

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

ОТЧЕТ
по летней практике

Оптимизация ССМО

(тема работы)

Работу выполнил:

43501.3

Волкова М.Д.

группа

Ф.И.О.

Преподаватель:

Сиднеев А.Г.

подпись

Ф.И.О.

Санкт-Петербург
2017

1. Постановка задачи

Написать программу Matlab для оптимизации однородной экспоненциальной замкнутой многоканальной ССМО.

$$\max \lambda = \omega_1 G_M(N-1)/G_M(N)$$

при ограничении

$$S = \sum_{i=1}^M c_i \mu_i^{\alpha_i} = S^*, \quad \mu > 0.$$

Дано: $\{S^*, M, N, p=\{p_{ij}\}_{i=0,M,j=0,M}, \vec{c}=(c_1, c_2, \dots, c_M), \vec{\alpha}=(\alpha_1, \alpha_2, \dots, \alpha_M), m=(m_1, m_2, \dots, m_M)\}$

Где N-число заявок в сети, M - число узлов, S* - стоимость сети,

$\vec{c}=(c_1, c_2, \dots, c_M)$ - вектор, определяющий число каналов в узле.

$\vec{\alpha}=(\alpha_1, \alpha_2, \dots, \alpha_M)$ — вектор, определяющий коэффициенты важности узлов. Коэффициенты важности α_i узлов ССМО входят в формулу расчета её стоимости.

2. Решение

2.1. Вычисление вероятностей ω_i

Для этого решается система уравнений:

$$\omega_j = \sum_{i=1}^M \omega_i * p_{ij}, \quad j = 1..M$$
$$\sum_{i=1}^M \omega_i = 1$$

Скрипт для решения системы:

```
w = fsolve(@wfun, [1;0;0;0]);
sum(w);

function F = wfun(w)
    for j = 1:M
        er = 0;
        for i = 1:M
            er = er + w(i)*p(i,j);
        end
        F(j) = er - w(j);
    end
    F = [F(1); F(2); F(3); F(4) ; sum(w) - 1];
end
```

2.2. Рекуррентная процедура вычисления времени и маргинального распределения числа заявок в сетях

$$\text{Шаг 1. } \bar{t}_i(r) = \sum_{n=1}^r \frac{n}{\mu_i(n)} P_i(n-1, r-1), \quad i = \overline{1, M} \quad (12.51)$$

$$\text{Шаг 2. } \lambda_1(r) = \frac{r}{\sum_{j=1}^M \frac{\omega_j}{\omega_1} \bar{t}_j(r)} \quad (12.52)$$

Шаг 3. Для каждого i -го узла сети, $i = \overline{1, M}$, определить набор вероятностей $P_i(n, r)$, $n = \overline{0, r}$:

$$\begin{cases} P_i(n, r) = \frac{\omega_i \lambda_1(r)}{\omega_i \mu_i(n)} P_i(n-1, r-1), & n = \overline{1, r}, \\ P_i(0, r) = 1 - \sum_{n=1}^r P_i(n, r), \end{cases} \quad (12.53)$$

$$\text{где } \mu_i(n) = \begin{cases} n \mu_i, & n < m_i, \\ m_i \mu_i, & n \geq m_i, \end{cases} \quad (12.54)$$

где m_i — число каналов в i -м узле.

Шаг 4.

Если $r = N$, Конеч,

Начальные условия: $P_i(0,0)=1, \quad i=\overline{1,M}; r=1$

Эта процедура и будет нашей оптимизируемой функцией.

Оптимизируемая функция на языке Matlab:

```
function [ lambda ] = findlambda(w,u)
    for i = 1:M
        P_old(i,1) = 1;
        u(i) = u(i)^a(i);
    end

    for r = 1:N
        % step 1
        for i = 1:M
            sum_1 = 0;
            for n = 1:r
                if (n >= m(i))
                    u(i) = m(i)*u(i);
                else
                    u(i) = n*u(i);
                end

                sum_1 = sum_1 + (n/u(i))*P_old(i,n);
            end
            t(i) = sum_1;
        end

        % step 2
        s=0;
        for i = 1:M
            s = s + (w(i)*t(i))/w(1);
        end
        lambda = r/s;

        % step 3
        for i = 1:M
            if (n >= m(i))
                u(i) = m(i)*u(i);
            else
                u(i) = n*u(i);
            end

            for n = 1:r
                P(i,n+1) = ((w(i)*lambda)/(w(1)*u(i)))*P_old(i,n);
            end

            sum = 0;
            for n = 1:r
                sum = sum + P(i,n);
            end

            P(i,1) = 1 - sum;
        end
        P_old = P;
    end
end
```

2.2. Решение оптимизационной задачи:

Решим задачу максимизации λ_1 с помощью средств Matlab.

Для этого потребуется составить функцию зависимости λ_1 от μ . Используется рекуррентная процедура описанная в прошлом пункте.

Для решения оптимизационной задачи, используем функцию Fmincon. Fmincon находит минимум функции с ограничениями:

$$\min_x f(x) \text{ such that } \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub, \end{cases}$$

Скрипт для решения задачи максимизации в Matlab:

```
for i = 1:M
    lb(i,1) = 0;
end

fun = @(x)(-findlambda(w,x));
[my_u,fval] = fmincon(fun,w,[],[],c,S,lb,[],
[],optimoptions('fmincon','Algorithm','sqp'))
```

В Приложении 1-2 находится полный код программы с решениями для некоторых ССМО.

Результаты выполнения программы приведены в таблице 1:

Таблица 1.		
Начальные данные	Lambda(max)	u
<p>N = 7; M = 4; S = 4;</p> <p>p = [0 0.1 0.3 0.6;... 0.2 0 0.2 0.6;... 0.4 0.1 0 0.5;... 0.3 0.2 0.5 0];</p> <p>m = [2;3;2;1]; c = [2 3 4 2]; a = [1;1;1;1];</p>	1.8912	<p>0.1050 0.0224 0.0784 1.7046</p>
<p>N = 8; M = 5; S = 9;</p> <p>p = [0 0.1 0.3 0.2 0.4;... 0.2 0 0.2 0.3 0.3;... 0.4 0.1 0 0.3 0.2;... 0.2 0.1 0.2 0 0.5;... 0.3 0.2 0.4 0.1 0];</p> <p>m = [1;1;1;1;1]; c = [1 1 1 1 1]; a = [1;1;1;1;1];</p>	0.4916	<p>2.3402 1.6292 1.3993 2.0711 1.5602</p>
<p>S = 8; N = 4; M = 3;</p> <p>p = [0 0.1 0.3;... 0.2 0 0.2;... 0.4 0.1 0];</p> <p>m = [5;4;7]; c = [4 2 3]; a = [2;3;4];</p>	85.1458	<p>0.7782 0.9406 1.0020</p>
<p>S = 7; N = 4; M = 4;</p> <p>p = [0 0.2 0.3 0.5;... 0.2 0 0.2 0.6;... 0.4 0.1 0 0.5;... 0.3 0.2 0.5 0];</p> <p>m = [8;2;2;2]; c = [2 8 2 2]; a = [2;2;8;2];</p>	10.6237	<p>0.4300 0.3487 0.9912 0.6839</p>

Приложение 1. Код программы:

```
function test_2
S = 8;
N = 4;      % число заявок в сети
M = 3;      % число узлов

p = [0    0.1  0.3;...
     0.2  0    0.2;...
     0.4  0.1  0   ];

m = [5;4;7]; % число каналов в i-м узле
c = [4 2 3]; % стоимостные коэффициенты
a = [2;3;4]; % коэффициент нелинейности

%% нахождение вероятностей w

for j = 1:M
    test(j,1) = 0;
end
test(1,1) = 1;

w = fsolve(@wfun,test);

function F = wfun(w)
    for j = 1:M
        sum_t = 0;
        for i = 1:M
            sum_t = sum_t + w(i)*p(i,j);
        end
        F(j) = sum_t - w(j);
    end
    F = [F(1); F(2); F(3); sum(w) - 1];
end

u = w;

%% find lambda
function [ lambda ] = findlambda(w,u)
    for i = 1:M
        P_old(i,1) = 1;
        u(i) = u(i)^a(i);
    end

    for r = 1:N
        % step 1
        for i = 1:M
            sum_1 = 0;
            for n = 1:r
                if (n >= m(i))
                    u(i) = m(i)*u(i);
                else
                    u(i) = n*u(i);
                end

                sum_1 = sum_1 + (n/u(i))*P_old(i,n);
            end
            t(i) = sum_1;
        end
    end
```

```

% step 2
s=0;
for i = 1:M
    s = s + (w(i)*t(i))/w(1);
end
lambda = r/s;

% step 3
for i = 1:M
    if (n >= m(i))
        u(i) = m(i)*u(i);
    else
        u(i) = n*u(i);
    end

    for n = 1:r
        P(i,n+1) = ((w(i)*lambda)/(w(1)*u(i)))*P_old(i,n);
    end

    sum = 0;
    for n = 1:r
        sum = sum + P(i,n);
    end

    P(i,1) = 1 - sum;

end
P_old = P;
end
end

%% ОПТИМИЗАЦИЯ
for i = 1:M
    lb(i,1) = 0;
end

fun = @(x) (-findlambda(w,x));
% минимизация с оптимизационными параметрами, определенными в структурной
опции
[my_u,fval] = fmincon(fun,w,[],[],c,S,lb,[],
[],optimoptions('fmincon','Algorithm','sqp'))
end

```


Приложение 2. Код программы:

```
function example_2
S = 7;
N = 4;
M = 4;

p = [0    0.2  0.3  0.5;...
     0.2  0    0.2  0.6;...
     0.4  0.1  0    0.5;...
     0.3  0.2  0.5  0];

m = [8;2;2;2];
c = [2 8 2 2];
a = [2;2;8;2];

for j = 1:M
    test(j,1) = 0;
end
test(1,1) = 1;

w = fsolve(@wfun,test);

function F = wfun(w)
    for j = 1:M
        sum_t = 0;
        for i = 1:M
            sum_t = sum_t + w(i)*p(i,j);
        end
        F(j) = sum_t - w(j);
    end
    F = [F(1); F(2); F(3); F(4); sum(w) - 1];
end

u = w;

%% find lambda
function [ lambda ] = findlambda(w,u)
    for i = 1:M
        P_old(i,1) = 1;
        u(i) = u(i)^a(i);
    end

    for r = 1:N
        % step 1
        for i = 1:M
            sum_1 = 0;
            for n = 1:r
                if (n >= m(i))
                    u(i) = m(i)*u(i);
                else
                    u(i) = n*u(i);
                end

                sum_1 = sum_1 + (n/u(i))*P_old(i,n);
            end
            t(i) = sum_1;
        end

        % step 2
```

```

s=0;
for i = 1:M
    s = s + (w(i)*t(i))/w(1);
end
lambda = r/s;

% step 3
for i = 1:M
    if (n >= m(i))
        u(i) = m(i)*u(i);
    else
        u(i) = n*u(i);
    end

    for n = 1:r
        P(i,n+1) = ((w(i)*lambda)/(w(1)*u(i)))*P_old(i,n);
    end

    sum = 0;
    for n = 1:r
        sum = sum + P(i,n);
    end

    P(i,1) = 1 - sum;

end
P_old = P;
end
end

for i = 1:M
    lb(i,1) = 0;
end

fun = @(x) (-findlambda(w,x));
[my_u,fval] = fmincon(fun,w,[],[],c,S,lb,[],
[],optimoptions('fmincon','Algorithm','sqp'))

end

```