УДК 519.876.2, 519.876.5

ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ ОЧЕРЕДЬЮ В СИСТЕМЕ МАССОВОГО ОБСЛУЖИВАНИЯ С ОГРАНИЧЕННОЙ ПРОИЗВОДИТЕЛЬНОСТЬЮ

Д. А. Зубок[a]*, канд. физ.-мат. наук, заместитель заведующего кафедрой*
А. В. Маятин[a]*, канд. пед. наук, доцент*
[a]*Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, Санкт-Петербург, РФ*

Постановка проблемы: *для систем массового обслуживания с бесконечным числом приборов отсутствует детальное исследование аналогов пороговому характеру оптимальных дисциплин обслуживания, установленных для систем массового обслуживания с конечным числом неоднородных приборов. В то же время модели систем с бесконечным числом приборов дают удовлетворительное описание вычислительных узлов с многопоточностью. Целью работы является выявление порогового характера оптимальной дисциплины управления системами с бесконечным числом приборов переменной производительности, зависящей от числа требований в системе.* ***Методы:*** *для описания процесса обслуживания применяются методы процессов с дискретным временем и счетным числом состояний. При описании выходящего потока используется авторегрессионная схема.* ***Результаты:*** *в приближении процессов с дискретным временем выведены уравнения авторегрессии и скользящего среднего для описания входящего и выходящего потоков процесса обслуживания. При этом предполагается, что порядок окончания выполнения требований совпадает с порядком их поступления на выполнение. Для проверки гипотезы о существовании оптимальной дисциплины управления очередью проводятся имитационные эксперименты. При проведении экспериментов интенсивность входящего потока принимала значения как меньше, так и больше производительности системы при выполнении только одного требования. Имитационные эксперименты показали зависимость среднего времени пребывания требования в системе от разрешенной величины очереди и интенсивности входящего потока. Таким образом, принцип повышения производительности заключается в том, что при заданной интенсивности потока требований однозначно определяется нижняя граница для величины очереди, начиная с которой производительность системы не растет.* ***Практическая значимость:*** *предложенные математическая и имитационная модели можно применять для изучения одно- и многофазных систем с различным распределением времени выполнения требований и различными законами падения производительности.*

Ключевые слова *— система массового обслуживания, бесконечное число приборов, управляемый марковский процесс, качество обслуживания, имитационная модель.*

Введение

Модели однопроцессорных систем с распределением процессорного времени по задачам изучаются с 70-х годов прошлого столетия [1, 2]. Задачи исследования времени отклика, времени выполнения операций решаются в рамках систем массового обслуживания (СМО) $M \mid M \mid 1\ PS$, которые обобщают системы $M \mid M \mid 1$ на уровне дисциплины обслуживания заявок тем, что предусмотрены циклы, охватывающие буфер с очередью задач и центральный процессор. Таким образом, для обработки заявки выделяется системное время, по окончании которого заявка переходит в режим ожидания, пока обрабатываются другие заявки. В работах [3, 4] изучены такие характеристики качества СМО $M \mid M \mid 1\ PS$, как среднее время ожидания и среднее время выполнения. Кроме того, математические модели таких задач близки к моделям задач управления транспортными потоками [5]. Модели управляемых случайных процессов и цепей находят применение при решении ряда других вопросов, связанных с организацией работы вычислительных средств, например, обработки информации, оптимального обмена данными.

Модели типа $M \mid G \mid 1$ и $M \mid G \mid c$, в том числе с учетом простоев приборов, изучены в работах [6–11], где рассматривается как детерминированный, так и стохастический процесс прерываний. Другой цикл работ [12, 13] посвящен исследованию систем типа $M \mid G \mid \infty$, в том числе с внешним марковским управлением, которые моделируют широкий класс систем от телекоммуникационных до биологических.

Однако довольно мало исследований в рамках моделей управляемых марковских систем или управляемых СМО для систем $M \mid M \mid 1\ PS$, $M \mid G \mid c$, $M \mid G \mid \infty$, т. е. проблема оптимального управления для систем класса $M \mid G \mid \infty$ является актуальной.

Задача оптимального управления СМО с $K > 2$ неоднородными приборами при различных предположениях относительно входного потока требований (пуассоновского, рекуррентного, марковского) и распределений длительностей их обслуживания (показательных, эрланговских или фазового типа) подробно изучена в работах [14–16]. Показано [14], что оптимальная дисциплина обслуживания требований в системе с K приборами фиксированной суммарной производительностью $M = \mu_1 + ... + \mu_K$ ($\mu_1 > \mu_2 > ... > \mu_K$) с ожиданием и конечной очередью по критерию минимизации среднего числа требований в системе имеет пороговый характер. Суть дисциплины обслуживания порогового типа заключается в том, что относительно критерия «среднее время

пребывания требования в системе» оптимальная дисциплина определяет для каждого состояния системы $x = (q, d_1, ..., d_K)$, где q — длина очереди; $d_i = 1$, если i-й прибор включен, и $d_i = 0$, если i-й прибор выключен, значение $q_j^*(x) = q^*(x)$ порога длины очереди q, начиная с которого ($q_j^*(x) \geq q^*(x)$, $q(x)$ — длина очереди в состоянии x) включается прибор j с наибольшей производительностью из оставшихся выключенных приборов. Значение порога $q_j^*(x)$ включения j-го прибора определяется соотношением

$$q_j^*(x) = \left[\frac{1}{\mu_j}\sum_{k=1}^{j-1}\mu_k\right] - (j-1),\ j = 2, 3, \ldots, K.$$

При этом система описывается управляемыми случайными процессами; для функции потерь в задаче оптимизации используется уравнение Беллмана. В работе [17] вычислены различные характеристики производительности СМО с неоднородными приборами при различных дисциплинах занятия приборов.

В то время как описанные выше многолинейные СМО являются адекватной моделью работы узла сети передачи данных, модель вычислительного узла слабо проработана в рамках теории управляемых СМО.

В настоящей работе проводится исследование типа оптимального управления системой $M \mid M \mid \infty$ в предположении, что производительность системы по выполняемому требованию зависит от числа уже выполняемых требований в системе. Для системы строится стохастическая модель процесса в дискретном времени как приближение к однородным управляемым марковским процессам с непрерывным временем, конечными пространствами состояний и управлений и аддитивным функционалом потерь. При этом используется модель авторегрессии и скользящего среднего [18]. Приближение дискретного времени позволяет сформулировать основные уравнения системы и проанализировать пространства решений, а также использовать имитационное моделирование для исследования поведения системы. Возможность построения моделей СМО в приближении непрерывного или дискретного времени подробно обсуждается в работе [19].

Математическая и имитационная модели системы $M \mid M \mid \infty$ с очередью переменной длины

Будем считать, что поведение системы описывается случайным процессом. На вход в систему подается однородный поток требований с интенсивностью $\lambda(t)$. В системе предусмотрена очередь ограниченной длины r. Дисциплина обслуживания: требования из очереди принимаются на выполнение в том случае, если система свободна или очередь полностью заполнена в момент поступления нового требования в систему. Величины $r, d_1, d_2, ..., d_n...$ — параметры управления; $L(r)$ — накладные расходы на содержание очереди длины r в системе; производительность системы $\mu(n)$ — количество требований, обрабатываемых в единицу времени при условии, что в системе обрабатывается n требований, требования начали обрабатываться в один и тот же момент времени и время выполнения требования распределено по показательному закону. Пусть $\mu(n)$ имеет вид $\mu(n) = \beta + (\alpha n)^{-1}$ при $n \geq 1$, не зависит от числа требований в очереди и не является случайной величиной при фиксированном значении n. Введем для дальнейшего изложения обозначение $\mu(1) = \mu_0$. Входящий поток опишем процессом восстановления, а именно введем случайную величину $S_k = T_1 + ... + T_k$, где S_k есть время поступления k-го требования в систему; величина k — случайная, зависящая от времени; T_i — независимые одинаково распределенные экспоненциально величины. Введем случайную величину Q_k — время окончания выполнения k-го требования.

Введем предположения:

1) порядок окончания выполнения требований совпадает с порядком их поступления в систему;

2) величина $\mu(n)$ не является случайной величиной при фиксированном значении n.

Пусть t_k — момент поступления k-го требования из очереди на выполнение. Выражение

$$\int_{t_k}^{\tau_k} \mu(t)\mathrm{d}t = 1$$

определяет для k-го требования, поступившего в момент времени t_k, момент времени его выполнения $\tau_k (Q_k = \tau_k)$. Производительность системы $\mu(t)$ — случайный скачкообразный процесс с разрывами в точках t_i и τ_j такой, что $\mu(t) = \mu(n_t)$, где n_t — случайная величина — число обрабатываемых требований в системе. Таким образом, время выполнения i-го требования зависит от случайных моментов времени поступления входящих требований в процессе его выполнения и моментов времени окончания выполнения требований в интервале времени выполнения i-го требования. При этом наличие очереди не предполагается.

В случае если верно предположение 1, момент времени окончания выполнения k-го требования будем определять из уравнения

$$\int_{t_1}^{\tau_k} \mu(t)\mathrm{d}t = k. \qquad (1)$$

Уравнение (1) можно переписать в терминах случайных величин S_k и Q_k следующим образом:

$$\alpha_n Q_n = -n + \mu S_1 - \beta \sum_{k:\, S_k \le t} S_k + \sigma \sum_{k<n} Q_k, \qquad (2)$$

где $\alpha_n = \mu - \beta N_t + \sigma(n-1)$; β — падение производительности при поступлении требования в систему; σ — возрастание производительности при окончании выполнения требования.

В том случае если допускается $\tau_k > \tau_j$ при $k<j$, то уравнение (1) перепишется в виде

$$\int_{t_1}^{\tau_k} \mu(t)\mathrm{d}t = G_t, \qquad (3)$$

где G_t — число выполненных требований к моменту времени t.

При этом выражение (2) заменится на выражение

$$\alpha_n Q_n = -G_t + \mu S_1 - \beta \sum_{k:S_k<t} S_k + \sigma \sum_{k:Q_k<t} Q_k, \qquad (4)$$

где $\alpha_n = \mu - \beta N_t + \sigma G_t$; $Q_n \ge t$.

Если предположить случайный характер времени обслуживания в расчете на одно требование, то выражения (2), (4) преобразуются к виду

$$\alpha_n Q_n = -G_t + \mu S_1 - \sum_{k:S_k<t} \beta_k S_k + \sum_{k:Q_k<t} \sigma_k Q_k. \qquad (5)$$

Для оценки статистических характеристик случайной последовательности Q_k, определяемой выражением (2), требуется это выражение преобразовать к модели авторегрессии и скользящего среднего с «белым шумом» и дать статистическую оценку математического ожидания, среднеквадратичного отклонения и функции ковариации. При этом аналитическая постановка задачи фильтрации, описывающей преобразование входящего потока в выходящий поток, затруднена.

В настоящей работе задача расчета величины очереди рассматривается в среде имитационного моделирования AnyLogic Professional 6.4.1, т. е. воспроизводится процесс преобразования входящего потока в выходящий поток и подбирается оптимальное значение очереди, минимизирующее среднее время пребывания требования в системе.

Для проведения экспериментов, подтверждения адекватности построенной модели и полученных результатов были приняты следующие параметры модели и их значения:

1) интенсивность потока требований (задач) требования (lambda) от 0,04 до 0,06 с шагом 0,01;

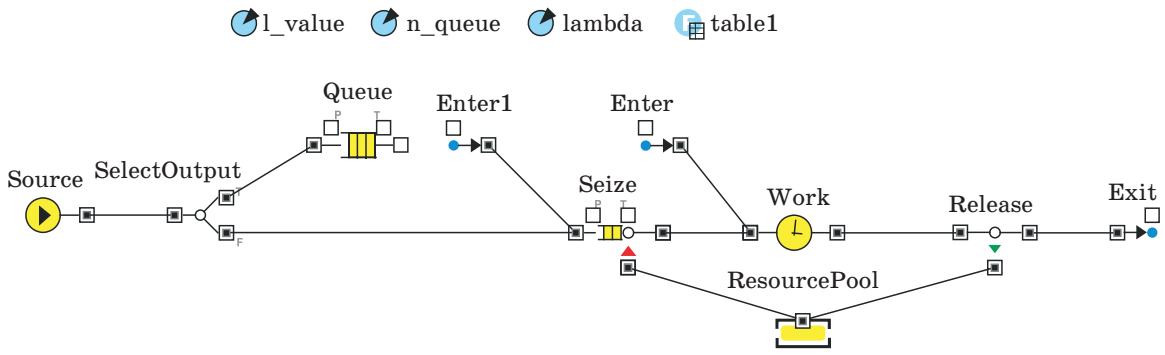
2) кортеж требований (l_value) от 1500 до 2500 с шагом 500 (ограничивает количество поступаемых требований в систему);

3) вместимость очереди (n_queue) от 1 до 50 с шагом 1 (ограничивает максимальное количество требований, находящихся в очереди);

4) закон падения производительности (table1). Значения зависимости количества одновременно обслуживаемых требований от времени их выполнения задаются в табличной форме, после чего производится экстраполяция. Исходные значения нами были получены с помощью эксперимента: производился запуск архивации файла и замер времени выполнения этой операции. Затем эксперимент проводился для двух файлов одинакового объема одновременно. Далее количество файлов на параллельную архивацию увеличивали.

Реализованы параметрические статистические оптимизационные эксперименты. Проводилось усреднение по серии более 100 экспериментов для каждого набора параметров. Вычислялось минимальное значение очереди по критерию минимального среднего времени обслуживания требований в системе с учетом пребывания требований в очереди.

Данная система (рис. 1) построена на стандартных элементах и классах библиотеки Enterprise Library пакета имитационного моделирования AnyLogic Professional 6.4.1.



■ ***Рис. 1.*** Схема модели системы в пакете AnyLogic Professional 6.4.1

Для моделирования объекта *Source* взят стандартный класс Entity, который модифицирован добавлением уникального идентификатора заявки и начального времени обслуживания заявки. Поток заявок является пуассоновским, table1.get(0) задает время обслуживания заявки для одной выполняемой заявки в системе.

Объект *SelectOutput* направляет входящие требования в один из двух выходных портов в зависимости от выполнения заданного условия. Поступившее требование покидает объект SelectOutput в тот же момент времени и поступает в очередь (queue) при условии, что в данный момент времени существует нагрузка на систему (resourcePool.busy()) и очередь не заполнена (queue.canEnter()). Если же очередь заполнена, то первое стоящее в очереди требование поступает на обработку (Enter1.take()), а текущее требование становится в очередь. В противном случае, если система не нагружена, требование отправляется на обслуживание (seize). Объект *Queue* моделирует очередь требований. Вместимость данного объекта задается параметром n_queue.

Требования покидают объект только путем вызова функции removeFirst(). *Enter1* — технический элемент модели, служащий для маршрутизации требований из очереди (queue) на обслуживание с помощью функции take(). *ResourcePool* задает набор ресурсов, которые могут захватываться и освобождаться требованиями с помощью объектов *Seize*, *Release*. В данной модели используется моделирование ресурса как числа (а не отдельного объекта), что позволяет получать с помощью метода resourcePool.busy() текущее количество одновременно обслуживаемых требований в каждый дискретный момент времени.

Попадая на обслуживание в объект *Seize*, очередное требование захватывает с собой ресурс и освобождает его только после обслуживания. Реализация данного элемента позволяет моделировать многопоточность обслуживания требования и перерасчет времени обслуживания для каждого требования.

Seize захватывает для требования заданное количество ресурсов определенного типа. При захвате ресурса требование мгновенно покидает этот объект. Для моделирования процесса (эффекта) падения производительности при увеличении количества одновременно выполняемых требований был реализован следующий алгоритм при выходе требования из объекта:

1. Изымаются все требования, находящиеся на обслуживании в данный момент, со значением времени, необходимым до конца обслуживания (work.getRemainingTime()).

2. Время обслуживания каждого требования (delay) пересчитывается с учетом коэффициента падения производительности.

3. Рассчитывается время обслуживания текущего требования из расчета текущей производительности системы.

4. Все требования отправляются на обработку одновременно (enter) с заново рассчитанными параметрами.

Enter — технический объект модели, служит либо для поступления требований с пересчитанными параметрами времени обслуживания, изъятых в момент поступления, либо для выхода требования из обслуживания.

Элемент *Work* построен на основе стандартного элемента Delay. Задерживает требование на заданный период времени, тем самым моделируя время обслуживания требования. Данное время задается параметром требования delay. При этом данный объект позволяет задерживать множество требований одновременно. Ограничения по вместимости нет.

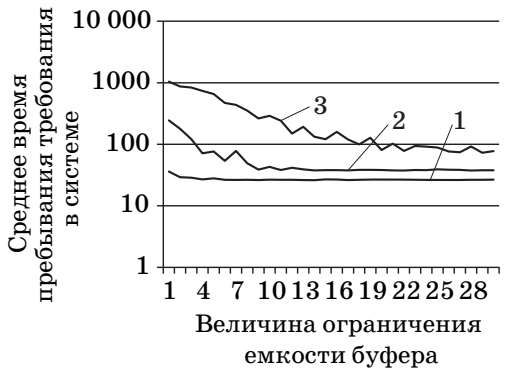
Объект *Release* освобождает после обслуживания требования захваченное им количество ресурсов, тем самым изменяется производительность системы. Соответственно, необходим пересчет времени обслуживания требований, находящихся в текущий момент на обслуживании. Рассматриваемый процесс был реализован с помощью следующего алгоритма:

1. Изымаются все требования, находящиеся на обслуживании в данный момент, со значением времени, необходимым до конца обслуживания (work.getRemainingTime()).

2. Время обслуживания каждого требования (delay) пересчитывается с учетом коэффициента падения производительности.

3. Все требования отправляются на обработку одновременно (enter) с заново рассчитанными параметрами.

4. Если система является не нагруженной (resourcePool.busy()=0) и существуют требования в очереди на обслуживание (queue.size()!=0), то



■ ***Рис. 2***. Зависимость среднего времени пребывания требования в системе от разрешенной величины буфера для значений интенсивности входящего потока: *1* — $\lambda = 0{,}04$; *2* — $\lambda = 0{,}05$; *3* — $\lambda = 0{,}06$; $\mu_0 = 0{,}066$

очередное требование из очереди отправляется на обслуживание.

Объект *Exit* служит для учета времени окончания обслуживания требования.

Результаты проведенных экспериментов, представленных на рис. 2, позволяют высказать гипотезу о существовании для значений $\lambda/\mu_0 \sim 1$ величины очереди (емкости буфера) $r_b(\lambda, \mu)$, минимизирующей функцию критерия «среднее время пребывания требования в системе». Дисперсия случайной величины также минимизируется, начиная с порогового значения очереди r_b.

Заключение

Предложенная модель позволяет использовать в качестве управляющего параметра длину очереди требований, ожидающих обработку, в зависимости от интенсивности потока заявок и закона падения производительности.

Дальнейшее исследование $r_b(\lambda, \mu)$ для системы необходимо проводить для значений $\lambda / \mu_0 > 1 + \varepsilon$, проводить расчет среднего времени пребывания требования в очереди, вводить функцию потерь, учитывающую затраты на содержание очереди.

Литература

1. **Kleinrock L.** Time-Shared Systems: a Theoretical Treatment // Journal of the ACM. 1967. Vol. 14. N 2. P. 242–261.
2. **Cohen J. W.** The Multiple Phase Service Network with Generalized Processor Sharing // Acta Information. 1979. Vol. 12. N. 3. P. 245–284.
3. **Guillemin F., Boyer J.** Analysis of the M/M/1 Queue with Processor Sharing via Spectral Theory // Queueing Systems. 2001. Vol. 39. P. 377–397.
4. **Kim J., Kim B.** Sojourn Time Distribution in the M/M/1 Queue with Discriminatory Processor-Sharing // Performance Evaluation. 2004. Vol. 58. N 4. P. 341–365.
5. **Рыков В. В.** Управляемые системы массового обслуживания // Итоги науки и техники. Сер. Теория вероятностей и математическая статистика. Теория кибернетики. 1975. Т. 12. С. 43–153.
6. **Bansal N.** Analysis of the M/G/1 Processor-Sharing Queue with Bulk Arrivals // Operations Research Letters. 2003. Vol. 31. N 5. P. 401–405.
7. **Avrachenkov K. E., Ayesta U., Brown P.** Batch Arrival Processor-Sharing with Application to Multi-Level Processor-Sharing Scheduling // Queueing Systems. 2005. Vol. 50. P. 459–480.
8. **Brandt A., Brandt M.** Workload and Busy Period for M/GI/1 with a General Impatience Mechanism // Queueing Systems. 2013. Vol. 75. P. 189–209.
9. **Yamamuro K.** The Queue Length in an M/G/1 Batch Arrival Retrial Queue // Queueing Systems. 2012. Vol. 70. P. 187–205.
10. **Sigman K.** Exact Simulation of the Stationary Distribution of the FIFO M/G/c Queue: the General Case for $\rho < c$ // Queueing Systems. 2012. Vol. 70. P. 37–43.
11. **Zhang Z. G., Tian N.** Analysis on Queueing Systems with Synchronous Vacations of Partial Servers // Performance Evaluation. 2003. Vol. 52. P. 269–282.
12. **Blom J., Kella O., Mandjes M., Thorsdottir H.** Markov-Modulated Infinite-server Queues with General Service Times // Queueing Systems. 2014. Vol. 76. P. 403–424.
13. **Baykal-Gursov M., Xiao W.** Stochastic Decomposition in M/M/∞ Queues with Markov Modulated Service Rates // Queueing Systems. 2004. Vol. 48. P. 75–88.
14. **Rykov V. V.** Monotone Control of Queueing Systems with Heterogeneous Servers // Queueing Systems. 2001. Vol. 37. P. 391–403.
15. **de Vericourt F., Zhou Y.-P.** On the Incomplete Results for the Heterogeneous Servers Problem // Queueing Systems. 2006. Vol. 52. P. 189–191.
16. **Rykov V. V., Efrosinin D. V.** Optimal Control of Queueing Systems with Heterogeneous Servers // Queueing Systems. 2004. Vol. 46. P. 389–407.
17. **Евфросинин Д. В., Рыков В. В.** К анализу характеристик производительности СМО с неоднородными приборами // Автоматика и телемеханика. 2008. № 1. С. 64–82.
18. **Shiryaev A. N.** Problems in Probability. Problem Books in Mathematics. — N. Y.: Springer, 2012. — 427 p.
19. **Бочаров П. П., Печенкин А. В.** Теория массового обслуживания / РУДН. — М., 1995. — 529 с.

UDC 519.876.2, 519.876.5

Optimal Control of Queues in Queueing Systems with Limited Performance

Zubok D. A.[a], PhD, Phys.-Math., Deputy Head of Chair, zubok@mail.ifmo.ru
Maiatin A. V.[a], PhD, Pedagogic, Associate Professor, mayatin@mail.ifmo.ru
[a]Saint-Petersburg National Research University of Information Technologies, Mechanics and Optics, 49, Kronverkskii St., 197101, Saint-Petersburg, Russian Federation

Purpose: For queuing systems with infinite servers, there is no detailed study of the analogues of the threshold nature of the optimal policy queueing systems established for queuing systems with non-homogeneous infinite servers. At the same time, the model systems with infinite servers provide a satisfactory description of the computing nodes with multi-threading. The aim of this study is to discover

the threshold nature of optimal policy control queueing systems for infinite servers with the performance of a server depending on the amount of the requests in the system. **Methods**: To describe queuing, we use processes with discrete time and a finite number of states. In describing the outgoing flow, an autoregressive scheme is used. **Results:** An approach was proposed to the solution of the basic equations of the model. A simulation experiment was carried out to test the hypothesis of the threshold nature of queue management. In the approximation of the processes with discrete time, equations of autoregression and moving average were derived to describe the incoming and outgoing flows of the queueing process. It is assumed that the order of closure compliance coincides with the order in which they are received to be performed. To test the hypothesis of the existence of an optimal queueing discipine, simulation experiments were conducted. In the experiments, the intensity of the incoming flow took values both less and greater than the performance of the system, meeting only one requirement. The simulation experiments showed that the mean time a requirement spends in the system depends on the allowed queue size and the intensity of the incoming stream. Thus, to improve the performance for a given flow rate, we need to uniquely determine the lower bound for the queue value, starting from which the performance of the system does not grow. **Practical relevance:** The proposed mathematical and simulation models can be used for studying single and multi-phase systems with various distributions of run-time requirements and various patterns of productivity drop.

Keywords — Queueing Systems, Infinite Servers, Markov Decision Processes, Servering Quality, Simulation Model.

References

1. Kleinrock L. Time-Shared Systems: a Theoretical Treatment. *Journal of the ACM*, 1967, vol. 14, no. 2, pp. 242–261.
2. Cohen J. W. The Multiple Phase Service Network with Generalized Processor Sharing. *Acta Information*, 1979, vol. 12, no. 3, pp. 245–284.
3. Guillemin F., Boyer J. Analysis of the M/M/1 Queue with Processor Sharing via Spectral Theory. *Queueing Systems*, 2001, vol. 39, pp. 377–397.
4. Kim J., Kim B. Sojourn Time Distribution in the M/M/1 Queue with Discriminatory Processor-Sharing. *Performance Evaluation*, 2004, vol. 58, no. 4, pp. 341–365.
5. Rykov V. V. Controllable Queuing Systems. *Itogi nauki i tekhniki. Seriia "Teoriia veroiatnostei i matematicheskaia statistika. Teoriia kibernetiki"*, 1975, vol. 12, pp. 45–152 (In Russian).
6. Bansal N. Analysis of the M/G/1 Processor-Sharing Queue with Bulk Arrivals. *Operations Research Letters*, 2003, vol. 31, no. 5, pp. 401–405.
7. Avrachenkov K. E., Ayesta U., Brown P. Batch Arrival Processor-Sharing with Application to Multi-Level Processor-Sharing Scheduling. *Queueing Systems*, 2005, vol. 50, pp. 459–480.
8. Brandt A., Brandt M. Workload and Busy Period for M/GI/1 with a General Impatience Mechanism. *Queueing Systems*, 2013, vol. 75, pp. 189–209.
9. Yamamuro K. The Queue Length in an M/G/1 Batch Arrival Retrial Queue. *Queueing Systems*, 2012, vol. 70, pp. 187–205.
10. Sigman K. Exact Simulation of the Stationary Distribution of the FIFO M/G/c Queue: the General Case for $\rho<c$. *Queueing Systems*, 2012, vol. 70, pp. 37–43.
11. Zhang Z. G., Tian N. Analysis on Queueing Systems with Synchronous Vacations of Partial Servers. *Performance Evaluation*, 2003, vol. 52, pp. 269–282.
12. Blom J., Kella O., Mandjes M., Thorsdottir H. Markov-Modulated Infinite-Server Queues With General Service Times. *Queueing Systems*, 2014, vol. 76, pp. 403–424.
13. Baykal-Gursov M., Xiao W. Stochastic Decomposition in M/M/∞ Queues with Markov Modulated Service Rates. *Queueing Systems*, 2004, vol. 48, pp. 75–88.
14. Rykov V. Monotone Control of Queueing Systems with Heterogeneous Servers. *Queueing Systems*, 2001, vol. 37, pp. 391–403.
15. de Veri court F., Zhou Y.-P. On the Incomplete Results for the Heterogeneous Servers Problem. *Queueing Systems*, 2006, vol. 52, pp. 189–191.
16. Rykov V. V., Efrosinin D. V. Optimal Control of Queueing Systems with Heterogeneous Servers. *Queueing Systems*, 2004, vol. 46, pp. 389–407.
17. Efrosinin D. V., Rykov V. V. On Performance Characteristics for Queueing Systems with Heterogeneous Servers. *Automation and Remote Control*, 2008, no. 1, pp. 64–82 (In Russian).
18. Shiryaev A. N. *Problems in Probability. Problem Books in Mathematics.* New York, Springer Publ., 2012. 427 p.
19. Bocharov P. P., Pechinkin A. V. *Teoriia massovogo obsluzhivaniia* [Theory of Queueing Systems]. Moscow, RUDN Publ., 1995. 529 p. (In Russian).