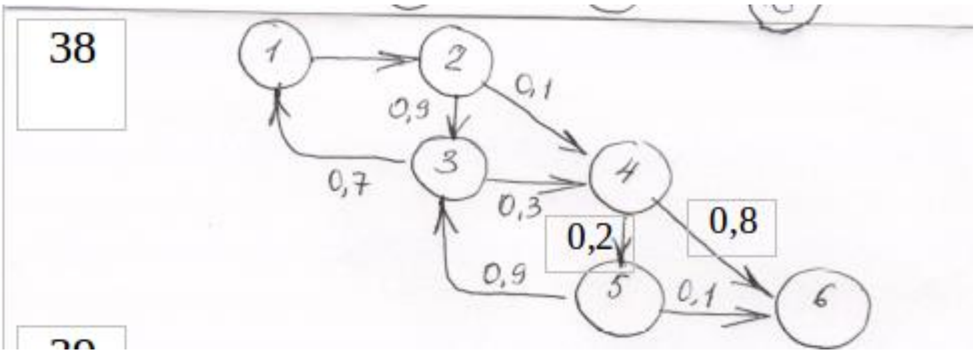

Рис.1. Исходный граф системы

3.2 Выполнение части 1

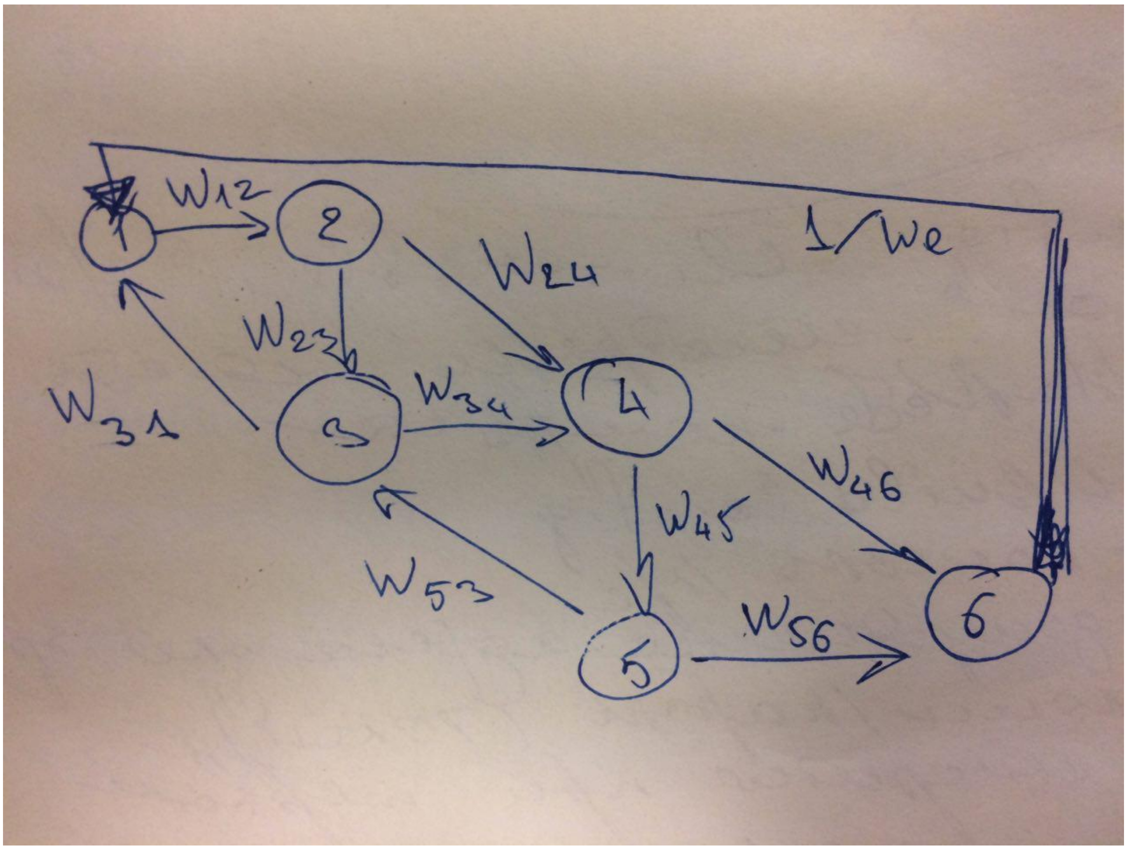

Рис.2. Граф GERT-сети

W-функция для заданной GERT-сети представлена в таблице 1.

Таблица 1.W-функция GERT сети

Начало	Конец	Вероятность p_{ij}	Мат. ожидание	Дисперсия	W-функция $W = p \cdot \exp(\mu t + t^2 \sigma^2/2)$
1	2	1	38	16	$0{,}5 \cdot \exp(10t + 8t^2)$
2	3	0,9	19	9	$0{,}9 \cdot \exp(19t + 4{,}5t^2)$
2	4	0,1	32	16	$0{,}1 \cdot \exp(32t + 8t^2)$
3	1	0,7	33	16	$0{,}7 \cdot \exp(33t + 8t^2)$
3	4	0,3	20	9	$0{,}3 \cdot \exp(20t + 4{,}5t^2)$
4	5	0,2	28	16	$0{,}2 \cdot \exp(28t + 8t^2)$
4	6	0,8	30	16	$0{,}8 \cdot \exp(30t + 8t^2)$
5	3	0,9	37	16	$0{,}9 \cdot \exp(37t + 8t^2)$
5	6	0,1	43	25	$0{,}1 \cdot \exp(43t + 12{,}5t^2)$

Замкнем вероятностный граф дугой Wa(s)=1/We(s)

Найдем все петли первого порядка:

$$W_{12}W_{23}W_{31}$$

$$W_{12}W_{24}W_{45}W_{53}W_{31}$$

$$W_{34}W_{45}W_{53}$$

$$W_{12}W_{24}W_{45}W_{56}\,1/We$$

$$W_{12}W_{23}W_{34}W_{45}W_{56}\,1/We$$

$$W_{12}W_{23}W_{34}W_{46}\,1/We$$

$$W_{12}W_{24}W_{46}\,1/We$$

Петлей большего порядка нет.

Запишем топологическое уравнение:

$$H = 1 - W_{12}W_{23}W_{31} - W_{12}W_{24}W_{45}W_{53}W_{31} - W_{34}W_{45}W_{53} - 1/We * (W_{12}W_{24}W_{45}W_{56} + W_{12}W_{23}W_{34}W_{45}W_{56} + W_{12}W_{23}W_{34}W_{46} + W_{12}W_{24}W_{46}) = 0$$

$$W_E = \frac{A}{B}, \text{где}$$

$$A = W_{12}W_{24}W_{45}W_{56} + W_{12}W_{23}W_{34}W_{45}W_{56} + W_{12}W_{23}W_{34}W_{46} + W_{12}W_{24}W_{46}$$

$$B = 1 - W_{12}W_{23}W_{31} - W_{12}W_{24}W_{45}W_{53}W_{31} - W_{34}W_{45}W_{53}$$

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

$W_E(s) = p_E M_E(s) \Rightarrow M_E(s) = W_E(s)/p_E = W_E(s)/W_E(0)$

Вычислим первую и вторую производные по s функции $M_E(s)$ при s=0, найдем математическое ожидание:

$$\mu_{1E} = \frac{\delta M_{E(s)}}{\delta s} s = 0$$

И дисперсию:

$$\sigma^2 = \mu_{2E} - (\mu_{1E})^2$$

Вероятность выхода в завершающий узел графа:

$p_E = W_E(0)$

Решим это в Matlab (скрипт приведен в приложении):

Для расчета параметров был написан следующий код в среде Matlab:

```
clc
P12 = 1; M12 = 38; D12 = 16;
P23 = 0.9; M23 = 19; D23 = 9;
P24 = 0.1; M24 = 32; D24 = 16;
P31 = 0.7; M31 = 33; D31 = 16;
P34 = 0.3; M34 = 20; D34 = 9;
P45 = 0.2; M45 = 28; D45 = 16;
P46 = 0.8; M46 = 30; D46 = 16;
P53 = 0.9; M53 = 37; D53 = 16;
P56 = 0.1; M56 = 43; D56 = 25;

syms s
W12 = P12*exp(M12*s+D12/2*s^2);
W23 = P23*exp(M23*s+D23/2*s^2);
W24 = P24*exp(M24*s+D24/2*s^2);
W31 = P31*exp(M31*s+D31/2*s^2);
```

```
W34 = P34*exp(M34*s+D34/2*s^2);
W45 = P45*exp(M45*s+D45/2*s^2);
W46 = P46*exp(M46*s+D46/2*s^2);
W53 = P53*exp(M53*s+D53/2*s^2);
W56 = P56*exp(M56*s+D56/2*s^2);

We =
(W12*W24*W45*W56+W12*W23*W34*W45*W56+W12*W23*W34*W46+W12*W2
W12*W23*W31-W12*W24*W45*W53*W31-W34*W45*W53);

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

Me = We/We0; %мат.ожидание и дисперсия

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

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

D = m2 - (m1)^2 %дисперсия времени выхода в завершающий узел
```

Результаты работы программы:

We =

-(400*exp(4*s*(6*s + 25)) + 1080*exp(s*(25*s + 107)) + 10*exp((s*(73*s + 282))/2) + 27*exp((s*(75*s + 296))/2))/(63*exp(8*s*(5*s + 21)) + 270*exp((s*(41*s + 170))/2) + 3150*exp((s*(41*s + 180))/2) - 5000)

We0 =

1

m1 =

478000/1517

m2 =

373807370155/2301289

D =

145323370155/2301289

Были получены следующие результаты:

1. Вероятность выхода в завершающий узел равна 1.
2. Математическое ожидание 315,1.
3. Дисперсия времени выхода в завершающий узел 63148,68

3.3 Выполнение части 2

Определим матрицу Q с обратной связью:

$$Q = \begin{matrix} 0 & q_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & q_{23} & q_{24} & 0 & 0 \\ q_{31} & 0 & 0 & q_{34} & 0 & 0 \\ 0 & 0 & 0 & 0 & q_{45} & q_{46} \\ 0 & 0 & q_{53} & 0 & 0 & q_{56} \\ w_{61} & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Определим матрицу коэффициентов A = $I_6 - Q^T$.

$$A = \begin{matrix} 1 & 0 & -q_{31} & 0 & 0 & -w_{61} \\ -q_{12} & 1 & 0 & 0 & 0 & 0 \\ 0 & -q_{23} & 1 & 0 & -q_{53} & 0 \\ 0 & -q_{24} & -q_{34} & 1 & 0 & 0 \\ 0 & 0 & 0 & -q_{45} & 1 & 0 \\ 0 & 0 & 0 & -q_{46} & -q_{56} & 1 \end{matrix}$$

Находим det(A), затем:

$$\frac{\delta \det(A)}{\delta w_{61}}$$

$$\det(A|w_{61} = 0)$$

Тогда $W_E(S)$ выводится с помощью формулы:

$$W_{E(S)} = -\frac{\frac{\delta \det(A)}{\delta w_{61}}}{\det(A|w_{61} = 0)}$$

Рассчитаем в Matlab:

```
clc; clearvars

syms q12
syms q23
syms q24
syms q31
syms q34
syms q45
syms q46
syms q53
syms q56
syms w61
syms s

Q = [0 q12 0 0 0 0;
0 0 q23 q24 0 0;
q31 0 0 q34 0 0;
0 0 0 0 q45 q46;
0 0 q53 0 0 q56;
w61 0 0 0 0 0];

A1 = eye(size(Q,1)) - transpose(Q);
disp(A1);

det_A1 = det(A1);
det_dw = diff(det_A1, w61);

det2_A1 = subs(det_A1, w61, 0);
We = -det_dw/det2_A1;
disp(We);
```

Результат:

```
[ 1, 0, -q31, 0, 0, -w61]
[ -q12, 1, 0, 0, 0, 0]
[ 0, -q23, 1, 0, -q53, 0]
[ 0, -q24, -q34, 1, 0, 0]
[ 0, 0, 0, -q45, 1, 0]
[ 0, 0, 0, -q46, -q56, 1]

-(q12*q24*q46 + q12*q23*q34*q46 + q12*q24*q45*q56 +
q12*q23*q34*q45*q56)/(q12*q23*q31 + q34*q45*q53 + q12*q24*q31*q45*q53 - 1)
```

Полученный $W_E(S)$ совпадает с полученным ранее с точностью до знаков. Найдем из него оставшиеся переменные следующим скриптом:

```
clc; clearvars

P12 = 1; M12 = 38; D12 = 16;
P23 = 0.9; M23 = 19; D23 = 9;
P24 = 0.1; M24 = 32; D24 = 16;
P31 = 0.7; M31 = 33; D31 = 16;
P34 = 0.3; M34 = 20; D34 = 9;
P45 = 0.2; M45 = 28; D45 = 16;
P46 = 0.8; M46 = 30; D46 = 16;
P53 = 0.9; M53 = 37; D53 = 16;
P56 = 0.1; M56 = 43; D56 = 25;


syms q12
syms q23
syms q24
syms q31
syms q34
syms q45
syms q46
syms q53
syms q56
syms w61
syms s

Q = [0 q12 0 0 0 0;
0 0 q23 q24 0 0;
q31 0 0 q34 0 0;
0 0 0 0 q45 q46;
0 0 q53 0 0 q56;
w61 0 0 0 0 0];

A1 = eye(size(Q,1)) - transpose(Q);
disp(A1);

det_A1 = det(A1);
disp(det_A1);

det_dw = diff(det_A1, w61);
```

```
disp(det_dw);

det2_A1 = subs(det_A1, w61, 0);
disp(det2_A1);

We = -det_dw/det2_A1;
disp(We);

We = subs(We, q12, P12*exp(M12*s+D12/2*s^2));
We = subs(We, q23, P23*exp(M23*s+D23/2*s^2));
We = subs(We, q24, P24*exp(M24*s+D24/2*s^2));
We = subs(We, q31, P31*exp(M31*s+D31/2*s^2));
We = subs(We, q34, P34*exp(M34*s+D34/2*s^2));
We = subs(We, q45, P45*exp(M45*s+D45/2*s^2));
We = subs(We, q46, P46*exp(M46*s+D46/2*s^2));
We = subs(We, q53, P53*exp(M53*s+D53/2*s^2));
We = subs(We, q56, P56*exp(M56*s+D56/2*s^2));

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

Me = We/We0; %мат.ожидание и дисперсия

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

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

D = m2 - (m1)^2 %дисперсия времени выхода в завершающий узел
```

Результат:

```
[ 1, 0, -q31, 0, 0, -w61]
[ -q12, 1, 0, 0, 0, 0]
[ 0, -q23, 1, 0, -q53, 0]
[ 0, -q24, -q34, 1, 0, 0]
[ 0, 0, 0, -q45, 1, 0]
[ 0, 0, 0, -q46, -q56, 1]

1 - q34*q45*q53 - q12*q24*q46*w61 - q12*q24*q31*q45*q53 - q12*q23*q34*q46*w61 -
q12*q24*q45*q56*w61 - q12*q23*q34*q45*q56*w61 - q12*q23*q31
- q12*q24*q46 - q12*q23*q34*q46 - q12*q24*q45*q56 - q12*q23*q34*q45*q56

1 - q34*q45*q53 - q12*q24*q31*q45*q53 - q12*q23*q31
-(q12*q24*q46 + q12*q23*q34*q46 + q12*q24*q45*q56 +
q12*q23*q34*q45*q56)/(q12*q23*q31 + q34*q45*q53 + q12*q24*q31*q45*q53 - 1)

We =
-(400*exp(4*s*(6*s + 25)) + 1080*exp(s*(25*s + 107)) + 10*exp((s*(73*s + 282))/2) +
27*exp((s*(75*s + 296))/2))/(63*exp(8*s*(5*s + 21)) + 270*exp((s*(41*s + 170))/2) +
```

```
3150*exp((s*(41*s + 180))/2) - 5000)

We0 =
1

m1 =
478000/1517

m2 =
373807370155/2301289

D =
145323370155/2301289
```

Вывод: в ходе данной лабораторной работы были получены навыки работы с вероятностными графами и их обработка с помощью методики GERT.

Глава 4. Поиск оптимальных параметров сети систем массового обслуживания

4.1 Постановка задачи

Задание по оптимизации многоканальной сети Джексона (алгоритм 2)

Задача 2, вариант 124.

$$\min\left(L(m) = \sum_{j=1}^{n} L_j \right)$$

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

Дано:

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

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

Необходимо сначала найти $\{\alpha_i\}$, потом найти $\min\left\{ \dfrac{\alpha_i}{\mu_i} \right\} = \dfrac{\alpha_s}{\mu_s}$ и далее выбрать $\lambda_0 \geq \dfrac{2\mu_s}{\alpha_s}$.

Принять M равным целой части удвоенного среднего числа заявок на обслуживании в сети: $M = \left[2\overline{n_{обсл}}\right] = \left[2\sum\limits_{j=1}^{n} \dfrac{\lambda_j}{\mu_j} \right]$, где

$[\cdot]$ — целая часть числа.

$\boldsymbol{Q} = \left\{ \boldsymbol{q_{ij}} \right\}_{\substack{i=\overline{0,n} \\ j=\overline{0,n}}}$:

0	0,1	0,3	0,3	0,3
0,2	0	0,2	0,2	0,4
0	0,5	0	0,4	0,1
0	0,6	0,2	0	0,2
0,1	0,2	0,3	0,4	0

$\vec{\boldsymbol{\mu}}$:

5	3	1	2

4.2 Решение

Решение поставленной задачи требует распределения (или перераспределения) обработчиков по узлам сети для оптимизации оценки производительности. Предположим, что между узлами распределяются в общей сложности M гомогенных обслуживающих механизмов (обработчиков). Таким образом, задача приобретает вид:

min (L(m) = $\sum_{j=1}^{n} L_j$)– целевая функция,

$\sum_{j=1}^{n} m_j = M$ – ограниченная.

Обозначим за $PI_i(m_i)$ индекс приоритета, определяемый как убыль показателя WIP (количество незавершенных работ) на обслуживающий механизм (обработчик) в узле j.

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

Применим следующий алгоритм:

1) Распределим $m_j = m_j^0$, j = 1..n (количество обработчиков по узлам),

$$m_0^j = \left\{ \left[\frac{\lambda_j}{\mu_j} \right] + 1 \right\} \geq 2$$

2) Каждая итерация пересчитывает показатель WIP L(m) и индекс приоритета $PI_j(m_j)$. В узел j* добавляется обработчик с наибольшим значением индекса PI_j :

$PI_j = \max \{PI_j(m_j), j = 1..n\}$

$L(m) = \sum_{j=1}^{n} v_j \, L_j m_j$

$$L_j\left(m_j, \lambda_j, \mu_j\right) = \frac{\frac{\lambda_j}{\mu_j m_j}\left(\frac{\lambda_j}{\mu_j}\right)^{m_j} \pi(0)}{\left(1-\frac{\lambda_j}{\mu_j m_j}\right)^2 m_j!} + \frac{\lambda_j}{\mu_j}$$

$\pi(0) = \left\{\sum_{t=0}^{m_j-1} \frac{\left(\frac{\lambda_j}{\mu_j}\right)^t}{t!} + \frac{\left(\frac{\lambda_j}{\mu_j}\right)^{m_j}}{\left(1-\frac{\lambda_j}{\mu_j m_j}\right) m_j!}\right\}^{-1}$ - вероятность, что система находится в положении равновесия в состоянии 0.

3) Алгоритм завершается после того, как суммарное количество выделенных обработчиков достигает предела M (допустимое решение).

Далее необходимо определить интенсивность на входи сети, на каждом узле и допустимое количество обработчиков.

С помощью известной матрицы переходных вероятностей и коэффициентов передачи узла найдем интенсивности: $\alpha_i = \frac{\lambda_j}{\lambda_0}$

Так как переходные вероятности однозначно определяют соотношение между интенсивностями потоков заявок, поступающих на входы элементов сети, можно составить систему линейных уравнений из n+1 уравнений и n неизвестных:

$$\begin{cases} q_{10}\alpha_1 + q_{20}\alpha_2 + \dots + q_{n0}\alpha_n = 1 - q_{00} \\ (q_{11} - 1)\alpha_1 + q_{21}\alpha_2 + \dots + q_{n1}\alpha_n = -q_{01} \\ \dots \\ q_{1n}\alpha_1 + q_{2n}\alpha_2 + \dots + (q_{nn} - 1)\alpha_n = -q_{0n} \end{cases}$$

Решим систему уравнений в Matlab следующим скриптом (исключим первое излишнее уравнение):

```
clc; clear

A = [-1 0.5 0.6 0.2;
0.2 -1 0.2 0.3;
0.2 0.44 -1 0.4;
0.4 0.1 0.2 -1];

B = [-0.1; -0.3; -0.3; -0.3];

a = A\B;
disp(a);
```

Результат:

```
4.0885
2.6763
3.4527
2.8936
```

С помощью найденных коэффициентов передачи узла определим интенсивность на входе сети. Для этого дополним предыдущий скрипт следующим образом:

```
clc; clear

A = [-1 0.5 0.6 0.2;
0.2 -1 0.2 0.3;
0.2 0.44 -1 0.4;
0.4 0.1 0.2 -1];

B = [-0.1; -0.3; -0.3; -0.3];

a = A\B;

intense = [5; 3; 1; 2];
idx = find(min(a./intense) == a./intense);
lam0 = 2*intense(idx)/a(idx);
disp(lam0);
```

Результат:

```
        2.4459
```

Округлим до целого значения вверх (3) и проверим условие

$$\min_j \left\{ \overline{n_{обсл\ j}} = \frac{\lambda_j}{\mu_j} \right\} \geq 2.$$

Для этого определим интенсивность на входе каждого узла сети с помощью решения систем линейных уравнений:

$$\begin{cases} (q_{00} - 1)\lambda_0 + q_{10}\lambda_1 + \cdots + q_{n0}\lambda_n = 0 \\ q_{01}\lambda_0 + (q_{11} - 1)\lambda_1 + \cdots + q_{n1}\lambda_n = 0 \\ \qquad\qquad\qquad \ldots \\ q_{0n}\lambda_0 + q_{1n}\lambda_1 + \cdots + (q_{nn} - 1)\lambda_n = 0 \end{cases}$$

В системе n+1 уравнений и n неизвестных (интенсивность на входе сети известна), поэтому первое уравнение рассматривать не будем:

```
clc; clear

A = [-1 0.5 0.6 0.2;
0.2 -1 0.2 0.3;
0.2 0.44 -1 0.4;
0.4 0.1 0.2 -1];

B = [-0.1; -0.3; -0.3; -0.3];

a = A\B;

intense = [5; 3; 1; 2];
idx = find(min(a./intense) == a./intense);
lam0 = 2*intense(idx)/a(idx);
disp(lam0);

lam0 = ceil(lam0);

b = lam0 * B;
lam = A\b;
disp(lam);
if(min(lam./intense) >= 2)
disp("Acepted");
else
disp("Denied");
return;
end
```

Результат:

```
2.4459

12.2654
8.0289
```

```
10.3581
8.6807

Acepted
```

Определим допустимое количество обработчиков как целую часть удвоенного среднего числа заявок на обслуживание в сети:

$$M = \left[2\overline{n_{\text{обсл } j}}\right] = \left[2\sum_{j=1}^{n}\frac{\lambda_j}{\mu_j}\right]$$

$$M = \left[2\sum_{j=1}^{n}\frac{\lambda_j}{\mu_j}\right] = \left[2\left(\frac{12.265}{5} + \frac{8.029}{3} + \frac{10.358}{1} + \frac{8.681}{2}\right)\right] = [39.656] = 39$$

Решение задачи оптимизации многоканальной сети Джексона:

```
clc; clear

A = [-1 0.5 0.6 0.2;
0.2 -1 0.2 0.3;
0.2 0.44 -1 0.4;
0.4 0.1 0.2 -1];

B = [-0.1; -0.3; -0.3; -0.3];

a = A\B;

intense = [5; 3; 1; 2];
idx = find(min(a./intense) == a./intense);
lam0 = 2*intense(idx)/a(idx);
disp(lam0);

lam0 = ceil(lam0);

b = lam0 * B;
lam = A\b;
disp(lam);
if(min(lam./intense) >= 2)
disp("Acepted");
else
disp("Denied");
return;
end

mj_0= round(lam./intense)+1;
% Вычисление предельного числа обработчиков в сети
M = round(2*sum(lam./intense));
L=zeros(1,4); % Среднее число заявок в узлах
```

```
L_next=zeros(1,4); % Среднее число заявок в узлах при увеличении обработчика в каждом узле
n=1;
while(sum(mj_0)~=M)
fprintf('\n--------------------------------------------------------------\n');
fprintf('Итерация %.0f\n', n);
for i=1:4
L(i)=wip(mj_0(i),lam(i),intense(i));
L_next(i)=wip(mj_0(i)+1,lam(i),intense(i));
end
fprintf('Lj(m): %.3f; %.3f; %.3f; %.3f\n',L);
% Индексы приоритета
PI = abs(L_next-L);
fprintf('PIj(mj): %.3f; %.3f; %.3f; %.3f\n',PI);
[PI_max,I]=max(PI);
% Увеличение числа обработчиков
mj_0(I)=mj_0(I)+1;
% Число итераций
n=n+1;
end
fprintf('\n--------------------------------------------------------------\n');
fprintf('mj_%.0f: %.0f; %.0f; %.0f; %.0f\n',n-1,mj_0);
fprintf('Использовано %.0f обработчиков из %.0f\n',sum(mj_0), M);

%% среднее число необслуженных заявок
function L = wip( m,lam,intense )
tmp = 0;
for i=1:m-1
tmp=tmp+(lam/intense)^i/factorial(i);
end
pi0=(tmp+((lam/intense)^m)/...
((1-lam/(m*intense))*factorial(m)))^(-1);
L=(lam/(m*intense)*(lam/intense)^m*pi0)/...
((1-lam/(intense*m))^2*factorial(m))+lam/intense;
end
```

В результате 17 итераций обнаружено, что оптимальное распределение обработчиков по узлам соответственно: 6; 7; 18; 9. Использовано 40 обработчиков из 40.

Вывод: изучен метод оптимизации структуры сети систем массового обслуживания, позволяющий определить оптимальное количество каналов для каждого узла сети.

Заключение

В работе рассмотрены некоторые математическое модели для решения задач выбора оптимального решения.

Решение задачи стохастического программирование привело к выводу, что увеличение доверительной вероятности приводит к ухудшению оценок получаемых решений.

При решение задачи на марковские модели был использован метод линейного программирования.

При решении задачи анализа потокового графа были использованы методика GERT и алгебра потоковых графов, выдавшие схожий результат.

Задача оптимизации системы массового обслуживания была произведена путем перераспределения мощностей в системе, что в итоге привело к понижению загруженности системы.

Список использованных источников

1. Д. Филлипс, А. Гарсиа-Диас. Методы анализа сетей: Пер. с англ. – М.: Мир, 1984.- 496 с.
2. Сиднев А.Г. Системный анализ. Часть 1 [Электронный ресурс] // Интернет-портал ИИТУ СПбПУ.2018.URL:http://intranet.ftk.spbstu.ru/docinfo.php?InfoFtkDocumentID=1386947 (дата обращения: 26.05.2018).
3. Горбунов В.М. Теория принятия решений: учебное пособие. Томск: Изд-во Томск. Политех. ун-та, 2010.
4. Таха Х.А. Введение в исследование операций: Пер. с англ. 7-е изд. М.: Издательский дом «Вильямс», 2005.