

УДК 004.852

## АППРОКСИМАЦИЯ ВЕРОЯТНОСТНО-ВРЕМЕННЫХ ХАРАКТЕРИСТИК СЛОЖНЫХ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ НА ОСНОВЕ РЕГРЕССИОННЫХ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ

Хабаров Р.С. (Санкт-Петербург), Левчик Э.С. (Москва),  
Лохвицкий В.А. (Санкт-Петербург)

В эпоху больших данных все больше основных вычислительных инфраструктур развертываются в распределенном режиме и неизбежно задействуют очереди Fork-Join для облегчения хранения и обработки крупномасштабных наборов данных. Например, Cassandra [1] и Dynamo [2], два популярных хранилища данных «ключ-значение», используют очереди Fork-Join для одновременного выполнения операций чтения и записи.

В системе очередей Fork-Join поступающая в систему заявка (в дальнейшем – задача) разветвляется на  $n$  подзадач, и каждая подзадача отправляется на один из свободных узлов. Результаты завершенных подзадач объединяются на центральном узле соединения. Когда интенсивность прибытия заявок высока, подзадаче, возможно, придется ждать обслуживания в подочереди своего узла в порядке «первым пришел – первым обслужен». Считается, что задание выполнено после того, как все результаты были получены в узле соединения.

На рисунке 1 приведен пример системы массового обслуживания (СМО) с дисциплиной Fork-Join и тремя каналами обслуживания. В данном примере подзадачи  $A_2$  и  $A_3$  задачи  $A$  закончили обслуживание и ожидают подзадачу  $A_1$ , а освободившиеся каналы обслуживания заняты подзадачами  $B_2$  и  $B_3$  задачи  $B$ . Как только подзадача  $A_1$  завершит обслуживание, произойдет объединение результатов, и задача  $A$  покинет систему.

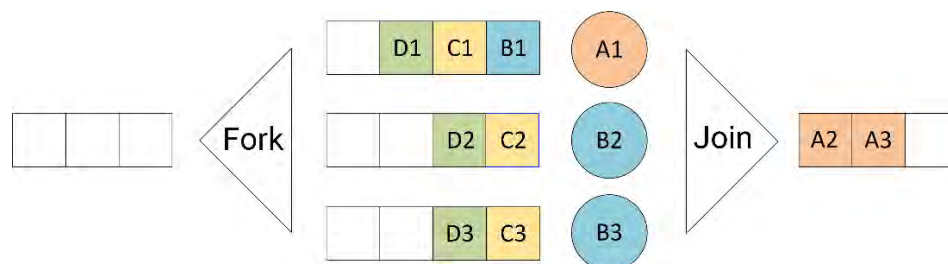


Рис. 1. Процесс обслуживания в СМО с дисциплиной Fork-Join

Задержка обычно является серьезной проблемой при создании и оптимизации кластеров больших данных. Например, в облачной платформе Amazon сервисы имеют требования к задержке, которые обычно измеряются на уровне 99,9-го перцентиля распределения [2]. Оценку перцентиля распределения можно получить лишь построив функцию распределения или, что удобнее на практике, дополнительную функцию распределения (ДФР). Одним из часто применяемых методов получения ДФР является аппроксимация распределением Вейбулла по начальным моментам распределения. Таким образом, при проектировании распределенных систем недостаточно ограничиться лишь оценкой среднего времени пребывания заявок в системе, необходима оценка высших моментов данного распределения.

Процессам Fork-Join посвящено достаточно большое количество работ [3-16]. Для систем Fork-Join точное выражение для среднего времени пребывания в системе с

произвольным распределением обслуживания было получено только для системы с двумя каналами [5, 6]. Для числа каналов  $n > 2$  и экспоненциального распределения времени обслуживания с помощью различных методов были получены аппроксимации среднего времени пребывания [8, 10, 11]. Приведем здесь кратко полученные результаты.

В [6] приведена точная формула для среднего время пребывания в системе *при  $n=2$  и экспоненциального распределения времени обслуживания*

$$v_1 = (H_2 - \rho/8)v_1^{(M)}, \quad (1)$$

где  $H_n = \sum_{i=1}^n 1/i$  и для  $n = 2$   $H_2 = 1.5$ ,

$\rho = \lambda/\mu$  – коэффициент загрузки системы,

$v_1^{(M)} = \frac{1}{(\mu-\lambda)}$  – среднее время пребывания в СМО М/М/1.

В [8] Nelson R. и Tantawi A.N. предложена аппроксимация для среднего времени пребывания в СМО Fork-Join с  $n$  каналами обслуживания

$$v_1 \approx \left[ \frac{H_n}{H_2} + \frac{4}{11} \left( 1 - \frac{H_n}{H_2} \right) \rho \right] \frac{12 - \rho}{8} \frac{1}{\mu - \lambda}, n \geq 2. \quad (2)$$

В [10] Varki E., Merchant A. и Chen H. предложили формулу

$$v_1 \approx \frac{1}{\mu} \left[ H_n + \frac{\rho}{2(1-\rho)} \left( \sum_{i=1}^n \frac{1}{i-\rho} + (1-2\rho) \sum_{i=1}^n \frac{1}{i(i-\rho)} \right) \right], n \geq 2. \quad (3)$$

В [11] Varma S. и Makowski A.M. предложен метод аппроксимации среднего времени пребывания на основе комбинации методов интерполяции высоких и низких значений коэффициентов загрузки системы:

$$v_1 \approx [H_n + (V_n - H_n)\rho] \frac{1}{\mu - \lambda}, \quad (4)$$

где  $V_n = \sum_i^n \binom{n}{i} (-1)^{i-1} \sum_{m=1}^n \binom{i}{m} \frac{(m-1)!}{i^{m+1}}$ .

В данной работе предлагается подход к получению начальных моментов времени пребывания заявок в СМО с дисциплиной Fork-Join на основе регрессионных моделей машинного обучения. В качестве последних предлагается использовать достаточно распространенные и зарекомендовавшие себя на практике модели Bagging [17], SVM [18] и Random Forest [19].

### Описание подхода к обучению моделей машинного обучения для СМО с дисциплиной Fork-Join

Подход к аппроксимации, основанный на машинном обучении (МО), предполагает наличие набора данных для обучения, который можно представить в виде таблицы, столбцы которой представляют собой так называемые признаки. В предложенном подходе предлагается получить такой набор данных средствами имитационного моделирования (ИМ). Традиционными вопросами при подготовке таких данных являются:

- 1) определение необходимого набора признаков;
- 2) определение необходимого количества записей в наборе данных.

Первая проблема связана с противоречием, которое заключается в том, что, с одной стороны, для обеспечения требуемой точности необходим учет как можно большего числа параметров (признаков), с другой стороны – излишнее их число сильно замедляет скорость обучения (это актуально в большей степени для модели машинного обучения SVM) и может привести к обратному эффекту – падению точности.

Вопрос накопления достаточного количества записей в наборе данных связан с несколькими проблемами:

- временными затратами на ИМ;
- замедлением скорости обучения моделей машинного обучения при большом наборе данных (это актуально в большей степени для моделей машинного обучения Random Forest и Bagging).

В данной работе предлагается в качестве признаков (параметров) для обучения выбрать следующие:

- число каналов обслуживания;
- начальные моменты распределения времени обслуживания в канале;
- интенсивность входного потока заявок (предполагается пуассоновский поток);
- коэффициент загрузки системы.

Число параметров предлагается расширить благодаря использованию квадратов приведенных величин, поскольку в (2-4) наблюдается нелинейная зависимость от упомянутых параметров. Также при подготовке набора данных предлагается произвести нормировку, а именно – предлагается нормировать интенсивность входного потока и начальные моменты распределения времени обслуживания в канале таким образом, чтобы значение среднего времени обслуживания всегда было равно единице, а значения остальных параметров масштабировались в соответствии с заданным коэффициентом загрузки системы и коэффициентов вариации времени обслуживания в канале. Также, для упрощения сбора данных, предлагается в качестве распределения времени обслуживания использовать Гамма-распределение, позволяющее задавать произвольный коэффициент вариации времени обслуживания. В данной работе мы ограничились значениями коэффициента вариации времени обслуживания  $v$  в диапазоне (0.1, 3.1), а значениями каналов обслуживания  $n$  (которое соответствует количеству подзадач) в диапазоне [2, 15), что достаточно для большинства практических задач. Коэффициент загрузки СМО  $\rho$  выбирался в диапазоне (0.1, 0.95). При каждой новой итерации ИМ указанные значения выбирались случайно в пределах упомянутых диапазонов. Всего произведено более 100 тыс. итераций ИМ. В таблице 1 приведен фрагмент полученного набора данных.

Таблица 1. Фрагмент набора данных для СМО Fork-Join

$\rho$	$n$	$v$	Начальные моменты времени обслуживания			$v_1$	$v_2$
			$b_1$	$b_2$	$b_3$		
0.724	9	2.863	1.000	9.199	160.1	43.82	2488
0.320	12	1.979	1.000	4.918	43.45	9.371	119.2
0.510	8	1.969	1.000	4.878	42.71	11.57	184.6
...							
0.534	13	0.482	1.000	1.233	1.806	3.190	12.51

0.618	10	0.522	1.000	1.273	1.967	3.661	17.11
0.259	7	2.000	1.000	4.998	44.96	6.707	69.52

Для аппроксимации второго начального момента времени пребывания заявок в системе использовались те же данные, однако в число признаков добавлялись значения среднего времени пребывания заявок в системе, а также квадраты данных значений. Т.е. создавались отдельные от полученных для средних времен пребывания модели МО Bagging, SVM, и Random Forest. Итого, в результате обучения по полученному набору данных получилось 6 моделей МО – три для среднего времени пребывания в СМО и три для второго начального момента времени пребывания. Таким образом, для получения оценок второго начального момента сперва находились оценки для среднего с помощью соответствующих моделей МО.

### Результаты аппроксимации

На рисунке 2 приведены результаты аппроксимации среднего времени пребывания заявок в системе, а также относительной ошибки его оценки по отношению к результатам ИМ, в зависимости от коэффициента загрузки  $\rho$  при  $n=5$  и  $v=1.0$ .

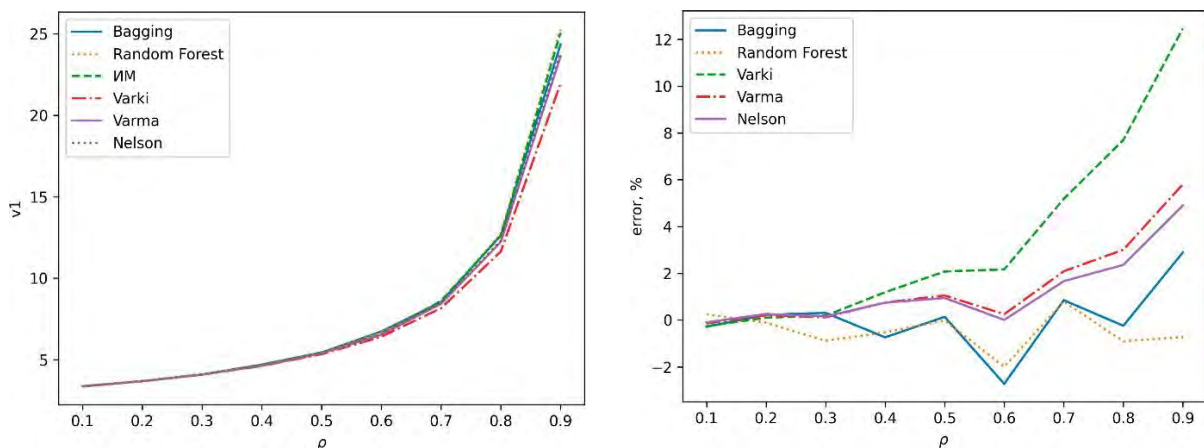


Рисунок 2. Сравнение аппроксимаций в зависимости от коэффициента загрузки при  $n=5$ ,  $v=1.0$

Результаты приведены для МО Random Forest, Bagging и формул (2-4). Данные ИМ выступали в качестве эталонных и соответствуют значениям, полученным после обслуживания 5 млн заявок. Модель МО SVM показала высокую погрешность оценок (до 60%), поэтому исключена из дальнейшего рассмотрения. Как можно наблюдать из полученных результатов, относительная погрешность оценок среднего времени пребывания для моделей МО находится в пределах 3-4%. Оценки аппроксимации по формулам (2) и (4) для невысоких значений коэффициента загрузки также находятся в этих пределах, однако при приближении коэффициента загрузки к значению 0.9 точность аппроксимации слегка уступает точности моделей МО. Точность оценок, полученных с помощью (3), значительно ниже вышеперечисленных.

Демонстрируемая точность моделей МО вполне приемлема, особенно если учесть известный факт увеличения дисперсии оценок среднего времени ожидания, полученных с помощью ИМ, при увеличении коэффициента загрузки. Кроме того, согласно [20] для оценок первого момента (среднего времени пребывания) считается допустимой погрешность в 20%, второго – 40 %, а третьего – 60 %.

На рисунке 3 приведены результаты аппроксимации среднего времени пребывания в системе, а также относительной ошибки его оценки по отношению к результатам ИМ, в зависимости от числа каналов при  $\rho=0.7$  и  $\nu=1.0$ .

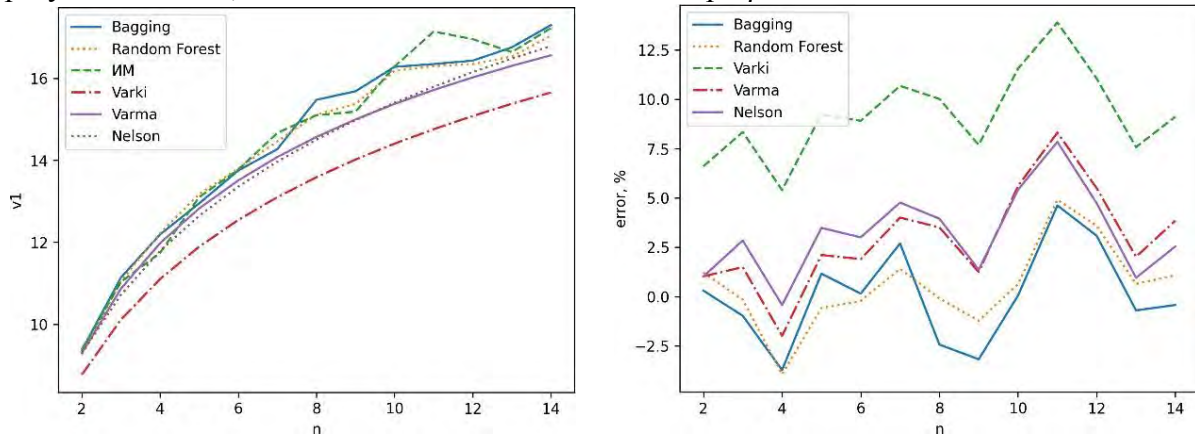


Рисунок 3. Сравнение аппроксимаций в зависимости от числа каналов обслуживания при  $\rho=0.7$ ,  $\nu=1.0$

Из полученных результатов видно, что значения относительной погрешности для моделей МО находятся в пределах 4-5 %, а для аппроксимаций по формулам (2) и (4) – достигают 7%. Аппроксимация по формуле (3) также продемонстрировала наибольшую погрешность. На рисунке 4 приведены результаты аппроксимации в зависимости от коэффициента вариации времени обслуживания при  $\rho=0.7$  и  $n=5$ . Значения аппроксимаций для формул (2-4) не приведены на рисунке 4 по причине высокой относительной погрешности при значениях  $\nu$  отличных от 1.0 – до 80% при  $\nu$ , приближающемся к значению 3.0 и до 60% при  $\nu$  приближающемся к значению 0.1.

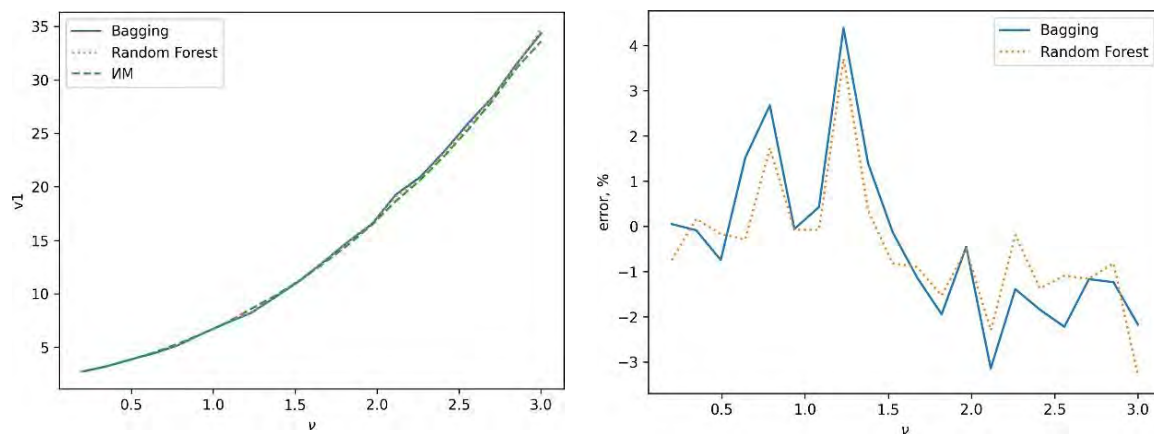


Рисунок 4. Результаты аппроксимаций в зависимости от коэффициента вариации времени обслуживания при  $\rho=0.7$ ,  $n=5$

Из полученных результатов видно, что значения относительной погрешности для моделей МО находятся в пределах 4 %.

На рисунке 5 приведен график ДФР времени пребывания заявок в СМО с помощью аппроксимации Вейбулла по найденным начальным моментам для  $\rho=0.7$ ,  $\nu=1.0$  и  $n=5$ . Также приведен график относительной ошибки полученных вероятностей. Перцентилю 0.999 соответствует значение  $t$  при ДФР, равном 0.001.

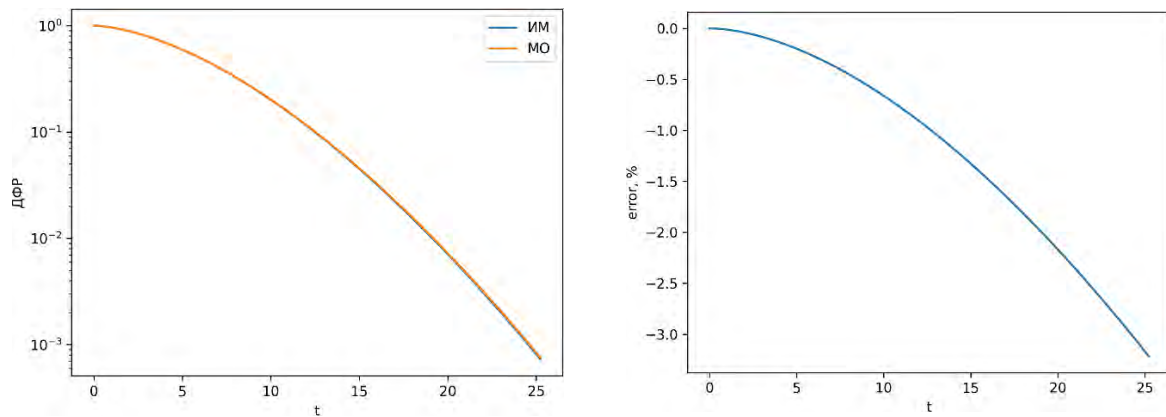


Рисунок 5. ДФР времени пребывания заявок в СМО  
с помощью аппроксимации Вейбулла для  $\rho=0.7$ ,  $v=1.0$  и  $n=5$

Из полученных соотношений видно, что погрешность оценки ДФР увеличивается с ростом  $t$ , но находится в пределах 4%, достаточных для практического применения.

Немаловажным аспектом при проектировании распределенных систем, задействующих очереди Fork-Join, являются временные затраты на получение оценок. Так, среднее время запуска ИМ для заданного набора параметров на ЭВМ с процессором Intel Core i9 составило 21.2 сек, тогда как среднее время аппроксимации с помощью модели МО –  $1.56e-3$  сек. Поскольку при проектировании может потребоваться исследование показателей оперативности при различных комбинациях возможных значений параметров, временные затраты для нахождения наилучших значений могут иметь существенное значение. Например, при числе комбинаций параметров, равном 1000 (например, при комбинациях из десяти возможных значений  $n$ , десяти  $\rho$  и десяти  $v$ ), на ИМ потребуется примерно 6 часов, тогда как аппроксимация с помощью моделей МО займет менее 2 сек.

### Заключение

В данной работе показано, что предложенный подход к получению начальных моментов времени пребывания заявок в СМО Fork-Join на основе регрессионных моделей МО Random Forest и Bagging, по сравнению с известными методами, позволяет получить оценки при произвольном коэффициенте вариации времени обслуживания с относительной погрешностью в большинстве случаев в пределах 4%, с максимальными значениями при экстремальных значениях коэффициента загрузки – до 7%. Данный подход позволяет оценить выполнение требования к задержке в виде процентиля распределения, с помощью аппроксимации ДФР Вейбулла. Показано, что полученные оценки для ДФР не превышают 4% (по крайней мере, для процентов до 0.999, соответствующих значению 0.001 ДФР), что достаточно для большинства практических задач.

В дальнейших работах планируется исследовать возможность построения подобных моделей для других СМО, например, с дисциплиной Fork-Join ( $n, k$ ), в которой, по сравнению с базовой моделью СМО Fork-Join, заявка требует завершения только  $k$  из  $n$  подзадач.

### Литература

1. **Lakshman A., Malik P.** Cassandra: A decentralized structured storage system // Operating Systems Review. – 2010. – Vol. 44(2). – P. 35-40. – DOI:10.1145/1773912.1773922.

2. **DeCandia G., Hastorun D., Jampani M., Kakulapati G., Lakshman A., Pilchin A., Sivasubramanian S., Vosshall P. and Vogels W.** Dynamo: Amazon's highly available key-value store // ACM SIGOPS symposium. – 2007. – P. 205–220. – DOI: 10.1145/1294261.1294281.
3. **Alomari F., Menasce D.A.** Efficient Response Time Approximation for Multiclass Fork and Join Queues in Open and Closed Queueing Networks // IEEE Transaction on Parallel and Distributed Systems. – 2014. – Vol. 25. – P. 1437-1446. – DOI: 10.1109/TPDS.2013.70.
4. **Baccelli F., Makowski A.M., Shwartz A.** The fork-join queue and related systems with synchronization constraints: stochastic ordering and computable bounds // Advanced in Applied Probability. – 1989. – Vol. 21. – P. 629-660. – DOI: 10.2307/1427640
5. **Baccelli F.** Two parallel queues created by arrivals with two demands. The M/G/2 symmetrical case // Technical report INRIA-Rocquencourt. – 1985. – Vol. 426.
6. **Flatto L., Hahn S.** Two parallel queues created by arrivals with two demands // SIAM Journal on Applied Mathematics. – 1979. – Vol. 44. – P. 1041-1053. – DOI: 10.1137/0144074.
7. **Khabarov R.S., Lokhvitckii V.A., Dudkin A.S.** Relationship invariants based sojourn time approximation for the fork-join queueing system // CEUR Workshop Proceedings: Proceedings of Models and Methods of Information Systems Research Workshop 2019. – 2020. – P. 63-68.
8. **Nelson R., Tantawi A.N.** Approximate analysis of fork/join synchronization in parallel queues // IEEE Transactions on Computers. – 1988. – Vol. 37. – P. 739-743. – DOI: 10.1109/12.2213.
9. **Qiu Z., Perez J.G., Harrison P.G.** Beyond the mean in fork-join queues: Efficient approximation for response-time tails // Performance evaluations. – 2015. – Vol. 91. – P. 99-106. – DOI: 10.1016/j.peva.2015.06.007.
10. **Varki E., Merchant A., Chen H.** The M/M/1 Fork-Join Queue with Variable Subtasks. [electronic resource] <http://www.cs.inh.edu/~varki/publication/2002-nov-open.pdf>
11. **Varma S., Makowski A.M.** Interpolation Approximations for Symmetric Fork-Join Queues // Performance Evaluation. – 1994. – Vol. 20. – P. 245-265. – DOI: 10.1016/0166-5316(94)90016-7.
12. **Wright P.E.** Two parallel processors with coupled inputs // Advances in Applied Probability. – 1992. – Vol. 24. – P. 986-1007. – DOI: 10.2307/1427722.
13. **Рыжиков Ю.И., Лохвицкий В.А., Хабаров Р.С.** Метод расчета длительности обработки задач в системе массового обслуживания с учетом процессов Split-Join // Известия высших учебных заведений. Приборостроение. – 2019. – № 62 (5). – С. 419-423.
14. **Хабаров Р.С., Лохвицкий В.А., Дудкин А.С.** Аппроксимация времени пребывания для системы массового обслуживания Fork-Join на основе инвариантов отношения // Интеллектуальные технологии на транспорте. – 2020. – № 2(22). – С. 46-50.
15. **Хабаров Р.С., Лохвицкий В.А.** Модель оценивания оперативности многопоточной обработки задач в распределенной вычислительной среде с учетом процессов Split-Join // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ и управление. – 2019. – № 1. – С. 26-34. – DOI: 10.25586/RNU.V9187.19.01.P.026.
16. **Хабаров Р.С., Лохвицкий В.А., Корчагин П.В.** Расчет временных характеристик системы массового обслуживания с процессами расщепления и слияния заявок и разогревом // Вестник Российского нового университета. Серия: Сложные

системы: модели, анализ и управление. – 2021. – № 2. – С. 10-19. – DOI  
10.25586/RNU.V9187.21.02.P.010.

17. **Breiman L.** Bagging predictors // Mach Learn. – 1996. – Vol. 24. – P. 123–140. – DOI:10.1007/BF00058655.

18. **Cortes C., Vapnik V.** Support-vector networks // Mach Learn. – 1995. – Vol. 20. – P. 273–297. –DOI: 10.1007/BF00994018.

19. **Tin Kam Ho.** Random decision forests // Proceedings of 3rd International Conference on Document Analysis and Recognition. – 1995. – Vol.1. – P. 278-282. – DOI: 10.1109/ICDAR.1995.598994.

20. **Вишневский В.М.** Теоретические основы проектирования компьютерных сетей / В.М. Вишневский. – М.: Техносфера, 2003. – 512 с.