

Программные системы: **теория и приложения**

УДК 004.932.75'1, 004.89

 10.25209/2079-3316-2024-15-1-3-30


Распознавание кадастровых координат с использованием свёрточно-рекуррентных нейронных сетей

Игорь Викторович **Винокуров** 

Финансовый Университет при Правительстве Российской Федерации, Москва, Россия

 igvvinokurov@fa.ru

Аннотация. В статье исследуется применение свёрточно-рекуррентных нейронных сетей (CRNN) для распознавания изображений кадастровых координат объектов на отсканированных документах ППК «Роскадастр». Комбинированная архитектура CRNN, объединяющая свёрточные нейронные сети (CNN) и рекуррентные нейронные сети (RNN), позволяет использовать преимущества каждой из них для обработки изображений и распознавания содержащихся в них непрерывных цифровых последовательностей. При проведении экспериментальных исследований были сформированы изображения, состоящие из заданного количества цифр, построена и исследована CRNN модель. Формирование изображений цифровых последовательностей заключалось в предобработке и конкатенации изображений образующих их цифр из собственного набора данных. Анализ значений функции потерь и метрик Accuracy, Character Error Rate (CER) и Word Error Rate (WER) показал, что использование предложенной CRNN модели позволяет достичь высокой точности распознавания кадастровых координат на их отсканированных изображениях.

Ключевые слова и фразы: свёрточно-рекуррентная нейронная сеть, CRNN, распознавание изображений, цифровые последовательности, глубокое обучение, Keras, Python

Для цитирования: Винокуров И. В. Распознавание кадастровых координат с использованием свёрточно-рекуррентных нейронных сетей // Программные системы: теория и приложения. 2024. Т. 15. № 1(60). С. 3–30. https://psta.psisiras.ru/read/psta2024_1_3-30.pdf

Введение

Распознавание буквенно-цифровых последовательностей на изображениях является значимой задачей в области компьютерного зрения и обработки изображений. Эффективное решение этой задачи имеет большое значение для автоматизации и оптимизации различных процессов, связанных с идентификацией и классификацией объектов.

В последние годы с использованием нейронных сетей глубокого обучения, особенно CNN, были получены впечатляющие результаты в распознавании и классификации изображений. Однако классические CNN ориентированы на выявление признаков входных данных, что ограничивает их применимость для распознавания последовательностей переменной длины. С целью преодоления этого ограничения была разработана архитектура CRNN, которая объединяет преимущества свёрточных и рекуррентных нейронных сетей. Основной особенностью архитектуры CRNN является совмещение свёрточных слоев CNN для извлечения локальных и пространственных признаков из изображений и рекуррентных слоёв RNN для учёта контекста и последовательности информации. Свёрточные слои позволяют обнаружить важные особенности изображений на разных уровнях абстракции, в то время как рекуррентные слои моделируют зависимости и последовательность входных данных [1].

В наиболее общем случае архитектура CRNN состоит из трёх основных компонентов – свёрточного, рекуррентного и компонента классификации. Свёрточный компонент содержит несколько свёрточных слоев, выполняющих обнаружение и извлечение признаков из изображений. Рекуррентный компонент включает рекуррентные слои долгой краткосрочной памяти (LSTM) [2] или управляемые рекуррентные блоки (GRU) [3], позволяющие учитывать контекст и последовательность входных данных. Компонент классификации выполняет распознавание и классификацию последовательностей на основе предсказаний рекуррентного компонента.

В данной статье проводится исследование эффективности использования CRNN для распознавания изображений цифровых последовательностей переменной длины. В разделе 1 осуществляется обоснование необходимости исследований и постановка задачи. Раздел 2 посвящён обзору и анализу работ по использованию сетей CRNN для распознавания текста и цифровых последовательностей. Создание набора данных для обучения модели описано в разделе 3. Формирование и исследование CRNN модели для распознавания цифровых последовательностей на изображениях приведено в разделе 4. В заключении приведены выводы по результатам проведённых исследований.

1. Постановка цели и задач исследования

Целью исследования является разработка CRNN модели, способной реализовать приемлемую точность распознавания кадастровых координат на отсканированных документах ППК «Роскадастр». В [4], [5] и [6] описаны реализованные в информационной системе (ИС) этой организации подходы к решению задачи преобразования изображений кадастровых координат в их текстовые аналоги с использованием моделей CNN. В [6] показывается эффективность использования CNN для последовательностей, состоящих не более чем из 4-х цифр. По результатам проведённых в [6] исследований, можно сделать вывод, что при распознавании большего количества цифр, структура CNN усложняется, качество распознавания при этом либо остаётся прежним, либо увеличивается незначительно. Учитывая, что максимальное количество символьно-цифровых элементов в кадастровых координатах может превышать 10, одним из целесообразных и эффективных средств решения задачи их распознавания может являться использование CRNN моделей. Поставленная в работе цель может быть достигнута за счёт решения следующих основных задач:

- (1) Формирование набора данных, заключающееся в подготовке набора изображений элементов кадастровых номеров и их аннотирование (сопоставление с кадастровым номером); формирование изображений кадастровых координат на заданное количество их цифровых элементов; разделении набора данных на выборки для обучения и валидации.
- (2) Формирование CRNN модели – выбор свёрточных и рекуррентных слоёв для извлечения признаков и распознавания цифровых последовательностей соответственно.
- (3) Обучение модели и анализ значений функций потерь и метрик точности, CER и WER.

2. Анализ основных работ по распознаванию буквенно-цифровых последовательностей

Впервые CRNN модель, объединяющая свёрточные и рекуррентные слои для обработки изображений с текстовыми последовательностями описана в [1]. Достоинством модели является сочетание свёрточных и рекуррентных слоёв, позволяющее выявлять как локальные, так и глобальные зависимости в изображениях, содержащих последовательности, и достаточно эффективно реализовывать их распознавание. К недостаткам предложенной модели могут быть отнесены достаточно большие объёмы данных для обучения и значительное время распознавания длинных последовательностей.

В [7] описана CRNN модель, которая может распознавать неизвестные ей слова, используя значимую контекстную информацию. Модель является устойчивой к различным искажениям изображения, не зависит от заранее определённого словаря и может обрабатывать произвольные предложения. Недостаток модели – плохое распознавание текста с низкой контрастностью, нечёткими границами и искажениями, что требует дополнительных методов предварительной обработки изображения.

Модель из [8] предназначена для распознавания текста на изображениях с искажением перспективы, изогнутым расположением символов и т. д. Предложенная CRNN модель способна обеспечить приемлемую читаемость и распознаваемость искажённого текста и превосходит аналогичные модели при распознавании текста с разными значениями входного шума и наклона. Кроме этого, модель демонстрирует высокую точность и производительность при обучении на больших наборах данных. К недостатку предложенной модели может быть отнесено плохое распознавание текста, если шум или деформация сильно изменяют форму символов, что особенно сильно заметно для текста с малым шрифтом или низким разрешением.

В статье [9] авторы применяют CRNN для распознавания текста на изображениях, сосредотачиваясь на сложных случаях, таких как сцены с плохим освещением или низким разрешением. Они предлагают модель, которая использует свёрточные слои для извлечения признаков из изображений и рекуррентные слои для моделирования последовательностей из этих символов. В экспериментах демонстрируется высокая точность распознавания текста на различных изображениях текстовых последовательностей.

Модель для многоуровневого распознавание рукописного текста на изображениях приведена в [10]. На первом уровне для распознавания слов, которые часто встречаются в тексте, используется CNN. Если слово не распознается этой моделью, оно переходит на второй уровень, на котором используется полностью свёрточной сети (FCN). Экспериментальное исследование модели проводилось с использованием NIST19 в качестве набора данных для обучения и рукописного текста в качестве набора тестовых данных и показало вполне приемлемый результат распознавания.

В [11] представлены две модели для распознавания последовательности цифр. В первой кодером и декодером последовательностей являются CNN и LSTM. Во второй – гистограмма ориентированного градиента (HOG) и параллельные полносвязные (Dense) слои соответственно. Обучение и тестирование осуществлялось на наборе данных Street View Number House

(SVHN). В результате проведённых исследований показано преимущество CNN в отношении кодирования изображений и преимущество LSTM в предсказании последовательностей.

Сеть глубокого обучения DIGI-Net, которая способна изучать общие характеристики трех различных форматов цифр (рукописные, естественные изображения, печатный шрифт) и распознавать их описана в [12]. Эксперименты, проведённые на наборах данных MNIST, CVL и Chars74K, продемонстрировали высокую точность распознавания непрерывных цифровых последовательностей.

Предложенная в [13] CRNN модель имеет свёрточный слой, слой слияния признаков, рекуррентный слой и слой транскрипции. Свёрточный слой, используемый для извлечения признаков, формирует два результата для входного текстового изображения. Слой объединения объектов объединяет результаты работы свёрточного слоя в один, из которого рекуррентный слой извлекает последовательности. Окончательный результат выводит слой транскрипции. Предложенная модель за счёт слияния признаков реализует лучшую точность распознавания текста на наборах текстовых данных Street View Text (SVT), IIT-5K, ICDAR2003 и ICDAR2013.

В [14] для распознавания рукописных цифр предлагается использовать гибридную архитектуру – визуальные трансформеры (ViTs) и многослойные перцептроны (MLP). Проведённые исследования на наборах данных EMNIST и DIDA показали хорошую точность распознавания машинописных цифровых, в том числе, и на зашумлённых изображениях.

В [15] предлагается симметричная многомасштабная архитектура под названием Circular Dilated Convolutional Neural Network (CDIL-CNN), где каждый элемент текущего уровня имеет равные шансы получить информацию от других элементов с предыдущих уровней. Предлагаемая CRNN модель позволяет сформировать логиты (logits) классификации для всех элементов, в результате чего становится возможным применение простого ансамблевого обучения для принятия лучшего решения. По результатам тестирования CDIL-CNN на длинных последовательных наборах данных показано, что CDIL-CNN позволяет получить приемлемый по точности результат распознавания.

Лучший подход к формированию моделей для распознавания последовательностей, с точки зрения автора этой работы, приведён в [16] и [17]. Для распознавания текстовых последовательностей предлагается использовать кодер в виде CNN и декодер в виде двунаправленной

долгосрочной краткосрочной памяти (BSTM) с использованием коннекционистской временной классификации (СТС). СТС является алгоритмом, используемым для обучения RNN на последовательностях с переменной длиной и сопоставления их с соответствующими метками. В задачах распознавания текста и цифровых последовательностей, СТС может справиться с проблемой их переменной длины. Он позволяет модели прогнозировать переменное количество букв или цифр в последовательности без их предварительного разделения или выравнивания. Алгоритм СТС вычисляет вероятность выходной последовательности и рекуррентно обновляет веса модели на основе разницы между прогнозами и метками последовательностей. Эффективность предложенного подхода показана на собственном наборе данных в [16] и на наборах данных арабских букв MADCAT, ANTID/MW и IFN/ENIT в [17].

3. Формирование набора данных

Для обучения CRNN модели и исследования её работы был сформирован собственный набор данных.

На первом этапе, по аналогии с [4], формировались чёрно-белые изображения элементов цифровых последовательностей с использованием основных шрифтов документов ППК «Роскадастр». Количество классов изображений выбрано равным 12 – 10 классов для цифр от 0 до 9, 1 класс для символов-разделителей «.» и «,» и ещё один на отсутствие символа в последовательности. Значения 2-х последних классов выбраны равными 10 и 11 соответственно. Для каждого класса изображений было сформировано по 10 и 5 изображений размером 20×25 пикселей для обучения и валидации модели соответственно.

На втором этапе формировались изображения кадастровых координат, состоящие из 2-х цифр в дробной части и от 4-х до 7-и в целой. Одновременно с формированием изображений формировались и их СТС-метки. Все изображения в наборе данных приводились к одному размеру 200×32 пикселей. Пример сформированных изображений кадастровых координат и соответствующих им СТС-меток приведён на рисунке 1.

Сформированный таким образом набор данных приведён в таблице 1 и состоит из 24240 и 12240 изображений кадастровых координат и их меток для обучения и валидации соответственно.

4. Формирование и исследование CRNN модели

Формирование модели осуществлялось с использованием библиотеки Keras. Все слои этой модели, помимо основных слоёв CRNN – свёрточных

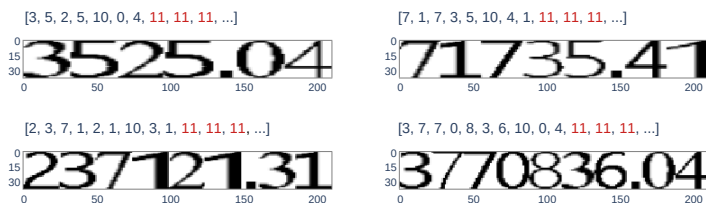


Рисунок 1. Примеры изображений кадастровых координат. Сверху указаны их CTC-метки

Таблица 1. Количество цифровых последовательностей для обучения и валидации

Цифр в последовательности	Последовательностей для обучения	Последовательностей для валидации
6	4848	2448
7	5656	2856
8	6464	3264
9	7272	3672

и рекуррентных (Conv2D и Bidirectional соответственно), приведены в таблице 2.

Субдискретизирующий слой MaxPooling2D уменьшает размерность пространства признаков, выделяя самые значимые из них.

Слой BatchNormalization нормализует данные по мини-пакетам, что позволяет ускорить сходимость обучения и уменьшить вероятность переобучения. Также он помогает стабилизировать распределение активаций между слоями.

Слой Dropout является регуляризатором, его цель – снижение переобучения за счёт предотвращения активации случайно выбранных нейронов. Это заставляет модель обучаться более устойчивыми признаками и уменьшает вклад каждого отдельного нейрона.

Слой Activation (активационная функция) применяет функцию активации к выводу предыдущего слоя. В данной модели это ReLU, которая активирует нейроны только при положительных значениях, Tanh – при положительных и отрицательных значениях и Softmax, генерирующая вероятности для разных классов.

Полносвязный слой Dense объединяет все выходы предыдущего слоя и применяет линейные преобразования для получения финального вывода модели. Он связывает выходные сигналы всех нейронов предыдущего

ТАБЛИЦА 2. Слои CRNN модели

Тип слоя	Функция активации	Количество фильтров	Входной вектор
InputLayer	–	–	[(None, 200, 32, 1)]
Conv2D	–	32	(None, 200, 32, 32)
BatchNormalization	–	–	(None, 200, 32, 32)
Activation	ReLu	–	(None, 200, 32, 32)
MaxPooling2D	–	–	(None, 100, 16, 32)
Conv2D	–	64	(None, 100, 16, 64)
BatchNormalization	–	–	(None, 100, 16, 64)
Activation	ReLu	–	(None, 100, 16, 64)
MaxPooling2D	–	–	(None, 50, 8, 64)
Dropout	–	–	(None, 50, 8, 64)
Conv2D	–	128	(None, 50, 8, 128)
BatchNormalization	–	–	(None, 50, 8, 128)
Activation	ReLu	–	(None, 50, 8, 128)
MaxPooling2D	–	–	(None, 50, 4, 128)
Dropout	–	–	(None, 50, 4, 128)
Conv2D	–	256	(None, 50, 4, 256)
BatchNormalization	–	–	(None, 50, 4, 256)
Activation	ReLu	–	(None, 50, 4, 256)
MaxPooling2D	–	–	(None, 50, 2, 256)
Dropout	–	–	(None, 50, 2, 256)
Reshape	–	–	(None, 32, 800)
Dense	–	–	(None, 32, 25)
Bidirectional	Tanh	–	(None, 32, 320)
Bidirectional	Tanh	–	(None, 32, 320)
Dense	–	–	(None, 32, 12)
Activation	Softmax	–	(None, 32, 12)

слоя с каждым нейроном в текущем слое и является основным слоем классификации в нейронных сетях.

Слой **Lambda** позволяет задать собственную лямбда-функцию для нестандартного преобразования данных, например, изменения их размерности. В сформированной модели он используется совместно с **ground-truth** метками, используемыми для задач распознавания непрерывных последовательностей. На этапе обучения эти метки подвергаются согласованию и сравниваются с истинными метками, в результате чего возникает ошибка, используемая для оптимизации модели.

Структура CRNN модели приведена на рисунке 2. Для её обучения был использован метод обратного распространения ошибки с оптимизатором **Adam**.

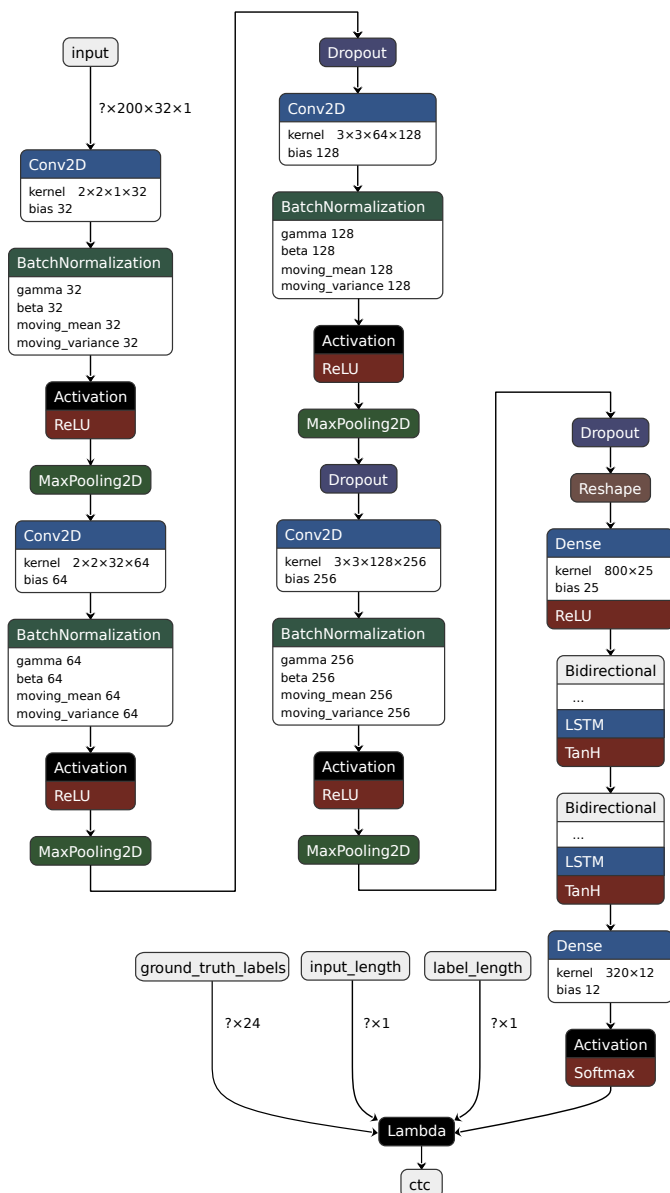


Рисунок 2. CRNN модель

Для оценки качества работы модели и её способности решать задачу осуществлялось вычисление значений функции потерь и метрик точности, CER и WER. Функция потерь (**Loss**) измеряет разницу между фактическими (истинными) значениями меток и значениями, предсказанными моделью. На рисунке 3 приведены потери модели для 9 эпох обучения. Такое количество эпох обучения найдено экспериментальным путём и является оптимальным.

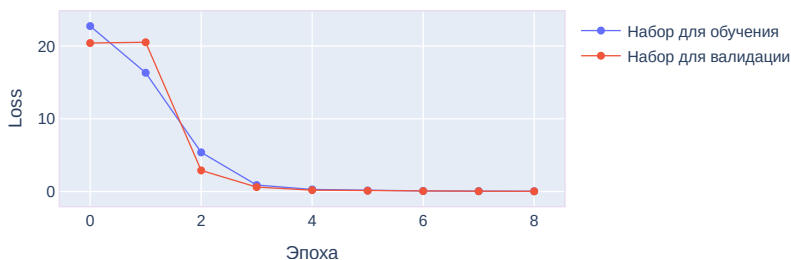


Рисунок 3. Потери (Loss) модели

Метрика точности (**Accuracy**) определяет долю правильных предсказаний, сделанных моделью, по отношению к общему количеству примеров. Она позволяет оценить, насколько хорошо модель может классифицировать или предсказывать правильный класс или значение для данного набора данных, рисунок 4.

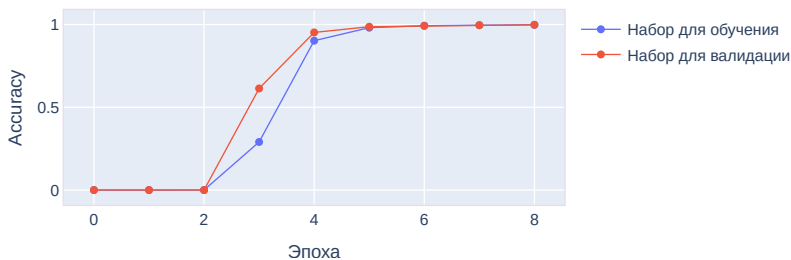


Рисунок 4. Точность (Accuracy) модели

Числовые значения функций потерь и метрик точности для наборов обучения и валидации на каждой из эпох обучения приведены в таблице 3.

Вычисление значений функции потерь и метрик точности является обычным подходом к оценке качества нейросетевых моделей. Для моделей, ориентированных на распознавание последовательностей, вычисляются ещё две – Character Error Rate (CER) и Word Error Rate (WER). Обе метрики используются для сравнительной оценки различных систем распознавания и анализа их точности.

ТАБЛИЦА 3. Числовые значения Loss и Accuracy

	Loss		Accuracy	
Номер эпохи	Набор для обучения	Набор для валидации	Набор для обучения	Набор для валидации
Epoch	Train set	Validation set	Train set	Validation set
0	22.7430438	20.4115753	0.0	0.0
1	16.3242683	20.5168876	0.0	0.0
2	5.3856644	2.9053363	0.0	0.0000817
3	0.8966413	0.6119849	0.2904703	0.6136437
4	0.2865071	0.1913659	0.9022276	0.9521241
5	0.1421806	0.1023578	0.9807755	0.9866012
6	0.0855829	0.0680020	0.9928630	0.9919934
7	0.0573494	0.0432046	0.9958745	0.9959967
8	0.0411685	0.0314824	0.9980198	0.9982843

CER позволяет оценить точность распознавания моделью отдельных символов. Зависимости значений этой метрики для набора данных и набора для валидации от номера эпохи обучения показано на рисунке 5.

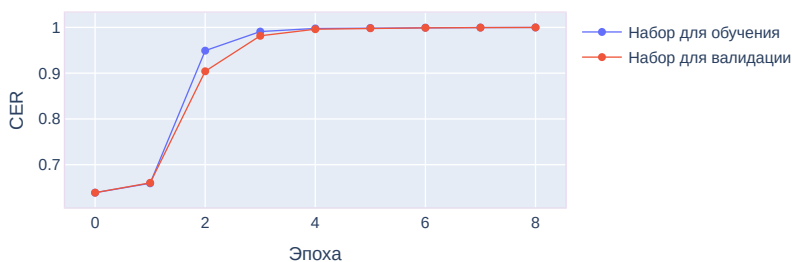


Рисунок 5. Точность распознавания символов (CER)

Точность распознавания моделью целых слов позволяет оценить метрика WER. Зависимости значений метрики для набора данных и набора для валидации от номера эпохи обучения показано на рисунке 6.

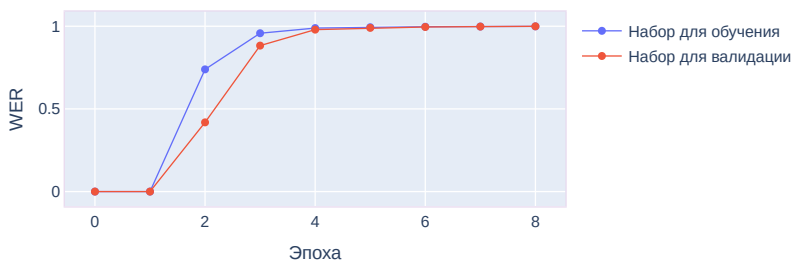


Рисунок 6. Точность распознавания слов (WER)

Числовые значения метрик CER и WER для наборов обучения и

валидации на каждой из эпох обучения приведены в таблице 4.

Таблица 4. Значения метрик CER и WER

Номер эпохи	CER		WER	
	Набор для обучения	Набор для валидации	Набор для обучения	Набор для валидации
0	0.6388888	0.6388888	0.0	0.0
1	0.6594093	0.6603758	0.0	0.0
2	0.9491560	20.9041156	0.7390676	0.4186274
3	0.9908536	0.9816210	0.9576320	0.8828431
4	0.9976399	0.9960001	0.9887788	0.9794117
5	0.9987451	0.9981072	0.9940182	0.9900327
6	0.9993949	0.9989787	0.9970297	0.9953431
7	0.9996046	0.9995132	0.9981435	0.9977942
8	0.999804	0.9998400	0.9992161	0.9994281

Результаты распознавания моделью нескольких кадастровых координат на 6,7,8 и 9 цифр соответственно из тестового набора данных приведены на рисунке 7.

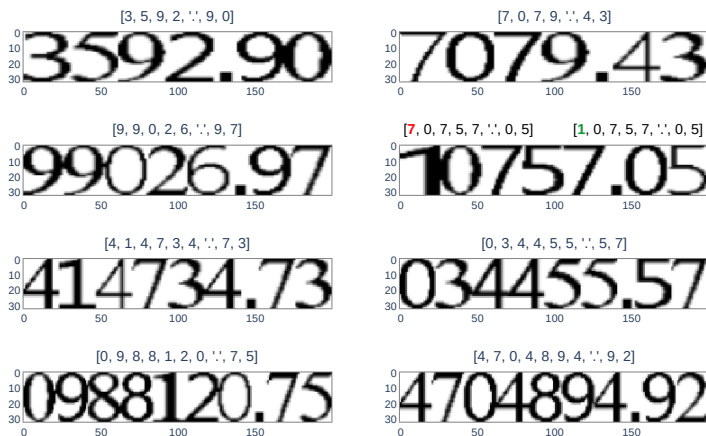


Рисунок 7. Результаты распознавания кадастровых координат. Одна из координат распознана с ошибкой

Предложенная CRNN модель реализована в ИС ППК «Роскадастр» и, как показали результаты её экспериментального исследования, точность распознавания отдельных символов и кадастровых координат в целом составила 99.98% и 99.94% соответственно. Изображения кадастровых координат выделялись из отсканированного документа по координатам, формируемым подсистемой контуризации этой ИС. Общие принципы работы этой подсистемы описаны в [4].

Заключение

В статье проведены исследования применения архитектуры CRNN для задачи распознавания изображений кадастровых координат. По результа-

там обучения модели были построены графики функции потерь (Loss) и точности (Accuracy). Графики показали, что модель успешно сходится и способна достичь высокой точности распознавания кадастровых координат. Для оценки качества работы модели были использованы метрики CER (Character Error Rate) и WER (Word Error Rate), которые позволили измерить процент ошибок на уровне символов и слов соответственно. По результатам экспериментальных исследований можно сделать вывод, что модель способна распознать кадастровые координаты с высоким уровнем точности и минимальным количеством ошибок. Как следствие, применение CRNN модели позволит значительно улучшить эффективность и достоверность геопространственных анализов и принятия решений. Дальнейшие исследования могут быть направлены на расширение набора данных, включение разных типов шрифтов и стилей, обучение модели на более разнообразных данных и исследование эффективности CRNN для других задач распознавания и классификации. Также возможно применение дополнительных методов предобработки данных или аугментации для улучшения точности модели.

Список литературы

- [1] Shi B., Bai X., Yao C. *An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence.– 2017.– Vol. **39**.– No. 11.– Pp. 2298–2304. 1507.05717 [cs.CV] ↑4, 5
- [2] Hochreiter S., Schmidhuber J. *Long short-term memory* // Neural Computation.– 1997.– Vol. **9**.– No. 8.– Pp. 1735–1780. ↑4
- [3] Chung J., Gulcehre C., Cho K., Bengio Y. *Gated feedback recurrent neural networks* // Proceedings of Machine Learning Research.– 2015.– Vol. **37**.– Pp. 2067–2075. 1502.02367 [cs.NE] ↑4
- [4] 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(54).– С. 29–43. ↑5, 8, 14
- [5] Винокуров И. В. *Распознавание табличной информации с использованием свёрточных нейронных сетей* // Программные системы: теория и приложения.– 2023.– Т. **14**.– № 1(56).– С. 3–30. ↑5
- [6] Винокуров И. В. *Распознавание цифровых последовательностей с использованием свёрточных нейронных сетей* // Программные системы: теория и приложения.– 2023.– Т. **14**.– № 3(58).– С. 3–36. ↑5
- [7] He P., Huang W., Qiao Y., Change Loy C., Tang X. *Reading scene text in deep convolutional sequences*, AAAI'16: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (Phoenix, Arizona, USA, February 12–17, 2016) // Proceedings of the AAAI Conference on Artificial Intelligence.– 2016.– Vol. **30** 1.– Pp. 3501–3508. ↑6
- [8] Shi B., Wang X., Lv P., Yao C., Bai X. *Robust scene text recognition with automatic rectification*, 2016 IEEE Conference on Computer Vision and Pattern Recognition

- (CVPR) (Las Vegas, NV, USA, June 27–30, 2016).– 2016.– Pp. 4168–4176. [doi](#) [arXiv:1603.03915 \[cs.CV\]](#) ↑6
- [9] Yin F., Wu Y.-C., Zhang X.-Y., Liu C.-L. *Scene text recognition with sliding convolutional character models*.– 2017.– 10 pp. [arXiv:1709.01727 \[cs.CV\]](#) ↑6
- [10] Nirmalasari D. A., Suciati N., Navastara D. A. *Handwritten text recognition using fully convolutional network* // IOP Conference Series: Materials Science and Engineering.– 2021.– Vol. **1077**.– No. 1.– id. 012030.– 9 pp. [doi](#) ↑6
- [11] Liu X., Deng Y., Sun Y., Zhou Y. *Multi-digit recognition with convolutional neural network and long short-term memory* // 2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD) (Huangshan, China, July 28–30, 2018).– IEEE.– 2018.– Pp. 1187–1192. [doi](#) ↑6
- [12] Madakannu A., Selvaraj A. *DIGI-Net: a deep convolutional neural network for multi-format digit recognition* // Neural Computing and Applications.– 2020.– Vol. **32**.– Pp. 11373–11383. [doi](#) ↑7
- [13] Zou L., He Z., Wang K., Wu Z., Wang Y., Zhang G., Wang X. *Text recognition model based on multi-scale fusion CRNN* // Sensors.– 2023.– Vol. **32**.– No. 16.– id. 7034.– 18 pp. [doi](#) ↑7
- [14] Agrawal V., Jagtap J. *Convolutional vision transformer for handwritten digit recognition*.– Research Square.– 2022.– 11 pp. [doi](#) ↑7
- [15] Cheng L., Khalitov R., Yu T., Yang Z. *Classification of long sequential data using circular dilated convolutional neural networks*.– 2022.– 16 pp. [arXiv:2201.02143 \[cs.LG\]](#) ↑7
- [16] Bhat R. S. *Text recognition with CRNN-CTC network*.– W&B Fully Connected.– 2022. [URL](#) ↑7, 8
- [17] Khamekhem S., Sourour A., Kessentini Y. *Domain and writer adaptation of offline Arabic handwriting recognition using deep neural networks* // Neural Computing and Applications.– 2022.– Vol. **34**.– Pp. 2055–2071. [doi](#) ↑7, 8

Поступила в редакцию 29.09.2023;
одобрена после рецензирования 27.11.2023;
принята к публикации 27.11.2023;
опубликована онлайн 11.03.2024.

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

Автор заявляет об отсутствии конфликта интересов.



Recognition of cadastral coordinates using convolutional recurrent neural networks

Igor Victorovich **Vinokurov**

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

 iguvinokurov@fa.ru

Abstract. The article examines the use of convolutional recurrent neural networks (CRNN) for recognizing images of cadastral coordinates of objects on scanned documents of the «Roskadastr» PLC. The combined CRNN architecture, combining convolutional neural networks (CNN) and recurrent neural networks (RNN), allows you to take advantage of each of them for image processing and recognition of continuous digital sequences contained in them. During experimental studies, images consisting of a given number of digits were generated, and a CRNN model was built and studied. The formation of images of digital sequences consisted of preprocessing and concatenation of images of the digits forming them from one's own data set. Analysis of the values of the loss function and Accuracy, Character Error Rate (CER), and Word Error Rate (WER) metrics showed that the use of the proposed CRNN model makes it possible to achieve high accuracy in recognizing cadastral coordinates in their scanned images.

Key words and phrases: convolutional recurrent neural network, CRNN, image recognition, digital sequences, deep learning, Keras, Python

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

For citation: Igor V. Vinokurov. *Recognition of cadastral coordinates using convolutional recurrent neural networks*. Program Systems: Theory and Applications, 2024, **15**:1(60), pp. 3–30. https://psta.psiras.ru/read/psta2024_1_3-30.pdf

Introduction

Recognition of alphanumeric sequences in images is a significant problem in the field of computer vision and image processing. An effective solution to this problem is of great importance for the automation and optimization of various processes associated with the identification and classification of objects.

In recent years, impressive results in image recognition and classification have been achieved using deep learning neural networks, especially CNNs. However, classical CNNs are focused on identifying features of the input data, which limits their applicability for recognizing sequences of variable length. To overcome this limitation, the CRNN architecture was developed, which combines the advantages of convolutional and recurrent neural networks. The main feature of the CRNN architecture is the combination of CNN convolutional layers to extract local and spatial features from images and RNN recurrent layers to take into account context and sequence information. Convolutional layers allow you to discover important features of images at different levels of abstraction, while recurrent layers model the dependencies and sequence of input data [1].

In the most general case, the CRNN architecture consists of three main components: convolutional, recurrent, and classification components. The convolutional component contains several convolutional layers that perform detection and feature extraction from images. The recurrent component includes recurrent layers of long short-term memory (LSTM) [2] or managed recurrent units (GRU) [3] to take into account the context and sequence of the input data. The classification component performs sequence recognition and classification based on the predictions of the recurrent component.

This article studies the effectiveness of using CRNN for image recognition of digital sequences of variable length. In section 1 the need for research and the formulation of the problem are substantiated. Section 2 is devoted to the review and analysis of works on the use of CRNN networks for text recognition and digital sequences. Creating a dataset for training the model is described in section 3. The formation and study of a CRNN model for recognizing digital sequences in images is given in section 4. In conclusion, conclusions are presented based on the results of the research.

1. Setting the purpose and objectives of the study

The purpose of the study is to develop a CRNN model capable of achieving acceptable recognition accuracy of cadastral coordinates on scanned documents of the «Roskadastr» PLC. [4], [5] and [6] describe approaches implemented in the information system (IS) of this organization to solving the problem of converting images of cadastral coordinates into their text counterparts using CNN models. [6] shows the effectiveness of using CNN for sequences consisting of no more than 4 digits. Based on the results of studies carried out in [6], we can conclude that when recognizing a larger number of digits, the structure of the CNN becomes more complex, while the quality of recognition either remains the same or increases slightly. Considering that the maximum number of symbolic-digital elements in cadastral coordinates can exceed 10, one of the expedient and effective means of solving the problem of their recognition may be the use of CRNN models.

The goal set in the work can be achieved by solving the following main tasks:

- (1) Formation of a dataset, which consists in preparing a set of images of elements of cadastral numbers and their annotation (comparison with the cadastral number); generation of images of cadastral coordinates for a given number of their digital elements; dividing the dataset into samples for training and validation.
- (2) Formation of a CRNN model – selection of convolutional and recurrent layers for feature extraction and digital sequence recognition, respectively.
- (3) Model training and analysis of loss function values and accuracy metrics, CER, and WER.

2. Analysis of the main works on alphanumeric sequence recognition

For the first time, a CRNN model combining convolutional and recurrent layers for processing images with text sequences is described in [1]. The advantage of the model is the combination of convolutional and recurrent layers, which makes it possible to identify both local and global dependencies in images containing sequences and to implement their

recognition quite effectively. The disadvantages of the proposed model include fairly large amounts of data for training and significant time for recognizing long sequences.

The paper [7] describes a CRNN model that can recognize words unknown to it using meaningful contextual information. The model is resistant to various image distortions, does not depend on a predefined dictionary, and can process arbitrary sentences. The disadvantage of the model is poor text recognition with low contrast, unclear boundaries, and distortions, which requires additional image pre-processing methods.

The model from [8] is designed for text recognition in images with perspective distortion, curved character placement, etc. The proposed CRNN model is able to provide acceptable readability and recognition of distorted text and outperforms similar models in recognizing text with different values of input noise and slope. In addition, the model demonstrates high accuracy and performance when trained on large datasets. A disadvantage of the proposed model may be poor text recognition if noise or deformation greatly changes the shape of the characters, which is especially noticeable for text with a small font or low resolution.

In the paper [9] the authors apply CRNN to recognize text in images, focusing on difficult cases such as scenes with poor lighting or low resolution. They propose a model that uses convolutional layers to extract features from images and recurrent layers to model sequences from these characters. The experiments demonstrate high accuracy of text recognition on various images of text sequences.

A model for multi-level recognition of handwritten text in images is given in [10]. At the first level, CNN is used to recognize words that appear frequently in text. If a word is not recognized by this model, it moves to the second layer, which uses a fully convolutional network (FCN). An experimental study of the model was conducted using NIST19 as the training dataset and handwriting as the test dataset and showed quite acceptable recognition result.

In [11] presented two models for recognizing sequences of digits. In the first, the encoder and decoder of the sequences are CNN and LSTM. In the second – histogram of oriented gradient (HOG) and parallel fully connected (Dense) layers, respectively. Training and testing were carried

out on the Street View Number House dataset (SVHN). As a result of the conducted research, the advantage of CNN in terms of image encoding and the advantage of LSTM in sequence prediction were shown.

The DIGI-Net deep learning network, which is capable of learning the common characteristics of three different digit formats (handwritten, natural images, printed font) and recognizing them, is described in [12]. Experiments conducted on the MNIST, CVL and Chars74K, demonstrated high accuracy of recognition of continuous digital sequences.

The CRNN model proposed in [13] has a convolutional layer, a feature fusion layer, a recurrent layer, and a transcription layer. The convolutional layer used for feature extraction produces two outputs for the input text image. The feature pooling layer combines the results of the convolutional layer into one, from which the recurrent layer extracts sequences. The final result is output by the transcription layer. The proposed model, due to the fusion of features, realizes better text recognition accuracy on text datasets Street View Text (SVT), IIIT-5K, ICDAR2003 and ICDAR2013.

In [14] it is proposed to use a hybrid architecture of visual transformers (ViTs) and multilayer perceptrons (MLP) to recognize handwritten digits. Conducted research on the EMNIST and DIDA showed good accuracy in recognizing typewritten digital data, including on noisy images.

Paper [15] proposes a symmetrical multi-scale architecture called Circular Dilated Convolutional Neural Network (CDIL-CNN), where each element in the current layer has an equal chance of receiving information from other elements in previous layers. The proposed CRNN model allows the generation of classification logits for all elements, as a result of which it becomes possible to use simple ensemble learning to make the best decision. Based on the results of testing CDIL-CNN on long sequential datasets, it is shown that CDIL-CNN allows one to obtain recognition results that are acceptable in terms of accuracy.

The best approach to forming models for sequence recognition, from the point of view of the author of this work, is given in [16] and [17]. To recognize text sequences, it is proposed to use an encoder in the form of a CNN and a decoder in the form of a bidirectional long-term short-term memory (BSTM) using connectionist temporal classification (CTC). CTC is an algorithm used to train an RNN on variable length sequences and

match them with corresponding labels. In text and digit recognition tasks, CTC can cope with the problem of their variable length. It allows the model to predict a variable number of letters or numbers in a sequence without first separating or aligning them. The CTC algorithm calculates the probability of the output sequence and recursively updates the model weights based on the difference between the predictions and the sequence labels. The effectiveness of the proposed approach is shown on its own dataset in [16] and on the Arabic letters datasets MADCAT, AHTID/MW and IFN/ENIT in [17].

3. Creating the dataset

To train the CRNN model and study its operation, our own dataset was generated.

At the first stage, by analogy with [4], black and white images of elements of digital sequences were formed using the main fonts of «Roskadastr» PLC documents. The number of image classes is chosen to be 12 – 10 classes for numbers from 0 to 9, 1 class for delimiter characters «.» and «,», and one more for the absence of a character in the sequence. The values of the last 2 classes are chosen to be 10 and 11, respectively. For each class of images, 10 and 5 images of 20×25 pixels were generated for training and validating the model, respectively.

At the second stage, images of cadastral coordinates were formed, consisting of 2 digits in the fractional part and from 4 to 7 in the whole. Simultaneously with the formation of images, their CTC labels were also formed. All images in the dataset were reduced to the same size of 200×32 pixels. An example of the generated images of cadastral coordinates and the corresponding CTC labels is shown in Figure 1.

The dataset thus generated is shown in Table 1 and consists of 24240 and 12240 images of cadastral coordinates and their labels for training and validation, respectively.

4. Formation and research of a CRNN model

The model was formed using the Keras library. All layers of this model, in addition to the main CRNN layers – convolutional and recurrent

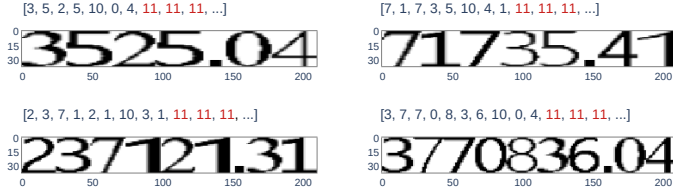


FIGURE 1. Examples of images of cadastral coordinates. Their CTC labels are indicated at the top

TABLE 1. Number of digital sequences for train and validation

Digits in sequence	Train sequences	Validation sequences
6	4848	2448
7	5656	2856
8	6464	3264
9	7272	3672

(Conv2D and Bidirectional respectively), are given in Table 2.

The downsampling layer MaxPooling2D reduces the dimension of the feature space, highlighting the most significant ones.

Layer BatchNormalization normalizes data in mini-batches, which speeds up training convergence and reduces the likelihood of overfitting. It also helps stabilize the distribution of activations between layers.

The Dropout layer is a regularizer, its goal is to reduce overfitting by preventing the activation of randomly selected neurons. This forces the model to learn more stable features and reduces the contribution of each individual neuron.

Layer Activation (activation function) applies an activation function to the output of the previous layer. In this model, this is ReLU, which activates neurons only for positive values, Tahn – for positive and negative values and Softmax, generating probabilities for different classes.

The fully connected layer Dense combines all outputs from the previous layer and applies linear transformations to produce the final model output. It connects the outputs of all neurons in the previous layer to each neuron in the current layer and is the main classification layer in neural networks.

TABLE 2. CRNN model layers

Layer	Activation	Filters	Input
InputLayer	–	–	[(None, 200, 32, 1)]
Conv2D	–	32	(None, 200, 32, 32)
BatchNormalization	–	–	(None, 200, 32, 32)
Activation	ReLu	–	(None, 200, 32, 32)
MaxPooling2D	–	–	(None, 100, 16, 32)
Conv2D	–	64	(None, 100, 16, 64)
BatchNormalization	–	–	(None, 100, 16, 64)
Activation	ReLu	–	(None, 100, 16, 64)
MaxPooling2D	–	–	(None, 50, 8, 64)
Dropout	–	–	(None, 50, 8, 64)
Conv2D	–	128	(None, 50, 8, 128)
BatchNormalization	–	–	(None, 50, 8, 128)
Activation	ReLu	–	(None, 50, 8, 128)
MaxPooling2D	–	–	(None, 50, 4, 128)
Dropout	–	–	(None, 50, 4, 128)
Conv2D	–	256	(None, 50, 4, 256)
BatchNormalization	–	–	(None, 50, 4, 256)
Activation	ReLu	–	(None, 50, 4, 256)
MaxPooling2D	–	–	(None, 50, 2, 256)
Dropout	–	–	(None, 50, 2, 256)
Reshape	–	–	(None, 32, 800)
Dense	–	–	(None, 32, 25)
Bidirectional	Tanh	–	(None, 32, 320)
Bidirectional	Tanh	–	(None, 32, 320)
Dense	–	–	(None, 32, 12)
Activation	Softmax	–	(None, 32, 12)

The `Lambda` layer allows you to define your own lambda function for non-standard data transformation, for example, changing its dimension. In the generated model, it is used in conjunction with `ground-truth` labels used for recognition tasks of continuous sequences. During the training phase, these labels are matched and compared with the ground truth labels, resulting in an error that is used to optimize the model.

The structure of the CRNN model is shown in Figure 2. To train it, the backpropagation method with the optimizer `Adam` was used.

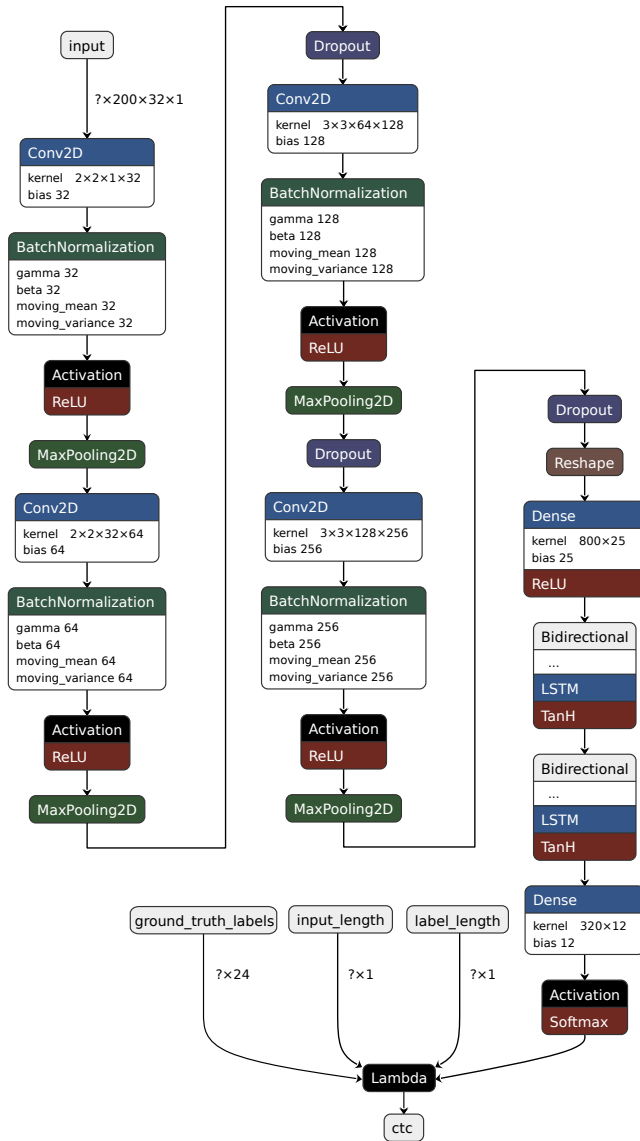


FIGURE 2. CRNN model

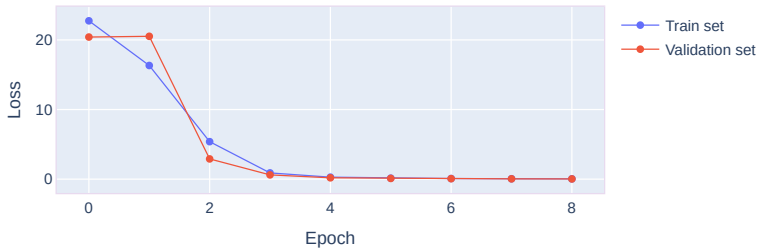


FIGURE 3. Model loss

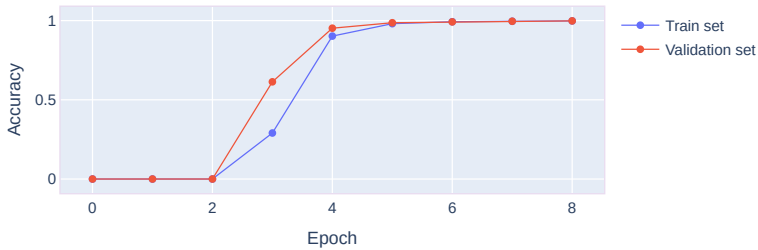


FIGURE 4. Model accuracy

To assess the quality of the model and its ability to solve the problem, the values of the loss function and accuracy metrics, CER, and WER were calculated. The loss function (**Loss**) measures the difference between the actual (true) label values and the values predicted by the model. In Figure 3 shows the model losses for 9 training epochs. This number of training epochs was found experimentally and is optimal.

The accuracy metric (**Accuracy**) measures the proportion of correct predictions made by the model relative to the total number of examples. It evaluates how well a model can classify or predict the correct class or value for a given set of data, Figure 4.

The numerical values of the loss functions and accuracy metrics for the train and validation sets at each of the training epochs are given in Table 3.

Calculating loss function values and accuracy metrics is a common approach to assessing the quality of neural network models. For models focused on sequence recognition, two more are calculated – Character Error Rate (**CER**) and Word Error Rate (**WER**). Both metrics are used to comparatively evaluate different recognition systems and analyze their accuracy.

TABLE 3. Numerical values of Loss and Accuracy

Epoch	Loss		Accuracy	
	Train set	Validation set	Train set	Validation set
0	22.7430438	20.4115753	0.0	0.0
1	16.3242683	20.5168876	0.0	0.0
2	5.3856644	2.9053363	0.0	0.0000817
3	0.8966413	0.6119849	0.2904703	0.6136437
4	0.2865071	0.1913659	0.9022276	0.9521241
5	0.1421806	0.1023578	0.9807755	0.9866012
6	0.0855829	0.0680020	0.9928630	0.9919934
7	0.0573494	0.0432046	0.9958745	0.9959967
8	0.0411685	0.0314824	0.9980198	0.9982843

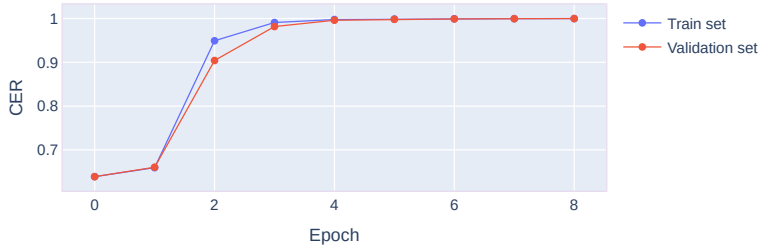


FIGURE 5. Character recognition accuracy

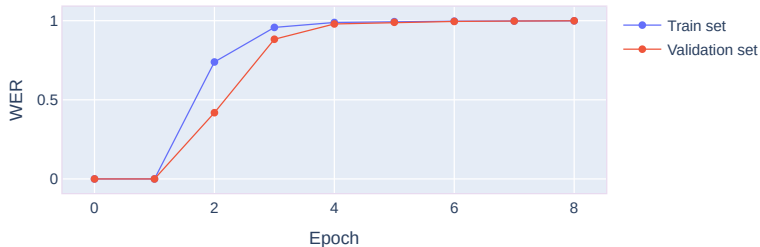


FIGURE 6. Word recognition accuracy

CER allows you to evaluate the accuracy of a model's recognition of individual characters. The dependences of the values of this metric for the training dataset and the validation dataset on the training epoch number are shown in Figure 5.

The accuracy of recognition of whole words by the model can be assessed by the WER metric. The dependences of the values of this metric for the training dataset and the validation dataset on the training epoch number are shown in Figure 6.

The numerical values of the CER and WER metrics for the train and validation sets at each training epoch are given in Table 4.

The results of recognition by the model of several cadastral coordinates

TABLE 4. Values of the CER and WER metrics

Epoch	CER		WER	
	Train set	Validation set	Train set	Validation set
0	0.6388888	0.6388888	0.0	0.0
1	0.6594093	0.6603758	0.0	0.0
2	0.9491560	20.9041156	0.7390676	0.4186274
3	0.9908536	0.9816210	0.9576320	0.8828431
4	0.9976399	0.9960001	0.9887788	0.9794117
5	0.9987451	0.9981072	0.9940182	0.9900327
6	0.9993949	0.9989787	0.9970297	0.9953431
7	0.9996046	0.9995132	0.9981435	0.9977942
8	0.999804	0.9998400	0.9992161	0.9994281



FIGURE 7. Results of recognition of cadastral coordinates.

One of the coordinates was recognized with an error

of 6, 7, 8, and 9 digits, respectively, from the test dataset are shown in Figure 7.

The model proposed by CRNN was implemented in the IS «Roskadastr» PLC and, as the results of its experimental study showed, the accuracy of recognition of individual symbols and cadastral coordinates as a whole was 99.98% and 99.94%, respectively. Images of cadastral coordinates were extracted from the scanned document according to the coordinates generated by the contourization subsystem of this IS. The general principles of operation of this subsystem are described in [4].

Conclusion

The article studies the use of CRNN architecture for the task of recognizing images of cadastral coordinates. Based on the results of training the model, graphs of the loss function (Loss) and accuracy (Accuracy) were constructed. The graphs showed that the model converges successfully and

is capable of achieving high accuracy in recognizing cadastral coordinates. To assess the quality of the model, the CER (Character Error Rate) and WER (Word Error Rate) metrics were used, which made it possible to measure the percentage of errors at the character and word levels, respectively. Based on the results of experimental studies, we can conclude that the model is capable of recognizing cadastral coordinates with a high level of accuracy and a minimum number of errors. As a result, the use of the CRNN model will significantly improve the efficiency and reliability of geospatial analyzes and decision making. Future research could focus on expanding the dataset, including different font types and styles, training the model on more diverse data, and investigating the effectiveness of CRNN for other recognition and classification tasks. It is also possible to use additional data preprocessing or augmentation methods to improve the accuracy of the model.

References

- [1] B. Shi, X. Bai, C. Yao. “An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**:11 (2017), pp. 2298–2304. [arXiv:1507.05717 \[cs.CV\]](#) [doi](#) ↑18, 19
- [2] S. Hochreiter, J. Schmidhuber. “Long short-term memory”, *Neural Computation*, **9**:8 (1997), pp. 1735–1780. [doi](#) [URL](#) ↑18
- [3] J. Chung, C. Gulcehre, K. Cho, Y. Bengio. “Gated feedback recurrent neural networks”, *Proceedings of Machine Learning Research*, **37** (2015), pp. 2067–2075. [arXiv:1502.02367 \[cs.NE\]](#) [URL](#) ↑18
- [4] I. V. Vinokurov. “Using a convolutional neural network to recognize text elements in poor quality scanned images”, *Program Systems: Theory and Applications*, **13**:3(54) (2022), pp. 45–59. [doi](#) [URL](#) ↑19, 22, 28
- [5] I. V. Vinokurov. “Tabular information recognition using convolutional neural networks”, *Program Systems: Theory and Applications*, **14**:1(56) (2023), pp. 3–30. [doi](#) [URL](#) ↑19
- [6] I. V. Vinokurov. “Recognition of digital sequences using convolutional neural networks”, *Program Systems: Theory and Applications*, **14**:3(58) (2023), pp. 3–36. [doi](#) [URL](#) ↑19
- [7] P. He, W. Huang, Y. Qiao, Change Loy C., X. Tang. “Reading scene text in deep convolutional sequences”, *AAAI’16: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (Phoenix, Arizona, USA, February 12–17, 2016), *Proceedings of the AAAI Conference on Artificial Intelligence*, **30**:1 (2016), pp. 3501–3508. [doi](#) ↑20
- [8] B. Shi, X. Wang, P. Lv, C. Yao, X. Bai. “Robust scene text recognition with automatic rectification”, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV, USA, June 27–30, 2016), 2016, pp. 4168–4176. [doi](#) [arXiv:1603.03915 \[cs.CV\]](#) ↑20

- [9] F. Yin, Y.-C. Wu, X.-Y. Zhang, C.-L. Liu. *Scene text recognition with sliding convolutional character models*, 2017, 10 pp. [arXiv:1709.01727](#) [cs.CV] [↑20](#)
- [10] D. A. Nirmalasari, N. Suciati, D. A. Navastara. “Handwritten text recognition using fully convolutional network”, *IOP Conference Series: Materials Science and Engineering*, **1077**:1 (2021), id. 012030, 9 pp. [doi](#) [↑20](#)
- [11] X. Liu, Y. Deng, Y. Sun, Y. Zhou. “Multi-digit recognition with convolutional neural network and long short-term memory”, *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)* (Huangshan, China, July 28–30, 2018), IEEE, 2018, pp. 1187–1192. [doi](#) [↑20](#)
- [12] A. Madakannu, A. Selvaraj. “DIGI-Net: a deep convolutional neural network for multi-format digit recognition”, *Neural Computing and Applications*, **32** (2020), pp. 11373–11383. [doi](#) [↑21](#)
- [13] L. Zou, Z. He, K. Wang, Z. Wu, Y. Wang, G. Zhang, X. Wang. “Text recognition model based on multi-scale fusion CRNN”, *Sensors*, **32**:16 (2023), id. 7034, 18 pp. [doi](#) [↑21](#)
- [14] V. Agrawal, J. Jagtap. *Convolutional vision transformer for handwritten digit recognition*, Research Square, 2022, 11 pp. [doi](#) [↑21](#)
- [15] L. Cheng, R. Khalitov, T. Yu, Z. Yang. *Classification of long sequential data using circular dilated convolutional neural networks*, 2022, 16 pp. [arXiv:2201.02143](#) [cs.LG] [↑21](#)
- [16] R. S. Bhat. *Text recognition with CRNN-CTC network*, W&B Fully Connected, 2022. [URL](#) [↑21, 22](#)
- [17] S. Khamekhem, A. Sourour, Y. Kessentini. “Domain and writer adaptation of offline Arabic handwriting recognition using deep neural networks”, *Neural Computing and Applications*, **34** (2022), pp. 2055–2071. [doi](#) [↑21, 22](#)

Received
approved after reviewing
accepted for publication
published online

29.09.2023;
27.11.2023;
27.11.2023;
11.03.2024.

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

The author declare no conflicts of interests.

УДК 519.876.5:62-713

doi 10.25209/2079-3316-2024-15-1-31-40



Оптимальное распределение площади радиаторов в погружных системах охлаждения высокопроизводительных вычислительных комплексов

Сергей Анатольевич Амелькин[✉]

Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации,
Москва, Россия

[✉]amelkin@ist.education

Аннотация. Рассмотрена задача минимизации температуры процессора при заданном тепловом потоке путем выбора распределения площади радиатора при контакте с омывающим его хладагентом. Эта задача эквивалентна задаче минимизации среднего (по координате) производства энтропии. Распределение тепловой нагрузки и ограничение общей площади радиатора являются условиями задачи. Показано, что оптимальное решение обеспечивает минимальную температуру процессора в погружных жидкостных системах охлаждения.

Ключевые слова и фразы: Погружные системы охлаждения, площадь радиатора, процессы минимальной диссипации

Благодарности: Исследование выполнено за счет гранта Российского научного фонда № 23-21-00173

Для цитирования: Амелькин С. А. *Оптимальное распределение площади радиаторов в погружных системах охлаждения высокопроизводительных вычислительных комплексов* // Программные системы: теория и приложения. 2024. Т. 15. № 1(60). С. 31–40. https://psta.psiras.ru/read/psta2024_1_31-40.pdf

Введение

Погружные жидкостные системы охлаждения высокопроизводительных вычислительных комплексов позволяют поддерживать необходимые для нормальной работы процессоров температурные режимы при различных климатических условиях. Благодаря тому, что коэффициент теплоотдачи от радиатора к жидкости приблизительно в 100 раз больше, чем к воздуху [1], использование стандартных, предназначенных для воздушных систем охлаждения, радиаторов обеспечивает охлаждение процессоров. Однако, эффективность системы охлаждения может быть повышена за счет оптимального распределения площади радиатора. Это особенно актуально при увеличении мощности процессоров.

Так, например, существенным для эффективности вычислительных процессов параметром является плотность процессоров, так что решение задач будет лучше организовано при небольшом количестве мощных процессоров, чем при большом количестве процессоров той же суммарной мощности [1]. Таким образом, для организации вычислительного процесса выгодно увеличить вычислительную мощность процессоров, т. е. «разогнать» процессоры вычислительного комплекса.

Задача оптимального распределения температур хладагента при различных моделях потока хладагента подробно рассмотрена в [3]. В [4] рассмотрена задача выбора оптимального режима для погружных однофазных и двухфазных жидкостных систем охлаждения. Критерием оптимальности в рассмотренных задачах является производство энтропии в теплообменном аппарате как термодинамической системе. Условия задачи является заданное общее количество переданного тепла и начальные значения температур теплоносителей. Минимум среднего производства энтропии позволяет обеспечить эффективное охлаждение хладагентом максимальной начальной температуры при заданной общей площади контакта между хладагентом и радиатором. При решении задач предполагалось, что коэффициент теплоотдачи и температура процессора заданы. В реальных системах теплопередача от процессора к хладагенту организуется через радиатор, площадь поверхности которого может быть функцией координаты, а температура процессора определяется заданным полем теплового потока [5].

Постановка задачи

Увеличение производительности процессоров сопровождается значительным повышением тепловыделения, а это значит, что при тех же условиях отвода тепла работа в режиме «турбо» приведет к нагреву процессора. Поскольку тепловыделение функционально зависит только от вычислительной производительности процессора, а в широком диапазоне температур верхний предел вычислительной производительности процессора можно считать постоянным, при постановке задачи следует в качестве параметра выбрать не температуру процессора, а интенсивность теплового потока.

Современные процессоры нуждаются в качественном охлаждении даже при работе в номинальном режиме. При разгоне процессоров наблюдается если не нестабильность работы, то по крайней мере температурный троттлинг. Поэтому производители процессоров устанавливают жесткие ограничения на температурный режим, в котором можно увеличивать производительность процессора. В результате режим «турбо» в существующих системах может поддерживаться только в течение непродолжительного времени, что нивелирует преимущества режима «турбо».

Задача повышения производительности вычислительного комплекса состоит в том, чтобы при заданном тепловом потоке минимизировать температуру процессора. Эта задача эквивалентна [3] задаче минимизации среднего (по координате) производства энтропии, формальная постановка которой для ньютоновского закона теплоотдачи $q(l) = \alpha(l)(T_0(l) - T(l))$ имеет вид

$$(1) \quad \sigma = \int_0^L q(l) \left[\frac{1}{T(l)} - \frac{1}{\frac{q(l)}{\alpha(l)} + T(l)} \right] dl \rightarrow \min_{\alpha(l)} \left\{ \begin{array}{l} \int_0^L \alpha(l) dl = A; \\ \frac{dT}{dl} = \frac{q(l)}{W}, \quad T(0) = T_1; \\ T_0(l) = \frac{q(l)}{\alpha(l)} + T(l) \leq T_m, \end{array} \right.$$

где: l – координата процесса (м), соответствующая направлению потока хладагента, омывающего радиатор ($l \in [0, L]$), L – длина радиатора);

$q(l)$ – удельный тепловой поток ($\text{Вт}/\text{м}$), определяемый вычислительной нагрузкой процессора;

$T_0(l)$ – температура радиатора (К) при контакте с хладагентом; будем предполагать, что вследствие высокой теплопроводности материала радиатора, небольшой высоты ламелей и низкого термического сопротивления при контакте радиатора с процессором температура радиатора в каждой точке l одинакова и приблизительно равна температуре процессора, при этом накладывается дополнительное ограничение на максимальное значение $T_0(l) \leq T_m$;

$T(l)$ – температура хладагента (К), изменение которой определяется тепловым потоком $q(l)$ от радиатора к хладагенту и теплоемкостью хладагента $W(l)$ (К/Вт), равной произведению массового расхода на удельную теплоемкость хладагента;

$\alpha(l)$ – удельный коэффициент теплоотдачи (Вт/м² · К), линейно зависящий от площади контакта радиатора с хладагентом; ограничение на общую площадь радиатора, пренебрегая зависимостью коэффициента теплоотдачи от скорости потока хладагента можно заменить на ограничение на среднее значение $\overline{\alpha(l)} = A/L$.

Решение задачи (1) определяет геометрические параметры радиатора, обеспечивающие оптимальный режим его работы в погружных системах охлаждения.

Условия оптимальности

Так как температура хладагента всегда ниже температуры радиатора, функция $T(l)$ является монотонно возрастающей. Следовательно, в задаче (1) допустима замена переменной:

$$(2) \quad dl = \frac{W(l)}{q(l)} dT.$$

Уравнение

$$(3) \quad \int_0^L dl = \int_{T(0)}^{T(L)} \frac{W(T)}{q(T)} dT = L,$$

позволяет определить значение температуры хладагента в точке L : $T(L) = T_2$ (функции $W(l)$ и $q(l)$ заданы по условию задачи), так что после

замены переменной задача (1) примет вид:

$$(4) \quad \sigma = \int_{T_1}^{T_2} W(T) \left[\frac{1}{T} - \frac{1}{\frac{q(T)}{\alpha(T)} + T} \right] dT \rightarrow \min_{\alpha(l)} \left| \int_{T_1}^{T_2} \frac{\alpha(T)W(T)}{q(T)} dT = A. \right.$$

Функция Лагранжа для изопериметрической задачи (4) имеет вид:

$$(5) \quad R = W(T) \left[\frac{1}{T} - \frac{1}{\frac{q(T)}{\alpha(T)} + T} \right] + \lambda \frac{\alpha(T)W(T)}{q(T)},$$

где λ – безразмерный неопределенный множитель Лагранжа. Необходимое условие оптимальности $\frac{dR}{d\alpha} = 0$ приводит к уравнению:

$$(6) \quad -\frac{q(T)}{\alpha(T)^2 \left(\frac{q(T)}{\alpha(T)} + T \right)^2} + \frac{\lambda}{q(T)} = 0,$$

откуда выражаем искомое значение $\alpha(T)$:

$$(7) \quad \alpha(T) = \frac{q(T)}{T} \frac{1 - \sqrt{\lambda}}{\sqrt{\lambda}} = \frac{\mu q(T)}{T},$$

где $\mu = \frac{1 - \sqrt{\lambda}}{\sqrt{\lambda}} = \text{const}$ находим из ограничения на общую площадь (средний коэффициент теплоотдачи) радиатора.

Полученное решение необходимо проверить на выполнение ограничения $T_0(l) = \frac{q(l)}{\alpha(l)} + T(l) \leq T_m$ во всех точках $l \in [0, L]$.

Реализация оптимальных радиаторов

Полученные условия оптимальности (7) определяют распределение площади (удельного коэффициента теплоотдачи) радиатора при известных зависимостях теплового потока и теплоемкости потока хладагента. Если для плоского ламельного радиатора скорость потока хладагента одинакова по длине радиатора, то более сложные конструкции характеризуются изменением скорости, а значит, и теплоемкости потока. Полученные распределения $\alpha(l)$ для разных зависимостей $q(l)$ и $W(l)$ показаны в таблице 1.

Таблица 1. Распределение удельного коэффициента теплоотдачи для разных характеристик радиаторов

Условия	Температура хладагента	Распределение удельного коэффициента теплоотдачи
$q(l) = q_0 = \text{const},$ $W = W_0 = \text{const}$	$T(l) = T_1 + \frac{q_0}{W_0} l$	$\alpha(l) = \frac{Ar}{(1 + rl) \ln(1 + rL)}, \quad r = \frac{q_0}{W_0 T_1}$
$q(l) = q_0 e^{-ml},$ $W = W_0 = \text{const}$	$T(l) = T_1 + \frac{q_0}{mW_0} (1 - e^{-ml})$	$\alpha(l) = \frac{Arm}{(m + r(1 - e^{-ml})) \ln(1 + \frac{rL}{m})}, \quad r = \frac{q_0}{W_0 T_1}$
$q(l) = q_0 = \text{const},$ $W = W_0 + wl$	$T(l) = T_1 + \frac{q_0}{w} \ln \frac{W_0 + wl}{W_0}$	$\alpha(l) = \frac{Aw\xi}{W_0 + \frac{q_0 W_0}{w T_1} \ln(1 + \frac{wl}{W_0})}, \quad \xi = \int_0^{\ln(1 + \frac{wL}{W_0})} \frac{e^x dx}{1 + \frac{q_0 x}{w T_1}}$

Последние условия характерны для радиатора типа «солнце»¹. Распределение площади радиально расположенных ламелей позволяет организовать охлаждение радиатора путем подачи жидкого хладагента температурой 20°C через трубку диаметром 0,5 мм так, что при изменении суммарной мощности теплового потока с 130 до 200 Вт (при увеличении производительности процессора Intel Core i7-4960X Extreme Edition с 137,7 до 151,7 Гфлопс, то есть на 10%) обеспечивается увеличение температуры процессора с 72°C до 100°C, что соответствует рабочим характеристикам процессора. Полученное решение удовлетворяет ограничению на максимальную температуру процессора, которая не должна превышать 110°C.

Для разработки вычислительного комплекса повышенной производительности, оснащенного погружной жидкостной системой охлаждения и использующего в качестве нормального рабочего режим «турбо», требуется установка специально разработанных радиаторов (например, типа «солнце»), поддерживающих высокие показатели энергоэффективности системы охлаждения. При установке радиатора с оптимальным распределением площади в штатном режиме работы процесс теплообмена с минимальным производством энтропии соответствует довольно низкой температуре процессора, значительно ниже установленного температурного ограничения. Такой запас позволяет использовать режим «турбо» без превышения температурного ограничения при заданной температуре подводимого хладагента. В случае увеличения температуры хладагента выше 30°C ограничение на максимально допустимую температуру процессора становится активным, следовательно, реализовать режим «турбо» в течение всего времени работы вычислительного комплекса невозможно.

Список литературы

- [1] Романков П.Г., Фролов В.Ф., Флисюк О.М. *Методы расчетов процессов и аппаратов химической технологии*. – СПб.: Химиздат. – 2009. – 544 с. ↑³²

¹см. доклад "Имитационные и математические модели компактных высокопроизводительных вычислительных комплексов с погружным охлаждением"¹⁰⁰ 26 ноября 2018, Переславль-Залесский, Россия: Национальный суперкомпьютерный форум НСКФ-2018, авторы Демидов А.А., Амелькин С.А., Чичковский А.А.

- [2] Стегайлов В.В., Норман Г.Э. *Проблемы развития суперкомпьютерной отрасли в России: взгляд пользователя высокопроизводительных систем* // Программные системы: теория и приложения. – 2014. – Т. 5. – № 1 (19). – С. 111–152.   ↑
- [3] Цирлин А.М. *Математические модели и оптимальные процессы в макросистемах*. – М.: Наука. – 2006. – ISBN 5-02-034084-7. – 500 с. ↑^{32, 33}
- [4] Ахременков А.А., Цирлин А.М. *Математическая модель жидкостного погружного охлаждения вычислительных устройств* // Программные системы: теория и приложения. – 2016. – Т. 7. – № 1 (28). – С. 187–199.   ↑³²
- [5] Elliott J.W., Lebon M.T., Robinson A.J. *Optimising integrated heat spreaders with distributed heat transfer coefficients: A case study for CPU cooling* // Case Studies in Thermal Engineering. – 2022. – Vol. 38. – id. 102354. – 12 pp.  ↑³²

Поступила в редакцию 07.12.2023;
 одобрена после рецензирования 27.12.2023;
 принята к публикации 28.12.2023;
 опубликована онлайн 11.03.2024.

Рекомендовал к публикации

д.т.н. А. М. Цирлин

Информация об авторе:



Сергей Анатольевич Амелькин

к.т.н., ст.научн.сотр. Института программных систем имени А. К. Айламазяна РАН. Доцент кафедры информатики и системного анализа института Экономики, математики и информационных технологий Российской академии народного хозяйства и государственной службы при Президенте Российской Федерации



0000-0002-4004-7159

e-mail: amelkin@ist.education

Автор заявляет об отсутствии конфликта интересов.



Optimal distribution of radiator area in immersion cooling systems of high-performance computing systems

Sergej Anatolyevich **Amelkin**

The Presidential Academy of National Economy and Public Administration, Moscow, Russia

 amelkin@ist.education

Abstract. The problem of minimizing the processor temperature for a given heat flow is considered. Control is the distribution of the radiator area in contact with the coolant. This problem is equivalent to the problem of minimizing the average (over coordinate) entropy production. The distribution of the thermal load and the total radiator area are the conditions of the problem. It is shown that the optimal solution ensures the minimum processor temperature in immersion liquid cooling systems of high-performance computer. (*In Russian*).

Key words and phrases: Immersion cooling systems, radiator area, minimal dissipation processes

2020 *Mathematics Subject Classification:* 80M50; 80A20, 41A46

Acknowledgments: The study was supported by Russian Science Foundation grant No. 23-21-00173

For citation: Sergej A. Amelkin. *Optimal distribution of radiator area in immersion cooling systems of high-performance computing systems*. Program Systems: Theory and Applications, 2024, **15**:1(60), pp. 31–40. (*In Russ.*). https://psta.psiras.ru/read/psta2024_1_31-40.pdf

References

- [1] P.G. Romankov, V.F. Frolov, O.M. Flisyuk. *Methods for calculating processes and apparatus of chemical technology*, Ximizdat, SPb., 2009 (in Russian), 544 pp.
- [2] V.V. Stegajlov, G.E. Norman. “Challenges to the supercomputer development in Russia: a HPC user perspective”, *Program Systems: Theory and Applications*, **5**:1 (19) (2014), pp. 111–152 (in Russian).  
- [3] A.M. Cirlin. *Mathematical models and optimal processes in macrosystems*, Nauka, M., 2006, ISBN 5-02-034084-7 (in Russian), 500 pp.
- [4] A.A. Axremenkov, A.M. Cirlin. “Mathematical model of liquid immersion cooling system for supercomputer”, *Program Systems: Theory and Applications*, **7**:1 (28) (2016), pp. 187–199 (in Russian).  
- [5] J.W. Elliott, M.T. Lebon, A.J. Robinson. “Optimising integrated heat spreaders with distributed heat transfer coefficients: A case study for CPU cooling”, *Case Studies in Thermal Engineering*, **38** (2022), id. 102354, 12 pp. 

УДК 536.757

 10.25209/2079-3316-2024-15-1-41-62

Фрактальная модель макросистем

Сергей Анатольевич **Амелькин**

Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации, Москва, Россия

Аннотация. Рассмотрена математическая модель макросистемы произвольной природы в виде фрактального графа. Такое представление позволяет вывести феноменологические закономерности макросистем, не основываясь на свойствах элементарных объектов, формирующих макросистему. Показано, что на множестве стационарных процессов можно ввести метрику; метрическими свойствами обладает производство энтропии в макросистеме.

Ключевые слова и фразы: Макросистемы, производство энтропии, процессы минимальной диссипации

Благодарности: Исследование выполнено за счет *гранта Российского научного фонда № 23-21-00173*^{}

Для цитирования: Амелькин С. А. *Фрактальная модель макросистем* // Программные системы: теория и приложения. 2024. **Т. 15.** № 1(60). С. 41–62.
https://psta.psir.ru/read/psta2024_1_41-62.pdf

Введение

При анализе закономерностей, возникающих в процессах ресурсообмена в системах различной природы: физических, химических, экономических, информационных, социальных целесообразно использовать макросистемный подход, учитывающий взаимосвязи между усредненными характеристиками большого количества неконтролируемых элементарных объектов. Феноменологические особенности макросистемного подхода связаны с тем, что математические модели макросистем основаны на наблюдаемых свойствах совокупности элементарных объектов, а эти свойства описываются макроскопическими величинами, в то время как внутренние механизмы взаимодействия элементарных объектов не рассматриваются ввиду их большого количества. При этом макроскопические величины не могут быть применимы к отдельным элементарным объектам. Для физических (прежде всего термодинамических) макросистем связь макроскопических величин – температуры, давления, внутренней энергии и пр. – с законами взаимодействия элементарных объектов (молекул) определяется статистической физикой, описывающей микросостояния элементарных объектов в соответствии с их координатами и импульсами. Таким образом, каждый элементарный объект является независимым от других, но его траектория в фазовом пространстве может изменяться при взаимодействии с другими объектами.

Взаимодействия в социальных макросистемах описываются, исходя из предположения, что каждый индивидум (актор), принимающий решения, строго рационален: его поведение полностью обусловлено наличием целевой функции, значение которой он стремится осознанно или неосознанно максимизировать в ходе взаимодействия с другими акторами. В качестве причин иррациональности описываются ненаблюдаемые и неизмеряемые физиологические и психологические особенности личности [1], взаимовлияние общества и личности [2] и др. Проведенные исследования [3] показали, что рациональность акторов ограничена и поведение акторов как элементарных объектов определяется социумом, в котором они взаимодействуют.

Взаимозависимость элементарных объектов и непрогнозируемость их поведения характерны для любых макросистем, отличных от физических: подобные явления наблюдаются в экологических, демографических, образовательных и иных видах социальных и информационных макросистем [4, 5]. Обоснование зависимостей макроскопических параметров в таких системах требует иного обоснования. Так, в [6] предложено деление макросистемы на микро-, мезо-, экзо- и макроуровни. На каждом

уровне рассматривается взаимодействие между системами, а уровень индивидуальных взаимодействий выводится за пределы наблюдений. Следующим шагом моделирования является переход от концентрической модели к спиральной [7]. Логически завершает этот путь фрактальное представление макросистем, предлагаемое в настоящей работе.

Основными задачами работы являются:

- построить модель* макросистемы произвольной природы, как фрактальную систему, свойства которой на каждом уровне описываются по распределению состояний макросистем на более низком уровне.
- описать свойства* экстенсивных и интенсивных параметров макросистем и показать, что можно ввести понятие расстояния между процессами в макросистемах, и таким образом количественно измерить отклонение процессов друг от друга.

1. Математическая модель ресурсообмена

Макросистемами называют системы, для которых выполняются следующие условия:

1. Макросистемы состоят из большого количества элементарных объектов, настолько большого, что любую макросистему можно разделить на любое конечное количество подсистем, каждая из которых может рассматриваться, как макросистема. Объединение нескольких макросистем является макросистемой более высокого уровня. Макросистема Y может рассматриваться, как объединение конечного числа макросистем более низкого уровня $X_i : Y = \bigcup_{i \in \mathcal{X}} X_i$. Это условие называют условием фрактальности [8]. Внешнее окружение макросистемы также будем рассматривать, как макросистемы (и макросистемой более высокого уровня является объединение макросистемы и ее внешнего окружения). Поскольку количество подсистем, на которое можно разделить макросистему может быть сколь угодно большое, в том числе достаточное для определения статистических характеристик любой заданной точности, то подсистемы наряду с макросистемными свойствами обладают всеми свойствами элементарных объектов. Таким образом нет необходимости отдельно рассматривать свойства элементарных объектов, составляющих макросистему.¹ Например, заданный объем идеального газа можно разделить на сколь угодно большое конечное число элементарных

¹При этом однако требуется аккуратно отслеживать, что части, содержащие статистически мало элементарных объектов, никогда в рассмотрении не возникают. (Примечание редактора.)

объемов, каждый из которых, тем не менее, будет обладать всеми термодинамическими свойствами идеального газа.

2. Состояние макросистемы определяется вектором запасов ресурсов Q , которые рассматриваем, как аддитивные величины: если Y – объединение нескольких макросистем $X_i : Y = U_{i \in \mathcal{X}} X_i$, то $Q_Y = \sum_i Q_{X_i}$. Макросистема может обмениваться ресурсами с ее внешним окружением; макросистемы, составляющие (то есть являющиеся подсистемами) макросистему более высокого уровня, могут обмениваться ресурсами между собой. Потоки ресурсов между подсистемами X и Y обозначим q_{XY} . Макросистемы любой природы могут быть описаны через запасы ресурсов. Л.И. Розоноэр вводит понятие ресурса не только для экономических, но и для физических систем: «в физике под ресурсами можно понимать такие величины, как энергия, импульс, масса, которыми обмениваются различные части физической системы» [9].
3. Каждым элементарным объектом управлять невозможно из-за их большого числа, управление макросистемой может быть организовано только путем воздействия на усредненные по множеству элементарных объектов параметры, а именно:
 - изменение параметров внешнего окружения макросистемы, взаимодействие с которым обуславливает изменение запасов ресурсов в макросистеме;
 - изменение запасов ресурсов (например, извлечение их) в макросистеме за счет внешних интервенций;
 - изменение характеристик инфраструктуры ресурсообмена в целях ускорения или, наоборот, замедления процессов ресурсообмена.

Перечисленные условия определяют макросистему вне зависимости от ее природы. В терминах ресурсов можно описать как экономические макросистемы на уровне отдельных потребителей [10], отраслей народного хозяйства [11] или регионов [12], так и информационные системы обмена синтаксической и семантической информацией [13, 14], а также экономические аспекты обмена информацией [15] термодинамические системы обмена веществом и энергией [16] и пр. В ходе обмена ресурсами общее его количество в макросистеме может сохраняться (в этом случае говорят о законе сохранения ресурса) или нет, что происходит в условиях производства, потребления или утилизации ресурса в ходе взаимодействия между подсистемами. Интенсивности потоков различных ресурсов в ходе обмена могут быть взаимосвязаны: ресурсы являются взаимодополняющими (взаимозаменяемыми) при монотонно возрастающей (убывающей) взаимосвязи между интенсивностями их потоков.

В физических системах тепло- и массообмена ресурсами являются энергия, количество вещества, объем и масса, в экономических системах к ресурсам можно отнести запасы материальных и финансовых активов, в системах, где наблюдается обмен информацией, ресурсами являются формирующие дискурс корпусы текстов, содержащих синтаксическую, семантическую и прагматическую информацию.

Макросистему можно представить в виде *фрактального цветного ориентированного взвешенного мультиграфа*, узлы которого соответствуют подсистемам и внешнему окружению макросистемы, а ребра – потокам ресурсов между подсистемами и между подсистемами и внешним окружением макросистемы:

Фрактального – каждый узел графа можно представить в виде графа, описывающего взаимодействие подсистем, образующих этот узел между собой и с их окружением, так что:

$$(1) \quad \left. \begin{aligned} \sum_{i \in \mathcal{X}} Q_i &= Q_X; & \sum_{j \in \mathcal{Y}} Q_j &= Q_Y; \\ \sum_{\substack{i \in \mathcal{X} \\ j \in \mathcal{Y}}} q_{ij} &= q_{XY}. \end{aligned} \right\}$$

Цветного – каждый узел графа (каждая подсистема) характеризуется вектором ресурсов $Q = (Q_0, \dots, Q_N)$, а потоки ресурсов между подсистемами могут быть функционально не связанными друг с другом, поэтому ребра графа, соответствующие потокам различных ресурсов, маркируются соответствующими цветами.

Ориентированного – направление каждого ребра определяет положительный знак потока каждого ресурса; так, если ребра направлены от узла X к узлу Y , то:

$$(2) \quad q_{XY, \nu} = -\frac{dQ_X}{dt} = \frac{dQ_Y}{dt}, \quad \nu = 0, \dots, N.$$

Впрочем, ребра разного цвета, соответствующие разным ресурсам, могут быть как сонаправленными, так и противоположно направленными. Так, например, в информационных макросистемах потоки синтаксической и семантической информации всегда сонаправлены, а в экономических системах добровольного ресурсообмена потоки благ: услуг и товаров и денег – противоположно направлены.

Взвешенного – вес ребра показывает интенсивность потока: положительное значение потока $q_{XY, \nu} > 0$, если реальный поток ресурса направлен по направлению ребра, $q_{XY, \nu} < 0$, если – противоположно направлению ребра. Кроме того, для всех потоков между узлами X и Y (то есть для всех потоков q_{XY}) определена матрица A инфраструктурных

коэффициентов. Эта матрица является метаданными для ребер всех цветов, соединяющих узлы X и Y , и определяет свойства взаимодополняемости и взаимозаменяемости ресурсов.

Интенсивность процесса ресурсообмена зависит как от параметров состояния подсистем, обменивающихся ресурсами, так и от условий (инфраструктуры) ресурсообмена. Параметры состояния взаимодействующих подсистем определяют движущие силы процессов ресурсообмена, а инфраструктура – коэффициенты пропорциональности между интенсивностями потоков и величинами движущих сил. Так, для процесса теплообмена к инфраструктуре можно отнести площадь контакта между обменивающимися теплом подсистемами, инфраструктурный коэффициент здесь – коэффициент теплопередачи. В экономике под инфраструктурой понимают комплекс учреждений и институций, обеспечивающих свободное движение товаров и услуг на рынке. Развитие инфраструктуры снижает барьеры, влияющие на интенсивность экономического обмена, тем самым увеличивая эластичность спроса и предложения товаров и услуг, определяющую инфраструктурные коэффициенты ресурсообмена в экономических макросистемах. При этом движущие силы и потоки ресурсов могут быть взаимосвязанными, именно величины инфраструктурных коэффициентов в матрице A показывают взаимное влияние движущих сил ресурсообмена и интенсивностей потоков.

Мультиграф – два узла графа могут быть соединены как ребрами различных цветов, так и группами ребер различных цветов, что соответствует множеству точек контакта между подсистемами. Отсутствие ребер некоторых цветов между узлами мультиграфа означает изолированность узлов друг от друга. Отсутствие ребра между узлами равносильно нулевому значению веса этого ребра, так что в общем случае можно говорить о том, что граф макросистемы является турниром (ориентированным графом, полученным из неориентированного полного графа). Пусть две макросистемы X и Y обмениваются вектором ресурсов Q . В каждый момент времени t запасы ресурсов, описывающие состояние макросистем, равны $Q_X(t), Q_Y(t)$. Макросистемы представим в виде объединения конечного множества подсистем: $\mathcal{X}$ и $\mathcal{Y}$ соответственно. Каждой паре узлов X_i и Y_j соответствует вектор потоков ресурсов $q_{ij}(t)$. Сумма всех векторов $q_{ij}(t)$ определяет интенсивность потоков ресурсов между макросистемами X и Y .

Состоянием равновесия назовем такое состояние (Q_X, Q_Y) , при

котором сумма потоков ресурсов в каждый момент времени t

$$(3) \quad q_{XY}(t) = \sum_{\substack{i \in \mathcal{X} \\ j \in \mathcal{Y}}} q_{ij}(t) = 0.$$

Таким образом, равновесие в макросистеме рассматривается, как динамическое. Равенство нулю вектора $q_{XY}(t)$ не означает, что $q_{ij}(t) = 0$, но что $q_{ij}(t)$ распределены с математическим ожиданием, равным нулю при любом t . Условие равновесия можно определить следующим образом: на уровне любых множеств подсистем $\mathcal{X}, \mathcal{Y} : X = U_{i \in \mathcal{X}} X_i; Y = U_{j \in \mathcal{Y}} Y_j$ потоки $q_{ij} (i \in \mathcal{X}, j \in \mathcal{Y})$ имеют распределение $f(\tilde{q})$, не зависящее от времени с математическим ожиданием $M[\tilde{q}] = 0$.

Отметим, что потоки $q_{ij}(t) (i \in \mathcal{X}, j \in \mathcal{Y})$, и случайная величина $\tilde{q}$, описывающая распределение потоков подсистем, – вектора. На этом уровне подсистем можно предполагать большое количество факторов, влияющих на распределение потоков, это означает, что распределение $f(\tilde{q})$ можно описать многомерным нормальным распределением. Параметрами этого распределения являются математическое ожидание $M[\tilde{q}]$, определяющее потоки ресурсов на уровне подсистем X и Y в соответствии с (3), и ковариационная матрица $C[\tilde{q}]$.

В предположении, что потоки ресурсов q_{XY} возникают вследствие действия некоторых движущих сил, которые на уровне подсистем мы можем также рассматривать, как случайный вектор, такой что:

- потоки ресурсов q_{XY} линейно зависят от движущих сил φ_{XY} ресурсообмена:

$$(4) \quad q_{XY} = A\varphi_{XY}$$

(здесь $A = (\alpha_{\nu k})$ – матрица инфраструктурных элементов), при таком предположении распределение движущих сил также является нормальным;

- ковариационная матрица $C[\tilde{q}]$ подсистем макросистемы $X \cup Y$ зависит от интенсивности движущих сил $\varphi_{XY} = M[\tilde{\varphi}]$, вызывающих эти потоки, так что матрицы $C[\tilde{q}]$ и $C[\tilde{\varphi}]$ совместно нормализуемы (их собственные вектора совпадают) – вследствие линейности связи между потоками и движущими силами;
- пределы коэффициентов корреляции, соответствующих ковариационной матрице $C[\tilde{q}]$, для любых $\nu, k = 0, \dots, N : \lim_{\varphi_{XY} \rightarrow 0} \rho_{\nu k} = 0$,
 $\lim_{\varphi_{XY} \rightarrow \infty} \rho_{\nu k} = 1$ – вследствие перераспределения ресурсов по множеству подсистем в зависимости от количества промежуточных узлов данного цвета в цепи мультиграфа до точки контакта.

Движущими силами для процесса ресурсообмена в физических макросистемах являются разницы температур и давлений, в экономических – разницы оценок ценности благ, а в информационных – мотивации для получения и распространения информации. Движущие силы определяются неравенством соответствующих параметров двух подсистем, между которыми возможен обмен ресурсами. При этом на интенсивность потоков влияют и инфраструктурные параметры, такие как площадь контакта двух подсистем в процессе теплообмена, реклама в экономических макросистемах или использование различных модальностей при передаче информации.

Если для макросистемы X условие равновесия выполняется при взаимодействии с каждой макросистемой из ее окружения, то такая макросистема называется замкнутой. Если для макросистемы X все ее подсистемы находятся в состоянии равновесия при взаимодействии друг с другом (но не обязательно с окружением макросистемы X):

$$(5) \quad \forall t, \forall i \in \mathcal{X} : \sum_{j \in \mathcal{X}} q_{ij}(t) = 0,$$

то такую макросистему назовем внутренне равновесной. Для внутренне равновесной макросистемы все потоки наблюдаются только на границах макросистемы и ее окружения. Состояние внутреннего равновесия макросистемы является устойчивым, что означает выполнение принципа Ле Шателье: если на внутренне равновесную макросистему воздействовать извне, изменяя какое-либо из условий равновесия, то в такой макросистеме возникают потоки ресурсов, направленные в сторону противодействия изменению.

2. Уравнения состояния макросистемы

Макросистема описывается множеством экстенсивных и интенсивных переменных:

- экстенсивными переменными называются такие, что для любых двух макросистем X и Y (не обязательно находящихся в состоянии равновесия)

$$(6) \quad Q_{X \cup Y} = Q_X + Q_Y;$$

количества ресурсов в макросистемах являются экстенсивными переменными;

- интенсивными переменными ν называются такие, что для любых двух систем X и Y , находящихся в состоянии равновесия,

$$(7) \quad \nu_{X \cup Y} = \nu_X = \nu_Y.$$

УТВЕРЖДЕНИЕ 1. *Экстенсивные переменные удовлетворяют условию нейтрального эффекта масштаба: если количества всех ресурсов во всех подсистемах макросистемы увеличить в $n > 0$ раз, то величины потоков в макросистеме не изменятся.*

В том числе, такое пропорциональное увеличение ресурсов не выведет макросистему из состояния внутреннего равновесия, если до того система находилась в состоянии внутреннего равновесия. Доказательство этого утверждения приведем ниже.

СЛЕДСТВИЕ 1. *Зависимости интенсивных переменных от количества ресурсов, определяющих состояние макросистемы, в силу (7) должны быть однородными функциями нулевого порядка однородности.*

ДОКАЗАТЕЛЬСТВО УТВЕРЖДЕНИЯ 1. Множество экстенсивных переменных описывает состояние макросистемы. Так как объединение макросистем является макросистемой более высокого уровня, то каждая экстенсивная величина на этом уровне равна сумме соответствующих экстенсивных величин макросистем низкого уровня (подсистем). Естественное направление процессов ресурсообмена – это то, которое соответствует увеличению количества возможных состояний подсистем, соответствующих данному состоянию макросистемы в целом. Таким образом мы вводим целевую функцию макросистемы, как количественную величину. Эта величина аддитивна, следовательно, является экстенсивной. Такая величина $S = Q_0$, характеризующая целевую функцию системы, и остальные экстенсивные переменные функционально связаны: $S = \mathcal{S}(Q)$, $Q = (Q_1, \dots, Q_N)$. Это уравнение назовем уравнением состояния системы. Для замкнутой системы $\mathcal{S}(Q) \rightarrow \max$, что определяет самопроизвольное направление ресурсообмена.

Так как S – экстенсивная переменная, то $\mathcal{S}$ – однородная функция первого порядка однородности: при масштабировании системы в n раз или объединении n одинаковых макросистем, находящихся в равновесии:

$$(8) \quad nS = \mathcal{S}(nQ).$$

В соответствии с соотношениями Эйлера для однородных функций

$$(9) \quad \mathcal{S}(Q) = Q \nabla \mathcal{S} = \sum_{\nu=1}^N Q_{\nu} \frac{\partial \mathcal{S}}{\partial Q_{\nu}}.$$

Обозначим $\nu_\nu = \frac{\partial S}{\partial Q_\nu}$. Так как $S(Q)$ – однородная функция первого порядка однородности, то $\nu_\nu(Q)$ ($\nu = 1, \dots, N$) однородны нулевого порядка однородности, то есть являются интенсивными переменными: для любого значения $n > 0$ $\nu_\nu(nQ) = \nu_\nu(Q)$. Это значит, что пропорциональное увеличение ресурсов не выведет макросистему из состояния внутреннего равновесия, если до того система находилась в состоянии внутреннего равновесия. Таким образом, утверждение 1 доказано. $\square$

В предположении дифференцируемости функции $S(Q)$ и непрерывности ее частных производных, в соответствии с необходимым условием дифференцируемости функции существует полный дифференциал

$$(10) \quad dS = \sum_{\nu=1}^N \frac{\partial S}{\partial Q_\nu} dQ_\nu = \sum_{\nu=1}^N \nu_\nu dQ_\nu.$$

Из уравнения (10) вытекают два следствия:

СЛЕДСТВИЕ 2. Дифференцируя соотношение Эйлера (9), получим:

$$(11) \quad dS = \sum_{\nu=1}^N (\nu_\nu dQ_\nu + Q_\nu d\nu_\nu).$$

Сравнивая (10) и (11), получаем:

$$(12) \quad \sum_{\nu=1}^N Q_\nu d\nu_\nu = 0.$$

СЛЕДСТВИЕ 3. Процесс ресурсообмена между подсистемами X и Y (при положительном направлении потоков от X к Y) можно описать, используя уравнения (2) и (10), в виде:

$$(13) \quad \frac{dS}{dt} = \frac{dS_Y}{dt} + \frac{dS_X}{dt} = \sum_{\nu=1}^N (\nu_{Y_\nu} - \nu_{X_\nu}) q_{XY,\nu}.$$

В соответствии с принципом Ле Шателье между знаками разности интенсивных переменных и потока для каждого ресурса имеет место взаимосвязь

$$(14) \quad \text{sign}(\nu_{Y_\nu} - \nu_{X_\nu}) = \text{sign}(q_{XY,\nu}) \quad \forall \nu = 1, \dots, N.$$

Это означает, что в ходе самопроизвольного процесса в макросистеме величина $\sigma = \frac{dS}{dt} > 0$. Если продолжительность процесса бесконечно велика, то естественная эволюция макросистемы приводит ее к состоянию внутреннего равновесия, что и соответствует достижению максимального

значения S . Это равносильно утверждению, что в состоянии внутреннего равновесия S максимальна. Таким образом, в процессе естественного, не вынужденного, ресурсообмена величина энтропии замкнутой макросистемы S не может убывать, а $S(Q)$ является целевой функцией системы:

- S – экстенсивная переменная,
- для замкнутой системы $S \rightarrow \max$, что определяет самопроизвольное направление ресурсообмена, причем максимум соответствует состоянию равновесия;
- существует полный дифференциал dS ;
- скорость изменения (прироста) $\frac{dS}{dt}$ представляет собой скалярное произведение векторов движущих сил и потоков ресурсов.

Рассматривая совокупность значений экстенсивных параметров статистически значимого количества подсистем, формирующих некоторую макросистему, как распределение реализаций многомерной случайной величины, состояние динамического равновесия макросистемы соответствует максимальному значению энтропии. Энтропия аддитивна. Поэтому параметр S статистически соответствует энтропии распределения многомерной случайной величины. При описании модели макросистемы через показатели элементарных объектов [17] экстенсивная переменная с перечисленными свойствами определена, как энтропия системы, а скорость ее прироста – производство энтропии системы. При ресурсном описании макросистемы энтропия может рассматриваться, как запас информации о состоянии макросистемы.

В произвольном состоянии макросистемы распределение состояний подсистем также стремится к увеличению энтропии, что приводит к нормальному многомерному распределению параметров подсистем (при отсутствии ограничений на возможные значения случайного вектора при конечной дисперсии нормальное распределение соответствует максимальному значению энтропии).

Действительно, энтропия нормального распределения складывается из двух слагаемых: $S = S_0 + 0,5 \ln \det R$, где R – корреляционная матрица. Первое слагаемое соответствует полной независимости элементов системы и характеризует структуру макросистемы, а второе – описывает взаимосвязи в макросистеме. С увеличением корреляции $\rho_{\nu k}(\nu, k = 1, \dots, N)$ между потоками, что наблюдается при увеличении величины движущих сил ресурсообмена, второе слагаемое, всегда отрицательное при $R \neq \mathbf{E}$, ($\mathbf{E}$ – единичная матрица) уменьшается. Это поясняет утверждение о S как целевой функции, достигающей своего максимума при достижении

макросистемой состояния равновесия, и функции состояния, которая задает направление процессов в замкнутой системе, увеличивающее рассеяние параметров подсистем низкого уровня при упрощении структуры макросистемы на высоком уровне.

Интенсивные переменные $\nu = \nabla \mathcal{S}$ могут рассматриваться, как удельные потенциалы [18]. Их разность $\varphi_{XY} = \nu_Y - \nu_X$ представляет собой движущую силу процесса ресурсообмена. Потоки ресурсообмена направлены от подсистем с меньшими значениями интенсивных параметров к подсистемам с большими значениями интенсивных параметров.

Таким образом, уравнение (13) можно переписать следующим образом:

$$(15) \quad \sigma(\nu_X, \nu_Y) = \sum_{\nu=1}^N \varphi_\nu(\nu_X, \nu_Y) q_{XY, \nu}(\varphi(\nu_X, \nu_Y)).$$

Если движущая сила в течение процесса постоянна, то такой процесс будем называть стационарным. Выделим классы обратимых ($\varphi(\nu_X, \nu_Y) \rightarrow 0$) процессов и процессов минимальной диссипации, для которых среднее производство энтропии достигает своего минимума $\bar{\sigma}(\varphi) \rightarrow \min_\varphi |\overline{q_{XY}}(\varphi)| = \text{fix}$.

Функция состояния может быть задана в дифференциальном виде:

$$(16) \quad \delta \Phi = \sum_{\nu=1}^N F_\nu(Q) dQ_\nu.$$

В этом случае возникает вопрос об интегрируемости функции $\Phi(Q)$. Уравнение (16) является пфаффовой формой. Пфаффова форма называется голономной, если существует такой интегрирующий множитель $w(Q)$, что

$$(17) \quad w(Q) \delta \Phi = \sum_{\nu=1}^N \frac{\partial \mathcal{S}}{\partial Q_\nu} dQ_\nu = d\mathcal{S}, \quad \text{где} \quad \frac{\partial \mathcal{S}}{\partial Q_\nu} = w(Q) F_\nu(Q), \quad \nu = 1, \dots, N.$$

Пфаффова форма двух независимых переменных всегда голономна, то есть всегда существует интегрирующий множитель $w(Q)$. Однако, при $N > 2$ интегрирующий множитель существует, если выполняются условия голономности: для любых трех различных k, μ, ν

$$(18) \quad \begin{aligned} & F_k(Q) \left(\frac{\partial F_\mu}{\partial Q_\nu} - \frac{\partial F_\nu}{\partial Q_\mu} \right) + F_\mu(Q) \left(\frac{\partial F_\nu}{\partial Q_k} - \frac{\partial F_k}{\partial Q_\nu} \right) + \\ & + F_\nu(Q) \left(\frac{\partial F_k}{\partial Q_\mu} - \frac{\partial F_\mu}{\partial Q_k} \right) = 0. \end{aligned}$$

Эти условия получены из равенства вторых смешанных производных по любым парам переменных

$$\frac{\partial^2 \mathcal{S}}{\partial Q_\nu \partial Q_k} = \frac{\partial(w(Q)F_\nu(Q))}{\partial Q_k}$$

(соотношения Максвелла) и исключением из этих равенств интегрирующего множителя $w(Q)$.

Кроме условий (18) необходимо, чтобы все произведения $w(Q)F_\nu(Q)$ были однородными нулевого порядка однородности.

Градиент функции $\mathcal{S}(Q)$ определяет вектор интенсивных переменных макросистемы. При увеличении запаса ресурса величина S также возрастает, но все с меньшей скоростью (закон убывающей отдачи), так что для всех

$$\nu = 1, \dots, N \quad \frac{\partial \mathcal{S}}{\partial Q_\nu} = \nu_\nu > 0, \quad \frac{\partial^2 \mathcal{S}}{\partial Q_\nu^2} = \frac{\partial \nu_\nu}{\partial Q_\nu} < 0.$$

УТВЕРЖДЕНИЕ 2. Матрица Гессе $\mathcal{H}_S = \left(\frac{\partial^2 \mathcal{S}}{\partial Q_\nu \partial Q_k} \right)$ для однородных функций первого порядка однородности является отрицательно полуопределенной.

ДОКАЗАТЕЛЬСТВО. Действительно, в соответствии со Следствием 1 из соотношений Эйлера, дифференцируя обе части (12) по Q_k , получаем:

$$(19) \quad \forall k = 1, \dots, N : \quad \sum_{\nu=1}^N Q_\nu \frac{\partial^2 \mathcal{S}}{\partial Q_\nu \partial Q_k} = 0.$$

Для произвольного вектора x необходимые условия экстремума квадратичной формы по x

$$(20) \quad x^T \mathcal{H}_S x = \sum_{\nu=1}^N x_\nu^2 \frac{\partial^2 \mathcal{S}}{\partial Q_\nu^2} + \sum_{\nu=1}^{N-1} \sum_{k=\nu+1}^N 2x_\nu x_k \frac{\partial^2 \mathcal{S}}{\partial Q_\nu \partial Q_k} \rightarrow \max_x$$

$$(21) \quad \frac{\partial(x^T \mathcal{H}_S x)}{\partial x_\nu} = \sum_{k=1}^N x_k \frac{\partial^2 \mathcal{S}}{\partial Q_\nu \partial Q_k} = 0,$$

что соответствует точке максимума (в соответствии с законом убывающей отдачи) квадратичной формы $x_\nu = Q_\nu$, $\nu = 1, \dots, N$. Поскольку произведение $Q^T \mathcal{H}_S$ (19) равно нулю, то и $x^T \mathcal{H}_S x$ в точке максимума тоже равна нулю. Таким образом, при любом значении x : $x^T \mathcal{H}_S x \leq 0$, что и требовалось доказать. Отметим, что поскольку матрица Гессе симметрическая, то все ее собственные значения – действительные числа. $\square$

Отрицательная полуопределенность матрицы Гессе $\mathcal{H}_S$ соответствует выпуклости вверх функции $\mathcal{S}(Q)$ и унимодальности S как целевого параметра. При увеличении запасов ресурсов в макросистеме S увеличивается, а интенсивные параметры уменьшаются, снижая величину движущей силы ресурсообмена. В соответствии с (13) такое поведение интенсивных параметров приводит к тому, что при воздействии на макросистему, изменяющем условия ее внутреннего равновесия, процессы ресурсообмена направлены в сторону противодействия изменениям, таким образом выполняется принцип Ле Шателье.

Частные производные $\frac{\partial^2 S}{\partial Q_\nu^2}$ описывают насыщенность системы ресурсами, а вторые производные $\frac{\partial^2 S}{\partial Q_\nu \partial Q_k}$ определяют взаимозаменяемость и взаимодополнительность ресурсов в макросистеме. Если ресурсы взаимозаменяемые, то увеличение одного из ресурсов снижает удельный потенциал другого ресурса, если ресурсы взаимодополняющие, то увеличение одного из ресурсов, наоборот, увеличивает удельный потенциал другого ресурса.

3. Метрические свойства производства энтропии

Рассмотрим частный случай ресурсообмена в макросистеме, состоящей из двух подсистем X и Y , где потоки ресурсов линейно зависят от разностей интенсивных переменных подсистем (4):

$$(22) \quad q_{XY,\nu}(\nu_X, \nu_Y) = \frac{dQ_{Y\nu}}{dt} = \sum_{k=1}^N \alpha_{\nu k} \varphi_k(\nu_{X_k}, \nu_{Y_k}),$$

где $\varphi_k(\nu_{X_k}, \nu_{Y_k}) = \nu_{Y_k} - \nu_{X_k}$.

УТВЕРЖДЕНИЕ 3. Матрица $A = (\alpha_{\nu k})$ – матрица инфраструктурных коэффициентов, описывающая возможности ресурсообмена на границе между подсистемами, – является положительно полуопределенной симметрической матрицей (условия Онсагера).

ДОКАЗАТЕЛЬСТВО. 1. Правая часть уравнения (15) при условии (4) представляет собой квадратичную форму

$$(23) \quad \sigma(\nu_X, \nu_Y) = \varphi^T(\nu_X, \nu_Y) A \varphi(\nu_X, \nu_Y).$$

При любых значениях движущих сил ресурсообмена $\varphi(\nu_X, \nu_Y)$ производство энтропии неотрицательно. Следовательно, A – положительно полуопределенная матрица.

2. Дважды продифференцируем обе части уравнения (15):

$$(24) \quad \frac{\partial^2 \sigma}{\partial \varphi_\mu \partial \varphi_k} = \frac{\partial^2}{\partial \varphi_\mu \partial \varphi_k} \sum_{\nu=1}^N \varphi_\nu q_{XY,\nu}(\varphi) = \frac{\partial q_{XY,\mu}}{\partial \varphi_k} + \sum_{\nu=1}^N \varphi_\nu \frac{\partial^2 q_{XY,\nu}}{\partial \varphi_\mu \partial \varphi_k}.$$

Для линейной зависимости

$$q_{XY} = A\varphi : \frac{\partial^2 q_{XY,\nu}}{\partial \varphi_\mu \partial \varphi_k} = 0,$$

а значит,

$$(25) \quad \frac{\partial^2 \sigma}{\partial \varphi_\mu \partial \varphi_k} = \alpha_{\mu k} = \alpha_{k\mu}, \quad \text{матрица Гессе } \mathcal{H}_\sigma = A.$$

□

УТВЕРЖДЕНИЕ 4. *Производство энтропии является метрикой в пространстве стационарных процессов и может быть использовано для определения расстояния между процессами.*

1. Нулем в пространстве стационарных процессов является обратимые процессы, для которых $\sigma = 0$.
2. Расстояние между двумя процессами a и b определяется, как $\delta(a, b) = (\varphi_a - \varphi_b)^T A(\varphi_a - \varphi_b)$. Очевидно, что $\delta(a, a) = 0$; $\delta(a, b) = \delta(b, a)$.
3. Расстояние $\delta(a, b)$ удовлетворяет неравенству треугольника.

ДОКАЗАТЕЛЬСТВО. Действительно, так как A — положительно полуопределенная симметрическая матрица, все ее собственные значения λ неотрицательны. Квадратичная форма может быть сведена к сумме квадратов, умноженных на собственные значения [19]:

$$(26) \quad (\varphi_a - \varphi_b)^T A(\varphi_a - \varphi_b) = \sum_{\nu=1}^N \lambda_\nu (\varphi_{a\nu} - \varphi_{b\nu})^2,$$

а, поскольку для любой разности положительных величин квадрат разности всегда меньше суммы квадратов, выполняется неравенство

$$(27) \quad (\varphi_a - \varphi_b)^T A(\varphi_a - \varphi_b) \leq \varphi_a^T A \varphi_a + \varphi_b^T A \varphi_b.$$

Для любых трех процессов a, b, c : $\varphi_a - \varphi_b = (\varphi_a - \varphi_c) + (\varphi_c - \varphi_b)$, откуда:

$$(28) \quad \begin{aligned} (\varphi_a - \varphi_b)^T A(\varphi_a - \varphi_b) &\leq \\ &\leq (\varphi_a - \varphi_c)^T A(\varphi_a - \varphi_c) + (\varphi_b - \varphi_c)^T A(\varphi_b - \varphi_c). \end{aligned}$$

С учетом $\delta(c, b) = \delta(b, c)$ из (28) следует неравенство треугольника.

Таким образом, производство энтропии в макросистеме характеризует расстояние между стационарными процессами ресурсообмена, протекающими в такой системе. Такое расстояние является характеристикой для каждого процесса, если его рассчитать между этим процессом и соответствующим (в смысле фиксированных условий протекания) обратимым процессом. Например, если зафиксировать температуры источников тепла, то процессами, находящимися на минимальном расстоянии от обратимых являются процессы минимальной диссипации [20]. В свою очередь, расстояние между реальными процессами и процессами минимальной диссипации можно описать, как экстропию [21].

Для нестационарных процессов ресурсообмена необходимо определить траекторию процесса: изменение во времени всех параметров подсистем в ходе приближения к равновесному состоянию. Для линейной зависимости потоков от движущих сил (22) выведем дифференциальное уравнение, определяющее изменение движущих сил процесса ресурсообмена в макросистеме, состоящей из двух подсистем X и Y . При заданных начальных условиях $\varphi_{XY}(0) = \varphi_0$ это уравнение описывает все параметры подсистем.

Полные дифференциалы функций $\nu_{ik}(Q_i)$, $i \in \{X, Y\}$, $k = 1, \dots, N$ записываются в виде:

$$(29) \quad d\nu_{ik} = \sum_{\nu=1}^N \frac{\partial \nu_{ik}}{\partial Q_{i\nu}} dQ_{i\nu} \Rightarrow \begin{cases} \frac{d\nu_{Xk}}{dt} = - \sum_{\nu=1}^N \frac{\partial \nu_{Xk}}{\partial Q_{X\nu}} q_{XY}(\varphi_{XY}) \\ \frac{d\nu_{Yk}}{dt} = \sum_{\nu=1}^N \frac{\partial \nu_{Yk}}{\partial Q_{Y\nu}} q_{XY}(\varphi_{XY}). \end{cases}$$

Вычитая уравнения для подсистемы X из уравнений для подсистемы Y с учетом линейной зависимости потоков от движущих сил (22) и того, что

$$\frac{d\varphi_{XY}}{dt} = \frac{d\nu_Y}{dt} - \frac{d\nu_X}{dt},$$

получаем

$$(30) \quad \begin{cases} \frac{d\nu_X}{dt} = -\mathcal{H}_{SX} A \varphi_{XY} \\ \frac{d\nu_Y}{dt} = \mathcal{H}_{SY} A \varphi_{XY} \end{cases} \Rightarrow \frac{d\varphi_{XY}}{dt} = (\mathcal{H}_{SX} + \mathcal{H}_{SY}) A \varphi_{XY}.$$

Уравнение (30) вместе с начальными условиями определяет траекторию процесса ресурсообмена. Отметим, что несмотря на то, что матрицы Гессе для подсистем X и Y и матрица инфраструктурных коэффициентов симметричны, $\mathcal{H}_{SX} + \mathcal{H}_{SY}$ и A могут не коммутировать друг с другом, поэтому их произведение может быть не симметрической матрицей. $\square$

Математическая модель макросистемы в виде фрактального графа позволяет вывести феноменологические закономерности макросистем, не основываясь на свойствах элементарных объектов, формирующих макросистему. Это обобщает понятие макросистемы на совокупность взаимодействующих систем произвольной природы. Показано, что на множестве стационарных процессов можно ввести метрику; метрическими свойствами обладает производство энтропии в макросистеме.

Заключение

Фрактальная модель макросистемы позволяет вывести ее свойства и определить ее состояние на основе рекуррентного подхода (состояние макросистемы, состоящей из большого количества подсистем, описывается суммой значений векторов экстенсивных параметров подсистем при неизменном уравнении состояния), не описывая процессы взаимодействия на уровне элементарных объектов (раздел 1). Скорость изменения экстенсивных величин – это интенсивность процессов, локализуемых на границах подсистем, зависит от движущих сил, которые определяются разностью интенсивных переменных взаимодействующих подсистем. В свою очередь, интенсивные переменные определяются из уравнения состояния макросистемы, связывающего экстенсивные переменные. Свойства экстенсивных и интенсивных характеристик макросистем рассмотрены в разделе 2. Самопроизвольные процессы протекают в направлении увеличения энтропии подсистемы, а скорость увеличения энтропии всей системы (производство энтропии) является метрикой в пространстве стационарных процессов (доказательство метрических свойств приведено в разделе 3).

Модель макросистем, основанная на их фрактальных свойствах, может быть использована при анализе больших систем физической природы, например, многоконтурных одно- и двухфазных систем охлаждения, информационных систем, таких как рекомендательные системы, систем обработки больших данных, систем человеко-машинного взаимодействия при принятии решений, систем обучения и иных типов обмена семантической информацией в реальном и виртуальном типах пространства. Стационарные процессы в макросистемах любой природы могут быть описаны в рамках представленного подхода, что особенно актуально для сложных иерархических систем, включающих физические, экономические, социальные и информационные аспекты своего функционирования и развития.

Список литературы

- [1] Corr P., Plagnol A. *Behavioral Economics: The Basics*.— Taylor & Francis.— 2019.— ISBN 978-1138228917.— 266 pp.  ^{↑42}
- [2] Schor J. B. *What's wrong with consumer society? // Consuming Desires: Consumption, Culture, and the Pursuit of Happiness*, ed. R. Rosenblatt, Washington: Island Press.— 1999.— ISBN 978-1559635356.— Pp. 37–50.  ^{↑42}
- [3] Simon H. A. *Bounded rationality // Utility and Probability*, The New Palgrave, eds. J. Eatwell, M. Milgate, P. Newman, London: Palgrave Macmillan.— 1990.— ISBN 978-0-333-49541-4.— Pp. 15–18.  ^{↑42}
- [4] Nordli S., Todd P. *Ecological rationality: Bounded rationality in an evolutionary light // Routledge Handbook of Bounded Rationality*, Routledge International Handbooks, ed. R. Viale, London: Routledge.— 2020.— ISBN 9780367563943.— Pp. 313–323. ^{↑42}
- [5] Lyons B. J., Scott B. A. *Integrating social exchange and affective explanations for the receipt of help and harm: A social network approach // Organ. Behav. Hum. Decis. Processes*.— 2012.— Vol. 117.— No. 1.— Pp. 66–79.  ^{↑42}
- [6] Bronfenbrenner U. *The Ecology of Human Development. Experiments by Nature and Design*.— London: Harvard University Press.— 1979.— ISBN 9780674224575.— 330 pp. ^{↑42}
- [7] Perepelkin E. E., Sadovnikov B. I., Inozemtseva N. G. *Ψ-model of micro- and macrosystems*.— 2016.— 44 pp. arXiv: 1701.00469 [physics.gen-ph] ^{↑43}
- [8] Пьетронеро Л., Тозатти Э. (ред.) *Фракталы в физике*, Труды VI международного симпозиума по фракталам в физике (МЦТФ, Триест, Италия, 9-12 июля, 1985).— М.: Мир.— 1988.— ISBN 5-03-001295-8.— 672 с. ^{↑43}
- [9] Розеноэр Л. И. *Обмен и распределение ресурсов (обобщенный термодинамический подход). I // Автомат. и телемех.*— 1973.— № 5.— С. 115–132.  ^{↑44}
- [10] Цирлин А. М. *Методы оптимизации в необратимой термодинамике и микроэкономике*.— М.: Физматлит.— 2003.— ISBN 978-5-9221-0265-0.— 416 с. ^{↑44}
- [11] Амелькин С. А., Логунова Н. Ю. *Иерархические макросистемы как модели технологических бизнес-процессов в пищевой промышленности // Хранение и переработка сельхозсырья*.— 2018.— № 4.— С. 84–91.  ^{↑44}
- [12] Попков Ю. С. *Макросистемные модели пространственной экономики*.— М.: Ленанд.— 2015.— ISBN 978-5-9710-2052-3.— 240 с. ^{↑44}
- [13] Поплавский Р. П. *Термодинамика информационных процессов*.— М.: URSS.— 2021.— ISBN 978-5-9710-8976-6.— 255 с. ^{↑44}
- [14] Гусаренко С. В. *Когнитивно-семантические структуры дискурса: системное взаимодействие и семантическая энтропия*.— Флинта.— 2021.— ISBN 978-5-9765-4407-9.— 356 с. ^{↑44}
- [15] Иванова О. С., Амелькин С. А. *Экономическая эффективность продажи программного обеспечения при наличии пиратского рынка // Программные системы: теория и приложения*.— 2014.— Т. 5.— № 5.— С. 45–54.   ^{↑44}

- [16] Berry R. S., Kasakov V. A., Sieniutycz S., Szwaab Z., Tsirlin A. M. *Thermodynamic Optimization of Finite Time Processes*.– Chichester: Wiley.– 1999.– ISBN 978-0-471-96752-1.– 450 pp. ^{↑44}
- [17] Пригожин И., Кондепуди Д. *Современная термодинамика. От тепловых двигателей до диссипативных структур*.– М.: Мир.– 2002.– ISBN 5-03-003538-9.– 462 с. ^{↑51}
- [18] Лежнин С. И. *Термодинамические процессы*.– Новосибирск: Изд-во НГУ.– 2010.– 112 с.  ^{↑52}
- [19] Конвей Дж. *Квадратичные формы, данные нам в ощущениях*.– М.: МЦНМО.– 2008.– ISBN 978-5-94057-268-8.– 144 с. ^{↑55}
- [20] Цирлин А. М. *Процессы минимальной диссипации в необратимой термодинамике*.– М.: URSS.– 2022.– ISBN 978-5-507-44649-0.– 400 с. ^{↑56}
- [21] Martínás K. *Thermodynamics and sustainability a new approach by extropy* // Periodica Polytechnica: Chemical Engineering.– 1998.– Vol. 42.– No. 1.– Pp. 69–83.  ^{↑56}

Поступила в редакцию 07.12.2023;
одобрена после рецензирования 27.12.2023;
принята к публикации 28.12.2023;
опубликована онлайн 21.03.2024.

Рекомендовал к публикации

д.т.н. А. М. Цирлин

Информация об авторе:



Сергей Анатольевич Амелкин

к.т.н., ст.научн.сотр. Института программных систем имени А. К. Айламазяна РАН. Доцент кафедры информатики и системного анализа института Экономики, математики и информационных технологий Российской академии народного хозяйства и государственной службы при Президенте Российской Федерации



0000-0002-4004-7159

e-mail: amelkin@ist.education

Автор заявляет об отсутствии конфликта интересов.



Fractal model of macrosystems

Sergej Anatolyevich **Amelkin**

The Presidential Academy of National Economy and Public Administration, Moscow, Russia

Abstract. A mathematical model of a macrosystem of arbitrary nature in the form of a fractal graph is considered. This representation allows one to obtain phenomenological dependencies of macrosystems without being based on the properties of elementary objects that form the macrosystem. It is shown that a metric can be introduced on a set of stationary processes; the entropy production in the macrosystem has metric properties. (*In Russian*).

Key words and phrases: Macrosystem, entropy production, minimal dissipation processes

2020 *Mathematics Subject Classification:* 28A80; 28D20, 82B35

Acknowledgments: The study was supported by *Russian Science Foundation grant No. 23-21-00173* 

For citation: Sergej A. Amelkin. *Fractal model of macrosystems*. Program Systems: Theory and Applications, 2024, **15**:1(60), pp. 41–62. (*In Russ.*). https://psta.psir.ru/read/psta2024_1_41-62.pdf

References

- [1] P. Corr, A. Plagnol. *Behavioral Economics: The Basics*, Taylor & Francis, 2019, ISBN 978-1138228917, 266 pp.
- [2] J. B. Schor. “What’s wrong with consumer society?”, *Consuming Desires: Consumption, Culture, and the Pursuit of Happiness*, ed. R. Rosenblatt, Island Press, Washington, 1999, ISBN 978-1559635356, pp. 37–50. 
- [3] H. A. Simon. “Bounded rationality”, *Utility and Probability*, The New Palgrave, eds. J. Eatwell, M. Milgate, P. Newman, Palgrave Macmillan, London, 1990, ISBN 978-0-333-49541-4, pp. 15–18. 
- [4] S. Nordli, P. Todd. “Ecological rationality: Bounded rationality in an evolutionary light”, *Routledge Handbook of Bounded Rationality*, Routledge International Handbooks, ed. R. Viale, Routledge, London, 2020, ISBN 9780367563943, pp. 313–323.
- [5] B. J. Lyons, B. A. Scott. “Integrating social exchange and affective explanations for the receipt of help and harm: A social network approach”, *Organ. Behav. Hum. Decis. Processes*, **117**:1 (2012), pp. 66–79. 
- [6] U. Bronfenbrenner. *The Ecology of Human Development. Experiments by Nature and Design*, Harvard University Press, London, 1979, ISBN 9780674224575, 330 pp.
- [7] E. E. Perepelkin, B. I. Sadovnikov, N. G. Inozemtseva. *Ψ-model of micro- and macrosystems*, 2016, 44 pp. arXiv  1701.00469 [physics.gen-ph]
- [8] (MCTF, Triest, Italiya, 9-12 iyulya, 1985), 1988, 672 pp.; L. Pietronero, Tozatti (eds.) E.. *Fractals in Physics: Proceedings of the Sixth Trieste International Symposium on Fractals in Physics, ICTP, Trieste, Italy, July 9-12, 1985*, Trudy VI mezhdunarodnogo simpoziuma po fraktalam v fizike, North-Holland, Amsterdam, ISBN 5-03-001295-8
- [9] L. I. Rozonoer. “Resource Exchange and Allocation (A Generalized Thermodynamic Approach). I”, *Avtomat. i telex.*, 1973, no. 5, pp. 115–132 (in Russian). 
- [10] A. M. Cirilin. *Optimization 'ethods in Shreversible Ehermodynamics and 'icroeconomics*, Fizmatlit, M., 2003, ISBN 978-5-9221-0265-0 (in Russian), 416 pp.
- [11] Amel'kin S. A. , N. Yu. Logunova. “Hierarchical macrosystems as models of technological business processes in the food industry”, *Xranenie i pererabotka sel'xozsyr'ya*, 2018, no. 4, pp. 84–91 (in Russian). 
- [12] Yu. S. Popkov. *Macrosystem Models of Spatial Economics*, Lenand, 2015, ISBN 978-5-9710-2052-3 (in Russian), 240 pp.
- [13] R. P. Poplavskij. *Thermodynamics of Information Processes*, Nauka, M., 1981, ISBN 978-5-9710-8976-6 (in Russian), 255 pp.
- [14] S. V. Gusarenko. *Cognitive-Semantic Structures of Discourse: System Interaction and Semantic Entropy*, SKFU, Stavropol', 2017, ISBN 978-5-9765-4407-9 (in Russian), 366 pp.

- [15] O. S. Ivanova, S. A. Amel'kin. "Economic efficiency of software sales in the presence of pirate market", *Program Systems: Theory and Applications*, **5**:5 (2014), pp. 45–54 (in Russian).  
- [16] R. S. Berry, V. A. Kasakov, S. Sieniutycz, Z. Szwast, A. M. Tsirlin. *Thermodynamic Optimization of Finite Time Processes*, Wiley, Chichester, 1999, ISBN 978-0-471-96752-1, 450 pp.
- [17] D. Kondepudi, I. Prigogine. *Modern Thermodynamics: From Heat Engines to Dissipative Structures*, John Wiley & Sons, Ltd 10.1002/9781118698723, 2014, ISBN 9781118698723, 523 pp.
- [18] S. I. Lezhnin. *Thermodynamic Processes*, Izd-vo NGU, Novosibirsk, 2010, 112 pp. 
- [19] H. J. Conway. *The Sensual Quadratic Form*, Carus Mathematical Monographs, vol. **26**, Mathematical Association of America, 1967, ISBN 088385-030-3, 167 pp.
- [20] A. M. Cirilin. *Processes of Minimal Dissipation in Irreversible Thermodynamics*, URSS, M., 2022, ISBN 978-5-507-44649-0 (in Russian), 400 pp.
- [21] K. Martínás. "Thermodynamics and sustainability a new approach by extropy", *Periodica Polytechnica: Chemical Engineering*, **42**:1 (1998), pp. 69–83. 



Обоснование методов ускорения гнёзд циклов итерационного типа

Елена Анатольевна **Метелица**✉

Южный Федеральный Университет, Ростов-на-Дону, Россия

✉metelica@sfedu.ru

Аннотация. Рассматривается ускорение итерационных алгоритмов, которые встречаются при решении задач математической физики, математического моделирования, обработки изображений и других. В программной реализации таких алгоритмов лежат гнёзда циклов (участки программы, состоящие из вложенных циклов). Такие гнёзда циклов ускоряются при помощи комбинации оптимизирующих преобразований, включающих тайлинг, метод гиперплоскостей и распараллеливание на общую память. Обосновывается эквивалентность комбинации используемых преобразований программ.

Предлагается и обосновывается метод изменения порядка обхода тайла. Метод даёт ускорение за счёт увеличения количества чтений данных из регистров, вместо чтений из более медленной памяти. С учётом этого метода получена формула вычисления оптимальных размеров тайлов.

Представленной в статье цепочкой преобразований достигается ускорение в 1.4 раза большее, чем в известном алгоритме оптимизации, реализованном в системе PLUTO. Приводятся численные эксперименты, которые в некоторых случаях на процессоре с 8 ядрами демонстрируют ускорение относительно исходных последовательных программ более чем на порядок. Результаты статьи могут использоваться для ручной и автоматизированной оптимизации программ.

Ключевые слова и фразы: тайлинг, метод гиперплоскостей, распараллеливание, общая память, гнёзда циклов итерационного типа

Благодарности: Автор благодарен д.т.н. Б.Я. Штейнбергу за руководство работой и Ар.В. Климову за внимание и интерес к работе.

Для цитирования: Метелица Е. А. Обоснование методов ускорения гнёзд циклов итерационного типа // Программные системы: теория и приложения. 2024. Т. 15. № 1(60). С. 63–94. https://psta.psiras.ru/read/psta2024_1_63-94.pdf

Введение

Рассматривается ускорение алгоритмов, которые встречаются при решении задач математической физики, математического моделирования, обработки изображений и других. В программной реализации рассматриваемых алгоритмов лежат гнёзда циклов (участки программы, состоящие из вложенных циклов) итерационного типа. Отдельные преобразования, которые используются для оптимизации таких гнёзд циклов описаны во многих работах. В частности, теория оптимизации гнёзд циклов представлена в работах М. Lam [1], М. Wolf [1, 2], У. Banerjee [2], У. Bondhugula [5], Л. Lamport [4] и основывается на преобразованиях пространства итераций в форме полиэдра. Автоматизация некоторых ключевых преобразований, используемых в данной работе, реализована в некоторых системах, таких как Polly LLVM¹, PLUTO², PolyMage [5] и др. В работах У. Bondhugula приводится подход к оптимизации гнёзд циклов итерационного типа. Этот подход автоматизирован в системе PLUTO. В статье [14] рассматривается подход к распараллеливанию гнёзд циклов итерационного типа с использованием алгоритма, основанного на преобразованиях «скошенный тайлинг» и «метод гиперплоскостей».

Данная работа рассматривает представленный в [10], [14] алгоритм распараллеливания гнёзд циклов итерационного типа для вычислительных систем с общей памятью. Расширяет его дополнительными преобразованиями (перестановка циклов внутри тайла, линейаризация [16], вынос инвариантных выражений) и даёт теоретическое обоснование его эквивалентности.

Описывается метод изменения обхода точек тайла, повышающий временную локальность данных. Изменение обхода точек тайла достигается за счёт перестановки циклов. Обосновывается целесообразность использования этого метода. Приводится модель вычисления оптимальных размеров тайлов, которая подтверждается численными экспериментами.

Численные эксперименты выполнения программ, преобразованных при помощи алгоритма, описанного в данной работе, демонстрируют ускорение относительно их исходных последовательных версий. Например, ускорение алгоритма Гаусса-Зейделя для численного решения обобщенной задачи Дирихле уравнения Лапласа составляет 16.32 раза. Достигнутый результат превосходит в 1.4 раза ускорение, полученное при помощи

¹Polly - Polyhedral optimizations for LLVM.  <https://polly.llvm.org/>.

²PLUTO - An automatic parallelizer and locality optimizer for affine loop nests.

 <https://pluto-compiler.sourceforge.net/>.

оптимизирующей системы PLUTO. Ускорение достигается посредством дополнительных преобразований, повышающих временную локальность данных. Также приводится ускорение, достигаемое описанным алгоритмом с использованием дополнительных преобразований.

Результаты статьи могут использоваться при разработке быстрых программ и оптимизирующих компиляторов.

1. Преобразования программ для ускорения целевых алгоритмов

Приведём некоторые определения для лучшего понимания результатов работы. Будут использоваться описания известных типов зависимости: истинная, антитезис, циклически порожденная зависимость и другие термины теории преобразований программ [12].

Вхождением переменной будем называть всякое появление переменной в тексте программы вместе с тем местом (строка и позиция имени переменной в строке) в программе, в котором эта переменная появилась. Всякому вхождению (при конкретном значении индексного выражения для массивов) соответствует обращение к некоторой ячейке памяти. Например, на рисунке 1 изображен текст программы, в которой присутствуют три вхождения массива *u*.

Генератор (out, output) – вхождение, при котором происходит запись в ячейку памяти.

Использованиями (in, input) называются остальные вхождения переменных.

Граф информационных зависимостей – это ориентированный граф, вершины которого соответствуют вхождениям переменных, а дуга (u, v) соединяет пару вершин, если эти вхождения порождают информационную зависимость (обращаются к одной и той же ячейке памяти), причем вхождение *u* раньше обращается к общей ячейке памяти, чем *v*, и хотя бы одно из этих вхождений является генератором.

Информационная зависимость между вхождениями называется *циклически независимой (loop independent dependence)*, если эти вхождения обращаются к одной и той же ячейке памяти на одной и той же итерации цикла. Иначе зависимость называется *циклически порожденной (loop carried dependence)*, а цикл называется *создающим* такую зависимость.

Носителем информационной зависимости называют цикл, создающий эту зависимость, которая является циклически порожденной. Носители нумеруются от внешнего цикла к внутреннему, начиная с нуля. Для одной

циклически порождённой зависимости может быть несколько циклов, которые её создают. Обозначим множество носителей - *carriers*.

Рассмотрим граф информационных зависимостей (рисунок 1) гнезда циклов. Для каждой дуги графа вычислим множество зависимостей:

$$\begin{aligned} \text{carriers}(u_{i,j}, u_{i,j}) &= (0), & \text{carriers}(u_{i,j}, u_{i-2,j-1}) &= (0, 1), \\ \text{carriers}(u_{i-2,j-1}, u_{i,j}) &= (0), & \text{carriers}(u_{i,j}, u_{i+1,j-1}) &= (0), \\ \text{carriers}(u_{i+1,j-1}, u_{i,j}) &= (0, 1). \end{aligned}$$

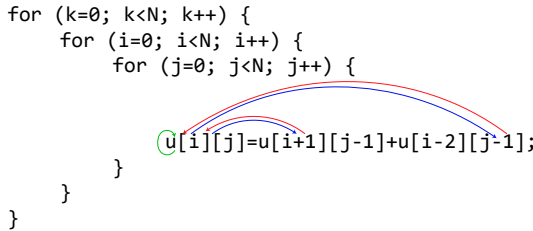


Рисунок 1. Граф информационных зависимостей трехмерного гнезда циклов

Рассмотрим гнездо из $n + 1$ вложенных друг в друга циклов. Прономеруем операторы циклов в этом гнезде в соответствии с порядком вложенности, начиная с самого внешнего цикла. Обозначим счетчик j -го цикла - I_j , где j от 0 до n . Множество значений, которые может принимать вектор счетчиков циклов $I = (I_0, I_1, \dots, I_n)$ в указанном гнезде, называется пространством итераций данного гнезда.

Если сделать раскрутку всех циклов гнезда, то получим код без циклов, состоящий из множества блоков, каждый из которых является копией тела исходного гнезда циклов. Каждая такая копия тела гнезда циклов соответствует набору значений счетчиков циклов гнезда.

Будем обозначать копию вхождения v в раскрутку исходного гнезда циклов при значениях счетчиков $I' = (I'_0, I'_1, \dots, I'_n)$ следующим образом: $v(I'_0, I'_1, \dots, I'_n)$ или $v(I')$ и называть представителем данного вхождения.

Для анализа и преобразований многомерных циклов используются решетчатые графы [15]. Приведём определение элементарного решетчатого графа из статьи [11].

Пусть X - это m -мерный массив, тогда будем рассматривать вхождение $X_{F(I)}$, где $F(I)$ - отображение пространства итераций в n -мерное целочисленное пространство индексов массива X , а I - точка пространства итераций. Далее отображение F будем рассматривать аффинным.

Например, для вхождения $X_{i_1-i_2, i_3+2}$ при размерности пространства итераций $n = 3$ и размерности массива X $m = 2$, отображение будет иметь вид: $F(i_1, i_2, i_3) = (i_1 - i_2, i_3 + 2)$.

Пусть в гнезде циклов имеется пара зависимых вхождений $X_{F(I)}$ и $X_{G(J)}$, и этой паре соответствует некоторая дуга графа информационных зависимостей $(X_{F(I)}, X_{G(J)})$ (листинг 1). Для этой дуги определим элементарный решетчатый граф (F, G) . Вершины решетчатого графа – точки пространства итераций гнезда циклов. Пусть (I, J) – пара точек пространства итераций и $I < J$ (лексикографический порядок, то есть I раньше J). Дуга (I, J) принадлежит графу, если $F(I) = G(J)$ и для любой точки пространства итераций K , для которой $K < J$ и $F(K) = G(J)$, выполняется $K \leq I$. Иными словами, вершина I является лексикографическим максимумом множества всех таких точек K , для которых $F(K) = G(J)$ и $K < J$. Если для любой переменной в теле гнезда циклов есть не более одного генератора, то существует взаимно однозначное соответствие между дугами графа (F, G) и не ложными дугами графа информационных зависимостей раскрутки всех циклов гнезда [11].

Листинг 1. Гнездо циклов с вхождениями массива X

```

for (int  $I = a$ ;  $I < b$ ;  $I++$ ) {
     $X[F(I)] = \dots$ 
     $\dots = X[G(I)]$ 
}
    
```

В каждую точку пространства итераций входит не более одной дуги элементарного решетчатого графа, соответствующей дуге $(X[F(I)], X[G(J)])$ графа информационных зависимостей.

Далее будем рассматривать решетчатый граф программы как объединение элементарных решетчатых графов.

Если в гнезде n циклов, то пространство итераций, т. е. множество наборов значений векторов счетчиков циклов после раскрутки образует подмножество целочисленной решетки n -мерного пространства. Поэтому графы информационных зависимостей между копиями тела гнезда циклов, естественно, называть решетчатыми [11].

Обозначим $u(i_1, i_2, \dots, i_d)$ представитель вхождения u в теле гнезда циклов с координатами $(i_1, i_2, \dots, i_d)$. Также $u(i_1, i_2, \dots, i_d)$ будем обозначать семейство вершин решетчатого графа, соответствующих вхождению u .

ПРИМЕР 1. Рассмотрим гнездо циклов.

```
for (i=1; i<=3; i=i+1)
  for (j=1; j<=3; j=j+1)
    a[j] = a[j+1];
```

Для этого гнезда циклов вхождению a_{j+1} соответствуют 9 представителей в раскрутке обоих циклов этого гнезда, каждой из которых соответствует точка пространства итераций решетчатого графа. Этому вхождению соответствует семейство представителей $a(i, j)$. При этом, например, $a(2, 3) = \langle a_4 \rangle$, $a(3, 1) = \langle a_2 \rangle$.

Результат полной раскрутки гнезда циклов

```
// i = 1
a[1]=a[2]; // j = 1
a[2]=a[3]; // j = 2
a[3]=a[4]; // j = 3
// i = 2
a[1]=a[2]; // j = 1
a[2]=a[3]; // j = 2
a[3]=a[4]; // j = 3
// i = 3
a[1]=a[2]; // j = 1
a[2]=a[3]; // j = 2
a[3]=a[4]; // j = 3
```

Решетчатый граф, построенный ОРС для примера 1, показан на рисунке 2.

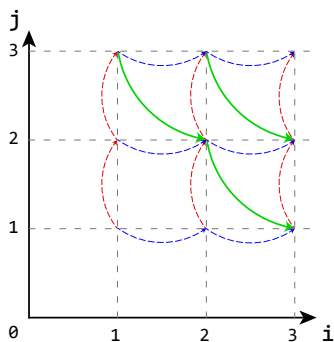


РИСУНОК 2. Решетчатый граф примера 1. Красная дуга - дуга антизависимости, зелёная - истинной зависимости, синяя - выходной зависимости

Скашивание (Loop Skewing) [7] – преобразование, которое изменяет пространство итераций гнезда циклов: вектор счетчиков нового гнезда циклов I' равен вектору счетчиков исходного гнезда I умноженного на нижнетреугольную матрицу skew с единицами на главной диагонали: $I' = \text{skew} \times I$. Число f , которое находится в строке m и в столбце k ($k < m$) будем называть параметром скашивания цикла номер k относительно цикла под номером m .

Матрица skew преобразования «скашивание циклов» для двумерного гнезда циклов имеет вид:

$$\text{skew} = \begin{pmatrix} 1 & 0 \\ f & 1 \end{pmatrix}.$$

Преобразование «скашивание циклов» (Loop Skewing) является эквивалентным.

На рисунке 3 представлено изменённое пространство итераций двумерного гнезда циклов после скашивания.

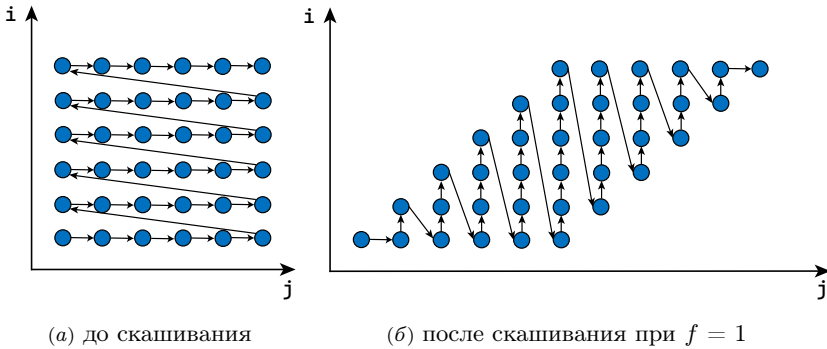


РИСУНОК 3. Последовательность выполнения итераций двумерного гнезда циклов

Понятие *вектор расстояния (distance vector)* [6] относится к дуге графа информационных зависимостей. Вектор расстояния зависимости используется для определения параметров скашивания гнезда циклов для последующего корректного выполнения прямоугольного тайлинга.

Приведём определение вектора расстояний (distance vector) дуги информационной зависимости.

Пусть в графе информационных зависимостей гнезда из n циклов есть дуга (u, v) . Тогда существует такая пара представителей $u(i'_1, i'_2, \dots, i'_n)$, $v(i''_1, i''_2, \dots, i''_n)$ для которых есть дуга в решетчатом графе. Заметим, что таких пар представителей может быть много. Рассмотрим разность

векторов счетчиков циклов $(i''_1 - i'_1, i''_2 - i'_2, \dots, i''_n - i'_n)$. Если значение этой разности одинаково для всех таких дуг $(u(i'_1, i'_2, \dots, i'_n), v(i''_1, i''_2, \dots, i''_n))$ соответствующих (u, v) , то будем называть эту разность *вектором расстояния* (*distance vector*) дуги информационной зависимости (u, v) .

Гнездо циклов итерационного типа — это такое гнездо циклов [13], которое обладает следующими свойствами. Внешний цикл отвечает за обход итераций, а внутренние за расчёт шаблона вычислений, который повторяется на итерациях внешнего гнезда. В данной работе рассматриваются такие гнезда циклов итерационного типа, которые являются тесными и в тело которых входят только операторы присваивания, безындексные переменные, константы, массивы в стиле языка C; массивы имеют линейные индексные выражения вида $i + \text{целочисленная константа}$, где i счетчик одного из циклов. Более точно, в теле итерационного гнезда циклов $(\text{LoopBody}(i_1, i_2, \dots, i_n))$ могут присутствовать генераторы вида $a_{i_1 - c_1, i_2 - c_2, \dots, i_n - c_n}$, (где $c_1, c_2, \dots, c_n$ — целочисленные константы), в которых последовательность индексов массива $(i_1 - c_1, i_2 - c_2, \dots, i_n - c_n)$ соответствует последовательности счетчиков гнезда циклов $(i_1, i_2, \dots, i_n)$. Тогда использования этой же переменной u в правой части оператора присваивания должны иметь вид: $a_{i_1 - d_1, i_2 - d_2, \dots, i_n - d_n}$, где $d_1, d_2, \dots, d_n$ — целочисленные константы. Общий вид такого гнезда циклов представлен в листинге 2. Из-за наличия информационных зависимостей ни один цикл данного гнезда не может выполняться параллельно. Поэтому перед распараллеливанием надо выполнить преобразование.

Листинг 2. Гнездо циклов итерационного типа

```

for (int i0 = 0; i0 < itera; i0++)
  for (int i1 = a1; i1 < b1; i1++)
    for (int i2 = a2; i2 < b2; i2++)
      ...
    for (int in = an; in < bn; in++) {
      LoopBody(i1, i2, ..., in);
    }

```

Сформулируем понятие вектора расстояния для информационных зависимостей в гнезде циклов итерационного типа. В случае гнезда итерационного типа массивы, генераторы которых входят в тело гнезда, имеют размерность на единицу меньше, чем количество циклов гнезда.

Пусть размерность гнезда циклов итерационного типа равна $n + 1$, u и v — вхождения n -мерного массива a , образующие дугу информационной зависимости $(u, v) = (a_{i_1 - c_1, i_2 - c_2, \dots, i_n - c_n}, a_{i_1 - d_1, i_2 - d_2, \dots, i_n - d_n})$. Счетчик

внешнего цикла (имеющего порядковый номер 0), вычисляющего итерации алгоритма, не входит в индексные выражения массивов (согласно определению гнезда циклов итерационного типа). Тогда существует такая пара представителей, для которых есть дуга в решетчатом графе $(u(i'_0, i'_1, i'_2, \dots, i'_n), v(i''_0, i''_1, i''_2, \dots, i''_n))$.

Рассмотрим вектор длины $n + 1$ разностей векторов счетчиков циклов $(i''_0 - i'_0, i''_1 - i'_1, i''_2 - i'_2, \dots, i''_n - i'_n)$. Все координаты, начиная с первой, не зависят от выбора пары представителей u, v . А нулевая координата, всегда неотрицательна (поскольку дуга информационной зависимости направлена от u к v). Тогда определим для гнезда циклов итерационного типа вектор расстояний дуги информационной зависимости (u, v) формулой

$$\begin{aligned} \text{dist}(a_{i_1 - c_1, i_2 - c_2, \dots, i_n - c_n}, a_{i_1 - d_1, i_2 - d_2, \dots, i_n - d_n}) \\ = (1, d_1 - c_1, d_2 - c_2, \dots, d_n - c_n). \end{aligned}$$

Рассмотрим пример 1. Для дуги истинной информационной зависимости (a_j, a_{j+1}) вектор расстояний $= (1, -1)$.

Скашивание меняет векторы расстояний информационных зависимостей: матрица преобразования «скашивание циклов» для примера 1 имеет вид

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix},$$

вектор расстояний для дуги истинной информационной зависимости (a_j, a_{j+1}) после скашивания будет иметь вид

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Метод гиперплоскостей (Loop Wavefront) [4] — это преобразование гнезда циклов, которое меняет обход точек пространства итераций следующим образом. Пространство итераций разбивается на параллельные гиперплоскости (с общим вектором нормали). Гиперплоскости обходятся в том порядке, который указывает вектор нормали. Точки, находящиеся на одной гиперплоскости, обходятся в лексикографическом порядке.

В данной работе, для преобразования гнёзд итерационного типа, метод гиперплоскостей будет применяться поэтапно к парам соседних циклов.

Преобразование метод гиперплоскостей является комбинацией преобразований скашивание и перестановка циклов [1]. Применяется к паре соседних циклов (один из которых непосредственно вложен во второй). Условие эквивалентности метода гиперплоскостей сводится к условию эквивалентности перестановки циклов после скашивания. Если все информационные зависимости в гнезде циклов имеют векторы

расстояний и эти векторы не имеют отрицательные координаты, то перестановка циклов эквивалентна [2].

Метод гиперплоскостей выполняется таким образом, чтобы точки находящиеся на одной гиперплоскости были информационно независимы, что позволяет выполнять их параллельно.

Тайлинг (Loop Tiling, Loop Blocking) [1, 8] – это преобразование программ, которое разбивает пространство итераций исходного гнезда цикла параллельными плоскостями (гиперплоскостями) на блоки (тайлы) меньшего размера и просматривает точки пространства итераций поблочно. Прямоугольный тайлинг – это тайлинг у которого блоки являются прямоугольными параллелепипедами, грани которых перпендикулярны соответствующим координатным осям, см. рисунок 4.

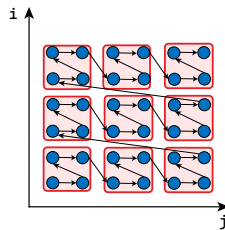


Рисунок 4. Пространство итераций двумерного гнезда циклов после тайлинга. Дуги показывают порядок выполнения точек пространства итераций. Точки, принадлежащие одному тайлу обведены.

Чтобы прямоугольный тайлинг оставался эквивалентным преобразованием должны отсутствовать информационные зависимости, для которых вектор расстояний имеет отрицательные координаты [2].

Листинг 3. Гнездо циклов полученное после применения тайлинга с размерами блоков d_1 , d_2 к двумерному гнезду циклов

```
for ( ii = 0; ii < N1/d1; ii++)
  for ( jj = 0; jj < N2/d2; jj++)
    for ( i = ii*d1; i < (ii+1)*d1; i++)
      for ( j = jj*d2; j < (jj+1)*d2; j++)
        LoopBody(i, j);
```

Скошенный тайлинг (Skewed Loop Tiling) [1] – преобразование, которое является комбинацией скашивания и прямоугольного тайлинга (см. рисунок 5).

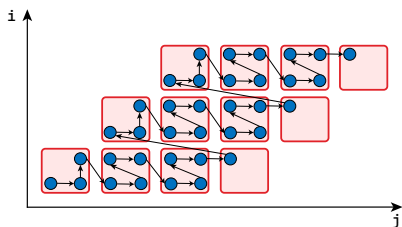


РИСУНОК 5. Пространство итераций двумерного гнезда циклов после скошенного тайлинга. Дуги показывают порядок выполнения точек пространства итераций. Точки, принадлежащие одному тайлу, обведены.

Листинг 4. Гнездо циклов полученное после применения скошенного тайлинга с размерами блоков $d1$, $d2$ и параметром скашивания $= 1$ к двумерному гнезду циклов

```

for ( ii=0; ii < (N-initN -1)/d1+1; ii++)
  for ( jj=0; jj < ((M-initM)+(N-initN)-2)/d2+1; jj++)
    for ( i=ii*d1; i<min((ii+1)*d1, N-initN); i++)
      for ( jjj=max(i, jj*d2);
           jjj < min((jj+1)*d2, (M-initM)+i); jjj++)
        LoopBody(i, jjj - i);

```

Гнёзда циклов, для которых прямоугольный тайлинг не применим, можно преобразовывать, используя скошенный тайлинг, поскольку за счёт скашивания изменяются векторы расстояний информационных зависимостей.

Инвариант цикла – это выражение, значение которого не меняется при каждом прохождении цикла. Предварительное вычисление инвариантов до выполнения цикла может увеличить производительность.

Вынос инвариантных выражений из циклов – это преобразование, которое находит инварианты в цикле и выносит их за пределы цикла.

2. Алгоритм оптимизации гнёзд циклов итерационного типа

Опишем процесс оптимизации гнёзд циклов с применением приведённых ранее преобразований. На вход алгоритму подается программа на

языке C, содержащая тесное гнездо циклов итерационного типа.

Алгоритм оптимизации итерационных гнёзд циклов:

- (1) Вычисление параметров скашивания на основе анализа информационных зависимостей;
- (2) Применение скашивания (если необходимо) и тайлинга;
- (3) Перестановка циклов внутри тайла для повышения временной локальности данных;
- (4) Применение метода гиперплоскостей;
- (5) Применение прагм OpenMP для параллельного выполнения тайлов.

Результатом применения алгоритма является преобразованная программа на языке C.

Автоматизация описанного алгоритма реализована в Оптимизирующей распараллеливающей системе (ОРС)³.

2.1. Вычисление параметров преобразований на основе анализа информационных зависимостей

Для определения подходящего алгоритма тайлинга (прямоугольный или скошенный) проводится анализ дуг графа информационных зависимостей. Для каждой дуги вычисляется вектор расстояний (distance vector). Если хотя бы в одном векторе расстояний присутствуют отрицательные координаты, то перед применением тайлинга выполняется скашивание.

2.2. Вычисление матрицы скашивания

Пусть на графе информационных зависимостей некоторого гнезда циклов присутствуют k дуг информационной зависимости. Анализируются векторы расстояний этих дуг.

Рассмотрим дугу информационной зависимости (u, v) .

Пусть n – количество циклов в гнезде. Определим массив $\text{carriers}(u, v)$ носителей [12] дуги (u, v) . Поскольку рассматриваются тесные гнёзда циклов, все вхождения переменных находятся в самом внутреннем цикле. Это означает, что для информационных зависимостей, которые порождены этими вхождениями, все циклы гнезда могут являться их носителями. Это означает, что элементы массива $\text{carriers}(u, v)$ принимают значения от нуля до $n - 1$, нумерация циклов производится от внешнего к внутреннему.

³Оптимизирующая распараллеливающая система:  <http://www.ops.rsu.ru/>.

Пусть $\text{dist}_m = \text{dist}(u, v)_m$, где m от 0 до $n - 1$, номер отрицательной координаты вектора расстояний дуги (u, v) . В таком случае нужно выполнить скашивание циклов с номерами, которые являются элементами массива $\text{carriers}(u, v)$, относительно цикла с номером m ; параметр скашивания $f = |\text{dist}_m|$. В результате получим композицию скашиваний. Матрица этой композиции – s . Пусть S множество таких матриц скашиваний, построенных для каждой дуги информационной зависимости. Тогда сформируем результирующую матрицу скашивания skew элементы которой вычисляются по формуле: $\text{skew}_{i,j} = \max(s^1_{i,j}, s^2_{i,j}, \dots, s^k_{i,j})$, где $s^1, s^2, \dots, s^k$ принадлежат множеству S .

ПРИМЕР 2. Рассмотрим вычисление матрицы скашивания для гнезда циклов, представленного на рисунке 1. В данном гнезде присутствуют 5 дуг графа информационных зависимостей, которые могут повлиять на эквивалентность выполнения тайлинга: две дуги антизависимости, две дуги истинной зависимости и одна дуга выходной зависимости. Вычислим для них векторы расстояний (dist) и массивы носителей зависимости (carriers):

Дуга информационной зависимости	Вектор расстояний информационной зависимости (dist)	Массив носителей информационной зависимости (carriers)
$(u_{i,j}, u_{i,j})$	$(1, 0, 0)$	(0)
$(u_{i,j}, u_{i-2,j-1})$	$(1, 2, 1)$	$(0, 1)$
$(u_{i-2,j-1}, u_{i,j})$	$(1, -2, -1)$	(0)
$(u_{i,j}, u_{i+1,j-1})$	$(1, -1, 1)$	(0)
$(u_{i+1,j-1}, u_{i,j})$	$(1, 1, -1)$	$(0, 1)$

Векторы расстояний дуг

$$(u_{i-2,j-1}, u_{i,j}), \quad (u_{i,j}, u_{i+1,j-1}), \quad (u_{i+1,j-1}, u_{i,j})$$

имеют отрицательные координаты. Вычислим для каждой дуги матрицу скашивания.

Вектор расстояний дуги $(u_{i-2,j-1}, u_{i,j})$ имеет две отрицательные координаты

$$\text{dist}(u_{i-2,j-1}, u_{i,j})_1 = -2, \quad \text{dist}(u_{i-2,j-1}, u_{i,j})_2 = -1.$$

Носитель информационной зависимости – цикл под номером 0. Значит нужно выполнить два скашивания: цикла под номером 0 относительно цикла под номером 1 с параметром скашивания 2 (матрица скашивания

$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$); цикла под номером 0 относительно цикла под номером 2 с параметром скашивания 1 (матрица скашивания $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$). Тогда матрица композиции скашиваний равняется:

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

После скашивания с такой матрицей все координаты вектора расстояний рассматриваемой дуги станут неотрицательными:

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ -2 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

ЗАМЕЧАНИЕ 1. Для любой матрицы скашивания, у которой элементы ниже главной диагонали не меньше, чем элементы полученной матрицы скашивание тоже приведёт к неотрицательным координатам вектора расстояний.

Вектор расстояний дуги $(u_{i,j}, u_{i+1,j-1})$ имеет одну отрицательную координату $\text{dist}(u_{i,j}, u_{i+1,j-1})_1 = -1$. Носитель информационной зависимости – цикл под номером 0. Значит нужно выполнить скашивание цикла под номером 0 относительно цикла под номером 1 с параметром скашивания 1. Матрица скашивания $= \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. После скашивания с такой матрицей все координаты вектора расстояний рассматриваемой дуги станут неотрицательными:

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}.$$

Вектор расстояний дуги $(u_{i+1,j-1}, u_{i,j})$ имеет одну отрицательную координату $\text{dist}(u_{i+1,j-1}, u_{i,j})_2 = -1$. Носители информационной зависимости – циклы под номером 0 и 1. Значит нужно выполнить два скашивания: цикла под номером 0 относительно цикла под номером 2 с параметром скашивания 1 (матрица скашивания $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$); цикла под номером 1 относительно цикла под номером 2 с параметром скашивания 1 (матрица скашивания $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$); Тогда матрица композиции скашиваний $= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$.

После скашивания с такой матрицей все координаты вектора расстояний рассматриваемой дуги станут неотрицательными:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Получим результирующую матрицу скашивания выбрав максимальные значения для каждой пары координат матриц

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} : \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}.$$

ЗАМЕЧАНИЕ 2. Скашивание должно применяться по строкам матрицы скашивания (по строкам, сверху-вниз). Так например, для матрицы скашивания $\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$:

- (1) применяется скашивание с матрицей $\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$;
- (2) применяются скашивания с матрицами $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$ и $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$ в произвольном порядке.

2.3. Применение тайлинга

Основываясь на анализе информационных зависимостей применяется прямоугольный тайлинг в комбинации со скашиванием, если требуется.

Параметрами прямоугольного тайлинга является целочисленный массив размеров тайлов (tile_sizes), который задаётся пользователем. Размер массива tile_sizes равен количеству циклов гнезда.

Параметры для скошенного тайлинга: целочисленный массив размеров тайлов (tile_sizes), матрица скашивания (skew). В начале к циклам гнезда применяется композиция скашиваний, определяемая матрицей skew. Затем применяется прямоугольный тайлинг с размерами тайлов (tile_sizes).

2.4. Перестановка циклов внутри тайла для повышения временной локальности данных

Для повышения временной локальности данных при вычислении тайла используется преобразование «перестановка циклов» (Loop Interchange). Его условия эквивалентности описаны в [2]. В данном случае, после применения тайлинга, условия эквивалентности выполняются (т.к. все векторы расстояний не будут иметь отрицательных координат).

Выбираются циклы, отвечающие за обход тайла. Определяется самый вложенный цикл (innermost loop) и переставляется с вышестоящими посредством преобразования «перестановка циклов», чтобы выбранный цикл стал внешним. До перестановки обход точек тайла проводился по целым точкам сечений тайла параллельным нижней грани. После перестановки обход производится по целым точкам сечений тайла параллельным боковой грани ближайшей к началу координат.

На рисунке (6) проиллюстрировано изменение порядка обхода итераций, которое получается после применения описанной перестановки.

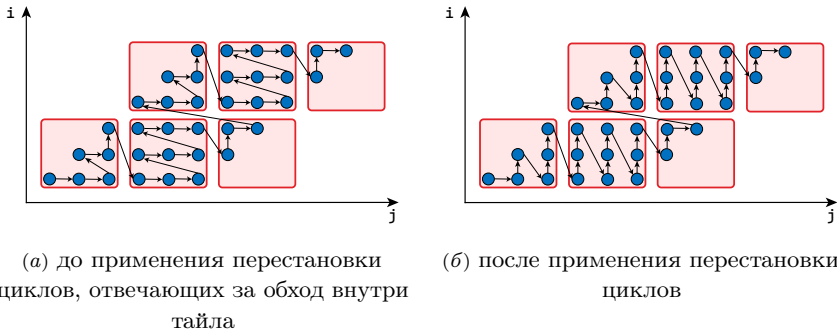


Рисунок 6. Порядок выполнения итераций двумерного гнезда циклов; точки – точки пространства итераций, дуги – порядок выполнения точек; красным выделены блоки (тайлы)

Данное преобразование может повышать временную локальность за счёт того, что вычисленные на предыдущих итерациях данные используются на следующих до того, как будут вытеснены из кэш-памяти и регистров. Ускорение, которое достигается описанной перестановкой, демонстрируется в параграфе 3.5 на примере алгоритма Гаусса-Зейделя для решения обобщённой задачи Дирихле уравнения Лапласа. Для данной задачи была выбрана матрица скашивания $skew = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$.

Для данной задачи определим откуда считываются данные при вычислении одного тайла без перестановки и с перестановкой. Обозначим M время чтения из оперативной памяти, C - из кэш-памяти и R из регистров.

Для алгоритма Гаусса-Зейделя двумерной задачи Дирихле уравнения Лапласа, на каждой итерации производится вычисление элемента массива на основе его соседних элементов

$$u_{i,j} = \frac{u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}}{4}.$$

Таким образом выполняется 4 чтения из памяти.

Модель чтения данных для первого случая, когда обход точек тайла осуществляется по горизонтальным плоскостям (сечениям), представлена в таблицах 1, 2.

Модель чтения данных для второго случая, при котором обход точек

Таблица 1. Чтение данных первого сечения при выполнении по горизонтальным сечениям

	Чтения	Количество точек
Первая выполняемая точка тайла	$4M$	1
Край сечения – нижний	$R + 3M$	$a - 1$
Край сечения – левый	$2C + 2M$	$b - 1$
Внутренние точки сечения	$2C + R + M$	$ab - (a + b - 1)$

Таблица 2. Чтение данных последующих сечений при выполнении по горизонтальным сечениям

	Чтения	Количество точек
Первая выполняемая точка тайла	$2C + 2M$	$c - 1$
Край сечения – нижний	$R + C + 2M$	$(c - 1)(a - 1)$
Край сечения – левый	$3C + M$	$(c - 1)(b - 1)$
Внутренние точки сечения	$R + 3C$	$(c - 1)(a - 1)(b - 1)$

тайла осуществляется по плоскостям (сечениям) параллельным боковой грани тайла, представлена в таблицах 3, 4.

Таблица 3. Чтение данных первого сечения при выполнении по боковым сечениям

	Чтения	Количество точек
Первая выполняемая точка тайла	$4M$	1
Край сечения – нижний	$C + 3M$	$a - 1$
Край сечения – левый	$2R + 2M$	$b - 1$
Внутренние точки сечения	$C + 2R + M$	$ab - (a + b - 1)$

Во втором случае, при обходе точек тайла по боковым сечениям, для внутренних точек будет больше чтений данных их регистров, чем в первом случае. Это увеличивает временную локальность данных и даёт ускорение.

2.5. Применение метода гиперплоскостей и прагм OpenMP для параллельного выполнения тайлов

Основная идея алгоритма, ускоряющего вычисление гнезд циклов итерационного типа, состоит в том, чтобы параллельно выполнять не отдельные точки пространства итераций, а блоки (тайлы) таких точек. На начальных этапах алгоритма производится анализ информационных

Таблица 4. Чтение данных последующих сечений при выполнении по боковым сечениям

	Чтения	Количество точек
Первая выполняемая точка тайла	$2C + 2M$	$c - 1$
Край сечения – нижний	$2C + 2M$	$(c - 1)(a - 1)$
Край сечения – левый	$2R + C + M$	$(c - 1)(b - 1)$
Внутренние точки сечения	$2R + 2C$	$(c - 1)(a - 1)(b - 1)$

зависимостей и, если требуется, выполняется скачивание. Получаем такое гнездо циклов, в котором для каждой информационной зависимости вектор расстояний не имеет отрицательных координат, что позволяет применять тайлинг.

После выполнения тайлинга, можно концептуально построить фактор-граф решетчатого графа по тайлам, т. е. вершинами такого фактор-графа являются тайлы. Дуга между вершинами фактор графа существует, если между точками тайлов, соответствующих этим вершинам, присутствует дуга информационной зависимости.

Покрываем вершины фактор-графа семейством параллельных гиперплоскостей. Удобно брать семейство гиперплоскостей с вектором нормали, имеющим все координаты единицы: $(1, 1, \dots, 1)$ (рисунок 7), поскольку

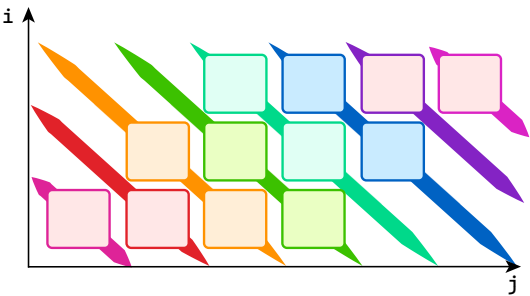


Рисунок 7. Фактор-граф по тайлам. Блоки – вершины фактор-графа. Фактор-граф покрыт семейством параллельных гиперплоскостей с вектором нормали $= (1, 1)$. Блоки, находящиеся на одной гиперплоскости, выделены одним цветом.

тогда получаются гиперплоскости более насыщенные тайлами, что выгодно для распараллеливания. Приведём обоснование того, что тайлы, лежащие в таких гиперплоскостях, попарно не связаны информационными

зависимостями.

ТЕОРЕМА 1. *После применения тайлинга тайлы, лежащие на одной гиперплоскости с вектором нормали $(1, 1, \dots, 1)$, попарно не связаны информационными зависимостями, и, следовательно, могут выполняться параллельно.*

ДОКАЗАТЕЛЬСТВО. Рассмотрим двумерный случай. Предположим, что существует пара, лежащих в одной гиперплоскости, информационно зависимых вершин фактор-графа решетчатого графа по тайлам. Тогда в соответствующих тайлах есть точки $u = (a, b), v = (c, d)$ пространства итераций, между которыми есть дуга решетчатого графа (u, v) , соответствующая некоторой дуге графа информационных зависимостей. Эта дуга решетчатого графа имеет вектор расстояния $\text{dist}(u, v) = (c - a, d - b)$. Покажем, что этот вектор будет иметь отрицательную координату.

Пусть d_1, d_2 - размеры тайлов, $(x_1, y_1), (x_2, y_2)$ - координаты этих тайлов на фактор-графе к которым принадлежат вхождения u, v соответственно, тогда выполняются неравенства:

$$\begin{aligned} x_1 d_1 &\leq a < (x_1 + 1) d_1 & y_1 d_2 &\leq b < (y_1 + 1) d_2 \\ x_2 d_1 &\leq c < (x_2 + 1) d_1 & y_2 d_2 &\leq d < (y_2 + 1) d_2 \end{aligned}$$

Поскольку $(x_1, y_1), (x_2, y_2)$ координаты тайлов лежащих на одной гиперплоскости с вектором нормали $(1, 1)$, $(x_2 - x_1) + (y_2 - y_1) = 0$. Обозначим $n = x_2 - x_1$, тогда $y_2 - y_1 = -n$,

$$\begin{aligned} x_1 d_1 &\leq a < (x_1 + 1) d_1 & y_1 d_2 &\leq b < (y_1 + 1) d_2 \\ (x_1 + n) d_1 &\leq c < ((x_1 + n) + 1) d_1 & (y_1 - n) d_2 &\leq d < ((y_1 - n) + 1) d_2 \end{aligned}$$

Из этого при $n \geq 1$ следует, что $a < c$ и $b > d$, т.е. $c - a > 0$ и $d - b < 0$. При $n < 1$ выполняется $a > c$ и $b < d$ т.е. $c - a < 0$ и $d - b > 0$. Получается, что одна из координат вектора расстояний будет отрицательной.

□

СЛЕДСТВИЕ 1. *Применение метода гиперплоскостей с вектором нормали $(1, 1, \dots, 1)$ и последующее распараллеливание является эквивалентными.*

Для того чтобы покрыть вершины фактор-графа семейством параллельных гиперплоскостей, с выбранным вектором нормали $(1, 1, \dots, 1)$, выполняется следующая последовательность действий.

- (0) Для гнезда циклов (размерности $n + 1$), после применения тайлинга (размерность гнезда $2(n + 1)$), выбираются циклы, отвечающие за обход тайлов (первые $n + 1$ циклов преобразованного гнезда).
- (1) Из выбранных циклов определяется внутренний, с порядковым номером n (нумерация циклов от 0). Переменной K присвоим номер n .
- (2) Преобразование Loop Wavefront, с входным параметром равным 1 (параметр скашивания), применимо к циклам под номерами K , $K - 1$. Порядковый номер выбранного цикла K , после применения преобразования, будет равняться $K - 1$.
- (3) Если порядковый номер выбранного цикла K не равняется нулю, то переходим к пункту (2).
- (4) Циклы с номерами от 1 до n отвечают за обход тайлов лежащих на одной гиперплоскости. К ним можно применить распараллеливание, например, за счёт добавления прагм OpenMP.

После применения описанного алгоритма может быть применено преобразование линейаризация выражений, которое дополнительно повышает производительность.

3. Эквивалентность алгоритма оптимизации итерационных гнёзд циклов

ТЕОРЕМА 2. *Алгоритм оптимизации итерационных гнёзд циклов является эквивалентным.*

ДОКАЗАТЕЛЬСТВО. Алгоритм состоит из преобразований: скашивание, тайлинг (гнездование циклов, перестановка циклов), перестановка циклов, метод гиперплоскостей (скашивание, перестановка циклов). Расширение алгоритма, дополняет представленный алгоритм преобразованиями «линейаризация» и «вынос инвариантных выражений». Доказательство эквивалентности алгоритма сводится к доказательству эквивалентности каждого преобразования последовательности.

Скашивание и гнездование не меняют порядок обращения к памяти, а потому являются эквивалентными. Если все информационные зависимости в гнезде циклов имеют векторы расстояний и эти векторы не имеют отрицательные координаты, то перестановка циклов эквивалентна [2, 9]. Для проверки эквивалентности тайлинга производится анализ информационных зависимостей. При наличии зависимостей, имеющих отрицательные координаты вектора расстояний, выполняется с соответствующими

параметрами скапливание так, чтобы все векторы расстояний имели неотрицательные координаты. Эквивалентность метода гиперплоскостей обосновывается теоремой 1. «Линеаризация» и «вынос инвариантных выражений» являются эквивалентными преобразованиями. Таким образом описанный алгоритм оптимизации итерационных гнёзд циклов и его расширение являются эквивалентными. $\square$

4. Численные эксперименты

Численные эксперименты были проведены на компьютере с процессором Intel i7-9700 (Coffee Lake), тактовая частота 3,00 GHz, 8 ядер, размер кэш-памяти: L1 — 256 Kb; L2 — 2 Mb; L3 — 12 Mb. В качестве компилятора использовался GCC-6.3.0-1 с опцией -O3. Ускорение вычислялось по формуле:

$$\text{Ускорение} = \frac{\text{Время выполнения исходной программы}}{\text{Время выполнения преобразованной программы}}.$$

4.1. Про выбор оптимальных размеров тайлов

Рассмотрим выбор оптимальных размеров тайлов для алгоритма Гаусса-Зейделя решения двумерной задачи Дирихле уравнения Лапласа (листинг 5). Размерность задачи $256 \times 4000 \times 4000$, тип данных double. Компилятор gcc, опция компилятора -O3. Применялся скошенный тайлинг с матрицей скапливания $\text{skew} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$ и перестановка циклов внутри тайла.

Листинг 5. Основное гнездо циклов алгоритма Гаусса-Зейделя для решения задачи Дирихле уравнения Лапласа

```
for (t=0; t<=T-1; t++)
    for (i=1; i<=N-2; i++)
        for (j=1; j<=N-2; j++)
            u[i][j] = (u[i-1][j]+u[i+1][j]+
                       u[i][j-1]+u[i][j+1])/4.0;
```

При тайлинге ускорение достигается за счет того, что некоторые данные при вычислении точек тайла читаются много раз, причем все чтения, кроме первого происходят из кэш-памяти или регистров. Например, в коде листинга (листинг 5) к ячейке памяти, в которой хранится элемент массива $u_{3,7}$, программа обращается при значениях вектора счетчиков циклов (t, i, j) при вычислении не только элемента массива $u_{3,7}$, но и элементов $u_{2,7}$ $u_{4,7}$ $u_{3,6}$ $u_{3,8}$. Идея оптимизации вычисления гнёзд циклов

итерационного типа состоит в том, чтобы, не завершив вычисления одной итерации t , начинать вычисление следующей итерации $(t + 1)$, пока в кэш-памяти находятся необходимые для вычислений данные.

Пусть d_1, d_2, d_3 – размеры тайлов двумерной задачи (трехмерное гнездо циклов). Пронаблюдаем влияние разных размеров тайлов на ускорение. Зафиксируем $d_2 = 50$, $d_1 = (32 \text{ либо } 64)$. Оценим влияние размера d_3 на ускорение (таблица 5).

Таблица 5. Влияние величины d_3 на ускорение программы из подраздела 4.1 при последовательном выполнении

d_3	$d_1 = 32$		$d_1 = 64$	
	Время выполнения (сек)	Ускорение	Время выполнения (сек)	Ускорение
4	3,823	2,915	3,643	3,059
8	3,359	3,318	3,257	3,421
16	3,142	3,547	3,028	3,680
20	3,088	3,609	2,990	3,727
25	3,064	3,637	2,951	3,777
40	3,007	3,706	2,880	3,869
50	2,984	3,735	2,869	3,884
80	2,959	3,766	2,815	3,959
100	2,932	3,801	2,813	3,961
125	2,920	3,816	2,825	3,944
200	2,899	3,844	2,755	4,045
250	2,895	3,849	2,754	4,046
400	2,907	3,833	2,754	4,046
500	2,887	3,860	2,731	4,080
1000	2,877	3,873	2,723	4,092
2000	2,876	3,875	2,709	4,114

При последовательном выполнении преобразованной программы наблюдается постепенное увеличение ускорения в пределах 1.1 секунды и при увеличении значения прирост уменьшается. Это связано с затратами на вычисление первого сечения тайла. Данные для первого сечения подгружаются из оперативной памяти, а для последующих – большая часть из кэш-памяти и регистров (таблицы 3, 4), чтобы компенсировать затраты на инициализацию значение должно быть как можно больше.

Таблица 6 иллюстрирует влияние размера d_3 при параллельном выполнении на 16 потоках.

Таблица 6. Влияние величины d_3 на ускорение программы из подраздела 4.1 при распараллеливании на 16 потоков

d_3	$d_1 = 32$		$d_1=64$	
	Время выполнения (сек)	Ускорение	Время выполнения (сек)	Ускорение
4	1,400	7,957	1,223	9,111
8	1,013	11,004	0,947	11,769
16	0,849	13,125	0,799	13,946
20	0,817	13,645	0,758	14,696
25	0,776	14,352	0,722	15,425
40	0,650	17,132	0,605	18,406
50	0,654	17,033	0,617	18,071
80	0,692	16,093	0,664	16,786
100	0,702	15,882	0,663	16,812
125	0,654	17,038	0,639	17,449
200	0,691	16,130	0,672	16,577
250	0,615	18,124	0,605	18,430
400	0,807	13,804	0,751	14,841
500	0,573	19,460	0,558	19,969
1000	0,970	11,483	0,891	12,512
2000	1,727	6,453	1,585	7,028

При распараллеливании наблюдается влияние значения d_3 на ускорение преобразованной программы. Наблюдается корреляция изменения ускорения для $d_1 = 32$ и $d_1 = 64$.

Предлагается теоретически оптимальное значение d_3 равным d_2 , которое подтверждается численными экспериментами, приведёнными в таблице 7.

Определим гипотезу для вычисления размеров d_1 и d_2 .

Для алгоритма Гаусса-Зейделя двумерной задачи Дирихле уравнения Лапласа тайл представляет собой целые точки параллелепипеда в пространстве Z^3 . Для вычислений, соответствующих внутренним точкам такого параллелепипеда все используемые данные уже использовались при вычислении этого же тайла. Эти ранее использованные данные находятся в кэш-памяти и некоторые даже в регистрах, если между повторными использованиями одного данного объем использованных других данных меньше объема кэш-памяти (не происходит вытеснения, данного из кэш-памяти).

Таким образом, объем данных между двумя использованиями одного и

ТАБЛИЦА 7. Результаты оптимизации программы из подраздела 4.1

Число потоков	Последовательно		16 потоков	
Размеры блоков	Время выполнения	Ускорение	Время выполнения	Ускорение
$d_1 = 32, d_2 = 25, d_3 = 25$	3,605	3,11	0,752	14,90
$d_1 = 32, d_2 = 40, d_3 = 40$	3,075	3,64	0,649	17,26
$d_1 = 32, d_2 = 50, d_3 = 50$	2,982	3,76	0,651	17,21
$d_1 = 32, d_2 = 80, d_3 = 80$	2,954	3,79	0,677	16,55
$d_1 = 32, d_2 = 100, d_3 = 100$	3,088	3,63	0,687	16,31
$d_1 = 64, d_2 = 20, d_3 = 20$	3,373	3,32	0,707	15,84
$d_1 = 64, d_2 = 25, d_3 = 25$	3,200	3,50	0,688	16,28
$d_1 = 64, d_2 = 40, d_3 = 40$	2,852	3,93	0,600	18,67
$d_1 = 64, d_2 = 50, d_3 = 50$	2,884	3,88	0,603	18,57
$d_1 = 64, d_2 = 80, d_3 = 80$	2,914	3,84	0,652	17,19
$d_1 = 64, d_2 = 100, d_3 = 100$	3,038	3,69	0,662	16,92

того же данного должен быть не более объема кэш-памяти. Объем памяти между двумя использованиями одного и того же данного обозначим V . Если мы хотим минимизировать количество промахов к L1-кэш, то должно выполняться неравенство

$$(*) \quad V \leq |L1|$$

После перестановки циклов внутри тайла циклы гнезда обхода тайла расположены так, чтобы выполнение тайла проводилось по сечениям, параллельным боковой грани параллелепипеда с длинами ребер d_1 и d_2 .

В этом случае количество точек трех соседних сечений (данные которых участвуют при вычислении среднего сечения в алгоритме) равно $d_1 d_2 + 2(d_1 + d_2)$ и, в соответствии с (*), должно выполняться условие: $V \leq (d_1 d_2 + 2(d_1 + d_2)) = ((d_1 + 2)(d_2 + 2) - 4) \leq |L1|$

Границей сечения является прямоугольник со сторонами d_1 и d_2 . При фиксированной площади прямоугольника длина (количество точек) границы будет минимальна. Если стороны этого прямоугольника равны: $d_1 = d_2$. $d_1 = d_2 \leq \sqrt{|L1| + 4} - 2$. Здесь V — объединение целых точек трех сечений параллелепипеда (тайла).

Для процессора, на котором проводились вычисления, объем L1 = 32КБ на ядро процессора, тогда количество чисел двойной точности (double, 8 байт), которые могут находиться в L1 равно 4096. Следовательно теоретические оптимальные размеры d_1 и d_2 : $d_1 = d_2 \leq 62$ числа типа double.

Численные эксперименты показали при наиболее близких значениях размеров тайла $d_1 = 64$, $d_2 = 50$, $d_3 = 50$ ускорение 18,57, которое незначительно отличается от лучшего ускорения 18,67 при $d_1 = 64$, $d_2 = 40$, $d_3 = 40$ (таблица 7).

Таким образом подтверждается гипотеза о размерах тайлов d_1 , d_2 .

4.2. Сравнение с оптимизирующей распараллеливающей системой PLUTO

Проведено сравнение времени выполнения программ, полученных путём преобразования алгоритма Гаусса-Зейделя для решения обобщённой задачи Дирихле с использованием алгоритма, описанного в статье и реализованного в OPC, и системой PLUTO.

Листинг 6. Основное гнездо циклов алгоритма Гаусса-Зейделя для решения обобщённой задачи Дирихле уравнения Лапласа

```
for (t=0; t<=T-1; t++)
  for (i=1; i<=N-2; i++)
    for (j=1; j<=N-2; j++)
      u[i][j] = (A[i][j]*u[i-1][j]+
                 B[i][j]*u[i+1][j]+
                 C[i][j]*u[i][j-1]+
                 D[i][j]*u[i][j+1]+
                 y0[i][j])/5.0;
```

Рассмотрен алгоритм Гаусса-Зейделя для численного решения обобщённой задачи Дирихле (листинг 6). Размерность задачи: 256 – количество шагов алгоритма, 2000×2000 – размер сетки. Тип данных float. Время выполнения исходного последовательного алгоритма: 7,0792 сек.

Наибольшее ускорение (в 16.32 раза) гнезда циклов, описанного в листинге, достигается алгоритмом, реализованным в OPC, с размерами блоков $64 \times 50 \times 50$. В то время как программа преобразованная, используя PLUTO, показывает своё лучшее ускорение (в 11.34 раза) на блоках меньшей размерности $d_1 \times d_2 \times d_3 = 8 \times 8 \times 8$ (рисунок 8). Эта разница обусловлена тем, что разработанный алгоритм применяет перестановку циклов внутри тайлов, что повышает временную локальность данных внутри блока (тайла).



Рисунок 8. Сравнение ускорений при выполнении на 16 потоках, полученных при помощи PLUTO и разработанного алгоритма оптимизации (в OPC) для алгоритма Гаусса-Зейделя.

4.3. Влияние преобразований «линеаризация выражений» и «вынос инвариантных выражений» на ускорение программ

Рассмотрим влияние дополнительных преобразований «линеаризация выражений» (реализованное в OPC обобщение вычислений на этапе компиляции [10], [16]) и известное в оптимизирующей компиляции преобразование «вынос инвариантных выражений» на ускорение программ, преобразованных алгоритмом, описанным в данной статье, на примере гнезда циклов (листинг 7).

В работе [10] приведён код (более 50 строк) преобразованного необобщённого алгоритма Гаусса-Зейделя для решения задачи Дирихле уравнения Лапласа. Так же в работе [10] приведён пример применения к рассматриваемому алгоритму преобразования линеаризация выражений для упрощения выражений.

Листинг 7. Трёхмерное гнездо циклов итерационного типа, вычисления производятся по восьми соседним точкам

```

for (int k = 0; k < K; ++k )
  for (int i = 1; i < N - 1; ++i )
    for (int j = 1; j < M - 1; ++j )
      u[i][j] = (u[i-1][j]+u[i+1][j]+
                 u[i][j-1]+u[i][j+1]+
                 u[i-1][j-1]+u[i+1][j-1]+
                 u[i+1][j-1]+u[i+1][j+1])/8.0;

```

Таким образом, за счёт применения дополнительных преобразований, после описанного алгоритма оптимизации из раздела 2, для гнёзда циклов из листинга 7, достигается дополнительное ускорение в до 1.8 раза (таблица 8).

Таблица 8. Сравнение влияния преобразований «линеаризация» и «вынос инвариантных выражений» на программу из листинга 7, предварительно преобразованную алгоритмом из раздела 2. Выполнение на 16 потоках

Размеры блоков	Время без дополнительных преобразований	Время с дополнительными преобразованиями	Ускорение
d1=32, d2=16, d3=16	3,03	1,99	1,52
d1=32, d2=20, d3=20	3,24	1,94	1,67
d1=32, d2=25, d3=25	3,39	1,92	1,77
d1=64, d2=16, d3=16	2,93	1,89	1,55
d1=64, d2=20, d3=20	3,18	1,85	1,72
d1=64, d2=25, d3=25	3,33	1,85	1,80

Заключение

В работе приводится алгоритм ускорения гнёзд циклов, основанный на комбинации оптимизирующих преобразований, и его расширение при помощи оптимизации линейных участков программы. Обосновывается эквивалентность алгоритма.

Метод изменения обхода точек тайла, который достигается перестановкой циклов внутри тайла, повышает временную локальность данных. Теоретическая модель вычисления оптимальных размеров тайлов подтверждается численными экспериментами.

Результаты численных экспериментов показывают ускорение в 16.32 раза алгоритма Гаусса-Зейделя, преобразованного при помощи приведённого алгоритма относительно исходной программы, что в 1.4 раза быстрее в сравнение с программой преобразованной известным алгоритмом оптимизации реализованном в системе PLUTO. Ускорение достигается за счёт перестановки циклов, отвечающих за обход тайлов, тем самым повышается временная локальность данных.

Численные эксперименты демонстрируют ускорение в 1.8 раза за счет применения дополнительных оптимизаций линейных участков программы, таких как линеаризация и вынос инвариантных выражений (после преобразования описанным алгоритмом оптимизации). Вынос инвариантных выражений из циклов уменьшает количество повторяющихся вычислений. Линеаризация упрощает индексные выражений и конструкции, отвечающие за вычисление границ циклов.

Список литературы

- [1] Wolf M. E., Lam M. S. *A loop transformation theory and an algorithm to maximize parallelism* // IEEE Transactions on Parallel and Distributed Systems.– 1991.– Vol. **2**.– No. 4.– Pp. 452–471.  ↑^{64, 71, 72, 73}
- [2] Wolf M., Banerjee U. *Data dependence and its application to parallel processing* // International Journal of Parallel Programming.– 1987.– Vol. **16**.– No. 2.– Pp. 137–178.  ↑^{64, 72, 77, 82}
- [3] Bondhugula U., Baskaran M., Krishnamoorthy S., Ramanujam J., Rountev A., Sadayappan P. *Automatic transformations for communication-minimized parallelization and locality optimization in the polyhedral model*, CC 2008: Compiler Construction, Lecture Notes in Computer Science.– vol. **4959**, Berlin–Heidelberg: Springer.– 2008.– ISBN 978-3-540-78791-4.– Pp. 132–146.  ↑⁶⁴
- [4] Lamport L. *The parallel execution of DO loops* // Commun. ACM.– 1974.– Vol. **17**.– No. 2.– Pp. 83–93.  ↑^{64, 71}
- [5] Mullapudi R. T., Vasista V., Bondhugula U. *PolyMage: automatic optimization for image processing pipelines* // ACM SIGPLAN Notices.– 2015.– Vol. **50**.– No. 4.– Pp. 429–443.  ↑⁶⁴
- [6] Maydan D. E., Hennessy J. L., Lam M. S. *Efficient and exact data dependence analysis* // ACM SIGPLAN Notices.– 1991.– Vol. **26**.– No. 6.– Pp. 1–14.  ↑⁶⁹
- [7] Wolfe M. *Loop skewing: the wavefront method revisited* // Int. J. Parallel. Program..– 1986.– Vol. **15**.– No. 4.– Pp. 279–293.  ↑⁶⁹
- [8] Irigoin F., Triolet R. *Supernode partitioning* // POPL '88: Proceedings of the 15th ACM SIGPLAN-SIGACT symposium on Principles of programming languages (San Diego, California, USA, January 10–13, 1988), New York: ACM.– 1988.– ISBN 978-0-89791-252-5.– Pp. 319–329.  ↑⁷²
- [9] Allen R., Kennedy K. *Automatic translation of FORTRAN programs to vector form* // ACM Transactions on Programming Languages and Systems.– 1987.– Vol. **9**.– No. 4.– Pp. 491–542.  ↑⁸²
- [10] Vasilenko A., Veselovskiy V., Metelitsa E., Zhivikh N., Steinberg B., Steinberg O. *Precompiler for the ACELAN-COMPOS package solvers*, PaCT 2021: Parallel Computing Technologies, Lecture Notes in Computer Science.– vol. **12942**, Cham: Springer.– 2021.– ISBN 978-3-030-86359-3.– Pp. 103–116.  ↑^{64, 88}
- [11] Штейнберг Б. Я. *О взаимосвязи между решетчатым графом программы и графом информационных связей* // Известия высших учебных заведений. Северо-Кавказский регион. Серия: Естественные науки.– 2011.– № 5(165).– С. 28–30.  ↑^{66, 67}

- [12] Савельев В. А., Штейнберг Б. Я. *Распараллеливание программ*, учебник. – Ростов-на-Дону: Изд-во Южного федерального университета. – 2008. – ISBN 978-5-9275-0547-0. – 192 с. ↑⁶⁵, 74
- [13] Christen M., Schenk O., Burkhart H. *PATUS: a code generation and autotuning framework for parallel iterative stencil computations on modern microarchitectures* // *2011 IEEE International Parallel & Distributed Processing Symposium* (Anchorage, AK, USA, 16–20 May 2011). – 2011. – Pp. 676–687. doi ↑⁷⁰
- [14] Bagliy A. P., Metelitsa E. A., Steinberg B. Ya. *Automatic parallelization of iterative loops nests on distributed memory computing systems*, PaCT 2023: Parallel Computing Technologies, Lecture Notes in Computer Science. – vol. **14098**, Cham: Springer. – 2023. – ISBN 978-3-031-41673-6. – Pp. 18–29. doi ↑⁶⁴
- [15] Воеводин В. В., Воеводин Вл. В. *Параллельные вычисления*. – СПб.: БХВ-Петербург. – 2002. – ISBN 5-94157-160-7. – 608 с. ↑⁶⁶
- [16] Баглий А. П., Дубров Д. В., Штейнберг Б. Я., Штейнберг Р. Б. 43–47 // *Научный сервис в сети Интернет: труды XIX Всероссийской научной конференции* (18–23 сентября 2017 г., г. Новороссийск), М.: ИПМ им. М.В.Келдыша. – 2017. – ISBN 978-5-98354-037-8. URL doi ↑⁶⁴, 88

Поступила в редакцию	21.01.2024;
одобрена после рецензирования	13.02.2024;
принята к публикации	15.02.2023;
опубликована онлайн	26.03.2024.

Рекомендовал к публикации

к.ф.-м.н. С. А. Романенко

Информация об авторе:



Елена Анатольевна Метелица

м.н.с. Института математики механики и компьютерных наук им. Воровича, Южный федеральный университет. Научные интересы в областях компиляторных преобразований, оптимизации и распараллеливания программ



0000-0001-6253-150X

e-mail: metelica@sfsedu.ru

Автор заявляет об отсутствии конфликта интересов.



Justification of methods for accelerating iterative loops nests

Elena Anatol'evna **Metelitsa**

Southern Federal University, Rostov-on-Don, Russia

 metelica@sfedu.ru

Abstract. The acceleration of iterative algorithms, found in solving problems of mathematical physics, mathematical modeling, and image processing, is considered. In the software implementation of these algorithms, there are nested loops (sections of the program that consist of nested loops). These loop nests can be accelerated by combination of optimizing transformations, including tiling, hyperplane method, and parallelization on shared memory. The equivalence of this combination of program transformations is substantiated.

A method for changing the order of tile traversal is proposed and justified. The method provides acceleration by increasing data readings from registers instead of slower memory. Considering this method, a formula for calculating optimal tile sizes is obtained.

The combination of transformations presented in this article results in an acceleration that is 1.4 times greater than the well-known optimization algorithm implemented in PLUTO. In some cases using an 8-core processor, numerical experiments show a significant increase in speed compared to the original sequential algorithms. The findings of this article can be applied to manual and automatic program optimization. (*In Russian*).

Key words and phrases: tiling, wavefront, parallelization, shared memory, iterative stencil loops

2020 *Mathematics Subject Classification:* 68W10; 68N20

Acknowledgments: The author is grateful to Dr. B.Ya. Steinberg for leadership of the work and Ar.V. Klimov for his attention and interest in the work.

For citation: Elena A. Metelitsa. *Justification of methods for accelerating iterative loops nests*. Program Systems: Theory and Applications, 2024, 15:1(60), pp. 63–94. (*In Russ.*). https://psta.psiras.ru/read/psta2024_1_63-94.pdf

References

- [1] M. E. Wolf, M. S. Lam. “A loop transformation theory and an algorithm to maximize parallelism”, *IEEE Transactions on Parallel and Distributed Systems*, **2**:4 (1991), pp. 452–471. [doi](#)
- [2] M. Wolf, U. Banerjee. “Data dependence and its application to parallel processing”, *International Journal of Parallel Programming*, **16**:2 (1987), pp. 137–178. [doi](#)
- [3] U. Bondhugula, M. Baskaran, S. Krishnamoorthy, J. Ramanujam, A. Rountev, P. Sadayappan. “Automatic transformations for communication-minimized parallelization and locality optimization in the polyhedral model”, CC 2008: Compiler Construction, Lecture Notes in Computer Science, vol. **4959**, Springer, Berlin–Heidelberg, 2008, ISBN 978-3-540-78791-4, pp. 132–146. [doi](#)
- [4] L. Lamport. “The parallel execution of DO loops”, *Commun. ACM*, **17**:2 (1974), pp. 83–93. [doi](#)
- [5] R. T. Mullanpudi, V. Vasista, U. Bondhugula. “PolyMage: automatic optimization for image processing pipelines”, *ACM SIGPLAN Notices*, **50**:4 (2015), pp. 429–443. [doi](#)
- [6] D. E. Maydan, J. L. Hennessy, M. S. Lam. “Efficient and exact data dependence analysis”, *ACM SIGPLAN Notices*, **26**:6 (1991), pp. 1–14. [doi](#)
- [7] M. Wolfe. “Loop skewing: the wavefront method revisited”, *Int. J. Parallel. Program.*, **15**:4 (1986), pp. 279–293. [doi](#)
- [8] F. Irigoin, R. Triolet. “Supernode partitioning”, *POPL ’88: Proceedings of the 15th ACM SIGPLAN-SIGACT symposium on Principles of programming languages* (San Diego, California, USA, January 10–13, 1988), ACM, New York, 1988, ISBN 978-0-89791-252-5, pp. 319–329. [doi](#)
- [9] R. Allen, K. Kennedy. “Automatic translation of FORTRAN programs to vector form”, *ACM Transactions on Programming Languages and Systems*, **9**:4 (1987), pp. 491–542. [doi](#)
- [10] A. Vasilenko, V. Veselovskiy, E. Metelitsa, N. Zhiviykh, B. Steinberg, O. Steinberg. “Precompiler for the ACELAN-COMPOS package solvers”, PaCT 2021: Parallel Computing Technologies, Lecture Notes in Computer Science, vol. **12942**, Springer, Cham, 2021, ISBN 978-3-030-86359-3, pp. 103–116. [doi](#)
- [11] B. Ya. Shtejnberg. “About the connection between the lattice graph and the data dependence graph”, *Izvestiya vysshix uchebnyx zavedenij. Severo-Kavkazskij region. Seriya: Estestvennye nauki*, 2011, no. 5(165), pp. 28–30 (in Russian).
- [12] V. A. Savel’ev, B. Ya. Shtejnberg. *Parallelization of Programs*, uchebnik, Izd-vo Yuzhnogo federal’nogo universiteta, Rostov-na-Donu, 2008, ISBN 978-5-9275-0547-0 (in Russian), 192 pp.
- [13] M. Christen, O. Schenk, H. Burkhardt. “PATUS: a code generation and autotuning framework for parallel iterative stencil computations on modern microarchitectures”, *2011 IEEE International Parallel & Distributed Processing Symposium* (Anchorage, AK, USA, 16–20 May 2011), 2011, pp. 676–687. [doi](#)

- [14] A. P. Bagliy, E. A. Metelitsa, B. Ya. Steinberg. “Automatic parallelization of iterative loops nests on distributed memory computing systems”, PaCT 2023: Parallel Computing Technologies, Lecture Notes in Computer Science, vol. **14098**, Springer, Cham, 2023, ISBN 978-3-031-41673-6, pp. 18–29. 
- [15] V. V. Voevodin, Vl. V. Voevodin. *Parallel Computing*, BXV-Peterburg, SPb., 2002, ISBN 5-94157-160-7 (in Russian), 608 pp.
- [16] A. P. Baglij, D. V. Dubrov, B. Ya. Shtejnberg, R. B. Shtejnberg. “43–47”, *Nauchnyj servis v seti Internet: trudy XIX Vserossijskoj nauchnoj konferencii* (18–23 sentyabrya 2017 g., g. Novorossijsk), IPM im. M.V.Keldysha, M., 2017, ISBN 978-5-98354-037-8 (in Russian).  