

**Программные системы:**  
**теория и приложения**

УДК 004.932.75'1+004.89

doi 10.25209/2079-3316-2023-14-1-3-30



## Распознавание табличной информации с использованием свёрточных нейронных сетей

Игорь Викторович **Винокуров**

Финансовый Университет при Правительстве Российской Федерации, Москва, Россия

✉ [igvvinokurov@fa.ru](mailto:igvvinokurov@fa.ru)

**Аннотация.** Показана актуальность выявления табличной информации и распознавания её содержимого для обработки отсканированных документов. Описано формирование набора данных для обучения, валидации и тестирования нейронной сети глубокого обучения (DNN) YOLOv5s для обнаружения простых таблиц. Отмечена эффективность использования этой DNN при работе с отсканированными документами. С использованием Keras Functional API сформирована свёрточная нейронная сеть (CNN) для распознавания основных элементов табличной информации – цифр, основных знаков препинания и букв кириллицы. Приведены результаты исследования работы этой CNN. Описана реализация выявления и распознавания табличной информации на отсканированных документах в разработанной ИС актуализации информации в базах данных системы ЕГРН Росреестра.

**Ключевые слова и фразы:** Свёрточные нейронные сети, нейронные сети глубокого обучения, CNN, DNN, YOLOv5s, Keras, Python

Для цитирования: Винокуров И. В. *Распознавание табличной информации с использованием свёрточных нейронных сетей* // Программные системы: теория и приложения. 2023. Т. 14. № 1(56). С. 3–30. [https://psta.psiras.ru/read/psta2023\\_1\\_3-30.pdf](https://psta.psiras.ru/read/psta2023_1_3-30.pdf)

## Введение

Реализуемый в настоящее время процесс цифровой трансформации бизнес-моделей различного вида организаций предполагает в том числе и их переход на электронные носители информации. Следствием этого является необходимость распознавания содержимого электронных документов с целью последующей обработки их содержимого. В случае эффективного распознавания содержимого электронного документа его автоматическая (в ряде случаев автоматизированная) обработка существенно повышает скорость, и в ряде случаев, качество этого процесса. Отсюда решение задачи распознавания информации на отсканированных копиях документов, являющейся первым этапом преобразования бумажного документа в его электронный аналог, представляет определённый практический и научный интерес.

Одним из возможных подходов к решению этой задачи с приемлемой точностью, наряду с традиционными методами, является использование CNN. В [1] описаны общие принципы распознавания отсканированного изображения с использованием CNN, обученной на характерном для содержимого текстовых документов наборе данных. Однако достаточно часто наиболее важная информация в документе представляется в табличном виде. Выявление только таблицы и распознавание её содержимого позволит для подобного вида документов повысить скорость распознавания необходимой для последующей обработки информации и, как следствие, эффективность этого процесса.

**Целью данной работы** является разработка ИС, реализующей локализацию таблицы на отсканированном документе и распознавание её содержимого. В этой работе осуществляется обнаружение в отсканированных документах простых типов таблиц, характерных для большинства документов и содержащих только текстовую или числовую информацию. Ограничение типа таблицы и особенности содержащейся в ней информации даёт основание сформировать свой собственный набор данных для обучения и выбрать любую из известных моделей DNN или CNN, демонстрирующих приемлемые результаты детектирования различного типа объектов. После выявления расположения таблицы в документе, для распознавания её содержимого предлагается использовать небольшую модификацию изложенного в [1] метода.

Полученные в работе результаты последовательного применения DNN для детектирования таблиц и разработанной CNN нашли практическое применение для актуализации базы данных системы ЕГРН Росреестра, переходящего в настоящее время на электронные носители информации.

## 1. Локализация таблицы на изображении

Существующие на данный момент времени методы локализации таблицы на изображении можно разделить на традиционные методы, реализующие обработку изображений и современные методы, в основе которых лежит использование нейронных сетей с архитектурами различных типов.

Традиционные методы локализации таблиц основаны на специфичном структурировании информации и присутствия, в большинстве случаев, горизонтальных и вертикальных линий, их признаков и наличия различных критериев близости и подобия. Так в работах [2] и [3] были предложены методы нахождения табличной информации между ключевыми словами или их комбинациями в начале и конце таблицы и в областях пересечения её горизонтальных и вертикальных линий соответственно. Недостатки подходов очевидны – отсутствие ключевых слов и линий (границ) таблицы приводит к невозможности или крайней неэффективности использования этих методов.

В основе другого подхода [4] лежит метод определения длины строк и столбцов таблицы. После выявления на изображении документа всех горизонтальных и вертикальных линий по их длине, для каждой из них формируется набор из признаков низкого уровня, которые передаются в машину опорных векторов (SVM), реализующей детектирование таблицы. В случае отсутствия линий метод не работает.

Еще один метод [5] предполагает реализовать локализацию и извлечение областей таблицы из изображения документа исходя локальных порогов для межсловного интервала и высоты строки. Основным ограничением этого метода является то, что он обнаруживает области таблицы вместе с окружающими текстовыми областями. Следовательно, его нельзя использовать только для локализации табличной информации.

К традиционным методам можно отнести и метод детектирования в документе табличных структур [6] на основе контура случайно выбранного слова. Метод предполагает, что аналогичные по форме контуры, выявленные по горизонтали и по вертикали и определяют таблицу. Наличие пустых строк в таблице может привести к неэффективности и даже невозможности обнаружения всей табличной информации.

Большую эффективность локализации таблиц на электронном документе реализуют методы, в основе которых лежат CNN или DNN. В таких методах нет явной привязки к структуре таблицы и отсутствует необходимость в предварительной подготовке документа. Однако обучение таких моделей требует большого количества изображений документов, содержащих таблицы – в этом и заключается основная сложность этих методов. Существует несколько наборов данных, которые могут быть использованы для обучения моделей детектирования таблиц – TableBlank

[7], Marmot [8], ICDAR [9], UNLV[10], ICDAR 2019 cTDaR [11] и других. Однако достаточно часто возникает необходимость формирования и собственных наборов данных, что является совсем не тривиальной задачей.

Наиболее эффективными моделями глубокого обучения в настоящее время считаются Fast RCNN и Faster RCNN [12], YOLO [13]. Особенности работы Fast RCNN и Faster RCNN являются нахождение потенциальных объектов на изображении и разбиение их на регионы с помощью метода селективного поиска, извлечение признаков каждого полученного региона с помощью CNN, последующая классификация методом опорных векторов и уточнение границ регионов с помощью линейной регрессии.

Первой для обнаружения таблиц в документе [14] была использована модель Faster RCNN. Однако в настоящее время лидером по детектированию объектов является модель YOLO. Эта модель реализует меньшее количество действий по сравнению с Faster RCNN и реализует поиск ограничивающих объекты рамок и вероятность их принадлежности тому или иному классу изображений.

Существует несколько версий этой DNN; для решения практических задач в настоящее время используются версии v5, v6 и v7 [15], предварительно обученные на наборе данных *COCO (Common Objects in COntext)*<sup>URL</sup>. Время обучения, работы и такие параметры модели как Precision, Recall и AP [16] повышаются от v5 к v7, следовательно, формально версию v7 можно считать более эффективной. Однако на практике эффективность той или иной версии модели определяется особенностями её использования и обучения. Поскольку особых требований к скорости и точности детектирования таблиц нет, а версия v5 [17] показывает вполне приемлемый результат в сочетании с низким потреблением ресурсов и при этом имеет развитые средства сбора информации о процессе обучения и метриках на каждой эпохе, для реализации задачи локализации таблицы на электронном документе была выбрана именно эта версия. Примеры эффективного использования DNN YOLO, обученных на наборе данных ICDAR [11] для детектирования таблиц в электронных копиях документах приведены в [18, 19].

При известной организации этой DNN, единственное, что необходимо сделать – это сформировать наборы данных для её дообучения, валидации и тестирования. Формирование дополнительного обучающего набора данных осуществлялась на нескольких сотнях (порядка 400) отсканированных документов Росреестра с простыми таблицами, содержащими небольшой поясняющий текст в заголовках таблицы и числовую информацию в виде совокупности координат объектов капитального строительства или земельных участков. Для каждого из этих документов с использованием ПО с открытым исходным кодом *labelImg*<sup>URL</sup> были сформированы текстовые файлы с метками таблиц – их нормализованными координатами внутри

документа. Обучение сети осуществлялось на 70% от общего количества изображений, валидация на 20% и тестирование на 10%. Как показали результаты проведённых экспериментальных исследований, 30 эпох обучения позволили получить вполне приемлемый результат по основным параметрам работы DNN и её обобщающих способностей, рисунок 1.

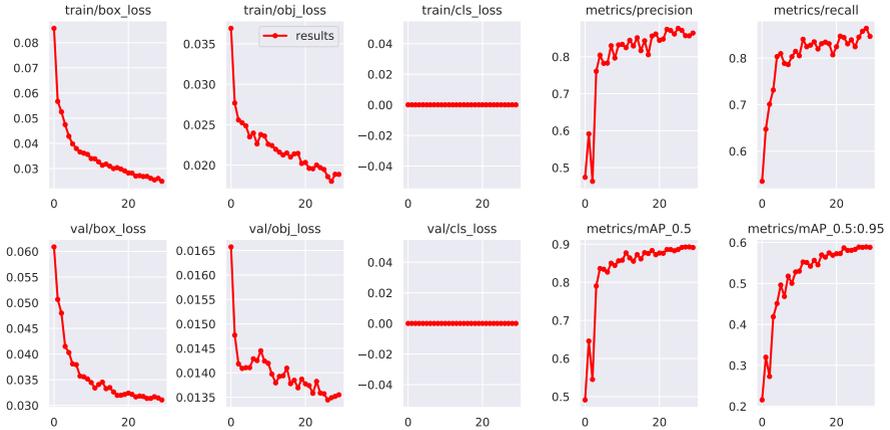


РИСУНОК 1. Графики функций потерь, precision, recall и mAP\_0.5:0.95 при 30 эпохах обучения DNN YOLOv5s

На валидационной выборке при 30 эпохах обучения были получены следующие результаты:  $Precision = 0.871$ ,  $Recall = 0.828$ ,  $mAP_{0.5} = 0.592$ . Зависимость Precision-Recall приведена на рисунке 2.

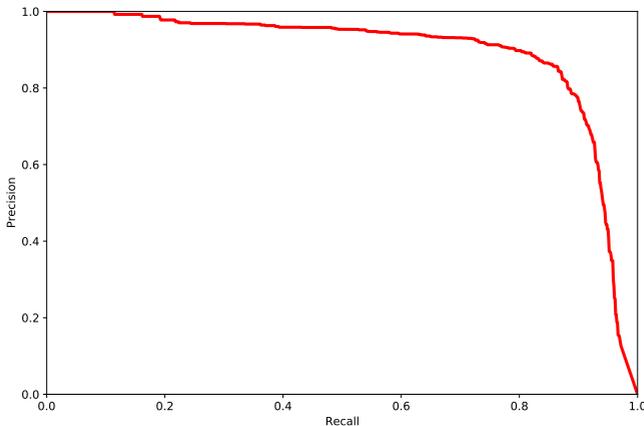


РИСУНОК 2. Precision-Recall при 30 эпохах обучения DNN YOLOv5s

Дообученная на своём наборе данных DNN YOLOv5s продемонстрировала хороший результат обнаружения таблиц на электронных копиях документах хорошего качества – порядка 98-99%. Пример результата работы сети приведен на рисунке 3.

**Table 0.78**

**Описание земельных участков. Раздел "Описание границ"**

Кадастровый квартал XX:XX:XX:XX			Изменение №		
Сведения о вновь образованных и прекращающих существование узловых и поворотных точках границ					
Условное обознач. точки	Координаты		f, м	Описание закрепления точки	Кадастровая запись
	X	Y			
1н	5976781.18	2226562.32	7.5	По естественным рубежам	
2н	5976907.56	2226513.13	7.5	По естественным рубежам	
3н	5976997.66	2226537.35	7.5	По естественным рубежам	
4н	5877213.76	2226611.14	7.5	По естественным рубежам	
5н	5977406.32	2226628.65	7.5	По естественным рубежам	
6н	5977596.63	2226652.87	7.5	По естественным рубежам	
7н	5977823.31	2226759.99	7.5	По естественным рубежам	
8н	5977829.71	2226853.41	7.5	По естественным рубежам	
9н	5977903.99	2226976.25	7.5	По естественным рубежам	
10н	5977956.51	2227056.59	7.5	По естественным рубежам	

Рисунок 3. Результат обнаружения таблицы с помощью DNN YOLOv5s

## 2. Распознавание табличной информации

В [1] приведены общие принципы формирования CNN для реализации многоклассовой классификации элементов текста на отсканированных изображениях плохого качества. Продолжающиеся в настоящее исследование этой CNN и набора данных для её обучения позволили сделать следующий вывод – улучшение результатов классификации может быть достигнуто в большей степени за счёт параллельного выявления устойчивых признаков и их последующего суммирования. Иными словами, усложнение последовательной модели CNN из [1] даёт худшие результаты по сравнению с её функциональной моделью, реализуемой *Keras Functional API*<sup>(URL)</sup>.

Для распознавания табличной информации предлагается структура CNN, имеющая два начальных контура. Первый контур состоит из 2-х свёрточных слоев (*Conv2D*<sup>(URL)</sup>) и одного слоя максимизирующего пуллинга (*MaxPooling2D*<sup>(URL)</sup>). Второй – из одного свёрточного слоя (*Conv2D*) и одного слоя максимизирующего пуллинга (*MaxPooling2D*). Все остальные слои CNN повторяют слои из [1] – один свёрточный слой (*Conv2D*) для выявления признаков из результатов суммирования работы 2-х входных контуров, один линеаризирующий слой (*Flatten*<sup>(URL)</sup>) и два полносвязных слоя (*Dense*<sup>(URL)</sup>). Структура CNN приведена на рисунке 4.

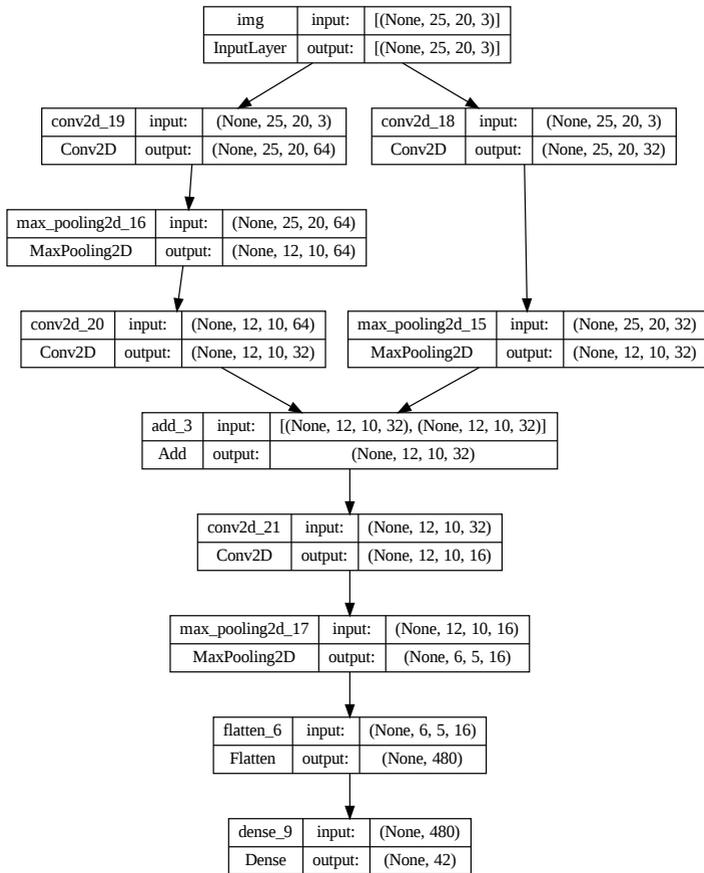


Рисунок 4. Структура CNN для распознавания табличной информации

Все изображения для обучения и валидации приводятся к размеру  $20 \times 25$  [1]. Функции активации всех нейронов CNN – «*sigmoid*»<sup>(URL)</sup>. Параметры компиляции и обучения выбраны стандартными для многоклассовой классификации: оптимизатор – «*adam*»<sup>(URL)</sup>, метрика – «*acc*»<sup>(URL)</sup>, функция потерь – «*categorical-crossentropy*»<sup>(URL)</sup>.

Наборы данных для обучения и валидации этой CNN остался таким же, как и в [1], – 42 класса различных изображений цифр, кириллических букв и 2-х знаков пунктуации «.» и «,». Поскольку эти наборы данных являются небольшими – по 10 и 5 изображений на каждый из символов соответственно, при обучении и последующем исследовании работы

CNN осуществлялась их аугментация с использованием генератора пакетов трансформированных данных *ImageDataGenerator*<sup>URL</sup>. Учитывая особенности предметной области, были выбраны следующие параметры трансформации и их значения:  $zoom\_range = 0.125$ ,  $shear\_range = 0.15$ ,  $rotation\_range = 0.15$ ,  $width\_shift\_range = 1.2$ ,  $height\_shift\_range = 1.2$ ,  $horizontal\_flip = \text{False}$  и  $fill\_mode = \text{«nearest»}$ .

На рисунках 5 и 6 приведены результаты экспериментальных

Результирующая точность модели: 0.975

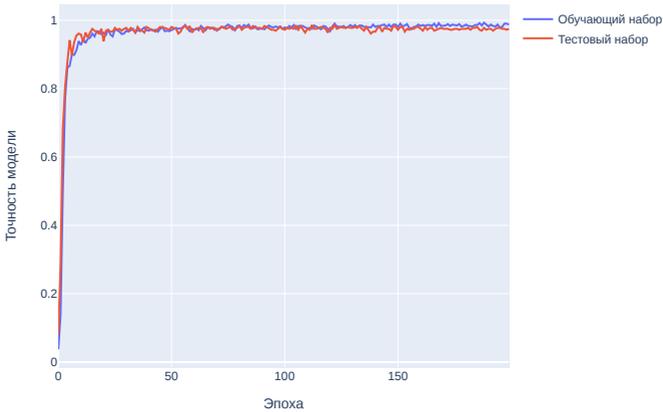


РИСУНОК 5. Точность модели

Результирующие потери модели: 0.056

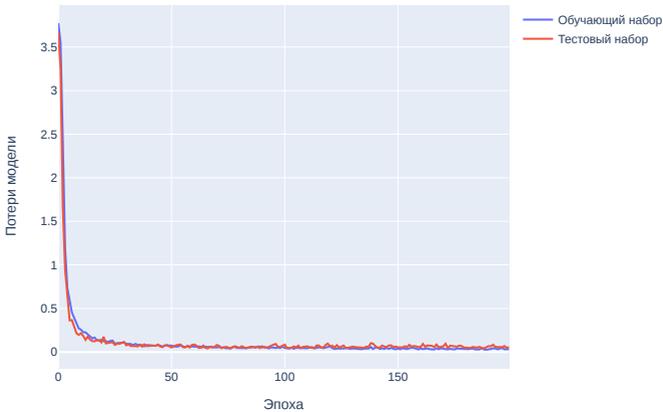


РИСУНОК 6. Потери модели

исследований соответственно точности и потерь модели при 200 эпохах её обучения, являющимся оптимальным количеством по результатам экспериментальных исследований.

Значения параметров gesall и F1 для модели составили 0.98055 и 0.98029 соответственно, что является достаточно хорошим результатом. Качество работы сформированной CNN при классификации элементов текста показывает следующая матрица ошибок CNN рисунок 7. Как

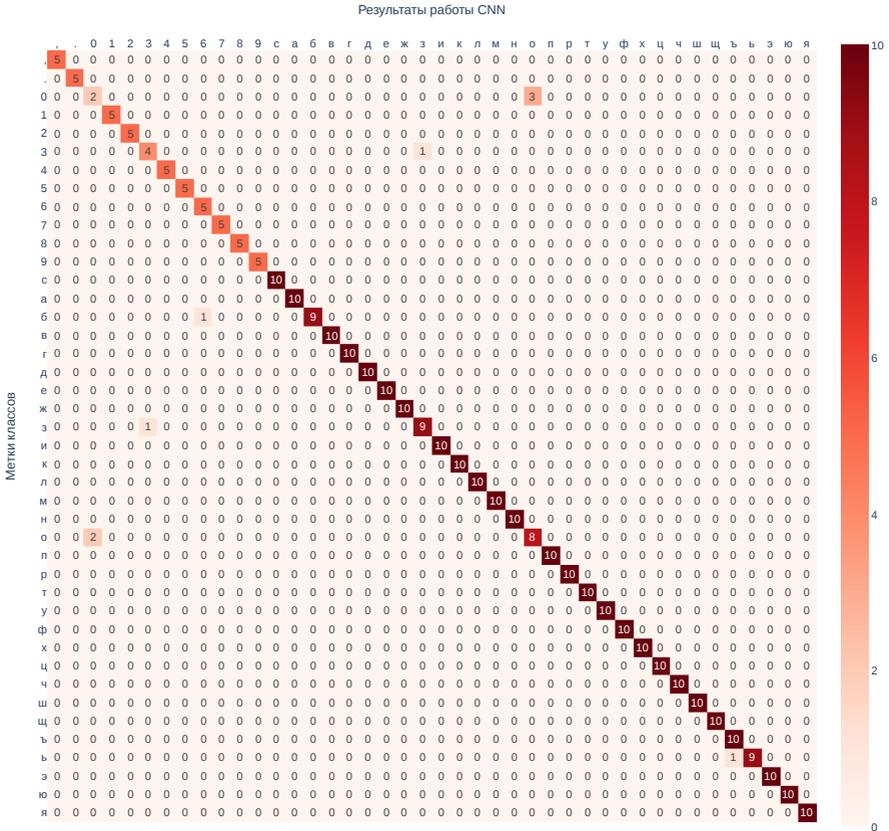


Рисунок 7. Матрица ошибок классификации

видно из этого рисунка, ошибки при классификации элементов текста заключались в схожести цифр 0, 3 и букв «о» и «з».

Пример ошибочной классификации этих цифр и букв приведён на рисунках 8а, и 8б.

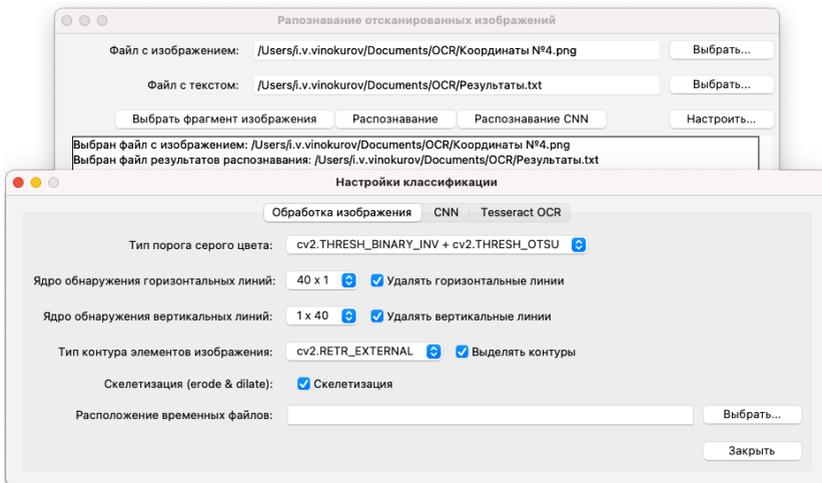


Рисунок 8. Ошибки классификации цифр

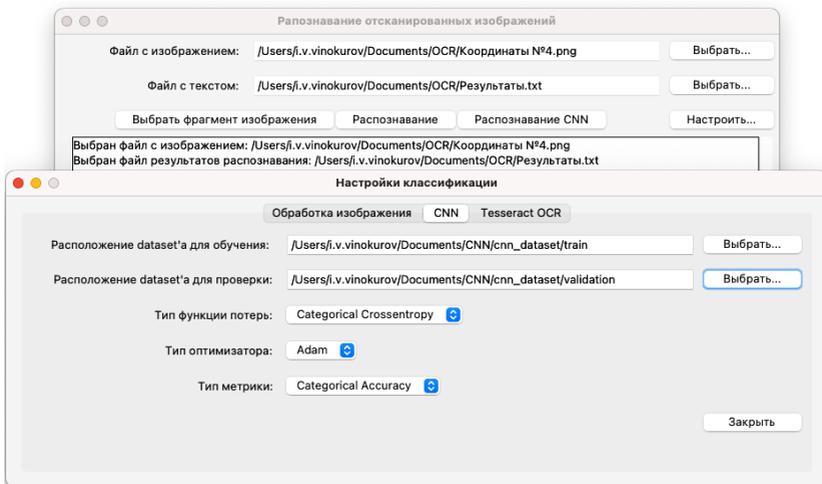
Ошибок распознавания кириллических букв и знаков пунктуации «.» и «,» нет, что является существенным улучшением результатов, полученных в [1].

### 3. Реализация ИС

По результатам проведённых исследований была доработана ИС, общие принципы организации и функционирования которой приведены в [1]. В доработанной ИС, с использованием DNN YOLOv5s, реализовано распознавание расположения таблиц. Координаты таблицы определяют область документа с табличной информацией. Собственно распознавание табличной информации обученной CNN осуществляется по описанным в [1] принципам – преобразование её в градации серого цвета, удаление горизонтальных и вертикальных линий таблицы на основе выбора значений ядер их детектирования и определение расположения границ элементов текста с возможным применением механизма скелетизации последних, рисунок 9а. Параметры обучения CNN остались такими же, как и в [1] – указание расположения наборов для обучения и валидации, выбор типа оптимизатора, функций оценки точности модели и её потерь, рисунок 9б.



(а) Параметры распознавания элементов текста



(б) Параметры обучения CNN

Рисунок 9. Выбор параметров распознавания текста и обучения CNN

В случае невозможности обнаружения таблицы с помощью DNN YOLOv5s на отсканированных документах плохого качества, её расположение в документе может быть выделено вручную (мышью), рисунок 10

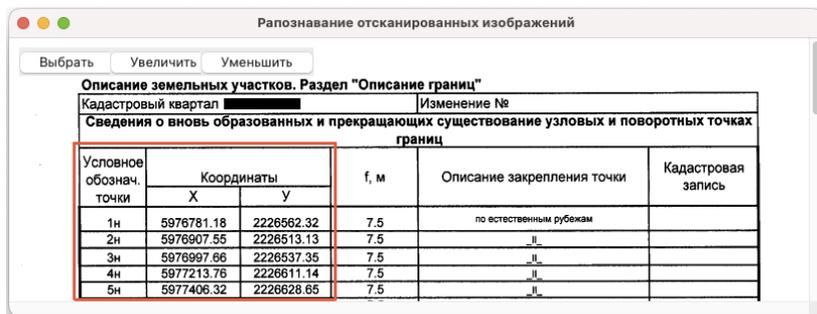


Рисунок 10. Выделение таблицы вручную

Используемое в текущей версии ИС автоматическое распознавание расположение таблицы с использованием DNN YOLOv5s и ограничение распознаванием только табличной информации как наиболее значимой в документах Росреестра позволило повысить скорость её распознавания на 20-30% по сравнению с начальной версией этой ИС.

## Заключение

Проведены экспериментальные исследования DNN YOLOv5s для обнаружения простых таблиц на отсканированных документах Росреестра. Для дообучения YOLOv5s был использован собственный набор данных из предметной области актуализации баз данных системы ЕНРН, состоящий из 400 изображений и соответствующих им текстовых файлов с нормализованными координатами таблиц. Дообученная DNN YOLOv5s показала вполне приемлемые результаты их детектирования на отсканированных документах.

Для распознавания элементов текста внутри найденной области с табличной информацией предложена CNN с двумя контурами выявления признаков. Результаты исследования этой CNN показали лучший по сравнению с [1] результат классификации цифр, кириллических букв и основных знаков пунктуации.

По полученным в результате работы результатам была повышена эффективность работы ИС [1], за счёт распознавания только табличной информации как наиболее значимой информации в документах Росреестра.

## Список литературы

- [1] Винокуров И. В. *Использование свёрточной нейронной сети для распознавания элементов текста на отсканированных изображениях плохого качества* // Программные системы: теория и приложения. – 2022. – Т. 13. – № 3. – С. 29–43.

- [2] Harit G., Bansal A. *Table detection in document images using header and trailer patterns* // *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP'12* (December 16–19, 2012, Mumbai, India), New York: ACM.– 2012.– ISBN 978-1-4503-1660-6.– id. 62.– 8 pp. doi ↑5
- [3] Gatos B., Danatsas D., Pratikakis I., Perantonis S. *Automatic table detection in document images*, ICAPR 2005: Pattern Recognition and Data Mining, Lecture Notes in Computer Science.– vol. **3686**, Berlin–Heidelberg: Springer.– 2005.– ISBN 978-3-540-28757-5.– Pp. 609–618. doi ↑5
- [4] Kasar T., Barlas P., Adam S., Chatelain C., Paquet T. *Learning to detect tables in scanned document images using line information*, 2013 12th International Conference on Document Analysis and Recognition (25–28 August 2013, Washington, DC, USA).– 2013.– Pp. 1185–1189. doi ↑5
- [5] Jahan M. A., Ragel R. G. *Locating tables in scanned documents for reconstructing and republishing*, 7th International Conference on Information and Automation for Sustainability (22-24 December 2014, Colombo, Sri Lanka).– 2014.– Pp. 1–6. doi ↑5
- [6] Kieninger T. G. *Table structure recognition based on robust block segmentation* // *Document Recognition V*, Photonics West'98 Electronic Imaging (1998, San Jose, CA, United States), Proc. SPIE.– vol. **3305**.– 1998.– Pp. 22–32. doi ↑5
- [7] Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, Zhoujun Li *TableBank: A benchmark dataset for table detection and recognition*.– 2020.– 9 pp. doi ↑6
- [8] Fang J., Tao X., Tang Z., Qiu R., Liu Y. *Dataset, ground-truth and performance metrics for table detection evaluation*, 2012 10th IAPR International Workshop on Document Analysis Systems (27–29 March 2012, Gold Coast, QLD, Australia).– 2012.– Pp. 445–449. doi ↑6
- [9] Gobel M., Hassan T., Oro E., Orsi G. *Icdar 2013 table competition*, 2013 12th International Conference on Document Analysis and Recognition (15 October 2013, Washington, DC, USA).– 2013.– Pp. 1449–1453. doi ↑6
- [10] Shahab A., Shafait F., Kieninger T., Dengel A. *An open approach towards the benchmarking of table structure recognition systems* // *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, DAS'10* (June 9–11, 2010, Boston, Massachusetts, USA).– 2010.– ISBN 978-1-60558-773-8.– Pp. 113–120. doi ↑6
- [11] Gao L., Huang Y., Dejean H., Meunier J.-L., Yan Q., Fang Y., Kleber F., Lang E. *ICDAR 2019 competition on table detection and recognition (cTDaR)*, 2019 International Conference on Document Analysis and Recognition (ICDAR) (20–25 September 2019, Sydney, NSW, Australia).– 2019.– Pp. 1510–1515. doi ↑6
- [12] Ren S., He K., Girshick R., Sun J. *Faster R-CNN: towards real-time object detection with region proposal networks* // *IEEE transactions on pattern analysis and machine intelligence*.– 2016.– Vol. **39**.– No. 6.– Pp. 1137–1149. doi ↑6
- [13] Redmon J., Divvala S., Girshick R., Farhadi A. *You only look once: Unified, real-time object detection* // *Proceedings of the IEEE conference on computer vision and pattern recognition*.– 2016.– Pp. 779–788. doi ↑6
- [14] Gilani A., Qasim S. R., Malik I., Shafait F. *Table detection using deep learning* // *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*.– V. 1 (09–15 November 2017, Kyoto, Japan).– 2017.– Pp. 771–776. doi ↑6

- [15] Banerjee A. *YOLOv5 vs YOLOv6 vs YOLOv7*, Learn With A Robot, <https://www.learnwitharobot.com/p/yolov5-vs-yolov6-vs-yolov7>.– 2022–2023. ↑<sup>6</sup>
- [16] Lebedzinski P. *A single number metric for evaluating object detection models*, Towards Data Science, <https://towardsdatascience.com/a-single-number-metric-for-evaluating-object-detection-models-c97f4a98616d>.– 2021. ↑<sup>6</sup>
- [17] Surya Gutta *Object Detection Algorithm — YOLO v5 Architecture*, <https://medium.com/analytics-vidhya/object-detection-algorithm-yolo-v5-architecture-89e0a35472ef>.– Analytics Vidhya.– 2021. ↑<sup>6</sup>
- [18] Zixin Ning, Xinjiao Wu, Jing Yang, Yanqin Yang *MT-YOLOv5: Mobile terminal table detection model based on YOLOv5*, The Fourth International Conference on Physics, Mathematics and Statistics (ICPMS) 2021 (19–21 May 2021, Kunming, China) // Journal of Physics: Conference Series.– 2021.– Vol. **1978**.– id. 012010.  ↑<sup>6</sup>
- [19] Yilun Huang, Qinqin Yan, Yibo Li, Yifan Chen, Zhi Tang *A YOLO-based table detection method*, 2019 International Conference on Document Analysis and Recognition (ICDAR) (20–25 September 2019, Sydney, NSW, Australia).– 2019.  ↑<sup>6</sup>

Поступила в редакцию 23.11.2022;  
 одобрена после рецензирования 28.11.2022;  
 принята к публикации 12.12.2022;  
 опубликована онлайн 13.02.2023.

Рекомендовал к публикации

д.ф.-м.н. А. М. Елизаров

## Информация об авторе:



### Игорь Викторович Винокуров

Кандидат технических наук (PhD), ассоциированный профессор в Финансовом Университете при Правительстве Российской Федерации. Область научных интересов: информационные системы, информационные технологии, технологии обработки данных.

 0000-0001-8697-1032  
 e-mail: [igvvinokurov@fa.ru](mailto:igvvinokurov@fa.ru)

Автор заявляет об отсутствии конфликта интересов.

UDC 004.932.75'1+004.89

 10.25209/2079-3316-2023-14-1-3-30

# Tabular information recognition using convolutional neural networks

Igor Victorovich **Vinokurov**<sup>✉</sup>

Financial University under the Government of the Russian Federation, Moscow, Russia

<sup>✉</sup>[igvvinokurov@fa.ru](mailto:igvvinokurov@fa.ru)

**Abstract.** The relevance of identifying tabular information and recognizing its contents for processing scanned documents is shown. The formation of a data set for training, validation and testing of a deep learning neural network (DNN) YOLOv5s for the detection of simple tables is described. The effectiveness of using this DNN when working with scanned documents is shown. Using the Keras Functional API, a convolutional neural network (CNN) was formed to recognize the main elements of tabular information — numbers, basic punctuation marks and Cyrillic letters. The results of a study of the work of this CNN are given. The implementation of the identification and recognition of tabular information on scanned documents in the developed IS updating information in databases for the Unified State Register of Real Estate system is described.

**Key words and phrases:** Convolutional Neural Networks, Deep Learning Neural Networks, CNN, DNN, YOLOv5s, Keras, Python

2020 *Mathematics Subject Classification:* 68T20; 68T07, 68T45

**For citation:** Igor V. Vinokurov. *Tabular information recognition using convolutional neural networks*. Program Systems: Theory and Applications, 2023, 14:1(56), pp. 3–30. [https://psta.psiras.ru/read/psta2023\\_1\\_3-30.pdf](https://psta.psiras.ru/read/psta2023_1_3-30.pdf)

## Introduction

The process of digital transformation of business models of various types of organizations currently being implemented involves, among other things, their transition to electronic media. This, in turn, leads to the need to recognize the contents of electronic documents for the purpose of subsequent processing of their contents. In the case of effective recognition of the contents of an electronic document, its automatic (in some cases automated) processing significantly increases the speed, and in some cases, the quality of this process. Hence, the solution of the problem of information recognition on scanned copies of documents, which is the first stage of converting a paper document into its electronic counterpart, is of particular practical and scientific interest.

One of the possible approaches to solve this problem with acceptable accuracy, along with traditional methods, is the use of CNN. In [1] describes the general principles of scanned image recognition using a CNN trained on a dataset specific to the content of text documents. Quite often, however, the most important information in a document is presented in tabular form. Revealing only the table and recognizing its contents will allow for this type of documents to increase the speed of recognition of the information necessary for subsequent processing and, as a result, the efficiency of this process.

**The purpose of this work** is to develop an IS that implements the localization of a table on a scanned document and recognition of its contents. In this work, simple types of tables are detected in scanned documents, which are typical for most documents and contain only textual or numeric information. The restriction of the type of table and the features of the information contained in it gives a reason to form your own data set for training and choose any of the known DNN or CNN models that demonstrate acceptable results for detecting various types of objects. After identifying the location of the table in the document, to recognize its contents, it is proposed to use a slight modification of the method described in [1].

The results obtained in the work of the consistent application of DNN for table detection and the developed CNN have found practical application for updating the database of the USRN system of Rosreestr, which is currently being transferred to electronic media.

## 1. Table localization on the image

The currently existing methods for localizing a table on an image can be divided into traditional methods that implement image processing and modern methods based on the use of neural networks with architectures of various types. Traditional methods of table localization are based on the specific structuring of information and the presence, in most cases, of horizontal and vertical lines, their signs and the presence of various proximity and similarity criteria. So in the works [2] and [3] methods for finding tabular information between keywords or their combinations at the beginning and end of the table and in the areas of intersection of its horizontal and vertical lines, respectively, were proposed. The disadvantages of the approaches are obvious — the absence of keywords and lines (borders) of the table leads to the impossibility or extreme inefficiency of using these methods.

Another approach [4] is based on the method of determining the length of rows and columns of a table. After identifying all horizontal and vertical lines along their length on the document image, for each of them a set of low-level features is formed, which are transferred to the support vector machine (SVM) that implements table detection. If there are no lines, the method does not work.

In [5] involves implementing the localization and extraction of table regions from a document image based on local thresholds for word spacing and line height. The main limitation of this method is that it detects table areas along with surrounding text areas. Therefore, it cannot be used only for tabular information localization.

Traditional methods include the method of detecting [6] tabular structures in a document based on the outline of a randomly selected word. The method assumes that contours similar in shape, identified horizontally and vertically, determine the table. Having empty rows in a table can lead to inefficiencies and even failure to discover all of the table information.

Greater efficiency of table localization on an electronic document is implemented by methods based on CNN or DNN. In such methods, there is no explicit binding to the structure of the table and there is no need for preliminary preparation of the document. However, the training of such models requires a large number of images of documents containing tables - this is the main difficulty of these methods. There are several datasets that can be used to train table detection models — TableBlank [7], Marmot

[8], ICDAR [9], UNLV[10], ICDAR 2019 cTDAr [11] and others. However, quite often there is a need to form your own data sets, which is not at all a trivial task.

Fast RCNN and Faster RCNN [12], YOLO [13] are currently considered the most efficient deep learning models. The features of Fast RCNN and Faster RCNN are finding potential objects in the image and dividing them into regions using the selective search method, extracting features of each obtained region using CNN, subsequent classification using the support vector method and refining the boundaries of regions using linear regression.

The Faster RCNN model was used first to detect tables in the [14] document. However, the YOLO model is currently the leader in object detection. This model implements a smaller number of actions compared to Faster RCNN and implements the search for bounding boxes and the probability of their belonging to one or another image class.

There are several versions of this DNN; to solve practical problems, versions v5, v6 and v7 [15] are currently used, pre-trained on the data set *COCO (Common Objects in COntext)*<sup>(URL)</sup>. Training time, work time and model parameters such as Precision, Recall and AP [16] increase from v5 to v7, therefore, formally v7 version can be considered more efficient. However, in practice, the effectiveness of a particular version of the model is determined by the peculiarities of its use and training. Since there are no special requirements for the speed and accuracy of table detection, and version v5 [17] shows quite acceptable results in combination with low resource consumption and, at the same time, has developed tools for collecting information about the learning process and metrics at each epoch to implement the localization problem table on the electronic document, this version was chosen. Examples of efficient use of YOLO DNNs trained on the ICDAR [11] dataset to detect tables in electronic copies of documents are given in [18,19].

With a known organization of this DNN, the only thing that needs to be done is to generate data sets for its additional training, validation and testing. The formation of an additional training data set was carried out on several hundred (about 400) scanned documents of Rosreestr with simple tables containing a small explanatory text in the table headers and numerical information in the form of a set of coordinates of capital construction objects or land plots. For each of these documents, using open source software *labelImg*<sup>(URL)</sup>, text files were generated with table labels— their normalized coordinates within the document. Network training was

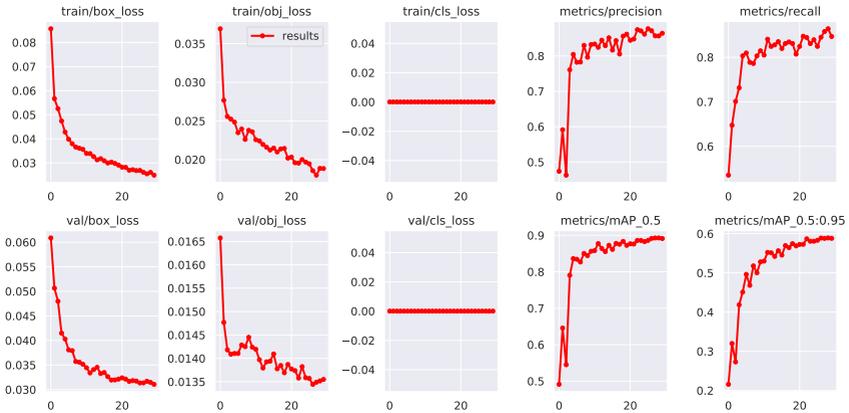


FIGURE 1. Plots of loss functions, precision, recall and  $mAP_{0.5:0.95}$  at 30 training epochs DNN YOLOv5s

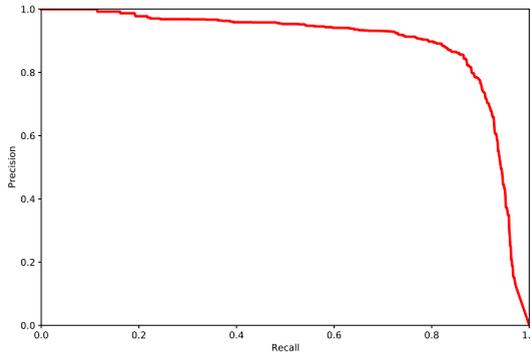


FIGURE 2. Precision-Recall at 30 training epochs DNN YOLOv5s

carried out on 70% of the total number of images, validation on 20% and testing on 10%. As the results of the conducted experimental studies showed, 30 training epochs made it possible to obtain a quite acceptable result in terms of the main parameters of the DNN operation and its generalizing abilities, Figure 1.

On the validation set with 30 training epochs, the following results were obtained:  $Precision = 0.871$ ,  $Recall = 0.828$ ,  $mAP_{0.5} = 0.592$ . The dependence Precision-Recall is shown in Figure 2.

**Table 0.78**

**Описание земельных участков. Раздел "Описание границ"**

Кадастровый квартал XX:XX:XX:XX		Изменение №			
<b>Сведения о вновь образованных и прекращающих существование узловых и поворотных точках границ</b>					
Условное обознач. точки	Координаты		f, м	Описание закрепления точки	Кадастровая запись
	X	Y			
1н	5976781.18	2226562.32	7.5	По естественным рубежам	
2н	5976907.56	2226513.13	7.5	По естественным рубежам	
3н	5976997.66	2226537.35	7.5	По естественным рубежам	
4н	5877213.76	2226611.14	7.5	По естественным рубежам	
5н	5977406.32	2226628.65	7.5	По естественным рубежам	
6н	5977596.63	2226652.87	7.5	По естественным рубежам	
7н	5977823.31	2226759.99	7.5	По естественным рубежам	
8н	5977829.71	2226853.41	7.5	По естественным рубежам	
9н	5977903.99	2226976.25	7.5	По естественным рубежам	
10н	5977956.51	2227056.59	7.5	По естественным рубежам	

FIGURE 3. Table discovery result with DNN YOLOv5s

DNN YOLOv5s, retrained on its own data set, showed a good result of detecting tables on electronic copies of good quality documents—about 98-99%. An example of the result of the network operation is shown in Figure 3.

## 2. Recognition of tabular information

In [1] provides the general principles of CNN formation for the implementation of multiclass classification of text elements on scanned images of poor quality. The ongoing research of this CNN and the dataset for its training led to the following conclusion – the improvement of classification results can be achieved to a greater extent due to the parallel detection of stable features and their subsequent summation. In other words, the complication of the sequential CNN model from [1] gives worse results compared to its functional model implemented by *Keras Functional API*<sup>[URL]</sup>.

To recognize tabular information, a CNN structure is proposed that has two initial contours. The first loop consists of 2 convolution layers (*Conv2D*<sup>[URL]</sup>) and one maximizing pooling layer (*MaxPooling2D*<sup>[URL]</sup>). The second one consists of one convolutional layer (*Conv2D*) and one maximizing pooling layer (*MaxPooling2D*). All other CNN layers repeat layers from [1]—one convolutional layer (*Conv2D*) for feature detection from summation results of 2 input circuits, one linearizing layer (*Flatten*<sup>[URL]</sup>) and

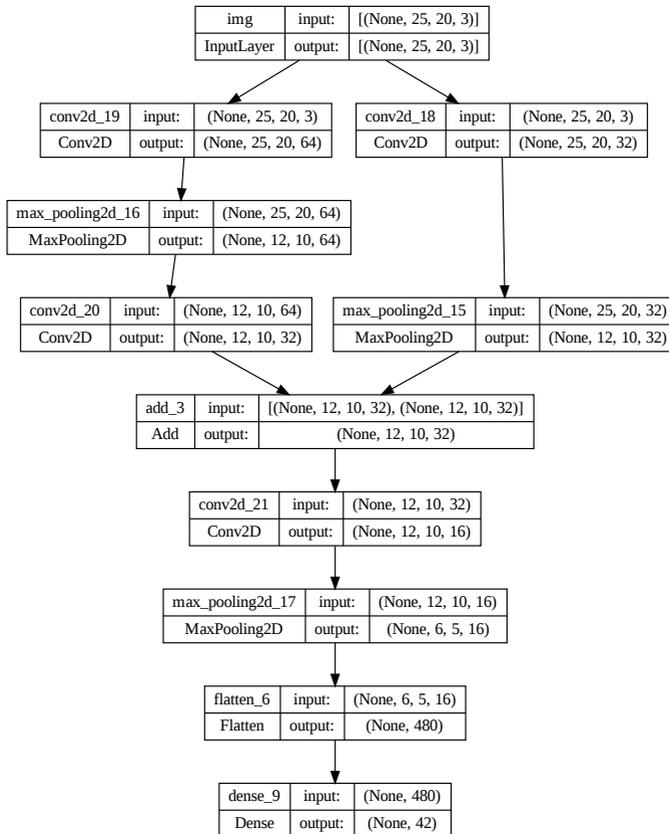


FIGURE 4. CNN structure for tabular information recognition

two fully connected layers ( $Dense^{\text{URL}}$ ). The structure of the CNN is shown in Figure 4.

All training and validation images are resized to  $20 \times 25$  [1]. The activation functions of all CNN neurons are  $\ll\textit{sigmoid}\gg^{\text{URL}}$ . Compilation and training parameters are chosen as standard for multiclass classification: optimizer —  $\ll\textit{adam}\gg^{\text{URL}}$ , metric —  $\ll\textit{acc}\gg^{\text{URL}}$ , the loss function is  $\ll\textit{ategorical-crossentropy}\gg^{\text{URL}}$ .

The data sets for training and validation of this CNN remained the same as in [1], — 42 classes of different images of numbers, Cyrillic letters and 2 punctuation marks «.» and «,». Since these datasets are small — 10 and 5

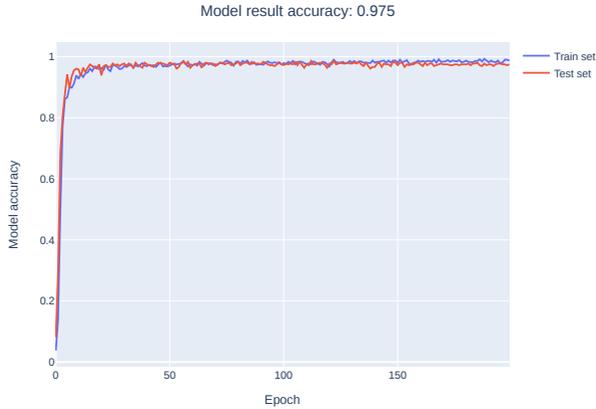


FIGURE 5. Model accuracy

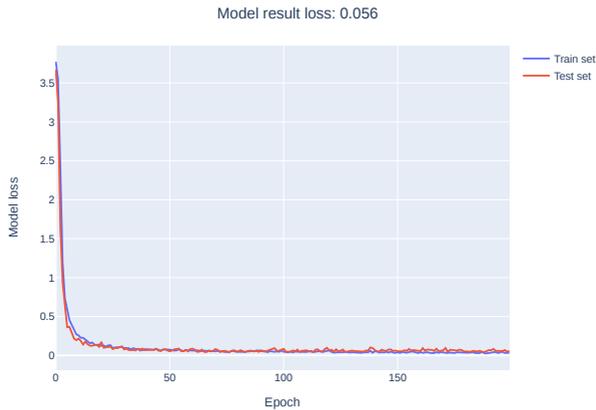


FIGURE 6. Model loss

images for each of the symbols, respectively, during training and subsequent study of the CNN operation, they were augmented using the transformed data batch generator *ImageDataGenerator*<sup>URU</sup>. Taking into account the specifics of the subject area, the following transformation parameters and their values were chosen: *zoom\_range* = 0.125, *shear\_range* = 0.15, *rotation\_range* = 0.15, *width\_shift\_range* = 1.2, *height\_shift\_range* = 1.2, *horizontal\_flip* = False and *fill\_mode* = «nearest».

Figure 5 and Figure 6 show the results of experimental studies, respec-

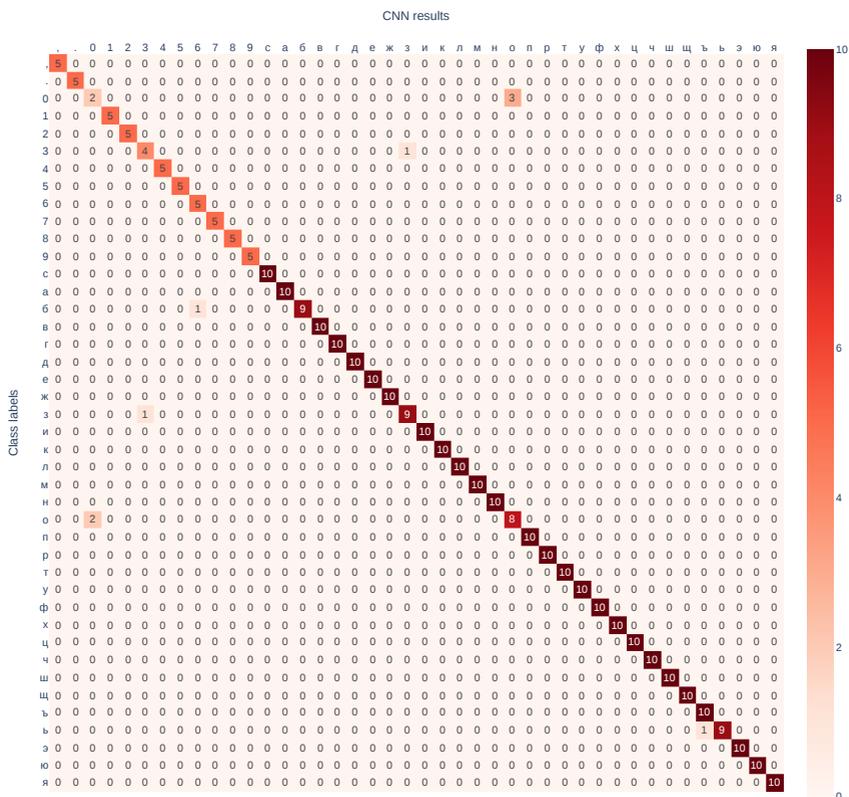


FIGURE 7. Confusion matrix

tively, of the accuracy and loss of the model at 200 epochs of its training, which is the optimal number according to the results of experimental studies.

The values of the recall and F1 parameters for the model were 0.98055 and 0.98029, respectively, which is a very good result. The quality of the generated CNN when classifying text elements is shown by the following CNN confusion matrix in Figure 7.

As can be seen from this figure, errors in the classification of text elements consisted in the similarity of the numbers 0, 3 and the letters «о» and «з». An examples of the erroneous classification of these numbers and letters are shown in Figure 8*a*, Figure 8*b*. No error appears in recognizing

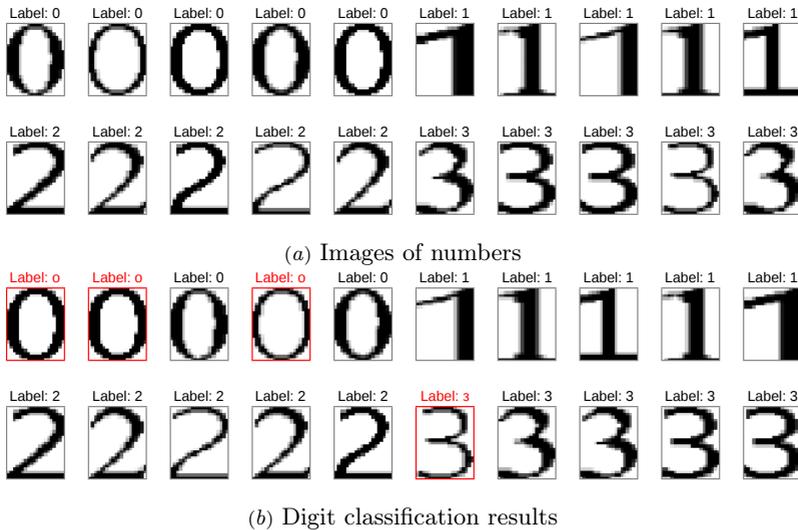


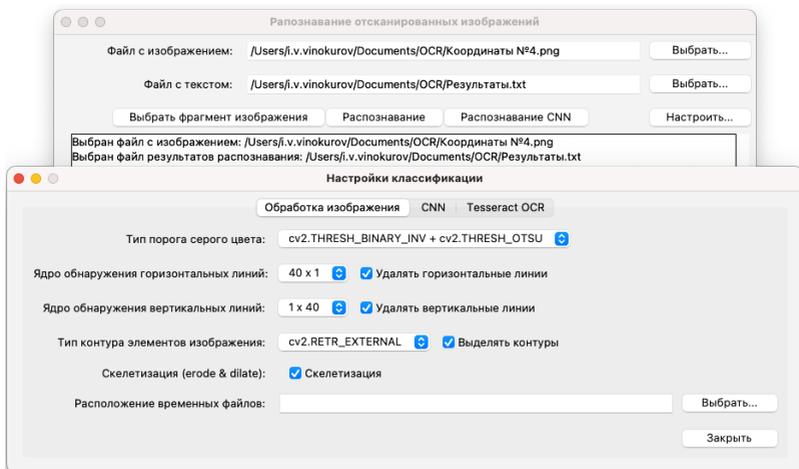
FIGURE 8. Digits classification errors

Cyrillic letters and punctuation marks «.» and «,», that is a significant improvement on the results obtained in [1].

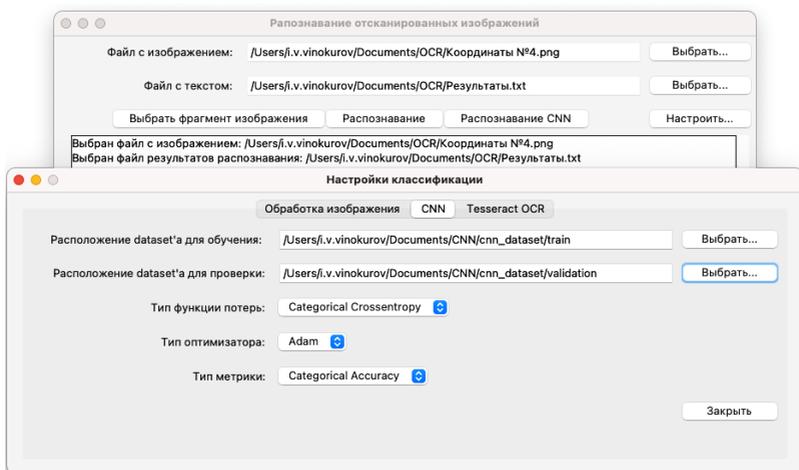
### 3. IS implementation

Based on the results of the research, the IS was finalized, the general principles of organization and functioning of which are given in [1]. In the modified IS, using DNN YOLOv5s, recognition of the location of tables is implemented. Table coordinates define the area of the document with tabular information.

The actual recognition of the tabular information of the trained CNN is carried out according to the principles described in [1]—converting it to grayscale, deleting horizontal and vertical lines of the table based on the choice of the values of their detection kernels and determining the location of the boundaries of text elements with the possible use of the skeletonization mechanism of the latter, Figure 9a. The training parameters of the CNN remained the same as in [1]—specifying the location of sets for training and validation, choosing the type of optimizer, functions for estimating the accuracy of the model and its losses, Figure 9b.



(a) Text recognition options



(b) CNN training parameters

FIGURE 9. Selecting text recognition and CNN training parameters

If it is impossible to detect a table using DNN YOLOv5s on scanned documents of poor quality, its location in the document can be selected manually (with the mouse), Figure 10.

The automatic recognition of the table location used in the current

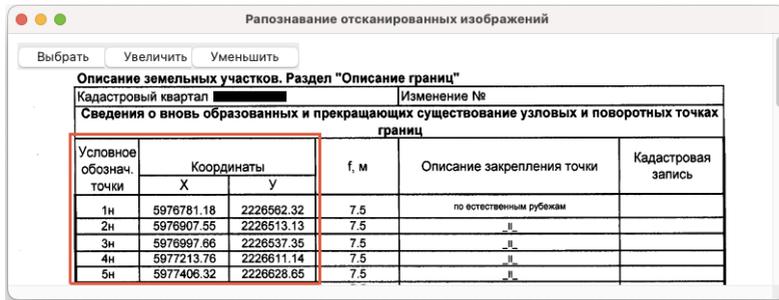


FIGURE 10. Selecting a table manually

version of the IS using DNN YOLOv5s and the limitation of recognizing only tabular information as the most significant in Rosreestr documents made it possible to increase the speed of its recognition by 20-30 compared to the initial version of this IS.

## Conclusion

Experimental studies of DNN YOLOv5s were carried out to detect simple tables on scanned documents of Rosreestr. For additional training of YOLOv5s, we used our own data set from the subject area of updating the databases of the ENPH system, consisting of 400 images and corresponding text files with normalized table coordinates. The retrained DNN YOLOv5s showed quite acceptable results of their detection on scanned documents.

To recognize text elements within the found area with tabular information, a CNN with two features detection contours is proposed. The results of the study of this CNN showed a better result in the classification of numbers, Cyrillic letters and basic punctuation compared to [1].

According to the results obtained as a result of the work, the efficiency of the IS [1] was increased, due to the recognition of only tabular information as the most significant information in the Rosreestr documents.

## References

- [1] I. V. Vinokurov. “Using a convolutional neural network to recognize text elements in poor quality scanned images”, *Programmnyye sistemy: teoriya i prilozheniya*, **13:3** (2022), pp. 29–43 (in Russian).    ↑18, 22, 23, 26, 28
- [2] G. Harit, A. Bansal. “Table detection in document images using header and trailer patterns”, *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP'12* (December 16–19, 2012, Mumbai, India), ACM, New York, 2012, ISBN 978-1-4503-1660-6, id. 62, 8 pp.

- [3] B. Gatos, D. Danatsas, I. Pratikakis, S. Perantonis. “Automatic table detection in document images”, ICAPR 2005: Pattern Recognition and Data Mining, Lecture Notes in Computer Science, vol. **3686**, Springer, Berlin–Heidelberg, 2005, ISBN 978-3-540-28757-5, pp. 609–618. doi ↑19
- [4] T. Kasar, P. Barlas, S. Adam, C. Chatelain, T. Paquet. “Learning to detect tables in scanned document images using line information”, 2013 12th International Conference on Document Analysis and Recognition (25–28 August 2013, Washington, DC, USA), 2013, pp. 1185–1189. doi ↑19
- [5] M. A. Jahan, R. G. Ragel. “Locating tables in scanned documents for reconstructing and republishing”, 7th International Conference on Information and Automation for Sustainability (22-24 December 2014, Colombo, Sri Lanka), 2014, pp. 1–6. doi ↑19
- [6] T. G. Kieninger. “Table structure recognition based on robust block segmentation”, *Document Recognition V*, Photonics West’98 Electronic Imaging (1998, San Jose, CA, United States), Proc. SPIE, vol. **3305**, 1998, pp. 22–32. doi ↑19
- [7] Li Minghao, Cui Lei, Huang Shaohan, Wei Furu, Zhou Ming, Li Zhoujun. *TableBank: A benchmark dataset for table detection and recognition*, 2020, 9 pp. doi ↑19
- [8] J. Fang, X. Tao, Z. Tang, R. Qiu, Y. Liu. “Dataset, ground-truth and performance metrics for table detection evaluation”, 2012 10th IAPR International Workshop on Document Analysis Systems (27–29 March 2012, Gold Coast, QLD, Australia), 2012, pp. 445–449. doi ↑20
- [9] M. Gobel, T. Hassan, E. Oro, G. Orsi. “Icdar 2013 table competition”, 2013 12th International Conference on Document Analysis and Recognition (15 October 2013, Washington, DC, USA), 2013, pp. 1449–1453. doi ↑20
- [10] A. Shahab, F. Shafait, T. Kieninger, A. Dengel. “An open approach towards the benchmarking of table structure recognition systems”, *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS’10 (June 9–11, 2010, Boston, Massachusetts, USA), 2010, ISBN 978-1-60558-773-8, pp. 113–120. doi ↑20
- [11] L. Gao, Y. Huang, H. Dejean, J. -L. Meunier, Q. Yan, Y. Fang, F. Kleber, E. Lang. “ICDAR 2019 competition on table detection and recognition (cTDaR)”, 2019 International Conference on Document Analysis and Recognition (ICDAR) (20–25 September 2019, Sydney, NSW, Australia), 2019, pp. 1510–1515. doi ↑20
- [12] S. Ren, K. He, R. Girshick, J. Sun. “Faster R-CNN: towards real-time object detection with region proposal networks”, *IEEE transactions on pattern analysis and machine intelligence*, **39**:6 (2016), pp. 1137–1149. doi ↑20
- [13] J. Redmon, S. Divvala, R. Girshick, A. Farhadi. “You only look once: Unified, real-time object detection”, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788. doi ↑20

- [14] A. Gilani, S. R. Qasim, I. Malik, F. Shafait. “Table detection using deep learning”, *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. V. 1 (09–15 November 2017, Kyoto, Japan), 2017, pp. 771–776.  [↑20](#)
- [15] A. Banerjee. *YOLOv5 vs YOLOv6 vs YOLOv7*, Learn With A Robot, <https://www.learnwitharobot.com/p/yolov5-vs-yolov6-vs-yolov7>, 2022–2023. [↑20](#)
- [16] P. Lebedzinski. *A single number metric for evaluating object detection models*, Towards Data Science, <https://towardsdatascience.com/a-single-number-metric-for-evaluating-object-detection-models-c97f4a98616d>, 2021. [↑20](#)
- [17] Surya Gutta. *Object Detection Algorithm — YOLO v5 Architecture*, <https://medium.com/analytics-vidhya/object-detection-algorithm-yolo-v5-architecture-89e0a35472ef>, Analytics Vidhya, 2021. [↑20](#)
- [18] Zixin Ning, Xinjiao Wu, Jing Yang, Yanqin Yang. “MT-YOLOv5: Mobile terminal table detection model based on YOLOv5”, The Fourth International Conference on Physics, Mathematics and Statistics (ICPMS) 2021 (19–21 May 2021, Kunming, China), *Journal of Physics: Conference Series*, **1978** (2021), id. 012010.  [↑20](#)
- [19] Yilun Huang, Qinjin Yan, Yibo Li, Yifan Chen, Zhi Tang. “A YOLO-based table detection method”, 2019 International Conference on Document Analysis and Recognition (ICDAR) (20–25 September 2019, Sydney, NSW, Australia), 2019.  [↑20](#)

Received	23.11.2022;
approved after reviewing	28.11.2022;
accepted for publication	12.12.2022;
published online	13.02.2023.

Recommended by

prof. A. M. Elizarov

### Information about the author:



Igor Victorovich Vinokurov

Candidate of Technical Sciences (PhD), Associate Professor at the Financial University under the Government of the Russian Federation. Research interests: information systems, information technologies, data processing technologies.

 0000-0001-8697-1032  
 e-mail: [igvvinokurov@fa.ru](mailto:igvvinokurov@fa.ru)

*The author declare no conflicts of interests.*

УДК 81'322

 10.25209/2079-3316-2023-14-1-31-54

## О декомпозиции метода построения энкодера языковой модели

Игорь Владимирович Трофимов<sup>✉</sup>

Институт программных систем им. А. К. Айламазяна РАН, Вельсково, Россия

<sup>✉</sup>[itrofimov@gmail.com](mailto:itrofimov@gmail.com)

**Аннотация.** Энкодер в составе языковой модели является механизмом преобразования текстовой информации в *эффективное* числовое представление, пригодное для решения широкого круга задач обработки текста при помощи нейросетевых методов. В данной статье предложен способ декомпозиции процесса обучения языкового энкодера. Рассматриваются вопросы целесообразности такой декомпозиции с точки зрения снижения вычислительных затрат, контроля качества на промежуточных стадиях обучения, обеспечения интерпретируемости результатов каждой стадии. Приводятся оценки качества энкодера.

**Ключевые слова и фразы:** обработка естественного языка, нейронные сети, языковая модель, энкодер, контекстно-зависимые представления, разрешение лексической неоднозначности

Для цитирования: Трофимов И. В. *О декомпозиции метода построения энкодера языковой модели* // Программные системы: теория и приложения. 2023. Т. 14. № 1(56). С. 31–54. [https://psta.psiras.ru/read/psta2023\\_1\\_31-54.pdf](https://psta.psiras.ru/read/psta2023_1_31-54.pdf)

## Введение

Современные языковые модели развиваются преимущественно в экстенсивной манере – рост качественных показателей сопровождается ростом размера нейронной сети. Так, уже BERT [1] имел 340 млн. оптимизируемых параметров. Более поздние модели T5 [2] вышли на уровень 11 млрд., GPT-3 [3] – 175 млрд., Switch Transformer [4] – более триллиона.

Эксперименты с обучением и даже использованием таких сетей вычислительно затратны и требуют применения довольно мощных аппаратных средств. Это обстоятельство влечет увеличение стоимости и сроков исследований, проблематичной стала верификация SOTA-подходов (а также сравнение со **state-of-the-art** на нестандартных задачах), исследования фрагментируются, т.к. многогранное исследование становится слишком затратным.

Выполнение исследований с соответствующими моделями меньших размерностей не всегда оправданно, т.к. полученный в таком эксперименте результат будет (с большой долей вероятности) ниже, чем в эксперименте с полноценной моделью, что обесценивает саму идею такого эксперимента. В этой связи видятся актуальными исследования в направлении интенсификации – поиска новых решений в архитектуре и новых малозатратных подходов к обучению языковых моделей.

Гигантский размер языковой модели является проблемой и для ее практического применения, где существенным показателем становится время предсказания, а также имеются ограничения на объем оперативной и долговременной памяти. Исследователями был предложен ряд решений, таких как дистилляция, квантизация и усечение. Общая идея дистилляции сводится к обучению небольшой модели подражать поведению крупноразмерной, квантизация заключается в снижении точности представления числовой информации, а усечение осуществляется за счет исключения из большой модели части весов с низкой практической значимостью.

Замечательный современный обзор этих методов представлен в [5], который можно дополнить разве что исследованиями по квантизации (например, [6]) и еще одним оригинальным подходом к дистилляции [7]. Однако эти подходы не решают проблему *исследования* больших SOTA-моделей и методов на их основе.

С вычислительной точки зрения существенным недостатком в методологии обучения языковых моделей является то, что модели учатся целиком. В данной работе предлагается подход, предполагающий разбиение процесса обучения языковой модели на отдельные этапы, на каждом из которых

выполняется обучение только части весов. Ожидается, что такой метод позволит значительно сократить вычислительные затраты на обучение.

Статья состоит из двух разделов. В первом описан предлагаемый метод построения энкодера – рассматриваются общие архитектурные идеи, описаны методы создания статической модели и модели контекстуализации. Во втором разделе изложены результаты эксперимента по обучению энкодера и его оценке на задаче разрешения лексической неоднозначности.

## 1. Поэтапный метод построения энкодера

### 1.1. Общие замечания об архитектуре

Типовой энкодер на основе механизма трансформеров включает в себя следующие элементы:

- токенизатор, разбивающий строку текста на фрагменты (наибольшее распространение получили статистические подходы, такие как `SentencePiece` [8], `BPE` [9], `WordPiece` [10]);
- матрицу эмбедингов для хранения «базовых» представлений каждого субтокена;
- механизм моделирования позиции субтокена в тексте;
- слои трансформеров для получения контекстуализованных (контекстно-зависимых) представлений субтокенов, кодирующих внутриязыковые и внутритекстовые связи между субтокенами.

Для обучения таких энкодеров распространен `MLM`-подход (от `masked language modeling`) [1], иногда в тандеме с декодером-генератором, обучаемым на различных языковых задачах.

Отталкиваясь от описанной типовой схемы, предложим ряд альтернативных идей, на основе которых может быть построена новая архитектура и подход к обучению (см. таблицу 1).

Архитектура предлагаемого энкодера и результаты его работы схематически изображены на рисунке 1.

В данной работе не рассматриваются верхние уровни, хотя необходимость их очевидна. Сосредоточимся на устройстве и обучении нижних уровней энкодера, результатом работы которых должно стать качественное представление *лексических значений слов* и *морфологической информации* с учетом разрешения неоднозначностей. Верхние уровни должны дополнить это представление позиционной информацией и закодировать внутритекстовые связи между токенами (синтактико-семантические, кореферентные и др.).

Таблица 1. Ключевые особенности предлагаемого подхода к построению энкодера языковой модели

Настоящее положение дел	Предлагаемая альтернатива
<p>В современных подходах проблема неполноты словаря (oov-проблема) решается за счет субтокенизации вплоть до отдельных литер с дальнейшим восстановлением «значения» слова путем комбинирования «значений» субтокенов и контекстуализации. Таким образом, ситуация <b>out-of-vocabulary</b> возникает крайне редко.</p>	<p>Предлагается крупногранулярная токенизация, чтобы лексическое «значение» слова содержалось в токене в готовом виде и не требовало восстановления из фрагментов. Восстановление «значения» в нередких теперь oov-случаях предлагается осуществлять за счет контекстуализации. Так как у модели нет необходимости обучаться комбинированию субтокенов, слои контекстуализации могут быть компактнее.</p>
<p>Так как «значения» субтокенов тесно связаны с механизмом их комбинирования, затруднительно разделить процессы их обучения. Матрица эмбедингов и слои трансформеров в языковых моделях учатся совместно.</p>	<p>Предлагается независимо обучать подмодель представления токенов быстрыми методами, такими как <b>word2vec [11]</b>. Затем обучать модель контекстуализации с «замороженной» подмоделью для токенов.</p>
<p>Обучение в соответствии с MLM-подходом опирается на предсказание замаскированных словарных единиц (субтокенов). Это ставит в прямую зависимость вычислительную эффективность и размер словаря. В языковых моделях обычно словарь делают умеренных размеров (например, в оригинальной BERT словарь включал 30 тыс. единиц).</p>	<p>При обучении контекстуализации предлагается в замаскированных позициях предсказывать не словарную единицу, а ее векторное представление (т. к. оно «заморожено»). То есть решать задачу регрессии. Размер словаря в этом случае почти не влияет на вычислительную эффективность. Это позволяет языковой модели иметь в готовом виде достаточно большой «словарный запас».</p>
<p>Векторные представления (токенов или субтокенов) не имеют какой-либо внутренней структуры и интерпретации, хотя и замечено, что механизм «внимания» в отдельных <b>head</b> трансформеров иногда отражает ту или иную теоретико-лингвистическую идею (синтаксическую связь, общность лексических значений и т.п.). Заранее неизвестно, какой <b>head</b> какой идее (а чаще смеси идей) научится. Это затрудняет осмысление выполняемых сетью операций.</p>	<p>Предлагается в явном виде моделировать разные виды лексико-семантической общности и закрепить их позиции в векторном представлении. А именно, предлагается моделировать близость значений в категориальном, ассоциативном и грамматическом аспектах.</p>

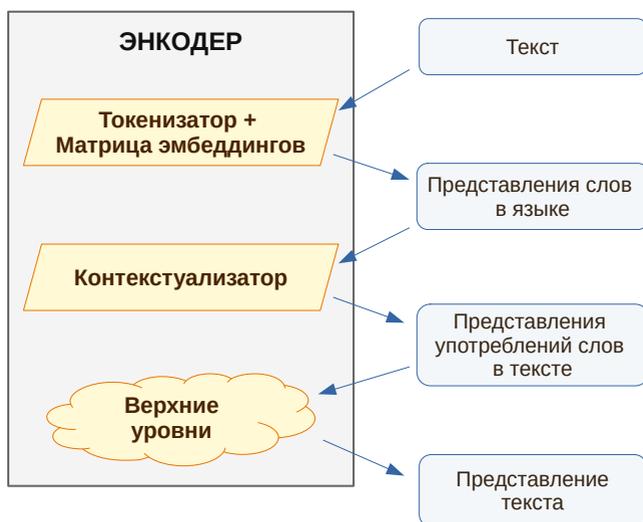


Рисунок 1. Общая схема энкодера

Первым шагом в создании и обучении такого энкодера будет построение статических<sup>1</sup> представлений слов в языке (матрицы эмбедингов).

## 1.2. Представление слов в языке

Основанные на идеях дистрибутивной семантики методы получения статических представлений слов зарекомендовали себя как довольно эффективное средство при решении задач классификации текстов, информационного поиска, машинного перевода и др. Появление быстрых алгоритмов построения статических эмбедингов (**skip-gram**), а также появление множества доступных готовых моделей позволило статическим представлениям стать ключевым компонентом в большом числе приложений, осуществляющих содержательную обработку текстовой информации (хоть и на непродолжительный период).

Очевидным недостатком статических подходов было отсутствие возможности эффективно моделировать «значение» многозначных слов.

<sup>1</sup> Будем пользоваться терминами *статическое* и *динамическое* представление для отсылки, соответственно, к контекстно-независимым представлениям (или представлениям слов в языке) и контекстно-зависимым представлениям (или представлениям употреблений слов в тексте). В литературе такое именование не получило широкого распространения, но, будучи компактным, оно более удобно для изложения. В англоязычной литературе для упоминания контекстно-зависимых представлений принято пользоваться термином *contextualized representation (embedding)*.

Со временем для устранения этой проблемы была предложена идея динамических моделей – моделей, которые позволяли бы получать контекстно-зависимые представления «значений» для отдельных употреблений слов в тексте.

Общий принцип моделирования «значения» слова в статических моделях следующий. Всякое слово представляет собой точку в многомерном пространстве. Чем больше общность значений пары слов, тем ближе они должны быть размещены в упомянутом многомерном пространстве в соответствии с выбранной мерой близости (обычно угловой). Замечено, что основания для определения общности значений могут быть различны. В данной работе предлагается оперировать тремя видами общности на основе выделения категориального, ассоциативного и грамматического компонентов.

Подробный анализ и обзор по теме категориальной и ассоциативной близости значений содержится в [12]. Кратко перескажем основные положения (в терминах стоящих за словами понятий):

- Категориальное сходство отражает общность неотъемлемых (физических или функциональных) свойств сущностей. Например, значительную степень категориального сходства можно усмотреть в парах *автомобиль – велосипед, лошадь – корова*.
- Ассоциация отражает связь сущностей через отношение, ситуацию, тему. Ассоциативно связанные сущности объединяет принадлежность одной предметной области, одной типовой жизненной ситуации и т. п. (*автомобиль – водитель, автомобиль – бензин, лошадь – седло, лошадь – аллюр*).
  - К ассоциациям принято относить и «языковые» связи, обусловленные, например, идиоматичностью (*страница – истории*) или терминологичностью (*коробка – передач*).

Грамматическая общность основана на сходстве морфологических признаков слов. В отличие от категориальной и ассоциативной общности эта составляющая векторного представления не лежит в плоскости лексической семантики. Ее задача кодировать информацию о синтаксических свойствах слов.

Дистрибутивный подход позволяет строить качественные модели для категориальной и ассоциативной компоненты представления. Известно, что для моделирования категориальной близости синтаксический контекст предпочтительней линейно-оконного [13]. Ассоциации лучше обнаруживаются широким линейным окном. Конкатенацию категориальной и ассоциативной компоненты в дальнейшем будем называть лексико-семантическим представлением.

В данном исследовании для получения статических представлений слов использовался инструментарий `conll2vec`<sup>2</sup>, способный порождать модель, включающую все три вышеперечисленные компоненты. В целом `conll2vec` можно охарактеризовать как функционально-насыщенную версию `word2vec`. Тезисно сформулируем ключевые особенности.

- Для построения векторной модели `conll2vec` опирается на размеченные текстовые данные (разметка включает морфологическую и синтаксическую информацию, а также результат лемматизации текста).
- Векторное представление состоит из трех частей. Категориальная компонента строится алгоритмом `skip-gram` на основе синтаксических контекстов; ассоциативная – модификацией `skip-gram` с одной матрицей (контекстом считаются все слова предложения); грамматическая – классификатором на основе морфологической разметки (подобно `skip-gram`, но вместо матрицы контекстов используется матрица представлений для отдельных граммем – логистическая регрессия по каждой граммеме).
- Векторная модель строится в два этапа: сначала представления лемм, затем представления словоформ.
- Имеется возможность использовать дополнительные ресурсы с информацией о близости значений, в том числе, векторные модели большей размерности.

Перечисленный функционал позволяет построить словарь  $V$  и соответствующую матрицу эмбедингов  $E$ . Кроме того, инструментарий `conll2vec` способен:

- строить грамматическую компоненту представления для распространенных суффиксов слов (суффикс понимается как цепочка букв, на которую заканчивается слово: *-щемся*, *-емому*, *-лась* и т.п.);
- частично группировать словоформы, относящиеся к одной «морфологической парадигме» (в кавычках, т.к. имеет место аналогия, а не строгое соответствие), – применяется эвристический алгоритм.

Этот дополнительный функционал позволяет реализовать идею ограниченной субтокенизации на основе следующих соображений.

Словарь векторной модели  $V$  разделяется на три части:

- словарь псевдооснов (для форм, которые удалось собрать в «морфологические парадигмы»);
- словарь суффиксов;

---

<sup>2</sup><https://github.com/parus-proj/conll2vec>

- словарь словоформ (для форм, не вошедших в словарь псевдооснов; сюда попадают омоформы и формы не отвечающие критериям эвристического алгоритма).

Матрица эмбедингов  $E$  преобразуется в две матрицы:

- матрицу лексико-семантических представлений;
- матрицу грамматических представлений.

Их соотношение со словарями отражено на рисунке 2. Эмбединги, относящиеся к словарю словоформ, образуются путем разделения исходного эмбединга на лексико-семантическую и грамматическую части. За эмбединг псевдоосновы принимается лексико-семантическое представление *Л.С.М.М.Ы.*

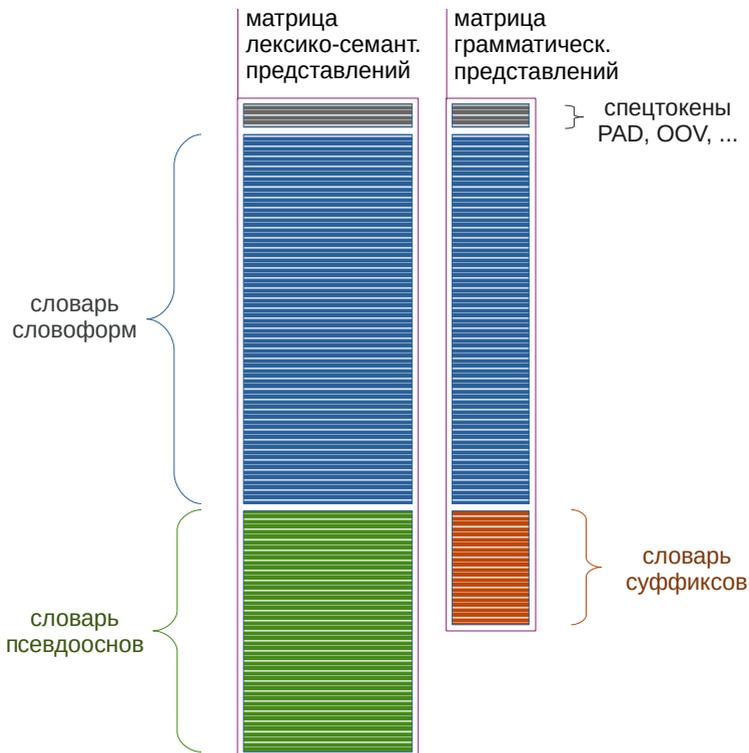


Рисунок 2. Матрицы эмбедингов в предлагаемой модели энкодера

Токенизация реализуется следующим образом. Сначала текст делится на слова по пробельным символам. Затем для каждого слова токенизатор возвращает пару индексов  $\langle i_{LEX}, i_{GR} \rangle$  (по одному для каждой матрицы

представлений) в соответствии со следующим алгоритмом.

- Если слово содержится в словаре словоформ, то индексы будут одинаковы и соответствовать лексико-семантической и грамматической частям представления для данного слова.
- Иначе ищем способы разделить слово на две части так, чтобы левой соответствовала словарная псевдооснова, а правой – словарный суффикс (или нулевой). Если это удастся, то  $i_{LEX}$  будет индексом наиболее длинной псевдоосновы (среди обнаруженных вариантов), а  $i_{GR}$  будет найден как наиболее длинный словарный суффикс, присутствующий в данной словоформе. Например, слово *абонентская* может быть разделено на псевдооснову *абонентск-* и суффикс *-ая*, но если есть представление для более длинного суффикса, например, *-ская*, то предпочтение будет отдано этому (более информативному) представлению.
- Если не удастся найти псевдооснову, то  $i_{LEX}$  берется для специального представления *out-of-vocabulary* (нулевой вектор), а  $i_{GR}$  соответствует наиболее длинному из найденных словарных суффиксов. Если и суффикс не найден, то для грамматической части также берется *oov*-представление.

В сравнении с непосредственным использованием единственной матрицы эмбедингов для словоформ предложенная субтокенизация имеет ряд преимуществ.

- Меньший объем памяти, необходимой для хранения матриц.
- Словарь суффиксов позволяет получить информативное грамматическое представление для большого числа несловарных слов.
- Словарь псевдооснов позволяет получить лексико-семантические представления для несловарных слов, относящихся к данной «морфологической парадигме».

Теперь перейдем к устройству контекстуализации.

### 1.3. Модель контекстуализации и ее обучение

Предлагаемый подход к контекстуализации опирается на следующие предположения:

- для разрешения лексической неоднозначности достаточно учесть тематическую информацию и ближайший контекст многозначного слова;
- для разрешения морфологической неоднозначности достаточно ближайшего контекста.

В основе модели контекстуализации лежит гибридная архитектура, сочетающая рекуррентные сети (для анализа ближайшего контекста) и

механизм внимания (для представления тематической информации). Компоненты представления (категор., ассоц. и грамм.) обрабатываются независимо, поэтому способы их контекстуализации могут быть различными. Рассмотрим общий подход к контекстуализации на примере категориальной части представления (после чего опишем особенности обработки других частей).

При обучении модели применяется MLM-подход. Для каждой текстовой позиции (за исключением *out-of-vocabulary*) модель учится предсказывать целевую компоненту исходного статического векторного представления за счет контекста. Будем иллюстрировать работу модели на примере следующего предложения:

(пр.-1) *Дети поели и пошли гулять.*

Пусть маскируемым (предсказываемым) словом будет союз 'и'<sup>3</sup>. Тогда прямой проход обучения и вычисление ошибки схематически можно изобразить так – рисунок 3.

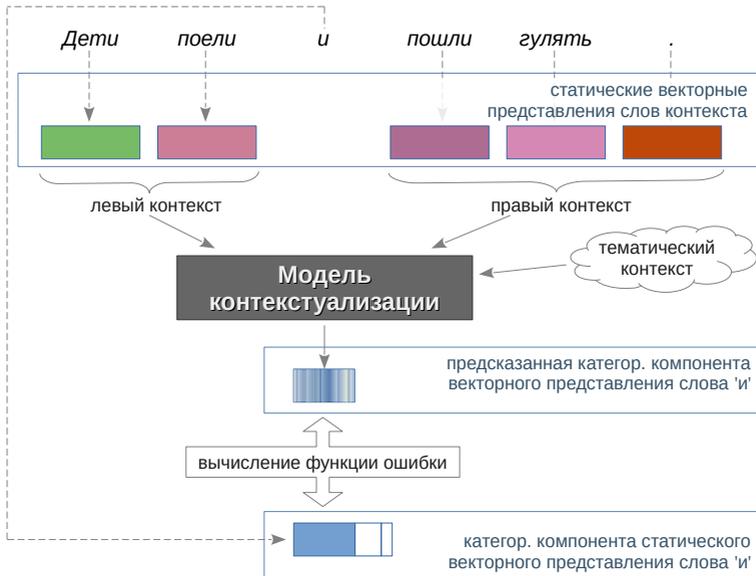


Рисунок 3. Обучение контекстуализации (предсказание векторного представления замаскированного слова)

<sup>3</sup>Заметим, что словоформа 'и' является категориально многозначной. Кроме употребления в роли союза 'и' может входить в состав сокращений (*и.о.*), быть инициалом (*В. И. Ленин*) или даже собственным именем (*Ван И* – министр иностранных дел КНР).

На вход контекстуализатора подаются левый и правый контексты замаскированного слова, а также представление тематического контекста (подробнее ниже). На выходе контекстуализатор стремится построить целевую компоненту векторного представления замаскированного слова. Величина ошибки вычисляется по формуле:

$$(1) \quad loss = CosLoss + \gamma \cdot MseLoss,$$

где  $CosLoss^4$  – угловая ошибка,  $MseLoss$  – среднеквадратичная ошибка,  $\gamma$  – коэффициент, подбираемый так, чтобы вклад среднеквадратичной ошибки в  $loss$  был небольшим в сравнении с более важной угловой составляющей. Первое слагаемое отвечает за корректное угловое размещение предсказываемого вектора в пространстве статической модели, второе – побуждает не пренебрегать длиной вектора.

Внутреннее устройство части контекстуализатора, отвечающей за анализ ближнего контекста, показано на рисунке 4.

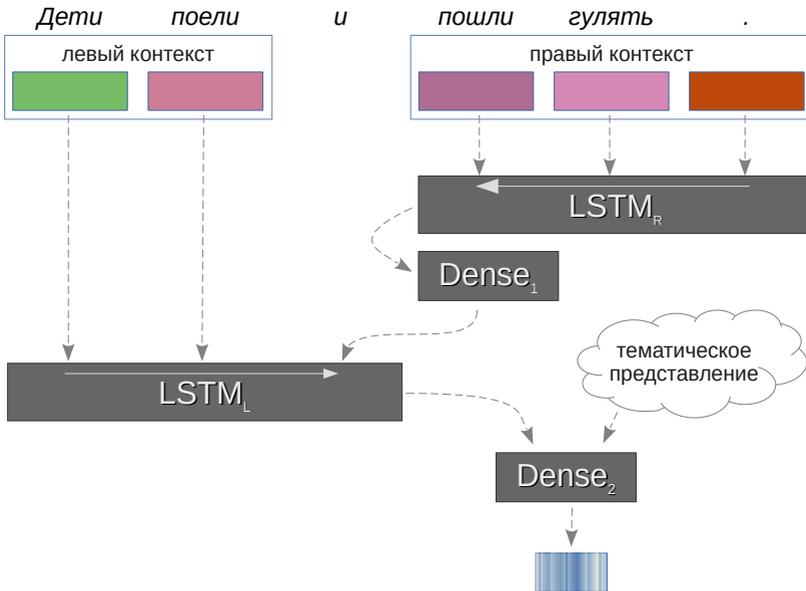


Рисунок 4. Схема обработки левого и правого контекстов в контекстуализаторе

<sup>4</sup> $CosLoss = -\cos(\vec{v}_1, \vec{v}_2)$ , где  $\vec{v}_1, \vec{v}_2$  – сравниваемые векторные представления.

Обработка<sup>5</sup> начинается с анализа правого контекста однослойной однонаправленной LSTM-сетью по направлению к замаскированному слову. На выходе LSTM<sub>R</sub> порождается *представление правого контекста*, которое затем масштабируется при помощи однослойной FF-сети (Dense<sub>1</sub>) до размерности входных векторов. После этого выполняется анализ левого контекста, за последний элемент которого принимается *представление правого контекста*. Для анализа вновь используется LSTM (LSTM<sub>L</sub>). Полученное в результате представление для всего локального контекста (левого и правого) путем конкатенации объединяется с тематическим представлением (о котором чуть ниже) и подается на вход еще одного FF-слоя (Dense<sub>2</sub>). На его выходе формируется представление замаскированного слова, выведенное на основе контекста.

Присоединение *представления правого контекста* ко входу LSTM<sub>L</sub> позволяет сократить пространство гипотез при предсказании. Так, при рассмотрении левого контекста независимо от правого LSTM<sub>L</sub> должна допускать очень разнородные продолжения текста:

- *Дети поели супа* ...
- *Дети поели горячего (супа)* ...
- *Дети поели напех* (приготовленного супа) ...
- *Дети поели за* (кухонным столом) ...
- и т.д.

Информация из правого контекста может помочь LSTM<sub>L</sub> заблокировать часть подобных гипотез<sup>6</sup>.

Тематическое представление является общим для всех слов одного предложения и формируется при помощи *attention*-механизма на базе ассоциативных компонент статических представлений. Коэффициенты внимания вычисляются не для токена целиком, а для каждого измерения в ассоциативной компоненте токена. За тематический вектор предложения принимается среднее ассоциативных компонент всех слов предложения,

---

<sup>5</sup>Для краткости здесь и далее опускаем вопросы регуляризации и соответствующие компоненты и гиперпараметры сети. Интересующиеся деталями реализации могут изучить исходные коды модели и скриптов ее обучения по адресу <https://github.com/parus-proj/rue>.

<sup>6</sup>Возможна реализация подобного информирования LSTM<sub>R</sub> о левом контексте.

взвешенных согласно коэффициентам внимания:

$$(2) \quad t_s = \frac{1}{N} \sum_{i=1}^N \vec{a}_i \odot \vec{v}_i,$$

где  $t_s$  – тематическое представление для предложения  $s$ ,  $\vec{a}_i$  – вектор коэффициентов внимания для  $i$ -ого слова,  $\vec{v}_i$  – вектор ассоциативной компоненты статического представления  $i$ -ого слова,  $N$  – количество слов в предложении.

Архитектура сети, вычисляющей коэффициенты внимания  $\vec{a}_i$ , изображена на рисунке 5 и представляет собой каскад из трех FF-слоев.

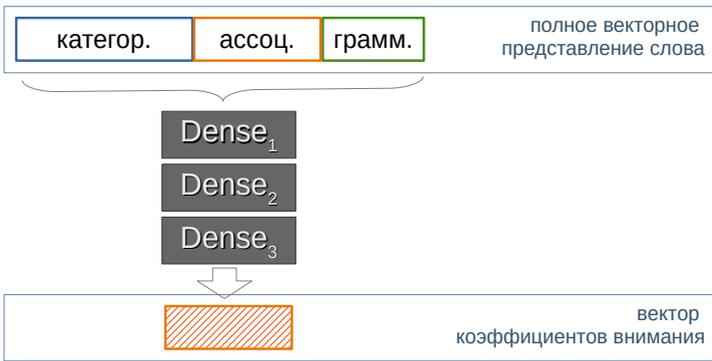


Рисунок 5. Вычисление коэффициентов внимания

Размерности ассоциативной компоненты на входе и вектора коэффициентов внимания на выходе одинаковы. Обучение этой сети выполняется только совместно с контекстуализатором ассоциативных частей<sup>7</sup>.

Таким образом, для каждого слова в предложении вычисляется не просто его значимость для определения темы<sup>8</sup>, а выполняется декомпозиция слова на признаки – каждый признак (измерение) в ассоциативной компоненте может вносить разный вклад в тематический вектор.

Выше была описана контекстуализация категориальной компоненты представления. Возвращаясь к вопросу специфики контекстуализации остальных частей, отметим следующее.

<sup>7</sup>При обучении контекстуализаторов для категориальной и грамматической компонент обратное распространение ошибки по этой ветви сети блокируется.

<sup>8</sup>Такой подход тоже приемлем. Очевидно, что, например, служебные слова практически не вносят вклада в формирование темы предложения. Поэтому даже простой индикатор, вроде IDF, мог бы здесь работать.

- Тематический вектор обучается только в рамках контекстуализатора для ассоциативной компоненты.
- В грамматическом контекстуализаторе вместо тематического вектора используется категориальная компонента предсказываемого слова с наложенной на него маской (dropout вектора с очень высоким коэффициентом зануления). Тема предложения практически бесполезна для предсказания морфологических признаков в какой-то конкретной текстовой позиции, а наличие категориальной информации делает контекстуализатор более точным.

После того как сформированы представления всех компонент слова, выполняется их конкатенация в том же порядке, в каком они размещены в статическом представлении.

Очевидно, что получить качественное (в значительной степени специфичное) представление только на основе контекста не всегда возможно. В этом нетрудно убедиться, рассмотрев следующее предложение с замаскированным словом  $X$ :

(пр.-2) *В цеху установили  $X$  станок.*

Здесь контекст допускает довольно разнообразные по значению подстановки: *деревобрабатывающий, новый, дорогостоящий, двухосный, малощумящий, безоператорный, 200-тонный, неоткалиброванный* и т.д. Выведенным на основе такого контекста представлением, вероятно, окажется малоинформативная смесь всех этих значений.

Вообще о самой идее контекстуализации можно заметить следующее:

- применительно к многозначному<sup>9</sup> слову контекстуализация уточняет представление;
- применительно к однозначному слову – размывает «значение».

Хороший контекстуализатор должен стремиться минимизировать размытие.

В предлагаемой здесь модели, чтобы ослабить эффект размытия, выполняется смешивание исходного статического представления и представления, выведенного сугубо из контекста. Для этого используется формула экспоненциального скользящего среднего:

$$(3) \quad R_{dynamic} = \alpha \cdot R_{ctx} + (1 - \alpha)R_{static},$$

---

<sup>9</sup>Out-of-vocabulary считается частным случаем многозначного слова.

где  $R_{dynamic}$  – окончательное контекстуализованное представление слова,  $R_{ctx}$  – представление, выведенное только на основе контекста,  $R_{static}$  – статическое представление слова в языке,  $\alpha$  – коэффициент сглаживания ( $\alpha \in [0, 1]$ ). Важно отметить, что в ходе обучения контекстуализатора на MLM-задаче смешивание отключено ( $R_{dynamic} = R_{ctx}$ ). Оно включается лишь после того, как контекстуализатор будет способен порождать  $R_{ctx}$ .

Каждая словарная единица имеет собственный набор коэффициентов сглаживания для каждой компоненты представления (категор., ассоц. и грамм.). Иными словами:

- каждому лексико-семантическому представлению (строке одноименной матрицы на рисунке 2) соответствует два коэффициента сглаживания (категор. и ассоц.);
- каждому грамматическому представлению – один коэффициент.

Для получения коэффициентов сглаживания организуется особый этап обучения модели (коэффициенты будут обучаемыми весами в этой модели). Архитектура нейросети временно расширяется до схемы энкодер–декодер (рисунок 6). Сеть обучается предсказанию текста исходного предложения в

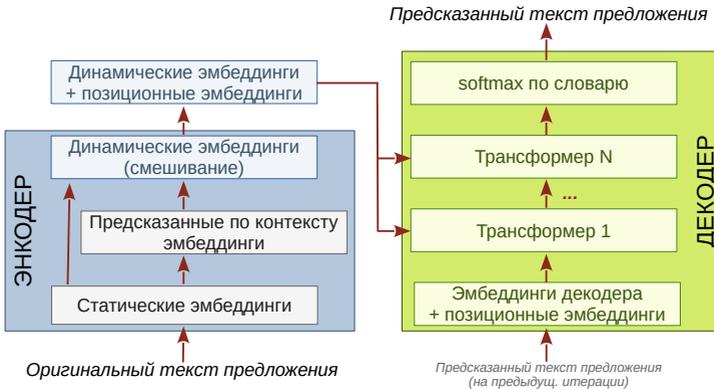


Рисунок 6. Архитектура для обучения коэффициентов сглаживания

условиях, когда словари энкодера и декодера различны. `SentencePiece`-словарь декодера строится на основе тех же текстовых данных, которые ранее использовались для MLM-обучения. Сам декодер реализуется на базе трансформерной архитектуры. Его особенностью является лишь то, что механизм внимания, обращенный в сторону энкодера, имеет только одну голову (хотя традиционно он `multi-head`). Это связано с тем,

что выходные вектора энкодера имеют предопределенную структуру, разбивать которую произвольным образом лишено смысла. В качестве функции ошибки используется перекрестная энтропия (`categorical_crossentropy loss`).

Во время обучения коэффициентов сглаживания в энкодере замораживаются не только статические эмбединги, но и веса, отвечающие за формирование представления на основе только лишь контекста ( $R_{ctx}$ ). Неиспользовавшаяся ранее логика смешивания статических и выведенных из контекста представлений включается в вычислительный процесс. Порождаемое энкодером представление дополняется генерируемой позиционной информацией (путем конкатенации).

В результате обучения происходит «настройка» коэффициентов сглаживания так, чтобы динамические эмбединги стали наиболее информативны для декодера. При этом декодер следует делать не слишком емким, для того чтобы ему затруднительно было восстанавливать текст предложения без качественных эмбедингов от энкодера (т.е. за счет собственных весов).

## 2. Эксперименты

### 2.1. Построение модели

Экспериментальная модель была обучена на данных корпуса русского литературного языка PaRuS [14]. Корпус содержит около 2.5 млрд. токенов. Текстовые данные в нем сегментированы, выполнены морфологическое и синтаксическое аннотирование, а также лемматизация.

Статические представления строились инструментарием `conll2vec`. Для эксперимента мы ограничились небольшими размерностями: 60 измерений для категориальной компоненты, 40 для ассоциативной и 20 для грамматической. Исследование статической модели на тестсетах для оценки семантической близости (RUSSE-15 [15] и RuSim1000 [12]) показало отсутствие *существенного* роста качества лексико-семантических представлений при дальнейшем увеличении размерности.

Построение статической модели включало следующие шаги: создание словарей, обучение лексико-семантических представлений для лемм (10 эпох), преобразование модели лемм в модель словоформ, обучение грамматических представлений (4 эпохи), преобразование словарей

и матриц к окончательному виду (группировка в «морфологические парадигмы»).

Наличие шумов в данных потребовало установить довольно высокие частотные пороги при формировании словарей. В результате были получены:

- словарь словоформ объемом 250 тыс. элементов;
- словарь псевдооснов – 40 тыс.;
- словарь суффиксов – 3 тыс. элементов.

Гиперпараметры модели контекстуализации были следующими:

- размер контекстного окна – по 5 токенов в каждом направлении;
- размерности LSTM – 768, 256 и 256 (для категориальной, ассоциативной и грамматической компонент, соответственно);
- размерности FF-слоев для порождения тематического представления – 256, 128 и 40.

Для обучения коэффициентов смешивания использовались следующие гиперпараметры:

- для кодирования позиционной информации использовалось 30 измерений (способ кодирования аналогичен BERT);
- в декодере использовалось 2 слоя трансформеров (размерность представлений 128, голов 8, размерность FF-слоев 256);
- словарь декодера на 10 тыс. входов построен алгоритмом *Sentence-Piece*.

В общей сложности обучение статической модели продолжалось около 10 часов. При этом необходимо принять во внимание, что для обучения использовался сервер с двумя CPU Intel Xeon E5540 – пиковая производительность данной установки равна 162 GFLOPS<sup>10</sup>. Для сравнения современные CPU high-end-сегмента имеют пиковую производительность 2–5 TFLOPS (например, десктопный Intel Xeon W-3175X). Таким образом, при использовании более современного оборудования построение статической модели заняло бы на порядок меньше времени на CPU и, вероятно, еще меньше при адаптации кодов *conll2vec* к GPU<sup>11</sup>.

---

<sup>10</sup>Всего 8 ядер с частотой 2.53 ГГц и 8 FLOP одинарной точности за такт.

<sup>11</sup>Пиковая производительность GPU Nvidia Tesla A100 составляет 19.5 TFLOPS (операций одинарной точности), Tesla V100 > 15 TFLOPS, Tesla P100 > 9 TFLOPS.

Модель контекстуализации обучалась на 6 Nvidia Tesla A100. Собственно контекстуализация (без коэффициентов смешивания статических и динамических представлений) обучалась в течение 3 часов, коэффициенты смешивания – в течение 2 часов.

В общей сложности обучение модели заняло 15 часов (из них 10 часов на несовременном оборудовании). Для сравнения BERT<sub>LARGE</sub> ее авторы обучали на 64 TPU в течение 4 дней.

## 2.2. Оценка модели

В качестве тестовой задачи для оценки построенного энкодера была выбрана задача разрешения лексической неоднозначности (WSD), а именно, разрешения неоднозначности для существительных, способных выражать событийное и несобытийное значение [16, 17]. Как и в указанных работах оценка производилась на шести тестовых словах. Для каждого слова в тестовом множестве представлено 100 примеров событийного значения и столько же несобытийного (в сумме 1200 примеров). Таким образом, random baseline по точности на данном тестсете составляет 50%.

Эксперимент повторял исследование с BERT, описанное в [17], за тем исключением, что вместо RuBERT [18] использовался предлагаемый энкодер. Таким образом, на «косвенных» данных тренировался классификатор, состоящий из энкодера (с замороженными весами), 128-мерного FF-слоя и softmax-слоя с двумя выходами. Для обучения использовался меньший объем данных, чем в эксперименте с RuBERT (тренировочное множество состояло из 368 тыс. примеров, валидационное – из 12 тыс.). Результат оценки классификатора и сравнение с аналогичным RuBERT-решением приведены в таблице 2 (для исследуемого энкодера приводится среднее в пяти экспериментах и СКО).

Из результатов оценки можно сделать следующие выводы.

- Существенное превосходство макроусредненной точности над random baseline говорит о работоспособности контекстуализации. Применение статических представлений в таком классификаторе дало бы ровно 50% точности.
- Энкодер уступает RuBERT в точности на этой задаче. Однако следует учитывать, что количество параметров в энкодере более чем в три раза меньше, чем в RuBERT.

Таблица 2. Результат оценки построенного энкодера на задаче разрешения лексической неоднозначности, точность в % (accuracy)

Тестовое слово	Исследуемый энкодер	RuBERT из [17]
<i>организация</i>	80.1 ± 1.2	<b>89.5</b>
<i>отделение</i>	<b>86.8 ± 0.9</b>	85.4
<i>отопление</i>	<b>58.9 ± 3.5</b>	51.4
<i>публикация</i>	50.5 ± 0	<b>77.3</b>
<i>стройка</i>	70.8 ± 2.1	<b>73.9</b>
<i>управление</i>	78.1 ± 2.4	<b>90.9</b>
<b>macro-average</b>	70.8	<b>78.1</b>

Таким образом, эксперимент показал лишь работоспособность подхода в целом. Для достижения более высоких показателей требуется дальнейшее совершенствование методологии и более тщательная настройка гиперпараметров энкодера.

### 3. Заключение

Несмотря на то, что исследованная модель энкодера пока не демонстрирует впечатляющих результатов на практических задачах, показана принципиальная возможность такого пути построения модели и работоспособность подхода в целом. Подход привлекателен тем, что методология может независимо совершенствоваться на разных уровнях: от методов построения статической модели до высокоуровневых надстроек над контекстуализацией. Преимуществами нового метода являются вычислительная эффективность за счет декомпозиции процесса обучения и естественная интерпретируемость промежуточных результатов. Последнее обеспечивается за счет явного выделения видов близости значений лексических единиц, а также за счет явного различения представлений слов в языке и в тексте.

Автор выражает искреннюю благодарность ЦКП «Центр данных ДВО РАН»  [19] за предоставление вычислительных ресурсов для выполнения экспериментов.

## Список литературы

- [1] Devlin J., Chang M-W., Lee K., Toutanova K. *BERT: Pre-training of Deep Bidirectional Transformers for language understanding.*– 2018.– 16 pp. [arXiv:1810.04805](#)  
- [2] Raffel C., Shazeer N., Roberts A., Lee K., Narang Sh., Matena M., Zhou Ya., Li W., Liu P. J. *Exploring the limits of transfer learning with a unified text-to-text transformer.*– 2020.– 67 pp. [arXiv:1910.10683](#)  
- [3] Brown T. B., Mann B., Ryder N., Subbiah M., Kaplan J., Dhariwal P., Neelakantan A., Shyam P., Sastry G., Askell A., Agarwal S., Herbert-Voss A., Krueger G., Henighan T., Child R., Ramesh A., Ziegler D. M., Wu J., Winter C., Hesse Ch., Chen M., Sigler E., Litwin M., Gray S., Chess B., Clark J., Berner Ch., McCandlish S., Radford A., Sutskever I., Amodei D. *Language models are few-shot learners.*– 2020.– 75 pp. [arXiv:2005.14165](#)  
- [4] Fedus W., Zoph B., Shazeer N. *Switch transformers: scaling to trillion parameter models with simple and efficient sparsity.*– 2021.– 40 pp. [arXiv:2101.03961](#)  
- [5] Kolesnikova A., Kuratov Y., Konovalov V., Burtsev M. *Knowledge distillation of Russian language models with reduction of vocabulary*, Proceedings of the International Conference «Dialogue 2022» (Moscow, June 15–18, 2022), Computational Linguistics and Intellectual Technologies.– vol. **21.**– 2022.– ISBN 978-5-7281-3205-9.– Pp. 295–310 . [URL](#)  [arXiv:2205.02340](#)  
- [6] Zafrir O., Boudoukh G., Izsak P., Wasserblat M. *Q8bert: Quantized 8bit bert*, 2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing – NeurIPS Edition (EMC2-NIPS) (13 December 2019, Vancouver, BC, Canada).– 2019.– Pp. 36–39.  [arXiv:1910.06188](#) 
- [7] Clark K., Luong M. -T., Le Q. V., Manning Ch. D. *Electra: Pre-training text encoders as discriminators rather than generators.*– 2020.– 18 pp. [arXiv:2003.10555](#)  
- [8] Kudo T., Richardson J. *SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing.*– 2018.– 6 pp. [arXiv:1808.06226](#)  
- [9] Sennrich R., Haddow B., Birch A. *Neural machine translation of rare words with subword units.*– 2016.– 11 pp. [arXiv:1508.07909](#)  
- [10] Schuster M., Nakajima K. *Japanese and Korean voice search*, 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (25–30 March 2012, Kyoto, Japan).– 2012.– Pp. 5149–5152.  
- [11] Mikolov T., Chen K., Corrado G., Dean J. *Efficient estimation of word representations in vector space.*– 2013.– 12 pp. [arXiv:1301.3781](#)  
- [12] Trofimov I. V., Suleymanova E. A. *A syntax-based distributional model for discriminating between semantic similarity and association*, Proceedings of the International Conference «Dialogue 2017».– V. 1, Computational Linguistics and Intellectual Technologies.– vol. **16.**– 2017.– Pp. 349–359. [URL](#)  
- [13] Трофимов И. В., Сулейманова Е. А. *Дистрибутивно-семантическая модель для выявления категориального сходства // Программные системы: теория и приложения.*– 2018.– Т. **9.**– № 4(39).– С. 443–460. [URL](#)  

- [14] Власова Н. А., Трофимов И. В., Сердюк Ю. П., Сулейманова Е. А., Воздвиженский И. Н. *PaRuS – синтаксически аннотированный корпус русского языка* // Программные системы: теория и приложения.– 2019.– Т. **10**.– № 4(43).– С. 181–199.    ↑
- [15] Panchenko A., Lukashevich N. V., Ustalov D., Paperno D., Meyer K. M., Konstantinova N. *RUSSE: The first workshop on Russian semantic similarity*, Proceedings of the International Conference «Dialogue 2015».– V. 2, Computational Linguistics and Intellectual Technologies.– vol. **14**.– RGGU.– 2015.– Pp. 89–105.   ↑
- [16] Трофимов И. В., Сулейманова Е. А., Власова Н. А., Подобрывев А. В. *Разрешение событийно-несобытийной неоднозначности существительных* // Программные системы: теория и приложения.– 2018.– Т. **9**.– № 4(39).– С. 3–33.    ↑
- [17] Трофимов И. В., Сердюк Ю. П., Сулейманова Е. А., Власова Н. А. *Разрешение событийно-несобытийной неоднозначности существительных: нейросетевой подход* // Программные системы: теория и приложения.– 2020.– Т. **11**.– № 4(47).– С. 31–53.    ↑
- [18] Kuratov Yu., Arkhipov M. *Adaptation of deep bidirectional multilingual transformers for Russian language*.– 2019.– 8 pp. arXiv: 1905.07213  ↑
- [19] Сорокин А. А., Макогонов С. В., Королев С. П. *Информационная инфраструктура для коллективной работы ученых Дальнего Востока России* // Научно-техническая информация. Серия 1: Организация и методика информационной работы.– 2017.– № 12.– С. 14–16.  ↑

Поступила в редакцию 13.11.2022;  
 одобрена после рецензирования 17.01.2023;  
 принята к публикации 09.02.2023;  
 опубликована онлайн 19.02.2023.

Рекомендовал к публикации

д.ф.-м.н. Н. В. Лукашевич

## Информация об авторе:



### Игорь Владимирович Трофимов

старший научный сотрудник Исследовательского центра искусственного интеллекта ИПС им. А. К. Айламазьяна, специалист по технологиям автоматической обработки естественного языка, извлечения информации, автоматического планирования.

 0000-0002-6903-4730

e-mail: [itrofimov@gmail.com](mailto:itrofimov@gmail.com)

Автор заявляет об отсутствии конфликта интересов.

UDC 81'322

 10.25209/2079-3316-2023-14-1-31-54

# Decomposition of construction method for a language encoder

Igor Vladimirovich **Trofimov**

Ailamazyan Program Systems Institute of RAS, Ves'kovo, Russia

 [itrofimov@gmail.com](mailto:itrofimov@gmail.com)

**Abstract.** An encoder as part of a language model is a mechanism for converting text information into an *effective* numerical representation which is suitable for solving a wide range of text processing tasks by means of neural network methods. This paper suggests a way of decomposing of the learning process for a language encoder. The author considers the issues of expediency of such decomposition taking into account reduction of computational costs, quality control at intermediate training stages, provision of the interpretability of the results on each stage. The quality evaluation of the encoder is given. (*In Russian*).

**Key words and phrases:** natural language processing, neural networks, language model, encoder, context-sensitive representations, lexical ambiguity resolution

2020 *Mathematics Subject Classification:* 68T07; 68T50

For citation: Igor V. Trofimov. *Decomposition of construction method for a language encoder*. Program Systems: Theory and Applications, 2023, **14**:1(56), pp. 31–54. (*In Russ.*). [https://psta.psiras.ru/read/psta2023\\_1\\_31-54.pdf](https://psta.psiras.ru/read/psta2023_1_31-54.pdf)

## References

- [1] J. Devlin, M-W. Chang, K. Lee, K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for language understanding*, 2018, 16 pp. [arXiv](#) [1810.04805](#) [doi](#)
- [2] C. Raffel, N. Shazeer, A. Roberts, K. Lee, Sh. Narang, M. Matena, Ya. Zhou, W. Li, P. J. Liu. *Exploring the limits of transfer learning with a unified text-to-text transformer*, 2020, 67 pp. [arXiv](#) [1910.10683](#) [doi](#)
- [3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, Ch. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, Ch. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei. *Language models are few-shot learners*, 2020, 75 pp. [arXiv](#) [2005.14165](#) [doi](#)
- [4] W. Fedus, B. Zoph, N. Shazeer. *Switch transformers: scaling to trillion parameter models with simple and efficient sparsity*, 2021, 40 pp. [arXiv](#) [2101.03961](#) [doi](#)
- [5] A. Kolesnikova, Y. Kuratov, V. Konovalov, M. Burtsev. “Knowledge distillation of Russian language models with reduction of vocabulary”, Proceedings of the International Conference “Dialogue 2022” (Moscow, June 15–18, 2022), Computational Linguistics and Intellectual Technologies, vol. **21**, 2022, ISBN 978-5-7281-3205-9, pp. 295–310. [URL](#) [doi](#) [arXiv](#) [2205.02340](#) [doi](#)
- [6] O. Zafrir, G. Boudoukh, P. Izsak, M. Wasserblat. “Q8bert: Quantized 8bit bert”, 2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing – NeurIPS Edition (EMC2-NIPS) (13 December 2019, Vancouver, BC, Canada), 2019, pp. 36–39. [doi](#) [arXiv](#) [1910.06188](#)
- [7] K. Clark, M. -T. Luong, Q. V. Le, Ch. D. Manning. *Electra: Pre-training text encoders as discriminators rather than generators*, 2020, 18 pp. [arXiv](#) [2003.10555](#) [doi](#)
- [8] T. Kudo, J. Richardson. *SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing*, 2018, 6 pp. [arXiv](#) [1808.06226](#) [doi](#)
- [9] R. Sennrich, B. Haddow, A. Birch. *Neural machine translation of rare words with subword units*, 2016, 11 pp. [arXiv](#) [1508.07909](#) [doi](#)
- [10] M. Schuster, K. Nakajima. “Japanese and Korean voice search”, 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (25–30 March 2012, Kyoto, Japan), 2012, pp. 5149–5152. [doi](#)
- [11] T. Mikolov, K. Chen, G. Corrado, J. Dean. *Efficient estimation of word representations in vector space*, 2013, 12 pp. [arXiv](#) [1301.3781](#) [doi](#)
- [12] I. V. Trofimov, E. A. Suleymanova. “A syntax-based distributional model for discriminating between semantic similarity and association”, Proceedings of the International Conference “Dialogue 2017”. V. 1, Computational Linguistics and Intellectual Technologies, vol. **16**, 2017, pp. 349–359. [URL](#)

- [13] I. V. Trofimov, E. A. Sulejmanova. “A dependency-based distributional semantic model for identifying taxonomic similarity”, *Program Systems: Theory and Applications*, **9**:4(39) (2018), pp. 443–460 (in Russian).  
- [14] N. A. Vlasova, I. V. Trofimov, Yu. P. Serdyuk, E. A. Sulejmanova, I. N. Vozdvizhenskij. “PaRuS — syntax annotated Russian corpus”, *Program Systems: Theory and Applications*, **10**:4(43) (2019), pp. 181–199 (in Russian).  
- [15] A. Panchenko, N. V. Lukashevich, D. Ustalov, D. Paperno, K. M. Meyer, N. Konstantinova. “RUSSE: The first workshop on Russian semantic similarity”, Proceedings of the International Conference “Dialogue 2015”. V. 2, Computational Linguistics and Intellectual Technologies, vol. **14**, RGGU, 2015, pp. 89–105.  
- [16] I. V. Trofimov, E. A. Sulejmanova, N. A. Vlasova, A. V. Podobryaev. “Disambiguation between eventive and non-eventive meaning of nouns”, *Program Systems: Theory and Applications*, **9**:4(39) (2018), pp. 3–33 (in Russian).  
- [17] I. V. Trofimov, Yu. P. Serdyuk, E. A. Sulejmanova, N. A. Vlasova. “Eventive vs. non-eventive sense of nouns: disambiguation using neural network approach”, *Program Systems: Theory and Applications*, **11**:4(47) (2020), pp. 31–53 (in Russian).  
- [18] Yu. Kuratov, M. Arkhipov. *Adaptation of deep bidirectional multilingual transformers for Russian language*, 2019, 8 pp. arXiv:1905.07213  
- [19] A. A. Sorokin, S. V. Makogonov, S. P. Korolev. “The information infrastructure for collective scientific work in the Far East of Russia”, *Scientific and Technical Information Processing*, **44**:4 (2017), pp. 302–304. 

УДК 519.248+519.872.8

 10.25209/2079-3316-2023-14-1-55-94

## Исследование стационарных характеристик многосерверных моделей с избыточностью

Александр Сергеевич Румянцев<sup>1✉</sup>, Диана Сергеевна Долгалева<sup>2</sup>,  
Александр Станиславович Головин<sup>3</sup>

Институт прикладных математических исследований КарНЦ РАН, Петрозаводск, Россия

<sup>1✉</sup>ar0@krc.karelia.ru

**Аннотация.** В работе предложены подходы к исследованию стохастических моделей вычислительных сетей из персональных компьютеров с помощью многосерверных систем обслуживания с избыточностью. Представлены перспективы и ограничения указанного подхода, а также возможные направления дальнейших исследований.

**Ключевые слова и фразы:** вычислительные сети из персональных компьютеров, многосерверные системы обслуживания, стохастическое моделирование, системы с избыточностью.

**Благодарности:** Авторы благодарны рецензентам за ценные замечания. Исследование выполнено за счет гранта РФФ 21-71-10135

Для цитирования: Румянцев А. С., Долгалева Д. С., Головин А. С. *Исследование стационарных характеристик многосерверных моделей с избыточностью* // Программные системы: теория и приложения. 2023. Т. 14. № 1(56). С. 55–94. [https://psta.psisras.ru/read/psta2023\\_1\\_55-94.pdf](https://psta.psisras.ru/read/psta2023_1_55-94.pdf)

## 1. Введение

Системы распределенных вычислений, такие как вычислительные сети из персональных компьютеров типа Desktop Grid (ВСПК), ВСПК уровня предприятия (Enterprise Desktop Grid), являются актуальным инструментом для объединения простаивающих ресурсов неспециализированных вычислителей (компьютеров, серверов) с целью решения вычислительно-математических задач [1]. Потенциально высокое быстродействие таких сетей компенсируется низкой надежностью, труднопрогнозируемым временем завершения расчетов, а также существенным ограничением на класс решаемых задач (решение которых можно разделить на множество слабо связанных подзадач, не требующих межпроцессорного обмена).

Для повышения надежности сети и достоверности получаемых результатов, в указанных сетях применяется *избыточность* на уровне подзаданий за счет *репликации* (создания копий подзадания с одними и теми же исходными данными) и *кворума* (числа копий, необходимых для успешного прохождения валидации и фиксации результата).

Избыточность информации является способом обеспечения надежной передачи данных между узлами вычислительных систем с помощью сетей передачи данных (в том числе беспроводных) [2]. Подобная задача возникает и в передовых системах хранения, основанных на технологии твердотельных накопителей, которые, в отличие от традиционных систем, обладают значительным внутренним резервом параллелизма, встроенной избыточностью, но существенно меньшим временем наработки на отказ отдельных модулей хранения [3]. В этой связи контролируемая избыточность, дублирование информации и высокая пропускная способность являются целевыми функциями для оптимизации параметров таких носителей [4].

Для анализа стационарных характеристик производительности телекоммуникационных и вычислительных систем целесообразно применять методы теории массового обслуживания в связи со случайным характером исследуемых процессов. Итогом применения указанных методов должно быть решение соответствующих задач по поиску оптимальных параметров конфигурации систем с избыточностью. Целью данной работы является анализ перспектив исследования класса многосерверных моделей с избыточностью, называемого системами разделения-слияния (split-merge), применяемых в качестве моделей вычислительных систем и систем передачи данных [5, 6]. Далее в работе кратко резюмированы основные результаты в области анализа систем разделения-слияния, получены новые результаты для стохастической упорядоченности ключевых характеристик и предложены направления дальнейших исследований указанного класса систем.

## 2. Обобщенные системы типа разделение-слияние

В данном разделе объектом исследования являются многосерверные системы обслуживания, в которых избыточность организована на уровне заявок (клиентов), поступающих в систему, с помощью репликации и кворума заявок. Такие системы являются моделями ВСПК (в том числе систем уровня предприятия) и ряда других систем (например, системы организации надежного хранения типа RAID [7]). Для исследования данного вида систем параллельного обслуживания можно применять многосерверные системы массового обслуживания с разделением и слиянием заявок. Именно, в момент поступления в систему заявка разделяется на несколько экземпляров, которые в дальнейшем обслуживаются в соответствии с выбранной дисциплиной. В соответствии с терминологией ВСПК, будем называть *подзаданием* экземпляр заявки на данном сервере. В зависимости от дисциплины обслуживания подзаданий, выделяют следующие виды систем:

- классические системы типа *разветвление-соединение* (Fork-Join, FJ), в которых подзадания попадают на независимые неограниченные очереди однотипных серверов, а обслуживание заявки завершается лишь в момент окончания обслуживания всех соответствующих подзаданий (в ряде случаев для целей накопления результатов выполнения подзаданий также рассматривается отдельная очередь, расположенная после серверов) [8–11];
- обобщенные системы типа *разветвление-соединение*, обозначаемые  $(r, q)$ -FJ, в которых для каждой заявки создаются  $r$  реплик (подзаданий), а окончание обслуживания соответствует кворуму из  $q$  результатов, при этом неостребованные  $r - q$  подзаданий снимаются с расчетов/из очередей (в частности, классической FJ является система  $(r, r)$ -FJ) [12];
- обобщенные системы типа *разветвление-ранняя отмена* (Fork-Early-Cancel), обозначаемые  $(r, q)$ -FEC, в которых из  $r$  реплик, в момент начала обслуживания не менее  $q$  подзаданий, неостребованные  $r - q$  подзаданий снимаются с расчетов/из очередей [12, 13];
- обобщенные системы типа *разделение-слияние* (Split-Merge), обозначаемые  $(c, r, q)$ -SM, в которых имеется одна общая очередь на обслуживание и  $c$  групп по  $r$  однотипных серверов каждая, подзадания каждого клиента ожидают освобождения (группы) серверов от подзаданий предшествующих клиентов и начинают

обслуживание на данной группе серверов одновременно, а окончанием обслуживания является достижение кворума  $q$  (невостребованные подзадания данной заявки снимаются с расчетов).

Отметим, что вероятностный анализ классических FJ-систем затруднен, поскольку входные потоки на сервера коррелированы, а размер очереди каждого сервера неограничен, что означает, потенциальную потребность в хранении неограниченного числа компонент для описания системы в виде цепи Маркова. В этой связи, как правило, для анализа классических FJ-систем прибегают к численному анализу или аппроксимации [7]. Однако, в ряде важных частных случаев удается свести анализ к известным классическим системам. В частности, в работах [12–14] доказаны следующие важные результаты:

- $(r, 1)$ -FJ система с типичным временем обслуживания подзадания  $S$  эквивалентна классической односерверной системе с типичным временем обслуживания  $S_{1,r}$ ;
- $(r, 1)$ -FEC система эквивалентна классической  $r$ -серверной системе обслуживания;
- время пребывания (время отклика) в  $(r, q)$ -FJ системе ограничено сверху временем отклика в  $(1, r, q)$ -SM системе;
- время пребывания (время отклика) в  $(c, r, q)$ -SM системе с типичным временем обслуживания подзадания  $S$  совпадает со временем пребывания в классической  $c$ -серверной системе с типичным временем обслуживания  $S_{q;r}$ ,

где  $S_{q;r}$  есть  $q$ -я порядковая статистика из  $r$  независимых одинаково распределенных случайных величин (с.в.), копий  $S$ . Это означает, в частности, что  $(1, r, 1)$ -SM система эквивалентна  $(r, 1)$ -FJ системе, а  $(c, 1, 1)$ -SM система эквивалентна  $(c, 1)$ -FEC системе. В этой связи далее приведем результаты для систем с разделяющимися клиентами типа  $(c, r, q)$ -SM.

Рассмотрим  $cr$ -серверную систему обслуживания, в которую с интенсивностью  $\lambda$  поступает поток заявок. Поступающая в систему, заявка ожидает одновременного освобождения группы из  $r$  серверов и создает  $r$  подзаданий, типичная (случайная) длительность которых,  $S$ , положительна. Подзадания данного клиента занимают освободившиеся  $r$  серверов одновременно, а при достижении кворума  $q$  (при окончании обслуживания  $q$  подзаданий), невостребованные  $r - q$  подзаданий покидают сервера немедленно. Это означает, что группа из  $r$  серверов занимается только

подзаданиями данного клиента, разделения данной группы серверов между различными клиентами не происходит. Таким образом, рассматриваемая система эквивалентна классической системе  $G/G/c$ , в которой типичное время обслуживания является порядковой статистикой  $S_{q:r}$ . Система стационарна в случае выполнения неравенства [15]

$$(1) \quad \lambda E S_{q:r} < c.$$

Целью анализа является, как правило, получение следующих ключевых характеристик производительности и эффективности систем в стационарном режиме в зависимости от числа серверов  $cr$ , параметров репликации  $r$  и кворума  $q$  (эти параметры использованы в обозначениях):

*время отклика системы*  $V_{r,q}$  – время с момента попадания клиента в систему до момента окончания его обслуживания (получения результатов  $q$  подзаданий)

$$(2) \quad V_{r,q} = W_{r,q} + S_{q:r},$$

где  $W_{r,q}$  есть стационарная задержка в системе;

*стоимость вычислений*  $C_{r,q}$  – ценовая характеристика, пропорциональная реальному времени вычислений клиента,

$$(3) \quad C_{r,q} = r S_{q:r}.$$

Отметим, что для ВСПК под стоимостью может подразумеваться энергопотребление, в системах облачных вычислений (в которых ресурсы размещаются в распределенном разделяемом пуле) стоимость представляет затраты в денежном эквиваленте, в системах хранения данных на основе твердотельных накопителей стоимость определяется как физический износ накопителя.

## 2.1. Стохастическая упорядоченность характеристик

Свойства указанных характеристик находятся в сложной зависимости от параметров репликации  $r$  и кворума  $q$  и, в частности, это связано с особенностями распределения порядковых статистик. Пусть с.в.  $S$  имеет функцию распределения  $F$ . Известно [16], что функция распределения порядковой статистики  $S_{q:r}$  имеет вид

$$(4) \quad F_{q:r}(x) = P\{S_{q:r} \leq x\} = \sum_{i=q}^r \binom{r}{i} F^i(x) \bar{F}^{r-i}(x),$$

где  $\bar{F}(x) = 1 - F(x)$  есть хвост распределения. Равенство (4) означает, в частности, что имеет место следующее (очевидное) стохастическое неравенство:

$$S_{1:r} \leq_{st} S \leq_{st} S_{r:r}.$$

Кроме того, при фиксированном  $r$  для  $q > 1$  имеет место следующее равенство [17]:

$$(5) \quad P(S_{q:r} \leq x) = P(S_{q-1:r} \leq x) - \binom{r}{q-1} F^{q-1}(x) \bar{F}^{r-q+1}(x).$$

Это означает, в частности, что

$$(6) \quad S_{q:r} \geq_{st} S_{q-1:r}.$$

Кроме того, при фиксированных  $r, q > 1$ , имеет место следующее равенство [17]

$$(7) \quad P(S_{q:r} \leq x) = P(S_{q-1:r-1} \leq x) - \binom{r-1}{q-1} F^{q-1}(x) \bar{F}^{r-q+1}(x).$$

В частности, это означает, что

$$(8) \quad S_{q:r} \geq_{st} S_{q-1:r-1}.$$

Наконец, при  $q \geq 1$  выполнено [17]

$$(9) \quad P(S_{q:r} \leq x) = P(S_{q:r-1} \leq x) + \binom{r-1}{q-1} F^q(x) \bar{F}^{r-q}(x).$$

Это означает, что  $S_{q:r} \leq_{st} S_{q:r-1}$ , что с учетом (6) и (8) окончательно влечет

$$(10) \quad S_{q:r} \geq_{st} S_{q-1:r-1} \geq_{st} S_{q-1:r}.$$

Из (3) и (6) следует, что  $C_{r,q}$  монотонно неубывает с возрастанием  $q$  (при фиксированном  $r$ ), т.е.

$$(11) \quad C_{r,q} \geq_{st} C_{r,q-1}, \quad q > 1.$$

Из (8) следует, что  $C_{r,q}$  монотонно неубывает при одновременном увеличении  $q$  и  $r$ , т.е.

$$(12) \quad C_{r,q} \geq_{st} C_{r-1,q-1}, \quad q, r > 1.$$

В то же время, в общем случае при фиксированном  $q$  с увеличением  $r$  величина  $C_{r,q}$  немонотонна, что следует из (10).

Для исследования монотонности времени пребывания в системе в зависимости от параметров репликации и кворума, нам потребуется следующий результат, являющийся прямым следствием Теоремы 4 в работе [18]:

**ТЕОРЕМА 1.** Пусть в системах  $\Sigma^{(1)}$  и  $\Sigma^{(2)}$  (управляющие последовательности которых обозначены соответствующим верхним индексом) имеет место следующий стохастический порядок для типичных времен между приходами  $T$  и времен обслуживания  $S$ ,

$$(13) \quad T^{(1)} \geq_{st} T^{(2)}, \quad S^{(1)} \leq_{st} S^{(2)}.$$

Тогда стационарные задержки  $W$  в системах упорядочены следующим образом,

$$(14) \quad W^{(1)} \leq_{st} W^{(2)}.$$

Тогда из равенства (2), Теоремы 1 и неравенства (10) следует, что справедливо утверждение.

**УТВЕРЖДЕНИЕ 1.** Для  $r, q > 1$  имеет место следующий стохастический порядок для стационарных времен отклика,

$$(15) \quad V_{r,q} \geq_{st} V_{r-1,q-1} \geq_{st} V_{r,q-1}.$$

Отметим, что стохастический порядок влечет, в частности, упорядоченность средних значений.

В общем случае выражение для среднего времени пребывания в системе  $EV_{r,q}$  для системы типа  $G/G/c$  неизвестно. Явный вид среднего времени отклика можно получить в случае  $c = 1$ , если входной поток является пуассоновским. Для этого можно воспользоваться формулой Поллачека–Хинчина

$$(16) \quad EW_{r,q} = \frac{\lambda ES_{q:r}^2}{2(1 - \lambda ES_{q:r})},$$

и получить из (3) выражение

$$(17) \quad EV_{r,q} = E(S_{q:r}) + \frac{\lambda E(S_{q:r}^2)}{2[1 - \lambda E(S_{q:r})]},$$

которое может быть использовано для расчета среднего времени отклика, если известны соответствующие выражения для моментов порядковых статистик.

Для иллюстрации стохастического порядка (15), построим графики функций распределения времени отклика в системе  $M/G/c$  при количестве серверов  $c = 3$ , интенсивности входного потока  $\lambda = 1$  и параметрах избыточности  $q = 3$ ,  $r = 5$  для случая, когда время обслуживания подзадачи имеет распределение Парето с плотностью

$$(18) \quad f(x) = \alpha x_0^\alpha x^{-\alpha-1}, \quad x > x_0, \quad (f(x) = 0, x \leq x_0).$$

Положим  $\alpha = 5$ ,  $x_0 = 1$  и заметим, что условие стационарности системы выполнено. Для указанных систем величина нагрузки  $\rho_{r,q} = \lambda E S_{q:r}$  равна  $\rho_{5,3} \approx 0.39$ ,  $\rho_{4,2} \approx 0.38$  и  $\rho_{5,2} = 0.37$ , соответственно. На рисунке 1 видна упорядоченность функций распределения, соответствующая стохастическому порядку (15). Графики построены с помощью метода идеальной выборки [19].

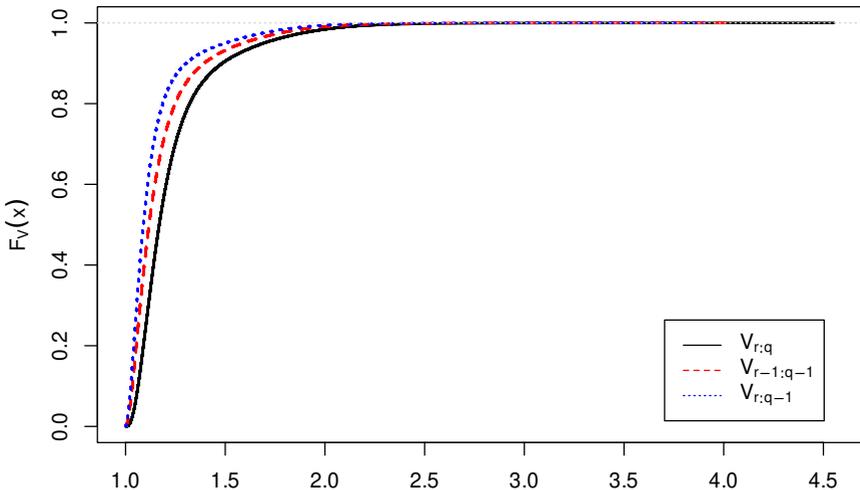


Рисунок 1. График функций распределения для системы  $M/G/c$  при единичной интенсивности входного потока на 3 сервера

## 2.2. Фазовый тип распределения времени обслуживания

Стоимость вычислений (3) может быть получена в явном виде для рассматриваемой модели в случае, когда время обслуживания имеет распределение фазового типа  $PH(\beta, \mathbf{B})$ , т.е. время обслуживания соответствует времени до поглощения конечной цепи Маркова с начальным распределением  $\beta$  и (субгенераторной) матрицей интенсивностей переходов между  $m$  невозвратными состояниями  $\mathbf{B}$  (подробнее об этом плотном классе распределений см., напр. [20]). В этом случае можно показать (см. [20, 21]), что распределение порядковой статистики  $S_{q:r}$  имеет вид  $PH(\beta_{q:r}, \mathbf{B}_{q:r})$ , где

$$(19) \quad \mathbf{B}_{q:r} = \begin{bmatrix} \mathbf{B}^{\oplus r} & \mathbf{b}_0^{\oplus r} & \mathbf{O} & \mathbf{O} & \dots & \mathbf{O} \\ \mathbf{O} & \mathbf{B}^{\oplus r-1} & \mathbf{b}_0^{\oplus r-1} & \mathbf{O} & \dots & \mathbf{O} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \dots & \mathbf{B}^{\oplus r-q+1} \end{bmatrix},$$

где неквадратные матрицы  $\mathbf{b}_0^{\oplus k}$  содержат интенсивности перехода между состояниями в связи с поглощением одной компоненты (реализацией одной из  $k$  независимых одинаково распределенных с.в. с распределением  $PH(\beta, \mathbf{B})$ ),

$$(20) \quad \mathbf{b}_0^{\oplus k} = \mathbf{b}_0 \otimes \mathbf{I}_{m^{k-1}} + \mathbf{I}_m \otimes \mathbf{b}_0 \otimes \mathbf{I}_{m^{k-2}} + \dots + \mathbf{I}_{m^{k-1}} \otimes \mathbf{b}_0,$$

$\mathbf{I}_j$  есть единичная матрица размера  $j \geq 1$ ,  $\otimes$  есть кронекерово произведение и  $\oplus$  есть кронекерова сумма матриц (соответственно, кратные произведение и сумма указаны в качестве верхнего индекса), а  $\mathbf{b}_0 = -\mathbf{B}\mathbf{1}$  есть вектор интенсивностей поглощения цепи, соответствующей распределению  $PH(\beta, \mathbf{B})$ . Начальное распределение процесса можно задать в следующем виде

$$(21) \quad \beta_{q:r} = (\beta^{\otimes r}, \mathbf{0}),$$

Это означает, что класс распределений фазового типа замкнут относительно порядковых статистик, а следовательно, условие стационарности (1) и стоимость  $EC_{r,q}$ , определенная в (3), принимают явный вид. Действительно, известно, что для с.в.  $X$  с распределением  $PH(\hat{\beta}, \hat{\mathbf{B}})$ , моменты первого и

второго порядка имеют вид [22]

$$(22) \quad \mathbf{E}X = -\hat{\boldsymbol{\beta}}\hat{\mathbf{B}}^{-1}\mathbf{1}, \quad \mathbf{E}X^2 = 2\hat{\boldsymbol{\beta}}\hat{\mathbf{B}}^{-2}\mathbf{1}.$$

Тогда, с учетом (22),

$$(23) \quad \mathbf{E}C_{r,q} = r\mathbf{E}S_{q:r} = -r\boldsymbol{\beta}_{q:r}\mathbf{B}_{q:r}^{-1}\mathbf{1},$$

где  $\boldsymbol{\beta}_{q:r}$  и  $\mathbf{B}_{q:r}$  определены в (21) и (19), соответственно.

Рассмотрим важный частный случай: пусть время обслуживания подзадания распределено экспоненциально с параметром  $\mu$ . Тогда  $\boldsymbol{\beta} = 1$ ,  $\mathbf{B} = -\mu$ ,  $\mathbf{b}_0 = \mu$  и можно преобразовать (23) к следующему виду

$$(24) \quad \mathbf{E}C_{r,q} = \frac{1}{\mu} \sum_{i=0}^{q-1} \frac{r}{r-i}.$$

В частности, при  $q = 1$ ,

$$(25) \quad \mathbf{P}\{S_{1:r} \leq x\} = 1 - e^{-r\mu},$$

поэтому выражение (24), примет вид

$$(26) \quad \mathbf{E}C_{r,1} = \mu^{-1},$$

т.е. стоимость вычислений не зависит от параметра  $r$  при  $q = 1$ .

В общем случае,

$$(27) \quad \mathbf{E}C_{r+1,q} - \mathbf{E}C_{r,q} = -\frac{1}{\mu} \left( \sum_{i=1}^{q-1} \frac{1}{(r-i+1)(r-i)} \right) \leq 0.$$

Значит, стоимость монотонно убывает по  $r$  при фиксированном  $q \geq 1$ .

Если входной поток в систему является пуассоновским (либо так называемым марковским входным потоком), то *время отклика* системы может быть найдено в явном виде, но на практике это может быть выполнено лишь численно. В ряде частных случаев это удается сделать явно.

Рассмотрим важный частный случай  $c = 1$ , т.е. репликация происходит на все узлы системы, и пусть входной поток в систему пуассоновский. Тогда  $(1, r, q)$ -SM система эквивалентна системе типа  $M/PH/1$ , для

которой время отклика (17), с учетом (22), примет вид

$$(28) \quad EV_{r,q} = \frac{\lambda \beta_{q:n} \mathbf{B}_{q:n}^{-2} \mathbf{1}}{1 + \lambda \beta_{q:n} \mathbf{B}_{q:n}^{-1} \mathbf{1}} - \beta_{q:r} \mathbf{B}_{q:r}^{-1} \mathbf{1}.$$

Если время обслуживания имеет экспоненциальное распределение с параметром  $\mu$ , и при этом  $q = 1$ , то анализ упрощается за счет выражения (25). В частности, в системе  $(c, r, 1)$ -SM для расчета времени отклика можно применить формулу Эрланга [23],

$$(29) \quad EV_{r,1} = \frac{1}{r\mu} + \frac{C(c, \lambda/(r\mu))}{c\mu - \lambda},$$

где так называемая С-формула Эрланга  $C(n, \rho)$  задает вероятность попадания в очередь в момент прихода в системе M/M/n при нагрузке  $\rho < n$ ,

$$(30) \quad C(n, \rho) = \left[ 1 + (1 - \rho) \frac{n!}{(n\rho)^n} \sum_{k=0}^{n-1} \frac{(n\rho)^k}{k!} \right]^{-1}.$$

В частности, время отклика системы  $(c, 1, 1)$ -SM совпадает с классическим временем отклика системы M/M/c:

$$(31) \quad EV_{1,1} = \frac{1}{\mu} + \frac{C(c, \lambda/\mu)}{c\mu - \lambda}.$$

Из (30) видно, что  $C(1, \rho) = \rho$ , поэтому время отклика (29) в системе  $(1, r, 1)$ -SM примет вид

$$EV_{r,1} = \frac{1}{r\mu} + \frac{\lambda}{r\mu(r\mu - \lambda)}.$$

### 2.3. Распределение времени обслуживания с тяжелым хвостом

С учетом (9), зависимость стоимостной характеристики  $C_{r,q}$  от параметров  $r, q$  необходимо исследовать для отдельных классов распределений. В данном разделе величина средней стоимости  $EC_{r,q}$  исследована для распределений Парето и Вейбулла, которые относятся (в последнем случае – при определенных параметрах) к классу распределений с тяжелым хвостом. Отметим, что такие распределения характерны для многих приложений в области вычислительных и телекоммуникационных систем [24].

Пусть типичное время обслуживания подзадания  $S$  имеет распределение Парето (18) с параметром  $\alpha > 1$ . Свойства порядковых статистик для с.в. с распределением Парето получены в работах [25, 26], именно,

$$(32) \quad \mathbb{E}(S_{q:r}^k) = x_0^k \frac{r!}{(r-q)!} \frac{\Gamma(r-q+1-k/\alpha)}{\Gamma(r+1-k/\alpha)}, \quad 0 < k < \alpha(r-q+1),$$

где  $\Gamma$  является гамма-функцией. Из (3) и (32) можно получить явный вид  $EC_{r,q}$ , а также следующее рекуррентное соотношение для величины  $EC_{r,q}$  [6]:

$$(33) \quad EC_{r+1,q} = \frac{(r+1)^2(r-q+1-1/\alpha)}{r(r-q+1)(r+1-1/\alpha)} EC_{r,q}.$$

Преобразовывая выражение (33), можно получить, что  $EC_{r,q}$  *возрастает* по  $r$ , если выполнено неравенство

$$(34) \quad \alpha r^2 + (2\alpha - \alpha q - q - 1)r - \alpha q + \alpha - 1 > 0,$$

в противном случае  $EC_{r,q}$  *невозрастает* по  $r$ . При  $\alpha > 1$  и фиксированном  $q \geq 1$ , квадратный трехчлен (относительно  $r$ ) в левой части (34) имеет два вещественных корня, левый из которых отрицателен, а правый положителен и имеет вид [6]:

$$(35) \quad r_0(q) = \frac{1+q+\alpha q - 2\alpha + \sqrt{(\alpha q + q - 1)^2 + 4q}}{2\alpha}.$$

Таким образом, при заданном фиксированном  $q$  неравенство (34) выполнено при (целых)  $r > r_0(q)$ , а минимум стоимости достигается в одном из двух ближайших целых к  $r_0(q)$ .

Из (35), в частности, следует, что  $r_0(1) < 1$  при  $\alpha > 1.5$ , т.е. стоимость  $EC_{r,1}$  возрастает по  $r$  для всех  $r \geq 1$ . Действительно,

$$(36) \quad EC_{r,1} = x_0 \frac{r^2 \alpha}{r\alpha - 1} = x_0 \left( r + \frac{1}{\alpha} + \frac{1}{\alpha(r\alpha - 1)} \right), \quad r > 1/\alpha.$$

Кроме того, из (36) следует, что имеет место асимптотически линейный рост стоимости

$$\lim_{r \rightarrow \infty} r^{-1} EC_{r,1} = x_0.$$

Из (36) следует, что  $EC_{1,1}$  может быть сколь угодно большим, если  $\alpha$  близко к 1. Действительно, из (32) или из (36) видно, что

$$EC_{1,1} = ES = x_0 \frac{\alpha}{\alpha - 1}.$$

В то же время, из (36) следует, что

$$EC_{2,1} < x_0 \left( 2 + \frac{2}{\alpha} \right),$$

поскольку  $\alpha r - 1 > 1$  для  $r \geq 2$ . Это означает, в частности, что  $EC_{2,1} < EC_{1,1}$  для  $\alpha < \sqrt{2}$ . Таким образом, с практической точки зрения репликация может быть полезна для существенного снижения стоимости вычислений при  $r < r_0(q)$ .

В отношении *времени отклика*, справедлив общий результат (15) о монотонности по параметрам избыточности, а также, с учетом (17) и (32), доступен явный вид в системе типа  $M/G/1$ . Однако более важной (в присутствии распределений с тяжелым хвостом) является проблема обеспечения конечности моментов (например, среднего или дисперсии) времени отклика системы. Конечность моментов, позволяет обеспечить надежное (доверительное) оценивание характеристик системы в стационарном режиме и (конечность в среднем) является необходимой для стационарности системы.

Обозначим  $\kappa(X)$  *моментный индекс* с.в.  $X$ , т.е. [27, 28]

$$(37) \quad \kappa(X) = \sup\{k > 0 : E(X^k) < \infty\} \leq \infty.$$

Тогда из (18) следует, что с.в.  $S$ , имеющая распределение Парето, имеет моментный индекс  $\alpha$ ,

$$(38) \quad \kappa(S) = \alpha,$$

при этом (32) означает, что

$$(39) \quad \kappa(S_{q;r}) = \alpha(r - q + 1) \geq \alpha.$$

Из (39) следует, что  $\kappa(S_{q;r})$  *линейно растет* с ростом  $r - q$ , т.е. с помощью параметров избыточности  $q, r$  можно обеспечить конечность требуемого момента (например, дисперсии) времени обслуживания заявки даже при малых значениях моментных индексов подзаданий (например, в случае если подзадания имеют неограниченную дисперсию длительности выполнения).

Известно, что моментный индекс времени ожидания в многосерверной системе типа  $G/G/c$  определяется моментным индексом времени обслуживания [29]. В рассматриваемой системе для заданных  $\lambda, \alpha, c, r, q$

моментный индекс времени ожидания  $W_{r,q}$  имеет вид [6]

$$(40) \quad \kappa(W_{r,q}) = (\alpha(r - q + 1) - 1)(c - \lfloor \rho \rfloor) \geq \alpha - 1,$$

где  $\rho = \lambda \mathbb{E}S_{q:r}$ , а  $\mathbb{E}S_{q:r}$  определено в (32). Тогда из (3), используя неравенство Минковского [30]

$$(\mathbb{E}(W_{r,q} + S_{q:r})^k)^{1/k} \leq (\mathbb{E}W_{r,q}^k)^{1/k} + (\mathbb{E}S_{q:r}^k)^{1/k},$$

можно заключить, что

$$(41) \quad \kappa(V_{r,q}) \geq \min(\kappa(W_{r,q}), \kappa(S_{q:r})),$$

где моментные индексы  $\kappa(W_{r,q})$  и  $\kappa(S_{q:r})$  определены в (40) и (39), соответственно.

Равенство (40) при  $c = 1$  и  $r = q$  согласуется с формулой Поллачека–Хинчина (16), из которой следует, что для обеспечения конечности среднего времени ожидания необходима конечность *второго момента* времени обслуживания. В заключение заметим, что линейный рост моментного индекса  $S_{q:r}$  по  $r - q$ , с учетом (41) и (40), может быть использован для подбора значений  $r, q$  таким образом, чтобы гарантировать требуемое значение моментного индекса (например, конечность дисперсии) времени пребывания клиента в системе даже если моментный индекс  $\kappa(S) = \alpha$  подзадания достаточно мал (например, при  $\alpha$  близком к единице).

Повторим анализ для системы, в которой типичное время выполнения подзадания  $S$  имеет распределение Вейбулла,

$$(42) \quad F(x) = 1 - e^{-x^\xi}, \quad \xi > 0.$$

Тогда моменты порядка  $k > 0$  для  $S_{q:r}$  могут быть получены в следующем виде [31, 32]

$$(43) \quad \mathbb{E}(S_{q:r}^k) = q \binom{r}{q} \Gamma(1 + k/\xi) \sum_{j=0}^{q-1} (-1)^j \binom{q-1}{j} (r - q + 1 + j)^{-1 - k/\xi}.$$

Таким образом, указанные моменты конечны для любого  $k > 0$ , т.е.  $\kappa(S_{q:r}) = \infty$ , а выражение (17) позволяет найти явный вид для  $\mathbb{E}V_{r,q}$  для системы типа  $M/G/1$ . Кроме того, имеет место следующее рекуррентное соотношение [17, 33]

$$(44) \quad \mathbb{E}(S_{q:r}) = \mathbb{E}(S_{q-1:r-1}) + \binom{r-1}{q-1} J(r - q + 1, q - 1),$$

где  $ES_{1:1} = J(1, 0) = \Gamma(1 + k/\xi)$ , а  $J(p, m)$  определяется рекуррентно:

$$J(p, m) = J(p, m - 1) + J(p + 1, m - 1), \quad J(p, 0) = p^{-1/\xi} \Gamma(1 + 1/\xi).$$

С учетом (43), имеем

$$(45) \quad EC_{r,q} = r \frac{r!}{(r-q)!} \frac{\Gamma(\xi^{-1})}{\xi} \sum_{j=0}^{q-1} (-1)^j \frac{(r-q+1+j)^{-1-\xi^{-1}}}{j!(q-1-j)!}.$$

Из (45) видно, что

$$(46) \quad EC_{r,1} = \Gamma(1 + 1/\xi) r^{1-1/\xi}, \quad r \geq 1.$$

Это означает, в частности, что  $EC_{r,1}$  неубывает (невозрастает) по  $r$  при  $\xi \geq 1$  ( $\xi \leq 1$ , соответственно). Кроме того, при  $\xi = 1$  распределение Вейбулла совпадает с экспоненциальным распределением, и при этом (46) согласуется с (26).

Напрямую использовать (45) для установления монотонности  $EC_{r,q}$  затруднительно. Однако, рассматривая разность  $\delta(r) = EC_{2,r+1} - EC_{2,r}$  и воспользовавшись рекурсией (44), можно установить, что

$$\delta(r) = \Gamma(1 + 1/\xi) \left( (2r^2 + r + 1)r^{-1/\xi} - r(r+1)^{1-1/\xi} - r^2(r-1)^{-1/\xi} \right).$$

В частности, при  $\xi < 1$  величина  $EC_{2,r}$  невозрастает по  $r$ . Действительно, исследуем функцию

$$\psi(r) = (2r^2 + r + 1)r^{-1/\xi} - r(r+1)^{1-1/\xi} - r^2(r-1)^{-1/\xi}.$$

Обозначим  $a = \xi^{-1} > 1$ . Преобразуем функцию к следующему виду

$$(47) \quad \psi(r) = r^2(2r^{-a} - (r+1)^{-a} - (r-1)^{-a}) + \phi(r),$$

где

$$\phi(r) = r(r+1)(r^{-a-1} - (r+1)^{-a-1}) = r^{1-a} + r^{-a} - (r+1)^{1-a} + (r+1)^{-a}.$$

Из выпуклости функции  $x^{-a}$  следует, что  $2r^{-a} - (r+1)^{-a} - (r-1)^{-a} \leq 0$ . Кроме того, можно убедиться, что  $\partial\phi(r)/\partial r \leq 0$  для всех  $a > 1$ , т.е. функция  $\phi(r)$  убывает по  $r \geq 2$ . При этом можно проверить, что  $\psi(2) \leq 0$  при любом  $\xi \in (0, 1)$ . Таким образом, из (47) следует, что рассматриваемая разность  $\delta(r)$  неположительна для всех  $r \geq 2$ , т.е.  $EC_{2,r}$  монотонно невозрастает по  $r$  при любом  $\xi \in (0, 1)$ .

### 3. Итоги и перспективы исследований

Таблица 1 представляет результаты описанного выше исследования влияния параметров репликации  $r$  и кворума  $q$  на стоимость  $C_{r,q}$  и время отклика  $V_{r,q}$  в многосерверных моделях систем с избыточностью типа  $(c, r, q) - SM$  для различных распределений времени обслуживания подзаданий  $S$ . Вид монотонности указан при увеличении  $r, q$ , в ряде случаев результат доступен для среднего значения, либо для соответствующей случайной величины (стохастический порядок). Результаты получены для случаев, когда время обслуживания имеет фазовый тип распределения, либо распределение Парето или Вейбулла (относящиеся к классу распределений с тяжелым хвостом).

Таблица 1. Результаты представленного исследования

Распределение	Характеристика	
	$C_{r,q}$	$V_{r,q}$
Общий случай	Немонотонна	Неубывает, неравенство (15)
Экспоненциальное	Невозрастает, неравенство (27); постоянна при $q = 1$ , формула (26); явный вид (24)	Явный вид (29) в системе типа $M/M/c$ при $q = 1$
Фазовый тип	Явный вид (23)	Явный вид (28) в системе типа $M/PH/1$
Парето	Глобальный минимум, выражение (35)	Явный вид в системе типа $M/G/1$ с учетом (17) и (32)
Вейбулла с параметром $\xi$	Невозрастает, если $\xi \leq 1, q = 1, 2$ ; неубывает при $\xi \geq 1, q = 1$ ; выражение (46)	Явный вид в системе типа $M/G/1$ с учетом (17) и (43)

С учетом полученных результатов, можно выделить следующие направления их дальнейшего развития:

- монотонность средней стоимости (27) получена для частного случая экспоненциально распределенных времен обслуживания подзадач, при этом интерес представляет общий случай произвольного фазового типа распределений;

- было бы целесообразно рассмотреть также неоднородный случай, например, когда сервера, на которые осуществляется репликация, имеют разную производительность (при этом корректировки требует формула (19) и дальнейшие результаты);
- монотонность стоимости для системы с распределением Вейбулла показана в разделе 2.3 для  $r = 1, 2$ , случай произвольного  $r$  требует отдельного исследования;
- наконец, в модели предполагается, что число блоков серверов  $c$  неизменно, при этом  $r$  является изменяемым параметром, целесообразно исследовать случай, когда число блоков определяется как  $s/r$  (целочисленно), где  $s$  является (неизменным) числом серверов в исходной системе.

Последнее направление открывает возможности для использования стохастических сравнений для систем с различным числом серверов, однако интересна также параметрическая монотонность систем (например, по параметру  $\alpha$  для распределения Парето или  $\xi$  для распределения Вейбулла).

### Список литературы

- [1] Chernov I., Nikitina N., Ivashko E. *Task scheduling in desktop grids: open problems* // Open Engineering.– 2017.– Vol. **7**.– No. 1.– Pp. 343–357. doi ↑56
- [2] Verma N., Singh D. *Data redundancy implications in wireless sensor networks* // Procedia Computer Science.– 2018.– Vol. **132**.– Pp. 1210–1217. doi ↑56
- [3] Rummyantsev A., Ivashko E., Chernov I., Kositsyn D., Shabaev A., Ponomarev V. *Latency/wearout in a flash-based storage system with replication on write*, 2019 24th Conference of Open Innovations Association (FRUCT) (08–12 April 2019, Moscow, Russia).– Pp. 360–366. doi ↑56
- [4] Chernov I., Ivashko E., Kositsyn D., Ponomarev V., Rummyantsev A., Shabaev A. *Flash-based storage deduplication techniques: A survey* // International Journal Of Embedded And Real-Time Communication Systems (IJERTCS).– 2019.– Vol. **10**.– No. 3.– Pp. 32–48. doi ↑56
- [5] Osipov O., Rogachko E. *Fork-join queueing systems with heterogeneous servers threshold control policy*, SMARTY2020 (August 16–20, 2020, Petrozavodsk, Russia), CEUR Workshop Proc.– vol. **2792**.– 2020.– Pp. 76–88. URL ↑56
- [6] Rummyantsev A., Chakravarthy S., Morozov E., Remnev S. *Cost and effect of replication and quorum in desktop grid computing* // *Information Technologies And Mathematical Modelling. Queueing Theory And Applications*, Springer CCIS Series.– vol. **912**, Cham: Springer.– 2018.– ISBN 978-3-319-97594-8.– Pp. 143–156.

- [7] Thomasian A. *Analysis of fork/join and related queueing systems* // ACM Computing Surveys.– 2014.– Vol. **47**.– No. 2.– id. 17.– 71 pp. doi ↑<sup>57, 58</sup>
- [8] Baccelli F., Makowski A., Shwartz A. *The fork-join queue and related systems with synchronization constraints: Stochastic ordering and computable bounds* // Advances in Applied Probability.– 1989.– Vol. **21**.– No. 3.– Pp. 629–660. doi ↑<sup>57</sup>
- [9] Gardner K., Zbarsky S., Doroudi S., Harchol-Balter M., Hyytiä E., Scheller-Wolf A. *Queueing with redundant requests: Exact analysis* // Queueing Systems.– 2016.– Vol. **83**.– No. 3–4.– Pp. 227–259. doi ↑<sup>57</sup>
- [10] Kim C., Agrawala A. K. *Analysis of the fork-join queue* // IEEE Transactions on Computers.– 1989.– Vol. **38**.– No. 2.– Pp. 250–255. doi ↑<sup>57</sup>
- [11] Nelson R., Tantawi A. N. *Approximate analysis of fork/join synchronization in parallel queues* // IEEE Transactions on Computers.– 1988.– Vol. **37**.– No. 6.– Pp. 739–743. doi ↑<sup>57</sup>
- [12] Joshi G., Soljanin E., Wornell G. *Queues with redundancy: Latency-cost analysis* // ACM SIGMETRICS Performance Evaluation Review.– 2015.– Vol. **43**.– No. 2.– Pp. 54–56. doi ↑<sup>57, 58</sup>
- [13] Joshi G. *Efficient redundancy techniques to reduce delay in cloud systems*, PhD Thesis.– Massachusetts Institute of Technology.– 2016. ↑<sup>57, 58</sup>
- [14] Joshi G., Soljanin E., Wornell G. *Efficient redundancy techniques for latency reduction in cloud systems* // ACM Transactions on Modeling and Performance Evaluation of Computing Systems.– 2017.– Vol. **2**.– No. 2.– id. 12.– 30 pp. doi ↑<sup>58</sup>
- [15] Asmussen S. *Applied Probability and Queues*, SMAP.– Vol. **51**.– Springer.– 2003.– ISBN 978-0-387-00211-8.– 452 pp. doi ↑<sup>59</sup>
- [16] Balakrishnan N. *Permanents, order statistics, outliers and robustness* // Revista Matemática Complutense.– 2007.– Vol. **20**.– No. 1.– Pp. 7–107. doi ↑<sup>59</sup>
- [17] Balakrishnan N., Joshi P. C. *A note on order statistics from Weibull distribution* // Scandinavian Actuarial Journal.– 1981.– Vol. **1981**.– No. 2.– Pp. 121–122. doi ↑<sup>60, 68</sup>
- [18] Whitt W. *Comparing counting processes and queues* // Advances in Applied Probability.– 1981.– Vol. **13**.– No. 1.– Pp. 207–220. doi ↑<sup>61</sup>
- [19] Blanchet J., Pei Y., Sigman K. *Exact sampling for some multi-dimensional queueing models with renewal input* // Advances in Applied Probability.– 2019.– Vol. **51**.– No. 4.– Pp. 1179–1208. doi ↑<sup>62</sup>
- [20] Bladt M., Nielsen B. *Matrix-Exponential Distributions in Applied Probability*, PTSM.– Vol. **81**.– Boston, MA: Springer US.– 2017.– ISBN 978-1-4939-7047-6.– 749 pp. doi ↑<sup>63</sup>
- [21] Rumyantsev A., Chakravarthy S. *Matrix-exponential distributions in applied probability* // Proceedings of the Third International Conference BOINC-based High Performance Computing: Fundamental Research and Development, BOINC:FAST 2017 (August 28–September 01, 2017, Petrozavodsk, Russia), CEUR Workshop Proc.– vol. **1973**.– 2017.– Pp. 27–34. URL ↑<sup>63</sup>

- [22] Buchholz P., Kriege J., Felko I. *Input Modeling with Phase-Type Distributions and Markov Models. Theory and Applications*, SpringerBriefs in Mathematics.– Cham: Springer International Publishing.– 2014.– ISBN 978-3-319-06673-8.– xii+127 pp. [doi](#) ↑<sup>64</sup>
- [23] Harchol-Balter M. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*.– Cambridge: Cambridge University Press.– 2013.– ISBN 9781107027503.– 574 pp. ↑<sup>65</sup>
- [24] Feitelson D. *Workload Modeling for Computer Systems Performance Evaluation*.– Cambridge: Cambridge University Press.– 2015.– ISBN 9781139939690. [doi](#) ↑<sup>65</sup>
- [25] Malik H. J. *Exact moments of order statistics from the Pareto distribution* // Scandinavian Actuarial Journal.– 1966.– Vol. **1966**.– No. 3–4.– Pp. 144–157. [doi](#) ↑<sup>66</sup>
- [26] Huang J. S. *A note on order statistics from Pareto distribution* // Scandinavian Actuarial Journal.– 1975.– Vol. **1975**.– No. 3.– Pp. 187–190. [doi](#) ↑<sup>66</sup>
- [27] Daley D. J., Goldie C. M. *The moment index of minima (II)* // Statistics & Probability Letters.– 2006.– Vol. **76**.– No. 8.– Pp. 831–837. [doi](#) ↑<sup>67</sup>
- [28] Daley D. J. *The moment index of minima* // Journal of Applied Probability.– 2001.– Vol. **38**, Issue A: Probability, Statistics and Seismology.– Pp. 33–36. [doi](#) ↑<sup>67</sup>
- [29] Scheller-Wolf A., Vesilo R. *Sink or swim together: necessary and sufficient conditions for finite moments of workload components in FIFO multiserver queues* // Queueing Systems.– 2011.– Vol. **67**.– No. 1.– Pp. 47–61. [doi](#) ↑<sup>67</sup>
- [30] Hardy G. H., Littlewood J. E., Pólya G. *Inequalities*, 2nd ed.– Cambridge: Cambridge University Press.– 1991.– 340 pp. ↑<sup>68</sup>
- [31] Lieblein J. *On moments of order statistics from the Weibull distribution* // Ann. Math. Statist..– 1955.– Vol. **26**.– No. 2.– Pp. 330–333. [doi](#) ↑<sup>68</sup>
- [32] Sultan K. S., Moshref M. E. *Moments of order statistics from Weibull distribution in the presence of multiple outliers* // Communications in Statistics – Theory and Methods.– 2014.– Vol. **43**.– No. 10–12.– Pp. 2214–2226. [doi](#) ↑<sup>68</sup>
- [33] Rinne H. *The Weibull Distribution. A Handbook*.– New York: Chapman and Hall/CRC.– 2009.– ISBN 9780367577469.– 808 pp. ↑<sup>68</sup>

Поступила в редакцию	06.12.2022;
одобрена после рецензирования	27.01.2023;
принята к публикации	05.02.2023;
опубликована онлайн	15.02.2023.

Рекомендовал к публикации

д.т.н. А. М. Цирлин

## Информация об авторах:



### Александр Сергеевич Румянцев

с.н.с., к.ф.-м.н., научные интересы: теория массового обслуживания, многосерверные системы, стохастическое моделирование и анализ, высокопроизводительные и распределенные вычисления, матрично-аналитический метод, регенеративный подход, трехуровневое моделирование

 0000-0003-2364-5939

*e-mail:* [ar0@krc.karelia.ru](mailto:ar0@krc.karelia.ru)



### Диана Сергеевна Долгалева

стажер-иссл., научные интересы: теория массового обслуживания, матрично-аналитический метод, биоинформатика, генные сети, распределенные вычисления

 0000-0002-0606-6583

*e-mail:* [abcdi\\_do@mail.ru](mailto:abcdi_do@mail.ru)



### Александр Станиславович Головин

вед.инж.-иссл., аспирант, научные интересы: теория массового обслуживания, матрично-аналитический метод, энергоэффективность, имитационное моделирование, распределенные вычисления, многосерверные системы обслуживания

 0000-0003-1325-3739

*e-mail:* [golovin@krc.karelia.ru](mailto:golovin@krc.karelia.ru)

*Все авторы сделали эквивалентный вклад в подготовку публикации.  
Авторы заявляют об отсутствии конфликта интересов.*

UDC 519.248+519.872.8

 10.25209/2079-3316-2023-14-1-55-94


## Steady-state performance analysis of multiserver queueing models with redundancy

Alexander Sergeevich **Rumyantsev**<sup>1</sup>, Diana Sergeevna **Dolgaleva**<sup>2</sup>,  
Alexander Stanislavovich **Golovin**<sup>3</sup>

Institute of Applied Mathematical Research KarRC of RAS, Petrozavodsk, Russia

<sup>1</sup>*✉* ar0@krc.karelia.ru

**Abstract.** An approach to study of the stochastic models of distributed computing systems by means of the multiserver queueing models with redundancy is suggested. Perspectives and limitations of the approach, as well as possible future research directions are presented.

**Key words and phrases:** distributed computing systems, multiserver queueing systems, stochastic modeling, systems with redundancy

2020 *Mathematics Subject Classification:* 60K30; 90B22, 65C30

**Acknowledgments:** The authors are grateful to the referees for valuable comments. The study was supported by the Russian Science Foundation grant 21-71-10135

**For citation:** Alexander S. Rumyantsev, Diana S. Dolgaleva, Alexander S. Golovin. *Steady-state performance analysis of multiserver queueing models with redundancy*. Program Systems: Theory and Applications, 2023, **14**:1(56), pp. 55–94. [https://psta.pspiras.ru/read/psta2023\\_1\\_55-94.pdf](https://psta.pspiras.ru/read/psta2023_1_55-94.pdf)

## 1. Introduction

Distributed computing systems, such as Desktop Grid (DG), Enterprise Desktop Grid (EDG), are widely used for utilizing the idle resources of the generic computing resources (personal computers/servers) in order to solve computationally intensive tasks [1]. The potentially high performance of such networks coincides with low reliability, hardly predictable task completion time, as well as a significant limitation on the class of tasks to be solved (the solution of such tasks needs to be divided into many loosely coupled subtasks that do not require interprocessor exchange).

To increase the reliability of the network and the reliability of the results obtained, *redundancy* at the task level is used in the DG by means of *replication* (i.e. copying a task with the same input) and *quorum* (the number of replicas required for successful validation and result assimilation).

Redundancy of information is a way to ensure reliable data transmission between nodes of computing systems using data transmission networks (including wireless) [2]. A similar problem arises in advanced storage systems based on solid-state drive technology, which, unlike traditional systems, have a significant internal reserve of parallelism, built-in redundancy, but significantly less time to failure of individual storage modules [3]. In this regard, controlled redundancy, duplication of information and high throughput are the target functions for optimizing the parameters of such media [4].

To analyze the stationary performance characteristics of telecommunication and computing systems, the methods of queuing theory are used due to the random nature of the processes under study. As a result of such a study, an optimal configuration parameters of such systems with redundancy may arise. The purpose of this work is to analyze the perspectives for the study of a class of multi-server models with redundancy, called split-merge systems, used as models of computing systems and data transmission systems [5, 6]. Further, the paper briefly summarizes the main results in the field of analysis of the split-merge systems, new results for stochastic ordering of key characteristics are obtained and directions for further research of this class of systems are proposed.

## 2. Generalized Split-Merge-type Systems

In this section we study the multiserver queueing systems with task-level redundancy organized by means of replication and quorum mechanisms. Such a systems are the models of the DG (including EDG) and many other systems such as reliable storage systems, e.g. RAID [7]. To study such systems, multiserver queueing systems with task split and merge may be used. Specifically, upon arrival, each task is split into multiple copies which are then processed according to the selected service discipline. Using the terminology adopted in DG systems, we call such tasks *workunits*. According to the service discipline, such systems fall into one of the following categories:

- classical *Fork-Join* (FJ) where the workunits enter independent queues of (identical) independent servers, whereas the service of a specific task is complete only upon completion of all the corresponding workunits (in such a case usually a specific unbounded buffer accumulating the completed workunits is used) [8–11];
- generalized *Fork-Join* systems, denoted as  $(r, q)$ -FJ, in which  $r$  replicas of a task (workunits) are created and service completion is considered to be the time at which  $q$  earliest results of the workunits of a specific task are obtained, whereas the unnecessary  $r - q$  workunits are immediately removed from the servers/queues (in particular, the classical FJ corresponds to the generalized  $(r, r)$ -FJ) [12];
- generalized *Fork-Early-Cancel* systems, denoted as  $(r, q)$ -FEC, where each task is copied into  $r$  replicas, and upon the service initialization of not less than  $q$  workunits, the unnecessary  $r - q$  workunits are cancelled from the servers/queues [12, 13];
- generalized *Split-Merge* systems, denoted as  $(c, r, q)$ -SM, in which a single queue and a pool of  $c$  groups of  $r$  identical servers each are used, and the workunits of each task are dispatched to a specific group of servers. The group of servers dedicated to a task is seized by the workunits simultaneously (for random independent service times), whereas the service completion epoch is the time of obtaining  $q$  completed workunits (the unnecessary workunits are cancelled, and the group of servers is simultaneously released).

We note that stochastic analysis of the classical FJ systems is rather complicated, since the arrival processes of the servers in such a system are correlated, and the queue sizes are unbounded, requiring a potentially unbounded storage needed to keep the system description Markovian. As such, the classical FJ systems are generally studied by means of numerical analysis and approximation [7]. However, in some important special cases such an analysis may be performed by using the results for classical queueing systems. In particular, the following important properties were obtained in the papers [12–14]:

- $(r, 1)$ -FJ system with (generic) workunit service time  $S$  is equivalent to a classical single-server queue with generic service time  $S_{1:r}$ ;
- $(r, 1)$ -FEC system is equivalent to a classical  $r$ -server system;
- sojourn (response) time in the  $(r, q)$ -FJ system is upper bounded by the sojourn time in  $(1, r, q)$ -SM system;
- sojourn time in the  $(c, r, q)$ -SM system with generic service time of a workunit  $S$  coincides with the sojourn time in a  $c$ -server system with generic service time  $S_{q:r}$ ,

where  $S_{q:r}$  is the  $q$ -th order statistics from  $r$  independent identically distributed (iid) random variables (r.v.), stochastic copies of  $S$ . In particular this means that  $(1, r, 1)$ -SM system is equivalent to  $(r, 1)$ -FJ system, whereas  $(c, 1, 1)$ -SM system is equivalent to  $(c, 1)$ -FEC system. In this regard, we focus on studying  $(c, r, q)$ -SM systems in what follows.

We consider the  $cr$ -server queueing system in which the tasks arrive with input rate  $\lambda$ . An arriving task is waiting for (simultaneous) release of the group of  $r$  servers and is split into  $r$  workunits, each having iid service times, stochastic copies of the generic positive random service time  $S$ . All the workunits of a given task simultaneously seize the group of  $r$  servers, and upon receiving the quorum of  $q$  completed workunits, the remaining (busy)  $r - q$  servers immediately are released. This means that the group of  $r$  servers is seized only by the workunits of a given task, and there is no split of such a group between different tasks. As such, the

system under consideration is equivalent to the classical  $G/G/c$  system in which the typical service time is the order statistics  $S_{q:r}$ . The stability condition of such a systems is as follows, [15]

$$(1) \quad \lambda E S_{q:r} < c.$$

The main goal of the analysis is, in general, to obtain the dependence of the following key performance and efficiency characteristics on the number of servers,  $cr$ , replication  $r$  and quorum  $q$  (these parameters are used hereafter in the notation):

*sojourn time*  $V_{r,q}$  – the time interval from the arrival of a (generic) task into the system until service completion (obtaining  $q$  completed workunits)

$$(2) \quad V_{r,q} = W_{r,q} + S_{q:r},$$

where  $W_{r,q}$  is the (stationary) generic delay in the system;

*computation cost*  $C_{r,q}$  – the characteristic proportional to the computation time of a task,

$$(3) \quad C_{r,q} = r S_{q:r}.$$

Note that in DG the cost may be thought of as the energy consumption of the system, whereas in the cloud computing such a cost directly coincides with the expenses for resources rent, in the solid state drive-based storage system such a cost may be considered as the wearout of the storage units.

## 2.1. Stochastic monotonicity of the performance

The properties of these characteristics are in a complex dependence on the replication parameters  $r$  and the quorum  $q$  and, in particular, this is due to the peculiarities of the order statistics distribution. Let r.v.  $S$  have a distribution function (cdf)  $F$ . It is known [16] that the order statistics cdf  $S_{q:r}$  has the form

$$(4) \quad F_{q:r}(x) = P\{S_{q:r} \leq x\} = \sum_{i=q}^r \binom{r}{i} F^i(x) \bar{F}^{r-i}(x),$$

where  $\bar{F}(x) = 1 - F(x)$  is the tail of distribution. The equality (4) means, in particular, that the following (obvious) stochastic inequality holds

$$S_{1:r} \leq_{st} S \leq_{st} S_{r:r}.$$

Moreover, for a fixed  $r$  and  $q > 1$  the following equality holds [17]

$$(5) \quad P(S_{q:r} \leq x) = P(S_{q-1:r} \leq x) - \binom{r}{q-1} F^{q-1}(x) \bar{F}^{r-q+1}(x).$$

This means, in particular, that

$$(6) \quad S_{q:r} \geq_{st} S_{q-1:r}.$$

Moreover, for fixed  $r, q > 1$  the following equality holds [17]

$$(7) \quad P(S_{q:r} \leq x) = P(S_{q-1:r-1} \leq x) - \binom{r-1}{q-1} F^{q-1}(x) \bar{F}^{r-q+1}(x).$$

In particular this means that

$$(8) \quad S_{q:r} \geq_{st} S_{q-1:r-1}.$$

Finally for  $q \geq 1$  holds [17]

$$(9) \quad P(S_{q:r} \leq x) = P(S_{q:r-1} \leq x) + \binom{r-1}{q-1} F^q(x) \bar{F}^{r-q}(x).$$

This means that  $S_{q:r} \leq_{st} S_{q:r-1}$  which taking account of (6) and (8) leads to

$$(10) \quad S_{q:r} \geq_{st} S_{q-1:r-1} \geq_{st} S_{q-1:r}.$$

It follows from (3) and (6) that  $C_{r,q}$  is monotonically non-decreasing as  $q$  increases (for a fixed  $r$ ), i.e.

$$(11) \quad C_{r,q} \geq_{st} C_{r,q-1}, \quad q > 1.$$

It follows from (8) that  $C_{r,q}$  is monotonically non-decreasing as  $q$  and  $r$  increase simultaneously, i.e.

$$(12) \quad C_{r,q} \geq_{st} C_{r-1,q-1}, \quad q, r > 1.$$

At the same time, in the general case, for a fixed  $q$ , as  $r$  increases, the quantity  $C_{r,q}$  is nonmonotonic, which follows from (10).

To study the monotonicity of the sojourn time in the system depending on the parameters of replication and quorum, we need the following result, which is a direct consequence of Theorem 4 in [18]:

**THEOREM 1.** *Let the systems  $\Sigma^{(1)}$  and  $\Sigma^{(2)}$  (whose escape sequences are denoted by the corresponding superscript) have the following stochastic order for typical times between arrivals  $T$  and service times  $S$ ,*

$$(13) \quad T^{(1)} \geq_{st} T^{(2)}, \quad S^{(1)} \leq_{st} S^{(2)}.$$

*then the stationary delays  $W$  in the systems are ordered as follows,*

$$(14) \quad W^{(1)} \leq_{st} W^{(2)}.$$

Then it follows from equality (2), Theorem 1 and inequality (10) that the assertion is true.

**STATEMENT 1.** *For  $r, q > 1$ , the following stochastic order for stationary sojourn times takes place,*

$$(15) \quad V_{r,q} \geq_{st} V_{r-1,q-1} \geq_{st} V_{r,q-1}.$$

Note that the stochastic order implies, in particular, the ordering of the mean values.

In general the expression for the average sojourn time  $EV_{r,q}$  for a system of type  $G/G/c$  is unknown. An explicit form of the average sojourn time can be obtained in the case of  $c = 1$  if the arrival process is Poisson. To do this, you can use the Pollacek—Khinchin formula

$$(16) \quad EW_{r,q} = \frac{\lambda ES_{q:r}^2}{2(1 - \lambda ES_{q:r})},$$

and get from (3) the expression

$$(17) \quad EV_{r,q} = E(S_{q:r}) + \frac{\lambda E(S_{q:r}^2)}{2[1 - \lambda E(S_{q:r})]},$$

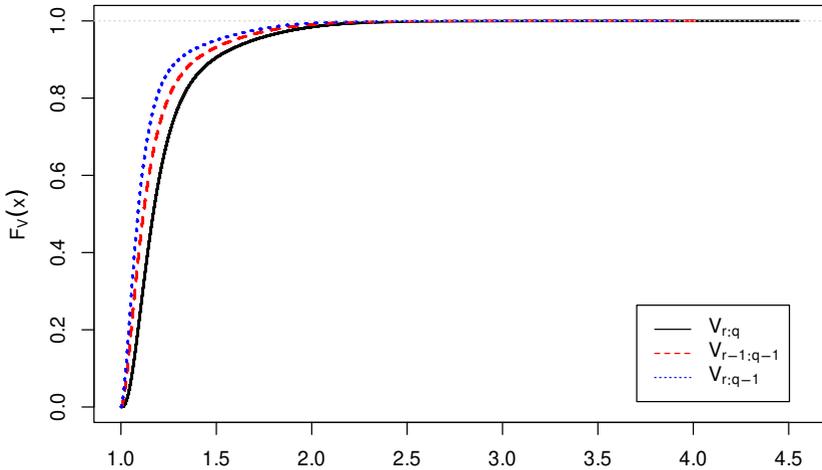


FIGURE 1. Cumulative distribution functions for the system  $M/G/c$  with the input rate  $\lambda = 1$  and the number of servers  $c = 3$

which can be used to calculate the average sojourn time if the corresponding expressions for the moments of the order statistics are known.

To illustrate the stochastic order (15), let's plot the sojourn time distribution functions in the  $M/G/c$  system for the case when the service time has Pareto distribution whose pdf has the form

$$(18) \quad f(x) = \alpha x_0^\alpha x^{-\alpha-1}, \quad x > x_0, \quad (f(x) = 0, x \leq x_0).$$

Let's set  $\alpha = 5$ ,  $x_0 = 1$  and redundancy parameters  $r = 5$ ,  $q = 3$  under input rate  $\lambda = 1$  and the number of servers  $c = 3$ . For these systems, the load  $\rho_{r,q} = \lambda ES_{q;r}$  is  $\rho_{5,3} \approx 0.39$ ,  $\rho_{4,2} \approx 0.38$  and  $\rho_{5,2} = 0.37$ , respectively. Note that the stationarity condition of the system is satisfied. Figure 1 shows the ordering of distribution functions corresponding to the stochastic order (15). Note that the graphics were built using the exact sampling method [19].

## 2.2. Phase-type service time distribution

The computation cost (3) may be obtained explicitly for the considered system in case the sojourn time has the so-called phase type distribution  $PH(\boldsymbol{\beta}, \mathbf{B})$ , i.e. the service time constitutes to the absorption time of a finite state space Markov chain with initial distribution  $\boldsymbol{\beta}$  and (subgenerator) transition rate matrix for  $m$  transient states  $\mathbf{B}$  (for more details on this class of distributions see e.g. [20]). In such a case it can be shown (see [20, 21]) that the distribution of order statistics  $S_{q:r}$  is given as  $PH(\boldsymbol{\beta}_{q:r}, \mathbf{B}_{q:r})$ , where

$$(19) \quad \mathbf{B}_{q:r} = \begin{bmatrix} \mathbf{B}^{\oplus r} & \mathbf{b}_0^{\oplus r} & \mathbf{O} & \mathbf{O} & \dots & \mathbf{O} \\ \mathbf{O} & \mathbf{B}^{\oplus r-1} & \mathbf{b}_0^{\oplus r-1} & \mathbf{O} & \dots & \mathbf{O} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \dots & \mathbf{B}^{\oplus r-q+1} \end{bmatrix},$$

where the (non-square) matrices  $\mathbf{b}_0^{\oplus k}$  consist of such transition rates between the states that are related to absorption of one of the components of the chain (i.e. absorption of one of the  $k$  iid r.v.'s with distribution  $PH(\boldsymbol{\beta}, \mathbf{B})$  each),

$$(20) \quad \mathbf{b}_0^{\oplus k} = \mathbf{b}_0 \otimes \mathbf{I}_{m^{k-1}} + \mathbf{I}_m \otimes \mathbf{b}_0 \otimes \mathbf{I}_{m^{k-2}} + \dots + \mathbf{I}_{m^{k-1}} \otimes \mathbf{b}_0,$$

the matrix  $\mathbf{I}_j$  is the identity matrix of size  $j \geq 1$ ,  $\otimes$  is the Kronecker product and  $\oplus$  is the Kronecker sum of matrices (repeated sums and products of this type are given as upper indices), whereas  $\mathbf{b}_0 = -\mathbf{B}\mathbf{1}$  is the absorption rate vector corresponding to the distribution  $PH(\boldsymbol{\beta}, \mathbf{B})$ . The initial distribution of such a chain may be given as

$$(21) \quad \boldsymbol{\beta}_{q:r} = (\boldsymbol{\beta}^{\otimes r}, \mathbf{0}),$$

This means that the phase-type distribution class is closed with respect to the order statistics, and hence the stability condition (1) and the cost  $EC_{r,q}$ , defined in (3), may be obtained in explicit form. Indeed, it is known that for the r.v.  $X$  having  $PH(\hat{\boldsymbol{\beta}}, \hat{\mathbf{B}})$  distribution, the first and second

order moments are given as [22]

$$(22) \quad EX = -\hat{\beta}\hat{B}^{-1}\mathbf{1}, \quad EX^2 = 2\hat{\beta}\hat{B}^{-2}\mathbf{1}.$$

Hence, it follows from (22) that

$$(23) \quad EC_{r,q} = rES_{q:r} = -r\beta_{q:r}\mathbf{B}_{q:r}^{-1}\mathbf{1},$$

where  $\beta_{q:r}$  and  $\mathbf{B}_{q:r}$  are defined in (21) and (19), respectively.

Consider the following important special case, let the workunit service time be exponentially distributed with service rate  $\mu$ . Then  $\beta = 1$ ,  $\mathbf{B} = -\mu$ ,  $\mathbf{b}_0 = \mu$  and (23) may be transformed into

$$(24) \quad EC_{r,q} = \frac{1}{\mu} \sum_{i=0}^{q-1} \frac{r}{r-i}.$$

In particular, if  $q = 1$ ,

$$(25) \quad P\{S_{1:r} \leq x\} = 1 - e^{-r\mu},$$

and hence (24) converts into

$$(26) \quad EC_{r,1} = \mu^{-1},$$

i.e. *the computation cost is independent on  $r$  when  $q = 1$* . In general,

$$(27) \quad EC_{r+1,q} - EC_{r,q} = -\frac{1}{\mu} \left( \sum_{i=1}^{q-1} \frac{1}{(r-i+1)(r-i)} \right) \leq 0.$$

Hence the cost monotonically decreases with  $r$  for given fixed  $q \geq 1$ .

If the arrival process is Poisson (or the so-called Markovian arrival process), then the *sojourn time* in the system may be obtained explicitly, however, in practice it is possible only numerically. However, in some important special cases this may be done in explicit form.

Consider the case when  $c = 1$ , i.e. the replication is performed on all the servers of the system, and let the arrival process be Poisson. Then the  $(1, r, q)$ -SM system is equivalent to the  $M/PH/1$  system which, due to (22), has the following form of the sojourn time (17):

$$(28) \quad EV_{r,q} = \frac{\lambda\beta_{q:n}\mathbf{B}_{q:n}^{-2}\mathbf{1}}{1 + \lambda\beta_{q:n}\mathbf{B}_{q:n}^{-1}\mathbf{1}} - \beta_{q:r}\mathbf{B}_{q:r}^{-1}\mathbf{1}.$$

If the workunit service time is exponentially distributed with rate  $\mu$  and if  $q = 1$ , then the analysis is simplified by using (25). In particular, for the  $(c, r, 1)$ -SM, the sojourn time may be computed by using the Erlang formula [23],

$$(29) \quad EV_{r,1} = \frac{1}{r\mu} + \frac{C(c, \lambda/(r\mu))}{cr\mu - \lambda},$$

where the so-called Erlang C-formula  $C(n, \rho)$  describes the probability of enqueueing in the M/M/n system having load  $\rho < n$ ,

$$(30) \quad C(n, \rho) = \left[ 1 + (1 - \rho) \frac{n!}{(n\rho)^n} \sum_{k=0}^{n-1} \frac{(n\rho)^k}{k!} \right]^{-1}.$$

In particular, the sojourn time in  $(c, 1, 1)$ -SM system coincides with the classical sojourn time in the  $M/M/c$  system:

$$(31) \quad EV_{1,1} = \frac{1}{\mu} + \frac{C(c, \lambda/\mu)}{c\mu - \lambda}.$$

It follows from (30) that  $C(1, \rho) = \rho$ , and hence the sojourn time (29) in the  $(1, r, 1)$ -SM system may be obtained as

$$EV_{r,1} = \frac{1}{r\mu} + \frac{\lambda}{r\mu(r\mu - \lambda)}.$$

### 2.3. Heavy-tailed service time distribution

It follows from (9) that the dependence of the cost  $C_{r,q}$  on  $r, q$  is not monotone in general, and needs to be studied for particular classes of distributions. In this section we consider the dependence of the average cost  $EC_{r,q}$  on the aforementioned parameters for the Pareto and Weibull service time distributions which (the latter - for specific parameter set) belong to the class of the heavy-tailed distributions. Note that the heavy-tailed distributions are widely used for studying the computing and communications systems [24].

Let the generic service time of the workunit  $S$  have the Pareto distribution (18) with parameter  $\alpha > 1$ . The properties of order statistics

from Pareto distribution are obtained in [25, 26], specifically,

$$(32) \quad E(S_{q;r}^k) = x_0^k \frac{r!}{(r-q)!} \frac{\Gamma(r-q+1-k/\alpha)}{\Gamma(r+1-k/\alpha)}, \quad 0 < k < \alpha(r-q+1),$$

where  $\Gamma$  is the gamma function. It follows from (3) and (32) that the explicit expression for  $EC_{r,q}$  may be obtained, and the following recurrence relation for  $EC_{r,q}$  may be obtained as well [6]:

$$(33) \quad EC_{r+1,q} = \frac{(r+1)^2(r-q+1-1/\alpha)}{r(r-q+1)(r+1-1/\alpha)} EC_{r,q}.$$

It follows from (33), after some algebra, that  $EC_{r,q}$  *increases* with  $r$ , if the following inequality holds good,

$$(34) \quad \alpha r^2 + (2\alpha - \alpha q - q - 1)r - \alpha q + \alpha - 1 > 0,$$

otherwise  $EC_{r,q}$  *non-increases* with  $r$ . For  $\alpha > 1$  and fixed  $q \geq 1$ , the quadratic polynomial (w.r.t.  $r$ ) on the l.h.s. of (34) has two real roots, such that the smaller is negative and larger is positive and has the following form [6]:

$$(35) \quad r_0(q) = \frac{1 + q + \alpha q - 2\alpha + \sqrt{(\alpha q + q - 1)^2 + 4q}}{2\alpha}.$$

As such, for given fixed  $q$  the inequality (34) holds good for (integer)  $r > r_0(q)$ , and the minimal cost is attained at one of the integers nearest to  $r_0(q)$ .

It follows from (35), in particular, that  $r_0(1) < 1$  for  $\alpha > 1.5$ , i.e. the cost  $EC_{r,1}$  increases with  $r$  for all  $r \geq 1$ . Indeed,

$$(36) \quad EC_{r,1} = x_0 \frac{r^2 \alpha}{r\alpha - 1} = x_0 \left( r + \frac{1}{\alpha} + \frac{1}{\alpha(\alpha r - 1)} \right), \quad r > 1/\alpha.$$

Moreover, it follows from (36) that the cost is asymptotically linear

$$\lim_{r \rightarrow \infty} r^{-1} EC_{r,1} = x_0.$$

It can be seen from (36) that  $EC_{1,1}$  may be arbitrary large if  $\alpha$  is close to 1. Indeed, it can be derived from (32) or from (36) that

$$EC_{1,1} = ES = x_0 \frac{\alpha}{\alpha - 1}.$$

At the same time, it follows from (36) that

$$EC_{2,1} < x_0 \left( 2 + \frac{2}{\alpha} \right),$$

since  $\alpha r - 1 > 1$  for  $r \geq 2$ . This means that, in particular,  $EC_{2,1} < EC_{1,1}$  for  $\alpha < \sqrt{2}$ . Thus, the replication may be used to significantly reduce the computation cost if  $r < r_0(q)$ .

With respect to the *sojourn time*, the general result (15) on the monotonicity w.r.t. replication/quorum parameters holds good, and, by taking into account (17) and (32), explicit expressions can be derived for the system of type  $M/G/1$ . However, an even more important problem to consider (in the presence of heavy tailed distributions) is to guarantee the finiteness of the moments (say, expectation or variance) of the sojourn time. This is especially important for obtaining reliable (interval) performance estimates in steady state and is important for the system stability.

Denote by  $\kappa(X)$  the *moment index* of the r.v.  $X$ , i.e.. [27, 28]

$$(37) \quad \kappa(X) = \sup\{k > 0 : E(X^k) < \infty\} \leq \infty.$$

Then it follows from (18) that the r.v.  $S$  with Pareto distribution has the moment index equal to  $\alpha$ ,

$$(38) \quad \kappa(S) = \alpha,$$

at the same time it follows from (32) that

$$(39) \quad \kappa(S_{q;r}) = \alpha(r - q + 1) \geq \alpha.$$

It follows from (39) that  $\kappa(S_{q;r})$  is *linear* w.r.t.  $r - q$ , i.e. the given moment conditions (say, variance) may be configured by appropriate choice of the redundancy parameters  $q, r$  even in the case when the moment index of the workunits is relatively small (e.g. has infinite variance).

It is known that the moment index in the multiserver system  $G/G/c$  may be obtained from the moment index of the service time [29]. In the system under consideration, for given fixed  $\lambda, \alpha, c, r, q$ , the moment index of the waiting time  $W_{r,q}$  may be obtained as follows [6]

$$(40) \quad \kappa(W_{r,q}) = (\alpha(r - q + 1) - 1)(c - [\rho]) \geq \alpha - 1,$$

where  $\rho = \lambda ES_{q:r}$ , and  $ES_{q:r}$  is given in (32). Thus from (3), using the Minkowski inequality [30], we obtain

$$(E(W_{r,q} + S_{q:r})^k)^{1/k} \leq (EW_{r,q}^k)^{1/k} + (ES_{q:r}^k)^{1/k},$$

and hence

$$(41) \quad \kappa(V_{r,q}) \geq \min(\kappa(W_{r,q}), \kappa(S_{q:r})),$$

where the moment indices  $\kappa(W_{r,q})$  and  $\kappa(S_{q:r})$  are given in (40) and (39), respectively.

Note that equality in (40) holds good, in particular, for  $c = 1$  and  $r = q$ , which agrees with the Pollacek–Khinchin formula (16), which states that finiteness of the *second moment* of the service time is necessary to ensure the finiteness of the average waiting time. Finally we note that, taking into account (41) and (40), the linear growth of the moment index  $S_{q:r}$  in  $r - q$  can be used to select the values of  $r, q$  in such way to guarantee the required value of the moment index (for instance, finiteness of the variance) of the task sojourn time even if the moment index  $\kappa(S) = \alpha$  of the workunit is small enough (for example, when  $\alpha$  is close to one).

Let's repeat the analysis for a system in which the typical workunit service time  $S$  has the Weibull distribution

$$(42) \quad F(x) = 1 - e^{-x^\xi}, \quad \xi > 0.$$

Then the moments of order  $k > 0$  for  $S_{q:r}$  can be obtained in the following form [31, 32]

$$(43) \quad E(S_{q:r}^k) = q \binom{r}{q} \Gamma(1 + k/\xi) \sum_{j=0}^{q-1} (-1)^j \binom{q-1}{j} (r - q + 1 + j)^{-1 - k/\xi}.$$

Thus, these moments are finite for any  $k > 0$ , i.e.,  $\kappa(S_{q:r}) = \infty$ , while the expression (17) allows to find an explicit form for  $EV_{r,q}$  for a system of type  $M/G/1$ . In addition, the following recurrence holds [17, 33]

$$(44) \quad E(S_{q:r}) = E(S_{q-1:r-1}) + \binom{r-1}{q-1} J(r - q + 1, q - 1),$$

where  $ES_{1:1} = J(1, 0) = \Gamma(1 + k/\xi)$ , and  $J(p, m)$  is defined recursively:

$$J(p, m) = J(p, m - 1) + J(p + 1, m - 1), \quad J(p, 0) = p^{-1/\xi} \Gamma(1 + 1/\xi).$$

Taking account of (43), we have

$$(45) \quad EC_{r,q} = r \frac{r!}{(r-q)!} \frac{\Gamma(\xi^{-1})}{\xi} \sum_{j=0}^{q-1} (-1)^j \frac{(r-q+1+j)^{-1-\xi^{-1}}}{j!(q-1-j)!}.$$

It can be seen from (45) that

$$(46) \quad EC_{r,1} = \Gamma(1 + 1/\xi)r^{1-1/\xi}, \quad r \geq 1.$$

This means, in particular, that  $EC_{r,1}$  is non-decreasing (non-increasing) in  $r$  for  $\xi \geq 1$  ( $\xi \leq 1$ , respectively). Moreover, for  $\xi = 1$  the Weibull distribution coincides with the exponential distribution, and at the same time (46) is consistent with (26).

It is difficult to use (45) directly to establish the monotonicity of  $EC_{r,q}$ . However, considering the difference  $\delta(r) = EC_{2,r+1} - EC_{2,r}$  and using recursion (44), we obtain

$$\delta(r) = \Gamma(1 + 1/\xi) \left( (2r^2 + r + 1)r^{-1/\xi} - r(r+1)^{1-1/\xi} - r^2(r-1)^{-1/\xi} \right).$$

In particular, for  $\xi < 1$  the quantity  $EC_{2,r}$  does not increase in  $r$ . Indeed, let us study the function

$$\psi(r) = (2r^2 + r + 1)r^{-1/\xi} - r(r+1)^{1-1/\xi} - r^2(r-1)^{-1/\xi}.$$

Denote  $a = \xi^{-1} > 1$ . Let's transform the function to the following form

$$(47) \quad \psi(r) = r^2(2r^{-a} - (r+1)^{-a} - (r-1)^{-a}) + \phi(r),$$

where

$$\phi(r) = r(r+1)(r^{-a-1} - (r+1)^{-a-1}) = r^{1-a} + r^{-a} - (r+1)^{1-a} + (r+1)^{-a}.$$

The convexity of the function  $x^{-a}$  implies that  $2r^{-a} - (r+1)^{-a} - (r-1)^{-a} \leq 0$ . Moreover, one can verify that  $\partial\phi(r)/\partial r \leq 0$  for all  $a > 1$ , i.e. the function  $\phi(r)$  is decreasing in  $r \geq 2$ . Moreover, one can check that  $\psi(2) \leq 0$  for any  $\xi \in (0, 1)$ . Thus, it follows from (47) that the difference  $\delta(r)$  is nonpositive for all  $r \geq 2$ , i.e.,  $EC_{2,r}$  is monotonically non-increasing in  $r$  for any  $\xi \in (0, 1)$ .

TABLE 1. A summary of the obtained results.

Distribution	Characteristic	
	$C_{r,q}$	$V_{r,q}$
In general	Non-monotonic	Non-decreasing, inequality (15)
Exponential	Non-increasing, inequality (27); is constant for $q = 1$ , formula (26); explicit form (24)	Explicit form (29) in system $M/M/c$ for $q = 1$
Phase type	Explicit form (23)	Explicit form (28) in $M/PH/1$
Pareto	Global minimum, expression (35)	Explicit form in system $M/G/1$ with respect (17) and (32)
Weibull with parameter $\xi$	Not-increasing if $q = 1, 2$ and $\xi \leq 1$ ; non-decreasing for $q = 1, \xi \geq 1$ , expression (46)	Explicit form in a system like $M/G/1$ taking account of (17) and (43)

### 3. Resume and perspectives of the research

Table 1 presents the results of studying the impact of replication parameter  $r$  and quorum  $q$  on the cost  $C_{r,q}$  and the sojourn time  $V_{r,q}$  in multi-server models with  $(c, r, q) - SM$  redundancy for different distributions of workunit service time  $S$ . The type of monotonicity is indicated with an increase in  $r, q$ , in some cases the result is available for the average value or for the corresponding random variable (stochastic order). The results are obtained for cases where the service time has a phase-type distribution, Pareto or Weibull distribution (belonging to the class of heavy-tailed distributions).

Taking account of the obtained results, the following directions of their further development can be distinguished:

- the monotonicity of the average cost (27) was obtained for the special case of exponentially distributed workunit service times, while the general case of an arbitrary phase type of distributions is of interest;

- it would be interesting to consider also a heterogeneous case, for example, when the servers to which replication is carried out have different performance (in this case, the formula (19) and further results require appropriate adjustment);
- cost monotonicity for a system with Weibull distribution is shown in section 2.3 for  $r = 1, 2$ , the case of arbitrary  $r$  requires a separate study;
- finally, the model assumes that the number of server blocks  $c$  is unchanged, while  $r$  is a variable parameter, it is advisable to study the case when the number of blocks is defined as  $s/r$  (integer), where  $s$  is (unchanged) number of servers in source system.

The latter direction opens up opportunities for using stochastic comparisons for systems with a different number of servers, but the parametric monotonicity of systems is also interesting (for example, in the parameter  $\alpha$  for the Pareto distribution or  $\xi$  for the Weibull distribution).

## References

- [1] I. Chernov, N. Nikitina, E. Ivashko. “Task scheduling in desktop grids: open problems”, *Open Engineering*, **7**:1 (2017), pp. 343–357.  [↑76](#)
- [2] N. Verma, D. Singh. “Data redundancy implications in wireless sensor networks”, *Procedia Computer Science*, **132** (2018), pp. 1210–1217.  [↑76](#)
- [3] A. Rumyantsev, E. Ivashko, I. Chernov, D. Kositsyn, A. Shabaev, V. Ponomarev. “Latency/wearout in a flash-based storage system with replication on write”, 2019 24th Conference of Open Innovations Association (FRUCT) (08–12 April 2019, Moscow, Russia), pp. 360–366.  [↑76](#)
- [4] I. Chernov, E. Ivashko, D. Kositsyn, V. Ponomarev, A. Rumyantsev, A. Shabaev. “Flash-based storage deduplication techniques: A survey”, *International Journal Of Embedded And Real-Time Communication Systems (IJERTCS)*, **10**:3 (2019), pp. 32–48.  [↑76](#)
- [5] O. Osipov, E. Rogachko. “Fork-join queueing systems with heterogeneous servers threshold control policy”, SMARTY2020 (August 16–20, 2020, Petrozavodsk, Russia), CEUR Workshop Proc., vol. **2792**, 2020, pp. 76–88.  [↑76](#)
- [6] A. Rumyantsev, S. Chakravarthy, E. Morozov, S. Remnev. “Cost and effect of replication and quorum in desktop grid computing”, *Information Technologies And Mathematical Modelling. Queueing Theory And Applications*, Springer CCIS Series, vol. **912**, Springer, Cham, 2018, ISBN 978-3-319-97594-8, pp. 143–156.  [↑76, 86, 87](#)

- [7] A. Thomasian. “Analysis of fork/join and related queueing systems”, *ACM Computing Surveys*, **47**:2 (2014), id. 17, 71 pp.  [↑77, 78](#)
- [8] F. Baccelli, A. Makowski, A. Shwartz. “The fork-join queue and related systems with synchronization constraints: Stochastic ordering and computable bounds”, *Advances in Applied Probability*, **21**:3 (1989), pp. 629–660.  [↑77](#)
- [9] K. Gardner, S. Zbarsky, S. Doroudi, M. Harchol-Balter, E. Hyttiä, A. Scheller-Wolf. “Queueing with redundant requests: Exact analysis”, *Queueing Systems*, **83**:3–4 (2016), pp. 227–259.  [↑77](#)
- [10] C. Kim, A. K. Agrawala. “Analysis of the fork-join queue”, *IEEE Transactions on Computers*, **38**:2 (1989), pp. 250–255.  [↑77](#)
- [11] R. Nelson, A. N. Tantawi. “Approximate analysis of fork/join synchronization in parallel queues”, *IEEE Transactions on Computers*, **37**:6 (1988), pp. 739–743.  [↑77](#)
- [12] G. Joshi, E. Soljanin, G. Wornell. “Queues with redundancy: Latency-cost analysis”, *ACM SIGMETRICS Performance Evaluation Review*, **43**:2 (2015), pp. 54–56.  [↑77, 78](#)
- [13] G. Joshi. *Efficient redundancy techniques to reduce delay in cloud systems*, PhD Thesis, Massachusetts Institute of Technology, 2016. [↑77, 78](#)
- [14] G. Joshi, E. Soljanin, G. Wornell. “Efficient redundancy techniques for latency reduction in cloud systems”, *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, **2**:2 (2017), id. 12, 30 pp.  [↑78](#)
- [15] S. Asmussen. *Applied Probability and Queues*, SMAP, vol. **51**, Springer, 2003, ISBN 978-0-387-00211-8, 452 pp.  [↑79](#)
- [16] N. Balakrishnan. “Permanents, order statistics, outliers and robustness”, *Revista Matemática Complutense*, **20**:1 (2007), pp. 7–107.  [↑79](#)
- [17] N. Balakrishnan, P. C. Joshi. “A note on order statistics from Weibull distribution”, *Scandinavian Actuarial Journal*, **1981**:2 (1981), pp. 121–122.  [↑80, 88](#)
- [18] W. Whitt. “Comparing counting processes and queues”, *Advances in Applied Probability*, **13**:1 (1981), pp. 207–220.  [↑81](#)
- [19] J. Blanchet, Y. Pei, K. Sigman. “Exact sampling for some multi-dimensional queueing models with renewal input”, *Advances in Applied Probability*, **51**:4 (2019), pp. 1179–1208.  [↑82](#)
- [20] M. Bladt, B. Nielsen. *Matrix-Exponential Distributions in Applied Probability*, PTSM, vol. **81**, Springer US, Boston, MA, 2017, ISBN 978-1-4939-7047-6, 749 pp.  [↑83](#)
- [21] A. Rumyantsev, S. Chakravarthy. “Matrix-exponential distributions in applied probability”, *Proceedings of the Third International Conference BOINC-based High Performance Computing: Fundamental Research and Development*, BOINC:FAST 2017 (August 28–September 01, 2017, Petrozavodsk, Russia), CEUR Workshop Proc., vol. **1973**, 2017, pp. 27–34.  [↑83](#)

- [22] P. Buchholz, J. Krieger, I. Felko. *Input Modeling with Phase-Type Distributions and Markov Models. Theory and Applications*, SpringerBriefs in Mathematics, Springer International Publishing, Cham, 2014, ISBN 978-3-319-06673-8, xii+127 pp.  [↑84](#)
- [23] M. Harchol-Balter. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*, Cambridge University Press, Cambridge, 2013, ISBN 9781107027503, 574 pp. [↑85](#)
- [24] D. Fietelson. *Workload Modeling for Computer Systems Performance Evaluation*, Cambridge University Press, Cambridge, 2015, ISBN 9781139939690.  [↑85](#)
- [25] H. J. Malik. “Exact moments of order statistics from the Pareto distribution”, *Scandinavian Actuarial Journal*, **1966**:3–4 (1966), pp. 144–157.  [↑86](#)
- [26] J. S. Huang. “A note on order statistics from Pareto distribution”, *Scandinavian Actuarial Journal*, **1975**:3 (1975), pp. 187–190.  [↑86](#)
- [27] D. J. Daley, C. M. Goldie. “The moment index of minima (II)”, *Statistics & Probability Letters*, **76**:8 (2006), pp. 831–837.  [↑87](#)
- [28] D. J. Daley. “The moment index of minima”, *Journal of Applied Probability*, **38**, Issue A: Probability, Statistics and Seismology (2001), pp. 33–36.  [↑87](#)
- [29] A. Scheller-Wolf, R. Vesilo. “Sink or swim together: necessary and sufficient conditions for finite moments of workload components in FIFO multiserver queues”, *Queueing Systems*, **67**:1 (2011), pp. 47–61.  [↑87](#)
- [30] G. H. Hardy, J. E. Littlewood, G. Pólya. *Inequalities*, 2nd ed., Cambridge University Press, Cambridge, 1991, 340 pp. [↑88](#)
- [31] J. Lieblein. “On moments of order statistics from the Weibull distribution”, *Ann. Math. Statist.*, **26**:2 (1955), pp. 330–333.  [↑88](#)
- [32] K. S. Sultan, M. E. Moshref. “Moments of order statistics from Weibull distribution in the presence of multiple outliers”, *Communications in Statistics – Theory and Methods*, **43**:10–12 (2014), pp. 2214–2226.  [↑88](#)
- [33] H. Rinne. *The Weibull Distribution. A Handbook*, Chapman and Hall/CRC, New York, 2009, ISBN 9780367577469, 808 pp. [↑88](#)

Received	06.12.2022;
approved after reviewing	27.01.2023;
accepted for publication	05.02.2023;
published online	15.02.2023.

Recommended by

*prof. A. M. Tsirlin*

**Information about the authors:****Alexander Sergeevich Rumyantsev**

senior researcher, PhD, key interests: queueing theory, multiserver systems, stochastic modeling and analysis, high-performance and distributed computing, matrix-analytic method, regenerative approach, three-level modeling

 0000-0003-2364-5939

*e-mail:* [ar0@krc.karelia.ru](mailto:ar0@krc.karelia.ru)

**Diana Sergeevna Dolgaleva**

intern-researcher, key interests: queueing theory, matrix-analytic method, bioinformatics, gene networks, distributed computing

 0000-0002-0606-6583

*e-mail:* [abcdi\\_do@mail.ru](mailto:abcdi_do@mail.ru)

**Alexander Stanislavovich Golovin**

leading engineer-researcher, PhD student, key interests: queueing theory, matrix-analytic method, energy efficiency, simulation, distributed computing, multiserver queues

 0000-0003-1325-3739

*e-mail:* [golovin@krc.karelia.ru](mailto:golovin@krc.karelia.ru)

*The authors declare no conflicts of interests.*

УДК 81'322+61

 10.25209/2079-3316-2023-14-1-95-123

## Система извлечения упоминаний симптомов из текстов на естественном языке с помощью нейронных сетей

Юрий Петрович Сердюк<sup>1</sup>, Наталья Александровна Власова<sup>2✉</sup>,  
Седа Рубеновна Момот<sup>3</sup>

Институт программных систем им. А. К. Айламазяна РАН, Вельсково, Россия

<sup>2✉</sup> [nathalie.vlassova@gmail.com](mailto:nathalie.vlassova@gmail.com)

**Аннотация.** В статье представлена система для извлечения упоминаний симптомов из медицинских текстов на естественном (русском) языке. Система осуществляет нахождение симптомов в тексте, их нормализацию (приведение к стандартной форме) и отождествление – отнесение найденного симптома к группе однотипных симптомов. Каждый этап обработки реализуется с помощью отдельной нейронной сети. Состав извлекаемых симптомов ограничен тремя видами заболеваний – аллергические и пульмонологические заболевания, а также коронавирусная инфекция (COVID-19). Представлен и описан аннотированный корпус предложений, использованный для обучения нейросети нахождению упоминаний симптомов, относящихся к этим трем заболеваниям. При разметке корпуса был использован простой XML-подобный язык. Для представления предложений, непосредственно поступающих на вход нейросети, предложен расширенный ВЮ-формат разметки. Для каждого этапа приведены оценки точности (для первого этапа точность оценивалась при строгом и гибком тестировании). Описаны подходы и реализация приведения к стандартной форме и отождествления упоминаний симптомов. Даны сравнения с аналогичными работами по извлечению симптомов из медицинских текстов на разных языках, а также показано место данной системы в системах поддержки принятия клинических решений.

**Ключевые слова и фразы:** автоматическая обработка языка, нейронные сети, автоматическое извлечение информации, аннотированный корпус, упоминания симптомов, BERT-модели, Covid-19

**Для цитирования:** Сердюк Ю.П., Власова Н.А., Момот С.Р. Система извлечения упоминаний симптомов из текстов на естественном языке с помощью нейронных сетей // Программные системы: теория и приложения. 2023. Т. 14. № 1(56). С. 95–123. [https://psta.psir.ru/read/psta2023\\_1\\_95-123.pdf](https://psta.psir.ru/read/psta2023_1_95-123.pdf)

## Введение

Медицинские системы поддержки принятия решений являются, с одной стороны, важной компонентой современного подхода к оказанию врачебной помощи, а с другой – высокотехнологичной областью, впитывающей в себя последние разработки в области искусственного интеллекта (ИИ) [1, 2]. Эффективность их применения во многом зависит от состава и качества информации, которая доступна врачам при постановке диагноза и проведении лечения. В свою очередь, эта информация распределена между ее источниками в структурированной форме – базах данных, и в неструктурированной – в виде разнообразных медицинских документов на естественном языке (ЕЯ), в состав которых входят истории болезней, анамнезы, эпикризы, отчеты о проведении клинических мероприятий (анализов, операций, обследований). Соответственно, извлечение и структурирование необходимых сведений из текстов на ЕЯ является критически важной задачей, от качества решения которой во многом зависит эффективность работы всей системы поддержки принятия клинических решений.

Из медицинских ЕЯ-текстов в общем случае требуется извлечь сущности разных типов – названия заболеваний и нарушений функционирования человеческого организма, симптомов болезней, медицинских процедур, лекарственных препаратов, упоминания частей тела и другие. Извлечение каждой такой сущности из текста на ЕЯ является достаточно сложной задачей, в состав которой включается не только сам этап извлечения, но и этапы нормализации (приведения к стандартной форме) и отнесения извлеченной сущности к предопределенному словарю (тезаурусу) или категории.

В данной работе мы предлагаем комплексный подход к извлечению упоминаний симптомов болезней из текстов на ЕЯ, включающий в себя все три отмеченных выше этапа – само извлечение, нормализацию и отождествление. В общем случае возможны два подхода к извлечению симптомов:

- (1) извлечение упоминаний симптомов и
- (2) извлечение (названий) симптомов с их значениями.

В первом случае под *упоминанием симптома* понимается некоторая непрерывная (без разрывов и пропусков) часть текста, которая используется врачами-практиками в качестве устойчивого словосочетания для описания некоторого симптома:

Мальчик поступил в отделение с жалобами на **затруднение носового дыхания**.

Во втором, более общем и сложном случае, выражение, представляющее собой описание симптома, рассматривается в структурированном виде, а именно в виде пары *⟨название симптома, значение симптома⟩*:

*Температура тела поднимается до 38°C, либо частота дыхательных движений доходит до 21-22 в минуту.*

Часто, как в приведенном выше примере, название симптома и его значение могут находиться в предложении на некотором «расстоянии» друг от друга, т.е. не составлять непрерывное выражение.

Соответственно, в первом случае при аннотировании упоминаний симптомов в тексте достаточно отметить начало и конец размечаемого выражения, что делает задачу извлечения упоминания симптомов схожей с задачей извлечения именованных сущностей (NER – named entity recognition), таких как «*Высшая школа экономики*» или «*Национальный Совет молодёжных и детских объединений России*».

Во втором случае требуется применение «двумерного» аннотирования – разметки по отдельности названий симптомов и их значений вместе с установлением отношений между ними. При таком подходе задача извлечения симптомов с их значениями становится схожей с задачей извлечения событий с их аргументами [3] или задачей извлечения семантических отношений [4].

В данной работе мы представляем решение первой задачи – извлечения упоминаний симптомов. Состав извлекаемых симптомов ограничен в нашем случае тремя видами заболеваний:

- аллергические заболевания,
- пульмонологические заболевания,
- коронавирусная инфекция (COVID-19).

Основу корпуса предложений с упоминаниями симптомов аллергических и пульмонологических заболеваний составил *корпус клинических текстов* <sup>(URL)</sup>, подготовленный Институтом системного анализа ФИЦ ИУ РАН и ФГНБНУ «Научный центр здоровья детей» (НЦЗД) в 2015 г. Корпус предложений с упоминаниями симптомов коронавирусной инфекции был подготовлен нами самостоятельно на основе медицинских документов и текстов, имеющихся в Интернете.

Для аннотирования упоминаний симптомов мы применяем простой XML-подобный язык. При подаче непосредственно на вход нейросети он переводится в расширенный BIO-формат (BIO – beginning, inside, outside) по сравнению с его стандартным вариантом, применяемым в NER-задачах. Для более формальной трактовки понятий «симптом» и

«упоминание симптома» нами были введены понятия «симптом-отклонение» и «симптом-характеристика». Кроме того, мы проводим различия между «простым симптомом» и «перечислением симптомов». Более подробные сведения о корпусе предложений для извлечения упоминаний симптомов, принципах и правилах их аннотирования приведены в разделе 1.

Для практического решения задачи извлечения упоминаний симптомов как задачи распознавания именованных сущностей мы применили систему *DeepPavlov*<sup>URL</sup> – библиотеку для анализа текстов на ЕЯ и создания виртуальных помощников (чат-ботов), разработанную в Лаборатории нейронных сетей и глубокого обучения МФТИ. Данная библиотека содержит несколько типов моделей для решения NER-задач, начиная от стандартных моделей на основе рекуррентных нейронных сетей, а также BERT-моделей и заканчивая гибридными моделями, объединяющими в себе несколько нейронных сетей различного типа. Нами была выбрана гибридная модель в силу ее небольшого размера и достаточно хороших показателей точности извлечения [5]. В частности, был разработан и использован собственный формат тегов разметки упоминаний симптомов – расширенный BIO-формат. Подробности практической реализации этого подхода даются в разделах 1.3 и 1.4.

После извлечения выражения, являющегося упоминанием симптома, оно приводится в нормальную форму. Так, например, из предложения

*Ребенок поступил в отделение впервые с жалобами на эпизоды  
малопродуктивного кашля*

должно быть извлечено выражение *малопродуктивного кашля* с дальнейшим переводом его в нормальную форму *малопродуктивный кашель*.

До недавнего времени для решения этой задачи применялись методы на основе морфо-синтаксического анализа, в которых каждое слово выражения преобразовывалось в корректную форму с учетом синтаксических связей этого слова с другими словами выражения. С точки зрения реализации эти методы основывались на использовании множества правил, описывающих соответствующие преобразования. Данный подход обладает тем недостатком, что для качественного выполнения нормализации требуется очень большое количество правил, составление которых является трудоемкой задачей. Кроме того, увеличение количества правил неизбежно приводит к появлению противоречий между ними.

В последнее время для решения задач нормализации выражений, например, именованных сущностей и им подобных, начали применяться методы на основе нейронных сетей [6]. В этом случае нейронная сеть обучается на предложениях с выделенными выражениями, для которых

подготовлены их стандартные представления. В частности, для более полного учета семантических свойств отдельных слов используются BERT-модели. Для решения данной задачи в настоящей работе мы воспользовались системой, разработанной Д. Анастасьевым [7]. Структура этой системы, состав и особенности корпуса для нормализации выражений, а также детали практической реализации и полученные результаты изложены в разделе 2.

Задачей нормализации выражений проблема извлечения упоминаний симптомов не исчерпывается. Конкретный симптом как семантическое понятие может иметь много форм своего выражения. Например, выражения *затруднение дыхания*, *затрудненное дыхание*, *нарушение функции дыхания*, *приступы затруднения дыхания*, *дыхательная недостаточность 2 степени*, *одышка* обозначают схожие явления. Привязывание каждого из такого типа выражений к некоторому стандартизованному (эталонному) выражению иногда называют задачей нормализации понятий (concept normalization) или связыванием сущностей (entity linking).

В некоторых случаях в качестве ресурса эталонных названий тех или иных медицинских сущностей (заболеваний, симптомов, лекарств, процедур и др.) используются разного вида тезаурусы и онтологии. Одним из главных примеров здесь является онтология UMLS (Unified Medical Language System) [9]. Составной частью онтологии UMLS является тезаурус предметных медицинских рубрик (Medical Subject Headings) [10], который имеет эквивалент на русском языке MSHRUS, поддерживаемый Центральной Научной Медицинской Библиотекой (ЦНМБ).

С другой стороны, в последнее время компанией Соцмедика<sup>URL</sup> разрабатывается объединенная база медицинских знаний UMKB (Unified Medical Knowledge Base) [11], составной частью которой является классификатор медицинских терминов.

Тем не менее имеются большие проблемы в совместном использовании различных онтологий и тезаурусов на русском языке. Поэтому в нашем случае мы разработали собственную систему базовых (эталонных) терминов для упоминаний симптомов аллергических и пульмонологических заболеваний, а также для наиболее важных симптомов коронавирусной инфекции. Механизм привязывания извлеченных упоминаний симптомов к выбранным эталонным терминам изложен в разделе 3.

В разделе 4 дан обзор современных работ по извлечению упоминаний симптомов, а также симптомов в виде пар *⟨название симптома, значение симптома⟩* и проведено их сравнение с нашей работой.

Раздел 5 посвящен кратким выводам и направлениям дальнейшей работы.

## 1. Извлечение упоминаний симптомов

### 1.1. Принципы и правила аннотации упоминаний симптомов

#### 1.1.1. Общие принципы выделения симптомов

Одно из классических определений понятия «симптом» описывает его как «признак болезни, качественно новый, не свойственный здоровому организму феномен» (Большая медицинская энциклопедия, 1989 г., 3-е изд.). Однако в реальной медицинской практике понятие «симптом» трактуется более широко – от описаний всевозможных отклонений (*сухой кашель*) до названий болезней, которые могут являться симптомами других болезней (*ринит, конъюнктивит*), выражений, включающих числовые данные (*частота дыхательных движений 21-22 в минуту*), характеристик отдельных органов и их функционирования (*сердечные тоны ясные*) и др.

Ввиду принципиальной невозможности строгим формальным образом определить понятие симптома (и, соответственно, его упоминания), в качестве основного принципа аннотирования предлагается выделение из текста в качестве упоминаний симптомов таких выражений, которые

- (1) используются врачами-практиками в качестве устойчивых словосочетаний,
- (2) обозначают либо «симптом-отклонение», либо «симптом-характеристику».

Далее в тексте статьи для легкости восприятия мы часто употребляем термин «симптом» в значении «упоминание симптома».

«*Симптомы-отклонения*» – симптомы, само название которых говорит о некотором отклонении от нормы (*малопродуктивный кашель, хрипы, дизурия, менингеальные знаки*):

- (а) жалобы на ухудшение общего состояния: *нарушение сна, повышенная утомляемость, снижение работоспособности*;
- (б) симптомы, непосредственно наблюдаемые или ощущаемые самим пациентом: *сухой кашель, одышка, боль в пояснице, высыпания*;
- (в) нарушения, фиксируемые врачом при осмотре/обследовании: *хрипы в легких, менингеальные знаки, нарушения бронхиальной проходимости*.

«*Симптомы-характеристики*» – это выражения, в которых присутствует указание на некоторую часть тела или физиологический процесс с соответствующими им характеристиками (*живот мягкий, тоны сердца ритмичные*). Необходимо отметить, что общепринятая врачебная практика состоит в использовании таких выражений в качестве упоминаний симптомов независимо от того, обозначают ли они некоторое отклонение от нормы (отрицательная характеристика), или же нормальное функционирование организма или его подсистемы (положительная характеристика):

- (1) отклонение от нормы (отрицательная характеристика): *носовое дыхание несколько затруднено, воронкообразная деформация грудной клетки;*
- (2) нормальное функционирование органа или системы (положительная характеристика): *кожные покровы бледно-розовые, зев спокоен, сердечные тоны ясные.*

Упоминания симптомов обоих типов могут сопровождаться указанием на:

- (1) характер проявления симптома (*малопродуктивный кашель, схваткообразные боли*);
- (2) интенсивность проявления симптома (*умеренные боли*);
- (3) часть тела, к которой относится симптом (*боли в суставах*);
- (4) временные рамки и условия проявления (*ночное недержание мочи, аллергический ринит в весенне-летний период, одышка при физической нагрузке*).

#### 1.1.2. Правила аннотирования симптомов

Для аннотирования упоминаний симптомов был использован простой XML-подобный язык разметки:

*Мальчик поступил в отделение с жалобами на <symp> затруднение носового дыхания </symp> в весенний и осенний периоды.*

Упоминания симптомов в тексте делятся на два вида:

- (а) простые (единичные) симптомы,
- (б) перечисления симптомов.

Простые симптомы выделяются с помощью тегов <symp> и </symp>:

*У ребенка 2 лет на фоне ОРВИ внезапно ночью возник <symp> лающий кашель </symp>, <symp> охриплость </symp>.*

Перечисление симптомов – это выражения с упоминаниями нескольких симптомов, которые должны быть извлечены по отдельности. Так, из выражения *Сердечные тоны ясные, ритмичные* должны быть выделены простые симптомы

- *Сердечные тоны ясные,*
- *Сердечные тоны ритмичные*

Соответственно, главная цель разметки таких выражений – отметить составные части отдельных симптомов.

Перечисления симптомов обычно состоят из основной части и подчиненной части. При этом возможны два варианта:

- (1) одна основная и несколько подчиненных, и
- (2) несколько основных частей и одна подчиненная.

(Случаи, когда выражение состоит из нескольких основных и нескольких подчиненных частей, возможны, но очень редки). В типичном случае основная часть содержит названия частей тела или физиологических процессов, а подчиненная – некоторые характеристики этих частей или процессов (эти характеристики мы называем аргументами).

**Вариант 1.** Перечисление симптомов с одной основной и несколькими подчиненными частями заключается в теги `<symp-args>` и `</symp-args>`, внутри которых отдельные аргументы отмечаются тегами `<arg>` и `</arg>`:

```
<symp-args> Сердечные тоны <arg> ясные </arg>, <arg> ритмичные </arg>
</symp-args>.
```

```
<symp-args> Живот <arg> мягкий </arg>, <arg> при пальпации безболезненный
</arg> </symp-args>.
```

**Вариант 2.** Перечисление симптомов с несколькими основными частями и одной подчиненной заключается в теги `<symp-args>` и `</symp-args>`, внутри которых отдельные основные части отмечаются тегами `<symp>` и `</symp>`, а подчиненная – тегами `<arg>` и `</arg>`.

```
<symp-args> <symp> Стул </symp>, <symp> диурез </symp> <arg> не нарушены
</arg> </symp-args>.
```

```
У пациентов возникают <symp-args> <symp> сухость </symp> и <symp>
неприятные ощущения </symp> <arg> в носоглотке </arg> </symp-args>.
```

### 1.1.3. Специальные случаи

Как уже было отмечено выше, в реальной медицинской практике понятие симптома трактуется очень широко. В данном разделе рассматриваются несколько сложных специальных случаев упоминания симптомов и правила их разметки, общепринятые у врачей-аннотаторов.

#### 1. Заболевание как симптом.

Встречаются следующие случаи использования названия заболеваний в качестве симптомов:

- (а) заболевание как симптом других заболеваний:

```
В последнее время все больше внимания привлекает <symp>
конъюнктивит </symp> как новый симптом коронавируса.
```

- (б) название заболевания, которому предшествует слово *симптомы*:

*Мальчик поступает в клинику впервые с жалобами на <symp> симптомы бронхиальной обструкции </symp>.*

- (6) название заболевания, которому предшествуют слова, указывающие на эпизодический характер его проявления, например, *эпизоды, проявления, случаи, приступы:*

*В течение нескольких лет отмечались <symp> проявления бронхиальной астмы </symp>.*

*<symp> Эпизоды бронхиальной обструкции </symp> отмечались 3-4 раза в год.*

С другой стороны, если аналогичные слова стоят перед упоминанием симптома-отклонения, то они не включаются в размечаемое выражение:

*эпизоды <symp> малопродуктивного кашля </symp>.*

- (2) заболевания, сопровождающееся указанием на регулярные рецидивы: *частые, рецидивирующие:*

*В дальнейшем <symp> частые ОРИ </symp>, <symp> рецидивирующие аденоидиты </symp>.*

К этой же группе относится и упоминание симптома вида *частые смены настроения*, иногда встречающееся в медицинских документах. С другой стороны, не отмечаются в качестве упоминаний симптомов выражения вида *часто болеющие дети, часто болеющий ребенок* и т. п., поскольку они самостоятельно классифицируются как заболевания в МКБ-10 (D84.9 «Иммунодефицит неуточненный»).

## 2. Симптомы с уточняющими характеристиками.

В медицинских текстах симптомы часто сопровождаются дополнительными характеристиками, такими как интенсивность их проявления, локализация, условия и временные особенности проявления. Все или некоторые такие характеристики могут быть включены в размечаемое выражение, если они составляют вместе с симптомом непрерывный фрагмент текста. Так, например, допускаются следующие варианты разметки одного и того же выражения:

*С января 2030 года беспокоят <symp> высыпания </symp> на коже ягодиц.*

*С января 2030 года беспокоят <symp> высыпания на коже </symp> ягодиц.*

*С января 2030 года беспокоят <symp> высыпания на коже ягодиц </symp>.*

Такой подход к разметке мы называем «гибким подходом», и он отображает реальную практику разметки аналогичных выраже-

ний врачами-аннотаторами. Приведем несколько дополнительных примеров разметки симптомов с уточняющими характеристиками:

*Девочка 8 лет, почувствовала <symp> боль в горле при глотании </symp>.*

*И больной астмой грудной ребенок, и его сверстник без признаков аллергии могут дать <symp> эпизод обструкции на фоне ОРВИ </symp>.*

## 1.2. Корпус предложений для обучения и тестирования

Для обучения и тестирования нейронной сети использовался корпус предложений из медицинских текстов объемом около 1,1 тыс предложений (неповторяющихся). Тематика текстов-источников – аллергические, пульмонологические заболевания и COVID-19. Все предложения аннотированы вручную с помощью специально разработанной системы тегов.

### 1.2.1. Корпус предложений с симптомами аллергических и пульмонологических заболеваний

В качестве основы корпуса предложений с симптомами аллергических и пульмонологических заболеваний использован *корпус медицинских текстов*<sup>URL</sup>, подготовленный ИСА РАН и Научным центром здоровья детей (НЦЗД) [4]. Этот корпус содержит анонимизированные медицинские истории более 60 пациентов НЦЗД (всего 112 текстов). Тексты корпуса включают в себя выписные эпикризы из историй болезней, заключения различного вида обследований, а также назначения и рекомендации различных врачей. Для использования в научных целях данный корпус доступен по запросу.

Корпус размечен практикующими врачами и включает аннотации таких сущностей, как заболевание (Disease), симптом (Symptom), тяжесть заболевания (Severity), течение болезни (Course), назначения (Treatment), медикаменты (Drug), части тела (Body location). Для отдельных видов сущностей установлены отношения между ними.

В силу сложности выделения перечисленных сущностей и отношений между ними разметка симптомов в данном корпусе иногда является непоследовательной и противоречивой. В частности, бывает трудно разбить выражение на сам симптом и место (часть тела) его проявления или выделить в сложном выражении отдельные симптомы. Так, например, в качестве симптома (Symptom) в данном корпусе встречаются такие выражения, как *мягкий; безболезненный; живот мягкий, безболезненный; боли; головные боли; боли в животе и головные боли; боли и потеря чувствительности.*

В нашем случае из этого корпуса было взято около 500 неповторяющихся предложений. Для некоторых симптомов, которые были представлены недостаточным образом в ранее отобранных предложениях,

было добавлено еще около 150 предложений из медицинских текстов (научные статьи, истории болезни, ситуативные задачи из учебников медвузов и т. п.).

### 1.1.2. Корпус предложений с симптомами коронавирусной инфекции

Вторая часть корпуса для обучения извлечению упоминаний симптомов подготовлена нами самостоятельно и состоит из множества предложений, относящихся к новой коронавирусной инфекции. Общее количество предложений в этом подкорпусе – около 500. В каждом предложении вручную размечены симптомы COVID-19 согласно разработанным принципам и правилам аннотации упоминаний симптомов (см. раздел 1.1.1).

Корпус создан на основе представленных в Интернете медицинских текстов, в которых содержатся описания симптомов и проявлений новой коронавирусной инфекции, например, таких как [13]. В некоторых работах [18, 19], упоминается от 10 до 60 разнотипных симптомов данной инфекции, которые извлекались из текстов. Нами было выделено 15 наиболее важных симптомов, характеризующих данное заболевание. В их состав вошли:

- |                    |                        |                        |
|--------------------|------------------------|------------------------|
| (1) температура;   | (7) потеря обоняния и  | (11) насморк;          |
| (2) кашель;        | вкуса;                 | (12) затруднение дыха- |
| (3) боль в груди;  | (8) диарея;            | ния;                   |
| (4) одышка;        | (9) головная боль;     | (13) кожные высыпания; |
| (5) слабость;      | (10) учащенное сердце- | (14) рвота;            |
| (6) боль в мышцах; | биение;                | (15) поражения глаз.   |

Для каждого из 15 симптомов было подобрано и размечено не менее, чем по 30 предложений с различными формами упоминания этих симптомов. Так, например, для симптома «кашель» были размечены такие выражения, как *кашель; частый кашель с небольшим количеством мокроты; малопродуктивный кашель; кашель с мокротой; постоянное покашливание; легкое покашливание; непрерывный кашель; кашель становится сухим и стойким; сухой, часто надсадный кашель; сухой ковидный кашель* и т. п.

Упоминания симптомов коронавирусной инфекции являются более разнообразными, а по своей структуре – более сложными, чем упоминания симптомов аллергических и пульмонологических заболеваний. В частности, они могут включать в себя выражения с числовыми данными:

*Большая 36 лет, госпитализирована с жалобами на <symp> повышение температуры до 39,8° С </symp>, <symp> слабость </symp>, <symp> утомляемость </symp>, <symp> кашель с трудноотделяемой мокротой </symp>, <symp> чувство нехватки воздуха </symp>.*

### 1.3. Расширенный ВГО-формат разметки упоминаний симптомов

Прежде чем текст с XML-подобной разметкой будет подан на вход нейронной сети для обучения, он переводится в так называемый «расширенный ВГО-формат». Расширение обычного ВГО-формата необходимо для представления перечисления симптомов (см. раздел 1.1.2). Для представления упоминаний простых симптомов используется стандартная ВГО-разметка:

```
У ребенка <symp> проявления atopического дерматита </symp> .
O      O                      B-SYM      I-SYM      I-SYM      O
```

Здесь метки **B-SYM** и **I-SYM** являются аналогами меток **B** и **I** стандартного ВГО-формата.

Таким образом, в процессе обучения на вход нейронной сети последовательно подаются отдельные слова (а точнее их векторные представления – эмбединги, см. раздел 3) с одновременным предъявлением ей правильных выходов – меток ВГО-формата.

Для представления перечислений симптомов используются дополнительные виды меток. Так, например, выражения с аргументами представляются следующим образом:

```
<symp-args> Живот <arg> мягкий </arg> , <arg> безболезненный </arg>
              B-SA          B-ARG      O          B-ARG
</symp-args>
              E-SA
```

Здесь метки **B-SA** (*begin-symptom with arguments*) и **E-SA** (*end-symptom with arguments*) обрамляют всё выражение с перечислением симптомов, а метки **B-ARG** (*begin-argument*), а также **I-ARG** (*inside-argument*) отмечают отдельные аргументы.

Аналогично выражение с перечислением симптомов второго типа переводится в расширенный ВГО-формат следующим образом:

```
<symptom-list> <symp> Стул </symp> , <symp> диурез </symp> <arg> не нарушены
              B-SL      O          B-SL          B-ARG I-ARG
</arg> </symptom-list>
              O
```

Здесь метки **B-SL** отмечают начало отдельного симптома в списке симптомов. Отдельный симптом может состоять из нескольких слов, а потому для разметки продолжения симптома используются метки **I-SYM**.

Соответственно, после обучения на этапе работы нейросети она пытается разметить входное предложение с помощью меток расширенного ВГО-формата. Используя эти метки, система из представления

```
Живот мягкий , безболезненный .
B-SA B-ARG O B-ARG E-SA
```

восстановит перечисление симптомов в виде *живот мягкий, живот безболезненный*.

Аналогично по расширенной BIO-разметке

*Стул , диурез не нарушены .*  
*B-SL O B-SL B-ARG I-ARG O*

будет восстановлено перечисление симптомов *стул не нарушен, диурез не нарушен*.

При этом будет выполнена попытка требуемого в общем случае синтаксического согласования отдельных частей выражения. Непосредственно выделенное выражение *стул не нарушены* будет переведено в выражение *стул не нарушен*.

Если нейронная сеть не формирует для некоторого предложения корректную BIO-разметку, то считается, что система не нашла в этом предложении упоминаний симптомов.

Машинная реализация расширенного BIO-формата базируется на возможности использования произвольных меток для разметки именованных сущностей в системе *DeepPavlov*<sup>URL</sup>.

#### 1.4. Практическая реализация и результаты

Для практического решения задачи извлечения упоминаний симптомов была использована система *DeepPavlov*<sup>URL</sup> – библиотека и набор моделей для анализа ЕЯ-текстов. В частности, данная система содержит подсистему для распознавания именованных сущностей, которая в прямом виде может быть применена для извлечения упоминаний симптомов. В этой подсистеме имеется несколько типов моделей для решения NER-задач. Нами выбрана гибридная модель, объединяющая в себе несколько нейронных сетей различного типа, в силу ее небольшого объема и достаточно хороших показателей точности распознавания [5]. (В последних версиях системы *DeepPavlov*<sup>URL</sup> предлагается только один тип моделей – BERT-модели).

Данная гибридная модель состоит из

- (1) сверточной (convolutional) нейронной сети, ответственной за распознавание морфологической структуры слов, в частности, за распознавание префиксов и суффиксов;
- (2) двунаправленной LSTM-сети (bidirectional long-short term memory – bi-LSTM) – специального вида рекуррентной сети, ответственной за распознавание слов, написанных с заглавной буквы (что является спецификой общей задачи распознавания именованных сущностей, но иногда применимо и для распознавания симптомов типа «отеки Квинке»);

- (3) еще одной bi-LSTM-сети, ответственной за распознавание правого и левого контекстов слова;
- (4) распознавателя на базе CRF-слоя (conditional random field) (вместо заключительного стандартного softmax-слоя), ответственного за более точное распознавание порядка следования тегов разметки.

Для числового векторного представления слов использовалась предобученная модель, построенная на основе корпуса новостных текстов с русскоязычного сайта *lenta.ru*. Длина вектора составляла 100 чисел. Обучающее множество содержало около 1000 предложений. Размер словаря различных словоформ упоминаний симптомов в этом множестве также составил около 1000 выражений. Тестовое множество состояло из 90 предложений со 160 упоминаниями симптомов. Это же множество использовалось как валидационное на этапе обучения нейронной сети.

Было проведено 2 вида тестирования:

- (1) «строгое», когда нейросеть должна была в точности выделить заданный список симптомов;
- (2) «гибкое» (см. раздел 1.1.3 про гибкий подход при разметке симптомов с уточнениями), когда возможны несколько вариантов выделения упоминания симптома из выражения, и выделение одного из них засчитывается за правильный ответ.

Точность распознавания упоминаний симптомов при строгом тестировании составила 81–82%, а при гибком – 85–86%.

## 2. Нормализация выражений

После этапа извлечения выражений, представляющих собой упоминания симптомов, выполняется нормализация этих выражений. Необходимость ее вызвана тем, что в общем случае извлеченные упоминания симптомов не находятся в нормализованной форме: *малопродуктивным кашлем; одышку, возникшую после стресса; обструктивного ларинготрахеита; хрипов* и т. п.

Способ решения задачи нормализации такого вида выражений с помощью нейронных сетей состоит в том, что на этапе обучения нейросети на вход подаются предложения с выделенными выражениями одновременно с примерами стандартных представлений этих выражений. Таким образом, приведенные выше выражения должны быть в итоге переведены нейронной сетью в выражения *малопродуктивный кашель; одышка, возникшая после стресса; обструктивный ларинготрахеит; хрипы*.

Формально нормализация выражения состоит в переводе главного слова выражения (корня синтаксической структуры) в именительный

падеж с соответствующим согласованием синтаксических связей данного слова с другими словами выражения. Следует отметить, что задача нормализации медицинских выражений отличается от классической задачи лемматизации – перевода слова еще и в единственное число, так как в медицинских документах некоторые выражения стандартно используются только во множественном числе: *хрипы*, *менингеальные знаки* и др. Таким образом, для нормализации такого рода выражений необходимо иметь в обучающем множестве соответствующие примеры с ними.

Возможны два подхода к решению задачи нормализации выражений, представляющих собой упоминания симптомов, с помощью нейронных сетей:

- (1) обучение нейронной сети на специальном датасете всевозможных форм упоминаний симптомов (из заданной предметной области) с соответствующими им стандартными представлениями;
- (2) обучение нейронной сети на датасете с выражениями общего вида с дальнейшим применением ее к нормализации выражений, представляющих симптомы.

При первом подходе требуется подготовка собственного, достаточно большого обучающего множества предложений с упоминаниями симптомов из заданной предметной области, что является довольно трудоемкой задачей. Нами выбран второй подход, поскольку уже существуют аннотированные русскоязычные корпуса, предназначенные для обучения нейронных сетей нормализации выражений общего вида, а также несколько практических реализаций систем, решающих эту задачу.

В нашем случае мы взяли систему, разработанную Д. Анастасьевым [7], которая, в свою очередь, базируется на морфо-синтаксическом анализаторе (а точнее, лемматизаторе из него) из работы [8]. Данная система показала один из наиболее высоких результатов в рамках соревнования RuNormAS-2021 «A shared task on Russian normalization of annotated spans», состоявшегося в рамках конференции «Диалог-2021» [20]. В качестве обучающего множества мы взяли часть корпуса, предоставленного на этом соревновании. Общее количество выражений (как общего вида, так и именованных сущностей) составило около 20 000. Для учета специфики некоторых медицинских терминов из области аллергических и пульмонологических заболеваний, а также коронавирусной инфекции, нами было подготовлено и добавлено в это множество еще около 100 предложений.

Базовой частью системы Д. Анастасьева является лемматизатор, который порождает все возможные леммы данного слова. Эти леммы, дополненные некоторой другой информацией, нейронная сеть учится классифицировать, предлагая уже на этапе практической работы ту или

иную лемму в качестве модификации заданного слова для нормализации всего выражения. В качестве общей модели в данной системе использован так называемый «Dependency Parser» [21] из библиотеки AllenNLP (<https://allennlp.org/allennlp>). Данный парсер предсказывает наличие дуги соответствующего типа между двумя словами в дереве зависимостей при синтаксическом анализе предложения. В системе Д. Анастасьева этот парсер адаптирован для предсказания связи между данной словоформой и одной из ее лемм. В качестве BERT-модели, порождающей векторные представления слов, в этой системе была использована модель RuBERT.

Сама система состоит из двух основных частей:

- (1) BERT-эмбеддера, который строит векторные представления слов входного предложения, и
- (2) LSTM-сети с классификатором, которые предсказывают необходимые модификации слов для нормализации всего выражения.

На вход LSTM-сети с классификатором подается комбинированное представление слов входного предложения, которое содержит

- (1) векторное представление самого слова, полученного от BERT-эмбеддера,
- (2) указатели на все возможные леммы данного слова из словаря,
- (3) информацию о положении данного слова внутри отмеченного выражения (т.е. одну из меток B, I, O из BIO-формата)

и некоторую другую информацию.

Данная система, примененная к нормализации выражений общего вида, показала результаты в 97–98% точности нормализации. Наши эксперименты по нормализации выражений, представляющих собой упоминания симптомов в выбранных предметных областях, дали результат 98,5%.

### 3. Отождествление выражений

После этапа нормализации выражений, представляющих собой упоминания симптомов, выполняется этап отождествления – отнесения найденного симптома к группе однотипных симптомов. Например, в медицинских анамнезах часто встречаются следующие семантически схожие понятия: *высыпания на коже, кожные высыпания, мелко-папулезные высыпания, эритематозные пятна, лихенизация, эритематозные бляшки.*

Отнесение подобного вида выражений к одной группе с выбором некоторого стандартного (эталонного) выражения в качестве представителя всей группы называют задачей нормализации понятий (concept normalization) или связыванием сущностей (entity linking). В области медицинской информатики уже делались попытки создания словарей

(тезаурусов) эталонных названий заболеваний, симптомов, лекарств и т.д., которые бы служили классификаторами медицинских терминов. Наиболее известной и разработанной системой является онтология UMLS (Unified Medical Language System) [9]. Однако русскоязычный эквивалент ее терминологической части – тезаурус Medical Subject Headings – имеет объем только 2% от английского варианта.

В UMLS основным понятием является так называемый «уникальный идентификатор понятия» (CUI – concept unique identifier). Такого рода идентификаторы выбраны большей частью для названий болезней, частей тела, лекарств и в меньшей степени для симптомов, в частности потому, что «симптомы» является трудно формализуемым понятием. Кроме того, что такие классификации неполны, они обладают и другими недостатками. В них часто отсутствуют выражения, которые мы называем «симптомами-характеристиками» (см. раздел 1.1.1). Например, в UMLS отсутствуют симптомы *тоны сердца ясные*, *тоны сердца ритмичные*, а имеется только понятие *тоны сердца*.

Введенная нами классификация на «симптомы-отклонения» и «симптомы-характеристики» вместе с реализованным механизмом отождествления схожих выражений позволяет в данном случае автоматически отнести выражения *тоны сердца ритмичные*, *сердечные тоны звучные*, *сердечные тоны ясные* к группе симптомов-характеристик с эталонным представителем *тоны сердца ясные*, а выражения *аритмия*, *учащенное разбиение*, *тахикардия*, *перебои в работе сердца* к группе симптомов-отклонений с эталонным представителем *сердцебиение*.

Таким образом, мы разработали собственную систему базовых (эталонных) терминов для упоминаний симптомов в выбранной нами предметной области (аллергические и пульмонологические заболевания), а также для наиболее важных симптомов коронавирусной инфекции. Также все упоминания симптомов из нашего корпуса предложений были вручную расклассифицированы по эталонным группам. Например, в группу *Зев(отклонения)* были отнесены выражения *катаральные явления*, *зев умеренно гиперемирован*, *зев рыхлый*, *зев: задняя стенка разрыхлена*, *зуд в зеве* и др., а в группу *зев спокоен* были отнесены выражения *зев спокоен*, *зев без катаральных явлений*, *миндалины не увеличены*.

Общее количество групп составило более 60. Так, например, в их состав вошли группы *Затруднение дыхания*, *Затруднение носового дыхания*, *Бронхообструктивный синдром*, *Дыхание в легких (отклонения)*, *Кашель*, *Кожные процессы*. В качестве базового метода отнесения выделенного выражения к некоторой группе (упоминаний) симптомов была использована возможность построения векторного представления этого выражения с помощью BERT-модели. Метод основывается на том, что если подать

на вход нейронной сети (BERT-модели) заданное выражение, которое заключено между специальными токенами **CLS** и **SEP**:

*CLS зев умеренно гиперемирован SEP,*

то кроме векторных представлений отдельных слов выражения нейросеть после последовательного чтения каждого из слов перейдет в заключительное состояние, числовое представление которого считается векторным представлением всего выражения и которое будет помещено в позицию токена **CLS**.

В практической реализации, как и на этапе нормализации выражений, в качестве BERT-модели нами была использована универсальная модель для русского языка RuBERT. Предварительные ее испытания показали, что она хорошо отождествляет семантически близкие термины, в том числе из медицинской области:

<i>затрудненное дыхание, одышка</i>	0.973
<i>экзема, сыпь</i>	0.968

Приведенные числа являются значениями (косинусной) меры близости соответствующих выражений.

Сам алгоритм отнесения заданного выражения к той или иной группе однотипных симптомов состоит из следующих основных шагов:

- (1) прямая проверка вхождения заданного выражения в ту или иную группу симптомов путем сопоставления текстовых представлений заданного выражения с каждым симптомом из группы; если такое вхождение найдено, то с данным выражением связывается название найденной группы;
- (2) если прямая проверка оказалась неудачной, то вычисляется векторное представление заданного выражения, после чего для каждой группы симптомов находится средневзвешенное значение близости данного выражения к каждому из симптомов в этой группе:

$$m = \frac{m_1 + m_2 + \dots + m_N}{N},$$

где  $m_i$  есть значение (косинусной) меры близости заданного выражения с  $i$ -ым симптомом из группы;  $N$  – количество симптомов в группе (векторные представления симптомов в группах вычисляются заранее); далее выбирается группа симптомов, для которой значение  $m$  оказалось максимальным, но не меньшим, чем 0,75;

- (3) если максимальное значение  $m$  оказалось меньше, чем 0,75, то исходное выражение считается неотнесенным к какой-либо группе и рассматривается как отдельная новая группа.

Проведенные эксперименты с упоминаниями симптомов из указанной

выше предметной области дали результаты в 92–93% по отнесению симптома к той или иной группе.

Следует, однако, отметить, что на модели RuBERT значения близости пар выражений *⟨одышка, тошнота⟩* и *⟨одышка, рвота⟩* оказались очень высокими: 0,96–0,97. Это объясняется тем, что модель RuBERT была построена на корпусе универсальных текстов, в которых данные пары слов чаще всего встречались в одинаковых контекстах. Следовательно, чтобы получить векторные представления медицинских терминов, которые более точно представляли бы их смысл, необходимо построение BERT-модели на достаточно большом корпусе специализированных медицинских текстов, в которых данные слова встречаются в более разнообразных контекстах.

#### 4. Обзор смежных работ

В данном разделе мы даем обзор наиболее важных работ, посвященных извлечению симптомов из неструктурированных (ЕЯ) медицинских текстов и их использованию в системах поддержки принятия диагностических решений. Для русского языка количество таких исследований невелико – все они перечислены ниже. Для них мы также проводим сравнение с нашей работой.

В работе [15] представлена система принятия диагностических решений на основе анамнезов пациентов, записанных в свободном неструктурированном виде. Система реализована в научном подразделении SberMedAI Сбербанка России и имеет название «Умный помощник врача «ТОР-3». По информации пресс-службы Сбербанка система внедрена во всех поликлиниках для взрослых г. Москвы. В качестве текста, поступавшего на вход нейросети, использовалась конкатенация содержимого разделов «симптомы» и «анамнез» из описания визита пациента к врачу. Общее количество использованных для обучения описаний визитов составило первоначально около 4 млн, а позднее было доведено до 14 млн. В качестве модели нейронной сети использовалась незначительно модифицированная стандартная BERT-модель. Модификация заключалась в добавлении в качестве последнего слоя нейронной сети полносвязного классификатора.

Классификатор выдает результат в виде одного из 265 кодов заболеваний, выбранных из Международной классификации болезней МКБ-10. Размер входных текстов был ограничен 128 словами (или, более точно, токенами).

Результат работы системы – выданный диагноз – оценивался врачом. Было использовано два основных вида оценки: по одному и трем выданным возможным диагнозам. В последнем случае, если хотя бы один из трех диагнозов, выданных нейросетью, совпадал с диагнозом врача, то такой

ответ засчитывался за правильный. Точность выдачи одного диагноза составила 47,5%, а по трем диагнозам – 68%.

Недостатком этого подхода является то, что такая система рассматривает текст «целиком», не выявляя в нем смысловые единицы, такие как заболевания, симптомы, названия диагностических и лечебных процедур, лекарства и т.д., а также отношения между ними. Поэтому данная система поставит «диагноз» по любому тексту, в том числе и по такому, в котором отсутствуют упоминания каких-либо симптомов, или вообще не являющемуся медицинским. Кроме того, низкая точность работы системы не позволяет рассматривать ее в качестве полезной компоненты систем поддержки принятия клинических решений.

В работе [16] подчеркивается необходимость извлечения из медицинских текстов отдельных сущностей, используемых в процессе постановки диагноза как врачом-практиком, так и в системах поддержки принятия клинических решений. В данной работе решаются задачи извлечения из текстов

- (1) упоминаний заболеваний,
- (2) названий лекарств.

Кроме того, для заболеваний предложены средства извлечения их атрибутов, таких как степень тяжести (*Severity*), этап течения заболевания (*Course*), часть тела (*Body location*) и некоторых других. Авторы работы отмечают отсутствие аннотированных корпусов клинических текстов на русском языке и предлагают собственный аннотированный корпус. Этот корпус кратко описан выше в разделе 1.2.1.

Напомним, что в данном корпусе размечаются не только сущности, но и отношения между ними, тогда как в нашей работе цель состояла в разработке системы извлечения только упоминаний симптомов, которые представляют собой непрерывные фрагменты текста, для чего и был подготовлен соответствующий корпус. Заявленная авторами работы [16] точность извлечения составила

- (1) для заболеваний – 95,1%,
- (2) для названий лекарств – 84,3%.

В работе [17] представлен подход к извлечению так называемых «смешанных упоминаний симптомов» (*mixture symptom mentions*) из текстов на китайском языке, посвященных традиционной китайской медицине. Под смешанным упоминанием симптомов понимаются выражения, в которых описывается в общем случае несколько симптомов с характеристиками их тяжести и указанием частей тела, к которым эти симптомы относятся. Задача извлечения для таких выражений состоит в выделении сущностей,

которые в них упоминаются, и отношений между ними. В качестве сущностей в данной работе рассматриваются симптом (Symptom), степень тяжести (Severity) и часть тела (Area of the body). Эти сущности могут находиться в отношениях `located_at` (Symptom, Area of the body) и `is_a_description_of` (Symptom, Severity).

Авторами работы предложена объединенная модель обучения (joint learning model), в которой нейронная сеть обучается одновременному извлечению как сущностей, так перечисленных выше отношений между ними. Обучающий корпус состоял из 2,2 тыс. текстов с 73 тыс. упоминаемых сущностей. В качестве модели нейронной сети была использована стандартная BERT-модель для китайского языка, дообученная на текстах по традиционной китайской медицине. Точность распознавания сущностей и отношений между ними составила 82,5%.

Если в нашей собственной системе мы извлекаем непрерывные фрагменты текста, являющиеся описанием одного или нескольких симптомов, то в работе [17] вначале извлекаются отдельные структурные части таких упоминаний, а затем из этих частей собираются описания отдельных симптомов. Наш механизм обработки перечисления симптомов (см. раздел 1.1.2) частично пересекается с подходом авторов данной работы.

В работе [18] заявлено создание алгоритма выявления подозрения на COVID-19 для использования в симптом-чекерах и системах поддержки принятия врачебных решений на основе информации, извлекаемой из текстов на русском языке. Исходными данными для алгоритма являются симптомы и их значения, извлекаемые из протоколов врачебных осмотров. Общее количество извлекаемых симптомов составило 14. Они были разделены на 2 группы:

- (1) *заложенность в груди, слабость, боль в мышцах, диарея, головная боль, кашель, конъюнктивит, сатурация, температура;*
- (2) *спутанность сознания, одышка, аносмия, поражения кожи, кровохарканье.*

Первая группа симптомов извлекалась, аналогично нашему подходу, с помощью методов распознавания именованных сущностей. Точность извлечения этих симптомов составила от 82,6% до 97,43% в зависимости от конкретного симптома. Вторая группа симптомов извлекалась с помощью специально сконструированных правил, в частности, включающих в себя поиск по синонимам этих симптомов – например, поиск симптома *аносмия* включал в себя поиск выражений *снижение обоняния, потеря обоняния*.

Отметим, что в нашей системе все такие формы упоминания симптомов извлекаются обученной нейросетью на первом этапе с последующим отождествлением. Для второй группы симптомов показатели точности

извлечения не приведены. Также в статье [18] не отмечено, как извлекались значения таких симптомов как *температура*, *частота дыхания*, *частота сердечных сокращений*, хотя эти значения используются в самом алгоритме предсказания наличия COVID-19. Обучающее множество состояло из 11 243 описаний первичных врачебных осмотров с установленными диагнозами ОРВИ и пневмония, т.е. данные документы не относились непосредственно к COVID-19. Размеченный корпус недоступен для исследовательских целей.

В работе [19] представлены

- (1) англоязычный «COVID-19 Annotated Clinical Text (CACT) Corpus»,
- (2) средства извлечения симптомов с их значениями и
- (3) система предсказания наличия COVID-19 на основе извлеченных симптомов.

Корпус текстов состоит из 1472 клинических описаний, содержащих около 30 тыс. упоминаний диагнозов, результатов тестов и симптомов. Данный корпус является первым аннотированным корпусом с информацией про COVID-19. В качестве симптомов в нем размечены не только симптомы как таковые (*cough*, *shortness of breath*), но и само слово *COVID*. Для слова *COVID* размечены его аргументы типа *Test Status* и *Assertion*, где, в свою очередь, аргумент типа *Assertion* может принимать значения *present*, *absent*, *possible*, *hypothetical*, *not patient*. Для самих симптомов размечены аргументы типа *Assertion*, *Change*, *Severity*, *Anatomy* и др. В силу сходства задачи извлечения симптомов со значениями с задачей извлечения событий с аргументами, что отмечено выше во Введении, в данной работе выделение симптомов реализовано как извлечение событий путем дообучения Bio+Clinical BERT-модели для решения этой задачи. Точность извлечения симптомов составила 81%, а их значений – от 45% до 78% в зависимости от конкретного значения.

Аналогично нашей работе в данной системе реализована нормализация выражений, описывающих симптомы, но этот этап отнесен уже к алгоритму предсказания наличия COVID-19. Нормализация проводилась с использованием подготовленной вручную таблицы, отображающей возможные выражения для каждого из симптомов в их нормализованные формы. Например, выражения *coughing*, *coughs*, *distress coughing*, *distressed coughing* отображались в слово *cough*. Стандартизованная терминология, например, из онтологии UMLS, в данной системе не использовалась.

В работе [12] представлен русскоязычный аннотированный корпус RuCCoN, имеющий в своей основе корпус, описанный выше в разделе 1.2.1 с дополнительной разметкой, которая состоит в приписывании каждой из 16 тыс. сущностей, отмеченных в исходном корпусе, одного из 2,4 тыс.

уникальных терминов (концептов) из русскоязычной части онтологии UMLS. Новый корпус и руководство по аннотированию размещены в открытом доступе. В корпусе RuCCoN выделено 10 сущностей (семантических типов):

- (1) заболевание или синдром,
- (2) часть тела, орган,
- (3) признак или симптом,

и другие, но симптомы составляют только 6% от всех семантических типов. Отличие данной работы от нашей состоит в том, что авторы работы [12] для привязывания словоформы выделенной сущности к эталонному термину используют отдельную нейронную сеть, обученную такому связыванию (*entity linking*). Мы же использовали возможность получения векторного представления (эмбединга) произвольных выражений с помощью некоторой BERT-модели и определения их близости с помощью стандартной косинусной меры. Для разного вида тестовых множеств авторы работы [12] сообщают о точности связывания с помощью нейросети в пределах 52–58%.

## 5. Выводы и направления дальнейшей работы

В данной статье была представлена система извлечения упоминаний симптомов из медицинских текстов на русском языке. Состав извлекаемых симптомов был ограничен аллергическими и пульмонологическими заболеваниями, а также коронавирусной инфекцией. Для обучения нейросети извлечению симптомов был подготовлен аннотированный корпус объемом в 1,1 тыс. предложений. Помимо извлечения упоминаний симптомов в системе предусмотрены их нормализация и отождествление.

Точность извлечения составила 81%, нормализации – 98%, отождествления – 92%. Для реализации последних двух операций была использована модель RuBERT, но опыт ее использования показал, что для дальнейшего повышения качества работы нейросетей необходимы BERT-модели, построенные на основе больших специализированных корпусов медицинских русскоязычных текстов. Дальнейшая работа будет заключаться в переходе к решению задачи извлечения информации о симптомах в виде пар *⟨название симптома, значение симптома⟩*, а также в расширении предметной области на симптомы кардиологических заболеваний.

Авторы статьи выражают благодарность ЦКП «Центр данных ДВО РАН»  [22] за предоставление вычислительных ресурсов на базе GPU.

## Список литературы

- [1] Sutton R. T., Pincock D., Baumgar D. C., Sadowski D. C., Fedorov R. N., Kroeker K. I. *An overview of clinical decision support systems: benefits, risks, and strategies for success* // npj Digit. Med.– 2020.– Vol. 6.– No. 3.– id. 17. doi ↑96
- [2] Kwan J. L., Lo L., Ferguson J., Goldberg H., Diaz-Martinez J. P., Tomlinson G., Grimshaw J. M., Shojania K. G. *Computerised clinical decision support systems and absolute improvements in care: meta-analysis of controlled clinical trials* // BMJ.– 2020.– Vol. 370.– id. m3216. doi ↑96
- [3] Sha L., Qian F., Chang B., Sui Zh. *Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction*, Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18) // Proceedings of the AAAI Conference on Artificial Intelligence.– 2018.– Vol. 32.– No. 1.– Pp. 5916–5923. doi ↑97
- [4] Smirnova A., Cudre-Mauroux Ph. *Relation extraction using distant supervision: A survey* // ACM Computing Surveys.– 2019.– Vol. 51.– No. 5.– id. 106.– 35 pp. doi ↑97, 104
- [5] Le Th. A., Burtsev M. S. *A deep neural network model for the task of named entity recognition* // International Journal of Machine Learning and Computing.– 2019.– Vol. 9.– No. 1.– Pp. 8–13. URL doi ↑98, 107
- [6] Ji Z., Wei Q., Xu H. *BERT-based ranking for biomedical entity normalization*, AMIA Jt Summits Transl Sci Proc.– 2020.– Pp. 269–277. doi arXiv:1908.03548 ↑98
- [7] Anastasyev D. G. *Annotated span normalization as a sequence labelling task*, Papers from the Annual International Conference “Dialogue” (2021), Computational Linguistics and Intellectual Technologies.– vol. 20.– 2021.– ISBN 978-5-7281-3032-1.– Pp. 8–15. doi ↑99, 109
- [8] Anastasyev D. G. *Exploring pretrained models for joint morpho-syntactic parsing of Russian*, Papers from the Annual International Conference “Dialogue” (2020), Computational Linguistics and Intellectual Technologies.– vol. 19.– 2020.– ISBN 978-5-7281-3032-1.– Pp. 1-12. doi ↑109
- [9] Bodenreider O. *The Unified Medical Language System (UMLS): Integrating biomedical terminology* // Nucleic Acids Res.– 2004.– Vol. 32, suppl. 1.– Pp. D267–D270. doi ↑99, 111
- [10] Coletti M. H., Bleich H. L. *Medical subject headings used to search the biomedical literature* // J. Am. Med. Inform. Assoc.– 2001.– Vol. 8.– No. 4.– Pp. 317–323; Erratum in: J. Am. Med. Inform. Assoc.– 2001.– Vol. 8.– No. 6.– Pp. 597. doi doi ↑99
- [11] Бледжянц Г. А., Исакова Ю. А., Осипов А. А. *Апробация и внедрение эффективного использования инструментов объединенной базы медицинских знаний системой дистанционного образования инновационных субъектов* // Человеческий капитал.– 2020.– № S12-1.– С. 199–205. \* ↑99
- [12] Nesterov A., Zubkova G., Miftahutdinov Z., Kokh V., Tutubalina E., Shelmanov A., Alekseev A., Avetisian M., Chertok A., Nikolenko S. *RuCCoN: Clinical concept normalization in Russian*, Findings of the Association for Computational Linguistics: ACL 2022 (Dublin, Ireland).– 2022.– Pp. 239–245. doi ↑116, 117

- [13] *Временные методические рекомендации Министерства здравоохранения Российской Федерации «Профилактика, диагностика и лечение новой коронавирусной инфекции (COVID-19)»,* Версия 14 (27.12.2021).– Министерство здравоохранения Российской Федерации.– 233 с. [URL](#) ↑<sup>105</sup>
- [14] *Краткое руководство по разметке тестового корпуса. Задача «Medicine light»,* Версия 1.6.– ИСА РАН и ИЦЗД.– 2014. [URL](#) ↑
- [15] Blinov P., Avetisian M., Kokh V., Umerenkov D., Tuzhilin A. *Predicting clinical diagnosis from patients electronic health records using BERT-based neural networks*, AIME 2020: Artificial Intelligence in Medicine, Lecture Notes in Computer Science.– vol. **12299**, eds. M. Michalowski, R. Moskovitch, Cham: Springer.– 2020.– ISBN 978-3-030-59136-6.– Pp. 111–121. [doi](#) ↑<sup>113</sup>
- [16] Shelmanov A. O., Smirnov I. V., Vishneva E. A. *Information extraction from clinical texts in Russian*, Papers from the Annual International Conference “Dialogue” (2015), Computational Linguistics and Intellectual Technologies.– vol. **14**.– 2015.– Pp. 560–572. [URL](#) ↑<sup>114</sup>
- [17] Sun Yu., Zhao Zh., Wang Zh., He H., Guo F., Luo Yu., Gao Q., Wei N., Liu J., Li G.-Zh., Li Z. *Leveraging a joint learning model to extract mixture symptom mentions from traditional Chinese medicine clinical notes* // BioMed Research International.– Vol. **2022**, Conference Issue: Big Data for Biomedical Research.– id. 2146236. [doi](#) ↑<sup>114, 115</sup>
- [18] Гаврилов Д. В., Кирилкина А. В., Серова Л. М. *Алгоритм формирования подозрения на новую коронавирусную инфекцию на основе анализа симптомов для использования в системах поддержки принятия врачебных решений* // Врач и информационные технологии.– 2020.– № 4.– С. 51–58. [doi](#) \* ↑<sup>105, 115, 116</sup>
- [19] Lybarger K., Ostendorf M., Thompson M., Yetisgen M. *Extracting COVID-19 diagnoses and symptoms from clinical text: A new annotated corpus and neural event extraction framework* // Journal of Biomedical Informatics.– 2021.– Vol. **117**.– id. 103761. [doi](#) ↑<sup>105, 116</sup>
- [20] Zolotukhin D., Smurov I. *RuNormAS-2021: A shared task on Russian normalization of annotated spans*, Papers from the Annual International Conference “Dialogue” (2021), Computational Linguistics and Intellectual Technologies.– vol. **20**.– 2021.– ISBN 978-5-7281-3032-1.– Pp. 1245–1250. [doi](#) ↑<sup>109</sup>
- [21] Dozat T., Manning C. D. *Deep biaffine attention for neural dependency parsing*.– 2017.– 8 pp. [doi](#) [arXiv](#) 1611.01734 ↑<sup>110</sup>
- [22] Сорокин А. А., Макогонов С. В., Королев С. П. *Информационная инфраструктура для коллективной работы ученых Дальнего Востока России* // Научно-техническая информация. Сер. 1: Организация и методика информационной работы.– 2017.– № 12.– С. 14–16. \* ↑<sup>117</sup>

Поступила в редакцию	26.12.2022;
одобрена после рецензирования	29.01.2023;
принята к публикации	29.01.2023;
опубликована онлайн	17.02.2023.

Рекомендовал к публикации

к.т.н. Я. И. Гулиев

**Информация об авторах:****Юрий Петрович Сердюк**

старший научный сотрудник Исследовательского центра искусственного интеллекта ИПС им. А. К. Айламазяна, научные интересы: параллельное программирование, формальные исчисления процессов, системы типов

 0000-0003-2916-2102

**e-mail:** [Yuri@serdyuk.botik.ru](mailto:Yuri@serdyuk.botik.ru)

**Наталья Александровна Власова**

младший научный сотрудник Исследовательского центра искусственного интеллекта ИПС им. А. К. Айламазяна, научные интересы: компьютерная лингвистика, автоматическая обработка естественного языка, корпусная лингвистика

 0000-0002-7843-6870

**e-mail:** [nathalie.vlassova@gmail.com](mailto:nathalie.vlassova@gmail.com)

**Седа Рубеновна Момот**

младший научный сотрудник Исследовательского центра искусственного интеллекта ИПС им. А. К. Айламазяна, научные интересы: компьютерная лингвистика, автоматическая обработка естественного языка

 0000-0002-6097-6545

**e-mail:** [morlot@mail.ru](mailto:morlot@mail.ru)

*Все авторы сделали эквивалентный вклад в подготовку публикации.  
Авторы заявляют об отсутствии конфликта интересов.*



# A system for extracting symptom mentions from texts by means of neural networks

Yuri Petrovich **Serdyuk**<sup>1</sup>, Natalia Aleksandrovna **Vlasova**<sup>2</sup>,  
Seda Rubenovna **Momot**<sup>3</sup>

Ailamazyan Program Systems Institute of RAS, Ves'kovo, Russia

<sup>2</sup>*nathalie.vlassova@gmail.com*

**Abstract.** This paper presents a system for extracting symptom mentions from medical texts in natural (Russian) language. The system finds symptom mentions in texts, brings them to a standard form and identifies the found symptom to a group of similar symptoms. For each stage of processing we use a separate neural network. We extract symptoms of three areas of diseases: allergic and pulmonological diseases, as well as coronavirus infection (COVID-19). We present and describe an annotated corpus of sentences that is used to train neural networks for extracting symptom mentions. These sentences were marked up with the help of a simple XML-like language. An extended BIO-markup format was proposed for the sentences directly received at the input of the neural network. We give the quality evaluation of the symptom extraction accuracy under strict and flexible testing. Possible approaches to normalization and identification of symptom mentions and their implementation are described. Our results are compared with those achieved in similar researches, thus we show the place of our system among clinical decision support systems. (*In Russian*).

**Key words and phrases:** natural language processing, neural networks, information extraction, symptom mentions, annotated corpus, BERT-models, Covid-19

2020 *Mathematics Subject Classification:* 68T07; 68T50

**For citation:** Yuri P. Serdyuk, Natalia A. Vlasova, Seda R. Momot. A system for extracting symptom mentions from texts by means of neural networks. Program Systems: Theory and Applications, 2023, **14**:1(56), pp. 95–123. (*In Russ.*). [https://psta.psiras.ru/read/psta2023\\_1\\_95-123.pdf](https://psta.psiras.ru/read/psta2023_1_95-123.pdf)

## References

- [1] R. T. Sutton, D. Pincock, D. C. Baumgar, D. C. Sadowski, R. N. Fedorak, K. I. Kroeker. “An overview of clinical decision support systems: benefits, risks, and strategies for success”, *npj Digit. Med.*, **6**:3 (2020), id. 17. 
- [2] J. L. Kwan, L. Lo, J. Ferguson, H. Goldberg, J. P. Diaz-Martinez, G. Tomlinson, J. M. Grimshaw, K. G. Shojania. “Computerised clinical decision support systems and absolute improvements in care: meta-analysis of controlled clinical trials”, *BMJ*, **370** (2020), id. m3216. 
- [3] L. Sha, F. Qian, B. Chang, Zh. Sui. “Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction”, Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), *Proceedings of the AAAI Conference on Artificial Intelligence*, **32**:1 (2018), pp. 5916–5923. 
- [4] A. Smirnova, Ph. Cudre-Mauroux. “Relation extraction using distant supervision: A survey”, *ACM Computing Surveys*, **51**:5 (2019), id. 106, 35 pp. 
- [5] Th. A. Le, M. S. Burtsev. “A deep neural network model for the task of named entity recognition”, *International Journal of Machine Learning and Computing*, **9**:1 (2019), pp. 8–13.  
- [6] Z. Ji, Q. Wei, H. Xu. “BERT-based ranking for biomedical entity normalization”, *AMIA Jt Summits Transl Sci Proc.*, 2020, pp. 269–277.   1908.03548
- [7] D. G. Anastasyev. “Annotated span normalization as a sequence labelling task”, Papers from the Annual International Conference “Dialogue” (2021), *Computational Linguistics and Intellectual Technologies*, vol. **20**, 2021, ISBN 978-5-7281-3032-1, pp. 8–15. 
- [8] D. G. Anastasyev. “Exploring pretrained models for joint morpho-syntactic parsing of Russian”, Papers from the Annual International Conference “Dialogue” (2020), *Computational Linguistics and Intellectual Technologies*, vol. **19**, 2020, ISBN 978-5-7281-3032-1, pp. 1-12. 
- [9] O. Bodenreider. “The Unified Medical Language System (UMLS): Integrating biomedical terminology”, *Nucleic Acids Res.*, **32**, suppl. 1 (2004), pp. D267–D270. 
- [10] M. H. Coletti, H. L. Bleich. “Medical subject headings used to search the biomedical literature”, *J. Am. Med. Inform. Assoc.*, **8**:4 (2001), pp. 317–323; Erratum in: *J. Am. Med. Inform. Assoc.*, **8**:6 (2001), pp. 597.  
- [11] G. A. Bledzhyancz, Yu. A. Isakova, A. A. Osipov. “Approbation and implementation of the effective use of the tools of the integrated medical knowledge base by the system of distance education of innovative disciplines”, *Chelovecheskij kapital*, 2020, no. S12-1, pp. 199–205 (in Russian).
- [12] A. Nesterov, G. Zubkova, Z. Miftahutdinov, V. Kokh, E. Tutubalina, A. Shelmanov, A. Alekseev, M. Avetisian, A. Chertok, S. Nikolenko. “RuCCoN: Clinical concept normalization in Russian”, Findings of the Association for Computational Linguistics: ACL 2022 (Dublin, Ireland), 2022, pp. 239–245. 

- [13] *Vremennye metodicheskie rekomendacii Ministerstva zdravooxraneniya Rossijskoj Federacii «Profilaktika, diagnostika i lechenie novoj koronavirusnoj infekcii (COVID-19)»*, Versiya 14 (27.12.2021), Ministerstvo zdravooxraneniya Rossijskoj Federacii, 233 pp. [URL](#)
- [14] *Kratkoe rukovodstvo po razmetke testovogo korpusa. Zadacha «Medicine light»*, Versiya 1.6, ISA RAN i NCzZD, 2014. [URL](#)
- [15] P. Blinov, M. Avetisian, V. Kokh, D. Umerenkov, A. Tuzhilin. “Predicting clinical diagnosis from patients electronic health records using BERT-based neural networks”, AIME 2020: Artificial Intelligence in Medicine, Lecture Notes in Computer Science, vol. **12299**, eds. M. Michalowski, R. Moskovitch, Springer, Cham, 2020, ISBN 978-3-030-59136-6, pp. 111–121. [doi](#)
- [16] A. O. Shelmanov, I. V. Smirnov, E. A. Vishneva. “Information extraction from clinical texts in Russian”, Papers from the Annual International Conference “Dialogue” (2015), Computational Linguistics and Intellectual Technologies, vol. **14**, 2015, pp. 560–572. [URL](#)
- [17] Yu. Sun, Zh. Zhao, Zh. Wang, H. He, F. Guo, Yu. Luo, Q. Gao, N. Wei, J. Liu, G. -Zh. Li, Z. Li. “Leveraging a joint learning model to extract mixture symptom mentions from traditional Chinese medicine clinical notes”, *BioMed Research International*, **2022**, Conference Issue: Big Data for Biomedical Research, id. 2146236. [doi](#)
- [18] D. V. Gavrilov, A. V. Kirilkina, L. M. Serova. “Algorithm for forming a suspicion of a new coronavirus infection based on the analysis of symptoms for use in medical decision support systems”, *Vrach i informacionnye texnologii*, 2020, no. 4, pp. 51–58 (in Russian). [doi](#)
- [19] K. Lybarger, M. Ostendorf, M. Thompson, M. Yetisgen. “Extracting COVID-19 diagnoses and symptoms from clinical text: A new annotated corpus and neural event extraction framework”, *Journal of Biomedical Informatics*, **117** (2021), id. 103761. [doi](#)
- [20] D. Zolotukhin, I. Smurov. “RuNormAS-2021: A shared task on Russian normalization of annotated spans”, Papers from the Annual International Conference “Dialogue” (2021), Computational Linguistics and Intellectual Technologies, vol. **20**, 2021, ISBN 978-5-7281-3032-1, pp. 1245–1250. [doi](#)
- [21] T. Dozat, C. D. Manning. *Deep biaffine attention for neural dependency parsing*, 2017, 8 pp. [doi](#) [arXiv:1611.01734](#)
- [22] A. A. Sorokin, S. V. Makogonov, S. P. Korolev. “Information infrastructure for the collective work of scientists from the Russian Far East”, *Nauchno-texnicheskaya informaciya. Ser. 1: Organizaciya i metodika informacionnoj raboty*, 2017, no. 12, pp. 14–16 (in Russian).

УДК 517.977.5

10.25209/2079-3316-2023-14-1-125-148



## Об одном классе дискретно-непрерывных систем с параметрами

Ирина Викторовна **Расина**<sup>1&2</sup>, Ирина Сергеевна **Гусева**<sup>2</sup>

<sup>1</sup> Институт программных систем им. А. К. Айламазяна РАН, Вельково, Россия

<sup>1</sup> Федеральный исследовательский центр «Информатика и управление» РАН, Москва, Россия

<sup>2</sup> Бурятский государственный университет, Улан-Уде, Россия

<sup>1&2</sup> [irinarasina@gmail.com](mailto:irinarasina@gmail.com)

**Аннотация.** Рассматривается частный случай гибридной системы: дискретно-непрерывные системы (ДНС) с параметрами и промежуточными критериями. Такая система является двухуровневой. Параметры входят лишь в непрерывные системы, действующие поочередно на нижнем уровне. Верхний уровень, описываемый дискретным процессом, играет связующую роль для всех систем нижнего, определяя политику их взаимодействия и обеспечивая минимизацию функционала. В работе выводится аналог достаточных условий оптимальности Кротова и строится метод улучшения управления и параметров. Приводится иллюстративный пример. На основе полученных общих условий исследуется частный случай: квазилинейные ДНС.

**Ключевые слова и фразы:** дискретно-непрерывные системы с параметрами, промежуточные критерии, оптимальное управление, квазилинейные дискретно-непрерывные системы

**Благодарности:** Работа поддержана грантом РНФ № 21-11-00202

Для цитирования: Расина И.В., Гусева И.С. *Об одном классе дискретно-непрерывных систем с параметрами* // Программные системы: теория и приложения. 2023. Т. 14. № 1(56). С. 125–148. [https://psta.psir.ru/read/psta2023\\_1\\_125-148.pdf](https://psta.psir.ru/read/psta2023_1_125-148.pdf)

## Введение

В 80-е годы прошлого столетия в теории оптимального управления появился новый класс управляемых систем, главная особенность которых заключалась в способности изменять структуру своего описания с течением времени. Также выяснилось, что существует не единственный вариант подобных неоднородных систем. Позже для них возникло общее название – гибридные системы. Наиболее распространенные и изученные различными научными школами представлены в публикациях [1–7].

Один из подходов, основанный на развитии и обобщении как абстрактной модели многошаговых управляемых процессов, предложенной Кротовым, так и его же достаточных условий оптимальности [8], позволил провести декомпозицию неоднородной системы на однородные подсистемы и построить двухуровневую иерархическую модель [2], получившую название дискретно-непрерывной системы (ДНС). Такая система является двухуровневой. Верхний уровень, описываемый дискретным процессом, играет связующую роль для всех систем нижнего, определяя политику их взаимодействия и обеспечивая минимизацию функционала. Далее естественным образом были построены аналоги достаточных условий оптимальности и алгоритмов оптимизации [2, 9–11], разработанные для однородных систем, под которыми понимаются системы с неизменной структурой, исследуемые в рамках классических представлений теории оптимального управления.

Заметим, что при таком подходе все однородные подсистемы связаны общей целью, представленной в модели функционалом. Однако это не исключает того факта, что каждая однородная подсистема может при этом иметь и свою собственную цель. Такое обобщение модели дано в [12].

В данной работе рассматриваются дискретно-непрерывные системы (ДНС) с параметрами, которые широко распространены на практике. Параметры входят лишь в непрерывные системы, действующие поочередно на нижнем уровне. К этому же классу систем приводят задачи идентификации моделей по серии экспериментов, когда требуется определить коэффициенты построенной модели, которые можно рассматривать как искомые параметры. Для рассматриваемой модели ДНС с параметрами приведен в форме теорем аналог достаточных условий оптимальности Кротова. На основе последних построен метод улучшения управления и параметров. Аналогичный подход к традиционным управляемым процессам с параметрами представлен в [13], там же рассмотрена задача идентификации по серии экспериментов.

В общем виде, когда параметры содержит и верхний уровень ДНС, достаточные условия оптимальности и метод улучшения получены в [14]. Поскольку, как показывает практика, а именно, те же задачи идентификации, часто параметры могут содержать лишь непрерывные процессы. В этом случае метод улучшения существенно упрощается. Эта ситуация и рассматривается далее.

Отдельный интерес представляют практически не изученные квазилинейные ДНС, в которых параметр входит лишь в одно слагаемое в непрерывных системах и представляет собой множитель при нелинейном по состоянию слагаемом. Этот вариант рассмотрен отдельно. Небольшой обзор [15] дает представление о квазилинейных системах, содержащих нелинейность с малым параметром.

Обе рассмотренные ситуации проиллюстрированы примерами.

### 1. Модель дискретно-непрерывной системы с параметрами

Пусть на некотором подмножестве  $\mathbf{K}'$  множества  $\mathbf{K} = \{k_I, k_I + 1, \dots, k_F\}$ ,  $\mathbf{K} \subset \mathbb{N}$ , являющегося областью определения абстрактной дискретной управляемой системы:

$$(1) \quad x(k+1) = f(k, x(k), u(k)), \quad k \in \mathbf{K} = \{k_I, k_I + 1, \dots, k_F\}, \\ u \in \mathbf{U}(k, x),$$

действует непрерывная система

$$(2) \quad \dot{x}^c = \frac{dx^c}{dt} = f^c(z, t, x^c, u^c, \epsilon^c), \quad t \in \mathbf{T}(z) = [t_I(z), t_F(z)].$$

В этих системах:

$k$  – номер шага (этапа), не обязательно физическое время,  
 $x, x^c$  и  $u, u^c$  – соответственно переменные состояния и управления,  
 $f$  и  $f^c$  – операторы.

Все указанные объекты – произвольной природы (возможно различной) для различных  $k$ .

$\mathbf{U}(k, x)$  – заданное при каждом  $k$  и  $x$  множество,  
 $k_I, k_F$  – начальный и конечный шаги соответственно,  
 $z = (k, x, u^d)$  – совокупность переменных верхнего уровня, играющих на нижнем уровне роль параметров,  
 $\epsilon^c$  – параметр,  
 $u^d$  – переменная управления произвольной природы,  
 $x^c \in \mathbf{X}^c(z, t, \epsilon^c) \subset \mathbb{R}^{n(k)}$ ,

$$u^c \in \mathbf{U}^c(z, t, x^c, \epsilon^c) \subset \mathbb{R}^{p(k)},$$

$$\epsilon^c \in \mathbf{E}^c(k) \subset \mathbb{R}^{(k)}.$$

Система (1) имеет такую же структуру, как и система, предложенная Кротовым в [8].

Для системы (2) на отрезке  $[t_I(z), t_F(z)]$  задана промежуточная цель в виде функционала:

$$I^k = \int_{\mathbf{T}(z(k))} f^k(t, \epsilon^c, x^c(k, t), u^c(k, t)) dt \rightarrow \inf.$$

На множестве  $\mathbf{K}'$  оператор правой части (1) имеет вид  $f(k, x, u) = \theta(z, \gamma^c, \epsilon^c)$ , где

$$\gamma^c = (t_I, x_I^c, t_F, x_F^c) \in \mathbf{\Gamma}^c(z, \epsilon^c),$$

$$\mathbf{\Gamma}^c(z, \epsilon^c) = \{\gamma^c : t_I = \tau(z), x_I^c = \xi(z), (t_F, x_F^c) \in \mathbf{\Gamma}_F^c(z, \epsilon^c)\}.$$

Здесь  $t_I = \tau(z)$ ,  $x_I^c = \xi(z)$  – заданные функции  $z$ ,  $\mathbf{\Gamma}^c(z, \epsilon^c)$  – заданное множество. Нетрудно видеть, что система (1) представляет верхний уровень рассматриваемой математической модели, тогда как система (2) – система нижнего уровня.

Решением этой двухуровневой системы считается такой набор

$$m = (x(k), u(k)),$$

где при  $k \in \mathbf{K}'$ :  $u(k) = (u^d(k), m^c(k))$ ,  $m^c(k, \epsilon^c) \in \mathbf{D}^c(z(k), \epsilon^c)$ , (называемый *дискретно-непрерывным процессом с параметрами*), где

$$m^c(k, \epsilon^c) \text{ – непрерывный процесс } (x^c(k, t, \epsilon^c), u^c(k, t, \epsilon^c), \epsilon^c(k)),$$

$$t \in \mathbf{T}(z),$$

$\mathbf{D}^c(z, \epsilon^c)$  – множество допустимых процессов  $m^c$ , удовлетворяющих указанной дифференциальной системе (2) с дополнительными ограничениями при кусочно-непрерывных  $u^c(k, t, \epsilon^c)$  и кусочно-гладких  $x^c(k, t, \epsilon^c)$  (на каждом дискретном шаге  $k$ ).

Подчеркнем, что на каждом дискретном шаге  $k \in \mathbf{K}'$  модель рассматривает непрерывный процесс  $m^c(k, \epsilon^c)$ , управление которым состоит из фиксированной на шаге дискретной части  $u^d$  и непрерывно изменяемой части  $m^c$ .

Совокупность элементов  $m$ , удовлетворяющих всем вышеперечисленным условиям, обозначим через  $\mathbf{D}$  и назовем множеством допустимых дискретно-непрерывных процессов с параметрами.

Для модели (1), (2) рассматривается задача о поиске минимума на  $\mathbf{D}$  функционала  $I = F(x(k_F))$  при фиксированных  $k_I = 0$ ,  $k_F = K$ ,  $x(k_I)$  и

дополнительных ограничениях

$$(3) \quad x(k) \in \mathbf{X}(k), \quad x^c \in \mathbf{X}^c(z, t, \epsilon^c),$$

где  $\mathbf{X}(k)$ ,  $\mathbf{X}^c(z, \epsilon^c)$  – заданные множества.

Очевидно, что на нижнем уровне модели (1), (2) представлены однородные процессы на отдельных этапах, а верхний процесс выполняет связующую для них роль и управляет функционированием всей системы в целом. Под однородными здесь и далее понимаются системы с постоянной структурой, исследуемые в традиционных задачах теории оптимального управления. Взаимодействие верхнего уровня с каждой подсистемой нижнего осуществляется через границу этой подсистемы и соответствующего непрерывного процесса  $\gamma^c$ .

## 2. Математический аппарат

Введем в рассмотрение функционалы  $\varphi(k, x)$ ,  $\varphi^c(z, t, x^c, \epsilon^c)$  и построим с их помощью ряд конструкций. Одна из основных конструкций – обобщенный лагранжиан – аналог лагранжианов Кротова для дискретных и непрерывных систем [8, 16]:

$$\begin{aligned} L = & G(x(k_F)) - \sum_{\mathbf{K} \setminus \mathbf{K}' \setminus k_F} R(k, x(k), u(k)) + \\ & + \sum_{\mathbf{K}'} \left( G^c(z, \gamma^c(z)) - \int_{\mathbf{T}(z)} R^c(k, z, t, x^c(k, t), u^c(k, t), \epsilon^c) dt \right), \\ G(x) = & F(x) + \varphi(k_F, x) - \varphi(k_I, x(k_I)), \\ R(k, x, u) = & \varphi(k+1, f(k, x, u)) - \varphi(k, x), \\ G^c(z, \gamma^c, \epsilon^c) = & -\varphi(k+1, \theta(z, \gamma^c, \epsilon^c)) + \varphi(k, x) + \\ & + \varphi^c(z, t_F, x_F^c, \epsilon^c) - \varphi^c(z, t_I, x_I^c, \epsilon^c), \\ R^c(z, t, x^c, u^c, \epsilon^c) = & \varphi_{x^c}^{\text{T}} f^c(z, t, x^c, u^c, \epsilon^c) - \\ & - f^k(t, z, \epsilon^c, x^c(k, t), u^c(k, t)) + \varphi_t^c(z, t, x^c, \epsilon^c), \\ \mu^c(z, t, \epsilon^c) = & \sup_{\substack{x^c \in \mathbf{X}^c(z, t, \epsilon^c), \\ u^c \in \mathbf{U}^c(z, t, x^c, \epsilon^c)}} R^c(z, t, x^c, u^c, \epsilon^c), \\ l^c(z, \epsilon^c) = & \inf_{\gamma^c \in \Gamma(z, \epsilon^c)} G^c(z, \gamma^c, \epsilon^c), \end{aligned}$$

$$\mu(k, \epsilon^c) = \begin{cases} \sup_{\substack{x \in \mathbf{X}(k), \\ u \in \mathbf{U}(k, x)}} R(k, x, u), & k \in \mathbf{K} \setminus \mathbf{K}', \\ - \inf_{x \in \mathbf{X}(k)} l^c(z, \epsilon^c), & k \in \mathbf{K}', \end{cases}$$

$$l = \inf_{x \in \Gamma \cap \mathbf{X}(k_F)} G(x),$$

$$\mu^*(k) = \sup_{\epsilon^c \in \mathbf{E}^c} \left\{ \mu(k, \epsilon^c) - \int_{\mathbf{T}(z(k))} \mu^c(z, t, \epsilon^c) dt \right\}.$$

Здесь  $\varphi_{x^c}^c$  – градиент  $\varphi^c$  в пространстве  $(x^c)$ ,  $\mathbf{T}$  – знак транспонирования.

Справедливы следующие утверждения.

**ТЕОРЕМА 1.** Для любого элемента  $m \in \mathbf{D}$  и любых  $\varphi$ ,  $\varphi^c$  имеет место оценка

$$I(m) - \inf_{\mathbf{D}} I \leq \Delta = I(m) - l.$$

Пусть имеются два процесса  $m^I \in \mathbf{D}$  и  $m^{II} \in \mathbf{E}$  и функционалы  $\varphi$  и  $\varphi^c$ , такие что  $L(m^{II}) < L(m^I) = I(m^I)$  и  $m^{II} \in \mathbf{D}$ .

Тогда  $I(m^{II}) < I(m^I)$ .

**ТЕОРЕМА 2.** Пусть заданы: последовательность дискретно-непрерывных процессов с параметрами  $\{m_s\} \subset \mathbf{D}$  и функционалы  $\varphi$ ,  $\varphi^c$ , для которых справедливы следующие условия:

- 1)  $\mu^c(z, t, \epsilon^c)$  – кусочно-непрерывна при каждом  $z$ ;
- 2)  $R(k, x_s(k), u_s(k)) \rightarrow \mu(k)$ ,  $k \in \mathbf{K}$ ;
- 3)  $\int_{\mathbf{T}(z_s)} (R^c(z_s, t, x_s^c(t), u_s^c, \epsilon_s^c) - \mu^c(z_s, t, \epsilon_s^c)) dt \rightarrow 0$ ,  $k \in \mathbf{K}'$ ,  $t \in \mathbf{T}(z_s)$ ;
- 4)  $G^c(z_s, \gamma_s^c, \epsilon_s^c) - l^c(z_s, \epsilon_s^c) \rightarrow 0$ ,  $k \in \mathbf{K}'$ ;
- 5)  $G(x_s(t_F)) \rightarrow l$ ;
- 6)  $\mu(k, \epsilon_s^c) - \int_{\mathbf{T}(z(k))} \mu^c(z, t, \epsilon_s^c) dt \rightarrow \mu^*(k)$ .

Тогда последовательность  $\{m_s\}_{s \in \mathbb{N}}$  – минимизирующая для  $I$  на  $\mathbf{D}$ .

Доказательство обоих утверждений непосредственно следует из принципа расширения [16, 17] и проводится по аналогии с доказательством для ДНС без параметра [12].

### 3. Метод улучшения управления и параметра

Предположим, что  $\mathbf{X}(k) = \mathbb{R}^{m(k)}$ ,  $\mathbf{X}^c(z, t, \epsilon^c) = \mathbb{R}^{n(k)}$ ,  $\mathbf{E}^c(k) = \mathbb{R}$ ,  $x_I^c = \xi(z, \epsilon^c)$ ,  $k_I$ ,  $x_I$ ,  $k_F$ ,  $t_I(k)$ ,  $t_F(k)$  – заданы,  $x_F^c \in \mathbb{R}^{n(k)}$  и системы нижнего уровня не зависят от управления  $u^d$ . Как указано в [18], задача улучшения состоит, по существу, в построении оператора  $\theta(m)$ ,  $\theta : \mathbf{D} \rightarrow \mathbf{D}$ , такого что  $I(\theta(m)) \leq I(m)$ . При некотором заданном начальном элементе такой оператор генерирует улучшающую, в частности, минимизирующую последовательность  $\{m_s\} : m_{s+1} = \theta(m_s)$ .

Используем для построения метода принципы расширения [17] и локализации [19]. Из последнего следует, что задачу улучшения некоторого элемента  $m^I$  можно заменить задачей о минимуме вспомогательного функционала

$$(4) \quad I_\alpha(m) = \alpha I(m) + \frac{1}{2}(1 - \alpha)J(m^I, m), \quad \alpha \in [0, 1].$$

Элемент, доставляющий минимум функционалу (4), обозначим через  $m_\alpha$ . Во втором слагаемом  $J(m^I, m)$  – функционал типа метрики. Изменяя  $\alpha$  от 0 к 1, можно достичь необходимой степени близости  $m_\alpha$  к  $m^I$ , что позволяет получить алгоритм с параметром  $\alpha$ . Выбор  $\alpha$  определяется при конкретном применении. Задача этого параметра – обеспечить наибольшее значение разности  $I(m^I) - I(m_\alpha)$ , тогда соответствующий элемент  $m_\alpha$  принимается за  $m^{II}$ .

Вспомогательный функционал зададим в виде:

$$I_\alpha = \alpha I + \frac{1}{2}(1 - \alpha) \left( \sum_{\mathbf{K} \setminus \mathbf{K}' \setminus k_F} |\Delta u(k)|^2 + \sum_{\mathbf{K}'} \int_{\mathbf{T}(z)} |\Delta u^c(k, t)|^2 dt \right).$$

В представленном выше выражении  $\alpha \in [0, 1]$ ,  $\Delta u = u - u^I$ ,  $\Delta u^c = u^c - u^{cI}$ .

Согласно указанному принципу расширения по заданному элементу  $m^I \in \mathbf{D}$  требуется найти элемент  $m^{II} \in \mathbf{D}$ , на котором  $I_\alpha(m^{II}) = L_\alpha(m^{II}) < I_\alpha(m^I) = L_\alpha(m^I)$  или  $L_\alpha(m^{II}) - L_\alpha(m^I) < 0$ . Рассмотрим приращение функционала  $L_\alpha(m)$ . Имеем:

$$\begin{aligned} \Delta L_\alpha \approx & G_{x_F}^T \Delta x_F - \sum_{\mathbf{K} \setminus \mathbf{K}' \setminus k_F} \left( R_x^T \Delta x + R_u^T \Delta u + \frac{1}{2} \Delta u^T R_{uu} \Delta u \right) + \\ & + \sum_{\mathbf{K}' \setminus k_F} \left( G_{x_F^c}^{cT} \Delta x_F^c + G_x^{cT} \Delta x + G_{\epsilon^c}^{cT} \Delta \epsilon^c - \right. \\ & \left. - \int_{\mathbf{T}(z)} \left( R_{x^c}^{cT} \Delta x^c + R_x^{cT} \Delta x + R_{u^c}^{cT} \Delta u^c + \right. \right. \end{aligned}$$

$$+ R_{\epsilon^c}^{\epsilon^c \Gamma} \Delta \epsilon^c + \frac{1}{2} \Delta u^{\epsilon^c \Gamma} R_{u^c u^c}^c \Delta u^c \Big) dt \Big),$$

где  $\Delta u = u - u^I$ ,  $\Delta x = x - x^I$ ,  $\Delta u^c = u^c - u^{cI}$ ,  $\Delta x^c = x^c - x^{cI}$ ,  $\Delta x_F^c = x_F^c - x_F^{cI}$ ,  $\Delta \epsilon^c = \epsilon^c - \epsilon^{cI}$ ,  $x_F = x(k_F)$ . Здесь функции  $R$ ,  $G$ ,  $R^c$ ,  $G^c$  определены для функционала  $I_\alpha$ , а их первые и вторые производные подсчитаны при  $u = u^I$ ,  $x = x^I$ ,  $x^c = x^{cI}$ ,  $u^c = u^{cI}$ ,  $\epsilon^c = \epsilon^{cI}$ .

Предположим, что матрицы  $R_{uu}$  и  $R_{u^c u^c}^c$  отрицательно определены (это всегда достижимо за счет выбора параметра  $\alpha$  [19]). Найдем  $\Delta u$ ,  $\Delta u^c$ , доставляющие максимум выражениям, стоящим под знаками сумм  $\sum_{\mathbf{K} \setminus \mathbf{K}' \setminus k_F}$

и  $\sum_{\mathbf{K}' \setminus t_F}$  соответственно. Нетрудно видеть, что

$$\Delta u = -(R_{uu})^{-1} R_u, \quad \Delta u^c = -(R_{u^c u^c}^c)^{-1} R_{u^c}^c.$$

Зададим функции  $\varphi(k, x)$  и  $\varphi^c(z, t, x, \epsilon^c)$  в виде:

$$\varphi = \psi^T(k)x(k), \quad \varphi^c = \lambda^T(k, t)x(k) + \psi^{cT}(k, t)x^c(k, t),$$

где  $\psi, \psi^c, \lambda$  – вектор-функции. Для выполнения неравенства  $\Delta L < 0$  потребуем далее, чтобы остальные слагаемые под знаками вышеуказанных сумм не зависели от  $\Delta x$ ,  $\Delta x_F^c$ ,  $\Delta x^c$ . Это требование будет достигнуто, если:

$$G_{x_F} = 0, \quad R_x = 0, \quad G_{x_F^c}^c = 0, \quad G_x^c = 0, \quad R_{x^c}^c = 0, \quad R_x^c = 0.$$

Дополним указанные равенства условием стыковки уровней:

$$(5) \quad \frac{d}{dx} (\varphi(k+1, \theta(k, x, x_F^c, x_I^c)) - \varphi^c(k, x, t_F, x_F^c)) = 0.$$

Расшифровка вышеприведенных равенств с учетом условия стыковки уровней приводит к задаче Коши для ДНС относительно  $\psi$ ,  $\psi^c$ ,  $\lambda$  с начальными условиями на правом конце:

$$(6) \quad \psi(k_F) = -\alpha F_{x(k_F)}, \quad \psi(k) = H_x(\psi(k+1)),$$

$$H(k, \psi, x, u) = \psi^T(k+1)f(k, x, u), \quad k \in \mathbf{K} \setminus \mathbf{K}' \setminus k_F,$$

$$\begin{aligned} \psi(k) &= H_x + (H_{x^c} \xi_x)^T + \lambda(k, t_F) - \\ &\quad - \lambda(k, t_I) + \xi_x^T \psi^c(t_I), \quad k \in \mathbf{K}' \setminus k_F, \end{aligned}$$

$$H(k, \psi, x_I^c, x_F^c) = \psi^T(k+1)\theta(z, x_I^c, x_F^c), \quad k \in \mathbf{K}',$$

$$\begin{aligned} \dot{\lambda} &= -H_x^c, & \lambda(k, t_F) &= H_x + \xi_x^T H_{x_I^c}, \\ \dot{\psi}^c &= -H_{x^c}^c, & \psi^c(k, t_F) &= H_{x_F^c}, \end{aligned}$$

$$H^c(z, t, \psi^c, x^c, u^c, \epsilon^c) = \psi^{cT} f^c(z, t, x^c, u^c, \epsilon^c(k)) - \\ - f^k(t, x^c(k, t), u^c(k, t), \epsilon^c(k)), \quad k \in \mathbf{K}'.$$

Здесь для краткости в правых частях указаны лишь аргументы  $\psi(k+1)$  и  $\psi^c(k, t)$ , необходимые для понимания соотношений. Отметим, что полученная система линейна, т.е. заведомо разрешается.

При этом приращения управляющих воздействий имеют вид:

$$(7) \quad \Delta u = -(H_{uu} - (1 - \alpha)E)^{-1} H_u, \\ \Delta u^c = -(H_{u^c u^c} - (1 - \alpha)E)^{-1} H_{u^c},$$

где  $E$  – единичные матрицы соответствующих размеров.

Запишем оставшиеся слагаемые в выражении для приращения функционала  $\Delta L_\alpha$ , учитывая подстановку в него формул для приращений управлений обоих уровней, и обозначив итог через  $\Delta M_\alpha$ . Имеем:

$$\Delta M_\alpha \approx - \sum_{\mathbf{K} \setminus \mathbf{K}' \setminus k_F} \left( -\frac{1}{2} \Delta u^T R_{uu} \Delta u \right) + \\ + \sum_{\mathbf{K}' \setminus k_F} \left( G_{\epsilon^c}^{cT} \Delta \epsilon^c - \int_{\mathbf{T}(z)} \left( R_{\epsilon^c}^{cT} \Delta \epsilon^c - \frac{1}{2} \Delta u^{cT} R_{u^c u^c} \Delta u^c \right) dt \right).$$

В приведенном выражении при сделанных ранее предположениях квадратичные формы от  $\Delta u$  и  $\Delta u^c$  отрицательно определены.

Рассмотрим теперь слагаемые, зависящие от  $\Delta \epsilon^c$ . Выберем  $\Delta \epsilon^c$  таким образом, чтобы рассматриваемое выражение обеспечивало выполнение неравенства  $\Delta M_\alpha < 0$ . Выбор становится очевидным, если прибегнуть к подробной записи скалярных произведений, входящих в выражение для  $\Delta M_\alpha$ . Итак:

$$\Delta M_\alpha \approx \frac{1}{2} \sum_{\mathbf{K} \setminus \mathbf{K}' \setminus k_F} \Delta u^T R_{uu} \Delta u + \sum_{\mathbf{K}' \setminus k_F} \sum_{i=1}^{b(k)} \left( G_{\epsilon_i^c}^{cT} \Delta \epsilon_i^c - \int_{\mathbf{T}(z)} R_{\epsilon_i^c}^{cT} \Delta \epsilon_i^c dt \right) + \\ + \frac{1}{2} \sum_{\mathbf{K}' \setminus k_F} \int_{\mathbf{T}(z)} \Delta u^{cT} R_{u^c u^c} \Delta u^c dt.$$

Отсюда следует, что

$$\Delta \epsilon_i^c(k) = \beta^c \left( G_{\epsilon_i^c}^c - \int_{\mathbf{T}(z)} R_{\epsilon_i^c}^c dt \right), \quad k \in \mathbf{K}' \setminus k_F, \quad i = 1, 2, \dots, b(k),$$

где  $\beta^c$  – коэффициент,  $\beta^c < 0$ .

#### 4. Итерационная процедура

На основе полученных соотношений можно сформулировать следующую итерационную процедуру.

1. Задаются  $\alpha$  и  $\beta^c(k)$ . «Слева направо» просчитывается ДНС (1), (2) при  $u = u_s(k)$ ,  $u^c = u_s^c(k, t)$ ,  $\epsilon_s^c$ , заданных начальных условиях, получаются соответствующие траектории  $(x_s(k), x_s^c(k, t))$ .
2. «Справа налево» разрешается ДНС относительно  $\psi(k)$ ,  $\psi^c(k, t)$  и  $\lambda(k, t)$ .
3. Находятся  $\Delta u$ ,  $\Delta u^c$ , новые управления  $u = u_s(k) + \Delta u$ ,  $u^c = u_s^c(k, t) + \Delta u^c$  и новое значение параметра  $\epsilon^c$  при каждом  $k$ .
4. При найденных управлениях, параметре и начальном условии  $x(k_I) = x_I$  просчитывается «слева направо» исходная ДНС (1), (2). Тем самым определяется новый элемент  $m_{s+1}$ .

Процесс итераций заканчивается, когда  $|I_{s+1} - I_s| \approx 0$  с заданной точностью.

Имеет место следующее утверждение о сходимости.

**ТЕОРЕМА 3.** *Предположим, что функционал  $I$  ограничен снизу и для ДНС (1), (2) построена указанная итерационная процедура. Тогда последовательность элементов  $\{m_s\} \in \mathbf{D}$  сходится по функционалу, т.е. существует число  $I^*$ , такое что  $I^* \leq I(m_s)$ ,  $I(m_s) \rightarrow I^*$ .*

**ДОКАЗАТЕЛЬСТВО.** Очевидно, что рассмотренный оператор улучшения обеспечивает монотонность построенной последовательности  $\{m_s\}$  (по функционалу). Таким образом, получается монотонная числовая последовательность

$$\{I_s\} = \{I(m_s)\}, \quad I_{s+1} \leq I_s,$$

ограниченная снизу, которая по известной теореме анализа сходится к некоторому пределу:  $I_s \rightarrow I^*$ .  $\square$

Продемонстрируем работу предложенного алгоритма на примере.

#### 5. Пример 1

Рассмотрим работу метода на примере системы, динамика которой включает в себя два этапа.

*1-ый этап:*

$$\begin{aligned} \dot{x}_1^c &= \epsilon^c (x_1^c)^2 + u^c, & \dot{x}_2^c &= (\epsilon^c)^2 x_1^c x_2^c + \frac{1}{2} (u^c)^2, \\ x_1^c(0) &= 0.3, & x_2^c(0) &= 1, \end{aligned}$$

$$t \in [0, 1], \quad I^0 = \int_0^1 ((x_1^c)^2 - x_2^c u^c) dt.$$

2-ой этап:

$$\dot{x}_1^c = x_2^c u^c, \quad \dot{x}_2^c = -\epsilon^c (x_1^c)^2 + (u^c)^2,$$

$$t \in [1, 3], \quad I^1 = \int_1^3 x_1^c (u^c)^2 dt.$$

Функционал в примере:  $I = x_1^c(3) \rightarrow \min$ . Построим ДНС. Нетрудно видеть, что  $\mathbf{K} = 0, 1, 2$ ;  $\mathbf{K}' = 1$ . Поскольку оба этапа связаны через переменную  $x_1^c$ , то она и играет роль  $x$ , а дискретный процесс верхнего уровня принимает вид:

$$\begin{aligned} x_1(0) = x_1^c(0, 0) = 0.3, & & x_2(0) = x_2^c(0, 0) = 1, \\ x(1) = x^c(0, 1), & & x(2) = x^c(1, 3), \\ I = x(2), & & x^c(1, 1) = x(1), \\ \xi = x(1), & & \theta(1) = x^c(0, 1). \end{aligned}$$

Модель ДНС может быть дополнена третьим мгновенным этапом, не имеющим протяженности во времени и состоящим лишь в передачи информации об окончании второго этапа на верхний уровень. Тогда  $x(3) = x(2) = x^c(1, 3)$ .

Основные конструкции имеют вид:

$$\begin{aligned} H^c(0, t, x_1^c, x_2^c, u^c, \psi_1^c, \psi_2^c, \epsilon^c) &= \psi_1^c (\epsilon^c (x_1^c)^2 + u^c) + \\ &+ \psi_2^c \left( (\epsilon^c)^2 x_1^c x_2^c + \frac{1}{2} (u^c)^2 \right) - (x_1^c)^2 + x_2^c u^c, \end{aligned}$$

$$H^c(1, t, x_1^c, x_2^c, u^c, \psi_1^c, \psi_2^c, \epsilon^c) = \psi_1^c x_2^c u^c + \psi_2^c (-\epsilon^c (x_1^c)^2 + (u^c)^2) - x_1^c (u^c)^2.$$

Поскольку процессы нижнего уровня не зависят от  $x$ , то на обоих этапах  $\lambda = 0$ . На первом этапе при  $k = 0$  имеем:

$$\begin{aligned} \dot{\psi}_1^c &= -2\psi_1^c \epsilon^c x_1^c - \psi_2^c (\epsilon^c)^2 x_2^c + 2x_1^c, \\ \dot{\psi}_2^c &= -\psi_2^c (\epsilon^c)^2 x_1^c - u^c, \\ H_{u^c}^c &= \psi_1^c + \psi_2^c u^c + x_2^c, \\ H_{u^c u^c}^c &= \psi_2^c, \\ H_{\epsilon^c}^c &= \psi_1^c (x_1^c)^2 + 2\psi_2^c \epsilon^c x_1^c x_2^c. \end{aligned}$$

На втором этапе при  $k = 1$  конструкции принимают вид:

$$\begin{aligned}\dot{\psi}_1^c &= 2\psi_2^c \epsilon^c x_1^c + (u^c)^2, \\ \dot{\psi}_2^c &= -\psi_1^c u^c, \\ H_{u^c}^c &= \psi_1^c x_2^c + 2u^c(\psi_2^c - x_1^c), \\ H_{u^c u^c}^c &= 2(\psi_2^c - x_1^c), \\ H_{\epsilon^c}^c &= -\psi_2^c (x_1^c)^2.\end{aligned}$$

Выпишем остальные конструкции. Поскольку  $\mathbf{K}' = 1$ , то

$$\begin{aligned}G^c &= -\psi_1(1)x_1^c(0, 1) - \psi_2(1)x_2^c(0, 1) + \psi_1(0)x_1^c(0, 0) + \\ &\quad + \psi_2(0)x_2^c(0, 0) + \psi_1^c(1, 3)x_1^c(1, 3) + \psi_2^c(1, 3)x_2^c(1, 3) - \\ &\quad - \psi_1^c(0, 0)x_1^c(0, 0) - \psi_2^c(0, 0)x_2^c(0, 0), \\ H(1, \psi_1, \psi_2, x_1^c(0, 1), x_2^c(0, 1)) &= \psi_1(2)x_1^c(0, 1) + \psi_2(2)x_2^c(0, 1), \\ G &= x_1(2) + \psi_1(2)x_1(2) + \psi_2(2)x_2(2) - \psi_1(0)x_1(0) - \psi_2(0)x_2(0), \\ \psi_1(2) &= -\alpha, \quad \psi_1^c(0, 1) = \psi_1^c(1, 3) = \psi_1(2), \\ \psi_2(2) &= 0, \quad \psi_2^c(0, 1) = \psi_2^c(1, 3) = \psi_2(2).\end{aligned}$$

Формулы для приращений управления и параметра на этапах имеют следующий вид. На первом этапе:

$$\begin{aligned}\Delta u^c(0, t) &= -\frac{\psi_1^c + \psi_2^c u^c + x_2^c}{\alpha - 1 + \psi_2^c}, \\ \Delta \epsilon^c(0) &= -\beta^c(0) \int_0^1 \left( \psi_1^c(x_1^c)^2 + 2\psi_2^c \epsilon^c x_1^c x_2^c \right) dt.\end{aligned}$$

На втором этапе:

$$\begin{aligned}\Delta u^c(1, t) &= -\frac{\psi_1^c x_2^c + 2u^c(\psi_2^c - x_1^c)}{\alpha - 1 + 2(\psi_2^c - x_1^c)}, \\ \Delta \epsilon^c(1) &= \beta^c(1) \int_1^3 \psi_2^c (x_1^c)^2 dt.\end{aligned}$$

На решение примера потребовалось 2 итерации. При этом значение функционала с  $I^0 = 0.8976$  уменьшилось до  $I^2 = 0.7043$ . Изменение функционала по итерациям представлено в таблице 1. Графики управляющих

Таблица 1. Изменение значений функционала по итерациям

$k$	$\beta^c(0)$	$\beta^c(1)$	$\epsilon^c(0)$	$\epsilon^c(1)$	$I^k$	$ I^k - I^{k-1} $
0			0.3	0.5	0.8976	
1	-0.6	-1	0.2327	0.5590	0.7053	0.1923
2	-0.4	-0.5	0.2076	0.5510	0.7043	0.001

воздействий и состояний процесса на первой и последней итерациях даны на рисунках 1, 2. При расчетах параметр  $\alpha = 0.3$ .

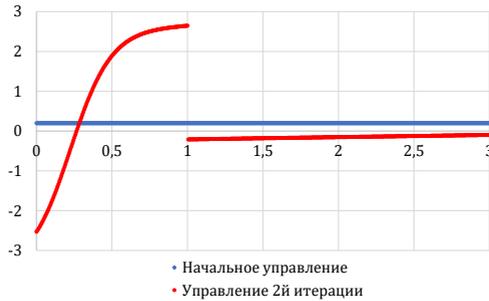


Рисунок 1. График управляющих воздействий по итерациям

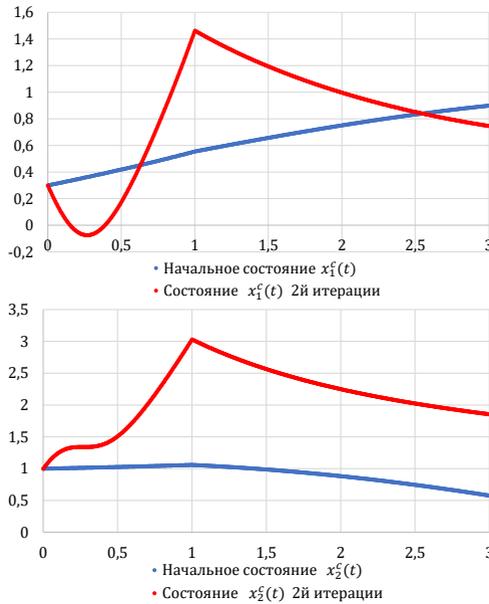


Рисунок 2. Графики состояний  $x_1^c(t)$  и  $x_2^c(t)$  процесса по итерациям

Рассмотрим далее частный случай приведенной выше модели.

### б. Квазилинейные дискретно-непрерывные системы

В данном случае система верхнего уровня имеет вид:

$$(8) \quad x(k+1) = A(k, x(k))x(k) + B(k, x(k))u(k),$$

а на нижнем уровне действуют непрерывные управляемые системы:

$$(9) \quad \dot{x}^c = \frac{dx^c}{dt} = A^c(k, t, x^c)x^c + B^c(k, t, x^c)u^c + \epsilon^c f(k, x^c),$$

$$t \in \mathbf{T}(k) = [t_I(k), t_F(k)].$$

Здесь

$A, B, A^c, B^c$  – матрицы размеров  $m(k) \times m(k)$ ,  $m(k) \times p(k)$ ,  $n(k) \times n(k)$ ,  $n(k) \times r(k)$  соответственно,

$f(k, x^c)$  – заданная при каждом  $k$  функция,

$\epsilon^c$  – параметр.

Кроме того, при каждом  $k \in \mathbf{K}'$  оператор правой части (8) имеет вид

$$x(k+1) = \theta(k)x_F^c,$$

где  $\theta(k)$  – матрица размера  $m(k) \times n(k)$ ,

$$x(k) \in \mathbb{R}^{m(k)}, \quad u(k) \in \mathbb{R}^{p(k)},$$

$$x^c \in \mathbf{X}^c(z, t, \epsilon^c) \subset \mathbb{R}^{n(k)},$$

$$u^c \in \mathbf{U}^c(z, t, x^c, \epsilon^c) \subset \mathbb{R}^{p(k)},$$

$$\epsilon^c \in \mathbf{E}^c(k) \subset \mathbb{R}^{(k)}, \quad z = (k, x).$$

Для системы (9) на отрезке  $[t_I(k), t_F(k)]$  задана промежуточная цель в виде функционала:

$$I^k = \frac{1}{2} \int_{\mathbf{T}(k)} (x^{cT} S^c(t, k, x^c)x^c + u^{cT} Q^c(k, t, x^c)u^c) dt +$$

$$+ \lambda^c(k, x_F^c)x_F^c + \frac{1}{2}(x_F^c)^T \Lambda^c(k, x_F^c)x_F^c \rightarrow \inf.$$

Здесь

$S^c, Q^c, \Lambda^c$  – матрицы размеров  $n(k) \times n(k)$ ,  $r(k) \times r(k)$ ,  $n(k) \times n(k)$  соответственно,

$\lambda^c$  – вектор размера  $n(k)$ ;

как и ранее,

$z = (k, x)$  – совокупность переменных верхнего уровня, играющих на нижнем уровне роль параметров,

$t_I = \tau(z)$ ,  $x_I^c = \xi(z)$  – заданные функции  $z$ .

Решением этой двухуровневой системы считается набор

$$m = (x(k), u(k)),$$

где при  $k \in \mathbf{K}'$ :

$$u(k) = (m^c(k, \epsilon^c)), \quad m^c(k, \epsilon^c) \in \mathbf{D}^c(z, \epsilon^c)$$

(называемый *квазилинейным дискретно-непрерывным процессом*). Как и ранее,

$m^c(k, \epsilon^c)$  – непрерывный процесс  $(x^c(k, t, \epsilon^c), u^c(k, t, \epsilon^c), \epsilon^c(k))$ ,  
 $t \in \mathbf{T}(z)$ ,

$\mathbf{D}^c(z, \epsilon^c)$  – множество допустимых процессов  $m^c$ , удовлетворяющих указанной дифференциальной системе (9) с дополнительными ограничениями при кусочно-непрерывных  $u^c(k, t, \epsilon^c)$  и кусочно-гладких  $x^c(k, t, \epsilon^c)$  (на каждом дискретном шаге  $k$ ).

Совокупность элементов  $m$ , удовлетворяющих всем указанным условиям, обозначим через  $\mathbf{D}$  и назовем множеством допустимых квазилинейных дискретно-непрерывных процессов. Минимизируемый функционал:  $I = F(x(k_F))$ .

Поскольку приведенная модель является частным случаем уже рассмотренной модели, то для нее справедливы теоремы 1 и 2. Для удобства изложения приведем основные конструкции достаточных условий оптимальности.

## 7. Основные конструкции и уравнения метода улучшения

Имеем:

$$L = G(x(k_F)) - \sum_{\mathbf{K} \setminus \mathbf{K}' \setminus k_F} R(k, x(k), u(k)) + \\ + \sum_{\mathbf{K}'} \left( G^c(z, \gamma^c(z)) - \int_{\mathbf{T}(z)} R^c(k, z, t, x^c(k, t), u^c(k, t), \epsilon^c) dt \right),$$

$$G(x) = F(x) + \varphi(k_F, x) - \varphi(k_I, x(k_I)),$$

$$R(k, x, u) = \varphi(k+1, A(k, x(k))x(k) + B(k, x(k))u(k)) - \varphi(k, x),$$

$$G^c(z, \gamma^c, \epsilon^c) = -\varphi(k+1, \theta(k)x_F^c) + \varphi(k, x) + \\ + \varphi^c(z, t_F, x_F^c, \epsilon^c) - \varphi^c(z, t_I, x_I^c, \epsilon^c),$$

$$R^c(z, t, x^c, u^c, \epsilon^c) = \varphi_{x^c}^{cT}(A^c(k, t, x^c)x^c + B^c(k, t, x^c)u^c + \epsilon^c f(k, x^c)) - \\ - f^k(t, z, x^c(k, t), u^c(k, t)) + \varphi_t^c(z, t, x^c, \epsilon^c),$$

$$\mu^c(z, t, \epsilon^c) = \sup_{\substack{x^c \in \mathbf{X}^c(z, t), \\ u^c \in \mathbf{U}^c(z, t, x^c)}} R^c(z, t, x^c, u^c, \epsilon^c),$$

$$\begin{aligned}
l^c(z, \epsilon^c) &= \inf_{\gamma^c \in \Gamma(z)} G^c(z, \gamma^c, \epsilon^c), \\
\mu(k) &= \begin{cases} \sup_{\substack{x \in \mathbf{X}(k), \\ u \in \mathbf{U}(k, x)}} R(k, x, u), & k \in \mathbf{K} \setminus \mathbf{K}', \\ - \inf_{x \in \mathbf{X}(k)} l^c(z, \epsilon^c), & k \in \mathbf{K}', \end{cases} \\
l &= \inf_{x \in \Gamma \cap \mathbf{X}(k_F)} G(x), \\
\mu^*(k) &= \sup_{\epsilon^c \in \mathbf{E}^c} \left\{ \mu(k, \epsilon^c) - \int_{\mathbf{T}(z(k))} \mu^c(z, t, \epsilon^c) dt \right\}.
\end{aligned}$$

Выпишем теперь уравнения метода улучшения.

$$\begin{aligned}
(10) \quad \psi(k_F) &= -\alpha F_{x(k_F)}, \quad \psi(k) = H_x(\psi(k+1)), \\
H(k, \psi, x, u) &= \psi^T(k+1)(A(k, x(k))x(k) + \\
&\quad + B(k, x(k))u(k)), \quad k \in \mathbf{K} \setminus \mathbf{K}' \setminus k_F, \\
\psi(k) &= H_x + (H_{x^c} \xi_x)^T + \xi_x^T \psi^c(t_I), \quad k \in \mathbf{K}' \setminus k_F, \\
H(k, \psi, x_I^c, x_F^c) &= \psi^T(k+1)\theta x_F^c, \quad k \in \mathbf{K}', \\
\dot{\psi}^c &= -H_{x^c}^c, \quad \psi^c(k, t_F) = H_{x_F^c}^c, \\
H^c(z, t, \psi^c, x^c, u^c, \epsilon^c) &= \psi^{cT} (A^c(k, t, x^c)x^c + B^c(k, t, x^c)u^c + \epsilon^c f(k, x^c)) - \\
&\quad - \frac{1}{2} (x^{cT} S^c(t, k, x^c)x^c + u^{cT} Q^c(k, t, x^c)u^c), \quad k \in \mathbf{K}'.
\end{aligned}$$

Здесь для краткости в правых частях указаны лишь аргументы  $\psi(k+1)$  и  $\psi^c(k, t)$ , необходимые для понимания соотношений. Отметим, что полученная система линейна, т.е. заведомо разрешается.

При этом приращения управляющих воздействий имеют вид:

$$\begin{aligned}
(11) \quad \Delta u &= -(H_{uu} - (1 - \alpha E))^{-1} H_u, \\
\Delta u^c &= -(H_{u^c u^c}^c - (1 - \alpha E))^{-1} H_{u^c}^c,
\end{aligned}$$

где  $E$  – единичные матрицы соответствующих размеров. Иначе:

$$\begin{aligned}
(12) \quad \Delta u &= -(Q - (1 - \alpha)E)^{-1} (B^T \psi), \\
\Delta u^c &= -(Q^c - (1 - \alpha)E)^{-1} (B^{cT} \psi^c - Q^c u^c), \\
\Delta \epsilon^c &= \beta^c \left( \int_{\mathbf{T}(z)} f^T \psi^c dt \right), \quad k \in \mathbf{K}' \setminus k_F.
\end{aligned}$$

Нетрудно видеть, что алгоритм метода состоит из тех же пунктов, что и ранее. Он был реализован на следующем примере.

## 8. Пример 2

Задана состоящая из двух этапов квазилинейная ДНС с коэффициентами, которые зависят от состояния:

1-ый этап:

$$\begin{aligned} \dot{x}_1^c &= (x_1^c)^2(x_1^c - u_1^c) + \epsilon^c (x_1^c)^3, \\ x_1^c(0) &= 0.3, \\ t \in [0, 2], \quad I^0 &= \int_0^2 (x_1^c - 1)^2 (u_1^c)^2 dt. \end{aligned}$$

2-ой этап:

$$\begin{aligned} \dot{x}_1^c &= x_1^c \exp(-x_1^c) - 2x_1^c u_1^c + \epsilon^c \sqrt{x_1^c}, \\ t \in [2, 3], \quad I^1 &= \int_2^3 (u_1^c x_1^c)^2 dt, \\ I &= (x_1^c(3))^3 - 3x_1^c(3) \rightarrow \min. \end{aligned}$$

Приведем задачу к стандартному виду. Очевидно, что  $\mathbf{K} = \{0, 1, 2\}$ . При  $k = 0$   $n = r = 1$ , а матрицы нижнего уровня имеют вид:

$$\begin{aligned} A^c(0) &= (x_1^c)^2, & B^c(0) &= -(x_1^c)^2, & Q^c(0) &= 2(x_1^c - 1)^2, \\ S^c(0) &= 0, & f(0, x_1^c) &= (x_1^c)^3. \end{aligned}$$

На следующем этапе при  $k = 1$  также  $n = r = 1$ , а матрицы нижнего уровня выглядят следующим образом:

$$\begin{aligned} A^c(1) &= \exp(-x_1^c), & B^c(1) &= -2x_1^c, & Q^c(1) &= 2(x_1^c)^2, \\ S^c(1) &= 0, & f(1, x_1^c) &= \sqrt{x_1^c}. \end{aligned}$$

Нетрудно видеть, что на верхнем уровне за  $x$  имеет смысл взять  $x_1^c$ , поскольку эта переменная проходит через оба этапа и через нее же определен общий функционал. Имеем

$$\begin{aligned} I &= (x(2))^3 - 3x(2), \\ \lambda &= -3, \quad \Lambda = (x(2))^2, \\ x(1) &= x_1^c(0, 2), \quad x_1^c(1, 2) = x(1), \\ \theta &= 1, \quad \theta^0 = 0, \quad \xi = 1. \end{aligned}$$

Выпишем основные конструкции, необходимые для расчетов. Имеем:

$$\begin{aligned} H^c(0, t, \psi_1^c, \psi_2^c, x_1^c, x_2^c, u_1^c, u_2^c) &= \psi_1^c ((x_1^c)^3 (\epsilon^c + 1) - (x_1^c)^2 u_1^c) - \\ &\quad - (x_1^c - 1)^2 (u_1^c)^2, \end{aligned}$$

$$H^c(1, t, \psi_1^c, x_1^c, u_1^c) = \psi_1^c \left( x_1^c \exp(-x_1^c) - 2x_1^c u_1^c + \epsilon^c \sqrt{x_1^c} \right) - (x_1^c u_1^c)^2,$$

На первом этапе:

$$\begin{aligned} \dot{\psi}_1^c &= -\psi_1^c \left( 3(x_1^c)^2(\epsilon^c + 1) - 2x_1^c u_1^c \right) + 2(x_1^c - 1)(u_1^c)^2, \\ H_{u_1^c}^c &= -\psi_1^c (x_1^c)^2 - 2(x_1^c - 1)^2 u_1^c, \\ H_{u_1^c u_1^c}^c &= -2(x_1^c - 1)^2, \\ \Delta u_1^c(0) &= \frac{\psi_1^c (x_1^c)^2 + 2(x_1^c - 1)^2 u_1^c}{\alpha - 1 - 2(x_1^c - 1)^2}, \\ \Delta \epsilon^c &= \beta^c \int_0^2 \psi_1^c (x_1^c)^3 dt. \end{aligned}$$

На втором этапе:

$$\begin{aligned} \dot{\psi}_1^c &= -\psi_1^c \left( \exp(-x_1^c) - x_1^c \exp(-x_1^c) - 2u_1^c + \frac{\epsilon^c}{2\sqrt{x_1^c}} \right) + 2(u_1^c)^2 x_1^c, \\ H_{u_1^c}^c &= -2\psi_1^c x_1^c - 2u_1^c (x_1^c)^2, \\ H_{u_1^c u_1^c}^c &= -2(x_1^c)^2, \\ \Delta u_1^c(1) &= \frac{2\psi_1^c x_1^c + 2u_1^c (x_1^c)^2}{\alpha - 1 - 2(x_1^c)^2}, \\ \Delta \epsilon^c &= \beta^c \int_2^3 \psi_1^c \sqrt{x_1^c} dt. \end{aligned}$$

Модель ДНС может быть дополнена третьим мгновенным этапом, не имеющим протяженности во времени и состоящим лишь в передаче информации об окончании второго этапа на верхний уровень. Тогда

$$\begin{aligned} x(3) &= x(2) = x^c(1, 3), \\ \psi(3) &= \psi(2) = -3\alpha((x(2))^2 - 1), \\ \psi_1^c(0, 2) &= \psi_1^c(1, 3) = \psi(2). \end{aligned}$$

На решение примера потребовалось 4 итерации. При этом значение функционала с  $I^0 = -1.05313$  уменьшилось до  $I^4 = -1.99979$ . Изменение функционала по итерациям представлено в таблице 2. Графики управляющих воздействий и состояний процесса на первой и последней итерациях даны на рисунке 3. При расчетах параметр  $\alpha = 0.4$ .

Таблица 2. Изменение значений функционала по итерациям

$k$	$\beta^c(0)$	$\beta^c(1)$	$\epsilon^c(0)$	$\epsilon^c(1)$	$I^k$	$ I^k - I^{k-1} $
0			0.9	0.5	-1.05313	
1	-0.8	-1	0.7689	0.1744	-1.99881	0.94568
2	-0.8	-1	0.7553	0.1514	-1.99974	0.00093
3	-0.8	-0.3	0.7498	0.1464	-1.99978	0.00004
4	-0.4	-0.5	0.7476	0.1373	-1.99979	0.00001

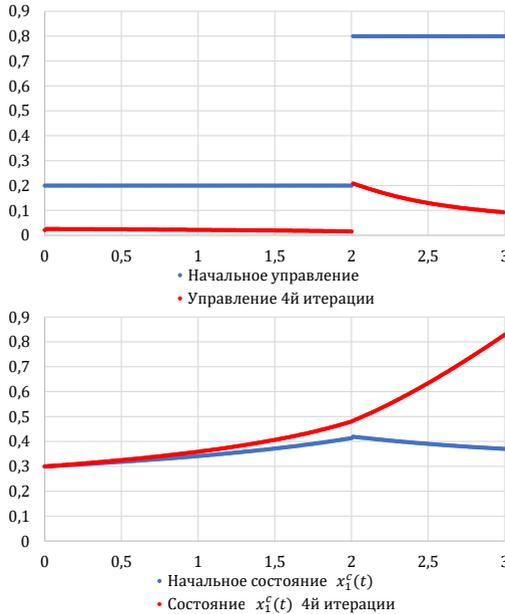


Рисунок 3. Графики управляющих воздействий и состояния  $x_1^c(t)$  процесса по итерациям

### 9. Заключение

В работе рассмотрена одна из гибридных систем: дискретно-непрерывная система (ДНС) с параметрами и промежуточными критериями. Для решения поставленной для нее задачи оптимального управления приведен аналог достаточных условий оптимальности Кротова в виде двух теорем. Этот аналог далее использован для построения метода улучшения управления и параметров. Сформулирован алгоритм. Отдельно приведены уравнения метода для частного случая: квазилинейных ДНС с параметром. Алгоритм апробирован на иллюстративных примерах, приведены результаты расчетов и графики.

## Список литературы

- [1] Емельянов С. В. (ред.) *Теория систем с переменной структурой*.– М.: Наука.– 1970.– 592 с. [↑<sub>126</sub>](#)
- [2] Гурман В. И. *К теории оптимальных дискретных процессов* // Автоматика и телемеханика.– 1973.– № 6.– С. 53–58.  [↑<sub>126</sub>](#)
- [3] Васильев С. Н. *Теория и применение логико-управляемых систем* // Труды 2-ой Международной конференции «Идентификация систем и задачи управления», SICPRO'03 (Москва, 29–31 января 2003), М.: Институт проблем управления им. В.А. Трапезникова РАН.– 2003.– С. 23–52. [↑<sub>126</sub>](#)
- [4] Бортакровский А. С. *Достаточные условия оптимальности детерминированными логико-динамическими системами*, Информатика. Сер. Автоматизация проектирования.– т. 2–3, М.: ВИМИ.– 1992.– С. 72–79. [↑<sub>126</sub>](#)
- [5] Миллер Б. М., Рубинович Е. Я. *Оптимизация динамических систем с импульсными управлениями*.– М.: Наука.– 2005.– ISBN 978-5-9710-5725-3.– 429 с. [↑<sub>126</sub>](#)
- [6] Lygeros J. *Lecture Notes on Hybrid Systems*, Notes for an ENSIETA short course, No 2-6/2/2004.– Rio, Patras: University of Patras.– 2004.– 82 pp.  [↑<sub>126</sub>](#)
- [7] Van der Shaft A. J., Schumacher H. *An Introduction to Hybrid Dynamical Systems*, Lecture Notes in Control and Information Sciences.– Vol. 251.– London: Springer-Verlag.– 2000.– ISBN 978-1-4471-3916-4.– 176 pp.  [↑<sub>126</sub>](#)
- [8] Кротов В. Ф. *Достаточные условия оптимальности для дискретных управляемых систем* // Докл. АН СССР.– 1967.– Т. 172.– № 1.– С. 18–21.  [↑<sub>126</sub>](#), 128, 129
- [9] Гурман В. И., Расина И. В. *Дискретно-непрерывные представления импульсных процессов в управляемых системах* // Автомат. и телемех.– 2012.– № 8.– С. 16–29.  [↑<sub>126</sub>](#)
- [10] Расина И. В. *Итерационные алгоритмы оптимизации дискретно-непрерывных процессов* // Автомат. и телемех.– 2012.– № 10.– С. 3–17.  [↑<sub>126</sub>](#)
- [11] Расина И. В. *Иерархические модели управления системами неоднородной структуры*.– М.: Физматлит.– 2014.– ISBN 978-5-94052-238-6.– 160 с. [↑<sub>126</sub>](#)
- [12] Расина И. В. *Дискретно-непрерывные системы с промежуточными критериями* // Материалы XX Юбилейной Международной конференции по вычислительной механике и современным прикладным программным системам, ВМСППС'2017 (24–31 мая 2017 г., Алушта, Россия), М.: Изд-во МАИ.– 2017.– ISBN 978-5-4316-0401-0.– С. 699–701. [↑<sub>126</sub>](#), 130
- [13] Батурин В. А., Урбанович Д. Е. *Приближенные методы оптимального управления, основанные на принципе расширения*.– Новосибирск: Наука.– 1997.– ISBN 5-02-031440-2.– 175 с. [↑<sub>126</sub>](#)
- [14] Расина И. В., Гусева И. С. *Дискретно-непрерывные системы с параметрами: метод улучшения управления и параметров* // Известия Иркутского государственного университета. Серия Математика.– 2022.– Т. 39.– С. 34–50.

- [15] Габасов Р., Калинин А. И., Кириллова Ф. М., Лавринович Л. И. *К асимптотическим методам оптимизации квазилинейных систем управления* // Тр. ИММ УрО РАН.– 2019.– Т. **25**.– № 3.– С. 62–72.   \*  ↑127
- [16] Кротов В. Ф., Гурман В. И. *Методы и задачи оптимального управления*.– М.: Наука.– 1973.– 448 с. ↑129, 130
- [17] Гурман В. И. *Принцип расширения в задачах управления*.– М.: Наука.– 1985.– 228 с. ↑130, 131
- [18] Гурман В. И. *Абстрактные задачи оптимизации и улучшения* // Программные системы: теория и приложения.– 2011.– № 5(9).– С. 21–29.  ↑131
- [19] Гурман В. И., Расина И. В. *О практических приложениях достаточных условий сильного относительного минимума* // Автомат. и телемех.– 1979.– № 10.– С. 12–18.  ↑131, 132

Поступила в редакцию 27.01.2022;  
 одобрена после рецензирования 16.03.2023;  
 принята к публикации 17.03.2023;  
 опубликована онлайн 20.03.2023.

Рекомендовал к публикации

*д.т.н. А. М. Цирлин*

## Информация об авторах:



### Ирина Викторовна Расина

д.ф.-м.н., гл. научн. сотр. Исследовательского центра системного анализа Института программных систем имени А. К. Айламазяна РАН и Федерального исследовательского центра «Информатика и управление» РАН. Специалист в области моделирования и управления гибридными системами, автор и соавтор более 130 статей и 5 монографий

 0000-0001-8939-2968

**e-mail:** [irinasina@gmail.com](mailto:irinasina@gmail.com)



### Ирина Сергеевна Гусева

к.ф.-м.н., ст.преп. Бурятского государственного университета имени Доржи Банзарова, РФ, респ. Бурятия, г. Улан-Удэ; автор и соавтор более 20 публикаций; область интересов – оптимальное управление, численные методы

 0000-0001-8720-3676

**e-mail:** [gulina.ig@gmail.com](mailto:gulina.ig@gmail.com)

*Все авторы сделали эквивалентный вклад в подготовку публикации.  
 Авторы заявляют об отсутствии конфликта интересов.*



## About One Class of Discrete-Continuous Systems with Parameters

Irina Viktorovna **Rasina**<sup>1</sup>, Irina Sergeevna **Guseva**<sup>2</sup>

<sup>1</sup> Ailamazyan Program Systems Institute of RAS, Ves'kovo, Russia

<sup>1</sup> Federal Research Center "Computer Science and Control" of RAS, Moscow, Russia

<sup>2</sup> Buryat State University, Ulan-Ude, Russia

<sup>1✉</sup> [irinarasina@gmail.com](mailto:irinarasina@gmail.com)

**Abstract.** The study focuses on a special case of a hybrid system: discrete-continuous systems (DCS) with parameters and intermediate criteria. Such systems are two-level. The parameters are included only in continuous systems operating alternately at the lower level. The upper level is described by a discrete process and plays a connecting role for all the lower-level systems. The upper level also determines the policy of interaction of lower-level systems and provides minimization of functionality. The authors formulate an analogue of sufficient Krotov optimality conditions and construct a method for improving control and parameters. The paper contains an illustrative example. Based on the general conditions obtained, we have researched a special case: quasilinear DNS. (*In Russian*).

**Key words and phrases:** discrete-continuous systems with parameters, intermediate criteria, optimal control, quasilinear discrete-continuous systems

2020 *Mathematics Subject Classification:* 49M99; 49K99

**Acknowledgments:** This work was supported by the Russian Science Foundation grant No. 21-11-00202

**For citation:** Irina V. Rasina, Irina S. Guseva. *About One Class of Discrete-Continuous Systems with Parameters*. Program Systems: Theory and Applications, 2023, **14**:1(56), pp. 125–148. (*In Russ.*).

[https://psta.psiras.ru/read/psta2023\\_1\\_125-148.pdf](https://psta.psiras.ru/read/psta2023_1_125-148.pdf)

## References

- [1] S. V. (red.) Emel'yanov. *Theory of Systems with Variable Structures*, Nauka, M., 1970 (in Russian), 592 pp.
- [2] V. I. Gurman. "Theory of optimum discrete processes", *Autom. Remote Control*, **34**:7 (1973), pp. 1082–1087. 
- [3] S. N. Vasil'ev. "Theory and application of logic-based controlled systems", *Trudy 2-oy Mezhdunarodnoj konferencii "Identifikaciya sistem i zadachi upravleniya"*, SICPRO'03 (Moskva, 29–31 yanvarya 2003), Institut problem upravleniya im. V. A. Trapeznikova RAN, M., 2003, pp. 23–52 (in Russian).
- [4] A. S. Bortakovskij. "Sufficient optimality conditions for control of deterministic logical-dynamic systems", *Informatika. Ser. Avtomatizaciya proektirovaniya*, vol. **2–3**, VIMI, M., 1992, pp. 72–79 (in Russian).
- [5] B. M. Miller, E. Ya. Rubinovich. *Optimization of the Dynamic Systems with Pulse Controls*, Nauka, M., 2005, ISBN 978-5-9710-5725-3 (in Russian), 429 pp.
- [6] J. Lygeros. *Lecture Notes on Hybrid Systems*, Notes for an ENSIETA short course, No 2-6/2/2004, University of Patras, Rio, Patras, 2004, 82 pp. 
- [7] der Shaft A. J. Van, H. Schumacher. *An Introduction to Hybrid Dynamical Systems*, Lecture Notes in Control and Information Sciences, vol. **251**, Springer-Verlag, London, 2000, ISBN 978-1-4471-3916-4, 176 pp. 
- [8] V. F. Krotov. "Sufficient conditions for the optimality of discrete control systems", *Dokl. AN SSSR*, **172**:1 (1967), pp. 18–21 (in Russian). 
- [9] V. I. Gurman, I. V. Rasina. "Discrete-continuous representations of impulsive processes in the controllable systems", *Autom. Remote Control*, **73**:8 (2012), pp. 1290–1300.
- [10] I. V. Rasina. "Iterative optimization algorithms for discrete-continuous processes", *Autom. Remote Control*, **73**:10 (2012), pp. 1591–1603. 
- [11] I. V. Rasina. *Hierarchical models of control systems of heterogeneous structure*, Fizmatlit, M., 2014, ISBN 978-5-94052-238-6 (in Russian), 160 pp.
- [12] I. V. Rasina. "Discrete-continuous systems with intermediate criteria", *Materialy XX Yubilejnoj Mezhdunarodnoj konferencii po vychislitel'noj mexanike i sovremennym prikladnym programmnym sistemam*, VMSPPS'2017 (24–31 maya 2017 g., Alushta, Rossiya), Izd-vo MAI, M., 2017, ISBN 978-5-4316-0401-0, pp. 699–701 (in Russian). 
- [13] V. A. Baturin, D. E. Urbanovich. *Approximate Methods of Optimal Control Based on the Extension Principle*, Nauka, Novosibirsk, 1997, ISBN 5-02-031440-2 (in Russian), 175 pp.
- [14] I. V. Rasina, I. S. Guseva. "Discrete-Continuous Systems with Parameters: Method for Improving Control and Parameters", *Izvestiya Irkutskogo gosudarstvennogo universiteta. Seriya Matematika*, **39** (2022), pp. 34–50 (in Russian). 
- [15] R. Gabasov, A. I. Kalinin, F. M. Kirillova, L. I. Lavrinovich. "On asymptotic optimization methods for quasilinear control systems", *Tr. IMM UrO RAN*, **25**:3 (2019), pp. 62–72 (in Russian).  

- [16] V. F. Krotov, V. I. Gurman. *Methods and Problems of Optimal Control*, Nauka, M., 1973 (in Russian), 448 pp.
- [17] V. I. Gurman. *Extension Principle In Control Problems*, Nauka, M., 1985 (in Russian), 228 pp.
- [18] V. I. Gurman. “Abstract problems of optimization and improvement”, *Program Systems: Theory and Applications*, 2011, no. 5(9), pp. 21–29 (in Russian). 
- [19] V. I. Gurman, I. V. Rasina. “On practical applications of conditions sufficient for a strong relative minimum”, *Autom. Remote Control*, **40**:10 (1980), pp. 1410–1415.