

ИССЛЕДОВАНИЕ И РАСЧЕТ ОСНОВНЫХ ПАРАМЕТРОВ ВЫЧИСЛИТЕЛЬНОЙ МАШИНЫ ДЛЯ ИНЖЕНЕРНЫХ РАСЧЕТОВ

В. Д. Кожухов

Харьков

В статье «Об одном способе повышения эффективности вычислительной машины для инженерных расчетов», опубликованной в сборнике «Приборы и системы автоматики», вып. 5, рассматривается цифровая вычислительная машина для инженерных расчетов с оптимальной ценой эффективного быстродействия. Это ЦВМ с полным распределением времени, имеющая систему автономных пультов вычислителей. В этой же статье приведены результаты исследования зависимости между величиной номинального быстродействия вычислительной машины и возможным количеством автономных пультов, а также результаты информационного анализа инженерных задач, решавшихся на ряде предприятий г. Харькова.

Статья посвящена исследованию зависимости между величиной возможного времени простоя задачи в очереди на решение и величиной номинального быстродействия вычислительной машины в зависимости от состава решаемых на ней задач. Кроме того, в статье предлагается методика определения объема буферной памяти (БЗУ) машины.

Пользуясь терминологией теории массового обслуживания [1 — 7], будем рассматривать вычислительную машину как некоторый обслуживающий аппарат, обрабатывающий одновременно не более одной заявки, в который заявки на обслуживание могут поступать по N каналам.

В силу того, что с каждого пульта вычислителя одновременно может вводиться только одна задача, т. е. вероятность поступления по одному каналу одновременно двух и более заявок равна нулю, входные потоки заявок являются ординарными.

Кроме того, каждый из входящих потоков можно рассматривать составленным из двух самостоятельных потоков: потока задач записи алгоритмов, решения которых производятся без применения операций условного или безусловного переходов или циклирования (задачи первого класса) и потока задач, для записи алгоритмов которых необходимы эти операции (задачи второго класса). Будем считать, что каждый из этих потоков обладает свойством стационарности, т. е. для любой группы из конечного числа достаточно малых отрезков времени вероятность получения определенного числа требований на протяжении каждого из них зависит только от этих чисел и от длительности промежутков времени, но не изменяется от сдвига всех временных интервалов на одну и ту же величину. В частности, вероятность появления k требований в течение промежутка времени от T до $T + t$ не зависит от T и является функцией только переменных t и k .

Нисколько не уменьшая общности изложения, объединим все входящие потоки задач, поступающие по различным каналам, в два потока — суммарный поток задач первого класса и суммарный поток задач второго класса.

Исходя из предельной теоремы о суммировании ординарных и стационарных потоков [4], заключаем, что оба полученных потока будут простейшими.

Как известно из теории массового обслуживания [1—7], простейший поток заявок можно полностью определить, если будет известен параметр этого потока λ . Последний определяется как математическое ожидание количества заявок на обслуживание, поступающих в единицу времени. Зная параметр λ , по формуле Пуассона

$$V_k(t) = e^{-\lambda t} \frac{(\lambda t)^k}{k!}$$

$$(k = 0, + \dots)$$

можно определить вероятность того, что в промежутке времени длиной t поступит точно k требований.

Однако для определения среднего времени простоя заявки в очереди на обслуживание недостаточно знать только характеристики входящего потока. Для этого необходимо знать также и характеристику интенсивности обслуживания заявок в системе. Такой характеристикой является среднее обслуживание одной заявки, для определения которого часто делают допущение [2, 3], что закон распределения времени обслуживания показательный, т. е.

$$F(t) = 1 - e^{-\nu t},$$

где ν — положительная постоянная.

В этом случае математическое ожидание времени обслуживания одной заявки

$$\mu_1 = \int_0^{\infty} t dF(t) = \int_0^{\infty} t d(1 - e^{-\nu t}) = \nu \int_0^{\infty} t e^{-\nu t} dt = \frac{1}{\nu}.$$

Мы будем определять среднее время обслуживания одной заявки, исходя из быстродействия вычислительной машины и количества операций, которые необходимо выполнить для решения одной задачи.

В выбранной нами структуре системы величина времени обслуживания заявки не зависит от времени ввода и вывода результатов задачи в машину. Величина времени обслуживания зависит только от времени перезаписи массива чисел на БЗУ в ОЗУ, от времени счета задачи и от времени перезаписи результатов вычислений из ОЗУ в БЗУ.

Учитывая, что интенсивность потока задач первого класса велика и может в тысячи раз превышать интенсивность потока задач второго класса, весьма рационально в целях экономии машинного времени добиться, чтобы во время перезаписи информации задач первого класса вычислительная машина продолжала счет задачи второго класса. Этого можно достичь, выделив в определенной памяти специальный массив ячеек, заведомо достаточный для записи максимальной задачи первого класса и предусмотрев возможность обмена информацией между БЗУ и этим участком ОЗУ в момент решения задачи второго класса. В этом случае время обслуживания заявки на решение задачи первого класса будет равно лишь времени счета задачи, т. е.

$$\mu_{1.1} = \frac{K_{01}}{n},$$

где K_{01} — количество операций задачи первого класса;

n — номинальное быстродействие вычислительной машины.

Для задач второго класса время счета

$$\mu_{1.2} = 2\tau_n + \frac{K_{02}}{n},$$

где τ_n — время перезаписи БЗУ-ОЗУ;

K_{02} — количество операций, которые необходимо выполнить машине для решения задачи второго класса.

Если известно время обслуживания одной заявки и математическое ожидание количества заявок в единицу времени при условии, что входящий поток заявок есть пуассоновский, время ожидания очереди на обслуживание определяется по формуле Хинчина [7]:

$$\bar{\gamma} = \frac{\lambda}{2(1-\alpha)} \frac{\mu_2}{\mu_1},$$

где μ_2 — математическое ожидание квадрата длительности обслуживания;
 α — вероятность того, что в произвольно выбранный момент времени машина занята, т. е.

$$\alpha = \lambda \mu_1.$$

Как видно из формулы, время простоя в очереди $\bar{\gamma}$ зависит не только от средней длительности времени обслуживания, но и от дисперсии. Именно поэтому для задач первого и второго класса должны быть созданы две отдельные очереди. Можно ввести приоритет для задач первого класса. В случае, если на машине решается задача второго класса, то при занесении в ОЗУ задачи первого класса решение задачи второго класса приостанавливается — решается задача первого класса, затем возобновляется решение задачи второго класса. При такой организации работы задачи второго класса никак не влияют ни на время решения задачи первого класса, ни на время их простоя в очереди. Что же касается задач второго класса, то здесь необходимо вносить поправки в величины μ_1 и μ_2 в зависимости от количества задач первого класса, которые будут решены во время счета соответствующих задач второго класса.

В табл. 1 приведены значения величин среднего времени ожидания очереди для задач первого класса в зависимости от номинального быстродействия вычислительной машины для различных значений параметра λ . Как видно из таблицы, время ожидания очереди для задач первого класса даже для машин со сравнительно невысоким номинальным быстродействием настолько мало, что практически им можно пренебречь.

Таблица 1

λ	50 оп/сек	100 оп/сек	250 оп/сек	500 оп/сек
1 тыс.	$4,16 \cdot 10^{-4}$	$1,04 \cdot 10^{-4}$	$1,66 \cdot 10^{-5}$	$4,16 \cdot 10^{-6}$
10 тыс.	$4,25 \cdot 10^{-3}$	$1,05 \cdot 10^{-3}$	$1,66 \cdot 10^{-4}$	$4,25 \cdot 10^{-5}$
100 тыс.	$5,17 \cdot 10^{-2}$	$1,15 \cdot 10^{-2}$	$1,6 \cdot 10^{-3}$	$5,17 \cdot 10^{-4}$
1 млн.	—	$3,75 \cdot 10^{-1}$	$1,2 \cdot 10^{-2}$	$3,48 \cdot 10^{-3}$

Перейдем теперь к определению среднего времени ожидания очереди для задач второго класса. Оно будет зависеть от интенсивности входящего потока задач второго класса, от статистики решаемых задач, от быстродействия вычислительной машины и от того, какое количество задач первого класса будет решено во время счета одной задачи второго класса.

Возьмем следующие значения интенсивностей входящего потока задач второго класса:

$$\text{одна задача в сутки } \lambda = \frac{1}{7,2 \cdot 10^{-4}};$$

$$\text{три задачи в сутки } \lambda = \frac{1}{2,4 \cdot 10^{-4}};$$

шесть задач в сутки $\lambda = \frac{1}{1,2 \cdot 10^4}$;

десять задач в сутки $\lambda = \frac{1}{7,2 \cdot 10^3}$;

двадцать пять задач в сутки $\lambda = \frac{1}{2,8 \cdot 10^2}$;

пятьдесят задач в сутки $\lambda = \frac{1}{1,4 \cdot 10^2}$;

сто задач в сутки $\lambda = \frac{1}{720}$;

двести задач в сутки $\lambda = \frac{1}{360}$.

Для определения среднего времени решения одной задачи будем пользоваться сведениями, полученными во время информационного анализа инженерных задач, проведенного автором на ряде предприятий Харькова [8]. Среднее значение времени простоя в очереди задачи второго класса для машин с различным номинальным быстродействием приведены в табл. 2—4. В табл. 2 — для случая, когда на одну задачу второго класса приходится 10^3 задач первого класса; в табл. 3 — для 10^4 ; в табл. 4 — для 10^5 . Время дано в секундах.

В табл. 2—4 имеются пропуски, которые объясняются тем, что для данной интенсивности входящего потока задач быстродействие вычислительной машины недостаточно — в этом случае возможно образование бесконечной очереди.

Таблица 2

n	$\lambda = \frac{1}{720000}$	$\lambda = \frac{1}{24000}$	$\lambda = \frac{1}{12000}$	$\lambda = \frac{1}{7200}$	$\lambda = \frac{1}{2800}$	$\lambda = \frac{1}{1440}$	$\lambda = \frac{1}{720}$	$\lambda = \frac{1}{360}$
50	406 000	—	—	—	—	—	—	—
100	30 600	—	—	—	—	—	—	—
250	2 850	14 500	—	—	—	—	—	—
500	645	2 380	7270	40 600	—	—	—	—
1000	152	502	1280	2 550	—	—	—	—
2500	24	74,7	158	285	1030	8130	—	—
5000	6	18,2	37,6	64,5	189	515	4050	—
10 000	1,46	4,43	9,48	15,2	41,1	92,8	109	2020
25 000	0,237	0,712	1,43	2,4	6,21	12,9	28,5	71,7
50 000	0,059	0,178	0,356	0,595	1,51	3,08	6,46	14,3
100 000	0,015	0,0438	0,0877	0,146	0,369	0,745	1,52	3,19

Таким образом, пользуясь формулой Хинчина и задаваясь приемлемым значением среднего времени простоя задачи в очереди на обслуживание, можно определить поправку к величине номинального быстродействия вычислительной машины.

Перейдем теперь к определению буферной памяти. БЗУ является буферным устройством при обмене информацией между быстродействующей вычислительной машиной и медленно действующими внешними устройствами.

Можно предложить два варианта работы БЗУ. По первому из них в БЗУ нет жестко закрепленных за внешними устройствами массивов памяти, а необходимый участок выделяется по поступлению заявки. В этом случае при вводе в вычислительную машину информация, поступающая с внешних устройств, формируется в БЗУ в некоторые стандартные массивы, содержимое которых по мере их заполнения перепиывается в накопитель на магнитной ленте. Как только занесение входной

информации будет закончено, задаче присваивается номер очереди на решение, о чем делается соответствующая запись в массиве автодиспетчера очередности. Таким образом, все занесенные алгоритмы хранятся в НМЛ, а в БЗУ хранится лишь алгоритм задачи, стоящей первой в очереди на решение. Результаты вычислений переписываются из ОЗУ в БЗУ, откуда посимвольно выдаются соответствующему внешнему устройству.

Таблица 3

n	$\frac{1}{72\,000}$ $\lambda =$	$\frac{1}{24\,000}$ $\lambda =$	$\frac{1}{12\,000}$ $\lambda =$	$\lambda = \frac{1}{7200}$	$\lambda = \frac{1}{2880}$	$\lambda = \frac{1}{1440}$	$\lambda = \frac{1}{720}$	$\lambda = \frac{1}{360}$
50	473 000	—	—	—	—	—	—	—
100	35 400	—	—	—	—	—	—	—
250	2 930	15 200	—	—	—	—	—	—
500	663	2 460	7510	47 300	—	—	—	—
1000	156	523	1320	2 680	—	—	—	—
2500	24,6	76,5	162	293	1070	9460	—	—
5000	6,1	18,6	38,3	66,2	193	536	4730	—
10 000	1,52	4,6	9,33	15,8	42,5	96,8	127	2370
25 000	0,242	0,729	1,47	2,46	6,32	13,2	29,3	74,4
50 000	0,0605	0,182	0,365	0,61	1,55	3,16	6,63	15,2
100 000	0,0151	0,0455	0,0911	0,152	0,383	0,774	1,58	3,31

Таблица 4

n	$\frac{1}{72\,000}$ $\lambda =$	$\frac{1}{24\,000}$ $\lambda =$	$\frac{1}{12\,000}$ $\lambda =$	$\lambda = \frac{1}{7200}$	$\lambda = \frac{1}{2880}$	$\lambda = \frac{1}{1440}$	$\lambda = \frac{1}{720}$	$\lambda = \frac{1}{360}$
50	—	—	—	—	—	—	—	—
100	37 500	—	—	—	—	—	—	—
250	3610	23 000	—	—	—	—	—	—
500	799	3 130	11 500	—	—	—	—	—
1000	188	576	1560	3750	—	—	—	—
2500	29,2	91,4	196	361	1500	—	—	—
5000	7,23	22,2	45,8	79,3	242	751	—	—
10 000	1,79	5,38	11,1	18,8	51,3	121	137	—
25 000	0,286	0,862	1,73	2,92	7,53	15,9	36,1	98,3
50 000	0,0716	0,215	0,432	0,723	1,84	3,77	7,98	18,1
100 000	0,0179	0,0536	0,107	0,179	0,452	0,917	1,88	3,99

В этом случае задача определения объема БЗУ сводится к отысканию величины стандартных массивов, необходимых для различных по своему назначению передач информации, к определению необходимого количества массивов каждого стандарта, к оценке вероятностей таких состояний системы, при которых объем БЗУ будет недостаточен и к принятию мер, с помощью которых вероятность таких состояний можно свести к минимуму.

Вначале определим величину стандартного массива, необходимого для записи входной информации. Поскольку занесение с пульта входной информации о задаче второго класса производится довольно долго, а уже занесенная задача еще некоторое время стоит в очереди на решение, то для уменьшения объема БЗУ входная информация накапливается и хранится в накопителе на магнитной ленте. В этом случае величина стандартного массива, предназначенного для занесения информации о задачах второго класса, может быть произвольной. Что же касается

величины стандартного массива, необходимого для записи входной информации о задаче первого класса, то здесь следует заметить, что при записи в стандартные массивы различных задач часть массива не будет использоваться. Кроме того, помимо размещения кодов, составляющих собственно символ входной информации, в массив будут записываться и некоторые служебные коды (например, коды, определяющие номер внешнего устройства, с которого заносится информация, номер массива и т. п.).

Условившись каким-либо образом кодировать символы входной информации и договорившись о способе записи кодов в память, а также используя статистическое распределение количества символов входной информации, при решении различных задач можно определить такой объем массива, при котором соотношение числа разрядов, занятых хранением кодов задачи, к общему количеству разрядов в массиве будет максимальным. Если исходить из требования экономии памяти, то найденный массив можно принять за стандартный, однако, может оказаться, что пойдя на незначительное увеличение потерь, можно добиться такого положения, когда большинство задач первого класса будет помещаться в один стандартный массив, что, несомненно, облегчит работу внешней памяти. В связи с этим выберем величину стандартного массива, выделяемого в БЗУ для занесения входной информации, равной массиву, необходимому для записи максимально большой задачи первого класса.

После выбора величины стандартного массива определим их количество. С этой целью представим участок буферной памяти, необходимый для занесения входной информации, как совокупность обслуживающих аппаратов, а каждый стандартный массив — одним из таких аппаратов. Заявкой на обслуживание будем считать заявку на выделение стандартного массива памяти. Учитывая, что поток заявок на обслуживание в вычислительную машину является пуассоновским, воспользуемся тем фактом, что для многоканальной системы обслуживания вероятность того, что в установившемся процессе было занято k обслуживающих аппаратов (стандартных массивов), независимо от начального состояния, система равна:

$$P_k = \frac{1}{k!} \left(\frac{\lambda}{v} \right)^k e^{-\frac{\lambda}{v}},$$

$$(k = 0, 1, 2, \dots)$$

где λ — параметр входящего потока;

$\frac{1}{v}$ — среднее время обслуживания одной заявки — среднее время занесения входной информации о всей задаче [6].

Вероятность P_k для различных значений k по мере увеличения k будут вначале возрастать и для некоторого $k = i$ достигнут максимальной величины. При дальнейшем увеличении k вероятность P_k начнет убывать, асимптотически приближаясь к нулю. Выберем количество стандартных массивов таким, чтобы k было больше i и при этом P_k было достаточно мало.

Определим теперь объем участка буферной памяти, необходимый для вывода на пульт вычислителей результатов решения задач. Здесь за стандартный можно принять массив, необходимый для записи результатов вычислений «средней» задачи второго класса. Количество стандартных массивов можно определить точно так же, как и количество стандартных массивов, необходимых для записи входной информации.

Перейдем теперь к определению суммарного массива, необходимого для хранения задачи, стоящей первой на очереди, задачи, заносимой с перфокарт и для хранения алгоритмов, выводимые на перфокарты.

Прежде всего заметим, что алгоритмы, выводимые на перфокарты, могут образовывать очередь у перфоратора. Если среднее время счета задачи вычислительной машины будет меньше среднего времени перфорации, то длина очереди может бесконечно возрастать. Чтобы этого не произошло, можно подключить к машине несколько перфораторов.

Пусть требований на перфорацию больше, чем перфораторов, тогда (считая поток заявок на перфорирование пуассоновским) вероятность того, что поступит k требований на перфорацию, равна [6]

$$P_k = \frac{m!}{n^{k-n} n! (m-k)!} \left(\frac{\lambda}{v}\right)^k P_0,$$

где λ — параметр входящего потока;

$\frac{1}{v}$ — среднее время перфорирования одного алгоритма;

m — наибольшее возможное число алгоритмов, предназначенных для перфорации и одновременно хранимых в памяти машины;

n — количество перфораторов;

P_0 — вероятность того, что перфораторы свободны.

$$P_0 = \left[\sum_{k=0}^m \frac{m!}{k!(m-k)!} \left(\frac{\lambda}{v}\right)^k + \sum_{k=n+1}^m \frac{m!}{n^{k-n} n! (m-k)!} \left(\frac{\lambda}{v}\right)^k \right]^{-1}.$$

Так как количество перфорируемых алгоритмов не может быть больше числа перфораторов, то имеет смысл хранить в БЗУ не более $m = n$ алгоритмов.

Присваивая k последовательный ряд значений $k = n, k = n-1, \dots$, подсчитываем для каждого случая соответствующую ему вероятность P_k . Задавшись приемлемой величиной P_k , примем соответствующее ей значение k за количество алгоритмов, место для хранения которых должно быть выделено в БЗУ.

Таким образом, в БЗУ должно быть отведено место для хранения информации об одной задаче, стоящей первой в очереди на решение, для хранения информации об одной задаче, заносимой с перфокарт и для хранения k алгоритмов, выдаваемых на перфокарты.

Для окончательного определения величины суммарного массива, необходимого для хранения входной информации $k+2$ задач, поступим следующим образом. Подсчитывают вероятность того, что, согласно статистическим рядам распределения количества символов входной информации, которое необходимо занести в ЦВМ для решения одной задачи второго класса, информация о $k+2$ задачах поместится в 1, 2... стандартных массивах, величина которых равна величине массива, необходимого для записи «средней» задачи второго класса. Получив в результате такого перебора достаточно большое значение вероятности, можно соответствующий ему объем памяти принять за искомый.

Описанный выше способ работы БЗУ обладает важным достоинством — он обеспечивает наиболее полное использование памяти, однако, при этом в БЗУ необходим автоматический диспетчер памяти, который должен распределять объем буферной памяти в зависимости от количества заявок, поступивших в текущий момент. Кроме того, в какой-либо момент времени при переполнении памяти может оказаться, что вновь поступившая заявка не будет выполнена, что приведет к ее потере. Этих недостатков лишено БЗУ, у которого за каждым внешним устройством жестко закреплен некоторый стандартный массив.

Работа в таком БЗУ может быть организована следующим образом. Входная информация (о задаче первого или второго класса) заносится

с пульта вычислителя в стандартный массив, отведенный для этого пульта, и по мере его заполнения переписывается в НМЛ. После занесения входной информации задача становится в очередь на решение.

Результаты вычислений переписываются из ОЗУ на ленту, а оттуда, по мере необходимости, в стандартный массив пульта, откуда посимвольно — на печать.

В этом случае объем суммарного массива, необходимого для хранения задачи, стоящей первой на очереди, задачи, заносимой с перфокарт и для хранения алгоритмов, выводимых на перфокарты, определяются так же, как в предыдущем случае, а объем суммарного массива, состоящего из массивов, закрепленных за пультами вычислителей, определим из следующих соображений.

Для работы с пультами вычислителей БЗУ должно иметь N стандартных массивов (по количеству пультов).

Чтобы БЗУ было минимальным по объему, естественно делать стандартные массивы пультов как можно меньшими, однако при этом увеличивается частота обращения к НМЛ и может случиться, что заявки на перезапись в НМЛ будут образовывать недопустимо длинную очередь.

Накопитель на магнитной ленте можно считать однолинейной обслуживающей системой. Допустим при этом, что время перезаписи в НМЛ стандартно. В результате того, что поток задач, поступающих в машину, является Пуассоновским, количество заявок, поступающих в НМЛ, также распределено по закону Пуассона. При этом вероятность, что вновь поступившему требованию на перезапись придется ожидать время, большее, чем t , равна [7]:

$$P\{\gamma < t\} = e^{\lambda t} (1 - \lambda \tau) \sum_{k=0}^n e^{-\lambda k \tau} \frac{(k\tau - t)^k}{k!} \lambda k,$$

где λ — поток заявок на обращение к НМЛ;

τ — длительность перезаписи.

Целое, неотрицательное число n определяется неравенствами

$$n\tau < t \leq (n+1)\tau.$$

Для облегчения работы НМЛ потребуем, чтобы величина стандартного массива, выделяемого для занесения входной информации, была достаточна для записи задач первого класса. В этом случае информация о задачах первого класса в НМЛ не переписывается вовсе.

Тогда поток заявок на НМЛ будет состоять из потока перезаписей, производимых во время занесения входной информации (количество символов, заносимых в единицу времени $\lambda_2 K_{c2}$; потока заявок задач, которые ставятся первыми в очередь на решение (количество символов, перезаписываемых в единицу времени $\lambda_2 K_{c2}$); потока заявок, возникающего при перезаписи результатов вычислений из ОЗУ в НМЛ (количество символов, перезаписываемых в единицу времени $\lambda_2 K_{b2}$) и потока заявок, возникающего при перезаписи результатов вычислений из НМЛ в БЗУ для их последующей печати (перезаписываемое количество символов в единицу времени $\lambda_2 K_{b2}$). Из этой информации лишь $\lambda_2 K_{c2} + 2\lambda_2 K_{b2}$ будет проходить через массивы пультов, так как задачи, которые ставятся первыми в очередь на решение, заносятся в свой массив БЗУ. По той же причине интенсивность потока заявок, проходящих через массивы пультов, равна интенсивности общего потока заявок на НМЛ λ_n минус интенсивность потока заявок задач, ставящихся первыми в очередь на решение. Разделив суммарный объем информации, перезаписываемый в единицу времени через массивы пультов на интенсивность этого потока, получим

объем информации, перезаписываемой во время одной заявки, т.е. получим объем массива пульта

$$Q_{\text{мп}} = \frac{\lambda_2 K_{c2} + 2\lambda_2 K_{b2}}{\lambda_n - \lambda_2}.$$

ЛИТЕРАТУРА

1. Н. П. Брусенко. Решение задач теории массового обслуживания методом моделирования на электронных цифровых вычислительных машинах. Сб. «Проблемы передачи информации», вып. 9. Изд-во АН СССР, 1961.
 2. Г. П. Башарин. Об одноканальной обслуживающей системе с конечной очередью и неоднородным входным потоком. Труды ЦНИИКА, вып. 12.
 3. А. А. Вдовин. Исследование некоторых двухфазных систем массового обслуживания. Труды ЦНИИКА, вып. 12.
 4. Б. В. Гнеденко, И. Н. Коваленко. Лекция по теории массового обслуживания. Изд-во «Квирту». 1963.
 5. Д. Риордан. Вероятностные системы обслуживания. Изд-во «Связь», 1966.
 6. В. Я. Розенберг, А. И. Прохоров. Что такое теория массового обслуживания. Изд-во «Советское радио», 1962.
 7. А. Я. Хинчин. Работы по математической теории массового обслуживания. Физматгиз, 1963.
 8. В. Д. Кожухов. Об одном способе повышения эффективности вычислительной машины для инженерных расчетов. Сб. «Приборы и устройства средств автоматки», вып. 5. Изд-во ХГУ, Харьков, 1967.
-