На правах рукописи

Андрейчук Антон Андреевич

Методы конфликтно-ориентированного поиска для планирования совокупности безопасных траекторий мобильных агентов с учетом возможности совершения действий произвольной продолжительности

Специальность
1.2.3 Теоретическая информатика, кибернетика

Автореферат
диссертации на соискание учёной степени
кандидата физико-математических наук

Москва — 2023

Работа выполнена в Федеральном государственном автономном образовательном учреждении высшего образования «Российский университет дружбы народов имени Патриса Лумумбы».

Научный руководитель: кандидат физико-математических наук, ведущий научный сотрудник Федерального государственного учреждения «Федеральный исследовательский центр «Информатика и управление» Российской академии наук»
Яковлев Константин Сергеевич

Официальные оппоненты: **Жилякова Людмила Юрьевна,** доктор физико-математических наук, Федеральное государственное бюджетное учреждение науки Институт проблем управления им. В.А. Трапезникова РАН, ведущий научный сотрудник

Скороходов Владимир Александрович, доктор физико-математических наук, Федеральное государственное автономное образовательное учреждение высшего образования «Южный федеральный университет», профессор

Ведущая организация: Федеральное государственное бюджетное учреждение «Национальный исследовательский центр «Курчатовский институт»

Защита состоится «___» ___________ 2023 г. в ____ часов на заседании диссертационного совета 24.1.224.03 при Федеральном исследовательском центре «Информатика и управление» Российской академии наук (ФИЦ ИУ РАН) по адресу: 119333, г. Москва, ул. Вавилова, д. 40.

С диссертацией можно ознакомиться в библиотеке ФИЦ ИУ РАН и на официальном сайте ФИЦ ИУ РАН: http://www.frccsc.ru.

Автореферат разослан «___» ___________ 2023 г.

Ученый секретарь
диссертационного совета
24.1.224.03,
канд. техн. наук И. А. Рейер

Общая характеристика работы

Актуальность темы исследования. Согласно отчету компании China Post Group [1] объем пересылаемых почтовых отправлений только в Китае по итогам 2021 года превысил 100 млрд единиц. Розничные и технологические гиганты вкладывают значительные средства в интеллектуальные склады, чтобы своевременно и с минимальными затратами обрабатывать возросший спрос на логистику электронной коммерции. Одним из наиболее ярких примеров автоматизированных складов являются сортировочные центры компании Amazon [2]. Аналогичные системы развиваются и другими гигантами в сфере электронной коммерции, такими как, например, Alibaba [3].

Одной из задач, возникающей при разработке интеллектуальных систем для управления группой роботов, является задача согласованного перемещения. Для того, чтобы роботы-грузчики могли эффективно выполнять задачи связанные с перемещением товаров, их сортировкой и компоновкой, необходимо иметь возможность автоматически строить для них согласованные траектории перемещения, которые позволяют с одной стороны выполнять поставленные роботам задачи как можно эффективнее, с другой – избегать столкновений между роботами, следующими вдоль построенных траекторий.

Аналогичные задачи, связанные с согласованными перемещениями, возникают и в других областях – интеллектуальных транспортных системах, при мониторинге территорий, ликвидации чрезвычайных происшествий. Не смотря на очевидные различия между примерами, в которых возникают задачи согласованного перемещения, принцип работы подходов к решению этих задач идентичен. В компьютерных науках, в частности в искусственном интеллекте, их называют задачами многоагентного планирования (англ. multi-agent pathfidning). Даже в случае использования ряда допущений, таких как дискретное представление пространства поиска в виде графа специального вида и дискретизации времени, поиск оптимального решения задачи многоагентного планирования относится к классу NP-полных задач. На сегодняшний день существует ряд алгоритмов, гарантирующих, что найденное ими решение является оптимальным. Однако большинство из них опирается на допущение о том, что все действия агентов имеют одинаковую продолжительность. Это допущение ограничивает набор возможных действий для агентов, и приводит к тому, что в действительности могут существовать решения, обладающие лучшим качеством за счет учета возможности совершения большего набора действий.

Ввиду вышеизложенного в диссертацию вошли исследования трех основных направлений: 1) исследование и разработка алгоритма решения задачи многоагентного планирования, который с одной стороны учитывает возможность совершения агентами действий произвольной продолжительности, с другой – гарантирует, что найденное им решение является оптимальным; 2) исследование и разработка модификаций алгоритма, позволяющих повысить его вычислительную эффективность, сохранив при этом гарантию нахождения оптимальных решений; 3) исследование и разработка модификаций алгоритма, позволяющих повысить его вычислительную эффективность за счет отказа от поиска оптимальных решений и переходу к поиску субоптимальных решений.

Степень разработанности. Для анализа текущего состояния области был проведен обзор существующих постановок задачи многоагентного планирования, а также методов и алгоритмов их решения. Основными отечественными учеными, занимающимися исследованиями и внесшими существенный вклад в область планирования, в том числе многоагентного, являются М.Л. Цетлин, В.И. Варшавский, Д.А. Поспелов, Г.С. Осипов, В.Г. Конюший, Б.С. Юдинцев, Ю.И. Нечаев, В.Э. Карпов, Л.Ю. Жилякова. Основными зарубежными исследователями, внесшими вклад в создание методов решения задачи многоагентного планирования, являются S. Koenig, M. Likhachev, R. Stern, A. Felner, N. Sturtevant, D. Silver, D. Harabor, H. Ma, W. Ruml, A. Botea, H. Choset, R. Bartak, P. Surynek.

Целью диссертационной работы является разработка и исследование методов и алгоритмов решения задачи многоагентного планирования траекторий с учетом возможности совершения агентами действий произвольной продолжительности.

Для достижения поставленной цели необходимо решить следующие **задачи**:

1. Разработать алгоритм планирования совокупности неконфликтных траекторий, опирающийся на принцип конфликтно-ориентированного поиска и допускающий возможность совершения действий произвольной продолжительности. Провести исследование теоретических свойств разработанного алгоритма, в частности, доказать утверждение об оптимальности решений, отыскиваемых алгоритмом.
2. Разработать модификации алгоритма, позволяющие повысить его вычислительную эффективность и сохраняющие при этом теоретические гарантии нахождения оптимальных решений. Провести исследование теоретических свойств разработанных модификаций.
3. Разработать модификации алгоритма, позволяющие повысить его вычислительную эффективность за счет перехода к поиску ограниченно-субоптимальных решений с возможностью настройки коэффициента субоптимальности. Провести исследование теоретических свойств разработанных модификаций.

Научная новизна:

1. Предложенный в работе алгоритм многоагентного планирования, основанный на подходе конфликтно ориентированного поиска, позволяющий учитывать произвольную продолжительность действий агентов, при этом гарантирующий, что отыскиваемые им решения являются оптимальными.
2. Предложенные в работе модификации разработанного алгоритма, такие как приоритизация конфликтов, непересекающееся разделение и

эвристики верхнего уровня позволяют повысить вычислительную эффективность разработанного алгоритма и при этом сохранить свойство оптимальности.

3. Предложенные в работе модификации разработанного алгоритма, комбинирующие подход конфликтно ориентированного поиска с субоптимальными алгоритмами планирования, отыскивают ограниченно субоптимальные решения и имеют регулируемый параметр, отвечающий за фактор субоптимальности, что позволяет найти требуемый баланс между качеством отыскиваемых решений и вычислительными ресурсами, затрачиваемыми на их поиск.

Методы исследования. В диссертационной работе применяются методы теории графов, линейного программирования, вычислительной геометрии, исследования операций.

Основные положения, выносимые на защиту:

1. Разработанный алгоритм планирования совокупности неконфликтных траекторий мобильных агентов с учетом возможности совершения действий произвольной продолжительности. Сформулированная и доказанная теорема об оптимальности решений, отыскиваемых предложенным алгоритмом.
2. Предложенные в работе модификации алгоритма многоагентного планирования, позволяющие существенно повысить его вычислительную эффективность и при этом сохраняющие свойство оптимальности. Сформулированная и доказанная теорема об оптимальности решений, отыскиваемых алгоритмом, использующим предложенные модификации.
3. Предложенные в работе модификации алгоритма многоагентного планирования, отыскивающие субоптимальные решения, позволяющие выбрать требуемый баланс между вычислительной эффективностью алгоритма и качеством отыскиваемых решений. Сформулированные и доказанные утверждения об ограниченной субоптимальности решений, отыскиваемых предложенными субоптимальными алгоритмами.

Обоснованность и достоверность результатов следует из корректного и строгого применения методов дискретной математики и математической логики при проведении исследования, в частности, при доказательстве теоретических свойств предложенных алгоритмов. Достоверность результатов дополнительно подтверждена результатами численных экспериментов, проведенных на данных, полученных из открытых источников, широко используемых в научном сообществе в области многоагентного планирования. Обоснованность и достоверность полученных в ходе исследования результатов также подтверждается их апробацией на ведущих научных конференциях и семинарах в области автоматического планирования.

Теоретическая и практическая значимость работы обусловлена тем, что разработанный алгоритм многоагентного планирования обладает уникальным набором свойств, включающим в себя учет возможности совершения действий произвольной продолжительности, а также гарантию оптимальности отыскиваемых решений. Свойство оптимальности было теоретически доказано. Благодаря учету возможности совершения агентами действий произвольной продолжительности алгоритм может находить решения, качество которых лучше, чем у существующих аналогов, а свойство оптимальности гарантирует, что отыскиваемые алгоритмом решения не могут быть улучшены (при тех же допущениях и входных данных).

Соответствие паспорту специальности. Диссертация выполнена в соответствии с паспортом научной специальности 1.2.3 «Теоретическая информатика, кибернетика». В соответствии с п.9 «Математическая теория исследования операций» в работе рассматривается задача многоагентного планирования и исследуются методы поиска её оптимальных решений. В соответствии с п.29 «Теоретические основы программирования, создания программных систем для новых информационных технологий» проведена разработка, реализация, теоретическое и экспериментальное исследование алгоритма, решающего задачу многоагентного планирования с возможностью совершения действий произвольной продолжительности, который может быть применен при разработке роботизированных интеллектуальных систем, в которых возникает задача согласованного перемещения группы роботов в общем рабочем пространстве.

Апробация работы. Результаты работы докладывались на следующих конференциях:

1. The 35th AAAI Conference on Artificial Intelligence, 2-7 February 2021, Online
2. The 28th International Joint Conference on Artificial Intelligence (IJCAI), 10-16 August 2019, Macao, China
3. IJCAI-19 Workshop on Multi-Agent Path Finding, 12 August 2019, Macao, China
4. The 14th International Symposium on Combinatorial Search, 26-30 July 2021, Online
5. Семнадцатая Национальная конференция по искусственному интеллекту с международным участием, КИИ-2019, 21-25 октября 2019 г., Ульяновск, Россия
6. Восемнадцатая Национальная конференция по искусственному интеллекту с международным участием, КИИ-2020, 10-16 октября 2020 г., Москва, Россия

Личный вклад. Автор принимал активное участие в исследованиях, в подготовке и представлении статей и докладов по теме работы. Программная реализация и тестирование алгоритмов, проведение модельных экспериментов, а также обработка полученных результатов производились автором лично. Доказательства теорем, приведенные в тексте диссертации, принадлежат автору и ранее не публиковались. Основные результаты и положения, выносимые на защиту, отражают персональный вклад автора.

Публикации. Основные результаты по теме диссертации изложены в 8 печатных изданиях, из которых 2[A1],[A2] работы изданы в журналах, рекомендованных ВАК, 5[A3],[A4],[A5],[A6],[A7] опубликованы в изданиях, индексируемых Scopus, в том числе 1[A3] статья опубликована в журнале первого квартиля по SJR, 1[A8] – в сборнике трудов конференции, индексируемый РИНЦ.

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** приводится постановка рассматриваемой задачи многоагентного планирования.

Задача задается набором $\langle G, \mathcal{M}, \mathcal{A}, Starts, Goals, K\rangle$, где $G = \langle V, E\rangle$ – граф, задающий возможные положения агентов, $\mathcal{M}$ – метрическое пространство, в которое вложен граф G, $\mathcal{A} = \{\mathcal{A}_w, \mathcal{A}_m\}$, $\mathcal{A}_w$ – множество действий ожиданий, $\mathcal{A}_m$ – множество действий перемещений, $Starts$, $Goals$ – множества стартовых и целевых положений, K – число агентов.

Каждое действие $a \in \mathcal{A}$ определено парой $\langle a_\varphi, a_D\rangle$, где a_D – продолжительность действия, а a_φ – функция, задающая положение агента в процессе исполнения действия. При этом действия перемещения привязаны к ребрам из множества E: $\forall a \in \mathcal{A} : \exists v, v' \in V : a_\varphi(0) = coord(v), a_\varphi(a_D) = coord(v'), (v,v') \in E$, где $coord(v)$ – координаты вершины v, определяющие её положение в метрическом пространстве $\mathcal{M}$. Действия ожидания также возможно совершать только в положениях соответствующих вершинам из множества V. В зависимости от типа действия a_φ, a_D определены следующим образом:

$$
\begin{aligned}
&\forall a \in A_m :\\
&\quad \forall t \in [0, a_D] : a_\varphi(t) = coord(v) + (coord(v') - coord(v))t/a_D\\
&\quad a_D = ||coord(v) - coord(v')||_2\\
&\forall a \in A_w :\\
&\quad \forall t \in [0, a_D] : a_\varphi(t) = coord(v)\\
&\quad a_D \in \mathbb{R}^+
\end{aligned}
\tag{1}
$$

Выражение 1 означает, что агент при исполнении действий перемещений движется с постоянной скоростью, причем такой, что время исполнения действия эквивалентно расстоянию между соответствующими вершинами. При исполнении действий ожиданий агент находится в вершине v на протяжении исполнения всего действия. Стоит отметить, что хотя множество всех действий перемещений $\mathcal{A}$ является конечным, множество всех действий ожиданий является бесконечным, так как допускается, что для действий ожиданий величина a_D может иметь произвольное положительное значение.

Определение 1. *Траектория π представляет собой последовательность пар действий и моментов времени $\pi_i = \{(a_1, t_1), \ldots, (a_k, t_k)\}$. При этом продолжительность траектории π_D и функция π_φ определены следующим образом:*

$$\pi_D = \sum_{a \in \pi} a_D \tag{2}$$

$$\pi_\varphi(t) = \begin{cases} a_{1\varphi}(t) & t \leq a_{1D} \\ \cdots & \cdots \\ a_{j\varphi}(t - (\pi[:j-1])_D) & (\pi[:j-1])_D < t \leq (\pi[:j])_D \\ \cdots & \cdots \\ a_{n\varphi}(t - (\pi[:n-1])_D) & (\pi[:n-1])_D < t \leq (\pi[:n])_D \\ a_{n\varphi}(a_{nD}) & t > (\pi[:n])_D \end{cases} \tag{3}$$

, где $\pi[:j]$ – это часть траектории π, состоящая из первых j действий.

Исполнив все действия, составляющие траекторию π_i, агент i перейдет из своего стартового положения $Starts(i)$ в целевое – $Goals(i)$. Выражение 3 определяет положение агента в любой момент времени в процессе исполнения заданной траектории. Для этого сперва определяется действие, которое агент совершает в момент времени t, после чего используется функция a_φ соответствующего действия. Последняя строка выражения означает, что после того как агент выполнил все действия, агент продолжает находиться в вершине, которая соответствует его целевому положению.

Находясь в положении вершины v, агент может совершить любое действие, начинающееся из этой вершины. При этом часть этих действий может приводить к конфликтам с другими агентами. Для определения конфликта введем функцию $InCollision : \{1, \ldots, K\} \times \{1, \ldots, K\} \times \mathcal{M} \times \mathcal{M} \rightarrow \{true, false\}$, где $InCollision(i, j, \pi_{i\varphi}(t), \pi_{j\varphi}(t)) = true$ означает, что между агентами i и j, которые в момент времени t находятся в положениях $\pi_{i\varphi}(t)$ и $\pi_{j\varphi}(t)$ соответственно, происходит конфликт. Без ограничения общности будем считать, что каждый агент представляет собой диск радиусом r. Тогда конфликт между агентами происходит в тех случаях, когда расстояние между ними меньше чем сумма их радиусов:

$$InCollision(i, j, \pi_{i\varphi}(t), \pi_{j\varphi}(t)) = \begin{cases} True & ||\pi_{i\varphi}(t) - \pi_{j\varphi}(t)||_2 < 2r \\ False & \text{В остальных случаях} \end{cases} \tag{4}$$

Для того, чтобы задача могла иметь решение, на множества $Starts$ и $Goals$ накладываются следующие ограничения:

3. Мощности множеств $Starts$ и $Goals$ должны быть эквивалентны: $|Starts| = |Goals|$.
4. Все элементы из множеств $Starts$ и $Goals$ должны принадлежать множеству вершин V: $n \in V, \forall n \in Starts \cup Goals$.
5. Между агентами, находящимися в своих стартовых или целевых положениях, не должно быть конфликтов:

$$\begin{gathered} InCollision(i,j, coord(Starts(i)), coord(Starts(j)) = False, \\ InCollision(i, j, coord(Goals(i)), coord(Goals(j)) = False, \\ \forall i,j = (1, K), i \neq j. \end{gathered} \tag{5}$$

Определение 2. *Пара траекторий π_i, π_j являются конфликтными, если существует момент времени, когда между агентами происходит конфликт в процессе исполнения этих траекторий:*

$$\exists t \in [0, \max(\pi_{iD}, \pi_{jD})] \;\; InCollision\big(i,j,\pi_{i\varphi}(t), \pi_{j\varphi}(t)\big) \tag{6}$$

Стоимость траектории эквивалента её продолжительности:

$$cost(\pi) = \pi_D \tag{7}$$

Качество (стоимость) решения оценивается с помощью суммарной стоимости всех траекторий:

$$cost(\Pi) = \sum_{i=1}^{K} cost(\pi_i) \tag{8}$$

Определение 3. *Совокупность траекторий $\Pi = \{\pi_1, ..., \pi_K\}$ является оптимальным решением задачи многоагентного планирования, если оно удовлетворяет следующим двум критериям:*

1. $\nexists \Pi' : cost(\Pi') < cost(\Pi)$

2. $\forall i,j \in [1,K] : \nexists t \in [0, \max(\pi_{iD}, \pi_{jD})] InCollision\big(i,j,\pi_{i\varphi}(t), \pi_{j\varphi}(t)\big)$

Таким образом, задача заключается в том, что имея заданный набор $\langle G, \mathcal{M}, \mathcal{A}, Starts, Goals, K \rangle$, необходимо построить совокупность неконфликтных траекторий, обладающую минимальной суммарной стоимостью, т.е. найти оптимальное решение Π.

<u>Глава 2</u> посвящена обзору существующих методов решения задачи многоагентного планирования. Данный обзор в первую очередь посвящен алгоритмам классического планирования, решающим задачу в детерминированный среде и графовым представлением пространства поиска.

Проведенный анализ литературы показал, что существующие алгоритмы многоагентного планирования, гарантирующие нахождение оптимального решения, опираются на допущение о дискретности времени и либо оперируют только действиями одинаковой продолжительности, либо с продолжительностями кратными шагу дискретизации. В связи с этим предлагается разработать алгоритм, который будет способен осуществлять планирование траекторий с возможностью осуществления действий произвольной продолжительности, что позволит повысить качество отыскиваемых решений.

Глава 3 посвящена описанию предлагаемого алгоритма многоагентного планирования CCBS (Continuous Conflict Based Search). Принцип его работы основан на подходе конфликтно-ориентированного поиска, который заключается в иерархическом (двухуровневом) подходе и декомпозиции задачи многоагентного планирования на множество более простых подзадач. На верхнем уровне алгоритма происходит перебор различных альтернативных наборов траекторий, а на нижнем - планирование индивидуальных траекторий агентов.

Первая ключевая особенность предлагаемого подхода – это само определение конфликта. В рассматриваемой постановке задачи, конфликт не всегда можно однозначно отнести к определенной вершине или ребру графа, т.к. он может произойти в произвольном месте метрического пространства, внутри которого оперируют агенты. В связи с этим предлагается определить конфликт через действия агентов. При этом существование конфликта между парами действий зависит не только от взаимного расположения агентов в процессе их исполнения, но и от моментов времени, в которые агенты начинают совершать соответствующие действия.

Определение 4. *Набор $\langle i,j,(a_i,t_i),(a_j,t_j)\rangle$ является конфликтом, обозначаемый как $InConflict((a_i,t_i),(a_j,t_j))$, в тех случаях, когда:*

$$\exists t \in [t_i,t_i+a_{iD}] \cap [t_j,t_j+a_{jD}] : \ InCollision\Big(i,j,a_{i\varphi}(t-t_i),a_{j\varphi}(t-t_j)\Big) \quad (9)$$

Для устранения конфликта необходимо наложить ограничение на действие агента. При этом недостаточно запретить агенту совершать действие в конкретный момент времени t. Ограничение накладывается на интервал времени $[t,t^u)$, где t^u – это первый момент времени, когда агент может начать совершать соответствующее действие без конфликта с действием другого агента.

Определение 5. *Для устранения конфликта $InConflict((a_i,t_i),(a_j,t_j))$ на одного из агентов накладывается ограничение, которое задается набором $\langle k,a_k,[t_k,t_k^u)\rangle$ и запрещает агенту k совершать действие a_k в течение конфликтного интервала $[t_k,t_k^u)$, где t_k^u – первый момент времени, когда агент k может начать совершать действие a_k не создавая конфликта с действием другого агента ($k=i,j$). В случае, если $k=i$, значение t_k^u определено следующим* *образом:*

$$t_i^u = argmin_{t \in [t_i, t_j + a_{j_D}]}\{InConflict((a_i,t),(a_j,t_j)) = False\} \tag{10}$$

В процессе работы алгоритм генерирует различные альтернативные наборы траекторий, которые отличаются друг от друга в зависимости от ограничений, наложенных на агентов. Для перебора различных альтернативных наборов траекторий алгоритм использует так называемое дерево ограничений CT (от англ. Constraint Tree), которое представляет из себя бинарное дерево, в котором каждая вершина задается набором $\langle N.cons, N.\Pi, N.cost\rangle$, где $N.cons$ – множество наложенных на агентов ограничений, $N.\Pi$ – набор траекторий, удовлетворяющих ограничениям $N.cons$, $N.cost$ – стоимость $N.\Pi$, т.е. суммарная стоимость всех траекторий, входящих в $N.\Pi$.

При инициализации дерева ограничений создается только одна вершина – корень. Корень дерева не содержит ограничений, т.е. траектория каждого агента фактически спланирована независимо от других агентов. На каждом шаге алгоритм выбирает из дерева вершину N, обладающую минимальной стоимостью. В случае, если выбранный набор траекторий $N.\Pi$ не содержит конфликтов, то решение задачи найдено. В противном случае выбирается один из конфликтов, содержащихся в $N.\Pi$, и для его устранения генерируются два потомка – N_i, N_j, где i,j – агенты, участвующие в выбранном конфликте. Множество ограничений, содержащееся в вершине N, включает в себя все ограничения вершин-предков, а также одно новое, наложенное либо на агента i, либо на агента j, устраняющее выбранный конфликт. Траектории агентов i и j перепланируются в соответствии с наложенными ограничениями. Этот процесс итеративно повторяется и продолжается до тех пор, пока не будет найден набор траекторий, не содержащий конфликтов.

Как уже упоминалось раннее, предлагаемый алгоритм CCBS является двухуровневым. Описанный выше процесс, отвечающий за перебор различных альтернатив – это верхний уровень алгоритма. На нижнем уровне алгоритма происходит перепланирование индивидуальных траекторий агентов с учетом наложенных ограничений. Для учета ограничений, содержащих в себе интервалы времени произвольной продолжительности, необходимо использовать алгоритм способный ими оперировать. За основу был взят алгоритм безопасно-интервального планирования SIPP (от англ. Safe Interval Path Planning) [4]. Отличительной особенностью алгоритма SIPP является описание состояния, которое задается парой $\langle v, [t_{begin}, t_{end}]\rangle$, где $[t_{begin}, t_{end}]$ – безопасный интервал, в течение которого агент может находиться в вершине v. В работе [4] было доказано, что если агент достигает каждое состояние в минимально возможное время, то найденная траектория обладает минимально возможной стоиимостью.

На вход алгоритму планирования индивидуальных траекторий помимо графа, стартового и целевого положений агента, подается набор ограничений, сформированных верхним уровнем алгоритма CCBS. Ограничения, наложенные на действия ожидания, преобразуются в конфликтные интервалы для соответствующих вершин, что приводит к возникновению новых состояний. Ограничения, наложенные на действия перемещения, учитываются в процессе генерации состояний-потомков. Предлагаемая модификация была названа CSIPP (Constrained SIPP). Подробное описание принципа его работы, псевдокод и анализ теоретических свойств приведен в тексте диссертации.

В разделе 3.3 исследуются теоретические свойства предлагаемого алгоритма CCBS.

Для любой решаемой задачи p существует множество решений $\mathbf{\Pi}(p)$. Множество решений делится на два подмножества: $\mathbf{\Pi}(p) = \mathbf{\Pi}_{opt}(p) \cup \mathbf{\Pi}_{subopt}(p)$, где $\mathbf{\Pi}_{opt}(p)$ – подмножество всех оптимальных решений, а $\mathbf{\Pi}_{subopt}(p)$ – подмножество всех субоптимальных решений:

$$\begin{gathered} \forall \Pi \in \mathbf{\Pi}_{opt}(p), \forall \Pi' \in \mathbf{\Pi}(p) : cost(\Pi) \leq cost(\Pi') \\ \forall \Pi \in \mathbf{\Pi}_{subopt}(p), \forall \Pi' \in \mathbf{\Pi}_{opt}(p) : cost(\Pi) > cost(\Pi') \end{gathered} \tag{11}$$

Определение 6. *Пусть дано множество решений $\mathbf{\Pi}(p)$ некоторой решаемой задачи p, а также пара интервальных ограничений $\langle i, a_i, [t_i, t_i^u) \rangle$ и $\langle j, a_j, [t_j, t_j^u) \rangle$. Будем называть пару ограничений согласованной, если любое решение удовлетворяет по крайней мере одному из ограничений, т.е. не включает в себя действие, на которое оно было наложено:*

$$\forall \Pi \in \mathbf{\Pi}(p) : (a_i, t_i') \notin \pi_i, \forall t_i' \in [t_i, t_i^u) \vee (a_j, t_j') \notin \pi_j, \forall t_j' \in [t_j, t_j^u), \pi_i, \pi_j \in \Pi \tag{12}$$

Лемма 1. *Для произвольного конфликта $InConflict((a_i, t_i), (a_j, t_j))$, ограничения $\langle i, a_i, [t_i, t_i^u) \rangle$ и $\langle j, a_j, [t_j, t_j^u) \rangle$, заданные в соответствии с Определением* **5**, *являются согласованной парой ограничений.*

Доказательство Леммы 1 строится от противного, т.е. предполагается, что существует такой набор траекторий, который не удовлетворяет ни одному из согласованной пары ограничений и при этом не содержит конфликтов.

Пусть дана произвольная задача многагентного планирования p, имеющая решение, $p \in \boldsymbol{P}^+$, а также некоторое решение этой задачи, найденное с помощью алгоритма CCBS, обозначаемое как Π_{CCBS}.

Теорема 1. $\forall p \in \boldsymbol{P}^+ : \Pi_{CCBS} \in \mathbf{\Pi}_{opt}(p)$ *– решение, найденное алгоритмом CCBS для любой решаемой задачи, является оптимальным.*

Доказательство. На первом шаге алгоритм CCBS создаст корень дерева ограничений N_0, содержащий набор траекторий, спланированных независимо. Корень дерева не содержит никаких ограничений, следовательно, любое решение из множества $\mathbf{\Pi}_{opt}$, во-первых, имеет стоимость не меньше чем стоимость $cost(N_0.\Pi)$, во-вторых, удовлетворяет всем его ограничениям ввиду их отсутствия.

Если $N_0.\Pi$ содержит конфликт $InConflict((a_i,t_i),(a_j,t_j))$, то на следующем шаге основного цикла работы алгоритма будут созданы две новых вершины дерева ограничений N_1, N_2, содержащие ограничения $\langle i,a_i,[t_i,t_i^u)\rangle$ и $\langle j,a_j,[t_j,t_j^u)\rangle$, наложенные на агентов i и j соответственно. При этом любое решение из множества $\mathbf{\Pi}_{opt}(p)$ удовлетворяет по крайней мере одному из ограничений, наложенному для устранения этого конфликта. Это утверждение является прямым следствием согласованности накладываемых ограничений (см. Лемма 1). Т.к. любое оптимальное решение удовлетворяет по крайней мере одному из ограничений, то его стоимость не может быть ниже минимальной стоимости среди имеющихся альтернативных частичных решений: $\forall\Pi \in \mathbf{\Pi}_{opt}(p) : cost(\Pi) \geq min(cost(N_1.\Pi), cost(N_2.\Pi))$.

Допустим, что эти утверждения выполняются на шаге k основного цикла работы алгоритма. Покажем, что они выполняются и на шаге $k+1$.

На шаге $k+1$ из списка $OPEN$ будет извлечена вершина N, имеющаяя наименьшую стоимость из всех, содержащихся в $OPEN$. Пусть в $N.\Pi$ найден конфликт $InConflict((a_l,t_l),(a_m,t_m))$. Для его устранения будут созданы две новых вершины дерева ограничений N_l, N_m, которые содержат по одному дополнительному ограничению $\langle l,a_l,[t_l,t_l^u)\rangle$ и $\langle m,a_m,[t_m,t_m^u)\rangle$. Если в множестве $\mathbf{\Pi}_{opt}(p)$ есть решения, удовлетворяющие множеству ограничений $N.cons$, то они будут удовлетворять по крайней мере одному из ограничений, добавленных в $N_l.cons$, $N_m.cons$. Следовательно их стоимость не может быть меньше минимальной стоимости среди новых созданных частичных решений: $\forall\Pi \in \mathbf{\Pi}_{opt}^N(p) : cost(\Pi) \geq min(cost(N_l.\Pi), cost(N_m.\Pi))$, где $\mathbf{\Pi}_{opt}^N(p)$ – подмножество всех оптимальных решений, удовлетворяющих ограничениям $N.cons$. Т.к. стоимости всех оптимальных решений эквивалентны, это неравенство справедливо для любого оптимального решения: $\forall\Pi \in \mathbf{\Pi}_{opt}(p) : cost(\Pi) \geq min(cost(N_l.\Pi), cost(N_m.\Pi))$. Вершины N_l и N_m добавляются в список $OPEN$, следовательно $\forall\Pi \in \mathbf{\Pi}_{opt}(p) : cost(\Pi) \geq argmin_{N'\in OPEN}\{cost(N'.\Pi)\}$, где $argmin_{N'\in OPEN}\{cost(N'.\Pi)\}$ – вершина с минимальной стоимостью решения из списка $OPEN$.

Если же набор траекторий $N.\Pi$ не содержит конфликтов, то искомое решение найдено. Учитывая неравенство $\forall\Pi \in \mathbf{\Pi}_{opt}(p) : cost(\Pi) \geq argmin_{N'\in OPEN}\{cost(N'.\Pi)\}$, где $argmin_{N'\in OPEN}\{cost(N'.\Pi)\}$ – это вершина N, а также определение множества всех оптимальных решений $\mathbf{\Pi}_{opt}$, решение $N.\Pi$ имеет ту же стоимость, что и любое решение из $\mathbf{\Pi}_{opt}$: $cost(N.\Pi) = cost(\Pi), \forall\Pi \in \mathbf{\Pi}_{opt}(p)$, т.е. $N.\Pi \in \mathbf{\Pi}_{opt}(p)$. □

Глава 4 посвящена описанию предлагаемых модификаций алгоритма.

Во-первых, был предложен оригинальный способ приоритизации конфликтов. В процессе работы алгоритма и генерации новых частичных решений, каждое из них может содержать множество различных конфликтов и выбор того, какой из них будет разрешен, влияет на дальнейший процесс поиска решения.

Начальный набор траекторий, содержащийся в корне дерева, состоит из траекторий, которые были спланированы без каких либо ограничений, и по сути имеет минимально возможную стоимость. Добавление новых ограничений в процессе работы алгоритма способно только увеличить стоимость решения. Так как алгоритм на каждом шаге выбирает из дерева ограничений вершину с минимальной стоимостью, целесообразно пытаться её увеличивать. Для этого было введено понятие добавочной стоимости, которое характеризует насколько сильно увеличится стоимость решения при разрешении определенного конфликта.

Определение 7. *Пусть дан конфликт $InConflict((a_i,t_i),(a_j,t_j))$, который содержится в некотором решении N. При разрешении этого конфликта будут созданы решения N_i и N_j, в которых было наложено ограничение на агента i или j, устраняющее этот конфликт. Разницу в стоимости между $cost(N.\Pi)$ и $cost(N_i.\Pi)$ обозначим как δ_i, а добавочную стоимость конфликта обозначим как $\Delta(InConflict((a_i,t_i),(a_j,t_j)))$. В таком случае добавочная стоимость конфликта равна минимальной разнице в стоимости между $cost(N.\Pi)$ и $cost(N_k.\Pi), k = \{i, j\}$:*

$$\Delta(InConflict((a_i,t_i),(a_j,t_j))) = min(\delta_i, \delta_j) \tag{13}$$

Имея значения добавочной стоимости каждого из конфликтов, содержащихся в решении, алгоритм выбирает то решение, которое обладает максимальной добавочной стоимостью. Критерий выбора конфликта может быть произвольным и не влияет на свойства алгоритма CCBS, т.к. накладываемые ограничения в любом случае остаются согласованными.

Следующая модификация, позволяющая повысить вычислительную эффективность алгоритма, заключается в использовании эвристической функции на верхнем уровне алгоритма. Оригинальный алгоритм CCBS в качестве критерия выбора вершины из дерева ограничений использует стоимость решения, что по сути является неинформированным поиском. Однако этот критерий может быть модифицирован и дополнительно учитывать то, насколько каждый из рассматриваемых наборов траекторий далек от искомого, не содержащего конфликты. Для этого было предложено использовать информацию о добавочной стоимости конфликтов. Добавление подобного рода эвристической функции на верхнем уровне алгоритма позволяет осуществить переход от неинформированного поиска к поиску по первому лучшему совпадению (англ. best-first search).

Предлагаемая эвристическая функция оценивает разницу в стоимости между рассматриваемым набором траекторий и искомым, который не содержит конфликтов, используя добавочную стоимость конфликтов. В работе было предложено два способа расчета эвристической функции. Первый способ основан на решении задачи линейного программирования, а второй – на эвристическом выборе конфликтов между непересекающимися парами агентов.

В алгоритме CCBS для устранения конфликтов используется подход, когда на одного из агентов, участвующих в конфликте, накладывается ограничение, запрещающее ему совершать конфликтное действие в течение определенного интервала времени. При этом рассматриваются оба альтернативных варианта, а соответствующая пара ограничений является согласованной, т.е. любое неконфликтное решение удовлетворяет по крайней одному из из этих ограничений. Однако, решение может удовлетворять и обоим ограничениям из этой пары и при этом быть оптимальным. Это приводит к тому, что в дереве ограничений могут содержаться вершины с одинаковым набором путей. Наличие идентичных решений в дереве приводит к замедлению работы алгоритма. Подход непересекающегося разделения использует два типа ограничений – *негативные* и *положительные*.

Определение 8. *Ограничение, которое запрещает агенту i начинать совершать действие a_i в течение интервала $[t_i, t_i^u)$, называется негативным и обозначается как $\overline{\langle i, a_i, [t_i, t_i^u) \rangle}$.*

Определение 9. *Ограничение, которое требует, чтобы агент i начал совершать действие a_i в течение интервала $[t_i, t_i^u)$, называется положительным и обозначается как $\langle i, a_i, [t_i, t_i^u) \rangle$.*

Пусть дана вершина дерева N, в решении которой содержится конфликт $InConflict((a_i, t_i), (a_j, t_j))$ и рассчитаны конфликтные интервалы $[t_i, t_i^u)$ и $[t_j, t_j^u)$. Для устранения этого конфликта создаются две новых вершины N_i, N_j, в которых для устранения этого конфликта накладываются ограничения $\overline{\langle i, a_i, [t_i, t_i^u) \rangle}$ и $\overline{\langle j, a_j, [t_j, t_j^u) \rangle}$ соответственно. Однако, помимо наложения негативных ограничений, в одной из ветвей накладывается положительное ограничение $\langle i, a_i, [t_i, t_i^u) \rangle$. Таким образом происходит непересекающееся разделение, т.к. агент i, на которого наложено положительное ограничение, в одной ветви может начать совершать действие a_i только внутри конфликтного интервала $[t_i, t_i^u)$, а в другой – только за его пределами.

Все три вышеописанных модификации могут быть скомбинированы вместе, в результате чего может быть получен алгоритм Improved CCBS (ICCBS), обладающий повышенной вычислительной эффективностью в сравнении с оригинальным алгоритмом и при этом сохраняющий свойство оптимальности.

Теорема 2. $\forall p \in \boldsymbol{P}^+ : \Pi_{ICCBS} \in \mathbf{\Pi}_{opt}(p)$ *– решение, найденное алгоритмом ICCBS для любой решаемой задачи, является оптимальным.*

Доказательство Теоремы 2 основывается на доказательстве оптимальности оригинального алгоритма CCBS. Дополнительные ограничения, накладываемые алгоритмом при использовании модификации непересекающегося разделения, являются согласованными. Способ выбора конфликта не влияет на свойства алгоритма, а оба предлагаемых способа расчета значений эвристической функции для верхнего уровня алгоритма обладают свойством допустимости. Т.к. ни одна из предложенных модификаций не нарушает свойств оригинального алгоритма, следовательно, решения, отыскиваемые алгоритмом ICCBS, также гарантированно являются оптимальными.

Иным способом повышения вычислительной эффективности алгоритма является отказ от поиска гарантированно оптимальных решений и переход к поиску субоптимальных решений. В разделе 4.2 приводится описание двух модификаций алгоритма CCBS, которые ищут субоптимальные решения и при этом гарантируют, что найденные ими решения не более чем w раз превысят стоимость оптимального решений, где w – фактор субоптимальности, значение которого задается пользователем.

Первая модификация, названная CCBS+FOCAL, базируется на подходе алгоритма A^*_ϵ [5] и использует дополнительный список FOCAL, который содержит подмножество вершин из списка OPEN. Стоимость решений, содержащихся в этом подмножестве не более чем в w раз превышает минимальную стоимость решения, содержащегося в OPEN. На каждом шаге алгоритм выбирает из списка FOCAL вершину, решение которой содержит минимальное число конфликтов.

Вторая модификация, названная CCBS+EES, базируется на подходе алгоритма Explicit Estimation Search (EES) [6] и использует два дополнительных списка – FOCAL и CLEANUP. Список CLEANUP по сути является регулярным списком OPEN, в котором все вершины отсортированы в порядке возрастания стоимости решения. Список OPEN в свою очередь отсортирован по $\hat{f}$-значению, которое равно сумме стоимости решения и значения эвристической функции $\hat{h}$: $\hat{f}(N) = cost(N.\Pi) + \hat{h}(N)$.

Теоретические свойства предложенных модификаций были исследованы и по их результатам были сформулированы и доказаны лемма и два утверждения:

Лемма 2. *Значение минимальной стоимости решения в списке OPEN монотонно не убывает.*

Доказательство Леммы 2 строится от противного. Делается предположение, что на некоторой итерации алгоритма k в список OPEN будет добавлена вершина N'_k, которая имеет стоимость $cost(N'_k.\Pi)$ меньшую, чем $cost(N_k.\Pi)$. Было показано, что данное предположение противоречит принципу построения дерева ограничений, используемое в алгоритме CCBS.

Утверждение 1. Стоимость решений, отыскиваемых алгоритмом CCBS+FOCAL не превышает стоимость оптимальных решений более чем в w раз, где w – фактор субоптимальности.

Утверждение 2. Стоимость решений, отыскиваемых алгоритмом CCBS+EES не превышает стоимость оптимальных решений более чем в w раз, где w – фактор субоптимальности.

Доказательство обоих этих утверждений строится на Лемме 2. Благодаря тому, что значение минимальной стоимости решений, содержащихся в списке OPEN, монотонно не убывает, решение, попавшее в список FOCAL на текущей итерации, будет удовлетворять ограничению на субоптимальность и на всех последующих итерациях работы алгоритма.

Глава 5 посвящена экспериментальным исследованиям разработанного алгоритма и его модификаций.

Экспериментальные исследования проводились на картах и заданиях взятых из открытой коллекции MovingAI[7]. Всего были протестированы 4 различные карты – `den520d`, `empty-16-16`, `rooms-32-32-4` и `warehouse-170-84-2-2`. Тестирование алгоритмов проводилось по принципу постепенного увеличения числа агентов в заданиях. Число агентов в заданиях увеличивалось до тех пор, пока алгоритм был способен найти решение в течение установленного лимита времени, составляющего 30 секунд. Каждая из карт представляет собой граф регулярной декмопозиции (ГРД) и была протестирована с 4 различными связностями графа – 4, 8, 16 или 32, что соотвтетсвует 2^k–связности с $k = 2{,}3{,}4{,}5$. Значение радиуса агентов r было установлено равным $\sqrt{2}/4$.

Раздел 5.3 посвящен изучению предложенных модификаций алгоритма CCBS – приоритизации конфликтов (PC), эвристикам верхнего уровня (HL), непересекающемуся разделению (DS). Всего было протестировано 5 различных версий алгоритма – CCBS, CCBS+PC, CCBS+DS, CCBS+DS+PC, CCBS+DS+PC+H. Результаты экспериментов, представленные на Рис. 1, показывают, что в большинстве случаев наибольшее число заданий решает версия, комбинирующая в себе все улучшения.

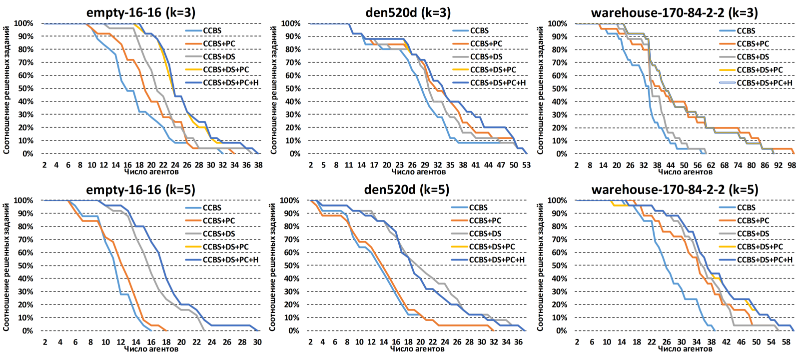

Рис. 1 — Сравнение различных модификаций алгоритма CCBS.

Заключительная серия экспериментов посвящена сравнению алгоритма CCBS с другими существующими алгоритмами, решающими задачу многоагнетного планирования, – E-ICTS, CBS+TAB, CBS-CT. Данные алгоритмы нельзя в полной мере назвать аналогами, т.к. они не предполагают возможности совершения действий произвольной продолжительности. При этом они могут осуществлять планирование с действиями различной продолжительности, однако требуют установления параметра шага дискретизации времени. По результатам проведенного предварительного тестирования, для данного параметра было выбрано значение 1/1000. Это значение позволяет максимально

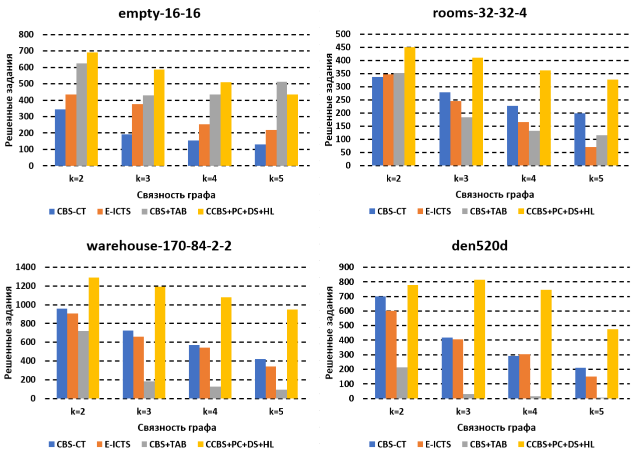

Рис. 2 — Сравнение алгоритмов CBS-CT, E-ICTS, CBS+TAB и CCBS+PC+DS+HL на различных картах и связностях графа по числу решенных заданий.

приблизить эти алгоритмы к CCBS по качеству решений и при этом не оказать существенного негативного влияния на скорость работы алгоритмов. При сравнении с другими алгоритмами использовалась лучшая из оптимальных модификаций CCBS – CCBS+PC+DS+HL. Все алгоритмы имели идентичные входные данные и одинаковый лимит времени работы – 30 секунд.

Результаты заключительной серии экспериментов показаны на Рис. 2. Почти во всех случаях предлагаемый алгоритм показывает результаты значительно превосходящие результаты всех других протестированных подходов. Единственный случай, в котором модифицированный алгоритм CCBS не показывает лучшие результаты наблюдается на карте `empty-16-16` с коэффициентом связности $k = 5$. В этом случае наибольшее число заданий сумел решить алгоритм CBS+TAB, однако на картах большого размера данный алгоритм показал худшие результаты из всех протестированных подходов вне зависимости от используемого коэффициента связности.

Более подробные результаты, касающиеся в том числе анализа качества отыскиваемых решений и вычислительной эффективности, тестирование алгоритма CCBS и его модифкаций на графах нерегулярной структуры, приведены в тексте диссертации.

В **заключении** представлены основные результаты диссертационной работы.

Основные результаты и выводы

1. Разработан алгоритм планирования совокупности неконфликтных траекторий для группы агентов с учетом возможности совершения действий произвольной продолжительности. Сформулирована и доказана

теорема, гарантирующая, что отыскиваемые алгоритмом решения являются оптимальнымы по стоимости.
2. Разработан ряд модификаций, в частности приоритизация конфликтов, непересекающееся разделение, а также эвристики верхнего уровня, позволяющие повысить вычислительную эффективность алгоритма. Сформулирована и доказана теорема, гарантирующая, что предложенные модификации не нарушают свойство оптимальности.
3. Разработаны две модификации алгоритма, которые позволяют отыскивать субоптимальные решения, обладают повышенной вычислительной эффективностью и позволяют устанавливать значение фактора субоптимальности, ограничивающие максимальное возможное превышение стоимости отыскиваемых решений в сравнении с оптимальными решениями. Ограниченная субоптимальность решений, отыскиваемых данными алгоритмами, была теоретически доказана.

Публикации автора по теме диссертации

A1. *Андрейчук, А. А.* Алгоритм планирования и согласования совокупности траекторий для группы интеллектуальных агентов [Текст] / А. А. Андрейчук // Искусственный интеллект и принятие решений. — 2018. — С. 72—85.

A2. *Андрейчук, А. А.* Эффективный поиск ограниченно-субоптимальных решений задачи многоагентного планирования [Текст] / А. А. Андрейчук // Искусственный интеллект и принятие решений. — 2022. — № 1. — С. 57—70.

A3. Multi-agent pathfinding with continuous time [Текст] / A. Andreychuk [и др.] // Artificial Intelligence. — 2022. — С. 103662.

A4. Improving Continuous-time Conflict Based Search [Текст] / A. Andreychuk [и др.] // Proceedings of The 14th International Symposium on Combinatorial Search, SoCS 2021. — 2021. — С. 145—146.

A5. Improving continuous-time conflict based search [Текст] / A. Andreychuk [и др.] // Proceedings of the AAAI Conference on Artificial Intelligence. Т. 35. — 2021. — С. 11220—11227.

A6. *Andreychuk, A.* Multi-agent path finding with kinematic constraints via conflict based search [Текст] / A. Andreychuk // Lecture Notes in Artificial Intelligence. Т. 12412. — Springer, Cham, 2020. — С. 29—45.

A7. Multi-agent pathfinding with continuous time [Текст] / A. Andreychuk [и др.] // Proceedings of the 28th International Joint Conference on Artificial Intelligence. — 2019. — С. 39—45.

A8. *Андрейчук, А. А.* Исследование алгоритма конфликтно-ориентированного поиска для решения задачи планирования совокупности неконфликтных траектория для множества агентов [Текст] / А. А. Андрейчук // Семнадцатая Национальная конференция по искусственному интеллекту с международным участием. КИИ-2019 (21–25 октября 2019 г., г. Ульяновск, Россия). Сборник научных трудов. Т. 1. — 2019. — С. 93—101.

Список литературы

1. *Xiaohe, D.* The annual business volume of express delivery in China exceeded 100 billion [Электронный ресурс] / D. Xiaohe. — 2021. — URL: http://www.news.cn/2021-12/08/c_1128141875.htm (дата обр. 25.05.2022).
2. *Banker, S.* New Robotic Solutions For The Warehouse [Text] / S. Banker. — 2017. — URL: https://www.forbes.com/sites/stevebanker/2017/03/07/new-robotic-solutions-for-the-warehouse (visited on 05/25/2022).
3. *Zhang, D.* Artificial intelligence in E-commerce fulfillment: A case study of resource orchestration at Alibaba's Smart Warehouse [Текст] / D. Zhang, L. Pee, L. Cui // International Journal of Information Management. — 2021. — Т. 57. — С. 102304.
4. *Phillips, M.* Sipp: Safe interval path planning for dynamic environments [Текст] / M. Phillips, M. Likhachev // 2011 IEEE International Conference on Robotics and Automation. — IEEE. 2011. — С. 5628—5635.
5. *Pearl, J.* Studies in semi-admissible heuristics [Текст] / J. Pearl, J. H. Kim // IEEE transactions on pattern analysis and machine intelligence. — 1982. — № 4. — С. 392—399.
6. *Thayer, J.* Learning inadmissible heuristics during search [Текст] / J. Thayer, A. Dionne, W. Ruml // Proceedings of the International Conference on Automated Planning and Scheduling. Т. 21. — 2011. — С. 250—257.
7. *Sturtevant, N. R.* Benchmarks for grid-based pathfinding [Текст] / N. R. Sturtevant // IEEE Transactions on Computational Intelligence and AI in Games. — 2012. — Т. 4, № 2. — С. 144—148.

Андрейчук Антон Андреевич

Методы конфликтно-ориентированного поиска для планирования совокупности безопасных траекторий мобильных агентов с учетом возможности совершения действий произвольной продолжительности

Диссертация посвящена исследованию методов и алгоритмов решения задачи многоагентного планирования. Рассмотрена постановка задачи, допускающая возможность совершения действий произвольной продолжительности, и предложен алгоритм её решения. Были исследованы и доказаны его теоретические свойства. Предложен ряд модификаций, позволяющих повысить его вычислительную эффективность и при этом сохранить свойство оптимальности. Также были предложены две модфикации алгоритма, позволяющие отыскивать ограниченно субоптимальные решения с возможностью регулирования фактора субоптимальности.

Andreychuk Anton

Conflict-Based Search Methods for Solving Multi-Agent Path Finding Problem with Ability of Performing Actions of Arbitrary Duration

The dissertation is devoted to the study of methods and algorithms for solving the multi-agent path finding problem. The considered problem statement admits the ability of performing actions of arbitrary duration, and an algorithm that is able to solve it is proposed. Its theoretical properties were investigated and proved. A number of modifications are proposed to increase its computational efficiency and at the same time preserve the optimality property. Two modifications of the algorithm were also proposed, which allow finding bounded suboptimal solutions with the ability to control the suboptimality factor.