

На правах рукописи



**Яковлев Константин Сергеевич**

**Методы и алгоритмы эвристического поиска на  
графах регулярной декомпозиции в задачах  
планирования траекторий мобильных роботов**

Специальность 1.2.1 —  
«Искусственный интеллект и машинное обучение»

**Автореферат**  
диссертации на соискание учёной степени  
доктора физико-математических наук

Москва — 2026

Работа выполнена в Федеральном исследовательском центре «Информатика и управление» Российской академии наук (ФИЦ ИУ РАН) в отделении 7 отделе 71.

Официальные оппоненты: **Скороходов Владимир Александрович**,  
доктор физико-математических наук, доцент,  
Южный федеральный университет,  
профессор

**Жилякова Людмила Юрьевна**,  
доктор физико-математических наук,  
Институт проблем управления им. В.А. Тра-  
пезникова Российской академии наук,  
ведущий научный сотрудник

**Карпов Валерий Эдуардович**,  
доктор технических наук, доцент,  
Национальный исследовательский центр  
«Курчатовский институт»,  
руководитель лаборатории робототехники

Ведущая организация: Федеральное государственное бюджетное  
учреждение науки Институт программных  
систем им. А.К. Айламазяна Российской  
академии наук

Защита состоится «\_\_\_» \_\_\_\_\_ 2026 г. в \_\_\_ часов на заседании  
диссертационного совета 24.1.224.03 при ФИЦ ИУ РАН по адресу: 119333,  
Москва, ул. Вавилова, д. 42.

С диссертацией можно ознакомиться в библиотеке ФИЦ ИУ РАН и  
на официальном сайте [https://www.frccsc.ru/diss-council/00207305/  
diss/list/jakovlev\\_ks](https://www.frccsc.ru/diss-council/00207305/diss/list/jakovlev_ks).

Автореферат разослан «\_\_\_» \_\_\_\_\_ 2026 г.

Ученый секретарь  
диссертационного совета  
24.1.224.03,  
к.т.н.



И.А. Рейер

## Общая характеристика работы

**Актуальность темы исследования.** Задачи планирования, т.е. задачи выбора последовательности действий для достижения поставленных целей на протяжении долгого времени являются одними из центральных задач в искусственном интеллекте – междисциплинарной области науки, ставящей своей целью разработку и исследование искусственных систем и устройств, способных к целенаправленному поведению и разумным рассуждениям<sup>1</sup>. Такой интерес к задачам планирования обусловлен рядом факторов. Во-первых, планирование подразумевает автоматизацию процесса рассуждения о действиях и последствиях действий, и такой тип высокоуровневых рассуждений, по-видимому, является неотъемлемым условием разумного, целенаправленного поведения. Во-вторых, методы решения задачи построения эффективных планов достижения целей находят множество применений на практике. Одним из распространенных и практически важных классов таких задач являются задачи планирования перемещений мобильных роботов, зачастую именуемых агентами. Эти в искусственном интеллекте зачастую формулируются как задачи поиска пути на графах, вложенных в метрическое пространство. Вершинам графов соответствуют различные положения агента, в ребрам – отрезки их соединяющие (следование вдоль которых может быть обеспечено системой управления в автоматическом режиме). Наиболее часто встречающейся в практических приложениях графовой моделью для широкого класса задач планирования траектории является граф, который может быть получен наложением регулярной сетки на рабочее пространство агента. Этот граф представляется в виде совокупности клеток, каждая из которых либо проходима, либо нет для агента. Вершины графа располагаются в центрах проходимых клеток, ребрами соединяются соседние вершины. Будем называть такие графы – графами регулярной декомпозиции (ГРД).

Стандартным подходом к поиска пути на ГРД в искусственном интеллекте является применение алгоритмов семейства  $A^*$ , использующих эвристические функции, оценивающие расстояния между вершинами ГРД, для элиминации перебора. К настоящему моменту известно множество подобных алгоритмов. Однако, большинство практически-мотивированных задач, особенно в контексте планирования для мобильных робототехнических систем, требуют учета дополнительных факторов, обуславливающих

---

<sup>1</sup> Несмотря на то, что существует множество различных определений искусственного интеллекта как области науки, указанное определение представляется, во-первых, справедливым, во-вторых, достаточно лаконичным и при этом ёмким. Его автор – Геннадий Семёнович Осипов, выдающийся советский и российский исследователь в области искусственного интеллекта и когнитивного моделирования, почетный член Европейской ассоциации искусственного интеллекта (ECAI Fellow), президент Российской ассоциации искусственного интеллекта с 1996 по 2022 г.

необходимость разработки и исследования новых техник сокращения перебора при поиске. Среди указанных факторов следует отметить следующие. Зачастую, необходимо планировать траекторию с учетом (прогнозной) динамики окружающей среды. Во многих приложениях встречается много-агентная постановка задачи, следовательно, необходимо осуществлять планирование множества координированных и безопасных траекторий в общей рабочей среде. Многие мобильные робототехнические системы обладают кинематическими ограничениями, то есть не могут мгновенно менять направление движения. Наконец, наиболее распространенные эвристические функции, используемые при поиске путей на ГРД, такие как расстояние Манхэттена или Евклида, не учитывают расположение препятствий в рабочем пространстве агента и их применение при поиске пути в среде с препятствиями, не приводит к существенному сокращению пространства поиска. Возникает потребность в методах автоматического конструирования более информативных эвристических функций, которые учитывали бы расположение препятствий в рабочем пространстве агента и эффективно сокращали пространство поиска.

Таким образом, разработка и исследование новых методов элиминации перебора для создания алгоритмов поиска и планирования, учитывающих важные особенности и ограничения, вытекающие из практических нужд, таких как много-агентная постановка, учет кинематических ограничений, учет динамики окружающей среды, является важной и актуальной научно-технической задачей современности. Именно разработке подобных методов и алгоритмов и посвящена данная работа, что обуславливает её актуальность.

**Степень разработанности темы.** Методы и алгоритмы поиска пути (путей) на графах активно развиваются с середины XX века. Среди основополагающих работ в этой области можно отметить работы Э. Дейкстры, Л. Беллмана, Р. Форда и др. В 70-х годах получили развитие методы эвристического поиска, среди которых основополагающим можно считать алгоритм  $A^*$ , предложенный Хартом, Нильсоном и Рафаэлем в статье. В 80-90х годах XX века в связи с развитием мобильной робототехники активно исследовались методы поиска путей на графах регулярной декомпозиции, которые применялись для автоматического планирования траектории мобильных роботов. В этом контексте стоит отметить таких авторов как Э. Стентс, М. Лихачев, С. Кениг, С. Тран и др., и предложенные ими алгоритмы  $D^*$ ,  $LPA^*$ ,  $ARA^*$ ,  $D^*Lite$  и др. К значимым результатам в области поиска путей на графах регулярной декомпозиции, полученным в начале XXI века, можно отнести алгоритмы поиска с достраиванием ребер графа по ходу поиска (алгоритм  $Theta^*$  и его модификации), алгоритмы отсечения симметрий при поиске (алгоритм  $JPS$  за авторством Д. Харабора). К другим темам, которые получили развитие можно отнести

двунаправленный поиск (работы А. Фелнера, Р. Холте), иерархический поиск (алгоритмы  $HPA^*$ ,  $HGA^*$ ,  $R^*$ ), поиск совокупности неконфликтных путей на графах. Последняя тема является одной из наиболее активно развивающихся в последнее десятилетие (2010 г. - н.в.). Это может быть объяснено двумя факторами. С одной стороны, даже при наличии существенного числа ограничивающих допущений, получение оптимальных решений является NP-трудной задачей. С другой стороны, методы построения совокупности неконфликтных путей высоко востребованы в практических приложениях, в частности в задачах роботизированной складской логистики, транспортных системах и др. В этой области следует отметить таких исследователей как Т. Стенли, А. Фелнер, Р. Штерн, Н. Стюртевант, С. Кёниг и др., и такие методы как  $A^*+ID+OD$ , CBS,  $M^*$ , ICTS и др.

Российские ученые также внесли существенный вклад в теорию графов и разработку методов и алгоритмов напрямую или косвенно связанных с вопросами построения пути/путей на графах. Так, например, известны труды Ерусалимского Я.М., Скороходова В.А., посвященные графам с нестандартной достижимостью, поиском потоков на таких графах, задачам о ресурсных сетях. Последним направлением также занимались и внесли существенный вклад в его развитие Жилякова Л.Ю., Кузнецов О.П. Известны работы Р.П. Агаева и П.Ю. Чеботарёва по алгебраической теории ориентированных графов. Значимый вклад в решение различных вариантов одной из классических графовых задач – задачи коммивояжера, – внесли Ченцов П.А., Ченцов А.Г., Курейчик В.М. Проблематикой случайных графов, задачами раскраски (гипер)графов продуктивно занимались Пузынина С.А., Райгородский А.М., Шабанов Д.А.

**Цели и задачи исследования.** Основной целью работы является повышение эффективности современных систем управления мобильными роботами при решении навигационных задач за счет разработки и исследования новых методов поиска пути (совокупности путей) на графах, вложенных в метрическое пространство, и применимых для решения задач автоматического планирования траектории (совокупности неконфликтных траекторий).

В соответствии с указанной целью задачами исследования являются:

1. Разработка новых методов поиска пути графах регулярной декомпозиции, применимых для планирования траектории в среде со статическими и динамическими препятствиями (траектории движения которых известны); проведение теоретических исследований предложенных методов; разработка алгоритмов их реализующих; проведение экспериментальных исследований разработанных алгоритмов.
2. Разработка новых методов поиска совокупности неконфликтных путей на графах регулярной декомпозиции, применимых для решения задачи централизованного много-агентного планирования

траекторий для группы агентов; проведение теоретических исследований предложенных методов; разработка алгоритмов их реализующих; проведение экспериментальных исследований разработанных алгоритмов, в том числе на реальных робототехнических платформах.

3. Разработка новых методов поиска пути на графах регулярной декомпозиции, косвенно учитывающих кинематические ограничения мобильного агента; проведение теоретических исследований предложенных методов; разработка алгоритмов их реализующих; проведение экспериментальных исследований разработанных алгоритмов.
4. Разработка новых методов автоматического конструирования эвристических функций для решения задач поиска пути на графе регулярной декомпозиции с помощью методов машинного обучения, в частности – глубокого обучения с применением современных нейросетевых архитектур, а также способов интеграции этих функций в алгоритмы эвристического поиска; программная реализация предлагаемых подходов и проведение экспериментальных исследований.

**Научная новизна.** В ходе работы созданы и теоретически обоснованы новые методы и алгоритмы эвристического поиска. В частности:

- Разработан новый метод поиска пути на динамическом графе регулярной декомпозиции, допускающий использование переходов между произвольными вершинами графа, основанный на принципе безопасно-интервального планирования и оригинальном методе обращения направления поиска в сочетании с принципом ленивого поиска (т.е. откладывания наиболее вычислительно ресурсоемких операций до определенного момента). Указанный метод позволяет решить задачу планирования траектории мобильного агента в среде с динамическими препятствиями. В отличие от аналогов разработанный метод допускает перемещение агента в произвольном направлении, что положительно сказывается на качестве отыскиваемых решений (длина траектории и время прибытия в заданное целевое положение сокращаются). Предложены новые алгоритмы поиска субоптимальных решений задачи поиска пути на динамическом графе регулярной декомпозиции.
- Разработаны новые методы и алгоритмы построения совокупности неконфликтных путей на графе регулярной декомпозиции, опирающиеся на принципы конфликтно-ориентированного поиска и приоритизированного планирования. Указанные методы и алгоритмы используют оригинальные техники элиминации перебора, что существенно повышает их эффективность при сохранении теоретических гарантий. Проведено экспериментальное исследование

- предложенных алгоритмов, в т.ч. на реальных робототехнических системах.
- Предложено семейство алгоритмов поиска на графе регулярной декомпозиции, косвенно учитывающих кинематические ограничения мобильного агента, а именно – ограничения на максимальный угол отклонения между прямолинейными сегментами, образующими траекторию. Исследованы теоретические свойства алгоритмов семейства. Сформулированы и доказаны гарантии отыскания решений в определенном классе. Проведено эмпирическое исследование предложенных алгоритмов.
  - Предложены новые типы эвристических функций для задачи поиска пути на графе регулярной декомпозиции, которые для каждого экземпляра задачи поиска учитывают его специфику и успешно аппроксимируются современными нейросетевыми моделями. Разработаны алгоритмы поиска, основанные на комбинации предложенных обучаемых эвристик и классических техник систематизации поиска, обладающие строгими теоретическими гарантиями. Эффективность применения разработанных алгоритмов на практике подтверждается многочисленными численными экспериментами (в том числе экспериментами по решению задач визуальной навигации, т.е. построения опорного маршрута мобильного робота по изображению подстилающей поверхности).

**Теоретическая и практическая значимость** заключается в следующем. С теоретической точки зрения предложен ряд новых методов, предназначенных для решения различных вариаций задач поиска пути (или совокупности путей) на графах регулярной декомпозиции, исследованы свойства предложенных методов (доказан ряд теорем о предоставляемых алгоритмами гарантиях). Их создание расширяет и обогащает основы теории графов и искусственного интеллекта и может служить фундаментом для дальнейших исследований в этой области. Отдельно стоит отметить, что предложенные в работе способы интеграции обучаемых (с помощью современных искусственных нейронных сетей) эвристик и алгоритмов классического поиска, гарантируют построение корректного решения, а также, в отдельных случаях, гарантируют отыскание решения, превышающего по стоимости оптимальное решение не более, чем в заданное пользователем число раз.

С практической точки зрения предложенные методы и алгоритмы могут применяться в робототехнических системах для повышения степени их автономности за счет повышения эффективности решения задач планирования траектории (с учетом различных ограничений). Так, ряд предложенных методов уже был апробирован в ходе работы на колесных мобильных роботах (в частности были апробированы методы построения

неконфликтных траекторий для группы роботов). Результаты проведенных экспериментальных исследований подтвердили применимость разработанных подходов на практике.

**Апробация работы.** Результаты работы неоднократно докладывались на профильных Российских и международных научных мероприятиях (конференциях, семинарах и пр.), среди которых можно отметить следующие: Международная конференция “Системный анализ и информационные технологии” (2015), Всероссийский научно-практический семинар “Беспилотные транспортные средства с элементами искусственного интеллекта” (2016, 2017, 2019, 2021), Национальная конференция по искусственному интеллекту (2016, 2018, 2021), Международная научно-техническая конференция “Экстремальная робототехника” (2016), Совместный семинар Российской ассоциации искусственного интеллекта и Федерального исследовательского центра “Информатика и управление” Российской академии наук (2023, 2025), German Conference on Artificial Intelligence (2015, 2019), International Workshop on Multi-agent Pathfinding (2016, 2019, 2026), International Conference on Automated Planning and Scheduling (2017, 2020, 2021), International Conference on Autonomous Agents and Multi-agent Systems (2018), European Conference on Mobile Robotics (2019), Conference on Interactive Collaborative Robotics (2019, 2020), AAAI Conference on Artificial Intelligence (2023), IEEE Conference on Intelligent Robots and Systems (2024, 2025).

Ряд результатов был получен в процессе выполнения работ по следующим грантам Российского фонда фундаментальных исследований (РФФИ), Российского научного фонда (РНФ), Министерства науки и высшего образования Российской Федерации (МНВО РФ): 075-15-2024-544, “Математические модели и численные методы как основа для разработки робототехнических комплексов, новых материалов и интеллектуальных технологий конструирования” (МНФО РФ, 2024-2026); 075-15-2020-799, “Методы построения и моделирования сложных систем на основе интеллектуальных и суперкомпьютерных технологий, направленные на преодоление больших вызовов” (МНФО РФ, 2020-2023); 16-11-00048 “Создание теории, методов и моделей децентрализованного управления поведением коллективов когнитивных робототехнических систем в недетерминированной среде” (РНФ, 2016-2018, 2019-2020); 18-37-20032 мол\_а\_вед “Методы управления автономными техническими объектами на основе планирования траектории в среде с динамическими препятствиями” (РФФИ, 2019-2020); 17-07-00281-а “Модели и методы решения задач интеллектуального управления коалицией сложных технических объектов” (РФФИ, 2017-2019); 14-11-00692 “Создание основ теории и технологии построения многоуровневых систем управления коалициями сложных технических объектов в динамической среде на основе взаимодействия методов искусственного интеллекта и теории динамических систем” (РНФ, 2014-2016).

**Достоверность результатов** подтверждается строгой теоретической обоснованностью и корректным применением используемого математического аппарата (дискретная математика, теория графов, математическая логика), согласованием полученных результатов с известными теоретическими положениями в рассматриваемой области, а также результатами экспериментальных исследований, выполненными как с использованием открытых данных, так и с использованием реальных робототехнических систем (колесных мобильных роботов).

**Публикации.** Основные результаты работы опубликованы в ведущих научных журналах и сборниках трудов конференций. Общее число публикаций – 54, среди которых можно выделить следующие группы публикаций:

- Публикации в ведущих отечественных журналах (K1/K2 Списка ВАК): 12 [1–12].
- Публикации в ведущих зарубежных журналах (Q1-Q3 по Scopus/WoS, приравненные к K1 Списка ВАК): 3 [13–15].
- Публикации в сборниках трудов конференций, имеющих рейтинг A/A\* по CORE: 6 [16–21].
- Публикации в сборниках трудов ведущих конференций, индексируемые в международных базах данных (Scopus/WoS): 20 [22–41].
- Русскоязычные публикации в сборниках трудов Российских профильных конференций, индексируемые в РИНЦ: 13 [42–54].

**Личный вклад автора.** Содержание диссертационной работы в полной мере отражает личный вклад автора в проведенное исследование. Автор принимал непосредственное участие в получении представленных в диссертации результатов, включая разработку и теоретическое исследование предлагаемых методов и алгоритмов, а также их экспериментальную апробацию. Подготовка к публикации полученных результатов проводилась совместно с соавторами, причем вклад диссертанта был определяющим.

**Соответствие паспорта специальности.** Диссертация и полученные в ней результаты соответствуют паспорту специальности 1.2.1. – Искусственный интеллект и машинное обучение, а именно: пункту 16 “Исследования в области специальных методов оптимизации, проблем сложности и элиминации перебора, снижения размерности”, пункту 6 в части “Разработка систем управления с использованием систем искусственного интеллекта и методов машинного обучения в том числе – управления роботами, автомобилями, БПЛА и т.п.”, пункту 5 в части “Исследования в области совместного применения методов машинного обучения и классического математического моделирования” и пункту 2 в части “Исследования в области оценки качества и эффективности алгоритмических

и программных решений для систем искусственного интеллекта и машинного обучения”.

### Основные положения, выносимые на защиту:

1. Метод поиска пути на динамических графах регулярной декомпозиции, и алгоритм его реализующий, допускающий возможность перемещения между произвольными вершинами графа и гарантирующий полноту и оптимальность отыскиваемых решений. Оценка сложности указанного алгоритма.
2. Метод поиска субоптимального решения задачи построения пути на динамическом графе регулярной декомпозиции и алгоритм его реализующий, использующий технику переназначения родителя в дереве поиска для повышения вычислительной эффективности. Теоретическое обоснование предоставляемых алгоритмом гарантий качества отыскиваемых решений.
3. Метод и алгоритм конфликтно-ориентированного планирования для построения совокупности неконфликтных путей на графе регулярной декомпозиции, допускающий возможность перемещения между произвольными вершинами графа, гарантирующий оптимальность отыскиваемых решений и использующий оригинальные техники сокращения перебора для повышения вычислительной эффективности.
4. Метод и алгоритм приоритизированного планирования для построения совокупности неконфликтных путей на графе регулярной декомпозиции, направленный на эффективный поиск субоптимальных решений и допускающий возможность перемещения между произвольными вершинами графа.
5. Семейство алгоритмов поиска пути на графе регулярной декомпозиции, учитывающих геометрические ограничения, и гарантирующее отыскание оптимальных решений в определенном классе.
6. Оригинальные эвристические функции для решения задачи поиска пути на графе регулярной декомпозиции и способы их интеграции с алгоритмами систематического поиска. Способы автоматического построения (обучения) указанных эвристических функций на основе глубоких нейронных сетей.
7. Гибридные алгоритмы поиска пути на графе регулярной декомпозиции, использующие предлагаемые обучаемые эвристики и техники систематического поиска, гарантирующие корректность отыскиваемых решений вне зависимости от выходных значений искусственной нейронной сети.

## Содержание работы

Во **введении** обосновывается актуальность исследований, проводимых в рамках данной диссертационной работы, формулируется цель, ставятся задачи работы, излагается научная новизна и практическая значимость представляемой работы.

В **главе 1** вводятся основные понятия, формализуется задача планирования траектории как задача поиска пути на графе особой структуры и проводится анализ современного состояния исследований.

Основным объектом исследования являются графы, вложенные в метрическое пространство  $\mathcal{M}^2$ , т.е. графы вида  $\mathcal{G} = (V, E, w)$ , где  $V$  – множество вершин,  $E$  – множество ребер, а  $w : E \rightarrow \mathbb{R}^+$  – функция, задающая веса ребер, для которых при этом задано отображение  $coord : V \rightarrow \mathcal{M}$ . При этом вес ребра  $e = (u, v)$  равен расстоянию:  $w(e) = \|coord(u) - coord(v)\|$ .

**Определение 1.** Граф регулярной декомпозиции (ГРД) – это граф, вложенный в  $\mathbb{R}^2$ , и содержащий вершины, которым соответствуют точки на плоскости, образующие регулярную сетку, т.е.:  $\forall v, u \in V : coord_x(u) = coord_x(v) + k \cdot \Delta$ ,  $coord_y(u) = coord_y(v) + k \cdot \Delta$ , где  $k \in \mathbb{N} \cup 0$ ,  $\Delta = const > 0$ . Ребрами соединяются ортогонально- и диагонально смежные вершины (8-связный ГРД).

Введённое определение ГРД весьма схоже с распространённым в области теории графов определением решетчатого графа (графа-решетки), который имеет вершины в точках плоскости с неотрицательными целыми координатами. Ключевое отличие состоит в том, что ГРД является не графом-решёткой, а его частичным подграфом, в котором отсутствуют некоторые вершины.

Далее в главе показана связь задачи поиска пути ГРД с задачей планирования траектории для мобильного робота. Для  $\Delta \geq 2r$ , где  $r$  – радиус безопасности робота, справедливо следующее утверждение:

**Утверждение 1.** *Путь на ГРД является решением задачи геометрического планирования.*

Здесь под геометрическим планированием подразумевается построение опорного пути (т.е. такого пути, следование вдоль которой обеспечивается системой управления в автоматическом режиме) в виде совокупности промежуточных точек, соединенных отрезками – см. Рис. 1.

Далее в главе проведен анализ литературы в области планирования путей в контексте мобильной робототехники. Показано, что современные системы управления мобильными роботами строятся по модульному

---

<sup>2</sup>Здесь и далее полагается, что  $\mathcal{M} = \mathbb{R}^2$  для простоты восприятия и визуализации. Все положения работы справедливы и для пространств более высоких размерностей

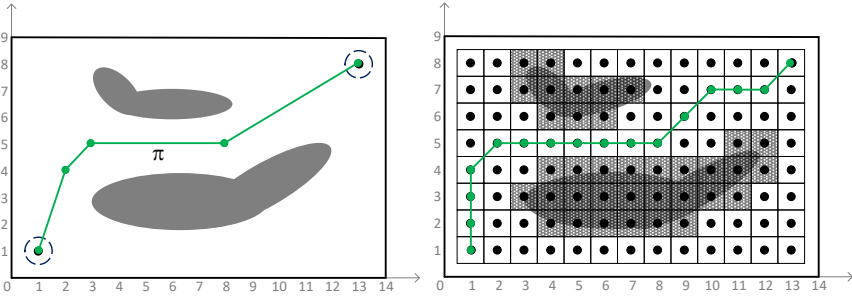


Рис. 1 — Непроходимые области рабочего пространства и геометрический путь (слева), аппроксимация рабочего пространства в виде ГРД (справа) и путь на нём.

принципу, часто иерархическому, где выделяется отдельный модуль планирования траектории. Обсуждаются альтернативные подходы – т.н. сквозное (end-to-end) обучение нейросетевых стратегий навигации и гибридные подходы, сочетающие обучаемые модули и классические алгоритмы. Описываются подходы к геометрическому планированию, учету кинематических ограничений и динамики окружающей среды. Проводится анализ методов решения задачи поиска совокупности неконфликтных путей на ГРД, которые активно используются в современных мульти-роботных системах. Анализируются методы повышения вычислительной эффективности эвристического поиска в задачах планирования в целом и задачах планирования пути на ГРД в частности. На основе проведенного анализа формулируются основные направления исследований, на которых предлагается сфокусироваться в дальнейшей работе.

**Глава 2** посвящена методам решения поиска пути на динамическом ГРД, в котором доступность вершин и рёбер зависит от времени, применимым для планирования траектории мобильного робота в среде с движущимися препятствиями. Ключевой особенностью предлагаемых методов, отличающих их от мировых аналогов, является поддержка переходов между произвольными вершинами графа – см. Рис. 2а.

Рассмотрим конечный ГРД  $\mathcal{G} = (V, E, w)$ , вложенный в  $\mathbb{R}^2$ . Будем считать, что на парах (различных) вершин ГРД определена функция

$$los : V \times V \rightarrow \{true, false\}, \tag{1}$$

которая определяет возможность перехода между произвольными вершинами графа. Также будем считать, что функция  $w$ , определяющая веса рёбер, определена и на произвольных парах вершин:  $w : V \times V \rightarrow \mathbb{R}^+$ .

Для моделирования динамических препятствий введем понятие *безопасных временных интервалов*. Для каждой вершины  $v \in V$  задано

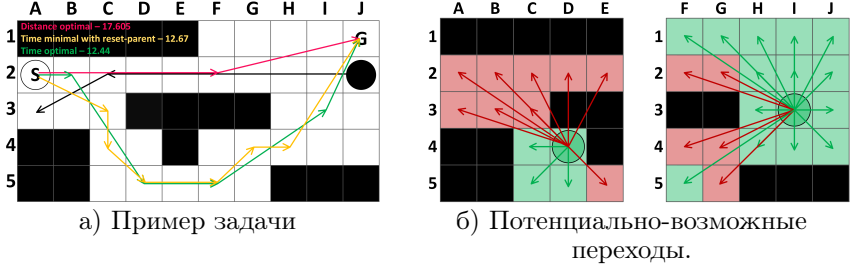


Рис. 2 — Задача поиска пути на динамическом ГРД.

конечное множество  $T_{safe}(v)$ , представляющее собой объединение непересекающихся интервалов  $[t_l, t_u] \subseteq T = [0, +\infty)$ , в течение которых агент может безопасно находиться в этой вершине. Аналогично, для каждой пары вершин задано множество  $T_{safe}(u, v)$ .

Определим действие перемещения (в произвольном направлении) как пару  $(u, v, t)$ , где  $u, v$  — вершины, а  $t$  — время начала движения. Действие допустимо, если  $t \in T_{safe}(u)$ ,  $t \in T_{safe}(u, v)$ ,  $t + w(e) \in T_{safe}(v)$  и  $los(u, v) = true$ .

**Определение 2.** Допустимым ПН-путем из  $v_{start}$  в  $v_{goal}$  на динамическом ГРД назовем последовательность вида:  $\pi(v_{start}, v_{goal}) = ((u_1, v_1, t_1), (u_2, v_2, t_2), \dots, (u_n, v_n, t_n))$ , такую что:

$$u_1 = v_{start}, v_n = v_{goal} \quad (2a)$$

$$\forall i = \overline{1, n-1} : v_i = u_{i+1}, \quad (2б)$$

$$\forall i = \overline{1, n} : t_i \in T_{safe}(u_i, v_i), los(u_i, v_i) = true, \quad (2в)$$

$$\forall i = \overline{1, n-1} : t_{i+1} \geq t_i + w(u_i, v_i) \quad (2г)$$

$$\forall i = \overline{1, n-1} : [t_i + w(u_i, v_i), t_{i+1}] \subseteq T_{safe}(u_{i+1}), \quad (2д)$$

$$t_1 \in SI_1(v_{start}), t_n + w(u_n, v_n) \in SI_{last}(v_{goal}) \quad (2e)$$

Приставка ПН означает, что путь содержит перемещения в произвольном направлении, т.е. не привязан к исходной топологии ГРД. ПН-путь, зачастую, короче простого пути на ГРД и содержит меньшее число поворотов.

**Определение 3.** Задача поиска допустимого ПН-пути на динамическом ГРД, AA-PFD (Any-Angle Path Finding on Dynamic Grid), — это набор:

$$AA-PFD = (\mathcal{G}, T, los, \{T_{safe}(v)\}, \{T_{safe}(u, v)\}, v_{start}, v_{goal}). \quad (3)$$

Решение задачи – допустимый ПН-путь из первого безопасного интервала  $v_{start}$  в последний безопасный интервал  $v_{goal}$ .

Стоимость решения – это время достижения целевой вершины:  $cost(\pi) = t_n + w(u_n, v_n)$ .

Далее в главе описываются предлагаемые методы и алгоритмы поиска оптимальных и субоптимальных решений задачи AA-PFD. Они основаны на принципе *безопасно-безопасно-интервального планирования* (англ. Safe Interval Path Planning или просто SIPP). Ключевая идея этого подхода состоит в том, что поиск производится в пространстве состояний, элементы которого задаются как  $n = (v, [t_l, t_u])$ , где  $[t_l, t_u]$  – безопасный интервал вершины  $v$ . Процедура генерации потомков для состояния  $n$  заключается в следующем: для каждого безопасного интервала  $[t'_l, t'_u]$  вершины  $v'$ , в которую необходимо совершить переход, вычисляется минимальная задержка  $\delta$ , которую необходимо совершить, чтобы переход  $(v, v')$  был возможен и  $v'$  достигалась в момент, принадлежащий  $[t'_l, t'_u]$ . Вычисление  $\delta$  осуществляется вспомогательной процедурой, которая учитывает все необходимые безопасные интервалы (способ реализации подобной процедуры приводится в работе). Время достижения состояния-потомка  $n'$ , т.н.  $g$ -значение, рассчитывается как  $g(n') = g(n) + \delta + w(e)$ . Использование интервалов вместо дискретных моментов времени позволяет алгоритму SIPP эффективно сокращать пространство поиска, гарантируя при этом нахождение оптимального решения.

Наивное расширение SIPP для решения задачи AA-PFD (генерация всех потомков для вершин, для которых  $los(n, v, v') = true$ ) приводит к взрывному росту коэффициента ветвления, см. Рис. 26, что замедляет алгоритм на практике. Для преодоления этой проблемы предложен новый алгоритм TO-AA-SIPP (Time-Optimal Any-Angle SIPP), основанный на принципиально иной парадигме поиска – *обратном раскрытии*.

Основная идея TO-AA-SIPP заключается в том, что на этапе инициализации создаются все возможные состояния поиска. Изначально все они, кроме начального, являются *несогласованными*: их  $g$ -значения равны  $\infty$ , а родитель неизвестен. Алгоритм итеративно пытается сделать состояния согласованными, находя для них наилучшего потенциального родителя среди уже согласованных состояний. Для этого каждое состояние  $n$  хранит список потенциальных родителей  $PARENTS(n)$  (согласованные состояния, из которых в  $n$  возможен переход) и указатель на наилучшего потенциально-го родителя  $bpp(n)$ . Для оценки перспективности состояния используется т.н.  $f$ -значение состояния:

$$f(n) = g_{low}(n) + h(n) = g(bpp(n)) + h(bpp(n), n) + h(n), \quad (4)$$

где  $g_{low}(n)$  – это оценка стоимости пути до  $n$  через его наилучшего потенциального родителя, а  $h(bpp(n), n)$  – эвристическая оценка стоимости

перехода между ними,  $h(n)$  – эвристическая оценка стоимости пути от  $n$  до целевого состояния<sup>3</sup>.

На каждой итерации из списка состояний-кандидатов на дальнейшее рассмотрение, OPEN, извлекается состояние  $n$  с минимальным  $f(n)$ . Затем выполняется попытка улучшить его  $g$ -значение путем валидации перехода от его текущего наилучшего родителя  $bpp(n)$  к  $n$ . Если валидация успешна и новое время достижения, лучше текущего, то  $g(n)$  и  $parent(n)$  обновляются. После этого для  $n$  осуществляется поиск в множестве  $PARENTS(n)$  нового кандидата, способного обеспечить еще более низкую оценку  $g_{low}(n)$ . Если такой кандидат находится,  $n$  возвращается в OPEN для дальнейшего рассмотрения. Если нет, то состояние  $n$  объявляется *согласованным* и перемещается в отдельный список CLOSED. В этот момент оно обавляется в списки  $PARENTS$  для всех состояний  $n' \in OPEN$ , переход в которые из  $n$  допустим ( $los(n.v, n'.v) = true$ ). Если согласованным становится целевое состояние, алгоритм завершается и возвращает путь, восстанавливаемый по родительским указателям. Псевдокод главного цикла TO-AA-SIPP представлен на Рис. 3.

Для алгоритма TO-AA-SIPP доказываются следующие утверждения, характеризующие его теоретические свойства.

**Лемма 1.** *Для любого состояния поиска  $n$  либо  $g_{low}(n) = g(bpp(n)) + h(bpp(n), n)$ , либо  $g_{low}(n) = g(n)$ .*

**Лемма 2.** *Для любого состояния поиска  $n$  всегда:  $g_{low}(n) \leq g(n)$ .*

**Лемма 3.** *Для любого  $i$  выполняется  $g_{low}^i(n) \leq g_{low}(n)$ , где  $g_{low}^i(n)$  – это  $g_{low}$ -значение состояния  $n$  извлеченного из списка OPEN в начале  $i$ -й итерации основного цикла, а  $g_{low}(n)$  – в конце  $i$ -й итерации.*

**Лемма 4.** *Последовательность  $\{f_{min}^1, f_{min}^2, \dots, f_{min}^K\}$ , где  $f_{min}^i$  –  $f$ -значение состояния, извлеченного из OPEN на  $i$ -й итерации основного цикла, является невозрастающей последовательностью.*

**Лемма 5.** *Список CLOSED содержит состояния поиска,  $g$ -значение которых не может быть уменьшено, т.е. такие состояния до которых известен путь минимальной стоимости.*

Для доказательств Лемм 1–5 используются методы математической индукции и доказательства от противного, анализ граничных случаев. Каждая следующая Лемма опирается на предыдущие. Наиболее, важным является утверждение о монотонности  $f$ -значений (Лемма 4). С помощью него Лемма 5 устанавливает, что состояния поиска, добавленные в CLOSED являются согласованными, т.е. кратчайший путь до них известен.

---

<sup>3</sup>Определение  $f$ -значения отличается от стандартного, принятого в классических алгоритмах эвристического поиска:  $f(n) = g(n) + h(n)$ .

---

---

**Процедура T0-AA-SIPP-Main:**

```
1  while  $\min_{n \in OPEN} f(n) < \infty$  do
2       $n \leftarrow \arg \min_{n \in OPEN} f(n)$ ;  $OPEN \leftarrow OPEN \setminus \{n\}$ 
3      if  $bpp(n) \in PARENTS(n)$  then
4           $PARENTS(n) \leftarrow PARENTS(n) \setminus \{bpp(n)\}$ 
5       $g\_new \leftarrow \text{ValidateTransition}(n, bpp(n))$ 
6      if  $g\_new < g(n)$  then
7           $g(n) \leftarrow g\_new$ ;  $parent(n) \leftarrow bpp(n)$ 
8      if  $\text{SetBestPotentialParent}(n)$  then
9           $OPEN \leftarrow OPEN \cup \{n\}$ 
10         continue
11     if  $g(n) + h(n) \leq \min_{n \in OPEN} f(n)$  then
12          $CLOSED \leftarrow CLOSED \cup \{n\}$ 
13         if  $n.v = v_{goal}$  and  $g(n) \in SI_{last}(v_{goal})$  then
14             return  $\text{ReconstructPath}(n)$ 
15         foreach  $n' \in OPEN$  do
16             if  $los(n', n) = true$  then
17                  $PARENTS(n') \leftarrow PARENTS(n') \cup \{n\}$ 
18                 if  $g(n) + h(n, n') < g_{low}(n')$  then
19                      $g_{low}(n') \leftarrow g(n) + h(n, n')$ 
20                      $bpp(n') \leftarrow n$ 
21                      $f(n') \leftarrow g_{low}(n') + h(n')$ 
22                     Update  $n'$  in  $OPEN$ 
23     else
24          $OPEN \leftarrow OPEN \cup \{n\}$ 
25 return failure
```

---

---

Рис. 3 — Основной цикл алгоритма T0-AA-SIPP.

Важным следствием этой леммы является теорема об оптимальности решения, отыскиваемого алгоритмом T0-AA-SIPP:

**Теорема 1.** *Решение задачи AA-PFD, возвращаемое алгоритмом T0-AA-SIPP, является оптимальным.*

**Лемма 6.** *Когда состояние поиска  $n$ , такое что  $bpp(n) = parent(n)$ , извлекается из OPEN на очередной итерации основного цикла, то это состояние будет добавлено в список CLOSED на этой же итерации.*

**Лемма 7.** *Если в основном цикле опустить проверку на достижение целевого состояния в строке (опустить строки 13–14), то алгоритм завершит свою работу за конечное число итераций.*

Следствием Леммы 7 является теорема, характеризующая вычислительную трудоемкость алгоритма TO-AA-SIPP.

**Теорема 2.** *Максимальное число итераций алгоритма TO-AA-SIPP составляет  $M(M - 1)$ , где  $M$  – это число состояний поиска.*

Если временная сложность одной итерации основного цикла алгоритма есть  $O(1)$ , то временная сложность алгоритма TO-AA-SIPP есть  $O(M^2)$ .  $M$  (число состояний поиска) ограничено числом безопасных интервалов на вершинах ГРД, т.е. константой. Следовательно, TO-AA-SIPP характеризуется квадратичной временной сложностью относительно числа вершин –  $O(V^2)$ .

**Лемма 8.** *Если в основном цикле опустить проверку на достижение целевого состояния в строке (опустить строки 13–14), то за конечное число итераций всякое достижимое состояние будет добавлено в список CLOSED.*

Леммы 7–8 используются для доказательства следующей теоремы.

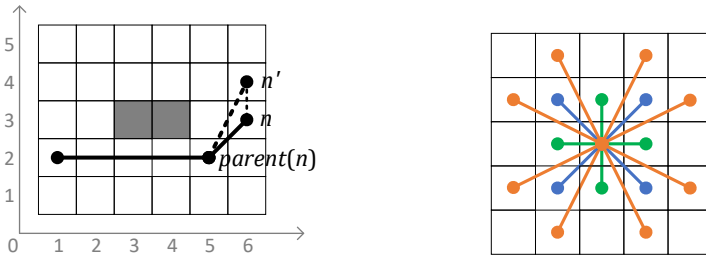
**Теорема 3.** *Алгоритм TO-AA-SIPP является полным, т.е. гарантирует отыскание решения задачи AA-PFD, если оно существует, и гарантирует корректное завершение работы за конечное число итераций, если решения задачи AA-PFD не существует.*

Несмотря на теоретическую обоснованность, использование алгоритма TO-AA-SIPP может быть затруднено на практике, т.к. логика его работы нетривиальна, а сложность квадратична, в то время как в практических задачах востребована простота реализации, скорость работы алгоритма, при этом требованием к получению оптимальных решений можно пренебречь. Эти соображения обосновывают разработку новых вычислительно-эффективных алгоритмов решения задачи AA-PFD, использующих базовый принцип безопасно-интервального планирования, т.е. алгоритм SIPP, и прямого поиска в пространстве состояний.

В частности предлагается алгоритм, опирающийся на прямой поиск и технику переназначения родительского указателя. При генерации потомков для состояния  $n$  и смежной вершины  $v'$ , помимо стандартного перехода  $n \rightarrow v'$ , рассматривается также возможность прямого перехода из  $parent(n)$ , в  $v'$ . Если такой переход возможен ( $los(parent(n), v, v') = true$ ) и время его выполнения оказывается меньше, то состояние-потомок  $n'$  добавляется в дерево поиска с родителем  $parent(n)$ , а не с  $n$ . Это позволяет, во-первых, спрямлять путь, во-вторых достигать вершины ГРД раньше по

времени – см. Рис. 4а. Соответствующий алгоритм назван **AA-SIPP** (Any-Angle Safe Interval Path Planning). Использование техники переназначения родительского указателя позволяет **AA-SIPP** избежать квадратичной сложности и одновременно предоставлять обоснованные гарантии на качество отыскиваемых решений.

**Теорема 4.** Для любой задачи *AA-PFD*, имеющей решение в исходной топологии ГРД, алгоритм **AA-SIPP** найдёт решение, при этом стоимость этого решения не будет превосходить стоимость решения, найденного с помощью стандартного алгоритма *SIPP*.



а) Переназначение родительского указателя. б) 16-связная решётка переходов на ГРД.

Рис. 4 — Способы элиминации перебора для получения субоптимальных решений задачи *AA-PFD*.

Другим способом модифицировать алгоритм *SIPP* для решения задачи *AA-PFD*, является применение  $2^k$ -решеток переходов, где параметр  $k$  определяет количество рассматриваемых направлений движения при генерации потомков (например,  $k = 4$  дает 16 направлений – см. Рис. 4б). На этой основе строятся следующие модификации безопасно-интервального планирования (алгоритма *SIPP*): *WSIPP* и *FocalSIPP*.

Алгоритм *WSIPP* (Weighted *SIPP*) использует взвешенную эвристическую функцию, т.е. функцию вида  $\hat{h}(n) = w \cdot h(n)$ , для приоритизации состояний в списке *OPEN*, где  $w > 1$  – входной параметр. Интуитивно, чем больше значение  $w$  тем более предпочтительны для дальнейшей генерации потомков состояния, расположенные ближе к цели. Для подобного алгоритма доказывается следующее утверждение.

**Утверждение 2.** Алгоритм *WSIPP*, использующий для своей работы взвешенную монотонную эвристику, не гарантирует отыскания решения задачи поиска пути на динамическом ГРД в случае, если процедура повторного рассмотрения состояний поиска не предусмотрена.

Для решения указанной проблемы предлагается использовать две модификации алгоритма – модификацию с поддержкой повторного рассмотрения состояний – **WrSIPP** и модификацию с дублирующими состояниями, обладающими различными  $f$ -значениями – **WdSIPP**.

Другой субоптимальный решатель задачи **AA-PFD**, предлагаемый в работе, **FocalSIPP**, использует дополнительный список **FOCAL** для выбора состояния на очередной итерации основного цикла. В этот список попадают состояния из **OPEN**, чьи  $f$ -значения ( $f(n) = g(n) + h(n)$ ) не превышают минимальное  $f$ -значение более чем в  $w$  раз. Выбор состояния для раскрытия производится из **FOCAL** по вторичной эвристике  $h_{FOCAL}$  (например, оценка оставшегося числа шагов). Это позволяет сильно фокусировать поиск, особенно при больших значениях  $w$ , в случае если вторичная эвристика достаточно информативна.

Проведено экспериментальное исследование предложенных алгоритмов. Использовались ГРД из известной в сообществе открытой коллекции **MovingAI**, моделирующие пространства различного вида (помещения, городские карты, лабиринты и т.д.). Безопасные интервалы индуцировались движущимися динамическими препятствиями, траектории которых были сгенерированы автоматически.

В предварительной серии сравнивались между собой алгоритмы, использующие решетку переходов (использовалась решетка с параметром  $k = 4$ , обеспечивающим коэффициент ветвления 16) и требующие настройки входного параметра  $w$  (веса эвристической функции) – **WrSIPP**, **WdSIPP**, **FocalSIPP**, для выбора наиболее подходящего варианта для дальнейших исследований. По итогам этих экспериментов был выбран алгоритм **WdSIPP** с  $w = 2$ , как обладающий наиболее подходящим балансом между скоростью работы и стоимостью отыскиваемых решений. Далее в основной серии экспериментов этот алгоритм обозначен просто как **WSIPP**.

В основной серии экспериментов проводилось исследование алгоритмов **TO-AA-SIPP**, **AA-SIPP**, **WSIPP** на трёх различных ГРД **random** ( $32 \times 32$ ), **arena** ( $49 \times 49$ ) и **warehouse** ( $170 \times 84$ ) с различным числом динамических препятствий (32–128). В качестве базового алгоритма для сравнения использовалась наивная адаптация **SIPP** для поиска оптимальных решений задачи **AA-PFD** – **nTO-AA-SIPP** (прямой поиск с генерацией всех возможных потомков при рассмотрении состояния). Медианное время работы представлено на Рис. 5а–5в, а нормированная стоимость отыскиваемых решений на Рис. 5г.

Анализ результатов позволяет сделать следующие выводы. Во-первых, алгоритм **TO-AA-SIPP** на порядки быстрее **nTO-AA-SIPP** (в 3–40 раз), что подтверждает эффективность предлагаемого подхода обратного раскрытия. Во-вторых, алгоритмы **AA-SIPP** и **WSIPP** являются ещё более эффективными на практике и обеспечивают дальнейшее ускорение на 1–2

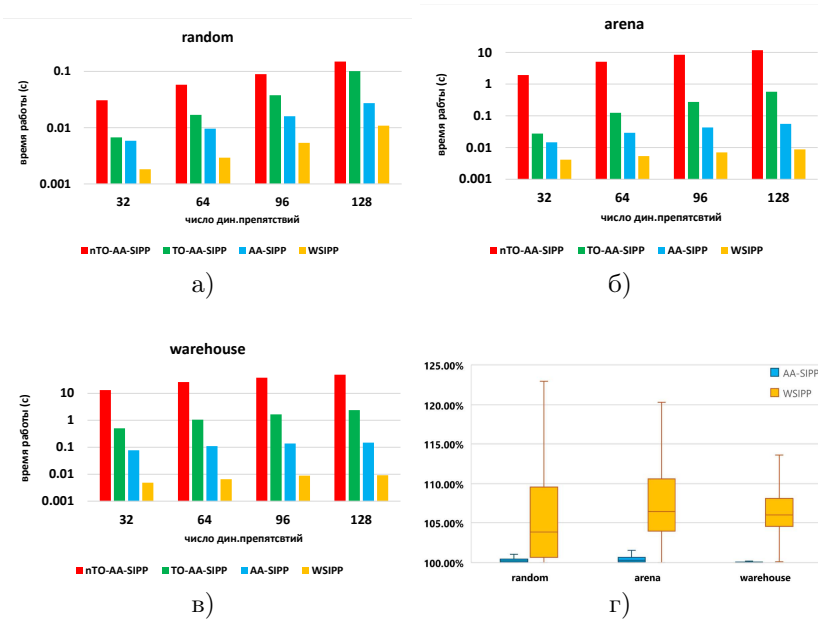


Рис. 5 — Медианное время работы алгоритмов решения задачи AA-PFD (а, б, в) и диаграмма размаха нормированной стоимости решения (г).

порядка относительно TO-AA-SIPP. В-третьих, стоимость решений, найденных AA-SIPP, в подавляющем большинстве случаев не превосходит стоимость оптимальных решений, более, чем на 1-2% (для WSIPP это превышение более существенно и составляет 2-12%), что говорит о том, что этот алгоритм обладает наиболее подходящим балансом между скоростью работы и качеством отыскиваемых решений на практике.

**Глава 3** посвящена задаче централизованного планирования безопасных траекторий для группы мобильных роботов (агентов), функционирующих в общем рабочем пространстве. Эта задача формализуется как поиск совокупности неконфликтных путей на ГРД. Основным отличием предлагаемых методов от известных аналогов является поддержка возможности перемещения агентов между произвольными вершинами графа, что позволяет сократить длину траекторий и время достижения целей.

Как и ранее в работе рассматривается взвешенный ГРД  $\mathcal{G} = (V, E, w)$ , вложенный в  $\mathbb{R}^2$ . Задана функция  $los : V \times V \rightarrow \{true, false\}$ , определяющая возможность перехода между произвольными вершинами. Заданы множества начальных и целевых вершин для  $n$  агентов:  $S = \{s^1, \dots, s^n\}$ ,  $G = \{g^1, \dots, g^n\}$ .

**Определение 4.** Действие – это тройка  $a = (u, v, \Delta)$ , где  $u, v \in V$  – вершины начала и окончания действия (также обозначаемые как  $source(a)$ ,  $target(a)$ ),  $\Delta \in T = [0, \infty)$  – продолжительность действия ( $dur(a)$ ).

Если  $u \neq v$ , то действие называется действием *перемещения* и его продолжительность равна  $\Delta = \|coord(u), coord(v)\|$ . Если  $u = v$ , то действие называется действием *ожидания* и его продолжительность произвольна.

**Определение 5.** Путь  $\pi^i$  из  $s^i \in S$  в  $g^i \in G$  – это последовательность пар (*действие, момент начала действия*),  $\pi^i = (a_1^i, t_1^i), (a_2^i, t_2^i), \dots, (a_k^i, t_k^i)$ , такая что:

$$\begin{aligned} source(a_1^i) &= s^i, target(a_k^i) = g^i, \\ \forall j \in [1, k-1] : source(a_{j+1}^i) &= target(a_j^i), t_j^i + dur(a_j^i) = t_{j+1}^i. \end{aligned} \quad (5)$$

Продолжительность пути определим как  $dur(\pi^i) = t_k^i + dur(a_k^i)$ .

Введем в рассмотрение функцию  $con : A \times T \times A \times T \rightarrow \{true, false\}$ , где  $A$  – множество всех возможных действий. Рассмотрим четвёрку  $(a_k^i, t_k^i, a_l^j, t_l^j)$ , где  $a_k^i$  – это  $k$ -е действие  $i$ -го агента,  $a_l^j$  – это  $l$ -е действие  $j$ -го агента, а  $t_k^i$  и  $t_l^j$  – моменты начала совершения этих действий, соответственно. Будем говорить, что имеет место *конфликт*, если  $con(a_k^i, t_k^i, a_l^j, t_l^j) = true$ .

Способ задания функции  $con$  может быть различным (исходя из особенностей решаемой задачи). Здесь и далее будем считать, что:

$$\begin{aligned} con(a_k^i, t_k^i, a_l^j, t_l^j) = true &\iff \\ \exists t \in [0, \min(dur(a_k^i), dur(a_l^j))] : & \\ dist(pos(t, a_k^i, t_k^i), pos(t, a_l^j, t_l^j)) < 2 \cdot r, & \end{aligned} \quad (6)$$

где  $pos(t, a, t_{start})$  – положение агента в момент времени  $t$  при совершении действия, начало которого было в момент  $t_{start}$ ,  $dist$  – расстояние,  $r$  – положительная константа, задающая радиус безопасности агента.

**Определение 6.** Пути  $\pi^i$  и  $\pi^j$  содержат конфликт если:

$$\exists (a_k^i, t_k^i) \in \pi^i, (a_l^j, t_l^j) \in \pi^j : con(a_k^i, t_k^i, a_l^j, t_l^j) = true \quad (7)$$

**Определение 7.** *Задача* построения множества неконфликтных путей, допускающих перемещения между произвольными вершинами ГРД, AA-MAPF (от англ. Any-Angle Multi-agent Pathfinding), – это набор:

$$AA-MAPF = (\mathcal{G}, S, G, los, con), \quad (8)$$

где  $S, G$  – множества начальных и целевых вершин на ГРД  $\mathcal{G}$ ,  $los$  – функция, определяющая возможность перехода между произвольными вершинами,  $con$  – функция, определяющая конфликт.

Решение задачи AA-MAPF – множество путей  $\Pi = \{\pi^1, \pi^2, \dots, \pi^n\}$ , допускающих перемещения между произвольными вершинами ГРД в соответствии с функцией *los*, т.ч. любая пара различных путей не содержит конфликтов (определяемых с помощью функции *con*).

Стоимость решения определяется как  $cost(\Pi) = \sum_{i=1}^n cost(\pi^i)$ .

Пример задачи AA-MAPF (и её решения) приведен на Рис. 6 справа. Для сравнения слева показана упрощенная постановка, когда допускаются лишь перемещения между ортогонально-смежными вершинами ГРД (и каждое действие занимает 1 такт времени) – задача, известная в сообществе, как MAPF.

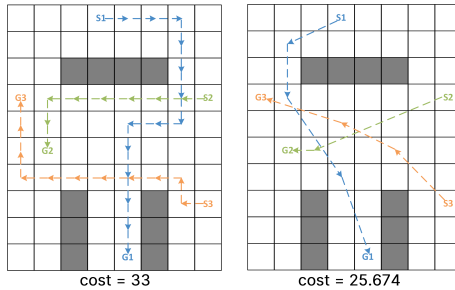


Рис. 6 — Пример решения задач MAPF и AA-MAPF.

Для решения задачи AA-MAPF предложены два класса алгоритмов: алгоритмы, предоставляющие теоретические гарантии о стоимости получаемых решений (оптимальные решения, либо решения, стоимостью которых не превосходит стоимость оптимальных более, чем в заданное число раз): AA-CCBS, Focal-AA-CCBS и алгоритм, направленный на быстрый поиск решений без предоставления гарантий об их стоимости (субоптимальные решения): AA-SIPP(m).

Алгоритм AA-CCBS основан на принципе конфликтно-ориентированного планирования и является двухуровневым иерархическим алгоритмом. На верхнем уровне строится *дерево ограничений*. Узел дерева  $N$  содержит пару  $(\Pi, \Psi)$ , где  $\Pi$  – набор из  $n$  индивидуальных путей, а  $\Psi$  – множество ограничений, которым эти пути удовлетворяют. На нижнем уровне происходит построение индивидуальных путей, учитывающих ограничения. Опишем основные шаги алгоритма.

- Инициализация:** Строятся индивидуальные пути без ограничений (с помощью ранее предложенного алгоритма TO-AA-SIPP, при этом в каждая вершина и ребро ГРД содержит единый безопасный интервал  $[0, \infty)$ ), формируется корневой узел  $N_{root} = (\Pi_0, \emptyset)$ .

1. **Цикл верхнего уровня:** На каждой итерации из списка OPEN (листьев дерева ограничений) выбирается узел  $N_{best}$  с минимальной стоимостью.
2. В наборе путей  $N_{best}$ . Ищется конфликт  $c = (a_k^i, t_k^i, a_l^j, t_l^j)$ . Если конфликта нет, решение найдено.
3. Для разрешения конфликта генерируются два ограничения – по одному на каждого участника конфликта. Ограничения являются интервальными и запрещают выполнение действия в определенный временной интервал. Для случая конфликта двух действий перемещения имеем:

$$\psi(i, c) = (a_k^i, [t_k^i, t_k^i + \alpha)), \quad (9)$$

$$\psi(j, c) = (a_l^j, [t_l^j, t_l^j + \beta)), \quad (10)$$

где  $\alpha$  и  $\beta$  – минимальные задержки, устраняющие конфликт. Для конфликта между ожиданием и перемещением ограничения определяются схожим образом, при этом ограничение на действие ожидания также задается через временной интервал. Ограничения индуцируют множества безопасных интервалов на рёбрах и вершинах ГРД. Таким образом задача поиска индивидуального пути с учетом этих ограничений – есть задача AA-PFD, рассмотренная во 2-й главе работы. Для поиска оптимальных решений этой задачи предлагается использовать алгоритм TO-AA-SIPP.

4. Создаются два узла-потомка  $N_{left}$  и  $N_{right}$ . В  $N_{left}$  к множеству ограничений добавляется  $\psi(i, c)$ , а в  $N_{right}$  –  $\psi(j, c)$ .
5. **Запуск алгоритма планирования нижнего уровня (индивидуальное планирование):** Для каждого потомка выполняется перепланирование пути для ограниченного агента с помощью алгоритма TO-AA-SIPP. Если путь найден, новый узел добавляется в OPEN.

Процесс продолжается до нахождения бесконфликтного набора путей. Псевдокод алгоритма представлен на Рис. 7.

Ранее в литературе было показано, что подход конфликтно-ориентированного планирования гарантирует отыскание оптимального решения, если на нижем уровне используется планировщик, гарантирующий оптимальность. Поскольку в алгоритме AA-CCBS для поиска индивидуальных путей используется алгоритм TO-AA-SIPP, всегда возвращающий решения минимальной стоимости (см. Теорему 1), то справедливо следующее утверждение:

**Утверждение 3.** *Алгоритм AA-CCBS гарантирует отыскание решения задачи AA-MAPF, если оно существует, при этом данное решение является оптимальным (т.е. минимальным по стоимости).*

---



---

**Алгоритм AA-CCBS**( $\mathcal{G}, s^1, s^2, \dots, s^n, g^1, g^2, \dots, g^n$ ):

**Входные данные:** Граф  $\mathcal{G}$ , множества начальных и целевых вершин  $s^1, \dots, s^n, g^1, \dots, g^n$

**Выходные данные:** Множество неконфликтных путей  
 $\Pi = \{\pi^1, \dots, \pi^n\}$

```

1  foreach  $i \in \{1, \dots, n\}$  do
2     $\pi_i \leftarrow \text{TO-AA-SIPP}(\mathcal{G}, s^i, g^i, \emptyset)$ 
3   $\Pi_0 \leftarrow \{\pi_1, \dots, \pi_n\}; \Psi_0 \leftarrow \emptyset; N_{root} \leftarrow (\Pi_0, \Psi_0)$ 
4   $OPEN \leftarrow \{N_{root}\}$ 
5  while  $OPEN \neq \emptyset$  do
6     $N_{best} \leftarrow \arg \min_{N \in OPEN} \text{cost}(N, \Pi)$ 
7     $c = (a_k^i, t_k^i, a_l^j, t_l^j) \leftarrow \text{GetConflict}(N_{best}, \Pi)$ 
8    if  $c = \emptyset$  then
9       $\text{return } N_{best}, \Pi$ 
10   foreach  $z \in \{i, j\}$  do
11      $\psi_{new} \leftarrow \text{GetConstraint}(z, c)$ 
12      $\Psi_{new} \leftarrow N_{best}, \Psi \cup \psi_{new}$ 
13      $\pi'_z \leftarrow \text{TO-AA-SIPP}(\mathcal{G}, s^z, g^z, \Psi_{new})$ 
14     if  $\pi'_z \neq \emptyset$  then
15        $\Pi_{new} \leftarrow N_{best}, \Pi \setminus \{\pi_z\} \cup \{\pi'_z\}$ 
16        $N_{new} \leftarrow (\Pi_{new}, \Psi_{new})$ 
17        $OPEN \leftarrow OPEN \cup \{N_{new}\}$ 

```

---

Рис. 7 — Алгоритм конфликтно-ориентированного планирования AA-CCBS

Для сокращения перебора на верхнем уровне алгоритма AA-CCBS в работе предлагается использовать *мульти-ограничения*, когда вместо одного действия запрещается сразу множество взаимно-конфликтных действий. Помимо наивного способа формирования таких ограничений, известного в литературе (мульти-ограничения типа 1 – MC1), предложены оригинальные способы конструирования мульти-ограничений, опирающиеся на геометрию движения агентов и не сокращающие, в отличие от MC1 продолжительность исходного конфликтного интервала (мульти-ограничения типа 2 и 3 – MC2, MC3).

Все три указанных способа опираются на общий подход, связанный с выделением двух множеств взаимно-конфликтных действий, и различаются в способах формирования этих множеств. Как было показано в ранних работах по схожей тематике такой принцип формирования мульти-ограничений не влияет на гарантии отыскания алгоритмом конфликтно-ориентированного планирования оптимальных решений. Следовательно, справедливо следующее утверждение.

**Утверждение 4.** *Решение задачи AA-MAPF, возвращаемое алгоритмом AA-CCBS с использованием мульти-ограничений является оптимальным.*

Для дальнейшего сокращения перебора предлагается использовать дополнительную фокусировку на верхнем уровне конфликтно-ориентированного планирования, т.е. использовать списка FOCAL, в который добавляются все листья дерева ограничений (частичные решения) удовлетворяющие условию:  $cost(N.\Pi) \leq w \cdot \min_{N' \in OPEN}(Cost(N'.\Pi))$ , где  $w > 1$  – задаваемый пользователем параметр. Теперь на каждой итерации алгоритма происходит выбор наиболее перспективного узла дерева ограничений из списка FOCAL, а не OPEN, с помощью дополнительной эвристической функции  $h_{FOCAL}$ :  $N_{best} \leftarrow \arg \min_{N \in FOCAL}(h_{FOCAL}(N))$ . Предлагается в качестве такой эвристики использовать число наложенных ранее ограничений:  $h_{FOCAL} = -\sum_{i=1}^n |N.\Psi_i|$ .

Известно, что метод конфликтно-ориентированного планирования, использующий фокусировку поиска, гарантирует что возвращаемое им решение задачи MAPF является ограничено субоптимальным (т.е. его стоимость не превосходит стоимостью оптимального решения, более чем в  $w$  раз), если на нижнем уровне используется планировщик, гарантирующий отыскание оптимальных индивидуальных путей. Поскольку в рассматриваемом случае для этой цели используется алгоритм TO-AA-SIPP, то справедливо следующее утверждение.

**Утверждение 5.** *Стоимость решения, отыскиваемого алгоритмом Focal-AA-SIPP не превосходит стоимости оптимального решения, более чем в  $w$  раз, где  $w > 1$  – параметр задаваемый пользователем.*

Далее, для ещё большего сокращения пространства поиска при решении задачи AA-MAPF предлагается отказаться от конфликтно-ориентированного планирования и перейти к приоритизированному планированию, которое состоит в следующем:

1. Каждому агенту присваивается уникальный приоритет.
2. В порядке убывания приоритета для каждого агента строится индивидуальный путь, не содержащий конфликтов с ранее построенными путями (они считаются фиксированными).

Такой подход не предоставляет теоретических гарантий, однако является простым в реализации и эффективным в вычислительном плане, т.к. исходная задача много-агентного планирования сводится к  $n$  задачам поиска пути с ограничениями. Для решения последней предлагается использовать эффективный субоптимальный планировщик, предложенный ранее в работе – AA-SIPP.

Для повышения эффективности приоритизированного планирования предлагаются две техники:

- *Безопасные интервалы в начальных вершинах*: каждой начальной вершине добавляется интервал  $[0, t_{ssi}]$ , в течение которого запрещается использовать эту вершину для построения какого-либо пути. Это предотвращает блокировку низкоприоритетных агентов высокоприоритетными.
- *Детерминированное переназначение приоритетов*: если для какого-то агента путь не найден, его приоритет становится наивысшим, а приоритеты остальных циклически сдвигаются.

Проведено обширное экспериментальное исследование предложенных алгоритмов решения задачи AA-MARF с использованием публичной, широко известной в сообществе и часто используемой в литературе коллекции задач много-агентного планирования путей MovingAI. Выполнено несколько серий экспериментов. Сначала между собой сравнивались различные модификации алгоритма AA-CCBS. В ходе этих экспериментов подтверждена эффективность предлагаемых в работе техник формирования мульти-ограничений, в особенности – использования мульти-ограничений типа 3 для поиска оптимальных решений и использования комбинации фокусировки поиска и мульти-ограничений типа 3 для отыскания ограниченно субоптимальных решений. Вторая серия экспериментов была направлена на исследование предлагаемого алгоритма приоритизированного планирования – AA-SIPP(m). В этих экспериментах подтверждена эффективность разработанных техник переназначения приоритетов и фиксации начальных безопасных интервалов в начальных вершинах. Обобщенные результаты первой и второй серии экспериментов представлены в Табл. 1–2.

Таблица 1 — Число успешно решенных заданий за ограниченный временной бюджет.

	AA-CCBS	AA-CCBS+MC	Focal-AA-CCBS+MC	AA-SIPP(m)	CCBS
empty	141	208	455	3 640	679
random	231	300	623	1 922	810
maze	62	92	199	1 968	277
den312	129	149	366	7 512	627
warehouse	308	361	797	15 441	1 268
<b>Total</b>	<b>871</b>	<b>1 110</b>	<b>2 440</b>	<b>30 483</b>	<b>3 661</b>

В таблицах строки соответствуют различным ГРД (картам) из коллекции MovingAI, столбцы – алгоритмам. При этом помимо предлагаемых в работе тестировался алгоритм конфликтно-ориентированного планирования CCBS, допускающий перемещения лишь между ортогонально-смежными вершинами. В ячейках Табл. 1 для каждого алгоритма приведено суммарное число задач из коллекции заданий для соответствующей карты, для которых алгоритм успешно нашел решение за время работы, не превышающее 300 с. Чем больше это число, тем выше практическая

Таблица 2 — Нормализованная стоимость решений.

	AA-CCBS	AA-CCBS+MC	Focal-AA-CCBS+MC	AA-SIPP (m)	CCBS
empty	100.0%	100.0%	101.4%	104.3%	127.4%
random	100.0%	100.0%	101.1%	104.4%	117.6%
maze	100.0%	100.0%	100.4%	104.2%	115.6%
den312	100.0%	100.0%	100.2%	102.3%	117.9%
warehouse	100.0%	100.0%	100.2%	103.6%	111.8%
<b>AVG</b>	<b>100.0%</b>	<b>100.0%</b>	<b>100.7%</b>	<b>103.8%</b>	<b>118.1%</b>

вычислительная эффективность алгоритма. Видно, что отыскание оптимальных решений весьма трудоемкая задача, однако переход к поиску ограниченно-субоптимальных решений и использование мульти-ограничений (Focal-AA-CCBS+MC) повышает число решенных задач почти в 3 раза. И хотя результат несколько хуже, чем CCBS, нужно помнить, что последний решает гораздо более простую задачу, в которой не допускаются перемещения в произвольном направлении. Отказ же от конфликтно-ориентированного планирования в пользу приоритизированного планирования повышает число успешно решённых заданий почти в 10 раз – см. результаты алгоритма AA-SIPP (m). Одновременно, решения, отыскиваемые этим алгоритмом, в среднем лишь на 3.8% превышают по стоимости оптимальные решения, в то время как для CCBS такое превышение составляет более 18%.

Помимо численных экспериментов, была проведена апробация разработанных алгоритмов, в частности алгоритма AA-SIPP (m), на реальных роботах. Для этого эксперимента использовался специализированный полигон размера  $4.8 \times 6$  м. и шесть колесных роботов YARP-2 с дифференциальным приводом – см. Рис. 8.



Рис. 8 — Внешний вид полигона во время проведения экспериментов с группировкой из 6 роботов – слева. Справа – соответствующий ГРД.

Было осуществлено более 60 запусков с различными настройками системы управления. В каждом эксперименте согласованное множество неконфликтных путей конструировалось алгоритмом AA-SIPP (m) и затем каждый путь передавался на индивидуального робота для локальной отработки. Была достигнута 100% успешность выполнения задания при использовании расширенных радиусов безопасности на этапе планирования для компенсации неточности последующей отработки траектории.

В **главе 4** рассматривается задача поиска пути на ГРД с учетом геометрических ограничений, а именно ограничений на максимальный угол отклонения между смежными прямолинейными сегментами траектории. Это целесообразно для косвенного учета кинематических ограничений мобильного робота и повышения безопасности последующего следования вдоль траектории на практике.

Для формализации задачи вводится понятие секции – упорядоченной пары произвольных вершин графа  $e = (u, v)$ . Допустимость секции, как и ранее, определяется с помощью функции  $los(u, v) : V \times V \rightarrow \{true, false\}$ . Две секции, имеющие общую вершину ГРД, –  $e = (u, v)$  и  $e' = (v, v')$ , – называются смежными. Углом отклонения секции  $e'$  от  $e$ ,  $\alpha(e, e')$ , называется модуль угла между векторами  $(v_x - u_x, v_y - u_y)$  и  $(v'_x - v_x, v'_y - v_y)$ . Путь  $\pi = (e_1, \dots, e_n)$  – это последовательность смежных допустимых секций. Его стоимость равна сумме длин секций. Максимальный угол поворота пути определяется как  $\alpha_m(\pi) = \max_i \alpha(e_i, e_{i+1})$ .

**Определение 8.** *Задача* поиска пути с ограничением на максимальный угол поворота, AC-PF (Angle-Constrained Path Finding, – это набор:

$$\text{AC-PF} = (\mathcal{G}, v_{start}, v_{goal}, \alpha_{max}, los), \quad (11)$$

где  $\mathcal{G} = (V, E)$  – заданный ГРД,  $v_{start} \in V$ ,  $v_{goal} \in V$  – зафиксированные начальная и конечная вершины соответственно,  $\alpha_{max} \in [0^\circ, 180^\circ]$  – заданное ограничение на максимальный угол поворота,  $los$  – функция, определяющая возможность перехода между произвольными вершинами ГРД.

*Решение задачи* – это набор смежных допустимых секций, образующих путь  $\pi$  из  $v_{start}$  в  $v_{goal}$ , т.ч.  $\alpha_m(\pi) \leq \alpha_{max}$  (см. Рис. 9).

*Критерием* оценки качества решения является длина пути (чем меньше, тем лучше).

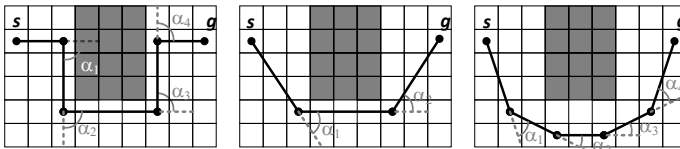


Рис. 9 – Примеры решений задачи AC-PF для различных  $\alpha_{max}$

Главная особенность задачи AC-PF – невозможность стандартного отсечения по стоимости при поиске (более длинный частичный путь до вершины может приводить к решению, в отличие от более короткого, из-за ограничений на угол отклонения). Поэтому в работе предлагается сфокусироваться на вычислительно-эффективных алгоритмах построения субоптимальных решений.

Предлагается новый алгоритм LIAN, который ищет решения в ограниченном, но практически важном классе  $\Delta$ -путей. Для этого вводится множество  $CIRCLE(v, \Delta)$ , аппроксимирующее окружность радиуса  $\Delta$  с центром в вершине  $v$  и состоящее из вершин ГРД, клетки которых пересекает эта окружность. Секция  $e = (u, v)$  называется  $\Delta$ -секцией, если  $v \in CIRCLE(u, \Delta)$ .  $\Delta$ -путем называется путь, в котором все секции, кроме, возможно, последней, являются  $\Delta$ -секциями – см. Рис. 10.

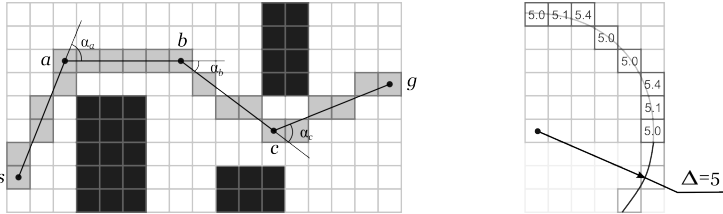


Рис. 10 —  $\Delta$ -путь на ГРД для  $\Delta = 5$ .

Алгоритм LIAN является алгоритмом эвристического поиска, опирающимся на стандартный порядок исследования пространства состояний (от начального до целевого), ключевым отличием которого является определение состояния и процедура генерации потомков. Состояние поиска в алгоритме LIAN определяется как пара  $n = (v, parent(n))$ , где  $parent(n)$  — родительское состояние. Это необходимо, так как достижение одной и той же вершины из разных предшественников ведет к разным углам поворота. С каждым состоянием поиска ассоциируется  $g(n)$  (стоимость пути от старта),  $h(n)$  (эвристическая функция, оценивающая расстояние до цели, например, Евклидово расстояние) и функция приоритета,  $f(n) = g(n) + h(n)$ , определяющая порядок рассмотрения состояний. Процедура генерации потомков для состояния  $n$  заключается в следующем: сначала формируется множество вершин-кандидатов  $SUCC_V = CIRCLE(n.v, \Delta)$ . Затем для каждой вершины  $v_{circ} \in SUCC_V$  проверяется, осуществим ли переход ( $los(n.v, v_{circ}) = true$ ) и не превышает ли угол между секцией от родителя к  $n$  и новой секцией заданного ограничения  $\alpha_{max}$ . Если обе проверки успешны, создается новое состояние-потомок  $n_{succ} = (v_{circ}, n)$  с  $g(n_{succ}) = g(n) + \|n.v, v_{circ}\|$ , которое затем добавляется в дерево поиска (список OPEN).

Для алгоритма LIAN доказывается следующая теорема.

**Теорема 5.** Алгоритм LIAN является полным и оптимальным относительно класса задач  $P^\Delta$ , то есть гарантирует отыскание  $\Delta$ -решения, если оно существует, и гарантирует, что стоимость найденного решения не превосходит стоимость любого другого  $\Delta$ -решения (т.е. гарантирует отыскание  $\Delta$ -пути минимальной длины).

Здесь  $\mathbf{P}^\Delta$  – подмножество всех решаемых задач AC-PF, таких что хотя бы одно из их возможных решений является  $\Delta$ -путем на ГРД. Доказательство опирается на конечность пространства состояний, монотонность используемой эвристики и тот факт, что алгоритм систематически перебирает все возможные  $\Delta$ -пути.

Основным недостатком LIAN является необходимость фиксации параметра  $\Delta$ . Неудачный выбор может привести к невозможности создания потомков для состояний, принадлежащих частичным путям, ведущих к цели. Для решения этой проблемы предлагается модифицированный алгоритм D-LIAN (Dynamic LIAN), который динамически изменяет значение  $\Delta$  в ходе поиска.

В D-LIAN каждое состояние поиска  $n$  ассоциируется помимо  $g$ -,  $h$ -,  $f$ -значений с  $\Delta$ -значением,  $\Delta(n)$ , равным тому значению  $\Delta$ , с которым оно было сгенерировано. Пользователем задаются границы изменения:  $\Delta_{min}$  и  $\Delta_{max}$ . На этапе инициализации  $\Delta(n_{start}) = \Delta_{max}$ . Процедура генерации потомков аналогична LIAN, но теперь, если множество потомков  $SUCC$  пусто и текущее  $\Delta(n) > \Delta_{min}$ , то значение  $\Delta(n)$  уменьшается вдвое, а само состояние  $n$  возвращается в список кандидатов для повторного рассмотрения. Благодаря этому появляется возможность использовать секции меньшей длины для построения фрагментов путей, обходящих препятствия сложной конфигурации. Для ускорения поиска на открытых участках предусмотрен обратный механизм: если  $\Delta(n) = \Delta(parent(n))$  и  $\Delta(n) < \Delta_{max}$ , то для состояний-потомков  $\Delta$  увеличивается (удваивается), что способствует более быстрому продвижению к цели. Псевдокод алгоритма приведён на Рис. 11.

Для D-LIAN доказываются ряд утверждений.

**Утверждение 6.** Алгоритм D-LIAN корректно завершается за конечное число шагов.

**Утверждение 7.** Алгоритм D-LIAN гарантирует отыскание решения задачи AC-PF, если для неё существует  $\Delta_{max}$ -решение, и корректное завершение в противном случае.

**Утверждение 8.** Длина пути, отыскиваемого алгоритмом D-LIAN, не превосходит длину оптимального  $\Delta_{max}$ -пути.

Следствием этих утверждений является следующая теорема.

**Теорема 6.** Алгоритм D-LIAN является полным и оптимальным относительно класса задач  $\mathbf{P}^{\Delta_{max}}$ , то есть гарантирует отыскание решения задачи AC-PF, если для неё существует  $\Delta_{max}$ -решение (и гарантирует корректное завершение в противном случае), более того стоимость отыскиваемого решения (длина пути) не превосходит стоимости оптимального  $\Delta_{max}$ -решения.

---

```

Алгоритм D-LIAN( $\mathcal{G}, v_{start}, v_{goal}, \alpha_{max}, \Delta_{min}, \Delta_{max}, h, los$ ):
1   $n_{start} \leftarrow \text{GenerateSearchNode}(v_{start}, \text{null})$ 
2   $\Delta(n_{start}) \leftarrow \Delta_{max}; g(n_{start}) \leftarrow 0; \text{parent}(n_{start}) \leftarrow \text{null}$ 
3   $OPEN \leftarrow \{n_{start}\}; CLOSED \leftarrow \emptyset$ 
4  while  $OPEN \neq \emptyset$  do
5       $n \leftarrow \arg \min_{n \in OPEN} (f(n) = g(n) + h(n))$ 
6       $OPEN \leftarrow OPEN \setminus \{n\}$ 
7      if  $n.v = v_{goal}$  then
8           $\text{return ReconstructPath}(n)$ 
9       $SUCC \leftarrow \text{GenerateDLIANSuccessors}(n, \alpha_{max}, \Delta_{max}, v_{goal}, los)$ 
10     if  $SUCC = \emptyset$  and  $\Delta(n) > \Delta_{min}$  then
11          $\Delta(n) \leftarrow \Delta(n)/2; OPEN \leftarrow OPEN \cup \{n\}$ 
12         continue
13      $\text{UpdateOPENandCLOSED}()$ 
14      $CLOSED \leftarrow CLOSED \cup \{n\}$ 
15 return failure

Процедура GenerateDLIANSuccessors( $n, \alpha_{max}, \Delta_{max}, v_{goal}, los$ ):
16  $SUCC \leftarrow \emptyset; SUCC_V \leftarrow \text{CIRCLE}(n.v, \Delta(n))$ 
17 if  $\|(n.v, v_{goal})\| < \Delta(n)$  then
18      $SUCC_V \leftarrow SUCC_V \cup \{v_{goal}\}$ 
19 foreach  $v_{circ} \in SUCC_V$  do
20      $e_1 = (\text{parent}(n).v, n.v); e_2 = (n.v, v_{circ})$ 
21     if  $los(n.v, v_{circ}) = \text{false}$  or  $\alpha(e_1, e_2) > \alpha_{max}$  then
22         continue
23      $n_{succ} \leftarrow \text{GenerateSearchNode}(v_{circ}, n)$ 
24      $g(n_{succ}) \leftarrow g(n) + \|e_2\|$ 
25     if  $\Delta(n) = \Delta(\text{parent}(n))$  and  $\Delta(n) < \Delta_{max}$  then
26          $\Delta(n_{succ}) \leftarrow 2 \cdot \Delta(n)$ 
27     else
28          $\Delta(n_{succ}) \leftarrow \Delta(n)$ 
29      $\text{parent}(n_{succ}) \leftarrow n$ 
30      $SUCC \leftarrow SUCC \cup \{n_{succ}\}$ 
31 return SUCC

```

---

Рис. 11 — Алгоритм D-LIAN.

Экспериментальное исследование проводилось на трех коллекциях карт: авторской Moscow Maps (модели реальной городской застройки), а также на картах Warcraft III (открытые пространства) и Baldur's

Gate (помещения) из известной открытой коллекции задач планирования MovingAI. Сравнивались алгоритмы LIAN (с  $\Delta = 5, 10, 20$ ), D-LIAN (с различными комбинациями  $\Delta_{max}$  и  $\Delta_{min}$ ) и известные мировые аналоги Theta\*-LA и wTheta\*-LA. Ограничение на угол отклонения составляло  $20^\circ$ ,  $25^\circ$ ,  $30^\circ$ . Обобщенные результаты первой серии экспериментов приведены в Табл. 3, показывающей долю успешно решенных задач из всех предъявленных.

Таблица 3 — Результаты экспериментального исследования алгоритмов решения задачи AC-PF.

	Moscow Maps			Baldur's Gate			Warcraft III		
	20°	25°	30°	20°	25°	30°	20°	25°	30°
LIAN-5	99.4%	100.0%	100.0%	100.0%	100.0%	100.0%	99.4%	100.0%	100.0%
LIAN-10	96.8%	99.0%	99.6%	87.7%	96.2%	98.4%	92.7%	96.1%	98.9%
LIAN-20	87.2%	91.4%	94.0%	61.6%	72.8%	81.3%	76.6%	80.5%	84.4%
wTheta*-LA	56.4%	86.8%	93.8%	33.8%	76.2%	91.4%	49.4%	80.0%	87.7%
Theta*-LA	4.4%	6.2%	14.0%	10.6%	10.9%	11.7%	4.0%	5.5%	10.0%

Как видно из таблицы, алгоритмы семейства LIAN существенно превосходят аналоги по эффективности, т.е. способности отыскивать решения с заданными ограничениями. Наибольшую эффективность демонстрирует алгоритм при  $\Delta = 5$  (LIAN-5). Однако, как было установлено в ходе экспериментов, пути отыскиваемые этим алгоритмом содержат в среднем в 3 раза больше точек поворота, по сравнению с LIAN-20.

Вторая серия экспериментов была осуществлена с целью дать ответ на вопрос – может ли техника динамической подстройки параметра  $\Delta$ , используемая алгоритмом D-LIAN, обеспечить более выгодный баланс между высокой вычислительной эффективностью LIAN-5 и низким число поворотов LIAN-20. Результаты этой серии экспериментов дали положительный ответ на этот вопрос. Более половины нерешаемых алгоритмом LIAN-20 заданий, решаются алгоритмом D-LIAN-20-5, при этом число поворотов в пути повышается незначительно и разница по этому показателю относительно LIAN-5 по-прежнему почти трехкратная (в пользу D-LIAN).

В [главе 5](#) рассматривается задача автоматического построения эвристических функций, учитывающих расположение препятствий в рабочей области и позволяющих более эффективно фокусировать поиск и повышать его вычислительную эффективность на практике. Предлагается подход, основанный на автоматическом конструировании информативных, обусловленных на конкретный экземпляр задачи, эвристик с помощью методов машинного обучения и их последующей интеграции в алгоритмы поиска.

Рассматриваются два типа задач. Первая – классическая задача поиска кратчайшего пути на 8-связном взвешенном ГРД, обозначаемая как PF-G (Path Finding on a Grid). Она формализуется как набор

$(\mathcal{G}, v_{start}, v_{goal}, los)$ , где  $\mathcal{G}$  – ГРД,  $v_{start}, v_{goal} \in V$  – начальная и целевая вершины, а функция  $los$  определяет допустимость перехода по ребру.

Вторая задача – поиск пути по спутниковому снимку местности, где информация о высотах (цифровая модель рельефа, DEM) недоступна для планировщика, но известна для оценки качества. Исходные данные представлены многомерным массивом  $MAP = (RGB, DEM)$ , где RGB – цветное изображение, а DEM – карта высот. Ему ставится в соответствие ГРД  $\mathcal{G}$ , все вершины и ребра которого считаются допустимыми. Стоимость перехода между вершинами  $u$  и  $v$  определяется формулой  $cost(u, v) = dist(u, v) + \alpha \cdot |dem(u) - dem(v)|$ , учитывающей как Евклидово расстояние, так и перепад высот. Требуется найти путь минимальной стоимости, используя только RGB-каналы. Эта задача обозначается как PF-RGB. Она проиллюстрирована на Рис. 12

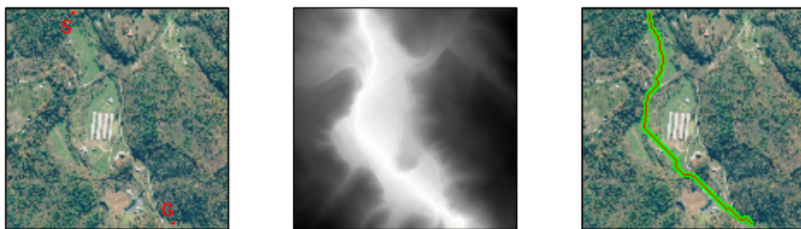


Рис. 12 – Пример задачи планирования по изображению.

Для обоих типов задач предполагается наличие конечной выборки задач, часть из которой может быть использована для обработки и анализа без каких-либо ограничений на вычислительные ресурсы (обучающая выборка), а оставшаяся часть используется для проверки эффективности разрабатываемого метода (тестовая выборка).

Подход к решению состоит в следующем. Доступная для предварительного анализа выборка используется для обучения искусственной нейронной сети, аппроксимирующей оригинальные эвристические функции. После этапа обучения, задачи из отложенной (тестовой) выборки решаются следующим образом. Сначала по входным данным (ГРД/RGB, расположение начальной и целевой вершины) нейросеть предсказывает эвристическую функцию, которая затем передается на вход алгоритму поиска, который решает задачу.

В качестве алгоритмов поиска предлагается использовать классические алгоритмы  $WA^*$  и  $FocalSearch$ . Первый использует взвешенную эвристику при поиске, т.е. эвристику вида  $\hat{h} = w \cdot h$ , где  $h$  – стандартная эвристическая функция, оценивающая расстояние на ГРД,  $w = const > 1$ . Второй алгоритм использует принцип двух-этапного выбора наиболее перспективного состояния на каждой итерации – сначала с помощью

стандартной эвристики и заданного коэффициента  $w > 1$  формируется предварительный список кандидатов, из которого затем выбирается наилучшее состояние с помощью вторичной эвристики  $h_{FOCAL}$ .

Предлагаются два вида обучаемых эвристических функций для этих алгоритмов. Первая функция – фактор коррекции,  $cf(n) \in (0,1]$ , определяется как  $cf(n) = h(n)/h^*(n)$ , где  $h(n)$  – стандартная эвристика, а  $h^*(n)$  – точная оценка стоимости кратчайшего пути до цели (доступная на этапе обучения). Эта эвристика интегрируется в алгоритм  $WA^*$  в качестве индивидуального веса для каждого состояния:  $\hat{h}(n) = \frac{1}{cf(n)}h(n)$ . Полученный гибридный алгоритм обозначается  $WA^*+CF$ .

Вторая эвристическая функция – вероятность вхождения в путь,  $pp(n) \in (0,1]$ . Она используется в качестве вторичной эвристики  $h_{FOCAL}$  в алгоритме **FocalSearch**. Результирующий алгоритм обозначается как **FS+PPM**. Предлагается также вариант этого алгоритма с  $w = \infty$ . В таком случае порядок рассмотрения состояний определяется по значениям  $pp(n)$ , а не первичной эвристики. Такая модификация обозначается как **GBFS+PPM**.

Предлагаемые гибридные алгоритмы поиска пути обладают важными гарантиями, формализуемыми в виде следующих утверждений.

**Утверждение 9.** *Алгоритмы  $WA^*+CF$ ,  $FS+PPM$ ,  $GBFS+PPM$  являются полными, т.е. гарантируют отыскание решения задачи, если оно существует, и если решения нет, корректно завершаются.*

**Утверждение 10.** *Алгоритм  $FS+PPM$  гарантирует отыскание ограниченно-субоптимального решения, т.е. пути, длина которого не превышает длину кратчайшего более, чем в заданное пользователем число раз.*

Для решения задач **PF-G** предлагается использовать все три представленных алгоритма, для задачи **PF-RGB** – алгоритм **GBFS+PPM**.

Для обучения эвристических функций используется подход с учителем, т.е. для каждой задачи из обучающей выборки формируются эталонные наборы  $cf$ - и  $pp$  значений, которые и обучаются предсказывать нейросеть (с помощью  $L_2$  функции потерь). Эталонные  $cf$ -значения формируются путем вычисления для каждой вершины стоимости кратчайшего пути до цели и последующего деления на значение стандартной эвристики. Для построения эталонных  $pp$ -значений ищется эталонный путь (отыскиваемый алгоритмом **Theta\*** для **PF-G** и **A\*** для **PF-RGB**). Вершинам этого пути присваивается  $pp = 1$ , а для остальных вершин  $pp$ -значение вычисляется по формуле, отражающей близость стоимости пути через данную вершину к стоимости эталонного пути.

Для аппроксимации эвристик предлагается архитектура искусственной нейронной сети, состоящая из трех основных блоков (см. Рис. 13), позволяющих выделять как локальные особенности задачи планирования, так и устанавливать их глобальные взаимосвязи между ними:

1. Блок кодирования входных данных (на основе сверток).
2. Блок-трансформер, реализующий механизм (пространственного) внимания (т.н. механизм attention).
3. Блок декодирования.

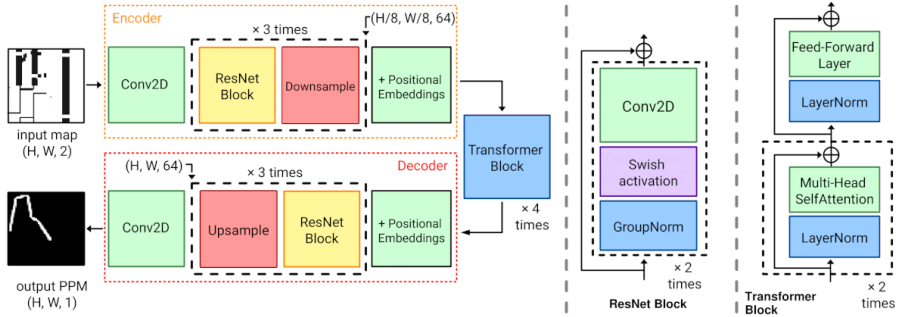


Рис. 13 — Архитектура нейронной сети, предлагаемая в работе для аппроксимации эвристических функций.

Для задачи PF-RGB также исследуются более сложные генеративные архитектуры: *StyleGAN3* (генеративно-состязательная сеть) и латентная диффузионная модель (LDM), в том числе с обуславливанием на координаты старта и цели (LDM-Cond).

Для обучения и тестирования созданы авторские наборы данных. Для задач PF-G разработан расширенный вариант известного в литературе набора данных TMP, насчитывающий 64 000 карт (ГРД) размера  $64 \times 64$ , обладающих сложной топологией. Для каждой карты создано 10 различных заданий. Таким образом общий объем набора данных составляет 640 000. Для задачи PF-RGB создан новый датасет пар RGB-DEM (18 316 карт, 183 160 задач PF-RGB), представляющих собой реальные земные ландшафты. Основным источником для составления набора данных послужил инструмент NOAA: *Data Access Viewer*, предназначенный для обработки и визуализации геопространственных данных.

Экспериментальное исследование для задач PF-G проводилось на тестовой выборке размера 64 000 заданий. Сравнивались предложенные алгоритмы ( $WA^*+CF$ ,  $FS+PPM$ ,  $GBFS+PPM$ ) с классическими ( $A^*$ ,  $WA^*$ ) и современными обучаемыми аналогами – *NeuralA\**,  $A^*+HL$ . Результаты представлены в Табл. 4.

Как видно из таблицы предложенные алгоритмы значительно превосходят аналоги. Так  $WA^*+CF$  показал наилучший результат по доле найденных оптимальных решений (85.4%). По числу итераций (основной показатель скорости) наилучшим оказался  $GBFS+PPM$ , сокращая их в среднем на 76.4% по сравнению с  $A^*$  (до 23.6%), при этом стоимость решений

Таблица 4 — Результаты экспериментального исследования различных методов решения задачи PF-G.

	Optimal Found Ratio (%) ↑	Cost Ratio (%) ↓	Expansions Ratio (%) ↓
A*	100	100	100
WA*	40.66	103.52 ± 4.85	44.43 ± 25.92
NeuralA*	29.82	104.90 ± 6.56	52.30 ± 30.47
A*+HL	79.11	100.27 ± 0.62	80.50 ± 74.40
WA*+CF	<b>85.40</b>	100.25 ± 1.13	36.98 ± 21.18
FS+PPM	82.97	<b>100.24 ± 0.74</b>	26.36 ± 21.08
GBFS+PPM	83.02	100.25 ± 0.90	<b>23.60 ± 18.34</b>

превышала оптимальную в среднем всего на 0.25%, а в 83% случаев решения были оптимальными.

Для задачи PF-RGB было проведено три серии экспериментов. В первой серии (изображения 64x64) GBFS+PPM сравнивался с NeuralA\* (единственным возможным конкурентом для решения подобного типа задач) и существенно превзошел его: относительная стоимость решений 106% против 121% у NeuralA\*, сокращение числа итераций 44% против 64% (относительно A\*, обладающего привилегированной информацией – значениями в канале DEM). Во второй серии (изображения 128x128) сравнивались между различные нейросетевые архитектуры. Наилучшие результаты продемонстрировала модель LDM-Cond (латентная диффузия, обусловленная на начальную и целевую позицию). Она обеспечила относительную стоимость решений 103% и сокращение числа итераций до 44.2%.

В заключении представлены основные результаты диссертационной работы.

## Основные результаты

1. Разработан новый метод, и алгоритм его реализующий, поиска пути на динамическом графе регулярной декомпозиции, допускающий переходы между произвольными вершинами, что позволяет сократить длину пути и время достижения цели. Метод основан на принципе безопасно-интервального планирования и использует оригинальный подход к обращению направления поиска для элиминации. Предложенный метод применим для планирования траектории мобильного агента (робота) в среде с динамическими препятствиями. При этом, в отличие от аналогов, он позволяет агенту перемещаться в произвольном направлении. Проведено теоретическое исследование алгоритма и установлены его важные свойства, в частности гарантия отыскания оптимального решения.
2. Предложен метод поиска пути на динамическом графе регулярной декомпозиции, использующий процедуру переназначения родительского узла в дереве поиска. Это позволило существенно

- сократить пространство поиска, при незначительном (на практике – менее 3%) ухудшении качества решения. Проведено теоретическое исследование алгоритма и установлены его свойства.
3. Разработан ряд новых вычислительно-эффективных субоптимальных методов поиска пути на динамическом графе регулярной декомпозиции. Методы основаны на применении решетки переходов и регулируемого коэффициента субоптимальности. Проведены их экспериментальные исследования.
  4. Предложен новый алгоритм построения совокупности неконфликтных траекторий на графе регулярной декомпозиции, основанный на принципе конфликтно-ориентированного поиска. Предложены техники сокращения пространства поиска, повышающие вычислительную эффективность без потери гарантий оптимальности.
  5. Разработан новый алгоритм приоритизированного многоагентного планирования. В отличие от аналогов, метод позволяет агентам выполнять действия произвольной продолжительности и перемещаться между любыми вершинами графа, что приводит к снижению суммарной длины траекторий. Предложены оригинальные эвристики для повышения практической эффективности. Проведено эмпирическое исследование алгоритма, в том числе на реальных роботах.
  6. Предложено семейство алгоритмов поиска на графе регулярной декомпозиции, допускающих перемещение между произвольными вершинами и учитывающих геометрические ограничения. Таким образом, становится возможен косвенный учет кинематических ограничений мобильных роботов на этапе планирования маршрута. Исследованы теоретические свойства разработанных алгоритмов, сформулированы и доказаны гарантии нахождения решений в заданном классе решений. Проведено эмпирическое исследование.
  7. Предложены новые типы эвристических функций для поиска пути на графе регулярной декомпозиции, учитывающие специфику отдельного экземпляра задачи и эффективно аппроксимируемые современными нейросетевыми моделями. Комбинация этих функций с классическими алгоритмами эвристического поиска существенно повышает их практическую эффективность, что подтверждено многочисленными вычислительными экспериментами.
  8. Разработанный подход к решению задачи поиска пути, сочетающий обучаемые эвристики и классические алгоритмы, успешно применен для планирования траектории по изображению (спутниковому снимку местности со сложным рельефом). Сравнительное исследование показало существенное преимущество предлагаемого метода над существующими мировыми аналогами.

## Публикации автора по теме диссертации

1. *Яковлев К. С., Макаров Д. А., Баскин Е. С.* Метод автоматического планирования траектории беспилотного летательного аппарата в условиях ограничений на динамику полета // Искусственный интеллект и принятие решений. — 2014. — № 4. — 3—17. **(ВАК К1)**.
2. *Яковлев К., Хитыков В., Логинов М., Петров А.* Система навигации группы БЛА на основе маркеров // Робототехника и техническая кибернетика. — 2014. — № 4. — 44—48. **(ВАК К2)**.
3. *Макаров Д. А., Панов А. И., Яковлев К. С.* Архитектура многоуровневой интеллектуальной системы управления беспилотными летательными аппаратами // Искусственный интеллект и принятие решений. — 2015. — № 3. — 18—33. **(ВАК К1)**.
4. *Яковлев К. С., Баскин Е. С., Андрейчук А. А.* Метод автоматического планирования совокупности траекторий для навигации беспилотных транспортных средств // Управление большими системами. — 2015. — № 58. — 306—342. **(ВАК К1)**.
5. *Панов А. И., Яковлев К. С.* Взаимодействие стратегического и тактического планирования поведения коалиции агентов в динамической среде // Искусственный интеллект и принятие решений. — 2016. — № 4. — 68—78. **(ВАК К1)**.
6. *Андрейчук А., Яковлев К.* Методы планирования траектории на плоскости с учетом геометрических ограничений // Известия Российской академии наук. Теория и системы управления. — 2017. — № 6. — 125—140. **(ВАК К1)**.
7. *Боковой А. В., Муравьев К. Ф., Яковлев К. С.* Система одновременного картирования, локализации и исследования неизвестной местности по видеопотоку // Информационные технологии и вычислительные системы. — 2020. — № 2. — 51—61. **(ВАК К1)**.
8. *Яковлев К. С.* AA-SIPP: Алгоритм планирования в среде с динамическими препятствиями // Искусственный интеллект и принятие решений. — 2020. — № 1. — 49—59. **(ВАК К1)**.
9. *Яковлев К. С., Белинская Ю. С., Макаров Д. А., Андрейчук А. А.* Безопасно-интервальное планирование и метод накрытий для управления движением мобильного робота в среде со статическими и динамическими препятствиями // Автоматика и телемеханика. — 2022. — № 6. — 96—117. **(ВАК К1)**.
10. *Яковлев К. С., Андрейчук А. А., Скрынник А. А., Панов А. И.* Методы планирования и обучения в задачах многоагентной навигации // Доклады Российской академии наук. Математика, информатика, процессы управления. — 2022. — Т. 508. — 88—93. **(ВАК К1)**.

11. *Миронов К., Юдин Д., Алхаддад М., Макаров Д., Пушкарев Д., Линок С., Белкин И., Криштопик А., Головин В., Яковлев К., Панов А.* STRL-ROBOTICS: интеллектуальное управление поведением робототехнической платформы в человеко-ориентированной среде // Искусственный интеллект и принятие решений. — 2023. — № 2. — 45–63. (**ВАК K1**).
12. *Головин В. А., Яковлев К. С.* Примитивы движения робота в задаче планирования траектории с кинематическими ограничениями // Информатика и автоматизация. — 2023. — Т. 22, № 6. — 1354–1386. (**ВАК K1**).
13. *Emel'yanov S., Makarov D., Panov A. I., Yakovlev K.* Multilayer cognitive architecture for UAV control // Cognitive Systems Research. — 2016. — Т. 39. — 58–72. (**WoS Q2**).
14. *Staroverov A., Muravyev K., Yakovlev K., Panov A. I.* Skill fusion in hybrid robotic framework for visual object goal navigation // Robotics. — 2023. — Т. 12, № 4. — 104. (**Scopus Q1**).
15. *Kirilenko D., Andreychuk A., Panov A. I., Yakovlev K.* Generative models for grid-based and image-based pathfinding // Artificial Intelligence. — 2025. — Т. 338. — 104238. (**WoS Q1**).
16. *Yakovlev K., Andreychuk A.* Any-Angle Pathfinding for Multiple Agents Based on SIPP Algorithm // Proceedings of The 27th International Conference on Automated Planning and Scheduling (ICAPS 2017). — 2017. — 586–593. (**Core A\*, Scopus**).
17. *Andreychuk A., Yakovlev K.* Two Techniques That Enhance the Performance of Multi-robot Prioritized Path Planning // Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2018). (**Core A\*, Scopus**). — 2018. — С. 2177–2179.
18. *Yakovlev K., Andreychuk A., Stern R.* Revisiting Bounded-Suboptimal Safe Interval Path Planning // Proceedings of the 30th International Conference on Automated Planning and Scheduling (ICAPS 2020). — 2020. — 300–304. (**Core A\*, Scopus**).
19. *Yakovlev K., Andreychuk A.* Towards Time-Optimal Any-Angle Path Planning With Dynamic Obstacles // Proceedings of the 31st International Conference on Automated Planning and Scheduling (ICAPS 2021). — 2021. — 405–414. (**Core A\*, Scopus**).
20. *Kirilenko D., Andreychuk A., Panov A., Yakovlev K.* TransPath: Learning Heuristics For Grid-Based Pathfinding via Transformers // Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI 2023). — 2023. — 12436–12443. (**Core A\*, Scopus**).

21. *Yakovlev K., Andreychuk A., Stern R.* Optimal and bounded suboptimal any-angle multi-agent pathfinding // Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2024). — 2024. — 7996–8001. (**Core A, Scopus**).
22. *Yakovlev K., Khithov V., Loginov M., Petrov A.* Distributed control and navigation system for quadrotor UAVs in GPS-denied environments // Proceedings of the 7th IEEE International Conference Intelligent Systems (IS 2014). — 2014. — 49–56. (**Scopus**).
23. *Yakovlev K., Baskin E., Hramoin I.* Grid-based angle-constrained path planning // Proceedings of the 38th German Conference on Artificial Intelligence (Künstliche Intelligenz, KI 2015). — 2015. — 208–221. (**Scopus**).
24. *Panov A. I., Yakovlev K.* Behavior and path planning for the coalition of cognitive robots in smart relocation tasks // Proceedings of the 4th International Conference on Robot Intelligence Technology and Applications (RITA 2015). — 2015. — 3–20. (**Scopus**).
25. *Khithov V., Petrov A., Tishchenko I., Yakovlev K.* Toward autonomous UAV landing based on infrared beacons and particle filtering // Proceedings of the 4th International Conference on Robot Intelligence Technology and Applications (RITA 2015). — 2015. — 529–537. (**Scopus**).
26. *Andreychuk A., Yakovlev K.* Applying MAPP Algorithm for Cooperative Path Finding in Urban Environments // Proceedings of the 2nd International Conference on Interactive Collaborative Robotics (ICR 2017). — 2017. — 1–10. (**Scopus**).
27. *Panov A. I., Suvorov R., Yakovlev K. S.* Grid Path Planning with Deep Reinforcement Learning: Preliminary Results // Proceedings of the 8th Annual International Conference on Biologically Inspired Cognitive Architectures (BICA 2017). — 2018. — 347–353. (**Scopus**).
28. *Andreychuk A., Yakovlev K.* Path Finding for the Coalition of Co-operative Agents Acting in the Environment with Destructible Obstacles // Proceedings of the 3rd International Conference on Interactive Collaborative Robotics (ICR 2018). — 2018. — 13–22. (**Scopus**).
29. *Andreychuk A., Soboleva N., Yakovlev K.* eLIAN: Enhanced Algorithm for Angle-Constrained Path Finding // Proceedings of the 16th Russian Conference on Artificial Intelligence (RCAI 2018). — 2018. — 206–217. (**Scopus**).

30. *Andreychuk A., Bokovoy A., Yakovlev K.* An Empirical Evaluation of Grid-based Path Planning Algorithms on Widely Used in Robotics Raspberry Pi Platform // Proceedings of the 2018 International Conference on Artificial Life and Robotics (ICAROB 2018). — 2018. — 383–386. (**WoS**).
31. *Bokovoy A. V., Fomin M. B., Yakovlev K. S.* Implementation of the Pathfinding System for Autonomous Navigation of Mobile Ground Robot // CEUR Workshop Proceedings. — 2018. — T. 2236. — 72–78. (**Scopus**).
32. *Yakovlev K., Andreychuk A., Belinskaya J., Makarov D.* Combining Safe Interval Path Planning and Constrained Path Following Control: Preliminary Results // Proceedings of the 4th International Conference on Interactive Collaborative Robotics (ICR 2019). — 2019. — 310–319. (**Scopus**).
33. *Yakovlev K., Andreychuk A., Vorobyev V.* Prioritized Multi-Agent Path Finding for Differential Drive Robots // Proceedings of the 2019 European Conference on Mobile Robots (ECMR 2019). — 2019. — 1–6. (**Scopus**).
34. *Soboleva N., Yakovlev K.* GAN Path Finder: Preliminary results // Proceedings of the 42nd German Conference on Artificial Intelligence (KI 2019). — 2019. — 316–324. (**Scopus**).
35. *Soboleva N., Yakovlev K.* LPLian: Angle-constrained Path Finding in Dynamic Grids // Proceedings of the 12th International Conference on the Developments in eSystems Engineering (DeSE 2019). — 2019. — 48–53. (**Scopus**).
36. *Yakovlev K., Andreychuk A., Rybecky T., Kulich M.* On the Application of Safe-Interval Path Planning to a Variant of the Pickup and Delivery Problem // Proceedings of the 17th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2020). — 2020. — 521–528. (**Scopus**).
37. *Khokhlov I., Litvinenko V., Ryakin I., Yakovlev K.* Planning to Score a Goal in Robotic Football with Heuristic Search // Proceedings of the 5th International Conference on Interactive Collaborative Robotics (ICR 2020). — 2020. — 148–159. (**Scopus**).
38. *Rybecky T., Kulich M., Andreychuk A., Yakovlev K.* Towards Narrowing the Search in Bounded-Suboptimal Safe Interval Path Planning // Proceedings of the 14th International Symposium on Combinatorial Search (SoCS 2021). — 2021. — 136–140. (**Scopus**).
39. *Ivanashev I., Andreychuk A., Yakovlev K.* Analysis of the Anytime MAPF Solvers Based on the Combination of Conflict-Based Search (CBS) and Focal Search (FS) // Proceedings of the 21st Mexican Conference on Artificial Intelligence (MICAI 2022). — 2022. — 368–382. (**Scopus**).

40. *Dergachev S., Muravyev K., Yakovlev K.* 2.5D Mapping, Pathfinding and Path Following For Navigation Of A Differential Drive Robot In Uneven Terrain // Proceedings of 13th IFAC Symposium on Robot Control (SyRoCo 2022). — 2022. — 80—85. (**Scopus**).
41. *Savinov V., Yakovlev K.* DHC-R: Evaluating “Distributed Heuristic Communication” and Improving Robustness for Learnable Decentralized PO-MAPF // Proceedings of the 8th Interactive Collaborative Robotics (ICR 2023). — 2023. — 151—163. (**Scopus**).
42. *Андрейчук А., Яковлев К.* Метод разрешения конфликтов при планировании пространственных траекторий для группы беспилотных летательных аппаратов // Третий Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта» (БТС-ИИ-2016): Труды семинара. — 2016. — 31—40. (**РИНЦ**).
43. *Макаров Д., Панов А., Яковлев К.* STRL: многоуровневая система управления интеллектуальными агентами // Пятнадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2016. Труды Конференции. Т. 1. — 2016. — 179—188. (**РИНЦ**).
44. *Андрейчук А., Боковой А., Яковлев К.* Оценка быстродействия некоторых алгоритмов планирования траектории на широко используемой в робототехнике платформе Raspberry PI // ЭР-2016: Экстремальная робототехника. Труды Международной научно-технической конференции. — 2016. — 184—189. (**РИНЦ**).
45. *Кириллов Д., Яковлев К.* Методы повышения вычислительной эффективности алгоритмов эвристического поиска в задачах планирования траектории // Интеллектуальные системы и технологии: современное состояние и перспективы. Сборник научных трудов IV Международной летней школы-семинара по искусственному интеллекту для студентов, аспирантов, молодых ученых и специалистов. — 2017. — С. 95—104.
46. *Яковлев К., Баскин Е.* Эвристическая оценка качества построения пути методом LIAN-Search // Шестая Международная конференция «Системный анализ и информационные технологии» САИТ-2015. Сборник трудов конференции. Т. 1. — 2015. — С. 132—138.
47. *Андрейчук А., Яковлев К.* Планирование траекторий для коалиции агентов. Алгоритм MAPF // Труды Второй молодежной научной конференции «Задачи современной информатики». — 2015. — 18—24. (**РИНЦ**).

48. *Андрейчук А., Яковлев К.* Планирование траектории на плоскости с учетом размера агента (мобильного робота, беспилотного транспортного средства) // Четвертый Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта» (БТС-ИИ-2017): Труды семинара. — 2017. — 107–117. **(РИНЦ)**.
49. *Соболева Н., Яковлев К.* Модификации алгоритма планирования траектории LIAN // Информатика, управление и системный анализ: Труды V Всероссийской научной конференции молодых ученых с международным участием. — 2018. — 117–126. **(РИНЦ)**.
50. *Андрейчук А., Яковлев К.* Метод планирования траекторий для группы агентов, обладающих возможностью модификации окружающей среды // Информатика, управление и системный анализ: Труды V Всероссийской научной конференции молодых ученых с международным участием. — 2018. — 73–82. **(РИНЦ)**.
51. *Соболева Н., Яковлев К.* LPLIAN: алгоритм планирования траектории с учетом геометрических ограничений в динамической среде // Пятый Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта» (БТС-ИИ-2019): Труды семинара. — 2019. — 56–65. **(РИНЦ)**.
52. *Дергачев С., Яковлев К.* Об одном вопросе реализации алгоритма планирования траектории  $A^*$  // Пятый Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта» (БТС-ИИ-2019): Труды семинара. — 2019. — 66–76. **(РИНЦ)**.
53. *Андрейчук А., Киселев Г., Яковлев К.* Интеграция методов планирования поведения и планирования траектории // Семнадцатая Национальная конференция по искусственному интеллекту с международным участием. КИИ-2019. Труды конференции. Т. 1. — 2019. — 66–74. **(РИНЦ)**.
54. *Петров А., Голубин С., Шилов А., Яковлев К.* Система управления робототехническим пожарным комплексом // Интеллектуальные системы, управление и мехатроника – 2020: Материалы Всероссийской научно-технической конференции. — 2020. — 54–58. **(РИНЦ)**.