Санкт-Петербургский Политехнический Университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Отчёт по расчетному заданию №3**

Поиск оптимальных параметров сети систем массового обслуживания

по курсу

Методы оптимизации

Работу выполнил студент группы № 13541/1 \_\_\_\_\_\_\_\_Гагаркин И.Ю.

Работу принял преподаватель \_\_\_\_\_\_\_\_\_Сиднев А.Г.

Санкт-Петербург

2018

# Задание

**Задача 5**

Минимизировать стоимость ССМО при ограничении на среднее число заявок в сети







Дано многоканальная сеть Джексона:



 — предельное число заявок в сети.

|  |  |  |  |
| --- | --- | --- | --- |
| **№** |  |  |  |
| **155** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **0** | **0,5** | **0** | **0,3** | **0,2** | | **0,2** | **0** | **0,2** | **0,2** | **0,4** | | **0,3** | **0,3** | **0** | **0,3** | **0,1** | | **0,3** | **0,4** | **0,1** | **0** | **0,1** | | **0** | **0,2** | **0,5** | **0,3** | **0** | | **7** | **6** |

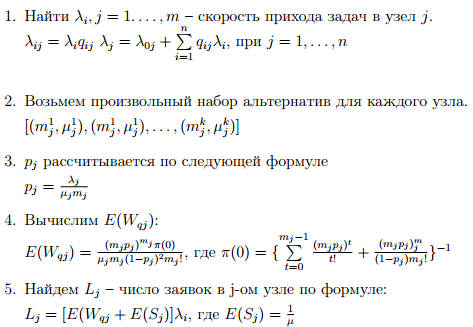
Самостоятельно сформировать набор альтернатив (по 2 альтернативы на каждый узел, обеспечивающих установивший режим в узле).

Решить задачу 5 двумя способами:

1. В соответствии с алгоритмом 5

2. Как задачу дискретного линейного программирования.

**Алгоритм:**

****

# Решение по алгоритму

***Первый шаг алгоритма***

По первой формуле из первого шага алгоритма найдем вектор .

= (3,5; 0; 2,1; 1,4)

Используя вектор , составим систему уравнений по второй формуле из шага 1.

В результате решения системы методом Гаусса был получен следующий результат:

Вычислим остальные 𝜆𝑖𝑗 , используя следующий скрипт. Здесь и в дальнейшем используется ограничение на предельное число заявок в сети – 4 шт. вместе с нулевой.

|  |
| --- |
| Clc  % Исходные данные  Q = [  0 0.5 0 0.3 0.2;  0.2 0 0.2 0.2 0.4;  0.3 0.3 0 0.3 0.1;  0.3 0.4 0.1 0 0.1;  0 0.2 0.5 0.3 0];  % Значение лямбды каждого узла  lambdaj=[7;19.75;12.3;15.9;13.52];  N=4; % Число узлов вместе с нулевой.  % Вычисление лямбда ij  lambdaij=[];  for i = 1:N  for j = 1:N  lambdaij(i,j) = lambdaj(i)\*Q(i, j);  end  end  lambdaij |

Результаты вычисления представлены в таблице 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i/j | 1 | 2 | 3 | 4 |
| 1 | 0 | 3.5 | 0 | 2.1 |
| 2 | 3.95 | 0 | 3.95 | 3.95 |
| 3 | 3.69 | 3.69 | 0 | 3.69 |
| 4 | 4.77 | 6.36 | 1.59 | 0 |

Таблица 1. Результат вычисления всех значений 𝜆ij

***Второй шаг алгоритма***

Далее сформируем набор альтернатив (по 2 альтернативы на каждый узел):

***Третий шаг алгоритма***

Для расчета коэффициента загрузки 𝑝𝑗 воспользуемся следующим скриптом:

|  |
| --- |
| K=2; % Задаем альтернативы  muj = [ 34 8 11 4;  5 23 7 9];  mj = [ 26 8 3 6;  9 13 10 4];  % Расчёт вероятностей  pj = [];  for i = 1:N  for k = 1:K  pj(k,i) = lambdaj(i+1)/(muj(k,i)\*mj(k,i));  end  end  pj |

Результаты занесены в таблицу 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| k\j | 1 | 2 | 3 | 4 |
| 1 | 0.22 | 0.2 | 0.48 | 0.56 |
| 2 | 0.44 | 0.41 | 0.23 | 0.37 |

Таблица 2. Результат расчета коэффициентов загрузки.

***Четвертый шаг алгоритма***

Расчет вектора 𝜋𝑗 (0).

|  |
| --- |
| %% Step 4  % Расчет начальной вероятности для каждого узла.  pij0 = [];  for i = 1:N  for k=1:K  sum = 0;  for t = 0:mj(k,i)-1  sum = sum + ((mj(k,i)\*pj(k,i))^t)/factorial(t);  end  sum = sum + ((mj(k,i)\*pj(k,i))^mj(k,i))/  ((1-pj(k,i))\*factorial(mj(k,i)));  pij0(k,i)=sum^(-1);  end  end  pij0 |

Результаты расчета начальной вероятности узлов занесены в таблицу 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| k\j | 1 | 2 | 3 | 4 |
| 1 | 0.5594 | 0.2149 | 0.2241 | 0.0329 |
| 2 | 0.0192 | 0.5858 | 0.1032 | 0.2205 |

Таблица 3. Расчет начальных коэффициентов загрузки.

***Пятый шаг алгоритма***

Рассчитаем 𝐸(𝑊𝑞𝑗 ).

|  |
| --- |
| % E(Wqj) для каждого узла.  EWqj = [];  for i = 1:N  for k=1:K  EWqj(k,i) = (((mj(k,i)\*pj(k,i))^mj(k,i))\*pij0(k,i))/(muj(k,i)\*mj(k,i)\*((1-pj(k,i))^2)\*factorial(mj(k,i)));  end  end |

Результаты занесены в таблицу 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| k\j | 1 | 2 | 3 | 4 |
| 1 | 0.0000 | 0.0000 | 0.0127 | 0.0149 |
| 2 | 0.0009 | 0.0000 | 0.0000 | 0.0033 |

Таблица 4. Расчет вектора 𝐸(𝑊𝑞𝑗 )

Теперь найдем 𝐸(𝑆𝑗 ) для каждого узла.

|  |
| --- |
| % Найдем E(sj) для каждого узла.  Esj = [];  for i = 1:N  for k=1:K  Esj(k,i) = 1/muj(k,i);  end  end  Esj |

Результаты занесены в таблицу 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| k\j | 1 | 2 | 3 | 4 |
| 1 | 0.0294 | 0.1250 | 0.0909 | 0.2500 |
| 2 | 0.2000 | 0.0435 | 0.1429 | 0.1111 |

Таблица 5. Расчет 𝐸(𝑆𝑗 ) для каждого узла

Теперь не составит труда посчитать 𝐿𝑗 – число заявок в j-ом узле, для каждой альтернативы.

|  |
| --- |
| % Найдем Lj  Lj = [];  for i = 1:N  for k=1:K  Lj(k,i) = lambdaj(i+1)\*(EWqj(k,i)+Esj(k,i));  end  end  Lj |

Результаты занесены в таблицу 6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| k\j | 1 | 2 | 3 | 4 |
| 1 | 0.5809 | 1.5375 | 1.6479 | 3.5814 |
| 2 | 3.9673 | 0.5348 | 2.2715 | 1.5473 |

Таблица 6. Расчет числа заявок в каждом узле

# Решение дискретным линейным методом программирования

Для решения задачи воспользуемся функцией linprog.

|  |
| --- |
| %Зададим входные параметры  A = [1 0 0 0 1 0 0 0;  0 1 0 0 0 1 0 0;  0 0 1 0 0 0 1 0;  0 0 0 1 0 0 0 1;  Lj(1,1) Lj(1,2) Lj(1,3) Lj(1,4) Lj(2,1) Lj(2,2) Lj(2,3) Lj(2,4)];  f = [170 184 77 36 234 104 30 24];  b = [1 1 1 1 6];  Aeq = [1 0 0 0 1 0 0 0;  0 1 0 0 0 1 0 0;  0 0 1 0 0 0 1 0;  0 0 0 1 0 0 0 1];  beq = [1; 1; 1; 1];  %дополнительно введем ограничения на значение lb и ub  lb=zeros(8,1);  ub=ones(8,1);  %решаем задачу  [x,fval] = linprog(f,A,b,Aeq,beq,lb,ub) |

Результаты занесены в таблицу 7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| k |  |  |  |  |
| 1 | 1 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 1 |

Таблица 7. Результат использования linprog.

Получаем следующий выбор альтернатив: (𝑢11, 𝑢21, 𝑢31, u42).