

Программные системы: теория и приложения

Двуязычный электронный научный журнал

№3  2025

Bilingual Online Scientific Journal

Program Systems: Theory and Applications

Том 16 Выпуск 3(66) 2025 г.

СОДЕРЖАНИЕ

Научная статья Программное и аппаратное обеспечение распределенных и суперкомпьютерных систем

Антонов И. А.[✉], Курочкин И. И.. *Методы квантового машинного обучения для обнаружения атак на программно-конфигурируемые сети* 3–19, 20–22

Научная статья Методы оптимизации и теория управления

Расина И. В.[✉], Гусева И. С.. *Метод сильного улучшения управления для неоднородных дискретных систем* 23–38, 39–40

Научная статья Программное и аппаратное обеспечение распределенных и суперкомпьютерных систем

Подлазов В. С.[✉]. *Быстродействие и отказоустойчивость идеальной системной сети через дополнительную параллельность* .. 41–65, 66–68

Научная статья Искусственный интеллект и машинное обучение

Смирнов А. В.[✉], Тищенко И. П.. *Определение здоровых и больных областей листьев растений при помощи нейронных сетей* 69–94, 95–97

Авторский указатель 101

Click the flag at a top corner of any page to switch the language, please!

Author index 102

Contents 103

УДК 004.272.45

 10.25209/2079-3316-2025-16-3-3-22

Методы квантового машинного обучения для обнаружения атак на программно-конфигурируемые сети

Илья Андреевич Антонов^{1✉}, Илья Ильич Курочкин²

¹ Университет науки и технологий МИСИС, Москва, Россия

² Институт проблем передачи информации им. А. А. Харкевича РАН, Москва, Россия

[✉] m1908142@edu.misis.ru

Аннотация. Программно-конфигурируемая сетевая архитектура является предпочтительным способом построения больших компьютерных сетей, требующих высокой скорости реагирования на изменения и высокой степени автоматизации. Основной особенностью данной архитектуры является централизованное управление всей сетью с одного контроллера. Тем не менее, такой подход открывает новые возможности для атак на сеть, делая контроллер их главной целью. В данной работе рассматривается возможность применения моделей квантового машинного обучения для обнаружения таких атак.

Ключевые слова и фразы: программно-конфигурируемые сети, информационная безопасность, машинное обучение, нейронные сети, квантовые вычисления, системы обнаружения вторжений, SDN, IDS

Для цитирования: Антонов И. А., Курочкин И. И. *Методы квантового машинного обучения для обнаружения атак на программно-конфигурируемые сети* // Программные системы: теория и приложения. 2025. Т. 16. № 3(66). С. 3–22. https://psta.psiras.ru/read/psta2025_3_3-22.pdf

Введение

В последние годы распространение сетевых атак и нарушений безопасности стало растущей проблемой как для организаций, так и для отдельных лиц [1]. Для защиты сетей от них разработаны системы обнаружения вторжений (СОВ, IDS). Однако традиционные системы обнаружения вторжений часто не могут идти в ногу с растущей сложностью и изощренностью современных атак. Следовательно, существует острая необходимость в инновационных подходах, которые могут повысить точность и эффективность обнаружения вторжений. Машинное обучение (ML) является одним из таких подходов, который показал многообещающие результаты в повышении точности и эффективности IDS. Системы обнаружения вторжений на основе ML оказались важным масштабируемым инструментом для защиты сетей от кибератак, они способны к самообучению, а также могут работать при относительно небольших мощностях с достаточной скоростью, в отличие от классических IDS [2].

Квантовые вычисления, являясь относительно молодой областью, предлагают значительное ускорение для многих задач за счет квантового параллелизма и других принципов квантовой физики, поэтому имеет смысл приложить их и к области ML. В данной работе рассматривается задача обнаружения вторжений в программно-конфигурируемых сетях с помощью методов квантового машинного обучения. В частности, рассматривается применение метода VQC, который пользуется большой популярностью у исследователей [3].

Сети традиционной архитектуры широко и успешно используются, но для больших сетей (например, дата-центров) необходима более высокая скорость реагирования на изменения и более высокая степень автоматизации, которые с помощью традиционной архитектуры не могут быть реализованы, даже при наличии высококлассных специалистов. Программно-конфигурируемая сеть (ПКС, SDN) – это концепция сети, которая отделяет плоскость управления от плоскости передачи данных [4]. Основной особенностью данной архитектуры является централизованное управление всей сетью с одного контроллера, что повышает гибкость сети и ее способность расти. Контроллер SDN может выполнять функции, связанные с управлением сетью, такие как конфигурация коммутаторов, упорядоченная доставка пакетов, получение статистических данных коммутатора и другие [5]. Архитектуру SDN можно представить схематично, как показано на рисунке 1 [6].

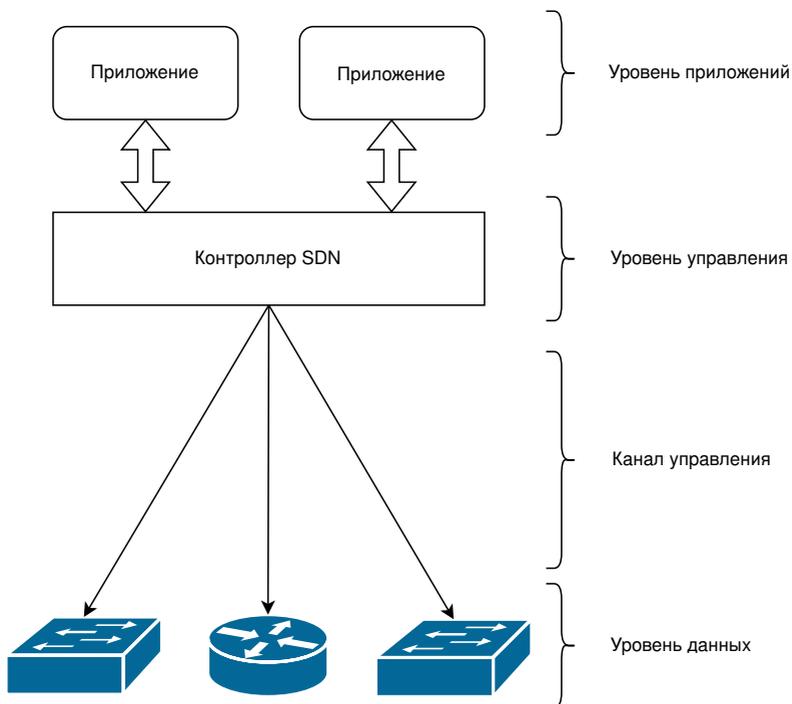


Рисунок 1. Архитектура SDN

Разделение уровня управления сетью и уровня передачи данных в сети SDN приводит к следующим преимуществам [7]:

- Отделение плоскости управления от плоскости данных облегчает управление сетью. Кроме того, сеть становится легче изменять и обновлять, что снижает количество человеческих ошибок.
- ИТ-администраторы могут легко добавлять сетевые устройства или модернизировать сетевую инфраструктуру, не привязываясь к конкретному поставщику оборудования.
- Низкоуровневые инфраструктурные устройства не требуют отдельного программирования, что значительно снижает эксплуатационные расходы по сравнению с обычной сетью.

Обнаружение вторжений – это задача классификации, распределяющая входящий сетевой трафик на классы: обычный и вредоносный [8]. Сами атаки на программно-конфигурируемые сети можно классифицировать на 4

категории [9]: удаленные атаки на плоскость управления сетью, локальные атаки на плоскость управления сетью, атаки на канал управления сетью (особенно, протокол OpenFlow, используемый в SDN) и атаки на уровень передачи данных.

1. Особенности квантовых вычислений

Квантовые вычисления — это относительно новая область, которая объединяет компьютерные науки, математику и физику. Эта область исследует возможности использования принципов квантовой физики для создания квантовых компьютеров, использующих квантовые биты (кубиты), которые могут содержать комбинации значений 0 и 1 в суперпозиции одновременно.

Так как кубиты могут находиться одновременно в нескольких состояниях, временные затраты на расчет каждого состояния не требуются, а общее время расчётов уменьшается. Именно такая возможность позволяет решать некоторые сложные задачи быстрее классических компьютеров [10, 11].

Классический компьютер с памятью из n бит может одномоментно выполнить некоторую операцию только над одним из $2n$ возможных наборов. Чтобы вычислить значение некоторой булевой функции от n аргументов для всех значений аргументов с помощью классического компьютера, придётся по очереди перебирать все $2n$ наборов и вычислять значение функции на каждом наборе по отдельности. Благодаря квантовой суперпозиции, в квантовом компьютере с n кубитами можно одновременно представить все $2n$ набора и выполнять операции над всеми наборами сразу. В результате возможно вычислить значение функции сразу для всех $2n$ комбинаций значений аргументов [12].

Кубит можно определить, как вектор единичной длины в двумерном гильбертовом пространстве над полем комплексных чисел. Состояния 0 и 1 вместе представляют собой базисные вектора. Каждое из состояний кубита может быть представлено суперпозицией двух базисных состояний, которые обычно обозначаются как $|0\rangle$ и $|1\rangle$. Формула, представляющая кубит в суперпозиции, выглядит следующим образом:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

α и β - комплексные числа, известные как амплитуды. Сумма их квадратов должна быть равна 1, поэтому можно сказать, что они определяют вероятности обнаружения кубита в каждом из базисных состояний.

Однако создание и манипулирование кубитами само по себе не является достаточным для выполнения сложных операций. Для реализации квантовых алгоритмов необходимо уметь манипулировать состояниями кубитов с высокой степенью контроля. Это достигается с помощью квантовых вентилях [10]. Квантовые вентили представляют собой аналоги классических логических вентилях, которые выполняют различные операции над кубитами, такие как логические, унитарные и нелинейные. Квантовые вентили используются для создания квантовых цепей и выполнения последовательности операций, составляющих квантовый алгоритм. Благодаря квантовым вентилям получается достичь эффективности и гибкости квантовых вычислений.

Квантовые вентили можно представить в виде унитарных матриц, которые преобразуют входной кубит в выходной кубит. Для любого входного состояния кубита, квантовый вентиль выполняет преобразования в соответствии с определенными правилами. Например, вентиль NOT инвертирует состояние кубита. Тем не менее, квантовые вычисления на данный момент используются только в исследовательских целях и не распространены для массового использования. Рассмотрим основные препятствующие этому проблемы:

- (1) Декогеренция представляет собой явление, при котором квантовая система взаимодействует с окружающей средой, что приводит к потере квантовых свойств и, следовательно, к ухудшению точности и стабильности вычислений [13]. Максимальное время жизни квантовой системы, когда она пригодна для квантовых вычислений, крайне мало, а по окончании этого времени система начнет выдавать белый шум вместо вероятностных распределений.
- (2) Одним из наиболее актуальных вызовов является масштабируемость квантовых систем. В настоящее время существуют квантовые компьютеры с небольшим числом кубитов, однако для решения реальных задач требуется значительное увеличение их мощности [11]. Разработчикам необходимо искать пути увеличения числа кубитов и улучшения их производительности, чтобы обеспечить масштабируемость и применение квантовых систем для решения широкого спектра задач.
- (3) Другим важным вызовом является сложность реализации квантовых систем. В отличие от классических вычислительных систем, которые имеют долгую историю и разработанную методологию, квантовые системы находятся на стадии активных исследований и разработок. Например, нет четких методик устранения “багов”, специфичных для

квантовых компьютеров, из-за чего во время их работы возникают ошибки [14].

Рассмотрим также основные приложения квантовых вычислений:

- (1) Алгоритм Шора [15, 16] представляет собой квантовый алгоритм, способный эффективно факторизировать большие целые числа. Факторизация является процессом разложения числа на простые множители. На классических компьютерах, работающих на классических алгоритмах, факторизация достаточно больших чисел требует экспоненциального времени, что делает ее практически невозможной для чисел с достаточной длиной для применения в криптографических системах. Алгоритм Шора, используя возможности квантовых компьютеров, способен произвести факторизацию числа не просто за полиномиальное время, а за время, ненамного превосходящее время умножения целых чисел. Таким образом, с помощью этого алгоритма (при использовании квантового компьютера с несколькими тысячами логических кубитов) становится возможным взлом криптографических систем с открытым ключом.
- (2) Алгоритм Гровера [17] позволяет найти нужный элемент базы данных за полиномиальное время, что делает его значительно более эффективным по сравнению с классическими алгоритмами. Это имеет потенциальное применение в таких областях, как поиск в больших базах данных, оптимизация и распознавание образов.
- (3) Квантовые компьютеры могут обрабатывать большие матрицы, а также ускорять различные операции линейной алгебры, значительно улучшая традиционные приложения машинного обучения [3]. В частности, было показано, что при использовании методов квантового машинного обучения можно достичь улучшения производительности (увеличение значений метрик или скорости работы) по сравнению с классическими методами, в том числе и для решения задачи обнаружения вторжений в сетях SDN [18, 19].

2. Методы квантового машинного обучения

Рассмотрим основные методы квантового машинного обучения в сравнении с их классическими аналогами.

k -ближайших соседей (k NN) – это алгоритм классификации, состоящий из трех шагов: вычисление расстояния относительно обучающих элементов; нахождение k элементов, ближайших к тестовому экземпляру; предсказание метки класса посредством голосования большинства.

В квантовом k NN вначале все элементы переводятся в пространство квантовых векторов состояний, затем вычисляются расстояния между классифицируемым объектом и всеми объектами тренировочной выборки с помощью, например, евклидовой метрики [20], SWAP-теста [21] или других. Далее класс присваивается объекту по большинству, находящемуся в числе его ближайших соседей, как и в классическом k NN. Благодаря свойству суперпозиции кубитов все расстояния от обучающих элементов вычисляются одновременно (квантовый параллелизм), что дает большой прирост к скорости работы алгоритма [20].

Метод опорных векторов (SVM) с помощью функции-ядра переводит выборку в пространство более высокой размерности, в котором данные становятся линейно разделимыми по классам, и проводит между ними разделяющую гиперплоскость. Квантовый SVM работает схожим образом: сначала происходит кодирование данных в квантовое пространство, затем к ним применяется квантовое ядро для перевода в пространство большей размерности, после чего проводится разделяющая гиперплоскость и выполняется классификация. В качестве методов оптимизации может выступать алгоритм Гровера [22] или алгоритм ННЛ [23], которые используют квантовый параллелизм и дают существенный прирост в скорости работы [24].

В квантовых сверточных нейронных сетях (QCNN) [25] изображение сначала кодируется в квантовую схему с использованием карты признаков. Затем к закодированному изображению применяются чередующиеся сверточные и объединяющие слои, уменьшая размерность схемы до тех пор, пока не останется только один кубит. Выход этого оставшегося кубита измеряется для классификации входного изображения. Квантовый сверточный слой состоит из серии двухкубитных унитарных операторов, которые распознают и определяют связи между кубитами в схеме; квантовый pooling-слой уменьшает количество кубитов, выполняя операции над каждым кубитом до определенной точки, а затем отбрасывая определенные кубиты в определенном слое. В QCNN каждый слой содержит параметризованные схемы, что означает, что выход может быть изменен путем настройки параметров каждого слоя. Во время обучения эти параметры корректируются для минимизации функции потерь.

Метод VQC (variational quantum circuit), использованный в данной работе, наиболее популярен у исследователей. Концептуально его показывает в виде схемы рисунок 2, навеянный рисунком 13 из [3]. Метод VQC представляет собой квантовый аналог нейронной сети. На входе квантовой

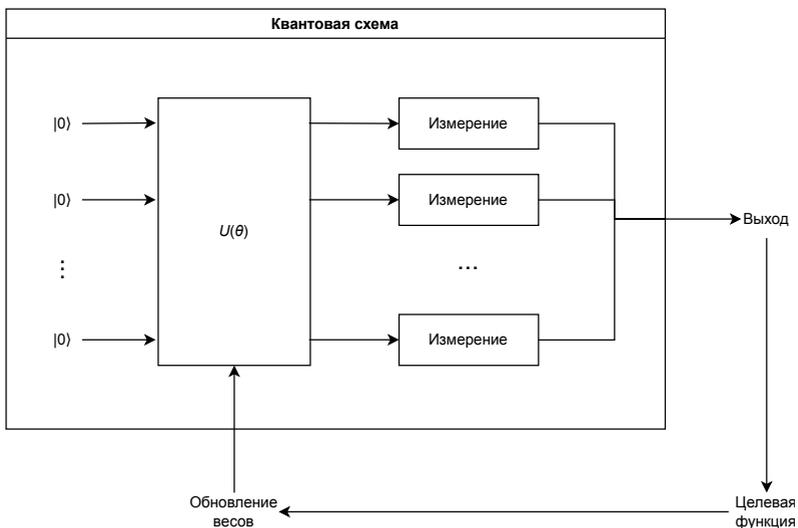


Рисунок 2. Представление метода VQC

схемы (Quantum circuit) расположено некоторое количество кубитов, соответствующее количеству признаков в наборе данных, а на выходе формируется решение задачи (например, вероятностное распределение для задачи классификации). Предварительно переведенные в пространство квантовых векторов (с помощью карты признаков) данные поступают на вход этой схемы, после чего она «обучается». Процесс обучения происходит с задействованием классического компьютера: на нем рассчитывается функция потерь и обновляются веса для квантовой схемы, после чего квантовая схема снова начинает работу.

3. Набор данных

В данной работе используется набор данных InSDN [7], опубликованный в 2020 году и содержащий различные виды трафика, актуального на сегодняшний день (YouTube, чаты, электронная почта и др.) и характерного для ПКС. Набор данных содержит типы сетевого трафика, представленные в таблице 1. Набор данных несбалансирован, поэтому для балансировки набора данных в некоторых экспериментах использовался алгоритм SMOTE (Synthetic Minority Over-sampling Technique), основная идея которого – генерация искусственных объектов в наборе данных, «похожих» на существующие, но не дублирующих их [26, 27].

Таблица 1. Виды трафика в наборе данных InSDN

Тип трафика	Количество записей
Нормальный	68424
Атака BFA	1405
Атака DDoS	121942
Атака DoS	53616
Атака Probe	98129
Атака Web-Attack	192
Атака BOTNET	164
Атака U2R	17

Учитывая, что для симуляции квантовых методов машинного обучения на классическом компьютере требуются значительные ресурсы, количество признаков в наборе данных было сокращено с 48 исходных до 10 с помощью функционала `SelectKBest` из библиотеки `Scikit-Learn`.

Оставшиеся признаки для экспериментов с задачей бинарной классификации: `Protocol`, `Fwd Pkt Len Min`, `Bwd Pkt Len Min`, `Bwd Pkt Len Mean`, `Bwd Pkt Len Std`, `Fwd IAT Tot`, `Pkt Len Min`, `Pkt Len Mean`, `Pkt Len Std`, `Pkt Size Avg`.

Оставшиеся признаки для экспериментов с задачей многоклассовой классификации: `Protocol`, `Bwd Pkt Len Min`, `Bwd Pkt Len Mean`, `Bwd Pkt Len Std`, `Flow Pkts/s`, `Bwd Pkts/s`, `Pkt Len Min`, `Pkt Len Mean`, `Pkt Len Std`, `Pkt Size Avg`.

Помимо сокращения количества признаков набор данных также был разделен на 3000 порций, и на вход методу `VQC` подавалось по одной порции за раз для минимизации использования памяти. Это тоже повлияло на значение метрики, поскольку качество модели, обучаемой на полном наборе данных, как правило, выше, чем для модели, обучаемой на том же наборе, но порционно.

4. Эксперименты и результаты

В процессе работы были проведены эксперименты с моделью квантового машинного обучения `VQC` и моделью классического машинного обучения `XGBoost` для сравнения. Эксперименты проводились на машине со следующими характеристиками: процессор `Intel Xeon E5-2695 v2`, 32 Гб оперативной памяти, ОС `Windows 10`.

Результаты работы моделей представлены в таблице 2. Модель XGBoost была взята как эталон, поскольку метод градиентного бустинга показывает лучшие результаты классификации табличных данных в большинстве случаев.

Для каждой модели было проведено 6 экспериментов:

- (1) Бинарная классификация (нормальный или атакующий трафик).
- (2) Бинарная классификация (нормальный или атакующий трафик) с предварительной нормализацией набора данных с помощью метода минимакс.
- (3) Бинарная классификация (нормальный или атакующий трафик) с предварительной нормализацией набора данных с помощью метода минимакс и балансировкой с помощью метода SMOTE.
- (4) Многоклассовая классификация (нормальный трафик или один из нескольких типов атак).
- (5) Многоклассовая классификация (нормальный трафик или один из нескольких типов атак) с предварительной нормализацией набора данных с помощью метода минимакс.
- (6) Многоклассовая классификация (нормальный трафик или один из нескольких типов атак) с предварительной нормализацией набора данных с помощью метода минимакс и балансировкой с помощью метода SMOTE.

Реализация модели VQC взята из библиотеки *Qiskit Machine Learning*^{[URL](#)} для языка Python, также в работе были использованы инструменты из библиотеки *Qiskit*^{[URL](#)}¹. Модель VQC^{[URL](#)} была взята с параметрами по умолчанию, в качестве метода оптимизации используется *COBYLA*^{[URL](#)} (Constrained Optimization By Linear Approximation optimizer), в качестве карты признаков использована *ZZFeatureMap*^{[URL](#)}, в качестве квантовой схемы использована *RealAmplitudes*^{[URL](#)}. Модель XGBoost была взята из одноименной библиотеки *XGBoost*^{[URL](#)} с параметрами по умолчанию.

Для оценки качества моделей были использованы меры ассигасу и F1. Значение ассигасу есть отношение количества правильно классифицированных объектов к общему количеству элементов выборки, а значение F1-меры сочетает в себе две другие функции оценки качества: precision (доля объектов, действительно принадлежащих данному классу

¹Прискорбно, что фирма IBM закрыла доступ к своим сайтам (прим. ред.)

относительно всех объектов, которые модель отнесла к этому классу) и recall (доля найденных классификатором объектов, принадлежащих классу, относительно всех объектов этого класса), что облегчает понимание, насколько качественно работает модель.

Для запуска экспериментов для модели VQC не был использован квантовый компьютер, все вычисления выполнялись с помощью симуляции квантовой архитектуры на классическом компьютере. Из-за этого появились ограничения по скорости работы моделей и используемой памяти: было принято решение сократить количество признаков (как уже говорилось в разделе «набор данных») и сам набор данных разделить на небольшие части и дообучать модель итеративно. Перечисленные ограничения сказались на времени обучения модели (как мы видим, оно на несколько порядков выше по сравнению с классическим методом) и значениях метрик.

Модель XGBoost обучалась также на наборе данных с сокращенным количеством признаков, но на всем наборе данных целиком, а не порциями, поскольку градиентный бустинг не поддерживает механизм дообучения и требует на вход полный набор данных за раз.

Из результатов экспериментов мы видим, что нормализация набора данных приводит к увеличению значений метрик VQC, а также балансировка набора данных методом SMOTE в эксперименте с многоклассовой классификацией, поскольку в этом случае присутствует несколько классов, представленных крайне малым количеством объектов. Нормализация не оказывает сильного влияния на классический метод XGBoost в силу особенностей его архитектуры, а балансировка набора данных не увеличивает значения метрик для обеих моделей в эксперименте с бинарной классификацией, поскольку в этом случае оба класса представлены достаточным количеством объектов.

Для экспериментов с нормализацией и балансировкой набора данных приведем графики, отражающие динамику изменения значений метрик во время обучения модели VQC по порциям (рисунки 3 и 4). Среди результатов экспериментов особое внимание хочется обратить на значения ассугасу в экспериментах с бинарной классификацией (выделены жирным шрифтом в таблице 2).

Таблица 2. Результаты экспериментов

Эксперимент	VQC			XGBoost		
	Ассурасу	F1-мера	время работы	Ассурасу	F1-мера	время работы
Бинарная классификация	0,771	0,656	721985 с	0,982	0,970	1,622 с
- она же после минимакс	0,916	0,841	722345 с	0,982	0,970	1,639 с
- она же после минимакс и SMOTE	0,915	0,841	954277 с	0,980	0,968	7,703 с
Многоклассовая классификация	0,423	0,170	835228 с	0,976	0,824	26,068 с
- она же после минимакс	0,480	0,201	836823 с	0,976	0,824	22,133 с
- она же после минимакс и SMOTE	0,624	0,210	984248 с	0,958	0,702	81,959 с

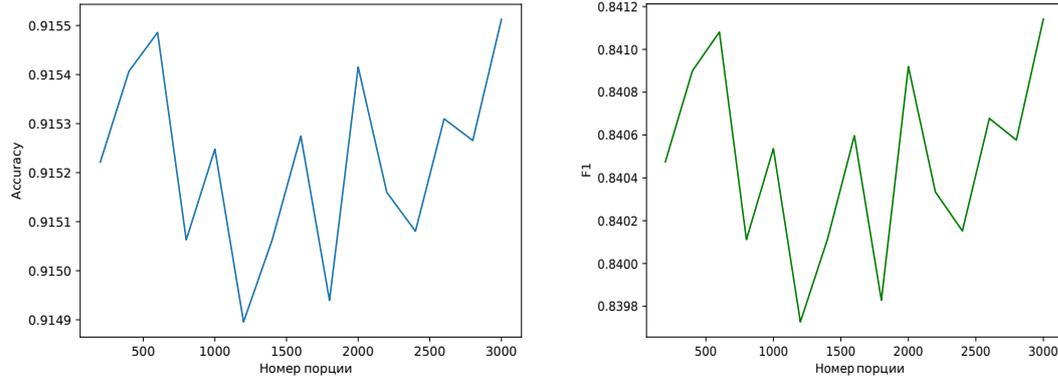


Рисунок 3. Динамика изменения метрик ассурасу и F1 во время обучения модели VQC в экспериментах с бинарной классификацией на нормализованных данных

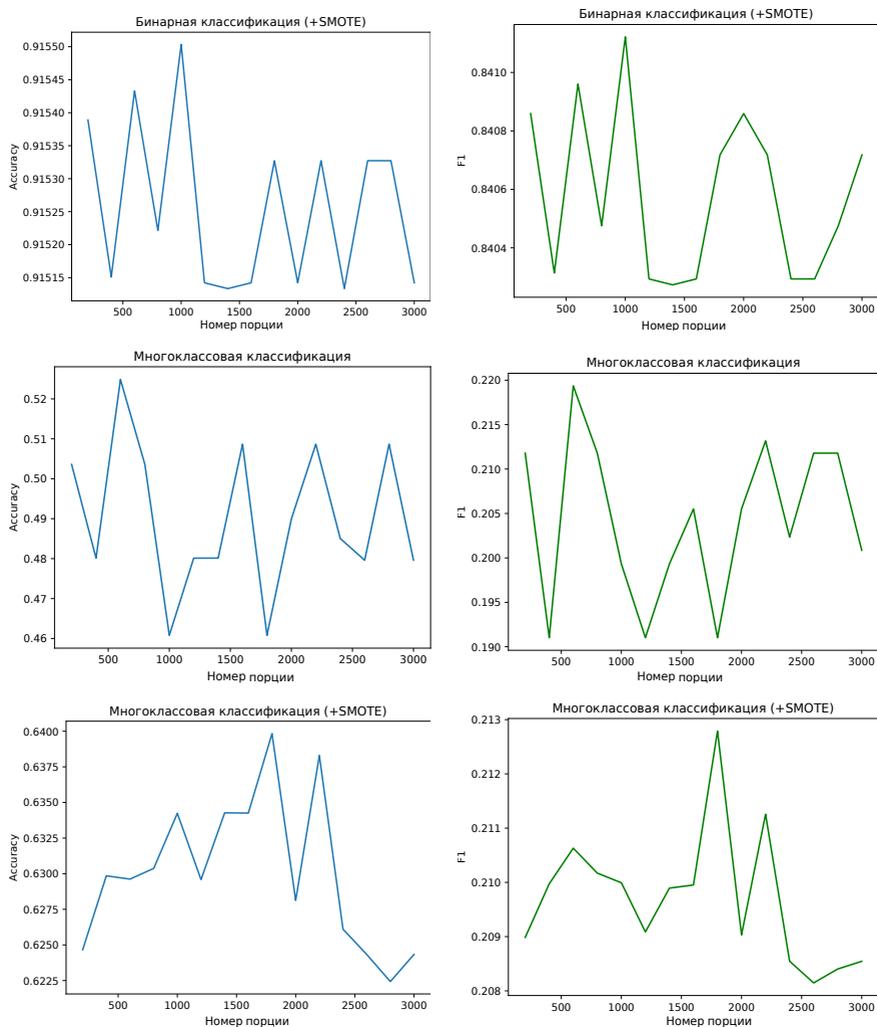


Рисунок 4. Динамика изменения метрик ассигасу и F1 во время обучения модели VQC в экспериментах на нормализованных данных

Пока результат работы квантового метода VQC хуже, чем эталонного для задачи XGBoost, но мы видим, что уже получается приблизиться к эталону, результаты работы моделей сопоставимы. При будущих доработках квантовых компьютеров вполне возможно, что квантовые

методы смогут превзойти классические, как по значениям метрик, так и по скорости работы.

Заключение

Проведенная работа показала, что квантовые методы машинного обучения уже сейчас можно ограниченно применять и исследовать. На данный момент они требуют гораздо больше ресурсов для симуляции на классическом компьютере, чем привычные нам классические методы, но при большем распространении квантовых компьютеров и увеличении числа кубитов в них QML-методы смогут составить конкуренцию традиционным ML-моделям (а возможно и превзойти их) благодаря квантовому параллелизму и существенному увеличению скорости работы. По этой причине уже сейчас необходимо изучать их, в частности, применительно к задаче обнаружения вторжений в сетях SDN, поскольку в будущем COB на основе методов квантового машинного обучения вероятно смогут анализировать гораздо большие объемы информации с большей скоростью, что сделает их значительно эффективнее, чем традиционные IDS.

В данной работе проведен ряд экспериментов для метода квантового машинного обучения VQC и показано, что он может достигать результатов, сопоставимых с классическим методом градиентного бустинга, который является эталоном для задачи классификации табличных данных. При запуске модели VQC на квантовом компьютере, а не в симуляции, можно получить и прирост в скорости работы, помимо приемлемых результатов метрик.

Полученный в работе результат не является конечным, можно выделить два направления исследований для его улучшения. Во-первых, можно улучшать производительность текущей модели VQC путем подбора гиперпараметров и дополнительной предобработки данных. Во-вторых, можно рассмотреть и другие методы квантового машинного обучения, такие как QSVM, квантовые нейронные сети, гибридные квантово-классические архитектуры. Движение в обоих направлениях может привести к более эффективному решению задачи обнаружения вторжений в программно-конфигурируемых сетях.

Список использованных источников

- [1] Kukliansky A., Orescanin M., Bollmann C., Huffmire T. *Network anomaly detection using quantum neural networks on noisy quantum computers* // IEEE Transactions on Quantum Engineering.– 2024.– Vol. 5.– Pp. 1–11. [doi](#) ↑4
- [2] Кажемский М. А., Шелухин О. И. *Многоклассовая классификация сетевых атак на информационные ресурсы методами машинного обучения* // Труды учебных заведений связи.– 2019.– Т. 5.– № 1.– С. 107–115. [doi](#) ↑4
- [3] Zeguendry A., Jarir Z., Quafafou M. *Quantum machine learning: A review and case studies* // Entropy.– 2023.– Vol. 25.– No. 2.– id. 287.– 41 pp. [doi](#) ↑4, 8, 9
- [4] Volkov S. S., Kurochkin I. I. *Extraction of traffic features in software-defined networks using an SDN-controller*, Conference: 9th International Conference "Distributed Computing and Grid Technologies in Science and Education" (Dubna, Russia, July 5–9, 2021), CEUR Workshop Proceedings.– vol. 3041.– 2021.– Pp. 553–557. [URL](#) ↑4
- [5] Фёдоров Н. К. *Программно-конфигурируемые сети. Проблемы при переходе к сетям ПКС* // StudNet.– 2022.– Т. 5.– № 4.– С. 3137–3148. [URL](#) ↑4
- [6] Forbacha S. C., Kinteh M. K., Hamza E. M. *Enhanced attacks detection and mitigation in software defined networks* // American Journal of Computing and Engineering.– 2024.– Vol. 7.– No. 3.– Pp. 40–80. [doi](#) ↑4
- [7] Elsayed M. S., Le-Khac N. A., Jurcut A. D. *InSDN: A novel SDN intrusion dataset* // IEEE Access.– 2020.– Vol. 8.– Pp. 165263–165284. [doi](#) ↑5, 10
- [8] Wu Z., Wang J., Hu L., Zhang Zh., Wu H. *A network intrusion detection method based on semantic Re-encoding and deep learning* // Journal of Network and Computer Applications.– 2020.– Vol. 164.– id. 102688. [doi](#) ↑5
- [9] Yoon C., Lee S., Kang H., Park T., Shin S., Yegneswaran V., Porras Ph., Gu G. *Flow wars: Systemizing the attack surface and defenses in software-defined networks* // IEEE/ACM Transactions on Networking.– 2017.– Vol. 25.– No. 6.– Pp. 3514–3530. [doi](#) ↑6
- [10] Vedral V., Plenio M. B. *Basics of quantum computation* // Progress in Quantum Electronics.– 1998.– Vol. 22.– No. 1.– Pp. 1–39. [doi](#) ↑6, 7
- [11] Yang Z., Zolanvari M., Jain R. *A survey of important issues in quantum computing and communications* // IEEE Communications Surveys & Tutorials.– 2023.– Vol. 25.– No. 2.– Pp. 1059–1094. [doi](#) ↑6, 7
- [12] Торгаев С. Н., Шульга И. Д., Юрченко Е. А., Громов М. Л. *Основы квантовых вычислений*, учебное пособие.– Томск: STT.– 2020.– ISBN 978-5-93629-656-7.– 88 с. ↑6
- [13] Bacciagaluppi G. *The role of decoherence in quantum mechanics* // *The Stanford Encyclopedia of Philosophy*, substantive revision Thu Jan 23, 2025, eds. Edward N. Zalta, Uri Nodelman.– Metaphysics Research Lab, Stanford University.– 2025. [URL](#) ↑7
- [14] Luo J., Zhao P., Miao Zh., Zhao J. *A comprehensive study of bug fixes in quantum programs* // 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering, SANER (Honolulu, HI, USA, 15–18 March 2022).– IEEE.– 2022.– ISBN 978-1-6654-3786-8.– Pp. 1239–1246. [doi](#) ↑8

- [15] Shor P. W. *Algorithms for quantum computation: discrete logarithms and factoring* // *Proceedings 35th Annual Symposium on Foundations of Computer Science* (Santa Fe, NM, USA, 20–22 November 1994).– IEEE.– 1994.– ISBN 0-8186-6580-7.– Pp. 124–134.  
- [16] Shor P. W. *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer* // *SIAM Journal on Computing*.– Т. **26**.– № 5.– С. 1484–1509. arXiv  quant-ph/9508027  
- [17] Grover L. K. *A fast quantum mechanical algorithm for database search* // *STOC'96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (Philadelphia, Pennsylvania, USA, 22–24 May 1996).– 1996.– ISBN 978-0-89791-785-8.– Pp. 212–219.  
- [18] Tychola K. A., Kalampokas T., Papakostas G. A. *Quantum machine learning – an overview* // *Electronics*.– 2023.– Vol. **12**.– No. 11.– id. 2379.– 21 pp.  
- [19] Kalinin M., Krundyshev V. *Security intrusion detection using quantum machine learning techniques* // *Journal of Computer Virology and Hacking Techniques*.– 2023.– Vol. **19**.– No. 1.– Pp. 125–136.  
- [20] Zardini E., Blanzieri E., Pastorello D. *A quantum k-nearest neighbors algorithm based on the Euclidean distance estimation* // *Quantum Machine Intelligence*.– 2024.– Vol. **6**.– No. 1.– id. 23.– 22 pp.  
- [21] Basheer A., Afham A., Goyal S. K. *Quantum k-nearest neighbors algorithm*.– 2021.– 21 pp. arXiv  2003.09187  
- [22] Anguita D., Ridella S., Rivieccio F., Zunino R. *Quantum optimization for training support vector machines* // *Neural Networks*.– 2003.– Vol. **16**.– No. 5–6.– Pp. 763–770.  
- [23] Harrow A. W., Hassidim A., Lloyd S. *Quantum algorithm for linear systems of equations* // *Physical review letters*.– 2009.– Vol. **103**.– No. 15.– id. 150502.  
- [24] Yang J., Awan A. J., Vall-Llosera G. *Support vector machines on noisy intermediate scale quantum computers*.– 2019.– 12 pp. arXiv  1909.11988  
- [25] Meedinti G. N., Sreekha K. S., Delhibabu R. *A quantum convolutional neural network approach for object detection and classification*.– 2023.– 16 pp. arXiv  2307.08204  
- [26] Chawla N. V., Bowyer K. W., Hall L. O., Kegelmeyer W. P. *SMOTE: Synthetic minority over-sampling technique* // *Journal of Artificial Intelligence Research*.– 2002.– Vol. **16**.– No. 1.– Pp. 321–357.  
- [27] Volkov S. S., Kurochkin I. I. *Network attacks classification using Long Short-term memory based neural networks in Software-Defined Networks* // *Procedia Computer Science*.– 2020.– Vol. **178**.– Pp. 394–403.  

Поступила в редакцию	10.03.2025;
одобрена после рецензирования	26.04.2025;
принята к публикации	22.05.2025;
опубликована онлайн	18.07.2025.

Рекомендовал к публикации

к.т.н. В. П. Фраленко

Информация об авторах:



Илья Андреевич Антонов

Выпускник магистратуры Национального исследовательского технологического университета МИСИС по направлению 09.04.03 Прикладная информатика. Области научных интересов: методы машинного обучения, нейронные сети, квантовые вычисления.



0009-0002-9820-1558

e-mail:



Илья Ильич Курочкин

Кандидат технических наук, заведующий лабораторией Ц-1 Института проблем передачи информации им. А.А. Харкевича Российской академии наук. Области научных интересов: моделирование телекоммуникационных сетей, облачные и распределенные вычисления, распределенное глубокое обучение.



0000-0002-0399-6208

e-mail:

Авторы внесли равный вклад в подготовку публикации.

Декларация об отсутствии личной заинтересованности: благополучие авторов не зависит от результатов исследования.

UDC 004.272.45

 10.25209/2079-3316-2025-16-3-3-22

Quantum machine learning methods for intrusion detection in software-defined networks

Il'ya Andreevich **Antonov**¹, Il'ya Il'ich **Kurochkin**²

¹ National University of Science and Technology MISIS, Moscow, Russia

² Institute for Information Transmission Problems of RAS, Moscow, Russia

 m1908142@edu.misis.ru

Abstract. Software-defined network architecture is the preferred way to build large computer networks that require high responsiveness to change and a high degree of automation. The main feature of this architecture is the centralized management of the entire network from a single controller. However, this approach opens new opportunities for attacks on the network, making the controller their main target. This paper explores the possibility of applying quantum machine learning models to detect such attacks. (*In Russian*).

Key words and phrases: software-defined networks, information security, machine learning, neural networks, quantum computing, intrusion detection systems, SDN, IDS

2020 *Mathematics Subject Classification:* 65Y05; 68Q10

For citation: Il'ya A. Antonov, Il'ya I. Kurochkin. *Quantum machine learning methods for intrusion detection in software-defined networks*. Program Systems: Theory and Applications, 2025, **16**:3(66), pp. 3–22. (*In Russ.*).

https://psta.psiras.ru/read/psta2025_3_3-22.pdf

References

- [1] A. Kukliansky, M. Orescanin, C. Bollmann, T. Huffmire. “Network anomaly detection using quantum neural networks on noisy quantum computers”, *IEEE Transactions on Quantum Engineering*, **5** (2024), pp. 1–11. [doi](#)
- [2] M. A. Kazhetskij, O. I. Sheluxin. “Multiclass classification of attacks to information resources with machine learning techniques”, *Trudy uchebnyx zavedenij svyazi*, **5:1** (2019), pp. 107–115 (in Russian). [doi](#)
- [3] A. Zeguendry, Z. Jarir, M. Quafafou. “Quantum machine learning: A review and case studies”, *Entropy*, **25:2** (2023), id. 287, 41 pp. [doi](#)
- [4] S. S. Volkov, I. I. Kurochkin. “Extraction of traffic features in software-defined networks using an SDN-controller”, Conference: 9th International Conference "Distributed Computing and Grid Technologies in Science and Education" (Dubna, Russia, July 5–9, 2021), CEUR Workshop Proceedings, vol. **3041**, 2021, pp. 553–557. [URL](#)
- [5] N. K. Fyodorov. “Software-defined networks. Problems in the transition to software-defined networks”, *StudNet*, **5:4** (2022), pp. 3137–3148 (in Russian). [URL](#)
- [6] S. C. Forbacha, M. K. Kinteh, E. M. Hamza. “Enhanced attacks detection and mitigation in software defined networks”, *American Journal of Computing and Engineering*, **7:3** (2024), pp. 40–80. [doi](#)
- [7] M. S. Elsayed, N. A. Le-Khac, A. D. Jurcut. “InSDN: A novel SDN intrusion dataset”, *IEEE Access*, **8** (2020), pp. 165263–165284. [doi](#)
- [8] Z. Wu, J. Wang, L. Hu, Zh. Zhang, H. Wu. “A network intrusion detection method based on semantic Re-encoding and deep learning”, *Journal of Network and Computer Applications*, **164** (2020), id. 102688. [doi](#)
- [9] C. Yoon, S. Lee, H. Kang, T. Park, S. Shin, V. Yegneswaran, Ph. Porras, G. Gu. “Flow wars: Systemizing the attack surface and defenses in software-defined networks”, *IEEE/ACM Transactions on Networking*, **25:6** (2017), pp. 3514–3530. [doi](#)
- [10] V. Vedral, M. B. Plenio. “Basics of quantum computation”, *Progress in Quantum Electronics*, **22:1** (1998), pp. 1–39. [doi](#)
- [11] Z. Yang, M. Zolanvari, R. Jain. “A survey of important issues in quantum computing and communications”, *IEEE Communications Surveys & Tutorials*, **25:2** (2023), pp. 1059–1094. [doi](#)
- [12] S. N. Torgaev, I. D. Shul’ga, E. A. Yurchenko, M. L. Gromov. *Fundamentals of quantum computing*, uchebnoe posobie, STT, Tomsk, 2020, ISBN 978-5-93629-656-7, 88 pp. (in Russian).
- [13] G. Bacciagaluppi. “The role of decoherence in quantum mechanics”, *The Stanford Encyclopedia of Philosophy*, substantive revision Thu Jan 23, 2025, eds. Edward N. Zalta, Uri Nodelman, Metaphysics Research Lab, Stanford University, 2025. [URL](#)

- [14] J. Luo, P. Zhao, Zh. Miao, J. Zhao. “A comprehensive study of bug fixes in quantum programs”, *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering*, SANER (Honolulu, HI, USA, 15–18 March 2022), IEEE, 2022, ISBN 978-1-6654-3786-8, pp. 1239–1246. [doi](#)
- [15] P. W. Shor. “Algorithms for quantum computation: discrete logarithms and factoring”, *Proceedings 35th Annual Symposium on Foundations of Computer Science* (Santa Fe, NM, USA, 20–22 November 1994), IEEE, 1994, ISBN 0-8186-6580-7, pp. 124–134. [doi](#)
- [16] P. W. Shor. “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”, *SIAM Journal on Computing*, **26**:5, pp. 1484–1509. [doi](#) [arXiv:quant-ph/9508027](#)
- [17] L. K. Grover. “A fast quantum mechanical algorithm for database search”, *STOC'96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (Philadelphia, Pennsylvania, USA, 22–24 May 1996), 1996, ISBN 978-0-89791-785-8, pp. 212–219. [doi](#)
- [18] K. A. Tychola, T. Kalampokas, G. A. Papakostas. “Quantum machine learning — an overview”, *Electronics*, **12**:11 (2023), id. 2379, 21 pp. [doi](#)
- [19] M. Kalinin, V. Krundyshev. “Security intrusion detection using quantum machine learning techniques”, *Journal of Computer Virology and Hacking Techniques*, **19**:1 (2023), pp. 125–136. [doi](#)
- [20] E. Zardini, E. Blanzieri, D. Pastorello. “A quantum k-nearest neighbors algorithm based on the Euclidean distance estimation”, *Quantum Machine Intelligence*, **6**:1 (2024), id. 23, 22 pp. [doi](#)
- [21] A. Basheer, A. Afham, S. K. Goyal. *Quantum k-nearest neighbors algorithm*, 2021, 21 pp. [arXiv:2003.09187](#) [doi](#)
- [22] D. Anguita, S. Ridella, F. Rivieccio, R. Zunino. “Quantum optimization for training support vector machines”, *Neural Networks*, **16**:5–6 (2003), pp. 763–770. [doi](#)
- [23] A. W. Harrow, A. Hassidim, S. Lloyd. “Quantum algorithm for linear systems of equations”, *Physical review letters*, **103**:15 (2009), id. 150502. [doi](#)
- [24] J. Yang, A. J. Awan, G. Vall-Llosera. *Support vector machines on noisy intermediate scale quantum computers*, 2019, 12 pp. [arXiv:1909.11988](#)
- [25] G. N. Meedinti, K. S. Sirekha, R. Delhibabu. *A quantum convolutional neural network approach for object detection and classification*, 2023, 16 pp. [arXiv:2307.08204](#)
- [26] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer. “SMOTE: Synthetic minority over-sampling technique”, *Journal of Artificial Intelligence Research*, **16**:1 (2002), pp. 321–357. [doi](#)
- [27] S. S. Volkov, I. I. Kurochkin. “Network attacks classification using Long Short-term memory based neural networks in Software-Defined Networks”, *Procedia Computer Science*, **178** (2020), pp. 394–403. [doi](#)

УДК 517.977.5

 10.25209/2079-3316-2025-16-3-23-40

Метод сильного улучшения управления для неоднородных дискретных систем

Ирина Викторовна **Расина**^{1✉}, Ирина Сергеевна **Гусева**²

¹Институт программных систем им. А. К. Айламазяна РАН, Вельсково, Россия

²Бурятский государственный университет, Улан-Удэ, Россия

[✉]irinarasina@gmail.com

Аннотация. Рассматривается класс неоднородных дискретных систем (НДС) с промежуточными критериями. Такие системы являются двухуровневыми и распространены на практике, а также получаются при дискретизации непрерывных систем в процессе решения задач оптимизации итерационными методами. Для указанного класса на основе аналога достаточных условий оптимальности Кротова строится метод сильного улучшения второго порядка.

Авторы статьи ставят под сомнение утверждение, что для классических дискретных управляемых систем, а также и для неоднородных, нет смысла вводить понятие сильного относительного минимума. Поэтому при построении метода улучшения ими выдвинуто требование близости соседних приближений из класса допустимых только по состояниям процесса на обоих уровнях. Полученный метод содержит векторно-матричную двухуровневую систему для сопряженных переменных. Приращение управлений на каждом из уровней линейно зависит от соответствующих состояний, что позволяет найти решение в форме приближенного линейного синтеза оптимального управления.

Проведена апробация метода на двух иллюстративных примерах, показавшая его работоспособность. Применение разработанного метода к более сложному примеру позволило получить меньшее значение функционала, чем найденное ранее аналогичным по структуре методом минимаксного улучшения.

Ключевые слова и фразы: неоднородные дискретные системы, промежуточные критерии, оптимальное управление

Для цитирования: Расина И. В., Гусева И. С. *Метод сильного улучшения управления для неоднородных дискретных систем* // Программные системы: теория и приложения. 2025. Т. 16. № 3(66). С. 23–40. https://psta.psir.ru/read/psta2025_3_23-40.pdf

Введение

Системы неоднородной структуры, т.е. процессы с изменяющимся во времени описанием, как непрерывные, так и дискретные, широко распространены на практике. Примерами могут служить сложные космические операции (например, перелет между планетами), динамика роботов, процессы химического производства, развитие организмов и популяций. Описание некоторых из них приведено в [1]. Далее будут рассматриваться неоднородные дискретные процессы (НДС).

Поскольку непосредственное использование необходимых и достаточных условий оптимальности для различных типов управляемых систем не представляется возможным из-за трудностей разрешимости в аналитическом виде содержащихся в них соотношений, то внимание исследователей всегда было сосредоточено на построении на их основе разнообразных итерационных методов. При этом следует заметить, что львиная доля таких методов позволяет найти лишь локальный, а не глобальный минимум функционала. Подобные методы более просты как в их разработке, так и реализации.

В классической задаче оптимального управления для непрерывных систем различают сильный и слабый относительный минимум [2]. Итак, пусть $\bar{z} = (\bar{x}, \bar{u})$, где \bar{x} — состояние, а \bar{u} — управление, элемент из класса допустимых, на котором функционал достигает относительного минимума. Будем называть такой относительный минимум сильным, если он достигается среди элементов z , удовлетворяющих условию $|x - \bar{x}| < \varepsilon$. И в свою очередь слабым, если он достигается среди элементов z , удовлетворяющих условиям $|x - \bar{x}| < \varepsilon, |u - \bar{u}| < \varepsilon$.

В дискретных системах в силу специфики их структуры говорят просто об относительном минимуме, который по своей сути является слабым [2]. Здесь достаточно требования близости элементов по управлению, из которого автоматически следует близость по состоянию.

Если же речь идет об управляемых системах неоднородной структуры, которые являются двухуровневыми, где на нижнем уровне чередуются непрерывные системы, а на верхнем — дискретные (ДНС), естественным для нижнего уровня будет требование близости элементов, как только по состоянию, так по состоянию и управлению. Для верхнего уровня только по управлению. По схеме: на нижнем близость по состоянию, на верхнем — по управлению, был построен метод улучшения в [3]. Когда же оба уровня дискретные (НДС), то по аналогии с классикой можно говорить о близости по управлению на обоих уровнях.

И все же возникает естественный вопрос: а можно ли для НДС построить метод, в котором на обоих уровнях элементы близки лишь по состоянию и будет ли этот метод работоспособным. Далее в статье рассматривается именно эта ситуация.

При постановке задачи оптимального управления использовалась математическая модель НДС с промежуточными критериями из [4], а для построения метода — аналог достаточных условий оптимальности Кротова, полученный в этой же работе.

1. Неоднородные дискретные процессы с промежуточными критериями

Рассмотрим двухуровневую модель, в которой нижний уровень составляют дискретные динамические системы однородной структуры. На верхнем уровне фигурирует дискретная модель общего вида

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

где k — номер шага (этапа), x и u — соответственно переменные состояния и управления произвольной природы (возможно различной) для различных k , $\mathbf{U}(k, x)$ — заданное при каждом k и x множество.

На некотором подмножестве $\mathbf{K}' \subset \mathbf{K}$, $k_F \notin \mathbf{K}'$, $u(k)$ интерпретируется как пара $(u^v(k), m^d(k))$, где $m^d(k)$ — процесс $(x^d(k, t), u^d(k, t))$, $t \in \mathbf{T}(k, z(k))$, $m^d(k) \in \mathbf{D}^d(k, z(k))$, а \mathbf{D}^d — множество допустимых процессов m^d , удовлетворяющих системе

$$(2) \quad \begin{aligned} x^d(k, t+1) &= f^d(k, z, t, x^d(k, t), u^d(k, t)), \\ t \in \mathbf{T} &= \{t_I(z), t_I(z) + 1, \dots, t_F(z)\}, \\ x^d \in \mathbf{X}^d(k, z, t), \quad u^d \in \mathbf{U}^d(k, z, t, x^d), \quad z &= (k, x, u^v). \end{aligned}$$

Здесь u^v управляющее воздействие верхнего уровня на нижний. Для этой системы на множестве \mathbf{T} задана промежуточная цель в виде функционала, который необходимо минимизировать:

$$I^k = \sum_{\mathbf{T}(z) \setminus t_F(z)} f^k(t, x^d(k, t), u^d(k, t)) \rightarrow \inf.$$

Здесь $\mathbf{X}^d(k, z, t)$, $\mathbf{U}^d(k, z, t, x^d)$ — заданные при каждом t, z и x^d множества. Оператор правой части (1) сводится к следующему:

$$\begin{aligned} f(k, x, u) &= \theta(z, \gamma^d(z)), \quad \gamma^d = (t_I, x_I^d, t_F, x_F^d) \in \mathbf{\Gamma}^d(k, z), \\ \mathbf{\Gamma}^d(z) &= \{\gamma^d: t_I = \tau(k, z), t_F = \vartheta(k, z), x_I^d = \xi(k, z), x_F^d \in \mathbf{\Gamma}_F^d(k, z)\}, \end{aligned}$$

где θ и τ, ϑ, ξ — заданные оператор и функции.

На множестве \mathbf{D} процессов

$$m = (x(k), u(k), x^d(k, t), u^d(k, t)),$$

удовлетворяющих (1), (2), рассматривается задача оптимального управления о минимизации конечного функционала $I = F(x(k_F))$ при фиксированных $k_I = 0, k_F, x(k_I)$ и дополнительных ограничениях $x(k) \in \mathbf{X}(k)$.

2. Используемый математический аппарат

Далее для удобства читателей приведем используемые для построения метода конструкции и теоремы.

Имеем [4]:

$$\begin{aligned} L &= G(x(k_F)) - \sum_{\mathbf{K} \setminus \mathbf{K}' \setminus k_F} R(k, x(k), u(k)) + \\ &+ \sum_{\mathbf{K}'} \left(G^d(z) - \sum_{\mathbf{T}(z) \setminus t_F} R^d(z, t, x^d(k, t), u^d(k, t)) \right), \end{aligned}$$

$$G(x(k_F)) = F(x(k_F)) + \varphi(k_F, x(k_F)) - \varphi(k_I, x(k_I)),$$

$$R(k, x, u) = \varphi(k+1, f(k, x, u)) - \varphi(k, x),$$

$$\begin{aligned} G^d(k, z, \gamma^d) &= -\varphi(k+1, \theta(k, z, \gamma^d)) + \varphi(k, x(k)) + \\ &+ \varphi^d(k, z, t_F, x_F^d) - \varphi^d(k, z, t_I, x_I^d), \end{aligned}$$

$$\begin{aligned} R^d(k, z, t, x^d, u^d) &= \varphi^d((k, z, t+1, f^d(k, z, t, x^d, u^d)) - \\ &- f^k(t, x^d(k, t), u^d(k, t)) - \varphi^d(k, z, t, x^d), \end{aligned}$$

$$\mu^d(k, z, t) = \sup \{ R^d(k, z, t, x^d, u^d) : x^d \in \mathbf{X}^d(k, z, t), u^d \in \mathbf{U}^d(k, z, t, x^d) \},$$

$$l^d(k, z) = \inf \{ G^d(k, z, \gamma^d) : \gamma^d \in \mathbf{\Gamma}^d(k, z), x^d \in \mathbf{X}^d(k, z, t_F) \},$$

$$\mu(k) = \begin{cases} \sup \{ R(k, x, u) : x \in \mathbf{X}(k), u \in \mathbf{U}(k, x) \}, & t \in \mathbf{K} \setminus \mathbf{K}', \\ -\inf \{ l^d(z) : x \in \mathbf{X}(k), u^v \in \mathbf{U}^v(k, x) \}, & k \in \mathbf{K}', \end{cases}$$

$$l = \inf \{ G(x) : x \in \mathbf{\Gamma} \cap \mathbf{X}(k_F) \}.$$

Здесь $\varphi(k, x)$ — произвольный функционал, $\varphi^d(k, z, t, x^d)$ — произвольное параметрическое семейство функционалов (с параметрами k, z). В дальнейшем указанные функционалы φ, φ^d подлежат определению.

ТЕОРЕМА 1. Для любого элемента $m \in \mathbf{D}$ и любых φ, φ^d имеет место оценка

$$I(m) - \inf_{\mathbf{D}} I \leq \Delta = I(m) - l.$$

Пусть имеются два процесса $m^I \in \mathbf{D}$ и $m^II \in \mathbf{E}$ и функции φ и φ^d такие, что $L(m^II) < L(m^I) = I(m^I)$, и $m^II \in \mathbf{D}$.

Тогда $I(m^II) < I(m^I)$.

Здесь \mathbf{E} — множество, включающее в себя множество \mathbf{D} , [2].

ТЕОРЕМА 2. Пусть имеются последовательность процессов $\{m_s\} \subset \mathbf{D}$ и функционалы φ , φ^d , такие что:

- 1) $R(k, x_s(k), u_s(k)) \rightarrow \mu(k)$, $k \in \mathbf{K}$;
- 2) $R^d(z_s, t, x_s^d(t), u_s^d(t)) - \mu^d(z_s, t) \rightarrow 0$, $k \in \mathbf{K}'$, $t \in \mathbf{T}(z_s)$;
- 3) $G^d(z_s, \gamma_s^d) - l^d(z_s) \rightarrow 0$, $k \in \mathbf{K}'$;
- 4) $G(x_s(t_F)) \rightarrow l$.

Тогда последовательность $\{m_s\}$ — минимизирующая для I на \mathbf{D} .

3. Метод улучшения

Здесь и ниже предполагается, что все используемые объекты обладают необходимыми для выполняемых операций свойствами, такими как непрерывность, дифференцируемость и т.д.

Предположим, что $\mathbf{X}(k) = \mathbb{R}^{m(k)}$, $\mathbf{X}^d(z, t) = \mathbb{R}^{n(k)}$, $x_I^d = \xi(z)$, k_I , x_I , k_F , $t_I(k)$, $t_F(k)$ — заданы, $x_F^d \in \mathbb{R}^{n(k)}$ и системы нижнего уровня не зависят от управления u^v . Последнее требование позволяет избежать излишней громоздкости метода. Рассматриваемая далее задача улучшения состоит, по существу, в построении оператора $\theta(m)$, $\theta: \mathbf{D} \rightarrow \mathbf{D}$, такого что $I(\theta(m)) \leq I(m)$ [4]. При некотором заданном начальном элементе такой оператор генерирует улучшающую, в частности, минимизирующую последовательность $\{m_s\}: m_{s+1} = \theta(m_s)$.

Будем строить метод на основе принципов расширения [5] и локализации [6]. Согласно последнему, задача улучшения некоторого элемента m^I сводится к задаче о минимуме вспомогательного функционала

$$(3) \quad I_\alpha(m) = \alpha I(m) + (1 - \alpha) J(m^I, m), \quad \alpha \in [0, 1],$$

где $J(m^I, m)$ — функционал типа метрики. Изменяя α от 0 к 1, можно достичь необходимой степени близости m_α к m^I и эффективно использовать аппроксимации конструкций достаточных условий в окрестности m^I . В итоге получается алгоритм с параметром α , играющим роль регулятора, настраиваемого при конкретном применении. Этот параметр выбирается так, чтобы разность $I(m^I) - I(m_\alpha)$ была наибольшей, тогда соответствующий элемент m_α принимается за m^II .

Вспомогательный функционал зададим в виде:

$$I_\alpha = \alpha I + \frac{1}{2} (1 - \alpha) \left(\sum_{\mathbf{K} \setminus \mathbf{K}'} |\Delta x(k)|^2 + \sum_{\mathbf{K}'} \sum_{\mathbf{T}(z) \setminus t_F} |\Delta x^d(k, t)|^2 \right),$$

где $\alpha \in [0, 1]$, $\Delta x = x - x^I$, $\Delta x^d = x^d - x^{dI}$. Из выше приведенной формулы следует, что в данном случае функционал $J(m^I, m)$ представляет собой половину квадрата по состояниям процесса.

Согласно указанному принципу расширения по заданному элементу $m^I \in \mathbf{D}$ требуется найти элемент $m^{II} \in \mathbf{D}$, на котором $I_\alpha(m^{II}) = L_\alpha(m^{II}) < I_\alpha(m^I) = L_\alpha(m^I)$, или $L_\alpha(m^{II}) - L_\alpha(m^I) < 0$. Рассмотрим приращение функционала $L_\alpha(m)$, по формуле Тейлора имеем:

$$\begin{aligned} \Delta L_\alpha &\approx G_{x_F}^T \Delta x_F + \frac{1}{2} \Delta x_F^T G_{x_F x_F} \Delta x_F - \\ &- \sum_{\mathbf{K} \setminus \mathbf{K}' \setminus k_F} \left(R_x^T \Delta x + R_u^T \Delta u + \frac{1}{2} \Delta u^T R_{uu} \Delta u + \frac{1}{2} \Delta x^T R_{xx} \Delta x + \right. \\ &+ \left. \Delta x^T R_{xu} \Delta u \right) + \sum_{\mathbf{K}'} \left(G_{x_F^d}^{dT} \Delta x_F^d + G_x^{dT} \Delta x + \frac{1}{2} \Delta(x_F^d)^T G_{x_F^d x_F^d}^d \Delta x_F^d + \right. \\ &+ \left. \Delta(x_F^d)^T G_{x_F^d x_F}^d \Delta x_F \right) - \sum_{\mathbf{T}(z) \setminus t_F} \left(R_{x^d}^{dT} \Delta x^d + R_x^{dT} \Delta x + R_{u^d}^{dT} \Delta u^d + \right. \\ &+ \left. \frac{1}{2} \Delta u^{dT} R_{u^d u^d}^d \Delta u^d + \frac{1}{2} \Delta x^{dT} R_{x^d x^d}^d \Delta x^d + \Delta x^T R_{xx}^d \Delta x + \right. \\ &+ \left. \Delta x^{dT} R_{x^d u^d}^d \Delta u^d + \Delta x^T R_{x u^d}^d \Delta u^d + \Delta x^{dT} R_{x^d x}^d \Delta x \right), \end{aligned}$$

где $\Delta u = u - u^I$, $\Delta x = x - x^I$, $\Delta u^d = u^d - u^{dI}$, $\Delta x^d = x^d - x^{dI}$, $\Delta x_F^d = x_F^d - x_F^{dI}$, $x_F = x(k_F)$. Здесь функции R , G , R^d , G^d определены для функционала I_α , а их первые и вторые производные подсчитаны при $u = u^I(k)$, $x = x^I(k)$, $x^d = x^{dI}(k, t)$, $u^d = u^{dI}(k, t)$.

Найдем Δu , Δu^d доставляющие максимум выражениям, стоящим под знаками сумм $\sum_{\mathbf{K} \setminus \mathbf{K}' \setminus k_F}$ и $\sum_{\mathbf{T}(z) \setminus t_F}$ соответственно. Имеем:

$$R_{u^d}^d + R_{u^d u^d}^d \Delta u^d + R_{x^d u^d}^d \Delta x^d + R_{x u^d}^d \Delta x = 0.$$

Тогда $\Delta u^d = -(R_{u^d}^d + R_{u^d x^d}^d \Delta x^d + R_{x u^d}^d \Delta x)^{-1} (R_{u^d}^d + R_{u^d x^d}^d \Delta x^d + R_{x u^d}^d \Delta x)$. Аналогично $\Delta u = -(R_{uu})^{-1} (R_u + R_{xu} \Delta x)$. Найденные формулы для приращений управлений обоих уровней подставим в выражение для приращения функционала ΔL_α и выполним необходимые преобразования, итог которых обозначим

через ΔM_α . Имеем:

$$\begin{aligned}
 \Delta M_\alpha &\approx G_{x_F}^\top \Delta x_F + \frac{1}{2} \Delta x_F^\top G_{x_F x_F} \Delta x_F - \\
 &- \sum_{\mathbf{K} \setminus \mathbf{K}' \setminus k_F} \left((R_x - R_{xu} (R_{uu})^{-1} R_u^\top) \Delta x + \right. \\
 &+ \frac{1}{2} \Delta x^\top (R_{xx} - R_{xu} (R_{uu})^{-1} R_{xu}^\top) \Delta x - \frac{1}{2} R_u^\top (R_{uu})^{-1} R_u \left. \right) + \\
 &+ \sum_{\mathbf{K}'} \left(G_{x_F^d}^{d\top} \Delta x_F^d + G_x^{d\top} \Delta x + \frac{1}{2} \Delta (x_F^d)^\top G_{x_F^d x_F^d}^d \Delta x_F^d + \right. \\
 &+ \Delta (x_F^d)^\top G_{x_F^d x_F}^d \Delta x_F \left. \right) - \\
 &- \sum_{\mathbf{T}(z) \setminus t_F} \left((R_{x^d}^d - R_{x^d u^d}^d (R_{u^d u^d}^d)^{-1} R_{u^d}^{d\top}) \Delta x^d + \right. \\
 &+ \frac{1}{2} \Delta x^{d\top} (R_{x^d x^d}^d - R_{x^d u^d}^d (R_{u^d u^d}^d)^{-1} R_{x^d u^d}^{d\top}) \Delta x^d - \\
 &+ (R_x^d - R_{x u^d}^d (R_{u^d u^d}^d)^{-1} R_{u^d}^{d\top}) \Delta x + \\
 &+ \Delta x^\top (R_{xx}^d - R_{x u^d}^d R_{u^d u^d}^{d-1} R_{x u^d}^{d\top}) \Delta x + \\
 &+ \Delta x^\top (R_{x x^d}^d - R_{x u^d}^d (R_{u^d u^d}^d)^{-1} R_{x^d u^d}^d) \Delta x^d - \frac{1}{2} R_{u^d}^{d\top} (R_{u^d u^d}^d)^{-1} R_{u^d}^d \left. \right).
 \end{aligned}$$

Зададим функции φ и $\varphi^d(z)$ в виде:

$$\begin{aligned}
 \varphi &= \psi^\top(k) x(k) + \frac{1}{2} \Delta x^\top(k) \sigma(k) \Delta x(k), \\
 \varphi^d &= \lambda^\top(k, t) x(k) + \psi^{d\top}(k, t) x^d(k, t) + \\
 &+ \frac{1}{2} \Delta x^{d\top}(k, t) \sigma^d(k, t) \Delta x^d(k, t) + \frac{1}{2} \Delta x^\top(k) \Lambda(k, t) \Delta x^d(k, t) + \\
 &+ \frac{1}{2} \Delta x^{d\top}(k, t) \Lambda^T(k, t) \Delta x(k) + \frac{1}{2} \Delta x^\top(k) \sigma(k, t) \Delta x(k),
 \end{aligned}$$

где ψ, ψ^d, λ – вектор-функции, $\sigma, \sigma^d, \Lambda$ – матрицы, и таким образом, чтобы приращение функционала ΔM_α не зависело от $\Delta x, \Delta x_F, \Delta x_F^d, \Delta x^d$.

Последнее требование будет достигнуто, если:

$$\begin{aligned}
 G_x &= 0, \quad G_{x_F^d}^d = 0, \quad G_{xx} = 0, \quad G_{x_F^d x_F^d}^d = 0, \quad G_{x_F^d}^d = 0, \\
 R_x - R_{xu} R_{uu}^{-1} R_u^\top &= 0, \quad R_{xx} - R_{xu} R_{uu}^{-1} R_{xu}^\top = 0, \\
 R_{x^d}^d - R_{x^d u^d}^d (R_{u^d u^d}^d)^{-1} R_{u^d}^{d\top} &= 0, \quad R_{x x^d}^d - R_{x u^d}^d (R_{u^d u^d}^d)^{-1} R_{x^d u^d}^{d\top} = 0.
 \end{aligned}$$

Расшифровка указанных условий приводит к задаче Коши для НДС относительно $\psi, \psi^d, \lambda, \sigma, \sigma^d, \Lambda$ с начальными условиями на правом

конце, в которых E — единичные матрицы соответствующих размеров:

$$\begin{aligned}
\psi(k_F) &= -\alpha F_{x(k_F)}, \\
\psi(k) &= H_x - (f_x^\top \sigma(k+1)f_u + H_{xu}) \times \\
&\quad \times (f_u^\top \sigma(k+1)f_u + (H_{uu})^{-1}H_u^\top), \quad k \in \mathbf{K} \setminus \mathbf{K}' \setminus k_F, \\
\sigma(k_F) &= -\alpha F_{x(k_F)x(k_F)} + (1-\alpha)E, \\
\sigma(k) &= f_x^\top \sigma(k+1)f_x + H_{xx} - (f_x^\top \sigma(k+1)f_u + H_{xu}) \times \\
&\quad \times (f_u^\top \sigma(k+1)f_u + H_{uu})^{-1} (f_x^\top \sigma(k+1)f_u + H_{xu})^\top, \\
&\quad k \in \mathbf{K} \setminus \mathbf{K}' \setminus k_F, \\
\psi^d(k, t_F) &= H_{x_F^d}^d(\psi(k+1)), \\
\psi^d(k, t) &= H_{x^d}^d + (f_{x^d}^{dT} \sigma^d(k, t+1)f_{u^d}^d + H_{x^d u^d}^d)(H_{u^d u^d}^d)^{-1} H_{u^d}^{dT}, \\
&\quad k \in \mathbf{K}', \\
\sigma^d(k, t_F) &= \theta_{x_F^d}^\top \sigma(k+1)\theta_{x_F^d} + H_{x_F^d x_F^d}^d, \\
\sigma^d(k, t) &= f_{x^d}^{dT} \sigma^d(k, t+1)f_{x^d}^d + H_{x^d x^d}^d - (f_{x^d}^{dT} \sigma^d(k, t+1)f_{u^d}^d + \\
&\quad + H_{x^d u^d}^d) \times (f_{u^d}^{dT} \sigma^d(k, t+1)f_{u^d}^d + H_{u^d u^d}^d)^{-1} \times \\
&\quad \times (f_{x^d}^{dT} \sigma^d(k, t+1)f_{u^d}^d + H_{x^d u^d}^d)^\top, \quad k \in \mathbf{K}', \\
\psi(k) &= H_x(\psi(k+1)) + \lambda(t_I) + \xi_x^\top \lambda(t_I) - \lambda(t_F), \\
\lambda(k, t_F) &= H_{x_F}(\psi(k+1)), \\
\lambda(k, t) &= H_x^d(\psi^d(k, t+1)).
\end{aligned}$$

Здесь для краткости в правых частях указаны лишь аргументы $\psi(k+1)$ и $\psi^d(k, t+1)$, необходимые для понимания соотношений.

$$\begin{aligned}
\sigma(k) &= \theta_x^\top \sigma(k+1)\theta_x + H_{xx} + \xi_x^\top \theta_{x_i^d} \sigma(k+1)\theta_x + \\
&\quad + \theta_x^\top \sigma(k+1)\theta_{x_i^d} \xi_x + \xi_x^\top \theta_{x_i^d}^\top \sigma(k+1)\theta_{x_i^d} \xi_x + \\
&\quad + \xi_x^\top \sigma^d(k, t_i) \xi_x + \xi_x \Lambda(k, t_i) + \Lambda^\top(k, t_i) \xi_x^\top, \\
\Lambda(k, t) &= f_{x^d}^d \Lambda(k, t+1) f_{x^d}^{dT} - f_{x^d}^d, \\
\Lambda(k, t_F) &= \theta_x^\top \sigma(k+1)\theta_{x_i^d} + H_{x^d x^d}(k, t_i), \\
H(k, \psi, x_I^d, x_F^d) &= \psi^\top \theta(z, x_I^d, x_F^d), \quad k \in \mathbf{K}', \\
H^d(k, t, z, \psi^d, x^d, u^d) &= \psi^{dT} f^d(k, z, x^d, u^d) - \\
&\quad - f^k(k, z, x^d, u^d) - (1-\alpha) \frac{1}{2} |\Delta x^d(k, t)|^2,
\end{aligned}$$

$$x(k_i) = x_i, \quad x(k_F) = x_F, \quad x^d(k, t_i) = x_i^d, \quad x^d(k, t_F) = x_F^d.$$

При этом приращения управляющих воздействий имеют вид:

$$\begin{aligned} \Delta u(k) &= -(f_u^T \sigma(k+1) f_u + H_{uu})^{-1} \times \\ &\quad \times (H_u + (f_x^T \sigma(k+1) f_u + H_{xu})^T \Delta x(k)), \\ \Delta u^d(k, t) &= -(f_{u^d}^{dT} \sigma^d(k, t+1) f_{u^d}^d + H_{u^d u^d}^d)^{-1} \times \\ &\quad \times \left(H_{u^d}^d + (f_{x^d}^{dT} \sigma^d(k, t+1) f_{u^d}^d + H_{x^d u^d}^d)^T \Delta x^d(k, t) + \right. \\ &\quad \left. + (f_x^{dT} \sigma^d(k, t+1) f_{u^d}^d + H_{x u^d}^d)^T \Delta x(k) \right). \end{aligned}$$

ЗАМЕЧАНИЕ 1. Отметим тот факт, что приращения управлений обоих уровней зависят от соответствующих состояний этих же уровней. Следовательно решение поставленной задачи представляет собой линейный приближенный синтез оптимального управления.

ЗАМЕЧАНИЕ 2. Уравнения для матриц σ и σ^d представляют собой матричные уравнения Риккати, которые могут и не иметь решения. В этом случае алгоритм должен быть дополнен специальной процедурой построения управляющих воздействий. Разработка такой процедуры требует проведения специального исследования и подбора примеров, что выходит за рамки этой статьи.

4. Итерационная процедура

На основе полученных соотношений можно сформулировать следующую итерационную процедуру.

1. «Слева направо» просчитывается НДС (1), (2) при $u = u_s(k)$, $u^d = u_s^d(k, t)$ и заданных начальных условиях, получается соответствующая траектория $(x_s(k), x_s^d(k, t))$.
2. «Справа налево» разрешается НДС относительно $\psi(k)$, $\psi^d(k, t)$, $\lambda(k, t)$, $\sigma(k)$, $\sigma^d(k, t)$, $\Lambda(k, t)$.
3. Находятся Δu , Δu^d и новые управления $u = u_s(k) + \Delta u$, $u^d = u_s^d(k, t) + \Delta u^d$.
4. При найденных управлениях и начальном условии $x(k_I) = x_I$ просчитывается «слева направо» исходная НДС (1), (2). Тем самым определяется новый элемент m_{s+1} .

Процесс итераций заканчивается, когда $|I_{s+1} - I_s| \approx 0$ с заданной точностью.

Работоспособность построенного метода была проверена на двух иллюстративных примерах. Пример 2 был ранее использован в работе [7] и специально решен предлагаемым методом для сравнения.

5. Примеры

ПРИМЕР 1. Пусть задана НДС, состоящая из двух этапов, обозначенных номерами 0 и 1. На нулевом этапе управляемый процесс описывается одним уравнением:

$$x^d(t+1) = x^d(t) + \sin(u_1^d(t)), \quad x^d(0) = 1, \quad t = 0, 1, 2, 3.$$

Задан промежуточный функционал этапа:

$$I^0 = (u_1^d(t) - t)^2.$$

Следующий первый этап характеризуется другим уравнением и своим промежуточным функционалом:

$$x^d(t+1) = x^d(t)u_2(t), \quad t = 4, 5, 6, \quad I^1 = \frac{1}{2}((x^d(t))^2 + (u_2^d(t))^2).$$

Задан общий функционал задачи:

$$I = x^d(7) \rightarrow \min.$$

Нетрудно видеть, что $\mathbf{K} = 0, 1, 2$, $\mathbf{K}' = 1$. Поскольку роль связующей переменной на двух рассматриваемых этапах играет x^d , то в терминах этой переменной легко записать процесс верхнего уровня:

$$x(0) = x^d(0, 0), \quad x(1) = x^d(0, 4), \quad x(2) = x^d(1, 7), \quad x^d(1, 4) = x(1).$$

Так как множество \mathbf{K}' состоит из одного элемента, то для удобства расчетов, модель может быть дополнена третьим мгновенным этапом, не имеющим протяженности во времени и состоящим лишь в передаче информации об окончании второго этапа на верхний уровень. Тогда

$$x(3) = x(2) = x^d(1, 7), \quad \mathbf{K}' = 1, 2, \quad \theta(k) = x^d(k, t_F), \quad \xi(k) = x(k).$$

Заметим, что на обоих этапах процесс нижнего уровня не зависит от переменных состояния верхнего уровня, тогда

$$\lambda(0, t) = \lambda(1, t) = 0, \quad \Lambda(0, t) = \Lambda(1, t) = 0.$$

Запишем задачу и конструкции в терминах НДС. Имеем:

$$\begin{aligned} f^d(0, t) &= x^d(t) + \sin(u_1^d(t)), & f^d(1, t) &= x^d(t)u_2(t), \\ f_{x^d}^d(0, t) &= 1, & f_{x^d}^d(1, t) &= u_2(t), \\ f_{u_1^d}^d(0, t) &= \cos(u_1^d(t)), & f_{u_2^d}^d(1, t) &= x^d(t); \end{aligned}$$

$$H^d(0, t) = \psi^d(0, t+1) (x^d(t) + \sin(u_1^d(t))) - \\ - (u_1^d(t) - t)^2 - \frac{1}{2}(1-\alpha)|\Delta x^d(0, t)|^2,$$

$$H_{x^d}^d(0, t) = \psi^d(0, t+1), \quad H_{x^d x^d}^d(0, t) = \alpha - 1,$$

$$H_{u_1^d}^d(0, t) = \psi^d(0, t+1) \cos(u_1^d(t)) + 2(t - u_1^d(t)),$$

$$H_{u_1^d u_1^d}^d(0, t) = -\psi^d(0, t+1) \sin(u_1^d(t)) - 2, \quad H_{x^d u_1^d}^d(0, t) = 0;$$

$$H^d(1, t) = \psi^d(1, t+1)x^d(t)u_2^d(t) - \frac{1}{2}((x^d(t))^2 + (u_2^d(t))^2) - \frac{1}{2}(1-\alpha)|\Delta x^d(1, t)|^2,$$

$$H_{x^d}^d(1, t) = \psi^d(1, t+1)u_2^d(t) - x^d(t), \quad H_{x^d x^d}^d(1, t) = \alpha - 2,$$

$$H_{u_2^d}^d(1, t) = -2\psi^d(1, t+1)(t - u_2^d(t)) - 1,$$

$$H_{u_2^d u_2^d}^d(1, t) = -1, \quad H_{x^d u_2^d}^d(1, t) = \psi^d(1, t+1).$$

Управляющие воздействия являются результатом решения следующих уравнений:

$$\psi(2) = \psi(3) = -\alpha, \quad \psi(1) = \psi(0) = 0,$$

$$\psi^d(1, 7) = \psi(2)u_2^d(6) - x^d(1, 7), \quad \psi^d(0, 4) = \psi(1),$$

$$\psi^d(1, t) = \psi^d(1, t+1)u_2^d(t) - x^d(t) + (x^d(t)u_2^d(t)\sigma^d(1, t+1) + \\ + \psi^d(1, t+1)) \times (2\psi^d(1, t+1)(t - u_2^d(t)) + 1),$$

$$\psi^d(0, t) = \psi^d(0, t+1) - (\cos(u_1^d(t))\sigma^d(0, t+1) \times \\ \times (\psi^d(0, t+1) \cos(u_1^d(t)) + 2(t - u_1^d(t)))) \times \\ \times (\psi^d(0, t+1) \sin(u_1^d(t)) + 2)^{-1};$$

$$\sigma(2) = 1 - \alpha, \quad \sigma(1) = \sigma(2) + \sigma^d(1, 7) = -\alpha,$$

$$\sigma^d(1, 7) = \sigma(2) + \alpha - 2 = -1, \quad \sigma^d(0, 4) = \sigma(1) + \alpha - 1 = -1,$$

$$\sigma^d(1, t) = (u_2^d(t))^2 \sigma^d(1, t+1) + \alpha - 2 - \\ - ((u_2^d(t)\sigma^d(1, t+1)x^d(t) + \psi^d(1, t+1))^2 \times \\ \times ((x^d(t))^2 \sigma^d(1, t+1) - 1)^{-1},$$

$$\sigma^d(0, t) = \sigma^d(0, t+1) + \alpha - 1 - (\sigma^d(0, t+1) \cos(u_1^d(t)))^2 \times \\ \times (\sigma^d(0, t+1) \cos^2(u_1^d(t)) - \psi^d(0, t+1) \sin(u_1^d(t)) - 2)^{-1};$$

$$\Delta u^d(0, t) = -(\psi^d(0, t+1) \cos(u_1^d(t)) + 2(t - u_1^d(t)) + \\ + \sigma^d(0, t+1) \cos(u_1^d(t))\Delta x^d(0, t)) \times$$

$$\begin{aligned} & \times (\sigma^d(0, t+1) \cos^2(u_1^d(t)) - \psi^d(0, t+1) \sin(u_1^d(t)) - 2)^{-1}, \\ \Delta u^d(1, t) = & (2\psi^d(1, t+1)(t - u_2^d(t)) + 1 - (x^d(t)u_2^d(t)\sigma^d(1, t+1) + \\ & + \psi^d(1, t+1))\Delta x^d(1, t)) \times ((x^d(t))^2\sigma^d(1, t+1) - 1)^{-1}. \end{aligned}$$

Решение получено за 2 итерации. Изменение функционала по итерациям представлено в таблице 1.

ТАБЛИЦА 1. Результаты расчетов примера 1

Итерация j	α	$I = x^d(7) \rightarrow \min$	$ I_j - I_{j-1} $
0		4.366	
1	0.2	4.024	0.342
2	0.6	3.947	0.077

При расчетах выбор значения α осуществлялся для обеспечения минимума функционала и находился перебором в диапазоне от 0 до 1 с шагом 0.1. Графики переменных состояния и управления приведены на рисунках 1а и 1б.

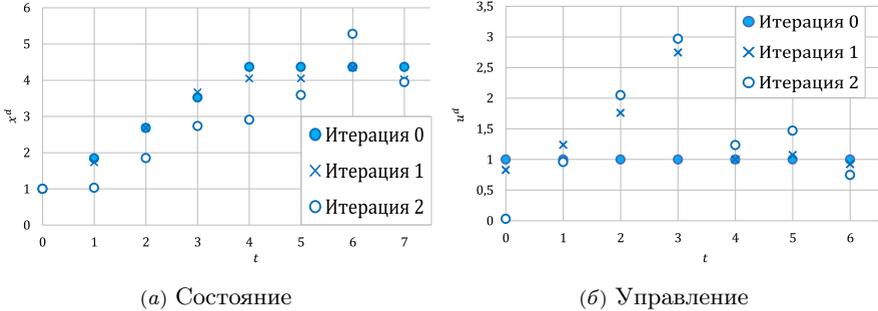


РИСУНОК 1. Результаты расчетов примера 1

ПРИМЕР 2. Двухэтапная задача:

1-й этап:

$$\begin{aligned} x^d(t+1) &= -2x^d(t) + (u_1^d(t))^2, \quad x^d(0) = 1, \quad t = 0, 1, 2, 3, \\ I^0 &= \frac{1}{2}(x^d(t))^2 + \frac{1}{3}(u_1^d(t))^3; \end{aligned}$$

2-й этап:

$$\begin{aligned} x^d(t+1) &= (t - u_2^d(t))^2, \quad t = 4, 5, 6, \quad I^1 = \frac{1}{2}(x^d(t))^2 + u_2^d(t); \\ I &= x^d(7) \rightarrow \min. \end{aligned}$$

Как и в предыдущем примере, представим эту систему в виде НДС. Имеем $\mathbf{K} = 0, 1, 2$, $\mathbf{K}' = 1$. Поскольку роль связующей переменной на двух рассматриваемых этапах играет x^d , то в терминах этой переменной легко записать процесс верхнего уровня:

$$x(0) = x^d(0, 0), \quad x(1) = x^d(0, 4), \quad x(2) = x^d(1, 7), \quad x^d(1, 4) = x(1).$$

Так как множество \mathbf{K}' состоит из одного элемента, то для удобства расчетов, модель может быть дополнена третьим мгновенным этапом, не имеющим протяженности во времени и состоящим лишь в передаче информации об окончании второго этапа на верхний уровень. Тогда

$$x(3) = x(2) = x^d(1, 7), \quad \mathbf{K}' = 1, 2, \quad \theta(k) = x^d(k, t_F), \quad \xi(k) = x(k).$$

На обоих этапах процесс нижнего уровня также не зависит от переменных состояния верхнего уровня, тогда

$$\lambda(0, t) = \lambda(1, t) = 0, \quad \Lambda(0, t) = \Lambda(1, t) = 0.$$

Имеем:

$$\begin{aligned} f^d(0, t) &= -2x^d(t) + (u_1^d(t))^2, & f^d(1, t) &= (t - u_2^d(t))^2, \\ f_{x^d}^d(0, t) &= -2, & f_{x^d}^d(1, t) &= 0, \\ f_{u_1^d}^d(0, t) &= 2u_1^d(t), & f_{u_2^d}^d(1, t) &= -2(t - u_2^d(t)); \end{aligned}$$

$$\begin{aligned} H^d(0, t) &= \psi^d(0, t+1)(-2x^d(t) + (u_1^d(t))^2) - \frac{1}{2}(x^d(t))^2 - \\ &\quad - \frac{1}{3}(u_1^d)^3 - \frac{1}{2}(1-\alpha)|\Delta x^d(0)|^2, \end{aligned}$$

$$\begin{aligned} H_{x^d}^d(0, t) &= -2\psi^d(0, t+1) - x^d(t), & H_{x^d x^d}^d(0, t) &= \alpha - 2, \\ H_{u_1^d}^d(0, t) &= 2\psi^d(0, t+1)u_1^d(t) - (u_1^d(t))^2, \\ H_{u_1^d u_1^d}^d(0, t) &= 2\psi^d(0, t+1) - 2u_1^d(t), & H_{x^d u_1^d}^d(0, t) &= 0; \end{aligned}$$

$$H^d(1, t) = \psi^d(1, t+1)(t - u_2^d)^2 - \frac{1}{2}(x^d)^2 - u_2^d(t) - \frac{1}{2}(1-\alpha)|\Delta x^d(1)|^2,$$

$$\begin{aligned} H_{x^d}^d(1, t) &= -x^d(t), & H_{x^d x^d}^d(1, t) &= \alpha - 2, \\ H_{u_2^d}^d(1, t) &= -2\psi^d(1, t+1)(t - u_2^d(t)) - 1, \\ H_{u_2^d u_2^d}^d(1, t) &= 2\psi^d(1, t+1), & H_{x^d u_2^d}^d(1, t) &= 0; \end{aligned}$$

$$\psi(2) = \psi(3) = -\alpha, \quad \psi(1) = \psi(0) = 0,$$

$$\psi^d(1, 7) = -x^d(1, 7), \quad \psi^d(0, 4) = -2\psi(1) - x^d(0, 4),$$

$$\psi^d(1, t) = -x^d(t),$$

$$\begin{aligned} \psi^d(0, t) &= -2\psi^d(0, t+1) - x^d(t) - \\ &\quad - (2(u_1^d(t))^2 \sigma^d(0, t+1)(2\psi^d(0, t+1) - u_1^d(t))) \times \\ &\quad \times (\psi^d(0, t+1) - u_1^d(t))^{-1}; \end{aligned}$$

$$\sigma(2) = 1 - \alpha, \quad \sigma(1) = \sigma(2) + \sigma^d(1, 7) = -\alpha,$$

$$\sigma^d(1, 7) = \sigma(2) + \alpha - 2 = -1, \quad \sigma^d(0, 4) = \sigma(1) + \alpha - 2 = -2,$$

$$\sigma^d(1, t) = \alpha - 2,$$

$$\begin{aligned} \sigma^d(0, t) &= 4\sigma^d(0, t+1) + \alpha - 2 - 8(u_1^d(t)\sigma^d(0, t+1))^2 \times \\ &\quad \times (2(u_1^d(t))^2 \sigma^d(0, t+1) + \psi^d(0, t+1) - u_1^d(t))^{-1}; \end{aligned}$$

$$\begin{aligned} \Delta u^d(0, t) &= -u_1^d(t) (2\psi^d(0, t+1) - u_1^d(t) - 4\sigma^d(0, t+1)\Delta x^d(0, t)) \times \\ &\quad \times (4(u_1^d(t))^2 \sigma^d(0, t+1) + 2\psi^d(0, t+1) - 2u_1^d(t))^{-1}, \end{aligned}$$

$$\begin{aligned} \Delta u^d(1, t) &= (2\psi^d(1, t+1)(t - u_2^d(t)) + 1) \times \\ &\quad \times (4\sigma^d(1, t+1)(t - u_2^d(t))^2 + 2\psi^d(1, t+1))^{-1}. \end{aligned}$$

Решение получено за 8 итераций. Изменение функционала по итерациям представлено в таблице 2.

ТАБЛИЦА 2. Результаты расчетов примера 2

Итерация j	α	$I = x^d(7) \rightarrow \min$	$ I_j - I_{j-1} $
0		36	
1	0.5	23.702	12.298
2	0.5	13.197	10.505
3	0.5	6.379	6.818
4	0.4	2.199	4.180
5	0.4	0.506	1.693
6	0.4	0.195	0.311
7	0.3	0.117	0.078
8	0.3	0.109	0.008

Графики переменных состояния и управления приведены на рисунках 2а и 2б.

Отметим, что этот пример имеет более сложную структуру. Его решение другим методом приведено в работе [7]. Варьирование параметра альфа позволило получить меньшее значение функционала по сравнению

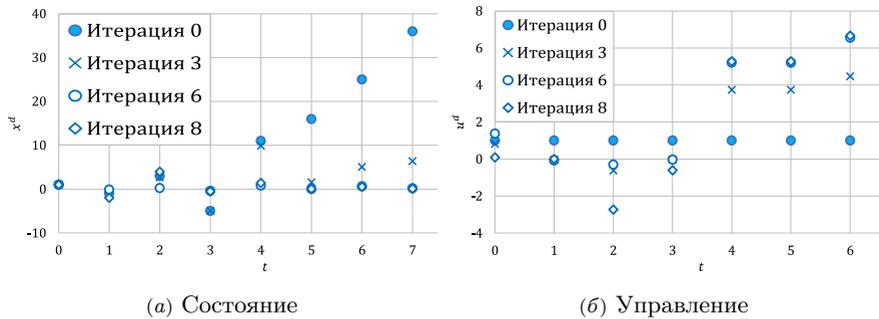


РИСУНОК 2. Результаты расчетов примера 2

с результатом работы [7].

Заключение

Таким образом, решенные примеры подтверждают работоспособность построенного метода, который имеет более сложную структуру по сравнению с методом из работы [7] за счет необходимости решения матричных уравнений Риккати для сопряженных переменных. При этом количество итераций незначительно увеличивается, но позволяет получить меньшее значение функционала, чем в [7].

Также дан ответ на вопрос, поставленный в начале статьи: понятие сильного относительного минимума можно использовать для построения метода улучшения для НДС.

Список использованных источников

- [1] Расина И. В. *Иерархические модели управления системами неоднородной структуры*. – М.: Физматлит. – 2014. – ISBN 978-5-94052-238-6. – 160 с. [↑](#)²⁴
- [2] Гурман В. И. *Принцип расширения в задачах управления*. – М.: Наука. – 1985. – 288 с. [↑](#)^{24, 27}
- [3] Rasina I., Danilenko O. *Second-order improvement method for discrete-continuous systems with intermediate criteria*, 17th IFAC Workshop on Control Applications of Optimization (Yekaterinburg, Russia, October 15–19, 2018) // IFAC-Papers Online. – 2018. – Pp. 184–188. [doi](#) [↑](#)²⁴
- [4] Расина И. В., Гусева И. С. *Метод улучшения управления для неоднородных дискретных систем с промежуточными критериями* // Программные системы: теория и приложения. – 2018. – Т. 9. – № 2(37). – С. 23–28. [doi](#) [URL](#) [↑](#)^{25, 26, 27}

- [5] Гурман В. И. *Абстрактные задачи оптимизации и улучшения* // Программные системы: теория и приложения.– 2011.– Т. 2.– № 5(9).– С. 21–29. [URL](#) ↑27
- [6] Гурман В. И., Расина И. В. *О практических приложениях достаточных условий сильного относительного минимума* // Автоматика и телемеханика.– 1979.– № 10.– С. 12–18.   ↑27
- [7] Расина И. В., Блинов А. О. *Метод минимаксного улучшения для неоднородных дискретных систем* // Программные системы: теория и приложения.– 2023.– Т. 14.– № 4(59).– С. 47–66. [URL](#)  ↑32, 36, 37

Поступила в редакцию 04.04.2025;
 одобрена после рецензирования 27.05.2025;
 принята к публикации 09.07.2025;
 опубликована онлайн 19.07.2025.

Рекомендовал к публикации

д.ф.-м.н. М. Г. Дмитриев

Информация об авторах:



Ирина Викторовна Расина

д.ф.-м.н., гл. научн. сотр. Исследовательского центра медицинской информатики Института программных систем имени А. К. Айламазяна РАН. Специалист в области моделирования и управления гибридными системами, автор и соавтор более 130 статей и 5 монографий

 0000-0001-8939-2968

e-mail:



Ирина Сергеевна Гусева

к.ф.-м.н., ст.преп. Бурятского государственного университета имени Доржи Банзарова, РФ, респ. Бурятия, г. Улан-Удэ; автор и соавтор более 20 публикаций; область интересов – оптимальное управление, численные методы

 0000-0001-8720-3676

e-mail:

Авторы внесли равный вклад в подготовку публикации.

Декларация об отсутствии личной заинтересованности: благополучие авторов не зависит от результатов исследования.



Strong control improvement method for non-homogeneous discrete systems

Irina Viktorovna **Rasina**¹, Irina Sergeevna **Guseva**²

¹ Ailamazyan Program Systems Institute of RAS, Ves'kovo, Russia

² Buryat State University, Ulan-Ude, Russia

 irinarasina@gmail.com

Abstract. The class of non-homogeneous discrete systems (NDS) with intermediate criterions is considered. These systems are two-level and are prevalent in practice. They can be also obtained via discretization of continuous systems in the process of solving optimization problems using iterative methods. For this class of systems a strong improvement method of the second order is constructed based on the analogue of Krotov type sufficient optimality conditions.

The authors of the article question the assertion that for classical discrete control systems, as well as for heterogeneous ones, there is no sense in introducing the concept of a strong relative minimum. Therefore, when constructing an improvement method, we put forward the requirement of proximity of neighboring approximations from the class of admissible only by the process states at both levels. The resulting method contains a vector-matrix two-level system for conjugate variables. The increment of controls at each level linearly depends on the corresponding states, which allows finding a solution in the form of approximate linear synthesis of optimal control.

The method was tested on two illustrative examples, which showed its efficiency. The application of the developed method to a more complex example allowed us to obtain a smaller value of the functional than that found earlier by a similar in structure minimax improvement method. (*In Russian*).

Key words and phrases: non-homogeneous discrete systems, intermediate criterions, optimal control

2020 *Mathematics Subject Classification:* 49M99; 49K99

For citation: Irina V. Rasina, Irina S. Guseva. *Strong control improvement method for non-homogeneous discrete systems*. Program Systems: Theory and Applications, 2025, **16**:3(66), pp. 23–40. (*In Russ.*). https://psta.psisras.ru/read/psta2025_3_23-40.pdf

References

- [1] I. V. Rasina. *Hierarchical control models of heterogeneous structure systems*, Fizmatlit, M., 2014, ISBN 978-5-94052-238-6, 160 pp. (in Russian).
- [2] V. I. Gurman. *Extension Principle in Control Problems*, Nauka, M., 1985, 228 pp. (in Russian).
- [3] I. Rasina, O. Danilenko. “Second-order improvement method for discrete-continuous systems with intermediate criteria”, 17th IFAC Workshop on Control Applications of Optimization (Yekaterinburg, Russia, October 15–19, 2018), *IFAC-Papers Online*, 2018, pp. 184–188. 
- [4] I. V. Rasina, I. S. Guseva. “Control improvement method for non-homogeneous discrete systems with intermediate criterions”, *Program Systems: Theory and Applications*, **9**:2(37) (2018), pp. 23–38 (in Russian).  
- [5] V. I. Gurman. “Abstract problems of optimization and improvement”, *Program Systems: Theory and Applications*, **2**:5(9) (2011), pp. 21–29 (in Russian). 
- [6] V. I. Gurman, I. V. Rasina. “On practical applications of conditions sufficient for a strong relative minimum”, *Autom. Remote Control*, **40**:10 (1980), pp. 1410–1415. 
- [7] I. V. Rasina, A. O. Blinov. “Minimax improvement method for inhomogeneous discrete systems”, *Program Systems: Theory and Applications*, **14**:4(59) (2023), pp. 47–66 (in Russian). 

УДК 004.724.2 + 004.272.43

 10.25209/2079-3316-2025-16-3-41-68

Быстродействие и отказоустойчивость идеальной системной сети через дополнительную параллельность

Виктор Сергеевич **Подлазов** 

Институт проблем управления имени В. А. Трапезникова РАН, Москва, Россия

 podlazov@ipu.ru

Аннотация. Рассмотрены методы повышения быстродействия системных сетей посредством двоякого повышения их параллельности – использования копий и параллельных каналов в их совокупности. Предложены методы построения неблокируемых системных сетей с прямыми каналами произвольной размерности. Рассмотрены методы построения неблокируемых сетей с заданным числом параллельных каналов как сетей со структурой квазиполных графов. Предложен метод построения системных сетей любого размера с заданным числом параллельных каналов посредством использования соответствующего числа их копий.

Ключевые слова и фразы: полный коммутатор, прямые каналы, мультиплексоры и демультимплексоры, многокаскадный коммутатор, бесконфликтная маршрутизация, неблокируемый коммутатор, статическая самомаршрутизация, квазиполный граф с заданным числом параллельных каналов, изоморфизм квазиполного графа и симметричной блок-схемы (block-design)

Для цитирования: Подлазов В. С. *Быстродействие и отказоустойчивость идеальной системной сети через дополнительную параллельность* // Программные системы: теория и приложения. 2025. **Т. 16.** № 3(66). С. 41–68.
https://psta.psir.ru/read/psta2025_3_41-68.pdf

1. Введение

В статье рассматриваются возможности повышения характеристик системных сетей (распределенных коммутаторов) многопроцессорных вычислительных комплексов. Решается задача многократного повышения пропускной способности системных сетей и частичного резервирования их абонентов и каналов посредством копирования сетей и использования разветвителей каналов для подключения абонентов к разным копиям.

Задача решается посредством повышения сетевого быстродействия абонентов в результате параллельной передачи разных частей пакетов по разным каналам. Задача решается для системных сетей любой структуры (идеальные неблокируемые сети, традиционные многомерные гиперкубы, многомерные торы и т.п.) и размера (числа абонентов). При этом накладными затратами являются повышенная сложность вследствие использования копий сети и многократное использование адреса пакета в каждой его части, передаваемой независимо.

Решение задачи основывается на методах построения идеальных системных сетей с разными свойствами. Новизна решения состоит в сопряжении этих сетей для решения задачи повышения быстродействия системных сетей.

Идеальной системной сетью (ИСС) [1] считается сеть, которая обеспечивает соединение любых абонентов по прямым каналам (без промежуточной буферизации) при параллельной передаче пакетов данных от всех абонентов. Кроме того системная сеть считается идеальной, если она является неблокируемой (бесконфликтной) и самомаршрутизируемой на произвольных перестановках пакетов между абонентами.

ИСС рассматриваются в двух вариантах [2–5], с разными свойствами. В первом варианте это сети с топологией квазиполного графа [2, 3], изоморфного симметричной блок-схеме (block-design), изучаемой в комбинаторике [6]. Квазиполный граф – это двудольный граф, одну долю которого (хорду) составляют узлы, задающие полные m -портовые коммутаторы $m \times m$, а другую долю – узлы, являющиеся m -портовыми абонентами. Любой узел в каждой доле связан ребрами (дуплексными каналами) с m разными узлами в другой доле. Кроме того, любые два абонента, связаны точно $\sigma \leq m$ путями длины 2 через разные узлы другой доли (коммутаторы). Число узлов N в каждой доле задается

через параметры m и σ соотношением $N = m(m - 1)/\sigma + 1$ и является максимально возможным.

Квазиполные графы (и симметричные блок-схемы) существуют не для всех значений параметров m и σ . Одни не существуют по определению (нецелое значение N), другие по теории [6], а третьи – просто не построены. Построение блок-схем это NP-сложная по m/σ задача, которая при $m > 12$ и $\sigma \gg 2$ практически еще не решена (блок-схемы не построены) [6, 7].

Если системная сеть имеет топологию квазиполного графа, то она является ИСС [2, 3] с σ разными путями между любыми двумя абонентами. Она также может задавать и распределенный самомаршрутизируемый полный коммутатор $N \times N$. В этом случае каждый m -портовый абонент заменяется на однопортового абонента, подключенного через двусторонний разветвитель каналов (демультиплексор $1 \times m$ от каждого абонента и мультиплексор $m \times 1$ к каждому абоненту). Такой коммутатор можно инвариантно масштабировать по числу абонентов посредством увеличения числа коммутаторов и добавления каскада разветвителей каналов. Инвариантность проявляется в том, что сохраняются прямые каналы между абонентами, неблокируемость сети, самомаршрутизируемость по сети и число σ разных каналов между абонентами. К сожалению, вследствие некратности N и m масштабирование числа абонентов происходит достаточно медленно с увеличением числа каскадов разветвителей особенно при больших σ .

Особо отметим, что на основе любой системной сети с m абонентами можно построить расширенную системную сеть с $N = m(m - 1)/\sigma + 1$ абонентами и σ разными путями между ними. Для этого достаточно в сети с топологией квазиполного графа использовать в хорде копии исходной сети вместо коммутаторов. При этом любой путь проходит только через одну копию заданной сети с ее внутренними задержками. В такой расширенной сети возможно теряется неблокируемость и увеличиваются задержки передачи, но сохраняется возможность увеличения ее пропускной способности посредством параллельной передачи пакетов по разным путям. Расширенную сеть также можно масштабировать описанным способом по числу абонентов, сохраняя ее полную связность и число разных каналов между любыми абонентами.

Существует комбинаторный метод приближенного построения квазиполных графов для любых значений параметров m и σ . Это построение

1-расширенных квазиполных графов, в которых некоторые абоненты связаны $\sigma + 1$ ребрами, а остальные — точно σ ребрами, и число абонентов N^* может быть на несколько единиц меньше N [8]. Такие расширенные квазиполные графы построены ниже для ряда $m < 14$ и $\sigma < 10$.

Использование ИСС с топологией квазиполного графа дает возможность многократно повышать скорость передачи пакетов по сети. Для этого достаточно разбивать пакеты на σ равных малых пакетов и параллельно передавать их по σ разным путям между абонентами. Накладными затратами при этом является необходимость передачи адреса приемника в каждом малом пакете и необходимость одновременной передачи малых пакетов от каждого абонента и приема их к нему. Кроме того, такой ИСС обеспечивает $(\sigma - 1)$ -канальную отказоустойчивость сети и даже канальную живучесть с понижением скорости передачи пакетов при отказах каналов.

Главным недостатком ИСС с топологией квазиполного графа является малое число его абонентов и слабая наращиваемость при каскадном масштабировании. В данной работе решается задача преодоления этого недостатка, т.е. построения ИСС с любым числом абонентов при сохранении возможности иметь σ разных путей между абонентами.

Оба недостатка преодолеваются посредством использования ИСС с топологией квазиполного орграфа [4, 5], который не имеет параллельных каналов. Эти ИСС изоморфны сетям со структурой двумерного гиперкуба. Они существуют при любых значениях m и только для $\sigma = 1$, и для них число абонентов задается соотношением $N = m^2$.

Эти ИСС образуют распределенные самомаршрутизируемые полные коммутаторы $N \times N$. Для этого каждый m -портовый абонент заменяется на однопортового абонента, подключенного через двусторонние разветвители каналов $1 \times m$ (мультиплексоры $m \times 1$, демультиплексоры $1 \times m$).

Маршрутизаторы с такой топологией (*YARC* и *ROSSETA* с внутренней буферизацией пакетов) с 64 и 48 портами используются в разных системных сетях [9–11]. Полные коммутаторы с аналогичными параметрами позволяют строить ИСС с сотнями и тысячами абонентов. Так коммутаторы 64×64 позволяют создать ИСС с $N = 4096$ абонентами. Такая ИСС будет содержать N коммутаторов 64×64 , каскад из N демультиплексоров 1×64 и каскад из N мультиплексоров 64×1 .

Коммутаторы с топологией квазиполных орграфов допускают простое масштабирование числа абонентов без утраты свойства неблокируемости и самомаршрутизируемости на произвольных перестановках.

Оно осуществляется посредством каскадирования с использованием дополнительных слоев разветвителей каналов $1 \times m$. При этом возможно как внутреннее масштабирование с ростом числа абонентов в m раз, так и внешнее масштабирование с ростом числа абонентов в квадрат раз при каждом каскадировании.

В данной статье решается задача построения комбинированных ИСС с большим числом абонентов $N \gg m^2$ и числом параллельных путей между абонентами в диапазоне $3 \leq \sigma \leq 10$ посредством разработки метода сопряжения ИСС с топологией квазиполных орграфов или любых системных сетей и ИСС с топологией квазиполных графов. Такие комбинированные ИСС обладают возможностью многократно повышать свою пропускную способность и сетевое быстродействие абонентов на основе исходной элементной базы.

В разделе 2 кратко представляются свойства ИСС с топологией квазиполного графа и строятся новые ИСС с большими значениями σ ($\sigma = 6$ и $\sigma = 10$). В разделе 2 представляется метод расширения любых ИСС посредством внутреннего масштабирования с добавлением еще одного каскада разветвителей канала. При этом ИСС расширяется как распределенный коммутатор с подключением однопортовых абонентов через каскад разветвителей каналов. Однако на заключительном шаге расширения крайний каскад собирается из многопортовых абонентов. Расширенная сеть состоит из набора копий исходной сети и приобретает возможность иметь σ разных каналов между любыми абонентами расширенной сети. Исследуются свойства этого метода при его каскадном применении.

Исследование показало, что при больших значениях σ это масштабирование эффективно только при однократном применении. В разделе 2 также показано, что метод внутреннего масштабирования применим не только для ИСС, но и для сетей с любой структурой (ИСС другой топологии, многомерных гиперкубов или торов). В этом случае данный метод дает возможность наделять группы сетей возможностью использовать разные каналы для повышения пропускной способности и канальной отказоустойчивости расширенной сети.

Для решения ранее поставленной задачи необходимо иметь возможность использовать системные сети любого размера (числа абонентов). В разделе 3 представляется метод построения ИСС с топологией квазиполного орграфа любого размера, имеющих $\sigma = 1$.

Сам метод основывается на начальном представлении ИСС как двудольного орграфа, который изоморфен двумерному гиперкубу. Он имеет хордовый каскад коммутаторов $m \times m$, каскад разветвителей каналов $1 \times m$ и объединяет $N = m^2$ абонентов. Такая ИСС многократно масштабируется как распределенный коммутатор посредством увеличения числа исходных малых коммутаторов и числа каскадов разветвителей каналов. При этом могут использоваться два метода масштабирования – внутренний и внешний.

Первый метод аналогичен методу расширения ИСС с топологией квазиполного графа, но поскольку в данном случае число абонентов кратно степени разветвителя каналов, то число абонентов (и число хордовых коммутаторов) увеличивается в m раз при каждом добавлении каскада разветвителей.

Второй метод масштабирует число абонентов (и число хордовых коммутаторов) посредством масштабирования уже имеющейся ИСС как исходной, но с использованием каскадов разветвителей большого размера. Для сохранения элементной базы последние представляются как деревья исходных разветвителей. Такой метод внешнего масштабирования позволяет увеличивать число абонентов в квадрат раз при каждом применении каскада разветвителей все большего размера.

На базе ИСС с топологией квазиполных графов и орграфов в разделе 4 строятся и рассчитываются комбинированные ИСС, сочетающие возможности и свойства базовых ИСС. Сначала строятся ИСС с топологией квазиполных графов необходимого размера (числом абонентов). Затем они расширяются методом внутреннего масштабирования посредством использования необходимого числа их копий и добавления ИСС с m -портовыми абонентами ($m \geq \sigma$), использующей топологию квазиполных графов с заданным числом σ разных каналов (конкретно с $\sigma = 6$ и $\sigma = 10$). Таким образом, строятся комбинированные ИСС заданного размера и заданного числа разных каналов между абонентами. Кроме того, комбинированные ИСС строятся с разным числом добавляемых абонентов, в частности, без изменения их числа ($m = \sigma$) или с их удвоением ($m \approx 2\sigma$).

Для повышения сетевого быстродействия абонентов, передаваемые пакеты разбиваются на σ равных частей. Для полной загрузки σ разных каналов каждый абонент должен иметь равное число внешних сдвиговых регистров, которые параллельно передают свои части пакетов в соответствующие каналы или принимают эти части из них. Передача по прямым каналам обеспечивает минимальный диаметр в один скачок, а наличие нескольких разных параллельных каналов позволяет сочетать отказоустойчивость и многократно повышенную пропускную способность.

Эти свойства достигаются посредством использования многокаскадной структуры ИСС. Базовый хордовый каскад содержит полные коммутаторы, промежуточные каскады – разветвители каналов, а внешний каскад содержит многопортовые абоненты, обеспечивающие наличие равное число разных каналов между ними. Такая ИСС имеет высокую схемную сложность [4], которая при измерении в числе точек коммутации оценивается как $O(N^2)$. Обычно эта сложность распределена между N узлами сети.

Совсем не так обстоит дело в известных системных сетях меньшей сложности, поскольку задача построения неблокируемых отказоустойчивых системных сетей до настоящего времени не имеет полного решения.

Системная сеть является неблокируемой, если в ней для любой перестановки пакетов можно проложить бесконфликтные пути от источников к приемникам. Системная сеть является самомаршрутизируемой, если бесконфликтные пути можно проложить локально по узлам сети без их взаимодействия только на основе маршрутной информации в пакетах. Наконец, самомаршрутизация является статической, если любой источник может самостоятельно наметить бесконфликтные пути к своему приемнику.

Существование неблокируемых сетей было доказано еще Клозом [7, 8]. Однако пока еще не построено самомаршрутизируемых неблокируемых сетей Клоза. Поэтому сети Клоза используются только в более простых самомаршрутизируемых вариантах, которые не являются неблокируемыми сетями, а только перестраиваемыми сетями. Неблокируемыми являются 64–48 портовые коммутаторы YARK и ROSETTA, используемые как строительные блоки в ряде сетей разной структуры: перестраиваемой сети Клоза [9], трехмерного гиперкуба [10, 11], иерархии полных орграфов [12]. При этом сами эти сети не являются неблокируемыми.

Сети со структурой толстого дерева являются перестраиваемыми сетями Клоза [9], в которых бесконфликтная передача осуществляется только по заранее составленным расписаниям для конкретных перестановок пакетов. Для произвольных перестановок эти сети оказываются блокируемыми, в которых произвольная перестановка полностью осуществляется за несколько скачков между узлами сети. Максимальное число таких скачков задает диаметр сети. В перестраиваемых сетях Клоза диаметр равен числу каскадов сети.

В литературе также широко представлены системные сети со структурой многомерного тора [13–15], использующие аналогичные небольшие коммутаторы в своих узлах. Эти сети для произвольных перестановок вообще не имеют возможности передавать пакеты по прямым каналам. В них произвольные перестановки осуществляются только за несколько скачков между узлами сети. Многомерные торы являются самыми простыми, но и самыми медленными сетями вследствие их больших диаметров, которые измеряются десятками скачков.

Наоборот сети со структурой иерархии полных или квазиполных орграфов [11, 12] имеют самый маленький диаметр в 3 скачка.

Сети со структурой обобщенного гиперкуба не являются даже перестраиваемыми сетями [16–18]. Их можно сделать таковыми посредством увеличения числа каналов в некоторых измерениях. Обобщенные гиперкубы имеют диаметр равный числу измерений или на 1 меньше в расширенном гиперкубе [17].

Большинство известных сетей не обладает канальной отказоустойчивостью без применения дополнительных мер. Например, гиперкуб обладает одноканальной отказоустойчивостью при удвоении числа каналов в каждом измерении [18].

Аналогичным свойством обладает 6-мерный тор [15], в котором только три измерения являются междуузловыми, а остальные измерения реализуются внутри узлов.

В сетях со структурой иерархии полных или квазиполных орграфов [11, 12] отказ канала приводит к увеличению диаметра сети с 3 до 5 скачков, т.е. к частичному уменьшению сетевого быстродействия абонентов и пропускной способности сети в целом.

Отказ каналов в сетях Клоза приводит к снижению пропускной способности сети, но не нарушает полной связности сети, т.к. в них

имеется несколько каналов между абонентами, которые, однако, не могут использоваться абонентами параллельно.

В разделе 2 рассматриваются различные аспекты построения ИСС с топологией квазиполных графов. В разделе 3 рассматриваются различные аспекты построения ИСС с топологией квазиполных орграфов. Эти разделы содержат краткое изложение работ [2–8]. В разделе 4 рассматриваются варианты построения ИСС с комбинированной топологией и характеристики такой ИСС.

2. ИСС на основе топологии квазиполного графа

Квазиполный граф – это однородный двудольный граф, каждую долю которого составляют N узлов степени m . Значение m выбирается минимальным, при котором любые два узла в одной доле связаны $\sigma \leq m$ прямыми путями длины два через разные узлы в другой доле. Если такой граф существует, то его параметры связаны соотношением $N = m(m - 1)/\sigma + 1$.

При схемной реализации узлы одной доли задаются как абоненты (процессоры) с m дуплексными портами, а узлы другой доли – как полные коммутаторы $m \times m$ с m дуплексными портами. Такую сеть мы называем простейшей и обозначаем ПС(N, m, σ). Она представлена на рисунке 1 при $m = 4$ и $\sigma = 2$.

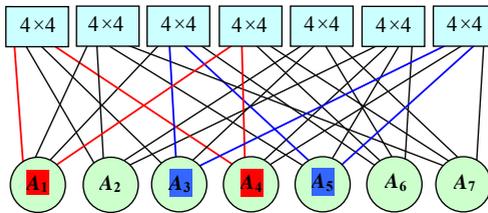


Рисунок 1. ИСС как ПС(7, 4, 2).

Сети ПС(N, m, σ) изоморфны симметричным блок-схемам, исследуемым в комбинаторике. Матрица инцидентности квазиполного графа является симметричной блок-схемой $B(N, m, \sigma)$, которая при $m = 4$ и $\sigma = 2$ задается в таблице 1. Теория блок-схем задает их существование в разрешимом виде, при котором они состоят из двух меньших блок-схем.

Таблица 1. Разрешимая блок-схема $B(7, 4, 2)$

Блоки 4×4	Абоненты			
1	1	2	3	4
2	1	2	5	7
3	1	3	5	6
4	1	4	6	7
5	2	3	6	7
6	2	4	5	6
7	3	4	5	7

Для построения блок-схем используются комбинаторные методы, которые представляют их в циклическом виде. Циклические блок-схемы задаются по столбцам циклическими последовательностями $(1, 2, \dots, N)$, размещенных в i -х столбцах и начинающихся в строках $a_i (i = 1, 2, \dots, m)$. При этом набор (a_1, a_2, \dots, a_m) полностью задает циклическую блок-схему. В частности, блок-схема в таблице 2 задается набором $(1, 3, 4, 5)$. Любые

Таблица 2. Циклическая блок-схема $B(7, 4, 2)$

Блоки 4×4	Абоненты			
1	1	6	5	4
2	2	7	6	5
3	3	1	7	6
4	4	2	1	7
5	5	3	2	1
6	6	4	3	2
7	7	5	4	3

циклические блок-схемы сводятся к разрешимым блок-схемам, но не наоборот, т.к. разрешимая блок-схема может не существовать в циклическом виде.

Для преодоления последнего недостатка было предложено использовать приближенные 1-расширенные циклические блок-схемы $B^*(N^*, m, \sigma^*)$, в которых числа абонентов N^* и разных путей σ^* между любыми двумя

абонентами подчиняется соотношениям $N^* \leq N$ и $\sigma^* \leq \sigma + 1$, а число N^* является максимальным для заданного значения m . Такие 1-расширенные блок-схемы могут быть построены комбинаторным способом для любых значения σ и m .

В данной работе это было сделано для разных значений m при $\sigma = 6$ и $\sigma = 10$, и полученные блок-схемы представлены в таблице 3. При

Таблица 3. Некоторые 1-расширенные циклические блок-схемы $B^*(N^*, m, 6)$ и $B^*(N^*, m, 10)$

$B^*(6, 6, 6)$	(1, 2, 3, 4, 5, 6)
$B^*(10, 8, 6)$	(1, 2, 3, 4, 5, 6, 7, 8)
$B^*(13, 9, 6)$	(1, 2, 3, 4, 5, 6, 8, 10, 11)
$B^*(15, 10, 6)$	(1, 2, 3, 4, 5, 6, 7, 9, 10, 12)
$B^*(10, 10, 10)$	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
$B^*(19, 11, 6)$	(1, 2, 3, 4, 5, 7, 8, 11, 13, 15, 16)
$B^*(22, 12, 6)$	(1, 2, 3, 4, 5, 6, 7, 9, 11, 14, 17, 18)
$B^*(14, 12, 10)$	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
$B^*(26, 13, 6)$	(1, 2, 3, 4, 5, 6, 8, 11, 14, 16, 18, 22, 23)
$B^*(16, 13, 10)$	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14)
$B^*(30, 14, 6)$	(1, 2, 3, 4, 5, 6, 8, 11, 13, 14, 19, 20, 24, 28)
$B^*(19, 14, 10)$	(1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 14, 15, 16)
$B^*(35, 15, 6)$	(1, 2, 3, 4, 5, 6, 8, 12, 15, 17, 21, 23, 28, 29, 33)
$B^*(25, 15, 10)$	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 15, 16)
$B^*(39, 16, 6)$	(1, 2, 3, 4, 5, 6, 8, 11, 14, 16, 20, 21, 25, 28, 32, 34)
$B^*(24, 16, 10)$	(1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 14, 17, 18)

этом некоторые 1-расширенные блок-схемы могут быть и простыми циклическими блок-схемами.

Очень часто m -портового абонента можно представить как однопортового абонента с разветвителем каналов $1 \times m$, в виде мультиплексора/демультиплексора во встречных каналах. В этом случае простейшая сеть ПС(N, m, σ) превращается в распределенный полный коммутатор $N \times N$,

обозначаемый как $\text{PK}_1(N_1, m_1, \sigma)$ с $N_1 = N$ и $m_1 = m$. На рисунке 2 он показан для $m = 4$ и $\sigma = 2$ и расположен выше интерфейса с абонентами, обозначенного пунктирной линией.

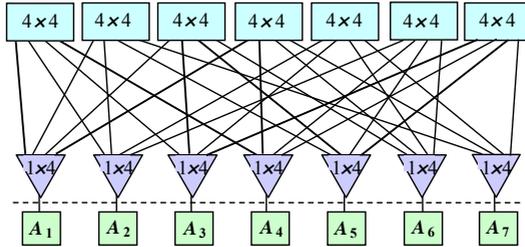


Рисунок 2. ИСС как $\text{PK}_1(7, 4, 2)$.

Заметим, что при построении $\text{PK}_1(N, m, \sigma)$ мы фактически решили задачу масштабирования малого полного коммутатора $m \times m$ до большого коммутатора $N \times N$, составленного из малых коммутаторов. При этом мы получили возможность иметь σ разных прямых путей между любыми абонентами в большом коммутаторе.

Последний коммутатор можно масштабировать и дальше двумя способами – внешним и внутренним каскадированием.

При внешнем каскадировании берется $m^2 = N_1$ и строится коммутатор $\text{PK}_2(N_2, m_2, \sigma)$ с топологией квазиполного графа с $N_2 = m_2(m_2 - 1)/\sigma + 1$. Например, при $m_1 = 4$ и $\sigma = 2$ имеем $m^2 = 7$ и $N_2 = 22$. Блок-схема $B(22, 7, 2)$ не существует по теории, но построена 1-расширенная блок-схема $B^*(21, 7, 2)$, задаваемая, например, набором $(1, 2, 3, 5, 9, 12, 17)$.

Однако масштабирование коммутатора $\text{PK}_2(N_2, m_2, \sigma)$ в коммутатор $\text{PK}_3(N_3, m_3, \sigma)$ с $m_3 = N_2$ уже практически невозможно, т.к. блок схема $B(N_3, m_3, \sigma)$ с $N_3 = m_3(m_3 - 1)/\sigma + 1$ скорее всего еще не построена ни в каком виде вследствие комбинаторных трудностей. В частности имеем $m_3 = 22$ и $N_3 = 232$.

При внутреннем каскадировании используются постоянная элементная база (коммутаторы $m \times m$ и разветвители каналов $1 \times m$), а масштабирование можно продолжать на любое число каскадов, но с более медленным ростом числа абонентов.

Внутреннее каскадирование состоит в построении расширенной простейшей сети, которая собирается из N_1 копий коммутатора $\text{PK}_1(N_1, m_1, \sigma)$.

Каждый из них разбивается на доли по m портов, и в каждой доле по всем коммутаторам строится отдельный составной коммутатор $PK_1(N_1, m_1, \sigma)$, но со сквозной нумерацией по всем ним. Эта нумерация является вторичной по отношению к внутренней нумерации в копиях исходного коммутатора и не зависит от нее.

В таблице 4 слева приводится пример таблицы инцидентности такой расширенной сети ($m_1 = 4$ и $\sigma = 2$) в предположении, что $N_1 = rm_1$.

Таблица 4. Таблица инцидентности для каскадирования $PK_1(7, 4, 2)$

ПК ₁	ППС(14, 4, 2)								ПК ₂ (11, 4, 2)							
	Порты	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7
1	1	2	3	4	8	9	10	11	1	2	3	4	8	9	10	
2	1	2	5	7	8	9	<u>12</u>	<u>14</u>	1	2	5	7	8	9	11	
3	1	3	5	6	8	10	<u>12</u>	<u>13</u>	1	3	5	6	8	10		
4	1	4	6	7	8	11	<u>13</u>	<u>14</u>	1	4	6	7	8	11		
5	2	3	6	7	9	10	<u>13</u>	<u>14</u>	2	3	6	7	9	10		
6	2	4	5	6	9	11	<u>12</u>	<u>13</u>	2	4	5	6	9	11		
7	3	4	5	7	10	11	<u>12</u>	<u>14</u>	3	4	5	7	10	11		

В этом случае число абонентов в расширенной сети оказывается $N_2 = rN_1$, и по построению не уменьшается число разных путей между абонентами, т.к. их число равно или σ или m_1 .

Однако в $PK_1(N_1, m_1, \sigma)$ число N_1 не кратно m_1 и расширенная сеть преобразуется в $PK_2(N_2, m_1, \sigma)$, в котором $N_2 < rN_1$. В таблице 4 справа приводится пример таблицы инцидентности $PK_2(N_2, m_1, \sigma)$. На рисунке 3 приводится схема $PK_2(N_2, m_1, \sigma)$, построенного по таблице 3.

Построение соединений осуществляется следующим образом. По номерам абонентов в таблице каскадирования коммутатора $PK_1(N_1, m_1, \sigma)$ находятся номера его копий через которые проходит соединение. Например, в таблице 4 абоненты 2 и 8 соединяются через коммутаторы 1 и 2. По порядковым номерам абонентов в таблице определяются номера портов соответствующих коммутаторов $PK_1(N_1, m_1, \sigma)$. Например, в таблице 4

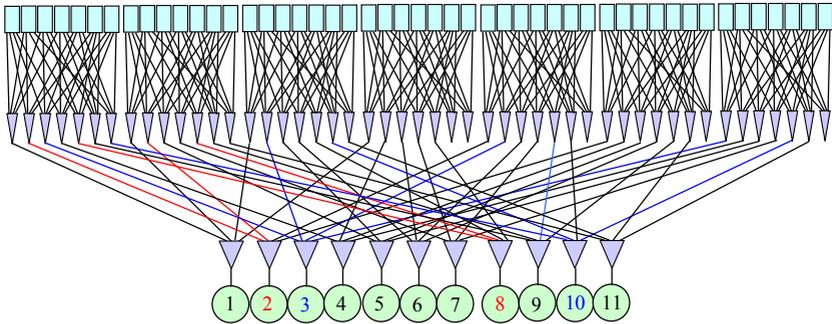


Рисунок 3. $PK_2(11, 4, 2)$, полученный посредством внутреннего расширения $PK_1(7, 4, 2)$.

порядковые номера абоненты 2 и 8 это 2 и 5, которые задают номера портов коммутаторов 1 и 2.

Коммутатор $PK_2(N_2, m_1, \sigma)$, задает ИСС с большим числом абонентов чем коммутатор $PK_1(N_1, m_1, \sigma)$, но не с меньшим числом разных путей между любыми двумя абонентами. Масштабирование ИСС можно продолжать и дальше посредством каскадирования коммутаторов $PK_i(N_i, m_1, \sigma)$. При этом для рассматриваемого примера образуется следующая последовательность коммутаторов $PK_3(18, 4, 2)$, $PK_4(31, 4, 2)$, $PK_5(52, 4, 2)$, $PK_6(91, 4, 2)$, $PK_7(152, 4, 2)$ и $PK_8(266, 4, 2)$, каждый из которых является ИСС с не меньше чем 2 разными путями между абонентами. Еще меньшее масштабирование числа абонентов происходит при каскадировании коммутатора $PK_1(5, 4, 3)$ при котором последовательность ИСС состоит из $PK_2(6, 4, 3)$, $PK_3(7, 4, 3)$, $PK_4(8, 4, 3)$ и $PK_5(10, 4, 3)$.

Фактически масштабирование ИСС с топологией квазиполного графа оказалось малоэффективным кроме первого шага построения коммутатора $PK_1(N_2, m_1, \sigma)$, при котором увеличилось как число абонентов, так и число разных каналов между ними. Но такое преобразование (инвариантное расширение) может быть применено к системной сети любой структуры.

Пусть у нас имеется исходная сеть $ИС(M)$ на M абонентов (рисунок 4 слева), к которой применяется процедура внутреннего расширения с использованием $N = m(m - 1)/\sigma + 1$ копий $ИС(M)$. При этом образуется расширенная сеть $РС(M^*)$ с $M^* = \lfloor M/m \rfloor N + \delta$ абонентами при $0 \leq \delta < N$, в которой имеется σ разных путей между любыми абонентами (рисунок 4 справа).

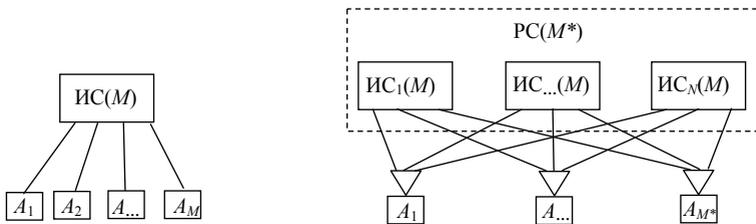


Рисунок 4. Инвариантное расширение системной сети

Рассмотренный подход уже использовался применительно к системной сети «Ангара» [12] для возможного повышения ряда ее характеристик [13, 14]. Ровно так же он может быть использован и для повышения ее быстродействия (раздел 4). В данной работе в качестве сетей $ИС(M)$ далее рассматриваются ИСС с топологией квазиполного орграфа с $\sigma = 1$, которые допускают любое масштабирование. Они кратко представляются в следующем разделе.

3. ИСС на основе топологии квазиполного орграфа

В данном разделе строятся неблокируемые самомаршрутизируемые распределенные коммутаторы на базе коммутаторов с топологией квазиполных орграфов. Для них возможна любая степень масштабирования числа каналов посредством внутреннего и внешнего каскадирования. Они базируются на двумерном гиперкубе, построенном из m -канальных коммутаторов $m \times m$. На рисунке 5 приводится пример двумерного гиперкуба.

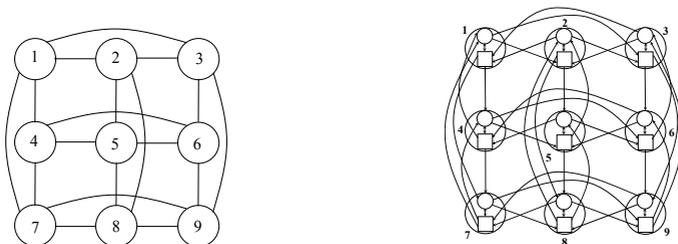


Рисунок 5. Двумерный тричный гиперкуб при $m = 3$ как граф и орграф

В орграфе (рисунок 5 справа) каждый узел содержит m -портового абонента (круг) и m -канального коммутатора $m \times m$ (квадрат), которые связаны m входными и m выходными дугами. В таком представлении двумерный m -ичный гиперкуб может быть представлен как двудольный орграф с $N = m^2$ узлами в каждой доле (рисунок 6), который именуется квазиполным орграфом.

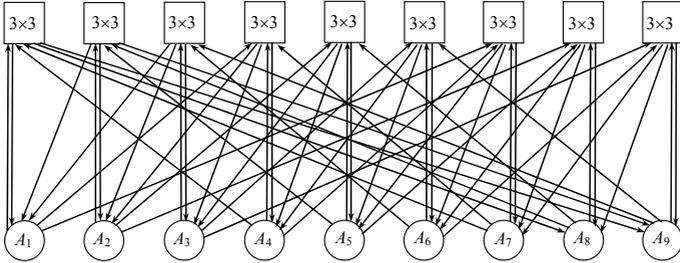


РИСУНОК 6. Квазиполный орграф при $m = 3$

Соединения в нем задаются разными таблицами инцидентности для дуг от абонентов и дуг к абонентам (таблицы 5 и 6). Они существуют при любых m .

Таблица 5. Таблица инцидентности для квазиполного орграфа при $m = 3$

Коммутаторы	Дуги от абонентов			Дуги к абонентам		
	1	2	3	1	4	7
1	1	2	3	1	4	7
2	1	2	3	2	5	8
3	1	2	3	3	6	9
4	4	5	6	1	4	7
5	4	5	6	2	5	8
6	4	5	6	3	6	9
7	7	8	9	1	4	7
8	7	8	9	2	5	8
9	7	8	9	3	6	9

Таблица 6. Таблица инцидентности для квазиполного орграфа при произвольном m

Коммутаторы	Дуги от абонентов				Дуги к абонентам			
	1	2	...	m	1	$1 + m$...	$1 + m(m - 1)$
1	1	2	...	m	1	$1 + m$...	$1 + m(m - 1)$
2	1	2	...	m	2	$2 + m$...	$2 + m(m - 1)$
...	1	2	...	m
m	1	2	...	m	m	$m + m$...	$m + m(m - 1)$
$m + 1$	$m + 1$	$m + 2$...	$m + m$	1	$1 + m$...	$1 + m(m - 1)$
$m + 2$	$m + 1$	$m + 2$...	$m + m$	2	$2 + m$...	$2 + m(m - 1)$
...	$m + 1$	$m + 2$...	$m + m$
$m + m$	$m + 1$	$m + 2$...	$m + m$	m	$m + m$...	$m + m(m - 1)$
...
$m^2 - m + 1$	$m^2 - m + 1$	$m^2 - m + 2$...	m^2	1	$1 + m$...	$1 + m(m - 1)$
...	$m^2 - m + 1$	$m^2 - m + 2$...	m^2	2	$2 + m$...	$2 + m(m - 1)$
$m^2 - 1$	$m^2 - m + 1$	$m^2 - m + 2$...	m^2
m^2	$m^2 - m + 1$	$m^2 - m + 2$...	m^2	m	$m + m$...	$m + m(m - 1)$

Для подсоединения однопортовых абонентов используются демультиплексоры $1 \times m$ и мультиплексоры $m \times 1$ соответственно (рисунок 7). Таким образом получается неблокируемый самомаршрутизируемый распре-

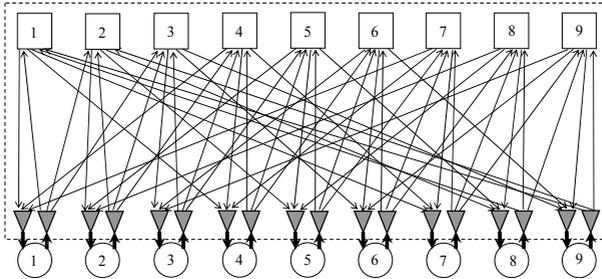


Рисунок 7. Коммутатор $PK_1(9, 3)$ с топологией квазицепоного орграфа при $m = 3$

деленный коммутатор $PK_1(N, m)$. В нем маршрутизация осуществляется от источника по адресам (номерам) выходных портов демультиплексоров и коммутаторов. Бесконфликтная передача пакетов на произвольной их перестановке осуществляется по прямым каналам за один скачок от источников к приемникам, т.е. с максимальным быстродействием.

Посредством внутреннего масштабирования можно построить ряд коммутаторов $PK_i(N_i, m)$ с $N_i = m^{i+1}$. Каждый коммутатор $PK_i(N_i, m)$ строится из $N = m^2$ коммутаторов $PK_{i-1}(N_{i-1}, m)$ и слоя из N_i демультиплексоров $1 \times m$ и мультиплексоров $m \times 1$.

Например, при $m = 4$ получим ряд коммутаторов с 16, 64, 256, и 1024 абонентами. Аналогично при $m = 16$ – ряд коммутаторов с 256, 4096 и 65536 абонентами.

При внешнем масштабировании можно построить ряд коммутаторов $PK_J(N_J, m_J)$ для $J = 2i (i = 1, 2, \dots)$ с $N_J = N_{J/2}^2$ и $m_J = N_{J/2}$, начиная с $PK_1(N_1, m_1)$ при $N_1 = N$ и $m_1 = m$.

Можно построить коммутатор $PK_2(N_2, m^2)$ на основе коммутатора $PK_1(N_1, m_1)$ и слоя мультиплексоров $m_1 \times 1$ и демультиплексоров $1 \times m_1$ с $m_1 = N_1 = m^2$. Он имеет $N_2 = N_1^2 = m^4$ каналов и по построению сохраняет неблокируемость и самомаршрутизируемость коммутатора $PK_1(N_1, m_1)$. Если демультиплексоры и мультиплексоры можно строить как деревья из исходных демультиплексоров $1 \times m$ и мультиплексоров $m \times 1$, то их общее число в дополнительном слое равно $N_1(m + 1)$.

Также, можно построить коммутатор $PK_4(N_4, m_4)$ на основе коммутатора $PK_2(N_2, m^2)$ и мультиплекторов $m_4 \times 1$ и демультиплекторов $1 \times m_4$ с $m_4 = N_2 = m^4$. Он имеет $N_4 = N_2^2 = m^8$ каналов сохраняет неблокируемость и самомаршрутизируемость коммутатора $PK_2(N_2, m^2)$.

Если демультиплексоры и мультиплексоры строить в виде деревьев из демультиплекторов $1 \times m$ и мультиплекторов $m \times 1$, то их общее число в дополнительном слое равно $N_2(m^3 + m^2 + m + 1)$.

Например, для $m = 4$ при внешнем каскадировании получим ряд коммутаторов с 16, 256 и 65536 абонентами.

4. ИСС на основе комбинированной топологии

Сопряжение коммутаторов разных топологий осуществляется как внутреннее расширение коммутатора $PK_i(N_i, m_i)$ с топологией квазиполного орграфа, при котором на последнем шаге используется коммутатор с топологией квазиполного графа $PK_1(N, m, \sigma)$. Такой расширенный коммутатор будем обозначать как $PPK(M, N_i, m_i, N, m, \sigma)$, где M задает число его абонентов определяемое как $M = \lfloor N_i/m \rfloor N + \delta$, где $0 \leq \delta < N$. Построенный расширенный коммутатор является неблокируемой самомаршрутизируемой системной сетью с прямыми каналами, т.е. ИСС, которая объединяет M абонентов и имеет σ разных прямых путей между любыми двумя абонентами.

Для примера в таблицах 7 и 8 приводятся таблицы инцидентности для внутреннего расширения коммутатора $PK_2(16, 4)$ коммутаторами $PK_1(5, 4, 3)$ и $PK_1(7, 5, 3)$. При этом последний коммутатор изоморфен 1-расширенной блок-схеме $B^*(7, 5, 3)$ с 3 или 4 разными путями между абонентами.

Таблица 7. Таблица инцидентности коммутатора $PPK(20, 16, 4, 5, 4, 3)$

PK_2	$PPK(20, 16, 4, 5, 4, 3)$ с $\delta = 0$															
Порты	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	2	3	4	6	7	8	9	11	12	13	14	16	17	18	19
2	1	2	3	5	6	7	8	10	11	12	13	15	16	17	18	20
3	1	2	4	5	6	7	9	10	11	12	14	15	16	17	19	20
4	1	3	4	5	6	8	9	10	11	13	14	15	16	18	19	20
5	2	3	4	5	7	8	9	10	12	13	14	15	17	18	19	20

ТАБЛИЦА 8. Таблица инцидентности коммутатора РПК(22, 16, 4, 7, 5, 3)

РК ₂	РПК(22, 16, 4, 7, 5, 3) с $\delta = 1$															
Порты	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	2	3	4	5	8	9	10	11	12	15	16	17	18	19	22
2	1	2	3	4	7	8	9	10	11	14	15	16	17	18	21	22
3	1	2	3	6	7	8	9	10	13	14	15	16	17	20	21	22
4	1	2	5	6	7	8	9	12	13	14	15	16	19	20	21	22
5	1	4	5	6	7	8	11	12	13	14	15	18	19	20	21	22
6	2	3	4	5	6	9	10	11	12	13	16	17	18	19	20	
7	3	4	5	6	7	10	11	12	13	14	17	18	19	20	21	

Можно на последнем шаге масштабирования вместо коммутатора $РК_1(N, m, \sigma)$ с однопортовыми абонентами (рисунок 2) использовать простейшую сеть $ПС(N, m, \sigma)$ с m -портовыми абонентами (рисунок 1). В этом случае коммутатор $РПК(M, N_i, m_i, N, m, \sigma)$ превращается в расширенную сеть $РПС(M, N_i, m_i, N, m, \sigma)$, в которой возможна одновременная бесконфликтная передача по σ разным путям между любыми двумя абонентами. Будем характеризовать расширенную сеть следующими параметрами: числом абонентов M , числом добавленных абонентов $D = M - N_i$ сложностью S , задаваемую как число копий коммутатора $РК_i(N_i, m_i)$, используемых на последнем шаге расширения, т.е. $S = N$. Выбор параметров сети $ПС(N, m, \sigma)$ будем осуществлять посредством максимизации критерия $g = \sigma D/S$. Значения критерия вычисляются по формуле $g = \sigma/m - \sigma^2/(m(m-1) + \sigma)$ и приводятся в таблице 9 и графиком на рисунке 8 с точностью до двух знаков.

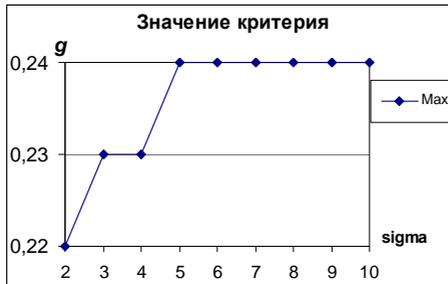
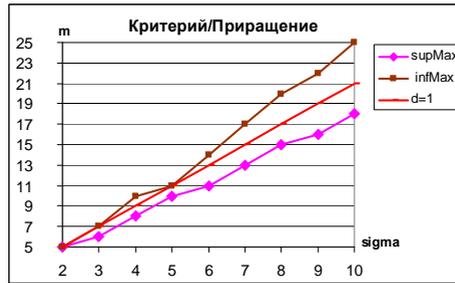


РИСУНОК 8. Значения критерия g

Таблица 9. Значения критерия g

$m \backslash \sigma$	2	3	4	5	6	7	8	9	10
2	0,00								
3	0,17	0,00							
4	0,21	0,15	0,00						
5	0,22	0,21	0,13	0,00					
6	0,21	0,23	0,20	0,12	0,00				
7	0,19	0,23	0,22	0,18	0,11	0,00			
8	0,18	0,22	0,23	0,22	0,17	0,10	0,00		
9	0,17	0,21	0,23	0,23	0,21	0,16	0,09	0,00	
10	0,16	0,20	0,23	0,24	0,23	0,19	0,15	0,08	0,00
11	0,15	0,19	0,22	0,24	0,24	0,22	0,18	0,14	0,08
12	0,14	0,18	0,22	0,23	0,24	0,23	0,21	0,18	0,13
13	0,13	0,17	0,21	0,23	0,24	0,24	0,23	0,20	0,17
14	0,12	0,17	0,20	0,22	0,24	0,24	0,23	0,22	0,19
15	0,11	0,16	0,19	0,22	0,23	0,24	0,24	0,23	0,21
16	0,11	0,15	0,18	0,21	0,23	0,24	0,24	0,24	0,23
17	0,10	0,14	0,18	0,20	0,22	0,24	0,24	0,24	0,23
18	0,10	0,14	0,17	0,20	0,22	0,23	0,24	0,24	0,24
19	0,09	0,13	0,16	0,19	0,21	0,23	0,24	0,24	0,24
20	0,09	0,13	0,16	0,19	0,21	0,22	0,24	0,24	0,24
21	0,09	0,12	0,15	0,18	0,20	0,22	0,23	0,24	0,24
22	0,08	0,12	0,15	0,17	0,20	0,21	0,23	0,24	0,24
23	0,08	0,11	0,14	0,17	0,19	0,21	0,22	0,23	0,24
24	0,08	0,11	0,14	0,16	0,19	0,20	0,22	0,23	0,24
25	0,07	0,11	0,13	0,16	0,18	0,20	0,21	0,23	0,24
26	0,07	0,10	0,13	0,15	0,18	0,19	0,21	0,22	0,23
27	0,07	0,10	0,13	0,15	0,17	0,19	0,21	0,22	0,23
28	0,07	0,10	0,12	0,15	0,17	0,19	0,20	0,22	0,23
29	0,06	0,09	0,12	0,14	0,16	0,18	0,20	0,21	0,22
30	0,06	0,09	0,12	0,14	0,16	0,18	0,19	0,21	0,22

Максимальные значения критерия g имеют верхнюю и нижнюю границы, которые представлены графиком на рисунке 9. В эти границы



Рисунк 9. Верхние и нижние границы максимальных значений критерия g

укладывается значения $D \approx N_i$, задаваемые отношением $d \approx D/N_i \approx 1$.

Примеры того, какие ИСС можно построить на сопряжении коммутатора РК₃(4096, 16) и сети ПС($N, m, 6$) (см. таблица 3 и таблица 9), приведены в таблице 10.

Таблица 10. Примеры ИСС на основе РК₃(4096, 16) и ПС($N, m, 6$)

Сеть	Расширение	D	S
ПС(6, 6, 6)	РПС(4096, 4096, 16, 6, 6, 6)	0	6
ПС(10, 8, 6)	РПС(5120, 4096, 16, 10, 8, 6)	1024	10
ПС(26, 13, 6)	РПС(7510, 4096, 16, 26, 13, 6)	4096	26
ПС(10, 10, 10)	РПС(4096, 4096, 16, 10, 10, 10)	0	10
ПС(14, 12, 10)	РПС(4783, 4096, 16, 14, 12, 10)	687	14
ПС(24, 16, 10)	РПС(6144, 4096, 16, 24, 16, 10)	2048	24

Основными целями построения ИСС на основе РПС($M, N_i, m_i, N, m, \sigma$) являются повышение ее быстродействия и отказоустойчивости.

Быстродействие повышается посредством разбиения пакетов на части и параллельной передачей частей по разным путям между источниками и приемниками. Накладными затратами при этом являются необходимость иметь отдельного заголовка с адресом приемника в каждой части и повышенная сложность ИСС вследствие использования копий исходной (расширяемой) сети.

Отказоустойчивость в ИСС повышается посредством резервирования в ней как каналов, так и абонентов. При этом если параллельно исполь-

звать $\sigma^* < \sigma$ путей между абонентами, то отказоустойчивость ИСС можно обеспечить и по ее быстродействию. При уменьшении числа разных путей меньше σ^* работоспособность ИСС может сохраняться посредством уменьшения ее быстродействия, что обеспечивает ее канальную живучесть.

Повышенное быстродействие ИСС можно использовать для создания в вычислительной системе групп из $\sigma + 1$ абонентов с повышенной отказоустойчивостью. Эти абоненты решают одну и ту же задачу (может по разным алгоритмам) с периодической сверкой результатов. Для сверки результатов каждый абонент посылает свой результат остальным участникам группы в одном сеансе по σ разным путям с предварительной синхронизацией этой рассылки. Правильным считается результат, совпадающий по большинству (может относительно) абонентов.

Порты m -портового абонента можно реализовать как набор виртуальных сдвиговых регистров (рисунок 10). Внутри абонента пакеты в них

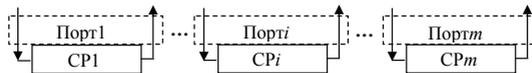


Рисунок 10. Структура m -портового абонента.

заносится независимо, но запускается на сдвиг (передачу) одновременно. Наоборот, пакеты в регистры принимаются из сети асинхронно, но считываются одновременно. В каждом сеансе задействуются только σ регистров. Поэтому сеанс задается не только заполнением регистров, но и маской, которая выделяет σ задействованных регистров.

Для выравнивания времен обмена пакетами со сдвиговыми регистрами внутри абонента и между абонентами по сети можно иметь несколько наборов сдвиговых регистров, используемых попеременно. Эти наборы в совокупности образуют многопортовую сетевую карту с параллельными портами.

Заключение

В статье рассмотрен метод дополнительного распараллеливания системных сетей. Он применим к любым системным сетям как метод их инвариантного расширения на основе простейших сетей с топологией квазиполного графа (раздел 2). Этот метод применим даже к идеальным системным сетям с казалось бы наибольшей параллельностью, которая обеспечивает их неблокируемость на произвольных перестановках при любых их размерах (раздел 3)

В статье рассмотрен метод использования дополнительной параллельности системных сетей для повышения их пропускной способности вместе с абонентским быстродействием и повышения канальной и абонентской отказоустойчивости. Рассмотренный метод может быть использован применительно к системной сети «Ангара», имеющей структуру одномерного, двумерного либо трёхмерного тора.

Список использованных источников

- [1] Kumar A., Peh L.-S., Kundu P., Jha N. K. *Toward ideal on-chip communication using express virtual channels* // IEEE Micro.– Jan/Feb 2008.– Vol. **28**.– No. 1.– Pp. 80–90.  [↑42](#)
- [2] Каравай М. Ф., Подлазов В. С. *Метод инвариантного расширения системных сетей многопроцессорных вычислительных систем. Идеальная системная сеть* // Автомат. и телемех.– 2010.– № 12.– С. 166–177.  [↑42, 43, 49](#)
- [3] Каравай М. Ф., Подлазов В. С. *Распределенный полный коммутатор как «идеальная» системная сеть для многопроцессорных вычислительных систем* // Управление большими системами.– 2011.– Т. **34**.– С. 92–116.  [↑42, 43, 49](#)
- [4] Подлазов В. С. *Самомаршрутизируемая неблокируемая системная сеть с прямыми каналами: сложность и быстродействие* // Программные системы: теория и приложения.– 2022.– Т. **13**.– № 4 (55).– С. 47–76.   [↑42, 44, 47, 49](#)
- [5] Подлазов В. С. *Разные неблокируемые самомаршрутизируемые системные сети с прямыми каналами* // Программные системы: теория и приложения.– 2023.– Т. **14**.– № 3 (58).– С. 115–138.   [↑42, 44, 49](#)
- [6] Холл М. *Комбинаторика. Главы 10–12*.– М.: Мир.– 1970.– 424 с. [↑42, 43, 49](#)
- [7] Каравай М. Ф., Пархоменко П. П., Подлазов В. С. *Комбинаторные методы построения двудольных однородных минимальных квазиполных графов (симметричных блок-схем)* // Автомат. и телемех.– 2009.– № 2.– С. 153–170.  [↑43, 47, 49](#)
- [8] Каравай М. Ф., Подлазов В. С. *Расширенные блок-схемы для идеальных системных сетей* // Проблемы управления.– 2012.– № 4.– С. 45–51.  [↑44, 47, 49](#)
- [9] Scott S., Abts D., Kim J., Dally W. *The BlackWidow high-radix Clos network* // 3rd International Symposium on Computer Architecture, ISCA'06 (17–21 June 2006, Boston, MA, USA).– IEEE.– 2006.– ISBN 0-7695-2608-X.– Pp. 16–28.  [↑44, 48](#)
- [10] Alverson R., Roweth D., Kaplan L., Roweth D. *Cray® XCTM series network*, WP-Aries01-1112.– Cray.– 28 pp.  [↑44, 48](#)
- [11] Kim J., Dally W. J., Scott S., Abts D. *Technology-driven, highly-scalable dragonfly topology* // 2008 International Symposium on Computer Architecture (21–25 June 2008, Beijing, China).– IEEE.– 2008.– ISBN 978-0-7695-3174-8.– Pp. 77–88.  [↑44, 48, 49](#)
- [12] Arimilli B., Arimilli R., Chung V., Clark S., Denzel W., Drerup B. *The PERCS high-performance interconnect* // 2010 18th IEEE Symposium on High Performance Interconnects (18–20 August 2010, Mountain View, CA, USA).– IEEE.– 2010.– ISBN 978-0-7695-4208-9.– Pp. 75–82.  [↑48, 49, 55](#)

- [13] Alverson R., Roweth D., Kaplan L. *The Gemini system interconnect // 2010 18th IEEE Symposium on High Performance Interconnects (18–20 August 2010, Mountain View, CA, USA).*– IEEE.– 2010.– ISBN 978-0-7695-4208-9.– Pp. 83–87. doi ↑48, 55
- [14] *Российский интерконнект Ангара-2: 200 Гбит/с при задержках до 0,8 мкс.*– ServerNews.– 2021. URL ↑48, 55
- [15] Ajima Yu., Inoue T., Hiramoto Sh., Shimiz T. *Tofu: Interconnect for the K computer // Fujitsu Scientific & Technical Journal.*– 2012.– Vol. 48.– No. 3.– Pp. 280–285. URL ↑48, 49
- [16] Bhuyan L. N., Agrawal D. P. *Generalized hypercube and hyperbus structures for a computer network // IEEE Trans. on Computers.*– 1984.– Vol. C-33.– No. 4.– Pp. 323–333. doi ↑48
- [17] Tzeng N., Wei S. *Enhanced hypercubes // IEEE Transactions on Computers.*– 1991.– Vol. 40.– No. 3.– Pp. 284–294. doi ↑48
- [18] Efe K. *A variation on the hypercube with lower diameter // IEEE Transactions on Computers.*– 1991.– Vol. 40.– No. 11.– Pp. 1312–1316. doi ↑48
- [19] Каравай М. Ф., Подлазов В. С. *Расширение возможностей системной сети «Ангара» // Проблемы управления.*– 2020.– № 2.– С. 47–56. doi ↑
- [20] Каравай М. Ф., Подлазов В. С. *Оптимальные отказоустойчивые многомерные торы на основе малопортовых маршрутизаторов и забов // Проблемы управления.*– 2020.– № 5.– С. 56–64. doi ↑

Поступила в редакцию 24.03.2025;
одобрена после рецензирования 17.04.2025;
принята к публикации 17.04.2025;
опубликована онлайн 04.08.2025.

Рекомендовал к публикации

к.ф.-м.н. С. А. Романенко

Информация об авторе:



Виктор Сергеевич Подлазов

Д. т. н., гл.н.с. Института проблем управления им. В.А. Трапезникова РАН, научные интересы: архитектуры интерконнекта и маршрутизация в суперкомпьютерных системах



0000-0002-9175-1138

e-mail:

Декларация об отсутствии личной заинтересованности: *благополучие автора не зависит от результатов исследования.*



Performance and fault tolerance of the ideal system network via additional parallelity

Viktor Sergeevich **Podlazov**

V. A. Trapeznikov Institute of Control Sciences of RAS, Moscow, Russia

 podlazov@ipu.ru

Abstract. The article considers methods for increasing the performance of system networks by increasing their parallelism in two ways: using their copies and parallel channels in each copy. The article considers methods for constructing non-blocking system networks with direct channels of arbitrary size. The article considers methods for constructing non-blocking networks with a given number of parallel channels as networks with the structure of quasi-complete graphs. The article considers a method for constructing system networks of any size with a given number of parallel channels by using the corresponding number of their copies. (*In Russian*).

Key words and phrases: complete switch, direct channels, multiplexers and demultiplexers, multi-stage switch, conflictfree routing, non-blocking switch, static self-routing, quasi-complete graph with a given number of parallel channels, isomorphism of a quasi-complete graph and a symmetric block-design

2020 *Mathematics Subject Classification:* 65Y05; 68Q10

For citation: Viktor S. Podlazov. *Performance and fault tolerance of the ideal system network via additional parallelity*. Program Systems: Theory and Applications, 2025, **16**:3(66), pp. 41–68. (*In Russ.*).
https://psta.psiras.ru/read/psta2025_3_41-68.pdf

References

- [1] A. Kumar, L.-S. Peh, P. Kundu, N.K. Jha. “Toward ideal on-chip communication using express virtual channels”, *IEEE Micro*, **28**:1 (Jan/Feb 2008), pp. 80–90. 

- [2] M. F. Karavaj, V. S. Podlazov. “An invariant extension method for system area networks of multicore computational systems. An ideal system network”, *Autom. Remote Control*, **71**:12 (2010), pp. 2644–2654.
- [3] M. F. Karavaj, V. S. Podlazov. “Distributed full switch as ideal system area network for multiprocessor computers”, *Upravlenie bol'shimi sistemami*, **34** (2011), pp. 92–116 (in Russian).
- [4] V. S. Podlazov. “Multichannel non-blocking system area network with direct channels”, *Program Systems: Theory and Applications*, **13**:4 (55) (2022), pp. 47–76 (in Russian).
- [5] V. S. Podlazov. “Multichannel non-blocking system area network with direct channels”, *Program Systems: Theory and Applications*, **14**:3 (58) (2023), pp. 115–138 (in Russian).
- [6] M. Hall (Jr.). *Combinatorial Theory*, A Blaisdell Book in Pure and Applied Mathematics, Blaisdell Publishing Company, Waltham–Toronto–London, 1967, ISBN 978-0471315186, 310 pp.
- [7] M. F. Karavaj, P. P. Parxomenko, V. S. Podlazov. “Combinatorial methods for constructing bipartite uniform minimal quasicomplete graphs (symmetrical block designs)”, *Autom. Remote Control*, 2009, no. 2, pp. 153–170.
- [8] M. F. Karavaj, V. S. Podlazov. “Expanded block-diagrams for “ideal” system area networks”, *Automation and Remote Control*, **74**:12 (2013), pp. 2180–2188.
- [9] S. Scott, D. Abts, J. Kim, W. Dally. “The BlackWidow high-radix Clos network”, *3rd International Symposium on Computer Architecture*, ISCA'06 (17–21 June 2006, Boston, MA, USA), IEEE, 2006, ISBN 0-7695-2608-X, pp. 16–28.
- [10] R. Alverson, D. Roweth, L. Kaplan, D. Roweth. *Cray® XCTM series network*, WP-Aries01-1112, Cray, 28 pp.
- [11] J. Kim, W. J. Dally, S. Scott, D. Abts. “Technology-driven, highly-scalable dragonfly topology”, *2008 International Symposium on Computer Architecture* (21–25 June 2008, Beijing, China), IEEE, 2008, ISBN 978-0-7695-3174-8, pp. 77–88.
- [12] B. Arimilli, R. Arimilli, V. Chung, S. Clark, W. Denzel, B. Drerup. “The PERCS high-performance interconnect”, *2010 18th IEEE Symposium on High Performance Interconnects* (18–20 August 2010, Mountain View, CA, USA), IEEE, 2010, ISBN 978-0-7695-4208-9, pp. 75–82.
- [13] R. Alverson, D. Roweth, L. Kaplan. “The Gemini system interconnect”, *2010 18th IEEE Symposium on High Performance Interconnects* (18–20 August 2010, Mountain View, CA, USA), IEEE, 2010, ISBN 978-0-7695-4208-9, pp. 83–87.
- [14] *Russian Angara-2 interconnect: 200 Gbps with delays up to 0.8 μs*, ServerNews, 2021 (in Russian).
- [15] Yu. Ajima, T. Inoue, Sh. Hiramoto, T. Shimiz. “Tofu: Interconnect for the K computer”, *Fujitsu Scientific & Technical Journal*, **48**:3 (2012), pp. 280–285.

- [16] L. N. Bhuyan, D. P. Agrawal. “Generalized hypercube and hyperbus structures for a computer network”, *IEEE Trans. on Computers*, **C-33**:4 (1984), pp. 323–333. 
- [17] N. Tzeng, S. Wei. “Enhanced hypercubes”, *IEEE Transactions on Computers*, **40**:3 (1991), pp. 284–294. 
- [18] K. Efe. “A variation on the hypercube with lower diameter”, *IEEE Transactions on Computers*, **40**:11 (1991), pp. 1312–1316. 
- [19] M. F. Karavaj, V. S. Podlazov. “Expanding the capabilities of the Angara system area network”, *Problemy upravleniya*, 2020, no. 2, pp. 47–56 (in Russian). 
- [20] M. F. Karavaj, V. S. Podlazov. “Optimum multidimensional tori based on low-port routers and hubs”, *Problemy upravleniya*, 2020, no. 5, pp. 56–64 (in Russian). 

УДК 004.93'11

10.25209/2079-3316-2025-16-3-69-97



Определение здоровых и больных областей листьев растений при помощи нейронных сетей

Александр Владимирович Смирнов^{1,2}, Игорь Петрович Тищенко²

^{1,2}Институт программных систем им. А. К. Айламазяна РАН, Вельсково, Россия

^{1,2}asmirmov_1991@mail.ru

Аннотация. В настоящей статье представлено исследование, направленное на разработку нейросетевого метода обнаружения здоровых и больных областей листьев растений по их изображениям с вычислением соотношения их площадей. В качестве нейросетевой модели использовалась базовая сеть архитектуры FPN с энкодером в виде архитектуры ResNet-34. Для обучения ИНС в качестве меток использовались бинарные маски целевых областей листьев растений, которые были получены программным путём без ручной разметки. Благодаря этому удалось достичь разумного компромисса между ресурсами, необходимыми для создания масок, и их точностью. При обучении нейросетевой модели была достигнута точность в 96.5% и 78.9% по метрике F1 для определения здоровых и больных областей соответственно. Далее был произведён инференс модели, в результате которого был рассчитан индекс «здоровья» для каждого из исследуемых изображений листьев. В контексте решаемых задач, индекс «здоровья» представляет собой разность между процентами долями здоровой и больной областей, который может быть использован при оценке тяжести заболевания, а также при мониторинге динамики развития болезни как индикатор эффективности используемых препаратов или методов ухода. Научная новизна представленного исследования заключается в создании метода автоматического определения соотношения площадей здоровых и больных участков листьев, который сочетает современные технологии компьютерного зрения, машинного обучения и практическую применимость для агрономии и растениеводства.

Ключевые слова и фразы: нейросетевой анализ, индекс «здоровья», здоровая область листа, больная область листа, модель

Для цитирования: Смирнов А. В., Тищенко И. П. *Определение здоровых и больных областей листьев растений при помощи нейронных сетей* // Программные системы: теория и приложения. 2025. Т. 16. № 3(66). С. 69–97. https://psta.psisras.ru/read/psta2025_3_69-97.pdf

Введение

В последние годы наблюдается тенденция к цифровизации агропромышленного комплекса, направленная на повышение производительности, снижение издержек и устойчивое управление природными ресурсами. Одной из актуальных задач современного сельского хозяйства является мониторинг состояния растений с целью своевременного выявления заболеваний и других негативных изменений, способных повлиять на урожайность. Особое внимание уделяется разработке и внедрению автоматизированных систем, способных выполнять визуальный анализ растений и оперативно выявлять признаки заболеваний.

Одним из наиболее перспективных направлений в области автоматизации ухода за растениями является использование методов искусственного интеллекта, в частности искусственных нейронных сетей (ИНС). ИНС позволяют выполнять анализ изображений и выделять сложные паттерны. Однако в контексте обработки изображений растений необходимо иметь релевантный набор обучающих данных. Так работа [1] посвящена сбору набора данных DiaMOS Plant, предназначенного специально для диагностики и мониторинга заболеваний растений. Представленный набор состоит из 3505 изображений плодов и листьев груши, пораженных четырьмя видами заболеваний. Хотя DiaMOS Plant предоставляет богатый источник данных, имеются и определенные ограничения. Во-первых, набор данных ограничен теми регионами, где проводились съемки, что может повлиять на универсальность модели. Во-вторых, некоторое число изображений может содержать артефакты или шумы, влияющие на качество результата.

В другой работе [2] также описывается формирование крупного набора данных Plant Disease Dataset (PDD). Набор PDD содержит 220592 изображения в 271 категории, каждая из которых отвечает за отдельный вид заболевания. Помимо обучающего набора данных, авторы также предлагают свой подход к реализации системы распознавания болезней растений, который основан на трёх принципах: Cluster-Based Region Reweighting, Training With Loss Reweighting и Weighted Feature Integration. Cluster-Based Region Reweighting позволяет обрабатывать небольшие участки изображений с очагами заболевания растения не пропуская их. Training With Loss Reweighting – усиливает влияние фрагментов изображения с болезнями, а Weighted Feature Integration служит для комплексного представления признаков для распознавания. В целом, авторам удалось улучшить точность распознавания болезней примерно на 5% по метрике $Top-1$ ^{URU}.

Исследования в работах [3], [4] посвящены обзору существующих методов диагностики заболеваний растений с использованием технологий глубокого обучения (*Deep Learning*^{URU}). Большое внимание уделяется методам, основанным на анализе изображений, в частности рассматривается применение свёрточных нейронных сетей (CNN) [5] в качестве классификатора изображений, рекуррентных нейронных сетей (RNN) [6] для анализа временных рядов и изменения состояния растения со временем, и генеративно-состязательных сетей (GANs) [7] для создания искусственных образцов здоровых и больных растений, которыми можно расширить обучающий набор данных. Также проанализировано влияние трансферного обучения (Transfer Learning) [8] на итоговую точность распознавания заболеваний. Рассматриваются конкретные архитектуры нейронных сетей такие как U-Net [9] и Mask R-CNN [10] в контексте анализа изображений растений. В итоге, авторы приводят описание проблем использования глубокого обучения для анализа заболеваний растений и методов их решения.

В статье [11] авторы предлагают метод диагностики заболеваний растений на основе анализа цвета и текстурных признаков, извлекаемых из изображений листьев. Данный метод состоит из нескольких последовательных шагов, включая предварительную обработку изображений, сегментацию заражённых областей листьев и расчёт признаков на основе матрицы совпадений уровня серого (GLCM) [12]. В качестве классификатора использовались различные алгоритмы такие, как метод опорных векторов (*SVM*^{URU}) и искусственные нейронные сети (ИНС). В результате авторам удалось добиться точности классификации больных и здоровых растений в 91.40%, и точности определения вида заболевания в 82.47%.

Статья [13] предлагает легковесную нейросетевую архитектуру на основе механизмов свёрточных нейронных сетей и модели *SSD*^{URU} для обнаружения и идентификация заболеваний листьев растений в режиме реального времени. Основной особенностью представленной архитектуры является то, что она способна работать на встраиваемом оборудовании таком, как *Nvidia Jetson TX1*^{URU}, с точностью классификации болезней составляет около 96%.

Следует отметить, что часть исследований направлены на идентификацию и анализ изображений отдельных групп или видов растений. Например, в работе [14] проводится нейросетевая классификация лекарственных растений. В качестве классификатора авторы используют

сеть VGG-19 [15]. Эксперимент был проведен с помощью набора данных *Flavia*^{URL}, который содержит 1907 изображения в 32 классах.

Другая работа [16] посвящена определению болезней на изображениях листьев яблони. Авторы предлагают собственную архитектуру свёрточной нейронной сети, состоящей из комбинации слоёв свёртки, пулинга и полносвязных слоёв. Для обучения авторы используют набор данных *PlantVillage*^{URL}. В результате предложенная архитектура обеспечивает точность в 98% при классификации болезней яблони.

В статьях [17], [18] рассматривается применение ИНС для распознавания болезней продовольственных растений, таких как томаты и томаты черри. Обсуждается выбор архитектуры сети и обучающего набора данных. Всего в представленных исследованиях были задействованы модели AlexNet [19], SqueezeNet [20], Faster R-CNN [21] и Mask R-CNN. Итоговая достигнутая точность составила 91.67% и 93.76% соответственно.

В последнее время стали появляться гибридные алгоритмы и отдельные *фреймворки*^{URL} для классификации болезней растений. Для примера, в работе [22] предложена гибридная модель, включающая алгоритм опыления цветка (FPA) [23] и метод опорных векторов (*SVM*^{URL}), а также классификатор на базе свёрточной нейронной сети. Обучающий набор данных состоял из изображений листьев яблок, винограда и томатов, характерные признаки которых извлекались с помощью двумерного дискретного вейвлет-преобразования (*2D-DWT*^{URL}). Разработанная гибридная модель была встроена в комплект разработчика *Nvidia Jetson Nano*^{URL} и протестирована с использованием БПЛА для классификации болезней растений в реальном времени. Полученные экспериментальные результаты показывают, что предложенная модель классифицирует указанные заболевания листьев растений с высокой точностью – более 90%.

Следующая статья [24] представляет фреймворк Agriculture Detection (AgriDet) для обнаружения болезней растений по изображениям. AgriDet включает в себя нейронную сеть Inception-Visual Geometry Group Network (INC-VGGN) [25] и *глубинные сети Кохонена*^{URL}. Здесь предварительно обученная модель INC-VGGN представляет собой глубокую свёрточную нейронную сеть для прогнозирования заболеваний растений. Чтобы преодолеть проблему переобучения, вводится слой исключения, а глубокое обучение выполняется с использованием обучающего слоя Кохонена. В результате достигается лучшая точность, в сравнении с другими моделями.

Исходя из выполненного обзора схожих по тематике работ, можно сделать вывод о том, что решения по автоматизации определения заболеваний растений остаются востребованными. Современные технологии позволяют создавать программные средства, обеспечивающие визуальный анализ состояния растений с высокой точностью и минимальным участием человека. Это особенно важно для раннего выявления заболеваний, проявляющихся в изменениях формы, структуры и окраски листовых пластин. Традиционные методы, основанные на ручной диагностике, характеризуются высокой трудоёмкостью, субъективностью оценок и ограниченными возможностями масштабирования.

В настоящей статье рассматривается определение процентного соотношения здоровых и больных областей листьев растения. Несмотря на то, что в большинстве случаев достаточно зафиксировать сам факт заболевания растения для принятия каких-либо действий, определение процентного соотношения здоровых и больных областей может использоваться при оценке тяжести заболевания, а также при мониторинге динамики развития болезни, в качестве индикатора эффективности используемых препаратов или методов ухода.

Научная новизна представленного исследования заключается в разработке метода автоматического определения соотношения площадей здоровых и больных участков листьев растений, основанного на использовании современных технологий компьютерного зрения, машинного обучения, и имеющего практическую применимость для агрономии и растениеводства.

1. Цель и задачи исследования

Целью исследования, представленного в настоящей статье, является разработка и тестирование метода обнаружения здоровых и больных областей листовых пластин (далее листьев) растений путём анализа их изображений с использованием технологий ИНС. Определение отношения площадей здоровых и больных областей листа растения поможет установить степень его заболевания, а также позволит выявлять патологии на ранних стадиях развития и контролировать динамику состояния растения.

Среди задач выполненного исследования можно выделить следующие:

- Выбор и подготовка обучающего набора данных.
- Выбор, обучение и инференс модели нейронной сети для определения здоровых и больных областей листьев растений.

- Визуализация и анализ результатов. Подсчёт индекса «здоровья» листа – отношения площади его здоровой области к больной.

Достигнутые результаты могут быть использованы в качестве инструмента «обратной связи» с растением для автоматизированных систем ухода с визуальным контролем растительности. Будущие исследования будут направлены на доработку разработанного метода, заключающиеся в подборе оптимальной нейросетевой архитектуры и добавлении функционала распознавания болезней растений.

2. Подготовка набора данных

2.1. Описание обучающего набора данных

В качестве обучающего набора данных использовался датасет PlantVillage, содержащий в общей сложности 54305 изображений в 38 классах. Представленный набор данных содержит изображения листьев различных видов растений таких как: Яблоня (Apple), Вишня (Cherry), Черника (Blueberry), Виноград (Grape) и др. Среди заболеваний растений представлены такие как:

- парша (scab);
- гниль (rot);
- ржавчина (rust);
- мучнистая роса (powdery mildew);
- пятнистость листьев (spot, leaf blight);
- ожог листьев (leaf scorch);
- бактериальные пятна (bacterial spot);
- чёрная корь (black measles);
- вирус курчавости листьев (leaf curl virus).

В процессе поиска обучающего набора данных, были рассмотрены и другие датасеты, например, *PDD271*^{[URL](#)}, *Crop Pest and Disease Detection*^{[URL](#)} и *FieldPlant*^{[URL](#)}. Основными критериями отбора были следующие показатели:

- Наличие разнообразных классов заболеваний растений, а также изображений со здоровыми растениями.
- Единый формат изображений растений/листьев растений – единое разрешение у всех изображений и одинаковый метод съёмки листьев.

- Возможность загрузить полную версию набора данных, а не только примеры.

Используемый набор PlantVillage удовлетворял всем вышеизложенным критериям. Данный набор состоит из изображений листьев растений с разрешением 256x256 пикселей. Съёмка листьев выполнялась на специальной подложке, цвет которой отличен от цвета листа растения. На рисунке 1 показан пример изображений из данного набора.



Рисунок 1. Пример изображений набора PlantVillage

Существует расширенная версия *PlantVillage-Dataset*^{URL} набора данных PlantVillage. Данная версия помимо оригинальных изображений листьев растений также содержит те же изображения, но без фона (фон заменён на чёрный цвет), которые использовались в настоящем исследовании для подготовки обучающих данных. На рисунке 2 показан пример изображений листьев без фона.

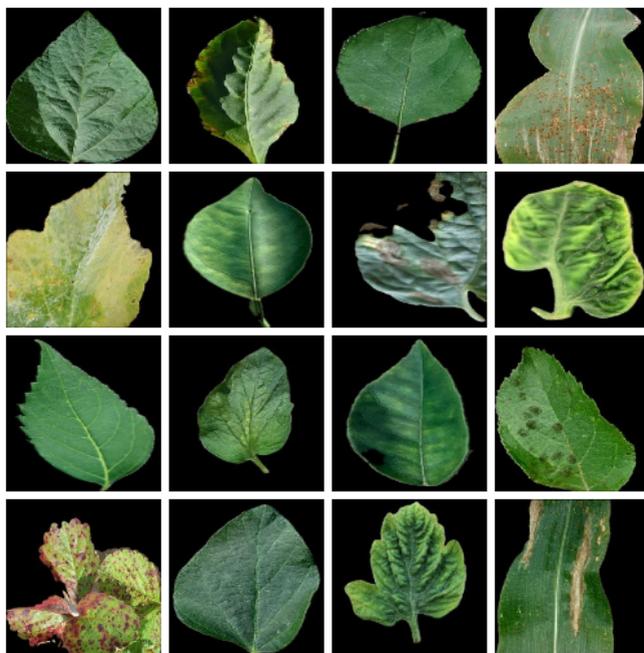


РИСУНОК 2. Пример изображений набора PlantVillage без фона

2.2. Формирование выборок и создание бинарных масок

Так как для решения поставленных задач нет необходимости в определении конкретного заболевания растения, используемый набор данных был разделён на две выборки «здоровую» и «больную». В «здоровую» выборку вошли все изображения, классы которых имели пометку «healthy». В «больную» выборку вошли изображения во всех остальных классах. Таким образом, удалось сгруппировать изображения здоровых и больных листьев, тем самым обобщая их по цвету и текстуре. Объединение всех изображений в «здоровую» и «больную» выборки было сделано для того, чтобы полученное решение обладало большей инвариантностью и не зависело от конкретного растения или заболевания.

Обычно обучение ИНС происходит с использованием промаркированных входных данных, то есть таких данных, которые помимо самих изображений содержат файлы с описанием классов (аннотационные файлы) в формате txt или csv. В процессе обучения происходит сравнение

предсказанного класса с целевым, описанном в аннотационном файле. Затем результат этого сравнения передаётся в архитектуру ИНС для выполнения коррекции весовых коэффициентов.

Набор данных PlantVillage уже содержит разделение изображений по классам, однако, в контексте решаемых задач требуется промаркировать определённые области на изображениях. Для этого используются так называемые бинарные маски, где белыми пикселями маркируются интересующие области изображения. Соответственно для «здоровой» выборки интересующими областями были сами листья растения, а для «больной» – области, поражённые заболеванием, то есть очаги заболеваний. Для создания бинарных масок использовались изображения без фона, доступные в расширенной версии набора данных, что позволило упростить алгоритм создания масок и потенциально увеличить их точность, благодаря отсутствию необходимости в удалении фона.

2.3. Алгоритм создания бинарных масок для «здоровой» выборки

Считается, что наиболее точные бинарные маски можно получить методом ручной разметки, при которой эксперт-разметчик непосредственно выполняет разметку данных. Тем не менее, оценивая ресурсозатраты, необходимые на ручную разметку, а также учитывая специфику входных данных, было принято решение реализовать программное формирование бинарных масок.

Алгоритм создания масок для «здоровой» выборки состоял из следующих шагов:

- (1) Перевод исходных изображений в цветовое пространство *CIE Lab*^(UR), определение диапазона значений для каналов L, a и b, и поканальная бинаризация.
- (2) Контурный анализ полученного на предыдущем шаге изображения с поиском и закрасиванием описывающего контура.

Цветовое пространство *CIE Lab* имеет ряд преимуществ таких как: независимость от устройств, более широкий цветовой охват и точное описание цвета, благодаря чему можно подобрать диапазон значений каналов для адекватного определения зелёного цвета и его оттенков, присутствующих на большинстве изображений здоровых листьев. Если цвет листа лежит в этом диапазоне, то он окрашивается в белый.

К сожалению, подобрать оптимальный диапазон значений каналов L , a и b , не представляется возможным, так как имеется вариативность в освещении листьев при съёмке, из-за которой цвет отдельных областей листа может меняться (засветка или тень). Этот факт приводит к появлению «пропусков» в бинарной маске. Чтобы убрать появившиеся «пропуски» в масках был применён контурный анализ. При контурном анализе с использованием маски, полученной на предыдущем шаге, выполняется поиск описывающего лист контура. Затем область под этим контуром окрашивается в белый цвет, тем самым формируя итоговую бинарную маску здорового листа. На рисунке 3 показан пример полученных бинарных масок.



Рисунок 3. Пример бинарных масок: левое изображение – исходное изображение, среднее – изображение после поканальной бинаризации с «пропусками», правое – итоговая бинарная маска

В итоге «здоровая» выборка содержала 13922 изображения здоровых листьев, каждому из которых соответствует своя бинарная маска. Для формирования выборки использовались оригинальные изображения, то есть те изображения, где присутствует фон.

2.4. Алгоритм создания бинарных масок для «больной» выборки

Алгоритм формирования бинарных масок больных областей растений для соответствующей выборки частично схож с алгоритмом формирования масок здоровых областей, но имеет ряд отличий и состоит из следующих шагов:

- (1) Перевод исходных изображений в цветовое пространство CIE Lab, определение диапазона значений для каналов a и b, и поканальная бинаризация. Диапазон значений каналов подбирался таким образом, чтобы фиксировался не только очаг заболевания, но и область листа, подверженная заболеванию и изменившая свой цвет.
- (2) Применение морфологических операций «открытия» и «закрытия» со структурным элементом (ядром) эллиптической формы размера 5x5 (рисунок 4).



Рисунок 4. Демонстрация морфологических операций (слева направо): бинарная маска до применения морфологических операций; бинарная маска после операции «открытия»; бинарная маска после операции «закрытия»

Как и в случае создания масок здоровых областей растений, при генерации масок больных областей использовалось цветовое пространство CIE Lab. Однако, в данном случае, диапазон значений каналов был подобран из расчёта цвета поражённых/областей растений и очагов заболеваний. В свою очередь, применение морфологических операций помогло избавиться от шума и «пропусков» в получившихся бинарных масках.

Объективно оценить «верность» полученных бинарных масок возможно только при использовании ручной разметки данных и расчёта *индекса Жаккара*^{URL}. Ручная разметка такого объёма данных весьма трудоёмка, а из-за специфики данных (не всегда удаётся определить чёткие границы областей заболевания) требуется проводить валидацию размеченных данных по разметчикам. Очевидно, что при автоматизированном создании бинарных масок программным способом нельзя гарантировать их 100% точность. Однако, благодаря такому подходу удалось достичь компромисса между затраченными ресурсами (человеко-часы) и их достоверностью.

Таким образом, для всех изображений «больной» выборки, были сгенерированы бинарные маски с использованием алгоритма, описанного выше. Затем, в целях расширения обучающего набора данных, все

оригинальные изображения были вертикально отражены, чтобы повторно пройти процедуру генерации бинарных масок. Итого «большая» выборка насчитывала 42731 изображение, каждому из которых соответствовала своя бинарная маска. Для формирования выборки использовались оригинальные изображения, то есть те изображения, где присутствует фон.

2.5. Описание обучающих наборов данных

Используемые выборки данных («здоровая» и «большая») по сути являются самодостаточными наборами данных для определения здоровых и больных областей листьев соответственно. В связи с этим каждый из наборов был дополнительно разделён на обучающую, тестовую и вариационную выборки.

В таблице 1 показана статистика используемых наборов данных.

Таблица 1. Статистика для наборов данных с изображениями здоровых и больных листьев растений

Набор данных	Обучающая выборка	Тестовая выборка	Валидационная выборка
Изображения здоровых листьев растений	11137	1392	1393
Изображения листьев растений с заболеваниями	34184	4273	4274

На рисунке 5 и рисунке 6 показаны примеры изображений и масок из полученных обучающих наборов данных.

Общая логика работы с исходным набором данных была следующая:

- (1) Формирование двух отдельных выборок – «здоровой» и «больной» из всех классов растений исходного набора.
- (2) Создание бинарных масок для «здоровой» (п. 2.3) и «больной» (п. 2.4) выборок. Формирование отдельных наборов данных для определения здоровых и больных областей листьев растений, состоящих из промаркированных данных (полноценное RGB-изображение + маска).
- (3) Формирование выборок для обучения теста и валидации из полученных на предыдущем шаге наборов данных.

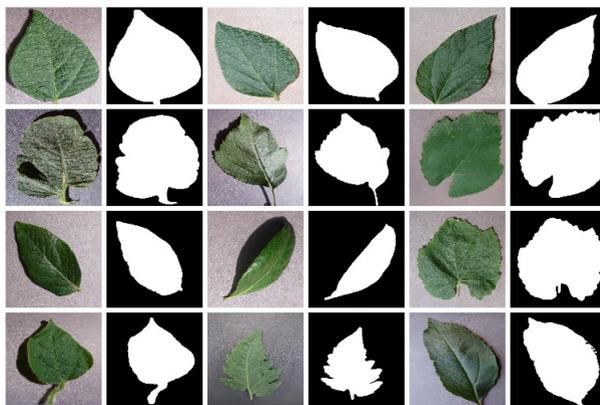


Рисунок 5. Пример изображений и бинарных масок набора данных для определения здоровых областей листа растения

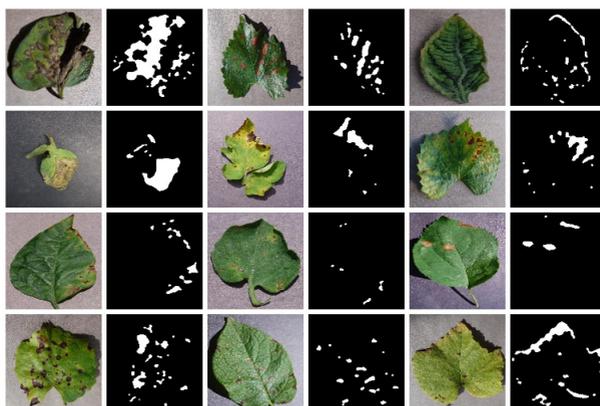


Рисунок 6. Пример изображений и бинарных масок набора данных для определения больных областей листа растения

3. Обучение нейронной сети

3.1. Архитектура модели

В качестве нейронной сети была выбрана модель сегментации изображений на основе PyTorch – *Segmentation Models Pytorch*^{URL} и фреймворка глубокого обучения для предварительной подготовки, настройки и развертывания моделей ИИ – *PyTorch Lightning*^{URL}. Выбор данного инструментария

был обусловлен тем, что библиотека Segmentation Models Pytorch является API высокого уровня и содержит 9 архитектур моделей для бинарной и многоклассовой сегментации, а также 124 энкодера. В свою очередь PyTorch Lightning предлагает широкие возможности по настройке и развёртыванию моделей нейронных сетей. Также преимуществом данных инструментов является наличие документации и поддержка сообщества.

Базовой нейронной сетью (backbone) для выбранной модели является архитектура FPN [26], тогда как архитектура ResNet-34 [27] используется как энкодер. Структура используемой нейросетевой модели представлена на рисунке 7.

Layer (type:depth-idx)	Output Shape
LeafModel	[1, 1, 256, 256]
FPN: 1-1	[1, 1, 256, 256]
— ResNetEncoder: 2-1	[1, 3, 256, 256]
— Conv2d: 3-1	[1, 64, 128, 128]
— BatchNorm2d: 3-2	[1, 64, 128, 128]
— ReLU: 3-3	[1, 64, 128, 128]
— MaxPool2d: 3-4	[1, 64, 64, 64]
— Sequential: 3-5	[1, 64, 64, 64]
— Sequential: 3-6	[1, 128, 32, 32]
— Sequential: 3-7	[1, 256, 16, 16]
— Sequential: 3-8	[1, 512, 8, 8]
— FPNDecoder: 2-2	[1, 128, 64, 64]
— Conv2d: 3-9	[1, 256, 8, 8]
— FPNBlock: 3-10	[1, 256, 16, 16]
— FPNBlock: 3-11	[1, 256, 32, 32]
— FPNBlock: 3-12	[1, 256, 64, 64]
— ModuleList: 3-13	-
— MergeBlock: 3-14	[1, 128, 64, 64]
— Dropout2d: 3-15	[1, 128, 64, 64]
— SegmentationHead: 2-3	[1, 1, 256, 256]
— Conv2d: 3-16	[1, 1, 64, 64]
— UpsamplingBilinear2d: 3-17	[1, 1, 256, 256]
— Activation: 3-18	[1, 1, 256, 256]
<hr/>	
Total params: 23,155,393	
Trainable params: 23,155,393	
Non-trainable params: 0	
Total mult-adds (Units.GIGABYTES): 6.86	
<hr/>	
Input size (MB): 0.79	
Forward/backward pass size (MB): 105.15	
Params size (MB): 92.62	
Estimated Total Size (MB): 198.56	

РИСУНОК 7. Структура используемой нейросетевой модели

3.2. Входные данные, обучение и тест модели

В данном случае применяется бинарная сегментация, то есть сегментация на два класса: фон и целевой объект. На вход ИНС подавались цветные, полноценные RGB-изображения с метками целевых областей (бинарными масками), то есть подавалось некоторое количество пар: изображение + маска. Количество одновременно поданных на вход пар задавалось параметром размера пакета/партии (`batch_size`) и составило 64 шт. Результатом работы нейросетевой модели также является бинарная маска, где чёрными пикселями обозначается фон, а белыми – целевые объекты. В зависимости от набора данных, целевым объектом является либо здоровая область листа, либо больная.

Стоит отметить, что обучение модели происходило отдельно для поиска здоровых и больных областей листовой пластины растения. В итоговом варианте применяются два набора весов для здоровых и больных областей соответственно. Решение об использовании одной модели с двумя наборами весов обусловлено тем, что так модель не будет сегментировать ничего кроме целевого класса (здоровые или больные области листа). Также упрощается сама задача сегментации, так как пространство признаков становится меньше благодаря меньшей вариации текстур.

Обучение модели для поиска здоровых областей листьев происходило на протяжении 5 эпох, а для поиска больных – 15 эпох. Данное количество эпох обучения модели является результатом проведённых предварительных экспериментов, в процессе которых было выяснено, что дальнейшее увеличение количества эпох уже не ведёт к какому-либо значительному увеличению показателей метрик. В роли метрик точности используемой модели выступали показатели IoU_{URL} и $F1_{URL}$. Метрики рассчитывались двумя способами:

- Суммирование истинно-положительных (TP), ложно-положительных (FP), ложно-отрицательных (FN) и истинно-отрицательных (TN) пикселей по всем изображениям и всем классам набора данных, а затем вычисление оценки.
- Вычисление оценки для каждого изображения и для каждого класса на этом изображении отдельно, затем вычисление средней оценки для каждого изображения по меткам и средней оценки изображений по набору данных.

Полученные в процессе обучения графики изменения значений метрик IoU и F1 в задаче определения здоровых областей листьев растений, показаны на рисунках 8 и 9.

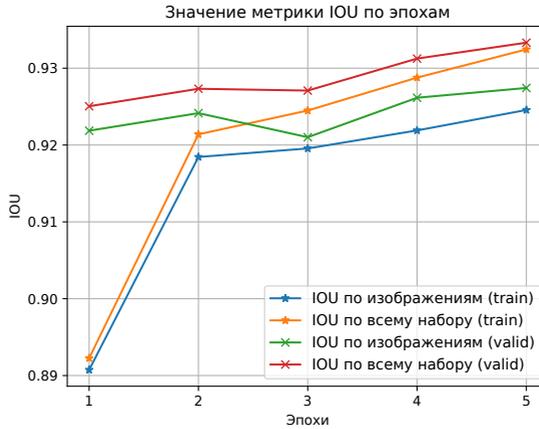


Рисунок 8. График изменения значений метрики IoU для определения здоровых областей листьев

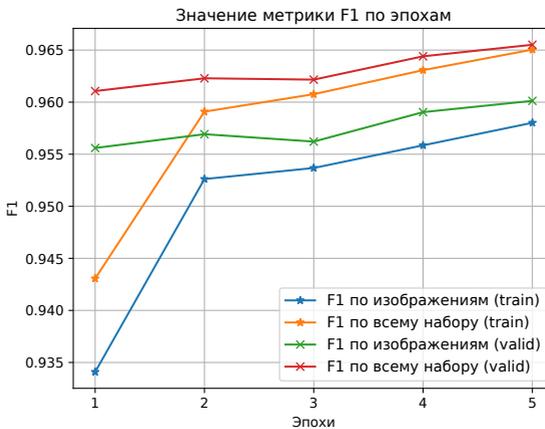


Рисунок 9. График изменения значений метрики F1 для определения здоровых областей листьев

Графики тех же метрик только в задаче определения больных областей листьев растений, представлены на рисунках 10 и 11.

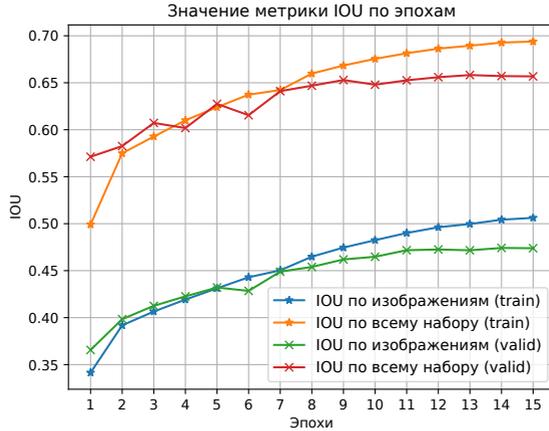


Рисунок 10. График изменения значений метрики IoU для определения больных областей листьев

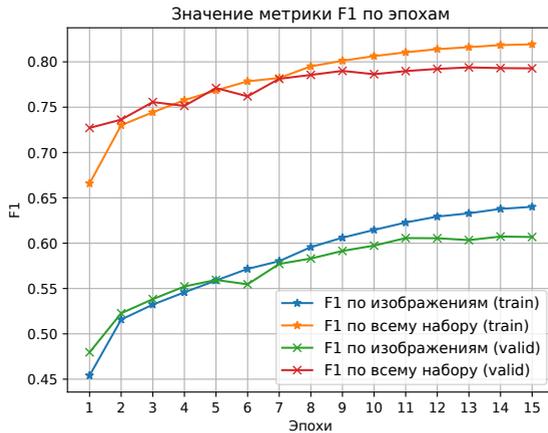


Рисунок 11. График изменения значений метрики F1 для определения больных областей листьев

По окончании обучения модели было выполнено её тестирование

с использованием изображений из валидационной выборки. На рисунках 12 и 13 показаны примеры полученных и эталонных масок для здоровых и больных областей листьев соответственно.

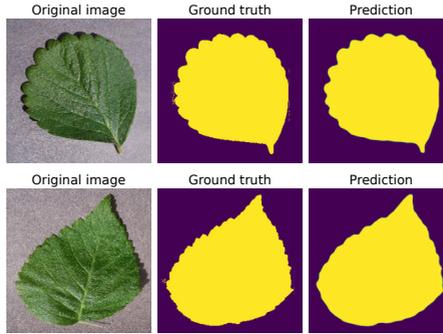


Рисунок 12. Образцы бинарных масок здоровых областей листьев: Ground truth – эталонная маска из набора данных; Prediction – полученная от модели маска

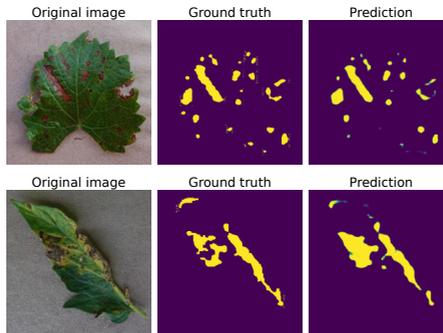


Рисунок 13. Образцы бинарных масок больных областей листьев: Ground truth – эталонная маска из набора данных; Prediction – полученная от модели маска

Итоговая точность определения здоровых областей листьев растений составила 93.2% по IoU и 96.5% по F1. Для больных областей листьев удалось достичь точности в 65.1% по IoU и 78.9% по F1.

Также были проведены эксперименты с обучением ИНС на конкретных видах растений из используемого набора данных. При обучении использовались изображения здоровых листьев растений, подготовка

бинарных масок была выполнена в соответствии с п. 2.3 настоящей статьи для «здоровой» выборки. Архитектура сети и параметры обучения были идентичны тем, что использовались при обучении на «здоровой» выборке. В таблице 2 представлен результат обучения ИНС на разных видах растений.

Таблица 2. Значение метрик IoU и F1 на тестовых выборках различных растений после обучения ИНС

Растение	IoU, %	F1, %
Apple	87.8	93.5
Blueberry	93.2	96.4
Cherry	92.6	96.1
Grape	88.8	94.1
Peach	80.9	89.4
Pepper	93.6	96.7
Potato	86.8	92.9
Raspberry	88.7	94.0
Soybean	92.9	96.3
Strawberry	91.6	95.6
Tomato	91.7	95.6
Среднее	89.8	94.6

Результаты проведённых экспериментов показали, что модель, обученная на «здоровой» выборке, превосходит по точности модели обученные на отдельных растениях в задаче определения здоровых областей листьев растений. Можно предположить, что для «больной» выборки был бы получен аналогичный результат, так как архитектура сети и параметры обучения остались неизменны.

4. Инференс модели и анализ результатов

После завершения процедур обучения и тестирования был выполнен инференс модели на тестовой выборке данных, результатом которого стали бинарные маски здоровых и больных областей листа растения. С их помощью можно рассчитать занимаемую ими площадь путём подсчёта количества принадлежащих им пикселей белого цвета. Сумма рассчитанных площадей в пикселях будет отражать общую площадь наблюдаемой листовой пластины растений. Таким образом, появляется возможность определить процентное отношение площадей, занимаемых здоровыми и больными областями относительно общей площади листа.

Далее определяется отношение процентных долей здоровой и больной области листа растения, которое по сути представляет собой разность

между этими значениями. Для простоты восприятия полученная разность делиться на 100, чтобы в итоге диапазон значений лежал в пределах от 1.0 до -1.0. Дабы не оперировать понятием «разность между процентными долями здоровой и больной областей» было решено обозначить полученное значение как индекс «здоровья» (*heal_index*) листа. Значение индекса «здоровья», близкое к 1.0 (рисунок 14), характерно для практически

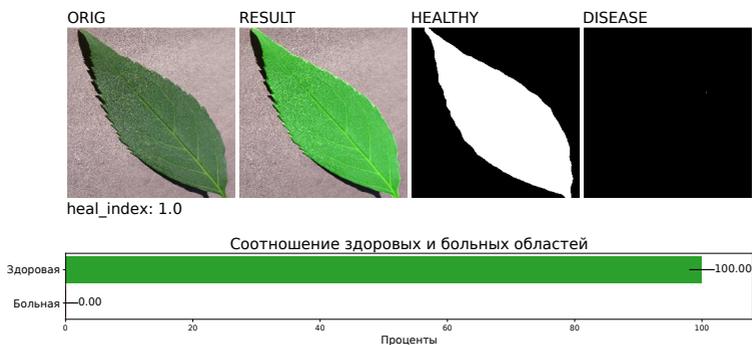


Рисунок 14. Пример здорового листа. Сверху: оригинальное изображение; сегментированное изображение; маска здоровой области; маска больной области. Снизу: столбец зелёного цвета – процентная доля здоровой области; красного цвета – процентная доля больной области

полностью здоровых с визуальной точки зрения листьев растений. Противоположенное значение, близкое к -1.0 (рисунок 15) свидетельствует

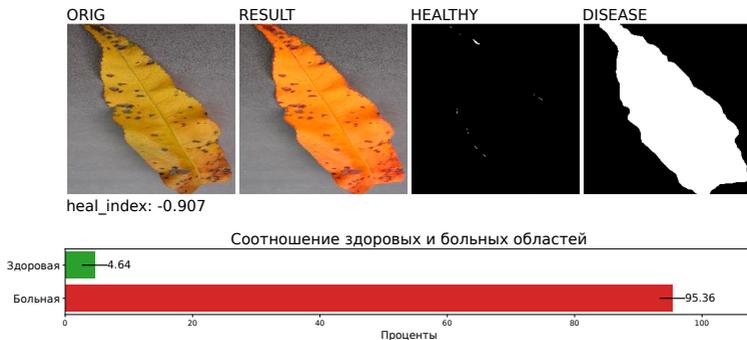


Рисунок 15. Пример полностью больного листа

о том, что наблюдаемый лист растения практически полностью поражён

заболеванием.

Следует отметить то, что значения индекса «здоровья» меньше нуля как правило указывают на то, что площадь больной области листа растения в определённой степени преобладает над здоровой (рисунок 16).

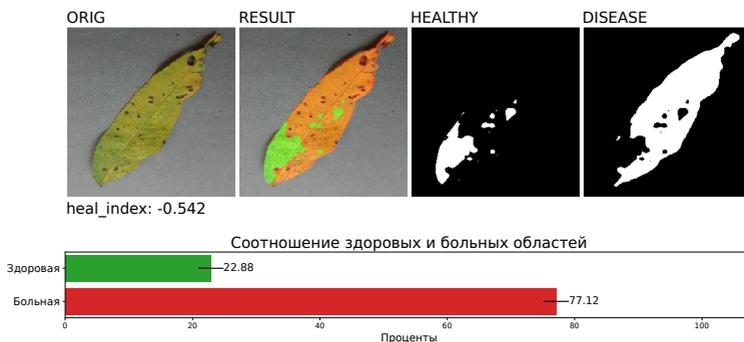


Рисунок 16. Пример больного листа

Опытным путём было выяснено, что если значение индекса «здоровья» больше 0.9, то с визуальной точки зрения наблюдаемый лист растения можно считать относительно здоровым (рисунок 17).

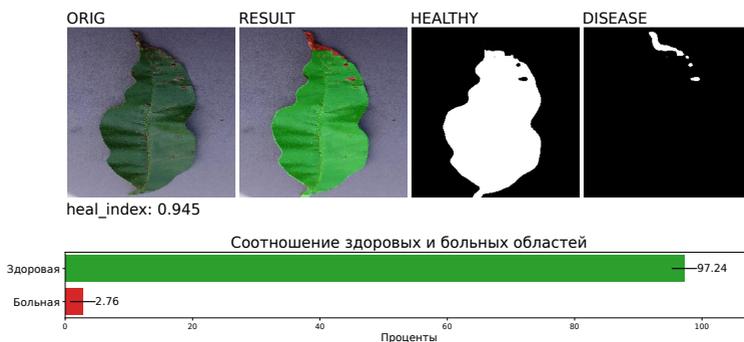


Рисунок 17. Пример относительно здорового листа

Если же значение индекса «здоровья» лежит в диапазоне от 0.9 до 0.5 (рисунок 18), то считается, что наблюдаемый лист уже имеет явные признаки заболевания, и чем ниже индекс, тем выше степень заболевания листа.

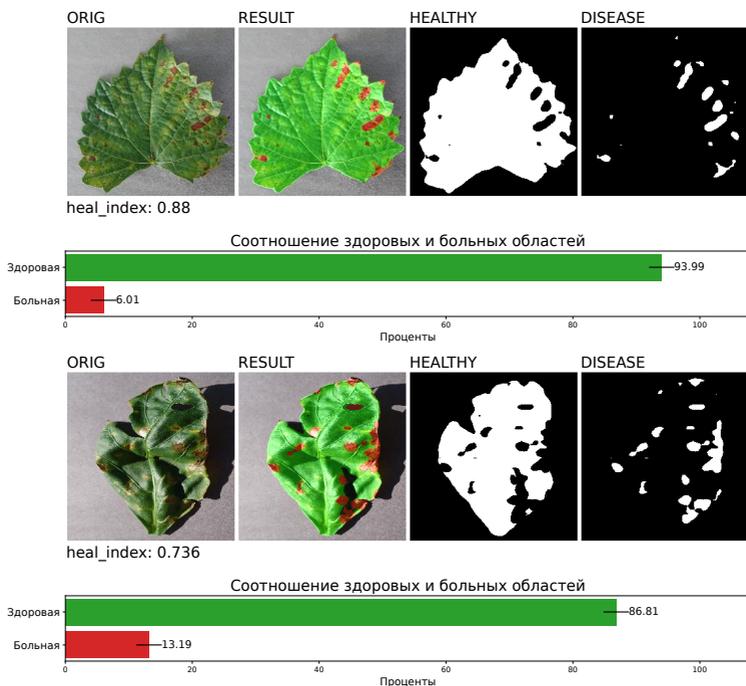


Рисунок 18. Примеры листьев с разной степенью заболевания

Заключение

В результате проведённых исследований был получен метод нейросетевого определения здоровых и больных областей листьев растений с вычислением отношения их процентных долей – индекса «здоровья». Анализ полученных данных показал, что значение индекса «здоровья» в определённой мере отражает степень заболевания листа. Так, например, здоровые листья имеют значения индекса более 0.9. Если индекс лежит в диапазоне от 0.9 до 0.5, то это указывает на то, что лист растения имеет явные признаки заболевания и чем ниже значение индекса, тем больше степень заболевания, с визуальной точки зрения. В случаях дальнейшего уменьшения индекса, а также при его отрицательных значениях наблюдаемый лист растения уже имеет крайне высокую степень заболевания.

Благодаря использованию технологий нейронных сетей удалось сформировать собирательный образ здоровых и больных областей листьев

растений в независимости от их цвета, текстуры и формы, что позволило анализировать различные виды растений и типы заболеваний.

Точность определения здоровых областей листовых платин растений, составила 96.5% по метрике F1. Относительно низкую точность в 78.9% по F1 удалось достичь при определении больных областей растений. Такая разница показателей точности объясняется различием в корректности/достоверности эталонных бинарных масок здоровых и больных областей. Для повышения достоверности бинарных масок необходимо использовать ручную разметку данных, или, по крайней мере, провести процедуру ручной корректировки масок. Однако, подход с автоматизированным формированием бинарных масок, используемый в настоящем исследовании, позволил существенно сократить время на их создание. Анализируя полученные результаты, можно сделать вывод о том, что создание бинарных масок без ручной разметки имеет место быть в задачах не требующих высоких показателей точности. Так, например, в рассматриваемой задаче по определению соотношения здоровой площади листа к больной, достаточно идентифицировать больную область листа, чтобы считать растение больным. В будущих работах планируется повысить точность определения больных областей листьев растений, а также добавить функционал определения типа заболевания.

Подводя итог, можно сказать, что в ходе выполнения исследования, представленного в настоящей статье, все поставленные задачи были успешно решены.

Список использованных источников

- [1] Fenu G., Mallocci F. *DiaMOS Plant: A dataset for diagnosis and monitoring plant disease* // *Agronomy*.– 2021.– No. 11.– id. 2107.– 12 pp.  [↑72](#)
- [2] Liu X., Min W., Mei S., Wang L., Jiang S. *Plant disease recognition: A large-scale benchmark dataset and a visual region and loss reweighting approach* // *IEEE Transactions on Image Processing*.– 2021.– Vol. 30.– Pp. 2003–2015.  [↑72](#)
- [3] Ahmad A., Saraswat D., El Gamal A. *A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools* // *Smart Agricultural Technology*.– February 2023.– Vol. 3.– id. 100083.  [↑73](#)
- [4] Shoaib M., Shah B., EI-Sappagh S., Ali A., Ullah A., Alenezi F., Gechev T., Hussain T., Ali F. *An advanced deep learning models-based plant disease detection: A review of recent research* // *Frontiers in Plant Science*.– 2023.– Vol. 14.– id. 1158933.– 22 pp.  [↑73](#)
- [5] O’Shea K., Nash R. *An introduction to convolutional neural networks*.– 2015.– 11 pp. [arXiv:1511.08458](#)[~][cs.NE] [↑73](#)
- [6] Schmidt R. *Recurrent Neural Networks (RNNs): A gentle introduction and overview*.– 2019.– 16 pp. [arXiv:1912.05911](#)[~][cs.LG]  [↑73](#)

- [7] Cohen G., Giryres R. *Generative adversarial networks*.— 2022.— 28 pp. arXiv:2203.00667~[cs.CV]  ↑73
- [8] Zhuang F., Qi Z., Duan K., Xi D., Zhu Y., Zhu H., Xiong H., He Q. *A comprehensive survey on transfer learning* // Proceedings of the IEEE.— 2021.— Vol. **109**.— No. 1.— Pp. 43–76.  ↑73
- [9] Ronneberger O., Fischer P., Brox T. *U-Net: Convolutional networks for biomedical image segmentation*, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 (Munich, Germany, October 5–9, 2015), Lecture Notes in Computer Science (LNIP).— vol. **9351**, Cham: Springer.— 2015.— ISBN 978-3-319-24573-7.— Pp. 234–241.  ↑73
- [10] He K., Gkioxari G., Dollár P., Girshick R. *Mask R-CNN* // *2017 IEEE International Conference on Computer Vision (ICCV)* (Venice, Italy, October 22–29, 2017).— 2017.— ISBN 9781538610336.— Pp. 2980–2988.  ↑73
- [11] Ahmad N., Asif H., Saleem G., Younus M., Anwar S., Anjum M. *Leaf image-based plant disease identification using color and texture features* // Wireless Personal Communications.— 2021.— Vol. **121**.— Pp. 1139–1168.  ↑73
- [12] V. B. Sebastian, Unnikrishnan A., Balakrishnan K. *Grey level co-occurrence matrices: Generalisation and some new features* // International Journal of Computer Science, Engineering and Information Technology.— 2012.— Vol. **2**.— No. 2.— Pp. 151–157.  ↑73
- [13] Gajjar R., Gajjar N., Thakor V., Patel N., Ruparelia S. *Real-time detection and identification of plant leaf diseases using convolutional neural networks on an embedded platform* // The Visual Computer.— 2021.— Vol. **38**.— No. 8.— Pp. 2923–2938.  ↑73
- [14] Kumari P., Ranjan P., Srivastava P. *Classification and assessment of visual content of medicinal plants using CNN* // *Handbook of Research on Innovative Approaches to Information Technology in Library and Information Science*.— IGI Global.— 2024.— ISBN 9798369308073.— Pp. 126–147.  ↑74
- [15] He W., Zhou T., Xiang Y., Lin Y., Hu J., Bao R. *Deep learning in image classification: Evaluating VGG19's performance on complex visual data*.— 2024.— 5 pp. arXiv:2412.20345~[cs.CV]  ↑74
- [16] Vishnoi V., Kumar K., Kumar B., Mohan S., Khan A. *Detection of apple plant diseases using leaf images through convolutional neural network* // IEEE Access.— 2023.— Vol. **11**.— Pp. 6594–6609.  ↑74
- [17] Sharma S., Sharma G., Menghani E. *Tomato plant disease detection with pretrained CNNs* // *Artificial Intelligence in Medicine and Healthcare*, 1st ed.— CRC Press.— 2025.— ISBN 9781003508595.— Pp. 67–85.  ↑74
- [18] Xu P., Fang N., Liu N., Lin F., Yang S., Ning J. *Visual recognition of cherry tomatoes in plant factory based on improved deep instance segmentation* // Computers and Electronics in Agriculture.— June 2022.— Vol. **197**.— id. 106991.  ↑74
- [19] Tang W., Sun J., Wang S., Zhang Y. *Review of AlexNet for medical image classification* // EAI Endorsed Transactions on e-Learning.— December 2023.— Vol. **9**.— 13 pp.  ↑74

- [20] Iandola F., Han S., Moskewicz M., Ashraf K., Dally W., Keutzer K. *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5MB model size.*– 2016.– 13 pp. arXiv:1602.07360 [cs.CV] doi ↑74
- [21] Ren S., He K., Girshick R., Sun J. *Faster R-CNN: Towards real-time object detection with region proposal networks.*– 2016.– 14 pp. arXiv:1506.01497 [cs.CV] ↑74
- [22] Yag I., Altan A. *Artificial intelligence-based robust hybrid algorithm design and implementation for real-time detection of plant diseases in agricultural environments // Biology.*– 2022.– Vol. **11**.– No. 12.– id. 1732.– 30 pp. doi ↑74
- [23] Yang X. *Flower pollination algorithm for global optimization*, Unconventional Computation and Natural Computation (UCNC 2012) (Orléans, France, September 3–7, 2012), Lecture Notes in Computer Science.– vol. **7445**, Berlin–Heidelberg: Springer.– 2012.– ISBN 978-3-642-32893-0.– Pp. 240–249. doi ↑74
- [24] Pal A. *AgriDet: Plant Leaf Disease severity classification using agriculture detection framework // Engineering Applications of Artificial Intelligence.*– March 2023.– Vol. **119**.– id. 105754. doi ↑74
- [25] Di Mascio T., Fantozzi P., Laura L., Ruggetti V. *Age and gender (face) recognition: A brief survey*, Methodologies and Intelligent Systems for Technology Enhanced Learning, 11th International Conference (MIS4TEL 2021) (Salamanca, Spain, September 6–8, 2021), Lecture Notes in Networks and Systems.– vol. **326**, Cham: Springer.– 2022.– ISBN 978-3-030-86617-4.– Pp. 105–113. doi ↑75
- [26] Lin T., Dollar P., Girshick R., He K., Hariharan B., Belongie S. *Feature pyramid networks for object detection // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI, USA, July 21–26, 2017).– 2017.– ISBN 9781538604588.– Pp. 936–944. doi ↑84
- [27] He K., Zhang X., Ren S., Sun J. *Deep residual learning for image recognition // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV, USA, June 27–30, 2016).– 2016.– ISBN 9781467388528.– Pp. 770–778. doi ↑84

Поступила в редакцию	30.07.2025;
одобрена после рецензирования	07.08.2025;
принята к публикации	19.08.2025;
опубликована онлайн	25.08.2025.

Рекомендовал к публикации

к.т.н. Е. П. Куршев

Информация об авторах:



Александр Владимирович Смирнов

Младший научный сотрудник Лаборатории методов обработки и анализа изображений, Институт Программных Систем имени А. К. Айламазяна РАН. Научные интересы: компьютерное зрение; нейронные сети; робототехника; автоматизация и управление



0000-0002-7104-1462

e-mail:



Игорь Петрович Тищенко

Кандидат технических наук, ИО директора, Институт Программных Систем имени А. К. Айламазяна РАН. Научные интересы: компьютерное зрение; нейронные сети; робототехника; автоматизация и управление



0000-0002-0369-0524

e-mail:

Вклад авторов: *А. В. Смирнов* – 90% (идея, методология, программное обеспечение, валидация, формальный анализ, расследование, написание черновой версии, доработка и редактирование, визуализация, наставничество); *И. П. Тищенко* – 10% (доработка и редактирование, наставничество, администрирование, финансирование).

Декларация об отсутствии личной заинтересованности: *благополучие авторов не зависит от результатов исследования.*

UDC 004.93'11

 10.25209/2079-3316-2025-16-3-69-97

Identifying healthy and diseased areas of plant leaves using neural networks

Alexander Vladimirovich **Smirnov**¹, Igor Petrovich **Tishchenko**²

^{1,2}Ailamazyan Program Systems Institute of RAS, Ves'kovo, Russia

¹ asmirnov_1991@mail.ru

Abstract. This paper presents a study aimed at developing a neural network method for detecting healthy and diseased areas of plant leaves based on their images and calculating the ratio of their areas. The basic network of the FPN architecture with an encoder in the form of the ResNet-34 architecture was used as a neural network model. To train the ANN, binary masks of target areas of plant leaves were used as labels; they were obtained programmatically without manual marking. Due to this, it was possible to achieve a reasonable compromise between the resources required to create masks and their accuracy. When training the neural network model, the accuracy of 96.5% and 78.9% was achieved according to the F1 metric for determining healthy and diseased areas, respectively. Next, the model was inferred, as a result of which the "health" index was calculated for each of the studied leaf images. In the context of the problems being solved, the "health" index is the difference between the percentages of healthy and diseased areas, which can be used to assess the severity of the disease, as well as to monitor the dynamics of the disease as an indicator of the effectiveness of the drugs or care methods used. The scientific novelty of the presented study lies in the creation of a method for automatically determining the ratio of healthy and diseased leaf areas, which combines modern computer vision technologies, machine learning and practical applicability for agronomy and plant growing. (*In Russian*).

Key words and phrases: neural network analysis, health index, healthy leaf area, diseased leaf area, model

2020 *Mathematics Subject Classification:* 68T10; 68T45, 68T07

For citation: Alexander V. Smirnov, Igor P. Tishchenko. *Identifying healthy and diseased areas of plant leaves using neural networks*. Program Systems: Theory and Applications, 2025, **16**:3(66), pp. 69–97. (*In Russ.*).

https://psta.psiras.ru/read/psta2025_3_69-97.pdf

References

- [1] G. Fenu, F. Mallocci. “DiaMOS Plant: A dataset for diagnosis and monitoring plant disease”, *Agronomy*, 2021, no. 11, id. 2107, 12 pp. [doi](#)
- [2] X. Liu, W. Min, S. Mei, L. Wang, S. Jiang. “Plant disease recognition: A large-scale benchmark dataset and a visual region and loss reweighting approach”, *IEEE Transactions on Image Processing*, **30** (2021), pp. 2003–2015. [doi](#)
- [3] A. Ahmad, D. Saraswat, El Gamal A.. “A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools”, *Smart Agricultural Technology*, **3** (February 2023), id. 100083. [doi](#)
- [4] M. Shoaib, B. Shah, S. El-Sappagh, A. Ali, A. Ullah, F. Alenezi, T. Gechev, T. Hussain, F. Ali. “An advanced deep learning models-based plant disease detection: A review of recent research”, *Frontiers in Plant Science*, **14** (2023), id. 1158933, 22 pp. [doi](#)
- [5] K. O’Shea, R. Nash. *An introduction to convolutional neural networks*, 2015, 11 pp. [arXiv:1511.08458](#) [cs.NE]
- [6] R. Schmidt. *Recurrent Neural Networks (RNNs): A gentle introduction and overview*, 2019, 16 pp. [arXiv:1912.05911](#) [cs.LG] [doi](#)
- [7] G. Cohen, R. Giryes. *Generative adversarial networks*, 2022, 28 pp. [arXiv:2203.00667](#) [cs.CV] [doi](#)
- [8] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He. “A comprehensive survey on transfer learning”, *Proceedings of the IEEE*, **109**:1 (2021), pp. 43–76. [doi](#)
- [9] O. Ronneberger, P. Fischer, T. Brox. “U-Net: Convolutional networks for biomedical image segmentation”, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 (Munich, Germany, October 5–9, 2015), Lecture Notes in Computer Science (LNIP), vol. **9351**, Springer, Cham, 2015, ISBN 978-3-319-24573-7, pp. 234–241. [doi](#)
- [10] K. He, G. Gkioxari, P. Dollár, R. Girshick. “Mask R-CNN”, *2017 IEEE International Conference on Computer Vision (ICCV)* (Venice, Italy, October 22–29, 2017), 2017, ISBN 9781538610336, pp. 2980–2988. [doi](#)
- [11] N. Ahmad, H. Asif, G. Saleem, M. Younus, S. Anwar, M. Anjum. “Leaf image-based plant disease identification using color and texture features”, *Wireless Personal Communications*, **121** (2021), pp. 1139–1168. [doi](#)
- [12] V. B. Sebastian, A. Unnikrishnan, K. Balakrishnan. “Grey level co-occurrence matrices: Generalisation and some new features”, *International Journal of Computer Science, Engineering and Information Technology*, **2**:2 (2012), pp. 151–157. [doi](#)
- [13] R. Gajjar, N. Gajjar, V. Thakor, N. Patel, S. Ruparelia. “Real-time detection and identification of plant leaf diseases using convolutional neural networks on an embedded platform”, *The Visual Computer*, **38**:8 (2021), pp. 2923–2938. [doi](#)
- [14] P. Kumari, P. Ranjan, P. Srivastava. “Classification and assessment of visual content of medicinal plants using CNN”, *Handbook of Research on Innovative*

- Approaches to Information Technology in Library and Information Science*, IGI Global, 2024, ISBN 9798369308073, pp. 126–147. [doi](#)
- [15] W. He, T. Zhou, Y. Xiang, Y. Lin, J. Hu, R. Bao. *Deep learning in image classification: Evaluating VGG19's performance on complex visual data*, 2024, 5 pp. arXiv:2412.20345~[cs.CV] [doi](#)
- [16] V. Vishnoi, K. Kumar, B. Kumar, S. Mohan, A. Khan. "Detection of apple plant diseases using leaf images through convolutional neural network", *IEEE Access*, **11** (2023), pp. 6594-6609. [doi](#)
- [17] S. Sharma, G. Sharma, E. Menghani. "Tomato plant disease detection with pretrained CNNs", *Artificial Intelligence in Medicine and Healthcare*, 1st ed., CRC Press, 2025, ISBN 9781003508595, pp. 67–85. [doi](#)
- [18] P. Xu, N. Fang, N. Liu, F. Lin, S. Yang, J. Ning. "Visual recognition of cherry tomatoes in plant factory based on improved deep instance segmentation", *Computers and Electronics in Agriculture*, **197** (June 2022), id. 106991. [doi](#)
- [19] W. Tang, J. Sun, S. Wang, Y. Zhang. "Review of AlexNet for medical image classification", *EAI Endorsed Transactions on e-Learning*, **9** (December 2023), 13 pp. [doi](#)
- [20] F. Iandola, S. Han, M. Moskewicz, K. Ashraf, W. Dally, K. Keutzer. *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5MB model size*, 2016, 13 pp. arXiv:1602.07360~[cs.CV] [doi](#)
- [21] S. Ren, K. He, R. Girshick, J. Sun. *Faster R-CNN: Towards real-time object detection with region proposal networks*, 2016, 14 pp. arXiv:1506.01497~[cs.CV]
- [22] I. Yag, A. Altan. "Artificial intelligence-based robust hybrid algorithm design and implementation for real-time detection of plant diseases in agricultural environments", *Biology*, **11:12** (2022), id. 1732, 30 pp. [doi](#)
- [23] X. Yang. "Flower pollination algorithm for global optimization", Unconventional Computation and Natural Computation (UCNC 2012) (Orléans, France, September 3–7, 2012), Lecture Notes in Computer Science, vol. **7445**, Springer, Berlin–Heidelberg, 2012, ISBN 978-3-642-32893-0, pp. 240–249. [doi](#)
- [24] A. Pal. "AgriDet: Plant Leaf Disease severity classification using agriculture detection framework", *Engineering Applications of Artificial Intelligence*, **119** (March 2023), id. 105754. [doi](#)
- [25] Di Mascio T., P. Fantozzi, L. Laura, V. Rughetti. "Age and gender (face) recognition: A brief survey", Methodologies and Intelligent Systems for Technology Enhanced Learning, 11th International Conference (MIS4TEL 2021) (Salamanca, Spain, September 6–8, 2021), Lecture Notes in Networks and Systems, vol. **326**, Springer, Cham, 2022, ISBN 978-3-030-86617-4, pp. 105–113. [doi](#)
- [26] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie. "Feature pyramid networks for object detection", *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI, USA, July 21–26, 2017), 2017, ISBN 9781538604588, pp. 936–944. [doi](#)
- [27] K. He, X. Zhang, S. Ren, J. Sun. "Deep residual learning for image recognition", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV, USA, June 27–30, 2016), 2016, ISBN 9781467388528, pp. 770–778. [doi](#)

АВТОРСКИЙ УКАЗАТЕЛЬ

Антонов Илья Андреевич

Методы квантового машинного обучения для обнаружения атак на программно-конфигурируемые сети **3**, 19

Гусева Ирина Сергеевна

Метод сильного улучшения управления для неоднородных дискретных систем **23**, 38

Курочкин Илья Ильич

Методы квантового машинного обучения для обнаружения атак на программно-конфигурируемые сети **3**, 19

Подлазов Виктор Сергеевич

Быстродействие и отказоустойчивость идеальной системной сети через дополнительную параллельность **41**, 66

Расина Ирина Викторовна

Метод сильного улучшения управления для неоднородных дискретных систем **23**, 38

Смирнов Александр Владимирович

Определение здоровых и больных областей листьев растений при помощи нейронных сетей **69**, 96

Тищенко Игорь Петрович

Определение здоровых и больных областей листьев растений при помощи нейронных сетей **69**, 96

VOL. 16 ISSUE 3(66) 2025

AUTHOR INDEX

Antonov Il'ya Andreevich
Quantum machine learning methods for intrusion detection in software-defined networks **3, 19**
Guseva Irina Sergeevna
Strong control improvement method for non-homogeneous discrete systems . **23, 38**
Kurochkin Il'ya Il'ich
Quantum machine learning methods for intrusion detection in software-defined networks **3, 19**
Podlazov Viktor Sergeevich
Performance and fault tolerance of the ideal system network via additional parallelity **41, 66**
Rasina Irina Viktorovna
Strong control improvement method for non-homogeneous discrete systems . **23, 38**
Smirnov Alexander Vladimirovich
Identifying healthy and diseased areas of plant leaves using neural networks . **69, 96**
Tishchenko Igor Petrovich
Identifying healthy and diseased areas of plant leaves using neural networks . **69, 96**

VOL. 16 ISSUE 3(66) 2025

CONTENTS

Research Article HARDWARE AND SOFTWARE FOR DISTRIBUTED AND SUPERCOMPUTER SYSTEMS

IL'YA A. ANTONOV[✉], IL'YA I. KUROCHKIN. *Quantum machine learning methods for intrusion detection in software-defined networks (In Russ.)* . 3–19, **20–22**

Research Article OPTIMIZATION METHODS AND CONTROL THEORY

IRINA V. RASINA[✉], IRINA S. GUSEVA. *Strong control improvement method for non-homogeneous discrete systems (In Russ.)* 23–38, **39–40**

Research Article HARDWARE AND SOFTWARE FOR DISTRIBUTED AND SUPERCOMPUTER SYSTEMS

VIKTOR S. PODLAZOV[✉]. *Performance and fault tolerance of the ideal system network via additional parallelity (In Russ.)* 41–65, **66–68**

Research Article ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

ALEXANDER V. SMIRNOV[✉], IGOR P. TISHCHENKO. *Identifying healthy and diseased areas of plant leaves using neural networks (In Russ.)* . 69–94, **95–97**

Author index **102**

Чтобы сменить язык страницы, кликните, пожалуйста, флаг в верхнем углу

Авторский указатель 101

Содержание 2