

УДК 004.01:006.72(470.22)

## МАТЕМАТИЧЕСКИЙ АНАЛИЗ ОДНОГО СПОСОБА ПРЕДСТАВЛЕНИЯ ДВУХ FIFO-ОЧЕРЕДЕЙ В ОБЩЕЙ ПАМЯТИ

Н. В. Каблукова<sup>1</sup>, А. В. Соколов<sup>2</sup>

<sup>1</sup> *Петрозаводский государственный университет*

<sup>2</sup> *Институт прикладных математических исследований  
Карельского научного центра РАН*

Во многих приложениях требуется работа с несколькими FIFO-очередями, расположенными в общем пространстве памяти. Для этого применяют различные программные или аппаратные решения [4, 7, 8].

В работе [8] поставлена задача построения математической модели процесса работы с несколькими FIFO-очередями в общей памяти, когда на нечетном шаге дискретного времени с известными вероятностями допускаются операции включения элементов в очереди, а на четном – операции исключения.

В [5] предложена математическая модель этого процесса для двух FIFO-очередей, и решается задача оптимального разбиения общей памяти для очередей в случае их последовательного циклического представления.

В данной работе построены математическая и имитационная модели процесса работы с двумя очередями, когда они двигаются по кругу друг за другом [9].

Ключевые слова: FIFO-очередь, случайное блуждание, регулярные цепи Маркова.

### N. V. Kablukova, A. V. Sokolov. MATHEMATICAL ANALYSIS OF A WAY OF PRESENTING TWO FIFO QUEUES IN SHARED MEMORY

Many applications need to work with several FIFO queues located in a common memory space. Various software or hardware solutions are used to this end [4, 7, 8]. In [8] the problem was formulated for building a mathematical model of the process of working with several FIFO queues in shared memory, where insertion of an item in the queue is allowed on an odd step of the discrete time, and deletion operations – on an even step. The probabilities of the operations are known. In [5] a mathematical model of this process for the number of queues  $n = 2$  was proposed, and the problem of optimal partitioning of the shared memory between the two FIFO queues in the case of successive cyclic representation was dealt with.

In this paper we construct a mathematical and a simulation models of the process of working with two queues, when they move around a circle, one after another [9].

Key words: FIFO queue, casual walk, regular Markov chains.

## ВВЕДЕНИЕ

Эффективные алгоритмы работы с несколькими FIFO-очередями в общем пространстве памяти необходимы при разработке различных сетевых устройств и встроенных операционных систем, управляющих потоками пакетов Internet, таких, например, как Cisco IOS, где требования на время обработки пакетов маршрутизатором очень жесткие. Механизм страничной виртуальной памяти здесь не используется, и вся работа происходит в нескольких пулах оперативной памяти. Количество очередей в таких устройствах может достигать нескольких сотен и тысяч, а в будущем, по экспертным оценкам, может достигнуть нескольких миллионов [14]. Для представления FIFO-очередей применяют различные программные или аппаратные решения [4, 7, 8].

В данной статье анализируются математические модели для некоторых способов представления нескольких FIFO-очередей в памяти одного уровня [7]. В качестве критерия оптимальности рассмотрена минимальная доля потерянных элементов при бесконечном времени работы очередей. Эту величину разумно минимизировать, когда переполнение очереди является не аварийной, а стандартной ситуацией (здесь мы подчеркиваем, что в некоторых приложениях при переполнении очереди работа программы заканчивается, и тогда в качестве критерия оптимальности надо рассматривать максимальное среднее время до переполнения памяти). Т. е., если очередь занимает всю предоставленную ей память, то все последующие элементы, поступающие в нее, отбрасываются до тех пор, пока не появится свободная память (т. е. до тех пор, пока не произойдет исключение элемента из очереди). Такая схема работы применяется, например, в работе сетевых маршрутизаторов [4] в том случае, когда по мере увеличения трафика очередь на исходящем интерфейсе маршрутизатора заполняется пакетами. Такое поведение маршрутизатора называется "сбросом хвоста". Потери пакетов приводят к нежелательному результату, поэтому число таких ситуаций необходимо свести к минимуму. Мы в этой работе строим математическую модель в виде случайных блужданий по целочисленной пирамиде. Первоначально такие модели в виде случайного блуждания в треугольнике [10, 12, 13] были построены для решения задачи анализа процесса работы с двумя стеками, растущими навстречу друг другу, поставленной в [7]. В этих моделях предполагается, что на

каждом шаге дискретного времени с заданными вероятностями происходят некоторые операции со структурами данных. Время выполнения операций это не случайная величина, а константа, поэтому фиксированным является и шаг времени. В [2, 3] предлагались модели работы со стеками в двухуровневой памяти. В [1, 11] предлагались модели для последовательного, связанного и страничного способов представления нескольких FIFO-очередей в памяти одного уровня.

В работе [8] приведены результаты имитационных экспериментов и поставлена задача построить математическую модель процесса работы с несколькими FIFO-очередями в общей памяти, когда операции с очередями выполняются по несколько другому принципу. В данной схеме работы на нечетном шаге допускаются операции включения элементов в одну из  $n$  очередей с равными вероятностями, а на четном шаге – операции исключения элементов из очередей с равными вероятностями. Исключение из пустой очереди не приводит к завершению работы. В [8] ставилась задача определить вероятность (как функцию от  $n$  и  $j$ ) того, что очередь, выбранная для операции на  $j$ -ом шаге, будет пустой, а также вычислить математическое ожидание количества элементов в очередях после  $j$  операций. В данной задаче не рассматривался конкретный способ представления очередей в памяти, т. е. предполагалось, что очереди могут быть неограниченной длины, что на практике невыполнимо.

В [5] предложена математическая модель этого процесса для числа очередей  $n = 2$ , и решается задача оптимального разбиения общей памяти для двух FIFO-очередей в случае их последовательного циклического представления в предположении, что операции с очередями выполняются по этому принципу, но с неравными вероятностями.

В данной работе построены математическая и имитационная модели процесса работы с двумя очередями, когда они двигаются по кругу друг за другом [9]. В случае, когда одна из очередей была пустой, но ожила, она начинает работу с середины пустого места.

Для решения поставленных задач используется аппарат управляемых случайных блужданий, регулярных цепей Маркова, система Intel® Math Kernel Library PARDISO\*. Анализируются результаты численных экспериментов, в которых производится сравнение двух методов представления очередей. Вычисления производились с помощью кластера КарНЦ РАН.

## МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Пусть в памяти размера  $m$  единиц мы работаем с двумя циклическими FIFO очередями, которые двигаются друг за другом по кругу. Возобновление движения пустой очереди начинается с середины промежутка между очередями.

Операции, производимые с очередями, выполняются по следующей схеме: на нечетном шаге происходит операция включения в одну из очередей, на четном шаге – операция исключения из какой-либо очереди, причем известны некоторые вероятностные характеристики операций, производимых с очередями. Пусть  $p_1$  и  $p_2$  – вероятности включения информации в первую и вторую очереди соответственно,  $q_1$  и  $q_2$  – вероятности исключения информации из первой и второй очередей соответственно.

Поскольку построенная на основе такой постановки задачи марковская цепь не будет являться регулярной и однородной, два последовательных шага объединяем в один, а также вводим следующие вероятности операций, не изменяющих длины очередей (например, чтение):  $r_1$  – на нечетном шаге и  $r_2$  – на четном шаге, при этом  $r_1 \neq 0$ ,  $r_2 \neq 0$ .

Соответственно,  $p_1 + p_2 + r_1 = 1$ ,  $q_1 + q_2 + r_2 = 1$ .

Тогда на каждом шаге могут быть выполнены следующие действия:

1. Включение в первую, исключение из второй очереди с вероятностью  $p_1 q_2$ ,
2. Включение во вторую, исключение из первой очереди с вероятностью  $p_2 q_1$ ,
3. Включение в первую очередь с вероятностью  $p_1 r_2$ ,
4. Включение во вторую очередь с вероятностью  $p_2 r_2$ ,
5. Исключение из первой очереди с вероятностью  $q_1 r_1$ ,
6. Исключение из второй очереди с вероятностью  $q_2 r_1$ ,
7. Остаться на месте с вероятностью  $p_1 q_1 + p_2 q_2 + r_1 r_2$ ,

где  $p_1 q_2 + p_2 q_1 + p_1 r_2 + p_2 r_2 + q_1 r_1 + q_2 r_1 + p_1 q_1 + p_2 q_2 + r_1 r_2 = 1$ .

Предполагается, что в очередях хранятся данные фиксированного размера. При исключении информации из пустой очереди не происходит завершения работы. Область блуждания показана на рис. 1.

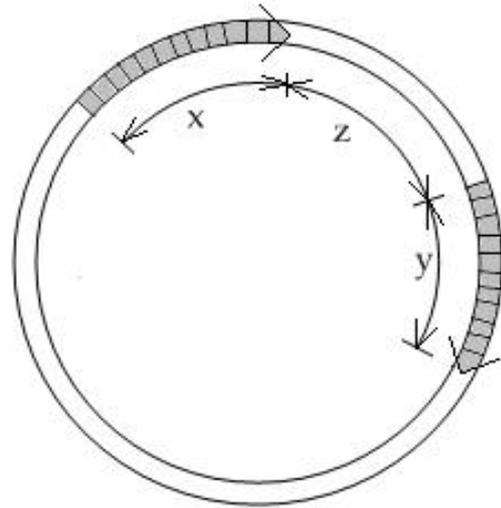


Рис. 1. Схема движения очередей

Целью исследования является определение средней доли потерянных элементов для сравнения со средней долей потерянных элементов при последовательном способе организации очередей в случае оптимального разбиения общей памяти [5].

Обозначим через  $x$  и  $y$  – текущие длины очередей, через  $z$  – расстояние между концом первой очереди и началом второй [7]. В качестве математической модели рассматриваем блуждание по целочисленной трехмерной пирамиде с вершиной  $(0, 0, 0)$  и основанием  $x + y + z = m$ .

Для учета потерянных элементов вводим два фиктивных экрана:

1. (а)  $z = -1$  и  $x + y' + z = m + 1$  ( $y' = y + 1$ ), где блуждание переходит на экран  $z = -1$ , если происходит включение в первую очередь в то время, когда  $z = 0$ , и на экран  $x + y' + z = m + 1$ , если происходит включение во вторую очередь в то время, когда  $x + y + z = m$ . В данном случае элемент будет потерян, а очереди будут находиться на экранах до тех пор, пока поступают новые элементы, в случае исключения элемента очереди вернуться в состояние, принадлежащие пирамиде.
- (б) В случае, когда происходит включение во вторую очередь и исключение из первой очереди, в то время когда  $x + y + z = m$ , происходит потеря элемента и одновременно увеличение расстояния между концом второй и началом первой очереди. Поэтому для этого также необходимо

ввести экран, который будет учитывать данную ситуацию ( $x' = x - 1$ ).

2. Если происходят операции включения и исключения из первой очереди при  $z = 0$  либо операции включения и исключения из второй очереди при  $x + y + z = m$ , происходит потеря элемента, а очереди возвращаются в состояния, принадлежащие пирамиде. Для учета потери элемента вводим экраны  $z = -2$  и  $x + y' + z = m + 2$  ( $y' = y + 2$ ) соответственно.

Определим схему переходов между состояниями. Пусть  $(x, y, z)$  – текущее состояние процесса, тогда блуждание по пирамиде можно описать следующим образом (1). Величина *distance* означает расстояние между концом первой очереди и началом второй. Если оживает первая очередь, то  $distance = \left\lceil \frac{m-y-1}{2} \right\rceil$ , и  $distance = \left\lceil \frac{m-x-1}{2} \right\rceil$ , если оживает вторая очередь. Черта сверху состояния указывает на то, что состояние принадлежит одному из фиктивных экранов.

$$\begin{aligned}
 (x, y, z) &\xrightarrow{r_1 r_2} (x, y, z) \\
 (x, y, z) &\xrightarrow{p_1 r_2} (x', y', z') = \begin{cases} (x+1, y, z-1) & x, y, z > 0 \\ (x+1, 0, 0) & y = 0, x < m \\ (1, y, \overline{distance-1}) & x = 0, y < m \\ (x, y, \overline{-1}) & z = 0, y \neq 0 \text{ либо } x = m \\ (x, y+1, z) & x = 0, y = m \end{cases} \\
 (x, y, z) &\xrightarrow{p_2 r_2} (x', y', z') = \begin{cases} (x, y+1, z) & x, y, z > 0 \\ (x, 1, \overline{distance}) & y = 0, x < m \\ (0, y, 0) & x = 0, y < m \\ (x, y+1, z) & x + y + z = m \end{cases} \\
 (x, y, z) &\xrightarrow{q_1 r_1} (x', y', z') = \begin{cases} (0, y, 0) & x = 0 \text{ либо } x = 1 \\ (x-1, 0, 0) & y = 0 \\ (x-1, y, z) & \text{иначе} \end{cases} \\
 (x, y, z) &\xrightarrow{q_2 r_1} (x', y', z') = \begin{cases} (x, 0, 0) & y = 0 \text{ либо } y = 1 \\ (0, y-1, 0) & x = 0 \\ (x, y-1, z+1) & \text{иначе} \end{cases} \\
 (x, y, z) &\xrightarrow{p_1 q_2} (x', y', z') = \begin{cases} (x+1, y-1, z) & x, y, z > 0 \\ (x+1, 0, 0) & x < m, y = 0 \text{ либо } y = 1 \\ (1, y-1, \overline{distance}) & x = 0, y < m \\ (x, y-1, \overline{-1}) & z = 0, y \neq 0 \text{ либо } x = m \\ (x, y-1, z+1) & x = 0, y = m \end{cases} \tag{1} \\
 (x, y, z) &\xrightarrow{p_2 q_1} (x', y', z') = \begin{cases} (x-1, y+1, z) & x, y, z > 0 \\ (x-1, 1, \overline{distance}) & y = 0, x < m \\ (0, y+1, 0) & y < m, x = 0 \text{ либо } x = 1 \\ (x', y, z) & x + y + z = m \end{cases} \\
 (x, y, z) &\xrightarrow{p_1 q_1} (x', y', z') = \begin{cases} (x, y, z-1) & x, y, z > 0 \\ (0, y, 0) & x = 0 \\ (x, 0, 0) & y = 0 \\ (x, y, \overline{-2}) & z = 0, x \neq 0 \\ (x, y+1, 0) & z = 0, y = m \end{cases} \\
 (x, y, z) &\xrightarrow{p_2 q_2} (x', y', z') = \begin{cases} (x, y, z+1) & x, y, z > 0 \\ (x, 0, 0) & y = 0 \\ (0, y, 0) & x = 0 \\ (x, y+2, z+1) & x + y + z = m \end{cases}
 \end{aligned}$$

Переходы с первого фиктивного экрана:

1) переходы с плоскости  $z = -1$  (2):

$$\begin{aligned}
& \overline{(x, y, -1)} \xrightarrow{r_1 r_2} (x, y, 0) \\
& \overline{(x, y, -1)} \xrightarrow{p_1 r_2} \overline{(x, y, -1)} \\
& \overline{(x, y, -1)} \xrightarrow{p_2 r_2} (x', y', z') = \begin{cases} \overline{(x, y + 1, 0)} & x + y = m \\ (x, y + 1, 0) & \text{иначе} \end{cases} \\
& \overline{(x, y, -1)} \xrightarrow{q_1 r_1} (x - 1, y, 0) \\
& \overline{(x, y, -1)} \xrightarrow{q_2 r_1} (x', y', z') = \begin{cases} (x, y, 0) & y = 0 \text{ либо } y = 1 \\ (x, y - 1, 1) & y > 1 \end{cases} \\
& \overline{(x, y, -1)} \xrightarrow{p_1 q_1} \overline{(x, y, -2)} \\
& \overline{(x, y, -1)} \xrightarrow{p_2 q_2} (x', y', z') = \begin{cases} \overline{(x, y + 2, 1)} & x + y = m \\ (x, y, 1) & \text{иначе} \end{cases} \\
& \overline{(x, y, -1)} \xrightarrow{p_1 q_2} (x', y', z') = \begin{cases} \overline{(x, y, -1)} & y = 0 \\ (x, y - 1, -1) & \text{иначе} \end{cases} \\
& \overline{(x, y, -1)} \xrightarrow{p_2 q_1} (x', y', z') = \begin{cases} \overline{(x', y, 0)} & x + y = m \\ (x - 1, y + 1, 0) & \text{иначе} \end{cases}
\end{aligned} \tag{2}$$

2) переходы из состояний  $x + y' + z = m + 1$   
( $y' = y + 1$ ) (3):

$$\begin{aligned}
& \overline{(x, y + 1, z)} \xrightarrow{r_1 r_2} (x, y, z) \\
& \overline{(x, y + 1, z)} \xrightarrow{p_1 r_2} (x', y', z') = \begin{cases} (x + 1, y, z - 1) & x, y, z > 0 \\ (x + 1, 0, 0) & y = 0, x < m \\ (1, y, \text{distance} - 1) & x = 0, y < m \\ \overline{(x, y, -1)} & z = 0, y \neq 0 \text{ либо } x = m \\ (x, y + 1, z) & x = 0, y = m \end{cases} \\
& \overline{(x, y + 1, z)} \xrightarrow{p_2 r_2} (x, y + 1, z) \\
& \overline{(x, y + 1, z)} \xrightarrow{q_1 r_1} (x', y', z') = \begin{cases} (0, y, 0) & x = 0 \text{ либо } x = 1 \\ (x - 1, 0, 0) & y = 0 \\ (x - 1, y, z) & \text{иначе} \end{cases} \\
& \overline{(x, y + 1, z)} \xrightarrow{q_2 r_1} (x', y', z') = \begin{cases} (x, 0, 0) & y = 0 \text{ либо } y = 1 \\ (0, y - 1, 0) & x = 0 \\ (x, y - 1, z + 1) & \text{иначе} \end{cases} \\
& \overline{(x, y + 1, z)} \xrightarrow{p_1 q_2} (x', y', z') = \begin{cases} (x + 1, y - 1, z) & x, y, z > 0 \\ (x + 1, 0, 0) & x < m, y = 0 \text{ либо } y = 1 \\ (1, y - 1, \text{distance}) & x = 0, y < m \\ \overline{(x, y - 1, -1)} & z = 0, y \neq 0 \text{ либо } x = m \\ (x, y - 1, z + 1) & x = 0, y = m \end{cases} \\
& \overline{(x, y + 1, z)} \xrightarrow{p_2 q_1} (x', y', z') = \begin{cases} \overline{(x - 1, y + 1, z + 1)} & x = 0 \\ (x, y + 1, z) & y = 0 \\ (x', y, z) & x \neq 0, y \neq 0 \end{cases} \\
& \overline{(x, y + 1, z)} \xrightarrow{p_1 q_1} (x', y', z') = \begin{cases} (x, y, z - 1) & x, y, z > 0 \\ (0, y, 0) & x = 0 \\ (x, 0, 0) & y = 0 \\ \overline{(x, y, -2)} & z = 0, x \neq 0 \\ (x, y + 1, 0) & z = 0, y = m \end{cases} \\
& \overline{(x, y + 1, z)} \xrightarrow{p_2 q_2} \overline{(x, y + 2, z)}
\end{aligned} \tag{3}$$

3) переходы из состояний  $(x', y, z)$  ( $x' = x - 1$ ) повторяют переходы из соответствующих состояний, принадлежащих пирамиде (4):

$$\overline{(x', y, z)} \Leftrightarrow \begin{cases} (0, y, 0) & x' = 0 \\ (x', y, z) & \text{иначе.} \end{cases} \tag{4}$$

Переходы из состояний второго фиктивного экрана повторяют переходы из соответствующих состояний, принадлежащих пирамиде (5):

$$\overline{(x, y, z)} \Leftrightarrow (x - 1, y, z + 2) \text{ для } z = -2$$

$$\overline{(x, y', z)} \Leftrightarrow \begin{cases} (x, 0, 0) & y = 0 \\ (x - 1, 0, 0) & y = 1 \\ (0, y - 1, 0) & y + z = m \\ (x, y, z + 1) & \text{иначе.} \end{cases} \text{ для } x + y' + z = m + 2 \quad (5)$$

Случайное блуждание будем рассматривать в виде регулярной конечной цепи Маркова с переходной матрицей  $P$ . Зададим нумерацию так, как показано на рис. 2: сначала нумеруем состояния, принадлежащие основанию пирамиды ( $z = 0$ ), далее поперечным сечениям ( $z = 1, 2, \dots, m - 1$ ) и высоте пирамиды ( $z = m$ ), затем состояния первого фиктивного экрана, последними нумеруем состояния второго фиктивного экрана.

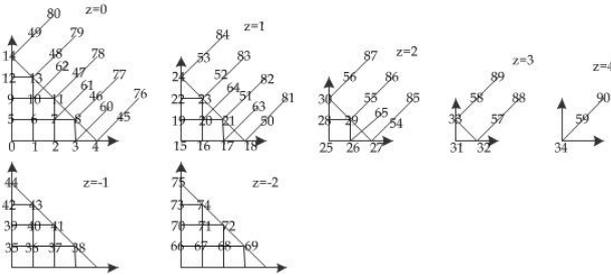


Рис. 2. Нумерация состояний при  $m = 4$

Нумерацию начинаем с нуля, общее количество состояний в цепи будет

$$\sum_{i=0}^m \frac{(i+1) \cdot (i+2)}{2} + \frac{5m^2 + 7m + 4}{2}.$$

Далее согласно введенной нумерации состояний и вышеуказанной схеме переходов составляется матрица переходных вероятностей  $P$ . В данном исследовании был установлен вид матрицы  $P$  для произвольных значений параметра  $m$ . При составлении матрицы для каждого конкретного состояния определяются те состояния, в которые процесс переходит при выполнении допустимых операций, и вычисляются соответствующие вероятности переходов. Данный процесс был автоматизирован при помощи программы на языке Си.

Следующим шагом является решение уравнения  $\alpha \cdot P = \alpha$ , где  $\alpha$  – предельный вектор для полученной марковской цепи. Для решения этого уравнения применялось средство решения Intel® Math Kernel Library PARDISO\* библиотеки Intel® MKL. Данный инстру-

мент используется для прямого решения систем линейных алгебраических уравнений с разреженной матрицей<sup>1</sup>, причем разработаны функции как для симметричных, так и для несимметричных матриц.

Элемент вектора  $\alpha_i$  – это средняя доля времени, которое процесс проводит в состоянии  $i$  [6]. Для вычисления времени, проводимого процессом в состояниях, где происходят потери элементов очередей, нужно просуммировать элементы вектора  $\alpha$ , соответствующие состояниям на экранах. При введенной нумерации это будут последние  $\left(\frac{5m^2+7m+4}{2}\right)$  элементов вектора  $\alpha$ .

## ЭКСПЕРИМЕНТЫ И РЕЗУЛЬТАТЫ

После завершения работы над моделью было проведено несколько экспериментов для возможного выявления присущих ей свойств и сравнения с ранее полученными данными и графиками.

1. Для выяснения ситуаций, когда выгоднее применить случай движения очередей по кругу, был проведен ряд вычислений с различными наборами исходных данных. Аналогичные вычисления были проведены для последовательного способа организации очередей в общей памяти при оптимальном разбиении памяти [5].

Также был рассмотрен случай, когда память заранее делится пополам при последовательном способе организации очередей. Это может быть полезно, если заранее неизвестны вероятностные характеристики операций, производимых с очередями. Некоторые полученные данные указаны в таблице, где столбцы (1) и (2) означают последовательное расположение очередей в памяти при оптимальном делении памяти и делении памяти пополам соответственно, столбец (3) – движение очередей по кругу друг за другом. Указанные результаты были подтверждены имитационными экспериментами.

<sup>1</sup>Матрица с большим количеством нулей.

Таблица 1. Величина потерь при различном представлении очередей при  $m = 16$

Входные данные	Величина потерь $T$ при переполнении		
	(1)	(2)	(3)
$p_1 = 0,4, p_2 = 0,45, q_1 = 0,4, q_2 = 0,45$	0,0568543452 ( $s = 8$ )	0,0568543452	0,0319323619
$p_1 = 0,3, p_2 = 0,45, q_1 = 0,35, q_2 = 0,5$	0,0190375345 ( $s = 8$ )	0,0190375345	0,0125850917
$p_1 = 0,8, p_2 = 0,1, q_1 = 0,1, q_2 = 0,8$	0,7000000000 ( $s = 10$ )	0,7000000040	0,6958703274
$p_1 = 0,75, p_2 = 0,1, q_1 = 0,15, q_2 = 0,7$	0,6000000000 ( $s = 10$ )	0,6000000099	0,5749273399
$p_1 = 0,7, p_2 = 0,15, q_1 = 0,2, q_2 = 0,65$	0,5000000008 ( $s = 9$ )	0,5000000032	0,4557056302
$p_1 = 0,6, p_2 = 0,3, q_1 = 0,15, q_2 = 0,75$	0,4500000039 ( $s = 9$ )	0,4500000354	0,4455273285
$p_1 = 0,4, p_2 = 0,45, q_1 = 0,6, q_2 = 0,35$	0,1022326838 ( $s = 6$ )	0,1030173486	0,0889535997
$p_1 = 0,3, p_2 = 0,6, q_1 = 0,45, q_2 = 0,45$	0,1502357136 ( $s = 6$ )	0,1514556020	0,1200671464
$p_1 = 0,25, p_2 = 0,65, q_1 = 0,45, q_2 = 0,45$	0,1901271731 ( $s = 5$ )	0,2002810614	0,1943398747
$p_1 = 0,2, p_2 = 0,3, q_1 = 0,45, q_2 = 0,05$	0,2500009312 ( $s = 7$ )	0,2500844509	0,3931085764
$p_1 = 0,3, p_2 = 0,5, q_1 = 0,1, q_2 = 0,8$	0,2000042218 ( $s = 8$ )	0,2000042218	0,3319594498

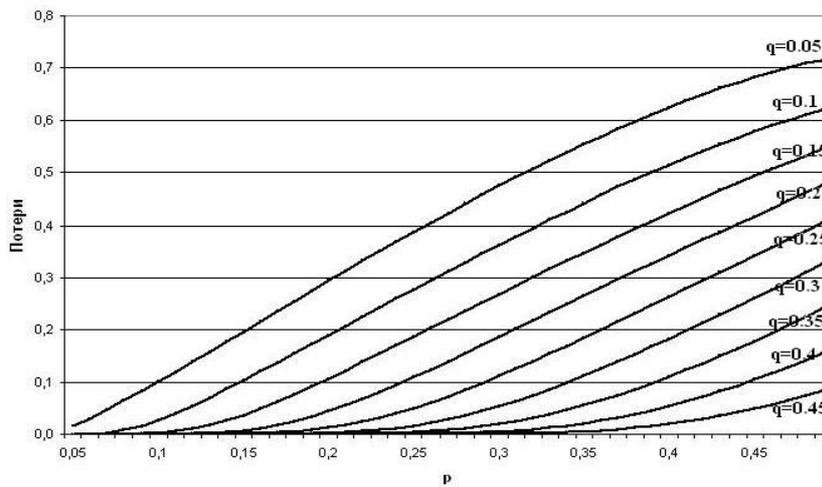


Рис. 3. Потери при постоянной величине  $q$

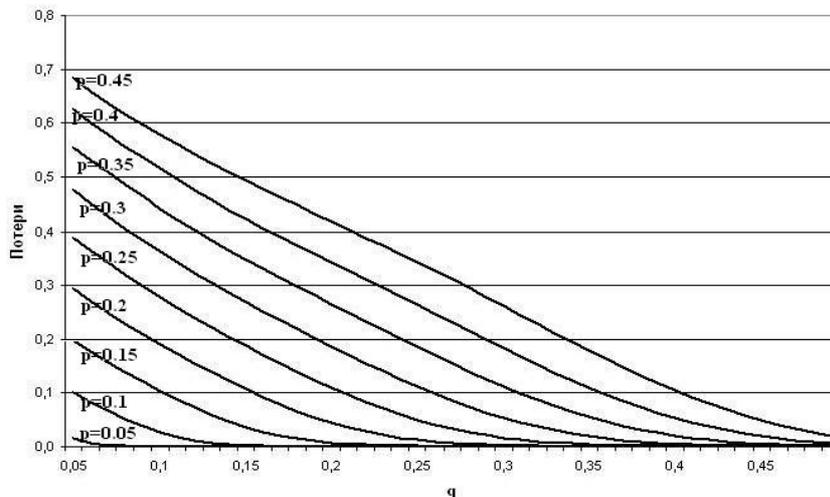


Рис. 4. Потери при постоянной величине  $p$

По этим данным видно, что при движении очередей по кругу потери меньше в двух случаях:

- (а) вероятности включения и исключения в очереди равны (либо практически равны). Это можно прокомментировать тем, что в такой ситуации обе очереди равномерно двигаются друг за другом по кругу, не переполняясь;
- (б) вероятность включения в одну из очередей намного больше, чем вероятность включения в другую очередь, а вероятность исключения из этой очереди – намного меньше, чем из другой. Данный случай объясняется тем, что при данных условиях та из очередей, вероятность включения в которую меньше, практически не занимает места в памяти, соответственно, теряется меньше включаемых элементов.

В других случаях лучше использовать оптимальное разбиение памяти в последовательном способе организации очередей. Если же вероятностные характеристики вообще неизвестны, предпочтительнее делить память пополам при последовательном расположении очередей.

2. Пусть  $p_1 = p_2 = p$ ,  $q_1 = q_2 = q$ , т. е. рассматриваем ситуацию равновероятного включения и исключения информации из очередей, тогда  $r_1 = 1 - 2p$ ,  $r_2 = 1 - 2q$ .

В данных условиях были построены графики зависимости суммарных потерь от изменения  $p$  при постоянной величине  $q$  (рис. 3), и от изменения  $q$  при постоянной величине  $p$  (рис. 4).

Подобные графики были также представлены в работе [5].

Работа выполнена при финансовой поддержке гранта РФФИ (проект 12-01-00253) и Программы стратегического развития ПетрГУ в рамках реализации комплекса мероприятий по развитию научно-исследовательской деятельности.

## ЛИТЕРАТУРА

1. Аксенова Е. А., Драц А. В., Соколов А. В. Оптимальное управление  $n$  FIFO-очередями на бесконечном времени // Информационно-управляющие системы. 2009. № 6. С. 46–54.
2. Аксенова Е. А., Лазутина А. А., Соколов А. В. Исследование немарковской модели управления стеком в двухуровневой памяти // Программирование. 2004. № 1. С. 1–10.
3. Аксенова Е. А., Соколов А. В. Оптимальное управление двумя параллельными стеками // Дискретная математика. 2007. № 1. С. 67–75.
4. Боллапрагада В., Мэрфи К., Расс У. Структура операционной системы Cisco IOS. М.: Вильямс, 2002. 208 с.
5. Каблукова Н. В., Соколов А. В. Оптимальное разбиение общей памяти для двух последовательных циклических FIFO-очередей // Прикладная информатика. 2012. № (40). С. 113–125.
6. Кемени Дж., Снелл Дж. Конечные цепи Маркова. М.: Наука, 1960. 272 с.
7. Кнут Д. Искусство программирования для ЭВМ. Т. 1. М.: Вильямс, 2001. 736 с.
8. Седжвик Р. Фундаментальные алгоритмы на C++. К.: Издательство «Диасофт», 2001. 688 с.
9. Соколов А. В. Математические модели и алгоритмы оптимального управления динамическими структурами данных. Петрозаводск: ПетрГУ, 2002. 216 с.
10. Соколов А. В. О распределении памяти для двух стеков // Автоматизация эксперимента и обработки данных. Петрозаводск: изд-во Карельского филиала АН СССР, 1980. С. 65–71.
11. Aksenova E. A., Sokolov A. V. The optimal implementation of two FIFO-queues in single-level memory // Applied Mathematics. 2011. Vol. 2, N 10. P. 1297–1302.
12. Flajolet P. The evolution of two stacks in bounded space and random walks in a triangle // Lecture Notes in Computer Science. 1986. Vol. 223. P. 325–340.
13. Louchard G., Schott R., Tolley M., Zimmermann P. Random walks, heat equation and distributed algorithms // Journal of Computational and Applied Mathematics. 1994. N 53. P. 243–274.
14. Nikolgiannis A., Katevenis M. Multi-Queue Management for Advanced QoS in High-Speed Communication Systems Computer Architecture and VLSI Systems Lab, Institute of Computer Science (ICS) Head.  
<http://archvlsci.ics.forth.gr/muqpro/queueMgt.html>

## СВЕДЕНИЯ ОБ АВТОРАХ:

### **Каблукова Наталья Вениаминовна**

студентка

Петрозаводский государственный университет  
ул. Ленина, 33, Петрозаводск, Республика Карелия,  
Россия, 185910

эл. почта: xituska@mail.ru

### **Соколов Андрей Владимирович**

ведущий научный сотрудник

Институт прикладных математических исследований  
Карельского научного центра РАН  
ул. Пушкинская, 11, Петрозаводск, Республика Каре-  
лия, Россия, 185910

эл. почта: avs@krc.karelia.ru

тел.: (8142) 766312

### **Kablukova, Natalia**

Petrozavodsk State University

33 Lenina St., 185910 Petrozavodsk, Karelia, Russia

e-mail: xitushka@mail.ru

### **Sokolov, Andrey**

Institute of Applied Mathematical Research, Karelian

Research Centre, Russian Academy of Sciences

11 Pushkinskaya St., 185910 Petrozavodsk, Karelia,  
Russia

e-mail: avs@krc.karelia.ru

tel.: (8142) 766312