

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:

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

Скрипт для решения системы:

```
w = fsolve(@wfun, [1;0;0;0]);  
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. Расчёт нормирующей константы многоканальной замкнутой сети

Нормирующую константу многоканальной СМО находим по следующей формуле:

$$G_r(k) = \sum_{l=0}^k Z_r(l) G_{r-1}(k-l).$$

Скрипт для нахождения нормирующей константы:

```
for r = 2:1:M  
    for k = 1:1:N  
        sum = 0;  
        for l = 0:1:k  
            sum = sum + z(r, l + 1)*G(r - 1, k - l + 1);  
        end  
        G(r, k + 1) = sum;  
    end  
end
```

, где

$$Z_i(n_i) = \frac{\omega_i^{n_i}}{\prod_{j=1}^{n_i} \mu_i(j)}, \text{ а } \mu_i(j) = \begin{cases} j \mu_i, & j < m_i \\ m_i \mu_i, & j \geq m_i \end{cases},$$

Соответствующий скрипт:

```
for i = 1:1:M
    for n = 0:1:N
        my = 1;
        for j = 1:1:n
            if (j >= m(i))
                my = my*m(i)*u(i);
            else my = my*j*u(i);
            end
        end
        z(i,n+1) = (w(i)^n)/my;
    end
end
```

2.3. Нахождение интенсивности на выходе 1 узла однородной замкнутой сети СМО

Находим интенсивность по следующей формуле:

$$\lambda_i(N) = \frac{\omega_i G_M(N-1)}{G_M(N)}$$

Реализация в Matlab:

```
lambda = G(M, N - 1) / G(M, N);
```

2.4. Функция оптимизации

Оптимизируем нашу функцию с помощью функцией `fmincon`

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

с следующим нелинейным ограничением:

```
function [ctmp , ceqtmp] = limitation(x)  
ctmp = 0;  
    for i = 1:M  
        ctmp = ctmp + c(i)*x(i)^a(i);  
    end  
    ctmp = ctmp - S;  
    ceqtmp = [];  
end
```

3. Результаты

Результаты программы приведены в Таблице 1.

Параметры сети / Начальные значения		Результат	
<pre> S = 8; N = 6; M = 4; 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]; m = [4;4;4;4]; c = [2 2 2 2]; a = [1;1;1;1]; </pre>		my_u =	
			<pre> 0.9335 0.6641 1.1237 1.2786 </pre>
		fval =	
			-4.9331
<pre> S = 8; N = 6; M = 3; p = [0 0.1 0.3;... 0.2 0 0.2;... 0.3 0.2 0.5]; m = [4;4;4;4]; c = [2 2 2 2]; a = [1;1;1;1]; </pre>		my_u =	
			<pre> 1.0906 0.8944 2.0150 </pre>
		fval =	
			-7.1560
<pre> S = 10; N = 7; M = 3; p = [0 0.1 0.3;... 0.2 0 0.2;... 0.3 0.2 0.5]; m = [4;3;3;3]; c = [3 3 3 2]; a = [3;1;2;1]; </pre>		my_u =	
			<pre> 0.7484 0.8152 1.4488 </pre>
		fval =	
			-5.7744

4. Приложение

```
function test
S = 10;
N = 7;
M = 3;
p = [0    0.1  0.3;...
     0.2  0    0.2;...
     0.3  0.2  0.5];
m = [4;3;3;3];
c = [3 3 3 2];
a = [3;1;2;1];

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);
        my = j;
    end

    for j=my:10
        F(j) = 0;
    end
    F = [F(1); F(2); F(3); F(4); F(5); F(6); F(7); F(8); F(9);
F(10); sum(w) - 1];
end
function [ lambda ] = findlambda(w,u)

for i = 1:1:M
    for n = 0:1:N
        my = 1;
        for j = 1:1:n
            if (j >= m(i))
                my = my*m(i)*u(i);
            else my = my*j*u(i);
            end
        end
        z(i,n+1) = (w(i)^n)/my;
    end
end
G(1, :) = z(1, :);

for r = 2:1:M
    for k = 1:1:N
        sum = 0;
        for l = 0:1:k
            sum = sum + z(r, l + 1)*G(r - 1, k - l + 1);
        end
        G(r, k + 1) = sum;
    end
end
end
```

```

lambda = 0;
lambda = G(M, N - 1)/G(M,N);
end

function [ctmp , ceqtmp] = limitation(x)
ctmp = 0;
    for i = 1:M
        ctmp = ctmp + c(i)*x(i)^a(i);
    end
    ctmp = ctmp - S;
    ceqtmp = [];
end

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

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

```