МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.
ЧЕРНЫШЕВСКОГО»

Отчёт по дисциплине
«Модели и методы теории массового обслуживания»

Задание №3

студентки 4 курса 481 группы
направления 27.03.03 – Системный анализ и управление
факультета компьютерных наук и информационных технологий
Гурковой Виктории Марковны

Преподаватель ________________ Е.С. Рогачко

Саратов 2020 г.

Задание 3. Анализ системы массового обслуживания M $_N$ / GI $_N$ / 1 с абсолютными приоритетами с дообслуживанием требований.

Задача. На вычислительную машину поступает пять типов потоков информации, обслуживаемых в соответствии с дисциплиной «абсолютный приоритет с дообслуживанием». Стоимость ожидания в единицу времени (стоимостные коэффициенты) для информации из каждого потока равна соответственно: 5; 4; 3; 2; 1.

Вычислить: среднее полное время обработки информации каждого типа; среднюю продолжительность ожидания обработки информации каждого типа; среднюю продолжительность ожидания обработки информации из общего потока.

При обработке поступающей информации управляющая программа осуществляет выбор порядка обслуживания информации. Требуется: а) определить, является ли оптимальным текущий порядок приоритетов среди потоков информации; б) назначить оптимальный порядок приоритетов среди потоков; в) сравнить результаты оптимального упорядочивания приоритетов для случаев абсолютных и относительных приоритетов, а также для бесприоритетного обслуживания (обслуживания в порядке поступления). Пояснить результаты.

Вариант 1. Каждый поток информации является пуассоновским с интенсивностью: 1; 2; 0,5; 1,1 и 3 соответственно. Время обработки поступающей информации имеет распределение Эрланга порядка 2 со средним значением соответственно: 0,2; 0,1; 0,1; 0,15; 0,01.

Метод анализа системы M $_N$ / GI $_N$ / 1 с абсолютными приоритетами.

Пусть в системе $M_N/GI_N/1$ потоки занумерованы в порядке убывания их значимости. Будем говорить, что в системе действуют абсолютные приоритеты, если поступающее в СМО i-требование замещает на приборе j-требование при $i < j$. Рассмотрим случай абсолютных приоритетов с дообслуживанием прерванных требований, при котором прерванное j-требование возвращается в начало j-очереди и вновь поступает на прибор, когда пусты очереди с номерами от 1 до $j-1$. Поскольку дообслуживание требования всегда продолжается с прерванного места, то чистое время его обслуживания не зависит от числа прерываний.

Для стационарной системы с коэффициентом использования прибора $\boldsymbol{R_N} < 1$ найдем средние характеристики через исходные параметры.

Среднее время пребывания j-требования с j-приоритетом в системе $\boldsymbol{u_j}$ складывается из среднего времени ожидания начала обслуживания w_j, среднего времени чистого обслуживания v_j и среднего времени всех перерывов за время обслуживания z_j:

$$\boldsymbol{u_j} = w_j + v_j + z_j, \boldsymbol{j} = \overline{\mathbf{1}, \boldsymbol{N}}.$$

Среднее время от начала обслуживания j-требования до завершения будем называть *средним полным временем обслуживания*. Оно равно $\theta_j = v_j + z_j$.

Начнем с вычисления z_j. В среднем за время обслуживания j-требования произойдет $\Lambda_{j-1} v_j$ прерываний, где $\Lambda_{j-1} = \sum_{i=1}^{j-1} \lambda_i$ – интенсивность суммарного потока требований, имеющего право прерывать j-требования. Каждое такое прерывание – это период занятости в системе с первыми j–1 потоками требований. Поскольку в такой системе коэффициент использования прибора равен $\boldsymbol{R_{j-1}} = \sum_{i=1}^{j-1} \psi_i$, а среднее время обслуживания требования из суммарного потока

$$\sum_{i=1}^{j-1} \frac{\lambda_i}{\Lambda_{j-1}} v_i = \frac{\boldsymbol{R_{j-1}}}{\Lambda_{j-1}},$$

то в соответствии с пунктом 1.2 среднее время одного прерывания

$$\pi_{j-1} = \frac{\boldsymbol{R_{j-1}}}{\Lambda_{j-1}(\mathbf{1} - \boldsymbol{R_{j-1}})}.$$

Теперь среднее время всех перерывов в обслуживании j-требования

$$\boldsymbol{z_j} = \Lambda_{j-1} v_j \pi_{j-1} = v_j \frac{\boldsymbol{R_{j-1}}}{\mathbf{1} - \boldsymbol{R_{j-1}}}, \boldsymbol{j} = \overline{\mathbf{2}, \boldsymbol{N}},$$

а среднее полное время обслуживания j-требования

$$\theta_j = v_j + z_j = v_j / (\mathbf{1} - \boldsymbol{R_{j-1}}), \boldsymbol{j} = \overline{\mathbf{1}, \boldsymbol{N}}. \qquad (2.11)$$

Вычислим w_j. Перед тем как поступившее в систему j-требование первый раз окажется на приборе, должны быть выполнены следующие действия:

а) обслужено или замещено обслуживаемое требование;

б) дообслужены прерванные требования из потоков от 1 до j;

в) обслужены наличные очереди с номерами от 1 до j;

г) обслужены вновь поступающие за время ожидания приоритетные требования из потоков от 1 до j–1.

Для средних длительностей указанных событий имеем уравнение

$$w_j = \boldsymbol{\sigma_j} + \boldsymbol{\eta_j} + \sum_{i=1}^{j} w_i \lambda_i v_i + w_j \sum_{i=1}^{j-1} \lambda_i v_i. \qquad (2.12)$$

Здесь $\boldsymbol{\sigma_j}$ – среднее время дообслуживания требования прибором в присутствии j-требования: $\boldsymbol{\sigma_j} = \sum_{i=1}^{j} \boldsymbol{\psi_i \Delta_i}$, $\boldsymbol{\Delta_i} = \boldsymbol{v_i^{(2)}}/2v_i$; $\boldsymbol{\eta_j}$ – среднее время дообслуживания прерванных требований из очередей с номерами от 1 до j. Заметим, что в каждой очереди может быть не более одного прерванного требования. По аналогии со средним полным временем обслуживания θ_j введем *среднее полное время дообслуживания* $\boldsymbol{\delta_j} = \boldsymbol{\sigma_j} + \boldsymbol{\eta_j}$.

Из (2.12) получаем рекуррентное соотношение

$$w_j = \frac{\sigma_j + \eta_j + \sum_{i=1}^{j-1} \psi_i w_i}{1 - R_j}. \qquad (2.13)$$

Непосредственно из него для первого потока имеем

$$w_1 = \boldsymbol{\sigma_1}/(\mathbf{1} - \boldsymbol{R_1}).$$

Предположим, что для всех потоков от 1 до j–1 справедливо решение

$$w_i = \frac{\sigma_i}{(1-R_{i-1})(1-R_i)} = \frac{\frac{1}{2}\sum_{k=1}^{i} \lambda_k v_k^{(2)}}{(1-R_{i-1})(1-R_i)}. \qquad (2.14)$$

Убедимся подстановкой в (2.13), что это решение справедливо и для j:

$$w_j = \left(\boldsymbol{\sigma_j} + \boldsymbol{\eta_j} + \sum_{i=1}^{j-1} \frac{\psi_i \sigma_i}{(1-R_{i-1})(1-R_i)}\right)/(\mathbf{1} - \boldsymbol{R_j}). \qquad (2.15)$$

В последнем уравнении осталось вычислить $\boldsymbol{\eta_j}$.

$$\begin{aligned}\boldsymbol{\eta_j} &= \sum_{k=2}^{j} \sum_{i=1}^{k-1} P_{ik} \Delta_k = \sum_{i=1}^{j-1} \sum_{k=i+1}^{j} P_{ik} \Delta_k = \\ &= \sum_{i=1}^{j-1} \frac{\psi_i(\sigma_j - \sigma_i)}{(1-R_{i-1})(1-R_i)} = \\ &= \sigma_j \frac{R_{j-1}}{1-R_{j-1}} - \sum_{i=1}^{j-1} \frac{\psi_i \sigma_i}{(1-R_{i-1})(1-R_i)}. \qquad (2.16)\end{aligned}$$

$$\boldsymbol{P_{ik} = \frac{\lambda_k v_k}{1 - R_i} - \frac{\lambda_k v_k}{1 - R_{i-1}} = \frac{\psi_i \psi_k}{(1 - R_{i-1})(1 - R_i)}}.$$

Подставив теперь (2.16) в (2.15), убеждаемся, что для w_j справедливо решение в форме (2.14), $\boldsymbol{j = \overline{1, N}}$.

Оптимизация относительных приоритетов.

Пусть в одноприборную систему с ожиданием поступают N пуассоновских потоков требований. Длительность обслуживания требований -го потока имеет произвольное распределение $B_i(t)$ со средним v_i. За единицу времени ожидания (пребывания в системе) требования i-го потока платится штраф c_i. Требуется назначить приоритеты так, чтобы суммарный штраф за единицу времени функционирования системы в стационарном режиме был минимальным, т. е. минимизировать величину

$$F = \sum_{i=1}^{N} c_i \lambda_i w_i,$$

где λ_i – интенсивность i-го потока, а w_i – среднее время ожидания (пребывания в системе) требований i-го потока.

Было показано, что приоритеты потоков должны быть тем выше, чем больше величина c_i/v_i , и этот факт имеет место в классе относительных приоритетов при произвольных $B_i(t)$ и в классе абсолютных приоритетов при $B_i(t) = 1 - e^{-t/v_i}$ (экспоненциальное распределение).

Последний результат может быть уточнен. Действительно, поскольку явный вид выражения для F зависит лишь от первых двух моментов распределений $B_i(t)$, то и ограничения должны быть наложены не на вид $B_i(t)$, а на соотношение между их первыми и вторыми моментами. Приведем эти соотношения.

Последовательность приоритетов, упорядоченная в соответствии со сказанным выше, будет оптимальна в классе абсолютных приоритетов, если дополнительно выполнены неравенства

$$\frac{2v_i c_i}{v_i^{(2)}} > \frac{c_j}{v_j}, \; i < j; \qquad \frac{2v_i c_i}{v_i^{(2)}} < \frac{c_j}{v_j}, \; i > j,$$

где $v_i^{(2)}$ – второй момент распределения $B_i(t)$.

Таким образом, если потокам приписаны приоритеты в порядке убывания отношения c_i/v_i и при этом выполняются вышеприведенные неравенства, то такая последовательность приоритетов оптимальна в классе абсолютных приоритетов, и она лучше оптимальных в классе относительных приоритетов.

Разработанная программа.

```
N = 5
k = 2
# начальные условия
# lambdas = [1, 2, 0.5, 1.1, 3]
# v = [0.2, 0.1, 0.1, 0.15, 0.01]
# c = [5, 4, 3, 2, 1]

# условия при оптимальном приоритете 5, 2, 3, 1, 4
lambdas = [3, 2, 0.5, 1, 1.1]
v = [0.01, 0.1, 0.1, 0.2, 0.15]
c = [1, 4, 3, 5, 2]

mu = [k / v[i] for i in range(N)]
print("mu =", mu)

disp = [k / mu[i] ** 2 for i in range(N)]
print("disp =", disp)

alpha_2 = [disp[i] + v[i] ** 2 for i in range(N)]
print("alpha_2 =", alpha_2)

psi = [lambdas[i] * v[i] for i in range(N)]

R = []
summa = 0
for j in range(N):
    summa += psi[j]
    R.append(summa)
print("R =", R)

# среднее полное время обслуживания j-го типа информации
theta = [v[j] / (1 - R[j - 1]) for j in range(N)]
theta[0] = v[0]
print("theta =", theta)

delta = [alpha_2[i] / (2 * v[i]) for i in range(N)]
```

```
sigma = []
summa = 0
for i in range(N):
    summa += psi[i] * delta[i]
    sigma.append(summa)

# средняя продолжительность ожидания обработки инф. каждого типа
w = []
w.append(sigma[0] / (1 - R[0]))
for i in range(1, N):
        w.append(sigma[i] / ((1 - R[i - 1]) * (1 - R[i])))
print("w =", w)

# средняя продолжительность ожидания обработки инф. из общего потока
w_random = sum([w[i] * lambdas[i] / sum(lambdas) for i in range(N)])
print("w_random =", w_random)
```

```python
# оптимизация абсолютных приоритетов
F = sum([c[i] * lambdas[i] * w[i] for i in range(N)])
print("F =", F)

cond1 = [c[i] / v[i] for i in range(N)]
print("condition 1 =", cond1)

cond2 = [2 * v[i] * c[i] / alpha_2[i] for i in range(N)]
print("condition 2 =", cond2)

print()

for i in range(N):
    print("i =", i, "condition 2 =", cond2[i])

print()

for j in range(N):
    print("j =", j, "condition 1 =", cond1[j])
```

Результат при заданном приоритете [1, 2, 3, 4, 5]:

```
theta = [0.2, 0.125, 0.16666666666666669, 0.2727272727272727, 0.025974025974025972]
w = [0.037500000000000006, 0.09375000000000003, 0.14772727272727276, 0.31788665879574973, 0.4941466983720506]
w_random = 0.2803919810730412
F = 3.3408816535577106
condition 1 = [25.0, 40.0, 30.0, 13.333333333333334, 100.0]
condition 2 = [33.33333333333333, 53.33333333333333, 40.0, 17.77777777777778, 133.33333333333331]

i = 0 condition 2 = 33.33333333333333
i = 1 condition 2 = 53.33333333333333
i = 2 condition 2 = 40.0
i = 3 condition 2 = 17.77777777777778
i = 4 condition 2 = 133.33333333333331

j = 0 condition 1 = 25.0
j = 1 condition 1 = 40.0
j = 2 condition 1 = 30.0
j = 3 condition 1 = 13.333333333333334
j = 4 condition 1 = 100.0
```

Результат при оптимальном приоритете [5, 2, 3, 1, 4]:

```
theta = [0.01, 0.10309278350515465, 0.12987012987012989, 0.2777777777777778, 0.28846153846153844]
w = [0.00023195876288659796, 0.020384254920337398, 0.03422619047619048, 0.13080929487179488, 0.3658586132177682]
w_random = 0.07787253299720655
F = 1.674044624803709
condition 1 = [100.0, 40.0, 30.0, 25.0, 13.333333333333334]
condition 2 = [133.33333333333331, 53.33333333333333, 40.0, 33.33333333333333, 17.77777777777778]

i = 0 condition 2 = 133.33333333333331
i = 1 condition 2 = 53.33333333333333
i = 2 condition 2 = 40.0
i = 3 condition 2 = 33.33333333333333
i = 4 condition 2 = 17.77777777777778

j = 0 condition 1 = 100.0
j = 1 condition 1 = 40.0
j = 2 condition 1 = 30.0
j = 3 condition 1 = 25.0
j = 4 condition 1 = 13.333333333333334
```

Сравнение результатов оптимального упорядочивания приоритетов для случаев абсолютных и относительных приоритетов, а также для бесприоритетного обслуживания:

Бесприоритетное обслуживание:

```
Средняя продолжительность ожидания в каждой из 3-х очередей равна 0.09968634686346864
Среднее время ожидания произвольно выбранного заказа на обслуживание равно 0.09968634686346864
Среднее число заказов, находящихся в 1 -й очереди равно 0.2990590405904059
Среднее число заказов, находящихся в 2 -й очереди равно 0.19937269372693728
Среднее число заказов, находящихся в 3 -й очереди равно 0.04984317343173432
Среднее число заказов, находящихся в 4 -й очереди равно 0.09968634686346864
Среднее число заказов, находящихся в 5 -й очереди равно 0.10965498154981551
Средняя длина очереди равна 0.7576162361623617
Значение функции стоимости: 1.963821033210332
```

Относительные приоритеты:

```
R = [0.03, 0.23, 0.28, 0.48000000000000004, 0.645]
Средняя продолжительность ожидания в каждой из 3-х очередей w = [0.06962628865979381, 0.09042375150622574, 0.12182088744588744,
0.18038862179487178, 0.3658586132177682]
Среднее время ожидания произвольно выбранного заказа:  0.13598288276963072
Среднее число заказов, находящихся в 1 -й очереди равно 0.20887886597938143
Среднее число заказов, находящихся в 2 -й очереди равно 0.18084750301245148
Среднее число заказов, находящихся в 3 -й очереди равно 0.06091044372294372
Среднее число заказов, находящихся в 4 -й очереди равно 0.18038862179487178
Среднее число заказов, находящихся в 5 -й очереди равно 0.402444474539545
Среднее число (по обслуживающей системе) находящихся в очереди заказов: 1.0334699090491934
F = 5.131832267251467
Скорости извлечения прибором штрафов из СМО:  [100.0, 40.0, 30.0, 25.0, 13.333333333333334]
```

Абсолютные приоритеты с дообслуживанием:

```
theta = [0.01, 0.10309278350515465, 0.12987012987012989, 0.2777777777777778, 0.28846153846153844]
w = [0.00023195876288659796, 0.020384254920337398, 0.03422619047619048, 0.13080929487179488, 0.3658586132177682]
w_random = 0.07787253299720655
F = 1.674044624803709
condition 1 = [100.0, 40.0, 30.0, 25.0, 13.333333333333334]
condition 2 = [133.33333333333331, 53.33333333333333, 40.0, 33.33333333333333, 17.77777777777778]

i = 0 condition 2 = 133.33333333333331
i = 1 condition 2 = 53.33333333333333
i = 2 condition 2 = 40.0
i = 3 condition 2 = 33.33333333333333
i = 4 condition 2 = 17.77777777777778

j = 0 condition 1 = 100.0
j = 1 condition 1 = 40.0
j = 2 condition 1 = 30.0
j = 3 condition 1 = 25.0
j = 4 condition 1 = 13.333333333333334
```

Ответ.

Таким образом, для системы массового обслуживания $M_N / GI_N / 1$ с абсолютными приоритетами с дообслуживанием были получены следующие результаты:

- среднее полное время обработки информации каждого типа при оптимальном порядке приоритетов равно:
  - 5-й тип — 0,01;
  - 2-й тип — 0,1;
  - 3-й тип — 0,13;
  - 1-й тип — 0,28;
  - 4-й тип — 0,29;
- средняя продолжительность ожидания обработки информации каждого типа при оптимальном порядке приоритетов равна:
  - 5-й тип — 0,01;
  - 2-й тип — 0,1;

3-й тип — 0,13;
1-й тип — 0,28;
4-й тип — 0,29;

- средняя продолжительность ожидания обработки информации из общего потока при оптимальном порядке приоритетов равна 0,078.

Так же были определены:

а) текущий порядок приоритетов [1, 2, 3, 4, 5] не является оптимальным;
б) оптимальный порядок приоритетов: [5, 2, 3, 1, 4];
в) самым оптимальным из трех вариантов обслуживания (бесприоритеное, с относительными приоритетами, с абсолютными приоритетами) является обслуживание с абсолютными приоритетами, так как при таком порядке обслуживания критерий эффективности работы СМО имеет наименьшее значение F = 1, 674.