

**Budko Natalia Aleksandrovna** – RDIRCS SFU; e-mail: natalia.tb13@mail.ru; Taganrog, Russia; phone: +79185670210; engineer; graduate student the department of electrical engineering and mechatronics.

**Budko Artem Yurievich** – e-mail: aptem\_budko@mail.com; phone: +79281545833; senior research officer; cand. of eng. sc.; associate professor the department of electrical engineering and mechatronics.

**Medvedev Mikhail Yurievich** – e-mail: medvmihal@gmail.com; phone: +79198927349; leading researcher; dr. of eng. sc.; professor, head the department of electrical engineering and mechatronics.

УДК 007.52:629.3.05

DOI 10.18522/2311-3103-2022-3-170-192

**В.Х. Пшихопов, М.Ю. Медведев, Д.О. Бросалин, М.А. Васильева,  
Б.В. Гуренко, Н. Хамдан**

**ИССЛЕДОВАНИЕ МЕТОДОВ ПЛАНИРОВАНИЯ ДВИЖЕНИЯ  
В ДВУМЕРНЫХ КАРТОГРАФИРОВАННЫХ СРЕДАХ\***

*Исследуются задача планирования движения в двумерных картографированных средах. Проводится обзор и анализ известных алгоритмов планирования, базирующихся на диаграммах Вороного, вероятностной дорожной карте, быстро растущих случайных деревьев, алгоритмах Дейкстры,  $A^*$ ,  $D^*$  и их модификациях, искусственных потенциальных полях и интеллектуальных эвристиках. На основе проведенного анализа делается вывод о том, что классические методы в динамических средах требуют значительных затрат по времени расчетов и объему используемой памяти. Делается вывод об актуальности разработки алгоритмов, повышающих эффективность известных методов планирования. В этой связи данная статья посвящена разработке модифицированного алгоритма быстро растущих случайных деревьев и исследованию его эффективности по сравнению с известными методами. В статье представлен модифицированный алгоритм быстро растущих случайных деревьев, отличающийся тем, что при проверке наличия пути в новый потенциальный узел графа проверяется путь в некоторую область возле указанного узла. Это позволяет снизить количество узлов в строящемся дереве. Разработанный алгоритм вначале сравнивается с традиционным алгоритмом быстрорастущих случайных деревьев. Сравнение производится по времени расчета траектории, объему требуемой памяти, длине пути и проценту ситуаций, в которых успешно найдена траектория в целевую точку. Далее осуществляется сравнение разработанного алгоритма с алгоритмами планирования других классов. При исследовании используются репрезентативные выборки численных экспериментов и различные среды, отличающиеся плотностью расположения препятствий и наличием лабиринтов. Также проводится исследование алгоритмов планирования с использованием результатов экспериментов на наземном колесном роботе. По результатам численных и реальных экспериментов делаются выводы о преимуществах и недостатках разработанного алгоритма планирования движения и о целесообразности его применения в различных средах.*

*Планирование движения; двумерная среда, метод случайных деревьев, оптимизация алгоритмов планирования, сравнительный анализ.*

---

\* Работа при поддержке РФФИ, проект № 22-29-00533.

**V.Kh. Pshikhopov, M.Yu. Medvedev, D.O. Brosalin, M.A. Vasileva, B.V. Gurenko,  
N. Hamdan**

**STUDY OF PATH PLANNING METHODS IN TWO-DIMENSIONAL MAPPED ENVIRONMENTS**

*The article studies the problem of motion planning in two-dimensional mapped environments. The review and analysis of known planning algorithms based on Voronoi diagrams, probabilistic road maps, rapidly growing random trees, Dijkstra algorithms, A\*, D\* and their modifications, artificial potential fields and intelligent heuristics are carried out. Based on the analysis, it is concluded that classical methods in dynamic environments require significant costs in terms of calculation time and the amount of memory used. The conclusion is made about the relevance of the development of algorithms that increase the efficiency of known planning methods. In this regard, this article is devoted to the development of a modified algorithm of rapidly growing random trees and the study of its effectiveness in comparison with known methods. The article presents a modified algorithm for rapidly growing random trees, characterized in that when checking for a path to a new potential node of the tree, the path to some area near the specified node is checked. This reduces the number of nodes in the tree under construction. The developed algorithm is first compared with the traditional algorithm of fast-growing random trees. The comparison is made by the trajectory calculation time, the amount of memory required, the path length and the percentage of situations in which the trajectory to the target point was successfully found. Next, the developed algorithm is compared with the planning algorithms of other classes. The study uses representative samples of numerical experiments and various environments that differ in the density of obstacles and the presence of mazes. A study of planning algorithms using the results of experiments on a ground-based wheeled robot is also being conducted. Based on the results of numerical and real experiments, conclusions are drawn about the advantages and disadvantages of the developed algorithm of motion planning and the feasibility of its application in various environments.*

*Motion planning; two-dimensional environment, random tree method, optimization of planning algorithms, comparative analysis.*

**Введение.** Развитие технологий позволяет повышать создавать робототехнические комплексы, способные функционировать при минимальном участии оператора. На сегодняшний день мобильные роботы подошли к уровню развития, который позволяет применять их автономно. В этой связи особую важность представляет задача оптимального планирования пути в реальном времени. Планирование пути – сложная и многогранная задача, которая не имеет универсального решения. В этой связи развитие и модификация различных алгоритмов, используемых при планировании пути, применимых и оптимальных для различных условий актуально.

В общем случае, задача планирования пути заключается в расчете последовательности состояний подвижного объекта, которые обеспечивают его перевод из текущего в целевое состояние. При этом под состоянием будем понимать минимально-необходимый набор значений переменных величин системы, способных однозначно и единственным образом определить положение системы в любой момент времени  $t$  [1].

Отметим, что переход из текущего в целевое состояние, обычно, можно осуществить множеством способов. В это связи предъявляются дополнительные требования к тому, как должен быть выполнен переход в целевое состояние. Как правило, предъявляются требования физической реализуемости и оптимальности траектории движения. К требованиям физической реализуемости относится, например, требование перехода в целевую точку за конечное время и при ограниченных управляющих воздействиях, требование отсутствия коллизий. Под оптимальностью траектории подразумевается минимизация или максимизация заданного критерия качества, например, времени движения, длины пути, затрачиваемой энергии и пр.

Таким образом, под алгоритмом планирования понимается набор правил, на основании которых для каждого момента планирования рассчитывается состояние, в которое должен перейти подвижный объект. В результате работы алгоритма планирования, на выходе, формируется набор состояний, который может быть преобразован в набор действий агента.

Согласно [2] планирование пути подвижного объекта в общем виде можно представить в виде последовательного выполнения действий, включающих расчет свободного от столкновений пути, сглаживание пути с учетом свойств объекта, расчет управляющих воздействий. Наиболее часто планирование пути осуществляется в конфигурационном пространстве [2]. Основным достоинством приведения задачи к виду конфигурационного пространства является то, что такой метод сводит задачу от вычислений для твердого тела к вычислениям для материальной точки, что значительно упрощает процесс [3].

В частности, для подвижного объекта в двумерном Евклидовом пространстве все конфигурации могут быть описаны вектором  $(x, y, \varphi)$ , в котором  $x, y$  – линейные координаты объекта, а  $\varphi$  – его угловая ориентация. Конфигурации пространства подразделяются на подмножества {«Свободное»}, {«Препятствия»} и {«Неопределенное»}.

**Обзор методов планирования движения.** В данной статье рассматриваются популярные алгоритмы планирования, которые можно разбить на четыре группы: а) методы, базирующиеся на применении теории графов (методы быстрорастущих случайных деревьев (RRT), вероятностных дорожных карт (PRM), диаграмм Вороного, диаграмм видимости); б) методы на основе клеточной декомпозиции (алгоритмы Дейкстры,  $A^*$ ,  $D^*$ ); в) методы на основе потенциальных полей (виртуальное силовое поле, метод Ньютоновских полей); методы на основе интеллектуальных технологий (нейронные сети, эволюционные вычисления, нечеткие вычисления). Рассмотрим более подробно некоторые алгоритмы из каждой группы.

Сегодня часто используются методы, основанные на клеточной декомпозиции, которые используют деление карты на ячейки. Это позволяет использовать дискретные итерационные алгоритмы построения пути. В случае представления конфигурационного пространства в виде решетки, популярными методами планирования является алгоритм Дейкстры и, базирующиеся на нем  $A^*$  и  $D^*$  [4 – 6].

Алгоритм  $A^*$  является развитием алгоритма Дейкстры. В  $A^*$  вычислительная сложность уменьшена за счет эвристической функции оценки стоимости пути. В то же время данный алгоритм требователен к объему используемой памяти, и его вычислительная сложность значительно растет с ростом числа ячеек на карте. В этой связи существует достаточно большое число модификация алгоритма  $A^*$ , которые нацелены на уменьшение указанных недостатков. К таким модификациям относятся  $A^*$  с итеративным углублением [7],  $A^*$  с ограничением памяти [8], иерархический  $A^*$  [9] и  $A^*$  с динамическим изменением весов ребер [10, 11]

Алгоритм  $D^*$  [6] оптимизирован для планирования пути в динамической среде, когда информация обновляется в процессе движения робота, например, при обнаружении новых препятствий.

Отметим, что рассмотренные методы планирования пути, как правило дают негладкие траектории. В этой связи они дополняются различными алгоритмами сглаживания траекторий движения [12, 13].

Гладкие траектории движения обеспечиваются при использовании для планирования метода потенциальных полей [14, 15]. В связи со своей простотой, высокой вычислительной эффективностью и указанной возможностью получения гладких траекторий движения метод потенциальных полей получил широкое рас-

пространение [16–19]. Однако имеется ряд проблем, к которым относятся локальные минимумы, сложность учета динамики подвижного объекта и обоснованный выбор отталкивающих и притягивающих сил [20]. Существующие подходы решают указанные проблемы в частных случаях. В этой связи потенциальные поля часто используются как составная часть алгоритмов планирования [21–23].

Популярными в последнее время стали методы планирования с использованием интеллектуальных технологий, в частности обучающихся систем [24–27]. Достоинством нейросетевых методов планирования является их высокая адаптивность к неопределенным динамическим средам. Основным недостаток таких систем планирования – высокие требования к бортовым вычислительным системам [16].

В данной статье изучаются графовые методы, обладающие рядом преимуществ, к которым относятся их масштабируемость, отсутствие длительной настройки, возможность учитывать кинематические и динамические ограничения, возможность применения в динамических средах и в условиях увеличения области видимости карты, возможность применения в многомерных пространствах. Имеются и недостатки графовых подходов, которые требуют большого времени и объема памяти для расчетов и, часто, медленно сходятся к оптимальному решению.

Алгоритмы, базирующиеся на представлении состояний подвижного объекта в виде графа, широко распространены [4]. К ним относятся алгоритм планирования с построением графа видимости [28], диаграмм Вороного [29], быстрорастущих случайных деревьев [30], дорожных карт [31].

Узлами графа видимости являются множество вершин полигональных препятствий и начальное и конечное положение робота. Ребрами графа являются связи между вершинами, которые не пересекаются с препятствиями. Основным ограничением алгоритма построения графа видимости является его высокая вычислительная сложность, которая оценивается как  $O(n^2 \log(n))$ , где  $n$  – число узлов. Высокая сложность построения графа определяется необходимостью проверять пересечения с препятствиями всех прямых, соединяющих попарно вершины. Тем не менее данный метод сегодня может успешно применяться в прикладных задачах.

Так в работе [32] рассмотрена задача планирования пути безэкипажного катера в акватории побережья Южной Кореи. Из-за большого числа островов граф содержит 54625 вершин. В этой связи прямое построение графа видимости нецелесообразно. Поэтому в статье вначале применяется адаптивное разбиение сетки, используя технологию дерева квадрантов (quadtree) [33]. Далее алгоритмом Дейкстры находится путь на дереве квадратов и строится граф видимости с использованием полученных точек пути. Кратчайший путь ищется на полученном графе видимости. В статье приводятся результаты моделирования, которые показывают более высокое быстродействие при использовании технологии дерева квадрантов по сравнению с остальными сетками, например, регуляторной сеткой. Однако сравнения метода построения графа видимости с другими методами не проведено.

В работе [34] граф видимости используется совместно с алгоритмом быстрорастущих случайных деревьев. Вначале используется двунаправленный метод случайных деревьев, в ходе которого строятся графы из начальной и целевой точек. Возможность соединить указанные деревья осуществляется с помощью графа видимости. Предложенный метод продемонстрирован на примерах с небольшим числом препятствий, т.е. в среде, которая не представляет большой вычислительной сложности для построения графа видимости.

Таким образом, графы видимости могут применяться для планирования пути при небольшом количестве препятствий или при предварительной обработке карты.

Диаграмма Вороного представляет собой разбиение плоскости с  $n$  центрами на множество выпуклых многогранников таким образом, чтобы любая точка внутри многогранника находилась ближе к своему центру, чем к другим центрам

[29, 35]. Это свойство позволяет планировать путь, наиболее удаленный от препятствий, если центры последних выступают в качестве точек для диаграммы Вороного. Вычислительная сложность построения диаграммы Вороного оценивается как  $O(n \cdot \log(n))$  [35], поэтому они, как и графы видимости, используются в совокупности с другими методами планирования или с дополнительными алгоритмами, которые позволяют уменьшить число рассматриваемых точек и применять алгоритм в динамических средах.

Так, в работах [36, 37] рассматривается задача планирования пути с использованием диаграмм Вороного с динамическим обновлением текущей и целевой точек. В [36] используется сочетание диаграмм Вороного и дорожных карт для построения пути. Предложен алгоритм, который осуществляет удаление старых и добавление новых текущей и целевой точки. Полученные результаты сравниваются путем моделирования с методом планирования, использующим граф видимости. В работе [37] рассматривается задача планирования пути, состоящего из прямых линий, с использованием метода обобщенных диаграмм Вороного. При этом также применяется процедура добавления и удаления точек, которая позволяет устранять лишние углы при планировании маршрута. Сравнение с другими методами аналогичного класса осуществляется на относительно простой карте.

В работе [38] метод диаграмм Вороного используется для построения начального пути, который затем оптимизируется дополнительными алгоритмами, включая сплайны и метод потенциальных полей. В данной статье рассмотрены среды с препятствиями, представленными окружностями. В силу того, что последние могут пересекаться, возникает проблема локальных минимумов, которую для указанного типа среды авторам удастся решить.

Таким образом, основными недостатками метода диаграмм Вороного является его низкая вычислительная эффективность, которая требует применения дополнительных алгоритмов и обработки карты.

Метод дорожных карт [39] применяется при решении задач локального и глобального планирования. Планирование здесь проводится в два этапа. На первом этапе осуществляется подготовка карты посредством генерации на ней случайным образом заданного числа путевых точек – узлов будущего графа. При генерации новой точки осуществляется проверка на пересечения новой точки с препятствием – сохраняются только те точки, которые сгенерированы в свободной от препятствий области карты. На втором этапе осуществляется Построение свободных от столкновений ребер между сгенерированными ранее вершинами. По полученному графу осуществляется поиск пути между любыми двумя узлами.

В силу высоких вычислительных затрат, основные усилия исследователей направлены на повышение эффективности метода дорожных карт, особенно для узких сред [40–42, 44]. Ключевую роль здесь играет способ генерирования новых потенциальных узлов графа. Так в работе [40] предложен метод планирования пути, использующий алгоритм случайной дорожной карты. Разработана новая процедура генерирования узлов в ограниченной области, эффективность которой показана сравнением с оригинальным методом случайной дорожной карты в стесненном пространстве. В работе [41] также изучается проблема повышения метода дорожных карт в узких пространствах. Для этого используются виртуальные поля, определяющие пространства, в которых генерируются потенциальные новые узлы дорожной карты. Проведены численные эксперименты, подтверждающие снижение времени построения траектории методом дорожной карты в узких местах.

Аналогичная проблема рационального создания дорожной карты решается в работах [42, 44]. В работе [42] предлагается метод нахождения ближайших соседей, использующий хеширование, зависящее от текущего положения [43]. Исследована эффективность методом моделирования по сравнению с другими метода-

ми, включая полный перебор. Недостатком работы является тот факт, что начальная дорожная карта задана пустой, что ограничивает использование метода для перестроения уже имеющегося пути. Такая необходимость может возникнуть в динамической среде.

В работе [44] предложена новая стратегия построения дорожной карты, нацеленная на максимизацию ее видимости при генерировании новых узлов. При этом узлы генерируются в узком коридоре на некотором расстоянии друг от друга. Смещение узлов осуществляется вдоль узкого коридора. В статье показана эффективность предложенной стратегии в трубах средах с окнами и другими узостями.

Таким образом, метод дорожных карт является распространенным при планировании пути, однако его вычислительная эффективность снижается в лабиринтах, коридорах, проходах и других узких местах.

Метод быстрорастущих случайных деревьев (RRT – rapidly random trees) вычислительно более эффективен по сравнению с методом дорожных карт. Он позволяет строить физически реализуемый путь, учитывая динамику мобильного робота [45, 46]. Однако основные ограничения данного метода для использования в динамических средах – низкая сходимости к оптимальному решению и высокие требования к используемой памяти. По этой причине повышение вычислительной эффективности позволит расширить область применения метода случайных быстрорастущих деревьев. В этой связи рассмотрим более детально работы, направленные на повышение эффективности указанного метода.

Проблема применения алгоритм RRT в условиях частичной неопределенности рассмотрена в работе [45], в которой предложено делить область функционирования на отдельные регионы. Для нахождения путей между регионами используется алгоритм RRT. Планирование пути внутри локальных регионов осуществляется простым алгоритмом, использующим отрезки прямых линий. Неопределенность модели мобильного робота учтена путем добавления возмущающего входа типа «белый шум» [47]. Неопределенность карты учтена введением вероятности того, что ячейка карты занята препятствием. Основным результатом, представленным в данной работе – метод, гарантирующий успешное достижение целевой точки с вероятностью не менее заданного значения  $p_{min}$ . Авторы работы [45] приводят результаты анализа и моделирования движения робота, представленного уравнениями кинематики материальной точки в статической среде. Основным недостатком метода – значительный рост вычислительной сложности при увеличении значения  $p_{min}$ .

Проблема уменьшения вычислительной сложности и требуемой памяти рассмотрена также в работах [46, 48]. В работе [46] комбинируются алгоритм RRT и искусственные потенциальные поля (метод P-RRT\*). В этом методе случайное дерево строится в направлении уменьшения потенциального отталкивающего поля, что позволяет повысить скорость сходимости к оптимальной траектории. В [46] используется улучшенная версия алгоритма RRT\* [49]. В работе [48] также используются потенциальные поля для генерирования новых точек, однако дерево растет как из начальной, так и из конечной точек. При этом оба дерева создают притягивающие поля друг для друга. Вычислительная сложность метода остается достаточно высокой, кроме того методы [46, 48] дают негладкие траектории.

Двунаправленный рост случайного дерева в алгоритме RRT также используется для повышения скорости сходимости к оптимальному решению в работе [50]. Основные результаты работ [48, 50] указывают на то, что двунаправленный случайный поиск в комбинации с потенциальными полями позволяет уменьшить число итераций для нахождения кратчайшего пути. Также указанные алгоритмы позволяют двигаться в узких коридорах, не попадая в локальные минимумы. Для уменьшения объема требуемой памяти деревья вычисляются последовательно во времени, что увеличивает время расчетов.

В работе [51] используется двухэтапная процедура планирования пути в динамической среде. Глобальный планировщик вычисляет начальный путь первого приближения, используя алгоритм RRT. Далее применяется динамический планировщик, который изменяет положение узлов пути, используя отталкивающие и инерционные силы. Для учета движущихся препятствий используется процедура прогнозирования. В этой работе использована эвристическая оптимизационная функция, выбор которой зависит от различных требований и ограничений. Основная проблема метода – рациональный выбор этой функции.

В работе [52] рассмотрена проблема вычисления оптимального пути при необходимости посетить заданное множество точек. Для решения этой проблемы в среде с препятствиями строится лес деревьев [53], в котором деревья стартуют из заданных целевых точек. Указанный лес строится до тех пор, пока деревья не соедятся друг с другом. В данной работе развивается метод, предложенный в работе [54], в которой лес деревьев оптимизируется благодаря свойствам очереди узлов. В [53] метод исследуется в двумерной и трехмерном пространствах, используя численное моделирование. Достигается снижение вычислительной сложности и используемой памяти за счет того, что каждое дерево строится отдельно друг от друга. В этой связи узлы отдельных деревьев хранятся в отдельных массивах.

В работе [55] предложен улучшенный алгоритм RRT\* – быстрый алгоритм Q-RRT\*, который рассматривает в качестве возможных родительских вершин не только вершин, содержащихся в гиперсфере, как в RRT\*, но и их собственные родительские вершины с точностью до определенного пользователем параметра. Это может приводить к получению путей с меньшей стоимостью, чем у RRT\*. Предложенный алгоритм позволяет устранять небольшие повороты и быстрее работает в узких проходах. Также разработаны предвзято-жадный RRT алгоритм [56] и мета-алгоритм [57], которые позволяют более эффективно строить начальный путь.

В работе [13] предложен двухэтапный алгоритм планирования, в котором для учета динамических ограничений мобильного робота на карту добавляются виртуальные препятствия, перекрывающие невыполнимые участки движения [59]. В качестве алгоритма глобального планирования используется модифицированная версия алгоритма RRT – алгоритм MPN-RRT. В этом алгоритме, в отличие от оригинальной версии, используется несколько родительских узлов, что уменьшает длину запланированной траектории по сравнению с исходной версией RRT с одним узлом. Методами численного моделирования показано, что использование двух родительских узлов позволяет уменьшить среднюю длину пути для городской среды с низкой плотностью застройки. Для решения проблемы медленной сходимости алгоритмов, основанных на случайном поиске и сглаживании путей, алгоритм RRT дополнен алгоритмом локальной оптимизации. Алгоритм RRT ищет глобальный путь, который сглаживается и оптимизируется итеративным локальным алгоритмом, предложенном в [59, 60].

Таким образом, различные варианты алгоритма RRT оптимизируются путем настройки параметров и модификации способов решения различных задач в ходе поиска пути. В общем случае, оптимизируются количество допустимых итераций построения дерева, рабочая область на карте, радиус исследуемой области вокруг узла дерева. Также варьируются закон выбора случайной точки на карте, в сторону которой растет дерево, алгоритм построения нового ребра и алгоритм выбора нового родительского узла.

В данной статье ставится задача разработать модифицированный алгоритм RRT, по времени расчета более эффективный, чем классический алгоритм RRT. Ставится задача провести также исследование предложенного алгоритма в сравнении с классическим методом RRT и другими методами планирования движения.

**Математическое описание движения робота и среды.** В общем случае модель робота может быть представлена уравнениями кинематики и динамики твердого тела и исполнительных механизмов [61, 62]. При планировании движения распространенной формой математической модели являются уравнения кинематики и динамики материальной точки, для наземного колесного робота могут быть представлены в виде [13, 16, 21]

$$\dot{x} = (\omega_l + \omega_r)r \cos \varphi, \quad (1)$$

$$\dot{y} = (\omega_l + \omega_r)r \sin \varphi, \quad (2)$$

$$\dot{\varphi} = (\omega_l - \omega_r) \frac{r}{l}, \quad (3)$$

$$\dot{\omega}_l = k_1 u_l, \quad (4)$$

$$\dot{\omega}_r = k_2 u_r, \quad (5)$$

где  $(x, y)$  – линейные координаты робота;  $\varphi$  – угол ориентации робота;  $\omega_l, \omega_r$  – скорости вращения левого и правого колес;  $u_l, u_r$  – управляющие воздействия;  $r$  – радиус колес;  $l$  – расстояние между колесами;  $k_1, k_2$  – постоянные коэффициенты.

Среда функционирования робота представлена на рис. 1.

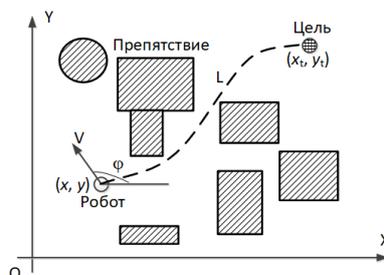


Рис. 1. Среда функционирования робота

Робот с координатами  $(x, y)$  обозначен на рис. 1 незаштрихованной окружностью. Цель с координатами  $(x_i, y_i)$  представлена в виде окружности с двойной штриховкой. Препятствия показаны заштрихованными прямоугольниками и окружностями. Отдельные препятствия могут образовывать сложные конфигурации, тушки, стенки, лабиринты и пр. Т.к. препятствия могут образовывать произвольные конфигурации, то нет большой разницы, какую форму имеют единичные препятствия. В силу этого далее будут использованы обе формы – прямоугольники и окружности. Траектория показана пунктирной линией.

Задача заключается в планировании траектории движения  $L$ , свободной от коллизии, таким образом, чтобы:

- ◆ при заданном времени расчета  $tp$  длина траектории  $L$  была минимизирована;
- ◆ при выполнении условия  $L \leq L_{\text{доп}}$  время расчета не превышало заданного значения  $t_{\text{доп}}$ .

В качестве базового алгоритма принимается алгоритм RRT, который позволяет обойти проблему локальных минимумов, может применяться в неопределенных условиях, но, как отмечалось ранее, требует длительного времени расчета.

**Модифицированный алгоритм быстрорастущих случайных деревьев.** Для повышения эффективности классического метода быстрорастущих случайных деревьев применяется следующая эвристика, используемая с целью компенсации существующих недостатков классического метода, позволяющая сократить время сходимости дерева при поиске пути и сократить объем требуемой памяти. Такая оптимизация достигается за счет следующих принципов:

◆ вместо проверки свободного пути из текущего узла в конечную точку осуществляется проверка наличия пути из текущей точки в свободную от препятствий область вокруг этой конечной точки, как показано на рис. 2;

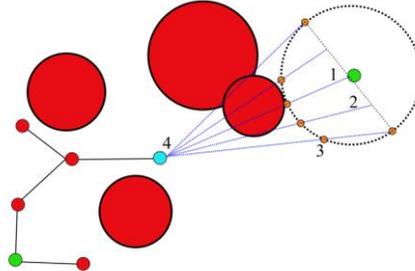


Рис. 2. Проверка пути в область вблизи целевой точки

◆ вместо проверки свободного пути из потенциального узла в родительский узел осуществляется проверка наличия пути из текущего потенциального узла в свободную от препятствий область вокруг этого родительского узла.

На рис. 3 представлена блок-схема модифицированного алгоритма с учетом перечисленных ранее принципов.

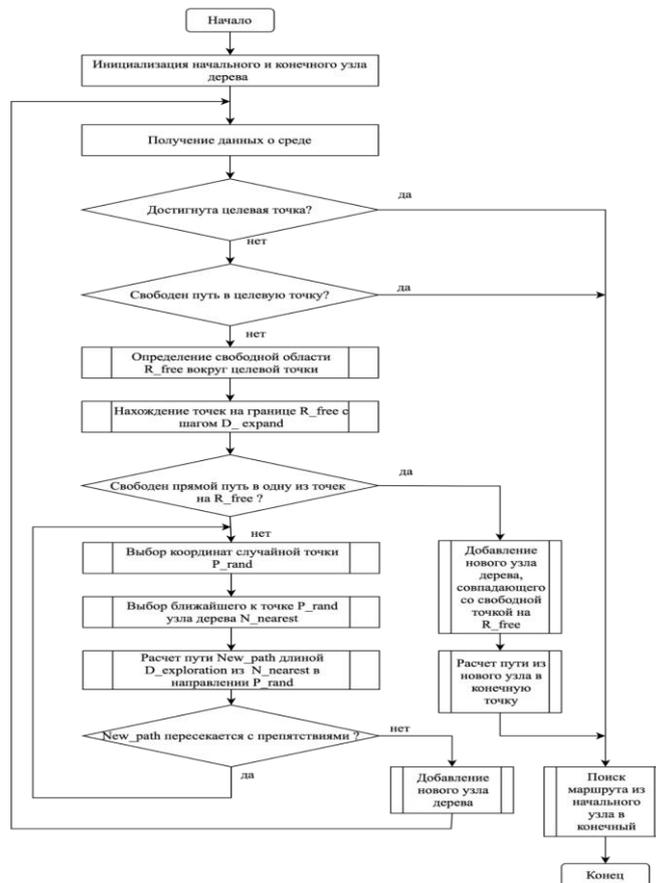


Рис. 3. Модифицированный алгоритм RRT

Основная часть алгоритма RRT осталась неизменной. Отличие предложенного алгоритма в дополнительной проверке. Перед формированием нового узла проверяется наличие свободного прямого пути из текущего узла 4 на рис. 2 в конечную точку 1. В случае, если прямой путь пересечен препятствиями, производится расчет свободной области вокруг целевой точки. Через центр свободной области строится диаметр, на него с заданным шагом  $d_{expand}$  из текущего узла строятся прямые. На пересечении прямых с границей свободной области инициализируются узлы (например, узел 3 на рис. 2). Осуществляется проверка наличия свободного прямого пути из текущего узла в узел на границе свободной области. Если в результате проверок был найден свободный прямой путь из текущего узла, то производится прокладка прямого пути.

Далее по построенному графу осуществляется нахождения маршрута из начальной точки в конечную и алгоритм завершает выполнение. В противном случае алгоритм продолжает строить граф в соответствии с классическим методом.

Аналогичным образом осуществляется проверка свободного пути из нового сгенерированного узла в родительский узел.

Таким образом применение предложенного эвристического метода расширяет список настраиваемых параметров алгоритма планирования. В реализованном классическом алгоритме настраиваются: размер области  $rand\_area$ , в рамках которой в конфигурационном пространстве осуществляется выбор случайной точки; максимальное расстояние  $expand\_dist$ , на которое от родительского узла в сторону случайной точки пространства, может вырасти ветка дерева; вероятность  $goal\_sample\_rate$ , с которой конечная точка пути выбирается в качестве точки, в направлении которой растет ветка дерева; максимальное количество  $max\_itert$  попыток построить новый узел дерева. При превышении этого числа алгоритм прекращает работу.

В силу того, что проверка пути в заданную область вносит дополнительные вычислительные затраты, она осуществляется не каждый шаг. В этой связи дополнительным настраиваемым параметром является частота проверки наличия пути на границу свободной зоны вокруг целевой точки  $straight\_path\_to\_area\_check$ .

Определение наличия прямого пути из текущего узла на границу свободной области вокруг целевой точки или сгенерированного нового узла осуществлено по следующему алгоритму:

Шаг 1. Расчет радиуса свободной зоны вокруг целевой точки или сгенерированного нового узла:  $R_{free}$  – равен расстоянию до границы ближайшего препятствия.

Шаг 2. Сегментация диаметра свободной зоны  $d_{free}$  на отрезки с заданным в параметрах алгоритма шагом  $d_{expand}$ .

Шаг 3. Нахождение точек пересечения прямых из текущего узла в точки на диаметре свободной зоны с границей свободной зоны.

Шаг 4. Определение наличия свободного от столкновений пути по одной из прямых, идущей из текущего узла в точку на границе свободной зоны вокруг целевой точки.

**Исследование модифицированного алгоритма быстрорастущих случайных деревьев по сравнению с классическим RRT.** В ходе исследования было проведено сравнительное моделирование двух методов: классического алгоритма RRT и модифицированного алгоритма, представленного на рис. 3. Алгоритмы были описаны на программном языке Python 3.9. Моделирование проведено посредством программной симуляции в интегрированной среде разработки PyCharm.

Моделируемая среда представлена в виде двумерной координатной сетки, в каждой точке которой может находиться робот, описываемый точкой позиционирования  $(x, y)$  и угол ориентации  $\varphi$  (рис. 1). Препятствия описаны набором окружностей, который включают к себя буферную область безопасности. Единичное

действие подразумевает изменение состояния агента, а именно одной или нескольких его координат  $(x, y, \varphi)$ . Моделирующий комплекс в среде PyCharm реализован в соответствии со схемой, представленной на рис. 4.

Подвижный объект представлен уравнениями (1)–(5). Моделирование блоков навигационной системы и системы технического зрения реализовано в соответствии с описанием, представленным в [63]. Расчет глобального пути реализуется в системе планирования пути, в рамках которой реализован как классический RRT, так и предложенный в данной статье модифицированный вариант. Датчики исполнительных механизмов приняты идеальными, а сами исполнительные механизмы – пропорциональными безынерционными звеньями. Регулятор синтезирован методом позиционно-траекторного управления [16, 62, 64, 65]

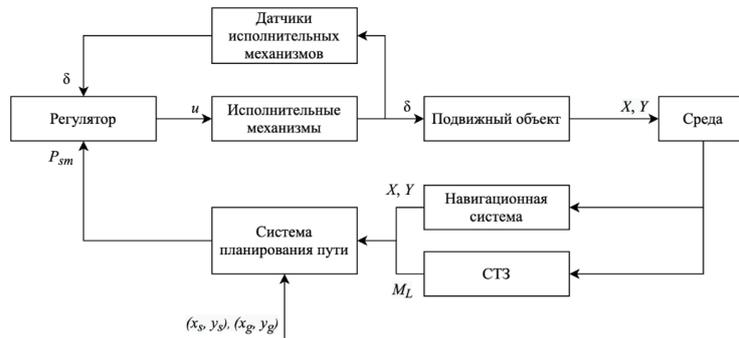


Рис. 4. Функциональная схема моделирующего комплекса

Каждое препятствие в виде круга описывается упорядоченным списком вида:

$$(x_{obst}, y_{obst}, r_{obst}), \quad (6)$$

где  $x_{obst}, y_{obst}$  – координаты центра препятствия,  $r_{obst}$  – радиус опасной зоны вокруг центра препятствия.

Прямоугольное препятствие описывается кортежем вида:

$$(x_{obst}, y_{obst}, a_{obst}, b_{obst}), \quad (7)$$

где  $x_{obst}, y_{obst}$  – координаты левого нижнего угла препятствия,  $a_{obst}, b_{obst}$  – ширина и длина препятствия.

В настоящем исследовании проведено 270 000 итераций моделирования. Каждая итерация включает в себя следующие шаги:

1. Генерация случайной ситуации на карте: координат начальной и конечной точек траектории, заданное число  $N_{obst}$  препятствий со случайными параметрами.

2. Моделирование движения робота на сгенерированной карте с использованием классического алгоритма RRT.

3. Моделирование движения робота на сгенерированной карте с использованием модифицированного алгоритма RRT.

4. Фиксация показателей работы классического и модифицированного алгоритмов RRT. К показателям относятся: время работы алгоритма ( $t_{RRT}, t_{new\_RRT}$ ); объем требуемой памяти ( $m_{RRT}, m_{new\_RRT}$ ); длина полученного пути ( $l_{RRT}, l_{new\_RRT}$ ); частота успешного достижения цели без коллизий ( $f_{RRT}, f_{new\_RRT}$ ).

Моделирование проведено в различных условиях. Серия запусков осуществлена на картах разной размерности  $map_{din} = [50:50, 500:500, 5000:5000]$ , с различным числом препятствий  $map_{obst} = [10, 30, 50]$ , с различными положениями начальной и целевой точек.

Настраиваемые параметры классического и модифицированного алгоритмов RRT на протяжении всех экспериментов были идентичны.

Результаты моделирования подставлены в табл. 1–3. Примеры работы классического и модифицированного алгоритмов RRT представлены на рис. 5 и 7. В таблицах приняты обозначения:  $t_{cp}$  – среднее время работы,  $m_{cp}$  – средний объем занимаемой памяти,  $l_{cp}$  – средняя длина пути,  $i_n$  – процент неудачных итераций моделирования,  $i_{max}$  – максимальное число итераций,  $map_{din}$  – размерность карты.

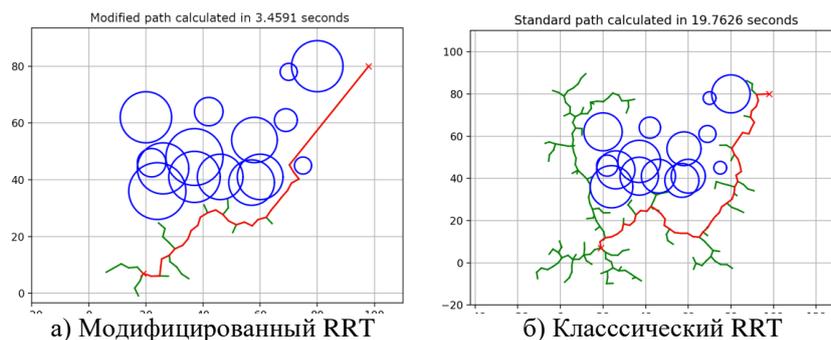


Рис. 5. Пример моделирования классического и модифицированного методов RRT для построения пути

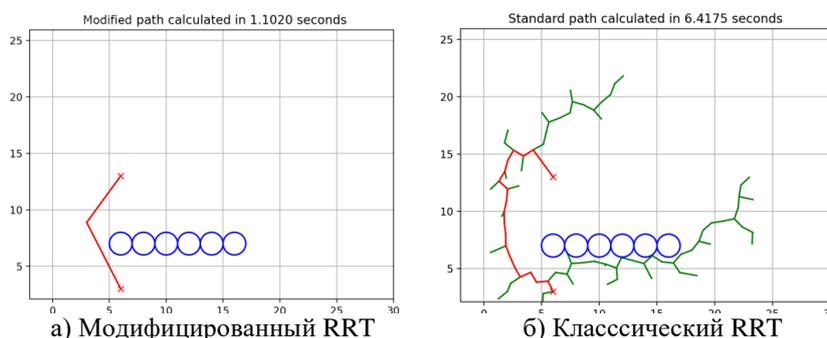


Рис. 6. Пример моделирования классического и модифицированного методов RRT для построения пути при наличии стены

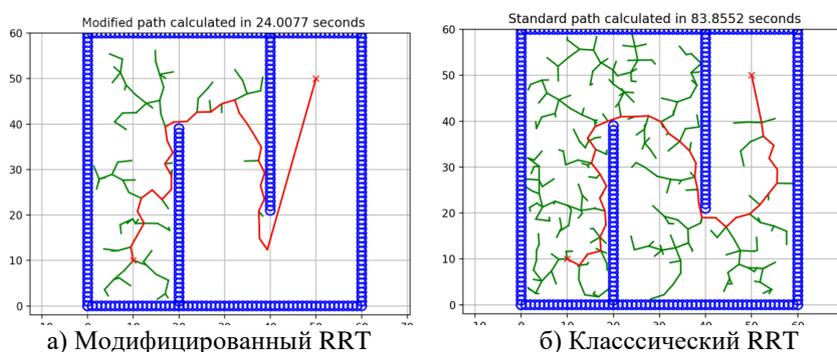


Рис. 7. Пример моделирования классического и модифицированного методов RRT для построения пути при наличии двух стен

Согласно данным, представленным в табл. 1 использование модифицированного метода в смоделированных условиях, в зависимости от размера карты и максимального числа итераций позволило сократить время расчета в пределах 64.0–90.3%, снизить объем используемой памяти на 84.0–95.0%, уменьшить длину траектории на 14.0–16.0%. Кроме того, классический метод RRT не успел найти текущий путь, в зависимости от размера карты и максимального числа итераций в 0,02–7.7% случаев. В то же время модифицированный алгоритм RRT нашел путь в 100% случаев. Всего было проведено 90000 экспериментов с описанными выше параметрами.

Таблица 1

**Результаты моделирования при  $N_{obst} = 10$** 

$i_{max}$	Метод	$t_{cp}$ , сек			$l_{cp}$ , ед.		
	$map_{din}$	50	500	5000	50	500	5000
700	RRT	0.0033	0.0202	0.0394	354629	3082035	4538360
	M-RRT	0.0003	0.0029	0.0141	298781	2608382	3806890
1500	RRT	0.0036	0.0244	0.1115	345404	3433538	21050822
	M-RRT	0.0003	0.0035	0.0365	355429	2917069	17944797
3000	RRT	0.0039	0.0260	0.1789	299590	3455666	33398372
	M-RRT	0.0003	0.0037	0.0642	300434	2937640	28738863
$i_{max}$	Метод	$m_{cp}$ , байт			$i_n$ , %		
	$map_{din}$	50	500	5000	50	500	5000
700	RRT	1053.80	3041.87	4000.77	0.36%	7.84%	76.85%
	M-RRT	169.66	219.89	216.09	0.0%	0.0%	0.0%
1500	RRT	1071.51	3320.04	7464.29	16.82%	0.82%	27.54%
	M-RRT	171.50	245.39	340.73	0.04%	0.0%	0.0%
3000	RRT	1088.93	3373.34	9584.43	0.0%	0.21%	1.9%
	M-RRT	172.61	257.74	488.63	0.0%	0.0	0.0%

Таблица 2

**Результаты моделирования при  $N_{obst} = 30$** 

$i_{max}$	Метод	$t_{cp}$ , сек			$l_{cp}$ , ед.		
	$map_{din}$	50	500	5000	50	500	5000
700	RRT	0.0097	0.0266	0.0591	338108	1709574	2760827
	M-RRT	0.0051	0.0141	0.0400	326533	1565291	2387731
1500	RRT	0.015	0.0474	0.1374	399790	2930239	11103805
	M-RRT	0.008	0.0309	0.1135	385079	2745523	10001515
3000	RRT	0.0190	0.1124	0.2827	413072	3305314	29437582
	M-RRT	0.0104	0.0619	0.2288	400049	3106780	27433970
$i_{max}$	Метод	$m_{среднее}$ , байт			$i_{неудачных}$ , %		
	$map_{din}$	50	500	5000	50	500	5000
700	RRT	1224.01	2502.32	3475.78	11.7%	43.22%	84.5 %
	M-RRT	636.28	713.05	488.05	1.69%	1.52%	0.13%
1500	RRT	1541.64	3455.61	6107.81	3.05%	18.16%	57.96%
	M-RRT	816.14	1229.59	1238.44	0.4%	1.41%	0.6%
3000	RRT	1679.25	5821.56	6905.18	0.74%	11.76%	17.52%
	M-RRT	914.10	2105.76	2254.93	0.08%	0.93%	1.25%

Согласно данным, представленным в табл. 2 использование модифицированного метода при увеличенном числе препятствий позволило сократить время расчета в пределах 17.4–46.8%, снизить объем используемой памяти на 45.6–86.0%, уменьшить длину траектории на 3.0–13.5%. Кроме того, классический метод RRT не успел найти текущий путь, в зависимости от размера карты и максимального числа итераций в 0,13–11.7% случаев. В то же время модифицированный алгоритм RRT на нашел путь в 0–1.7% случаев. Всего было проведено 90000 экспериментов с описанными выше параметрами.

Анализируя данные, приведенные в табл. 3, определяем, что использование модифицированного метода, при числе препятствий 50, позволило сократить время расчета в пределах 14.3–47.3%, снизить объем используемой памяти на 29.8–80.2%, уменьшить длину траектории на 1.9–8.9%. Кроме того, классический метод RRT не успел найти текущий путь, в зависимости от размера карты и максимального числа итераций в 6,9–88.7% случаев. В то же время модифицированный алгоритм RRT на нашел путь в 0.2–4.6% случаев. Также было проведено 90000 экспериментов с описанными выше параметрами

Отметим, что с увеличением числа препятствий и размерности карты эффект от примененной в данной статье эвристики несколько снижается. Так на рис. 8–10 данные табл. 1–3 представлены в виде графиков, из которых видно, что на карте размерностью 5000×5000 при числе препятствий, равном 50, модифицированный алгоритм RRT по времени проигрывает классическому алгоритму, если максимальное число итераций более 1000. Это связано с тем, что на большой карте и при большом числе препятствий сложность определения прямого пути из текущей точки в заданную область возрастает.

Таблица 3

Результаты моделирования при  $N_{obst} = 50$

$i_{max}$	Метод	$t_{cp}$ , сек			$l_{cp}$ , ед.		
	$map_{din}$	50	500	5000	50	500	5000
700	RRT	0.0133	0.0326	0.0772	222692	570338	1826683
	M-RRT	0.0088	0.0172	0.0652	216445	519433	1608241
1500	RRT	0.0268	0.0666	0.1600	345404	1370839	6354044
	M-RRT	0.0188	0.0486	0.1866	338594	1288650	5814408
3000	RRT	0.0463	0.1328	0.2980	427098	2317289	14420732
	M-RRT	0.0336	0.1062	0.4708	419115	2198826	13550898
$i_{max}$	Метод	$m_{cp}$ , байт			$i_n$ , %		
	$map_{din}$	50	500	5000	50	500	5000
700	RRT	1073.89	2456.67	3095.97	35.96 %	74.55 %	88.73 %
	M-RRT	679.88	691.59	612.80	4.63 %	1.14 %	0.2 %
1500	RRT	1625.69	3733.38	5158.43	16.82 %	53.0 %	72.33 %
	M-RRT	1089.77	1463.61	1390.18	3.14 %	3.0 %	1.29 %
3000	RRT	2174.86	5289.81	7529.37	6.94 %	33.07 %	51.21 %
	M-RRT	1526.72	2466.74	2863.20	1.11 %	3.53 %	2.67 %

В табл. 4 приведен процент выигрыша модифицированного алгоритма по всем критериям за все 270 000 проведенных экспериментов.

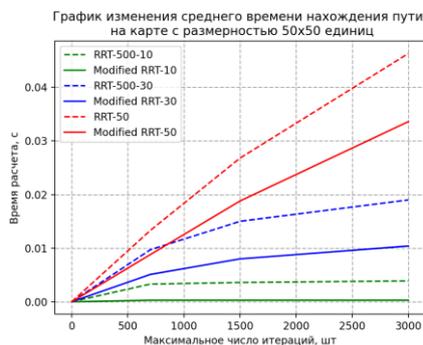


Рис. 8. График изменения времени поиска пути на карте 50x50

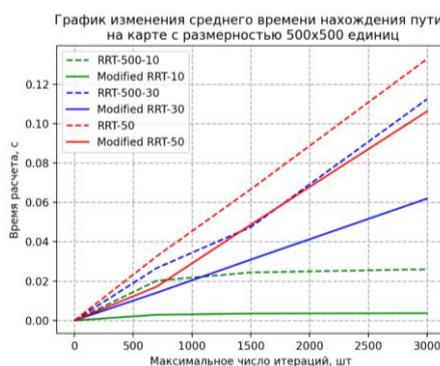


Рис. 9. График изменения времени поиска пути на карте 500x500

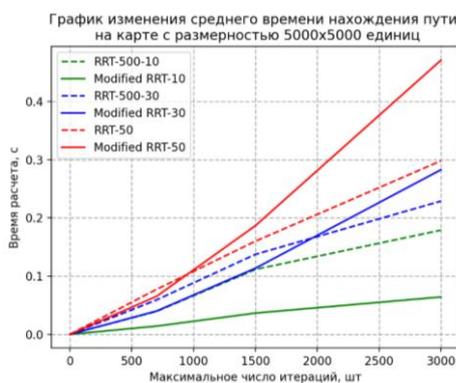


Рис. 10. График изменения времени поиска пути на карте 5000x5000

Таблица 4

**Результаты моделирования, сгруппированные по размеру среды**

Размерность карты	$t_{cp}$ , сек	$m_{cp}$ , байт	$l_{cp}$ , ед.
50x50	55,48%	54,67%	8,03%
500x500	53,06%	70,56%	9,56%
5000x5000	21,16%	81,46%	11,27%

**Заключение.** Анализ результатов серии экспериментов позволяет сделать ряд выводов, чтобы определить, в каких ситуациях эффективно применение модифицированного алгоритма RRT, а в каких классического.

С ростом сложности среды – увеличением числа препятствий, преимущество модифицированного алгоритма по  $t_{\text{среднее}}$ ,  $m_{\text{среднее}}$  и  $l_{\text{среднее}}$  падает, но сохраняется. Это обусловлено увеличением вычислительных затрат на определение наличия прямого пути – продолжительность этой процедуры прямо пропорционально связана с числом препятствий на карте. Этот недостаток возможно частично компенсировать оптимизацией расчетов вплоть до применения искусственных нейронных сетей. Снижение величины преимущества по  $m_{\text{ср}}$  и  $l_{\text{ср}}$  связано с увеличением количества маневров, которые агент должен совершать, огибая большое число препятствий. Большое число препятствий на карте снижает число участков, в прохождении которых модифицированный алгоритм дает выигрыш.

С ростом размерности среды преимущество модифицированного алгоритма по  $t_{\text{ср}}$  снижается, но сохраняется. А преимущество по  $m_{\text{ср}}$  и  $l_{\text{ср}}$  увеличивается. Изменение преимущества по  $t_{\text{ср}}$  связано с увеличением числа итераций роста дерева, которые выполняет алгоритм, соответственно модифицированный алгоритм выполняет большее число расчетов эвристики и затрачивает на них больше времени. Изменение по  $m_{\text{ср}}$  и  $l_{\text{ср}}$  связано с тем, что модифицированное RRT стремится расти в направлении целевой точки – при любой возможности путь будет проложен либо сразу в целевую точку, либо на границу свободной области вокруг нее, классическое дерево, напротив, растет хаотично – итоговый граф получается большим и содержит много изгибов, соответственно, на хранение его требуется больше памяти, а путь получается более ломаным и за счет этого немного длиннее.

Использование модифицированного алгоритма увеличивает процент вероятности нахождения решения в рамках заданных требований к допустимому числу итераций роста дерева  $i_{\text{max}}$ . Из всех экспериментов модифицированный алгоритм не нашел решение в 1,07% случаев, классический в 30,13%.

Таким образом, модифицированный алгоритм продемонстрировал преимущество:

- ◆ в случаях незагруженной среды;
- ◆ в случаях, когда есть ограничение по максимальному числу итераций;
- ◆ в случаях, когда ограничен объем памяти, допустимый для хранения рассчитанного дерева.

К недостаткам алгоритма можно отнести зависимость временной вычислительной эффективности от размерности среды при средней и высокой загрузке карты.

Работа выполнена при поддержке гранта Российского научного фонда № 22-29-00533, выполняемого в АО "Научно-конструкторское бюро робототехники и систем управления".

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Гайдук А.П.* Применение пространства состояния к исследованию систем автоматического управления. – Таганрог: ТРТИ, 1979. – 100 с.
2. *LaValle S.* Planning Algorithms. – Cambridge University Press, 2006. – 842 p.
3. *Lozano-Perez T.* Spatial planning: A configuration space approach // IEEE Transactions on Computers. – 1983. – Vol. 32 (2). – P. 108-120.
4. *Казakov К.А., Семенов В.А.* Обзор современных методов планирования пути // Тр. ИСП РАН. – 2016. – Т. 28 (4). – С. 241-294.
5. *Hart P.E., Nilsson N.J., Raphael B.A.* Formal Basis for the Heuristic Determination of Minimum Cost Paths // IEEE Transactions on Systems Science and Cybernetics. – 1968. – Vol. 2. – P. 100-107.

6. Stentz A. Optimal and efficient path planning for partially known environments // In Intelligent Unmanned Ground Vehicles. – Springer, Boston, MA, USA, 1997. – P. 203-220.
7. Wang Q., Hao Y., Chen F. Deepening the IDA\* algorithm for knowledge graph reasoning through neural network architecture // Neurocomputing. – 2021. – Vol. 429. – P. 101-109.
8. Zhou R., Hansen E.A. Memory-Bounded {A\*} Graph Search // The Florida AI Research Society Conference – FLAIRS. – 2002. – P. 203-209.
9. Holte R., Perez M., Zimmer R., MacDonald A. Hierarchical A\*: Searching abstraction hierarchies efficiently // Proceedings of the thirteenth national conference on Artificial intelligence. – 1996. – Vol. 1. – P. 530-535.
10. Liu B., Xiao X., Stone P. A Lifelong Learning Approach to Mobile Robot Navigation // In IEEE Robotics and Automation Letters. – 2021. – Vol. 6 (2). – P. 1090-1096.
11. Chen B.Y., Chen X.-W., Chen H.-P., Lam W.H.K. Efficient algorithm for finding k shortest paths based on re-optimization technique // Transportation Research Part E: Logistics and Transportation Review. – 2020. – Vol. 133. Article number 101819.
12. Zhang X., Wylie B., Oscar C., Moore C.A. Time-Optimal and Collision-Free Path Planning for Dual-Manipulator 3D Printer // IEEE Transactions on Systems, Man, and Cybernetics: Systems. – 2020. – P. 2389-2396.
13. Пилюхов В.Х., Медведев М.Ю., Костоков В.А., Хуссейн Ф., Кадим А. Алгоритмы планирования траекторий в двумерной среде с препятствиями // Информатика и автоматизация. – 2022. – Vol. 21 (3). – P. 459-492. – <https://doi.org/10.15622/ia.21.3.1>.
14. Khaitib O. Real-Time Obstacles Avoidance for Manipulators and Mobile Robots // International Journal of Robotics Research. – 1986. – Vol. 5 (2). – P. 90-98.
15. Платонов А.К., Карнов И.И., Кирильченко А.А. Метод потенциалов в задаче прокладки трассы // Препринт Института прикладной математики АН СССР. – М.: 1974. – 27 с.
16. Pshikhopov V.Kh. (Ed.), Beloglazov D., Finaev V., Guzik V., Kosenko E., Krukhmalev V., Medvedev M., Pereverzev A., Pyavchenko A., Saprykin R., Shapovalov I., Soloviev V. Path Planning for Vehicles Operating in Uncertain 2D Environments. – Elsevier, Butterworth-Heinemann, 2017. – 312 p. – ISBN: 9780128123058.
17. Филимонов А.Б., Филимонов Н.Б. Вопросы управления движением мобильных роботов методом потенциального наведения // Мехатроника, автоматизация, управление. – 2019. – Т. 20 (11). – С. 677-685.
18. Woods A.C., La H.M. A Novel Potential Field Controller for Use on Aerial Robots // IEEE Transactions on Systems, Man, and Cybernetics: Systems. – 2019. – Vol. 49 (4). – P. 665-676.
19. Koren Y., Borenstein J. Potential field methods and their inherent limitations for mobile robot navigation // International Conference on Robotics and Automation. – 1991. – Vol. 2. – P. 1398-1404.
20. Пилюхов В.Х., Медведев М.Ю. Групповое управление движением мобильных роботов в неопределенной среде с использованием неустойчивых режимов // Тр. СПИИРАН. – 2018. – Вып. 60. – С. 39-63.
21. Malone N., Chiang H.-T., Lesser K., Oishi M., Tapia L. Hybrid Dynamic Moving Obstacle Avoidance Using a Stochastic Reachable Set-Based Potential Field // IEEE Transactions on Robotics. – 2017. – Vol. 33 (5). – P. 1124-1138.
22. Friudenberg P., Koziol S. Mobile Robot Rendezvous Using Potential Fields combined With Parallel Navigation // IEEE Access. – 2018. – Vol. 6. – P. 16948-16957.
23. Fedele G., D'Alfonso L., Chiaravalloti F., D'Aquila G. Obstacles Avoidance Based on Switching Potential Functions // Journal of Intelligent and Robotic Systems: Theory and Applications. – 2018. – Vol. 90 (3-4). – P. 387-405.
24. Гайдук А.Р., Мартьянов О.В., Медведев М.Ю., Пилюхов В.Х., Хамдан Н., Фархунд А. Нейросетевая система управления группой роботов в неопределенной двумерной среде // Мехатроника, автоматизация, управление. – 2020. – Т. 21 (8). – С. 470-479.
25. Medvedev M., Pshikhopov V. Path Planning of Mobile Robot Group Based on Neural Networks // Lecture Notes in Artificial Intelligence. – 2020. – P. 51-62.
26. Wang Y., Cheng L., Hou Z.-G., Yu J., Tan M. Optimal Formation of Multirobot Systems Based on a Recurrent Neural Network // IEEE Transactions on Neural Networks and Learning Systems. – 2016. – Vol. 27 (2). – P. 322-333.

27. Güzel M., Ajabshir V., Nattharith P., Gezer E., Can S. A Novel Framework for Multi-Agent Systems Using a Decentralized Strategy // *Robotica*. – 2019. – Vol. 37 (4). – P. 691-707. – DOI: 10.1017/S0263574718001261.
28. De Berg M., Cheong O., Van Kreveld M., Overmars M. *Computational Geometry: Algorithms and Applications*. – 3rd ed. – Springer-Verlag, 2008. – 386 p.
29. Guibas L.J., Knuth D.E., Sharir M. Randomized incremental construction of Delaunay and Voronoi diagrams // *Algorithmica*. – 1992. – Vol. 7 (1). – P. 381-413.
30. LaValle S.M., Kuffner J.J. Rapidly-exploring random trees: Progress and prospects // *Workshop on the Algorithmic Foundations of Robotics*. – 2000. – P. 293-308.
31. Kedem K., Sharir M. An efficient motion planning algorithm for a convex rigid polygonal object in 2-dimensional polygonal space // *Discrete Computational Geometry*. – 1990. – Vol. 5 (1). – P. 43-75.
32. Lee W., Choi G.-H., Kim T.-W. Visibility graph-based path-planning algorithm with quadtree representation // *Applied Ocean Research*. – 2021. – Vol. 117.
33. Finkel R., Bentley J.L. Quad Trees: A Data Structure for Retrieval on Composite Keys // *Acta Informatica*. – 1974. – Vol. 4 (1). – P. 1-9. – DOI: 10.1007/BF00288933.
34. Pradhan S., Mandava R.K., Vundavilli P.R. Development of path planning algorithm for biped robot using combined multi-point RRT and visibility graph // *International Journal of Information Technology*. – 2021. – Vol. 13. – P. 1513-1519. – <https://doi.org/10.1007/s41870-021-00696-w>.
35. Белоглазов Д.А., Гузик В.Ф., Косенко Е.Ю., Крухмалев В.А., Медведев М.Ю., Переверзев В.А., Пшихопов В.Х., Пьявченко А.О., Сапрыкин П.В., Соловьев В.В., Финаев В.И., Чернухин Ю.В., Шаповалов И.О. Интеллектуальное планирование траекторий подвижных объектов в средах с препятствиями / под ред. В.Х. Пшихопова. – М.: Физматлит, 2014. – 300 с. – ISBN 978-5-9221-1595-7.
36. Bhattacharya P., Gavrilova M.L. Roadmap-Based Path Planning - Using the Voronoi Diagram for a Clearance-Based Shortest Path // *IEEE Robotics & Automation Magazine*. – 2008. – Vol. 15 (2). – P. 58-66. – DOI: 10.1109/MRA.2008.921540.
37. Al-Dahhan M.R.H., Schmidt K.W. Voronoi Boundary Visibility for Efficient Path Planning // *IEEE Access*. – 2020. – Vol. 8. – P. 134764-134781. – DOI: 10.1109/ACCESS.2020.3010819.
38. Magid E., Lavrenov R., Svinin M., Khasianov, A. Combining Voronoi Graph and Spline-Based Approaches for a Mobile Robot Path Planning. In: Gusikhin O., Madani K. (eds) // *Informatics in Control, Automation and Robotics. ICINCO 2017. Lecture Notes in Electrical Engineering*. – Vol 495. – Springer, Cham, 2020. – [https://doi.org/10.1007/978-3-030-11292-9\\_24](https://doi.org/10.1007/978-3-030-11292-9_24).
39. Kavradi L.E., Svestka P., Latombe J.C., Overmars M.H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces // *IEEE transactions on Robotics and Automation*. – 1996. – Vol. 12 (4). – P. 566-580. – DOI: 10.1109/70.508439.
40. Kamil A.R.M., Shithil S.M., Ismail Z.H., Mahmud M.S.A., Faudzi A.A.M. Path Planning Based on Inflated Medial Axis and Probabilistic Roadmap for Duct Environment // *Lecture Notes in Electrical Engineering*. – Vol. 834. – Springer, Singapore, 2022. – [https://doi.org/10.1007/978-981-16-8484-5\\_42](https://doi.org/10.1007/978-981-16-8484-5_42).
41. Chen G., Luo N., Liu D., Zhao Z., Liang Ch. Path planning for manipulators based on an improved probabilistic roadmap method // *Robotics and Computer-Integrated Manufacturing*. – 2021. – Vol. 72.
42. Rantanen M.T., Juhola M. Speeding up probabilistic roadmap planners with locality-sensitive hashing // *Robotica*. – 2015. – Vol. 33 (7). – P. 1491-1506.
43. Buaba R., Homaiyar A., Gebril M., Kihn E. Satellite Image Retrieval Application Using Locality Sensitive Hashing in L2-Space // *Proceedings of IEEE Aerospace Conference, Big Sky, MT, USA, 2011*. – P. 1-7.
44. Kala R. Increased Visibility Sampling for Probabilistic Roadmaps // *IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots, Brisbane, AUSTRALIA, 2018*. – P. 87-92.
45. Chakravorty S., Kumar S. Generalized Sampling-Based Motion Planners // *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics*. – 2011. – Vol. 41 (3).
46. Qureshi A., Ayaz Y. Potential functions based sampling heuristic for optimal path planning // *Autonomous Robot*. – 2016. – Vol. 40. – P. 1079-1093.
47. Cheng G.-R., Ma M.-C., Tan L.-G., Song S.-M. Gaussian estimation for non-linear stochastic uncertain systems with time-correlated additive noises and packet dropout compensations // *IET Control Theory Appl.* – 2022. – Vol. 16. – P. 600-614. – <https://doi.org/10.1049/cth2.12252>.

48. Wang X., Li X., Guan Y., Song J., Wang R. Bidirectional Potential Guided RRT\* for Motion Planning // *IEEE Access*. – 2019. – Vol. 7. – P. 95046-95057.
49. Karaman S., Frazzoli E. Sampling-based algorithms for optimal motion planning // *The International Journal of Robotics Research*. – 2011. – Vol. 30 (7). – P. 846-894.
50. Chen L., Shan Y., Tian W., Li B., Cao D. A Fast and Efficient Double-Tree RRT\*-Like Sampling-Based Planner Applying on Mobile Robotic Systems // *IEEE/ASME Transactions on Mechatronics*. – 2018. – Vol. 23 (6). – P. 2568-2578.
51. Wang J., Meng M.Q.-H., Khatib O. EB-RRT: Optimal Motion Planning for Mobile Robots // *IEEE Transactions on Automation Science and Engineering*. – 2020. – Vol. 17 (4). – P. 2063-2073.
52. Janos J., Vonasek V., Penicka R. Multi-Goal Path Planning Using Multiple Random Trees // *IEEE Robotics and Automation Letter*. – 2021. – Vol. 6 (2). – P. 4201-4208.
53. Devaurs D., Siméon T., Cortés J. A multi-tree extension of the transition-based RRT: Application to ordering-and-pathfinding problems in continuous cost spaces // *In Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* – 2014. – P. 2991-2996.
54. Vonásek V., Penička R. Space-filling forest for multi-goal path planning // *In Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Automat.* – 2019. – P. 1587-1590.
55. Jeong I.-B., Lee S.-J., Kim J.-H. Quick-RRT\*: Triangular inequality-based implementation of RRT\* with improved initial solution and convergence rate // *Expert Systems with Applications*. – 2019. – Vol. 123. – P. 82-90.
56. Yang K. Anytime synchronized-biased-greedy rapidly-exploring random tree path planning in two dimensional complex environments // *International Journal of Control, Autom. Syst.* – 2011. – Vol. 9, No. 4. – P. 750.
57. Wang J., Chi W., Shao M., Meng M. Q.-H. Finding a high-quality initial solution for the RRTs algorithms in 2D environments // *Robotica*. – 2019. – Vol. 37, No. 10. – P. 1677-1694.
58. Medvedev M., Pshikhopov V., Gurenko B., Hamdan N. Path planning method for mobile robot with maneuver restrictions // *Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME) 7-8 October 2021, Mauritius*. – 10.1109/ICECCME52200.2021.9591090.
59. Медведев М.Ю., Костюков В.А., Пушхопов В.Х. Оптимизация движения мобильного робота на плоскости в поле конечного числа источников-репеллеров // *Тр. СПИИРАН*. – 2020. – Т. 19 (1). – С. 43-78.
60. Kostjukov V., Medvedev M., Pshikhopov V. Method for Optimizing of Mobile Robot Trajectory in Repeller Sources Field // *Informatics and Automation*. – 2021. – Vol. 20 (3). – P. 690-726.
61. Пушхопов В.Х., Медведев М.Ю., Гайдук А.Р., Нейдорф Р.А., Беляев В.Е., Федоренко Р.В., Костюков В.А., Крухмалев В.А. Система позиционно-траекторного управления роботизированной воздухоплавательной платформой: математическая модель // *Мехатроника, автоматизация и управление*. – 2013. – № 6. – С. 14-21.
62. Pshikhopov, V., Medvedev, M. Multi-Loop Adaptive Control of Mobile Objects in Solving Trajectory Tracking Tasks // *Automation and Remote Control*. – 2020. – Vol. 81 (11). – P. 2078-2093. – <https://doi.org/10.1134/S0005117920110090>.
63. Пушхопов В.Х., Медведев М.Ю. Сравнительный анализ централизованного и децентрализованного алгоритмов движения строим БЛА мультикоптерного типа // *Известия ЮФУ. Технические науки*. – 2022. – № 1 (225). – С. 121-139.
64. Pshikhopov V.Kh., Medvedev M.Yu., Gaiduk A.R., Fedorenko R.V., Krukhmalev V.A., Gurenko B.V. Position-Trajectory Control System for Unmanned Robotic Airship // *IFAC Proceedings Volumes (IFAC-PapersOnline)*. The 19th World Congress the International Federation of Automatic Control. Cape Town, South Africa. August 24-29, 2014. – P. 8953-8958.
65. Пушхопов В.Х., Медведев М.Ю. Синтез систем управления подводными аппаратами с нелинейными характеристиками исполнительных органов // *Известия ЮФУ. Технические науки*. – 2011. – № 3 (116). – С. 147-156.

## REFERENCES

1. Gayduk A.R. *Primenenie prostranstva sostoyaniya k issledovaniyu sistem avtomaticheskogo upravleniya* [Application of the state space to the study of automatic control systems]. Taganrog: TRTI, 1979, 100 p.
2. LaValle S. *Planning Algorithms*. Cambridge University Press, 2006, 842 p.
3. Lozano-Perez T. Spatial planning: A configuration space approach, *IEEE Transactions on Computers*, 1983, Vol. 32 (2), pp. 108-120.

4. Kazakov K.A., Semenov V.A. Obzor sovremennykh metodov planirovaniya puti [Review of modern methods of path planning], *Tr. ISP RAN* [Proceedings of ISP RAS], 2016, Vol. 28 (4), pp. 241-294.
5. Hart P.E., Nilsson N.J., Raphael B.A. Formal Basis for the Heuristic Determination of Minimum Cost Paths, *IEEE Transactions on Systems Science and Cybernetics*, 1968, Vol. 2, pp. 100-107.
6. Stentz A. Optimal and efficient path planning for partially known environments, *In Intelligent Unmanned Ground Vehicles*. Springer, Boston, MA, USA, 1997, pp. 203-220.
7. Wang Q., Hao Y., Chen F. Deepening the IDA\* algorithm for knowledge graph reasoning through neural network architecture, *Neurocomputing*, 2021, Vol. 429, pp. 101-109.
8. Zhou R., Hansen E.A. Memory-Bounded {A\*} Graph Search, *The Florida AI Research Society Conference – FLAIRS*, 2002, pp. 203-209.
9. Holte R., Perez M., Zimmer R., MacDonald A. Hierarchical A\*: Searching abstraction hierarchies efficiently, *Proceedings of the thirteenth national conference on Artificial intelligence*, 1996, Vol. 1, pp. 530-535.
10. Liu B., Xiao X., Stone P. A Lifelong Learning Approach to Mobile Robot Navigation, *In IEEE Robotics and Automation Letters*, 2021, Vol. 6 (2), pp. 1090-1096.
11. Chen B.Y., Chen X.-W., Chen H.-P., Lam W.H.K. Efficient algorithm for finding k shortest paths based on re-optimization technique, *Transportation Research Part E: Logistics and Transportation Review*, 2020, Vol. 133. Article number 101819.
12. Zhang X., Wylie B., Oscar C., Moore C.A. Time-Optimal and Collision-Free Path Planning for Dual-Manipulator 3D Printer, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020, pp. 2389-2396.
13. Pshikhopov V.Kh., Medvedev M.Yu., Kostyukov V.A., Khusseyn F., Kadim A. Algoritmy planirovaniya traektoriy v dvumernoy srede s prepyatstviyami [Algorithms for trajectory planning in a two-dimensional environment with obstacles], *Informatika i avtomatizatsiya* [Informatics and automation], 2022, Vol. 21 (3), pp. 459-492. Available at: <https://doi.org/10.15622/ia.21.3.1>.
14. Khatib O. Real-Time Obstacles Avoidance for Manipulators and Mobile Robots, *International Journal of Robotics Research*, 1986, Vol. 5 (2), pp. 90-98.
15. Platonov A.K., Karpov I.I., Kiril'chenko A.A. Metod potentsialov v zadache prokladki trassy [The method of potentials in the task of laying a route], Preprint Instituta prikladnoy matematiki AN SSSR [Preprint of the Institute of Applied Mathematics of the USSR Academy of Sciences]. Moscow: 1974, 27 p.
16. Pshikhopov V.Kh. (Ed.), Beloglazov D., Finaev V., Guzik V., Kosenko E., Krukhmalev V., Medvedev M., Pereverzev A., Pyavchenko A., Saprykin R., Shapovalov I., Soloviev V. Path Planning for Vehicles Operating in Uncertain 2D Environments. Elsevier, Butterworth-Heinemann, 2017, 312 p. ISBN: 9780128123058.
17. Filimonov A.B., Filimonov N.B. Voprosy upravleniya dvizheniem mobil'nykh robotov metodom potentsial'nogo navedeniya [Questions of controlling the movement of mobile robots by the method of potential guidance], *Mekhatronika, avtomatizatsiya, upravlenie* [Mechatronics, automation, control], 2019, Vol. 20 (11), pp. 677-685.
18. Woods A.C., La H.M. A Novel Potential Field Controller for Use on Aerial Robots, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019, Vol. 49 (4), pp. 665-676.
19. Koren Y., Borenstein J. Potential field methods and their inherent limitations for mobile robot navigation, *International Conference on Robotics and Automation*, 1991, Vol. 2, pp. 1398-1404.
20. Pshikhopov V.Kh., Medvedev M.Yu. Gruppovoe upravlenie dvizheniem mobil'nykh robotov v neopredelennoy srede s ispol'zovaniem neustoychivyykh rezhimov [Group control of the movement of mobile robots in an uncertain environment using unstable modes], *Tr. ISP RAN* [Proceedings of ISP RAS], 2018, Issue 60, pp. 39-63.
21. Malone N., Chiang H.-T., Lesser K., Oishi M., Tapia L. Hybrid Dynamic Moving Obstacle Avoidance Using a Stochastic Reachable Set-Based Potential Field, *IEEE Transactions on Robotics*, 2017, Vol. 33 (5), pp. 1124-1138.
22. Friudenberg P., Koziol S. Mobile Robot Rendezvous Using Potential Fields combined With Parallel Navigation, *IEEE Access*, 2018, Vol. 6, pp. 16948-16957.
23. Fedele G., D'Alfonso L., Chiaravalloti F., D'Aquila G. Obstacles Avoidance Based on Switching Potential Functions, *Journal of Intelligent and Robotic Systems: Theory and Applications*, 2018, Vol. 90 (3-4), pp. 387-405.

24. Gayduk A.R., Mart'yanov O.V., Medvedev M.Yu., Pshikhopov V.Kh., khamdan N., Farkhud A. Neyrosetevaya sistema upravleniya gruppy robotov v neopredelennoy dvumernoy srede [Neural network control system for a group of robots in an indefinite two-dimensional environment] *Mekhatronika, avtomatizatsiya, upravlenie* [Mechatronics, automation, control], 2020, Vol. 21 (8), pp. 470-479.
25. Medvedev M., Pshikhopov V. Path Planning of Mobile Robot Group Based on Neural Networks, *Lecture Notes in Artificial Intelligence*, 2020, pp. 51-62.
26. Wang Y., Cheng L., Hou Z.-G., Yu J., Tan M. Optimal Formation of Multirobot Systems Based on a Recurrent Neural Network, *IEEE Transactions on Neural Networks and Learning Systems*, 2016, Vol. 27 (2), pp. 322-333.
27. Güzel M., Ajabshir V., Nattharith P., Gezer E., Can S. A Novel Framework for Multi-Agent Systems Using a Decentralized Strategy, *Robotica*, 2019, Vol. 37 (4), pp. 691-707. DOI: 10.1017/S0263574718001261.
28. De Berg M., Cheong O., Van Kreveld M., Overmars M. Computational Geometry: Algorithms and Applications. 3rd ed. Springer-Verlag, 2008, 386 p.
29. Guibas L.J., Knuth D.E., Sharir M. Randomized incremental construction of Delaunay and Voronoi diagrams, *Algorithmica*, 1992, Vol. 7 (1), pp. 381-413.
30. LaValle S.M., Kuffner J.J. Rapidly-exploring random trees: Progress and prospects, *Workshop on the Algorithmic Foundations of Robotics*, 2000, pp. 293-308.
31. Kedem K., Sharir M. An efficient motion planning algorithm for a convex rigid polygonal object in 2-dimensional polygonal space, *Discrete Computational Geometry*, 1990, Vol. 5 (1), pp. 43-75.
32. Lee W., Choi G.-H., Kim T.-W. Visibility graph-based path-planning algorithm with quadtree representation, *Applied Ocean Research*, 2021, Vol. 117.
33. Finkel R., Bentley J.L. Quad Trees: A Data Structure for Retrieval on Composite Keys, *Acta Informatica*, 1974, Vol. 4 (1), pp. 1-9. DOI: 10.1007/BF00288933.
34. Pradhan S., Mandava R.K., Vundavilli P.R. Development of path planning algorithm for biped robot using combined multi-point RRT and visibility graph, *International Journal of Information Technology*, 2021, Vol. 13, pp. 1513-1519. Available at: <https://doi.org/10.1007/s41870-021-00696-w>.
35. Beloglazov D.A., Guzik V.F., Kosenko E.Yu., Krukhmalev V.A., Medvedev M.Yu., Pereverzev V.A., Pshikhopov V.Kh., P'yavchenko A.O., Saprykin R.V., Solov'ev V.V., Finaev V.I., Chernukhin Yu.V., Shapovalov I.O. Intellektual'noe planirovanie traektoriy podvizhnykh ob"ektov v sredakh s prepyatstviyami [Intelligent planning of trajectories of moving objects in environments with obstacles], ed. by V.Kh. Pshikhopova. Moscow: Fizmatlit, 2014, 300 p. ISBN 978-5-9221-1595-7.
36. Bhattacharya P., Gavrilova M.L. Roadmap-Based Path Planning - Using the Voronoi Diagram for a Clearance-Based Shortest Path, *IEEE Robotics & Automation Magazine*, 2008, Vol. 15 (2), pp. 58-66. DOI: 10.1109/MRA.2008.921540.
37. Al-Dahhan M.R.H., Schmidt K.W. Voronoi Boundary Visibility for Efficient Path Planning, *IEEE Access*, 2020, Vol. 8, pp. 134764-134781. DOI: 10.1109/ACCESS.2020.3010819.
38. Magid E., Lavrenov R., Svinin M., Khasianov, A. Combining Voronoi Graph and Spline-Based Approaches for a Mobile Robot Path Planning. In: Gusikhin O., Madani K. (eds.), *Informatics in Control, Automation and Robotics. ICINCO 2017. Lecture Notes in Electrical Engineering*, Vol 495. Springer, Cham, 2020. Available at: [https://doi.org/10.1007/978-3-030-11292-9\\_24](https://doi.org/10.1007/978-3-030-11292-9_24).
39. Kavragi L.E., Svestka P., Latombe J.C., Overmars M.H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE transactions on Robotics and Automation*, 1996, Vol. 12 (4), pp. 566-580. DOI: 10.1109/70.508439.
40. Kamil A.R.M., Shithil S.M., Ismail Z.H., Mahmud M.S.A., Faudzi A.A.M. Path Planning Based on Inflated Medial Axis and Probabilistic Roadmap for Duct Environment, *Lecture Notes in Electrical Engineering*, Vol. 834. Springer, Singapore, 2022. Available at: [https://doi.org/10.1007/978-981-16-8484-5\\_42](https://doi.org/10.1007/978-981-16-8484-5_42).
41. Chen G., Luo N., Liu D., Zhao Z., Liang Ch. Path planning for manipulators based on an improved probabilistic roadmap method, *Robotics and Computer-Integrated Manufacturing*, 2021, Vol. 72.
42. Rantanen M.T., Juhola M. Speeding up probabilistic roadmap planners with locality-sensitive hashing, *Robotica*, 2015, Vol. 33 (7), pp. 1491-1506.
43. Buaba R., Homatfar A., Gebril M., Kihn E. Satellite Image Retrieval Application Using Locality Sensitive Hashing in L2-Space, *Proceedings of IEEE Aerospace Conference, Big Sky, MT, USA, 2011*, pp. 1-7.

44. Kala R. Increased Visibility Sampling for Probabilistic Roadmaps. *IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, Brisbane, AUSTRALIA. 2018, pp. 87-92.
45. Chakravorty S., Kumar S. Generalized Sampling-Based Motion Planners, *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics*, 2011, Vol. 41 (3).
46. Qureshi A., Ayaz Y. Potential functions based sampling heuristic for optimal path planning, *Autonomous Robot*, 2016, Vol. 40, pp. 1079-1093.
47. Cheng G.-R., Ma M.-C., Tan L.-G., Song S.-M. Gaussian estimation for non-linear stochastic uncertain systems with time-correlated additive noises and packet dropout compensations, *IET Control Theory Appl.*, 2022, Vol. 16, pp. 600-614. Available at: <https://doi.org/10.1049/cth2.12252>.
48. Wang X., Li X., Guan Y., Song J., Wang R. Bidirectional Potential Guided RRT\* for Motion Planning, *IEEE Access*, 2019, Vol. 7, pp. 95046-95057.
49. Karaman S., Frazzoli E. Sampling-based algorithms for optimal motion planning, *The International Journal of Robotics Research*, 2011, Vol. 30 (7), pp. 846-894.
50. Chen L., Shan Y., Tian W., Li B., Cao D. A Fast and Efficient Double-Tree RRT\*-Like Sampling-Based Planner Applying on Mobile Robotic Systems, *IEEE/ASME Transactions on Mechatronics*, 2018, Vol. 23 (6), pp. 2568-2578.
51. Wang J., Meng M.Q.-H., Khatib O. EB-RRT: Optimal Motion Planning for Mobile Robots, *IEEE Transactions on Automation Science and Engineering*, 2020, Vol. 17 (4), pp. 2063-2073.
52. Janos J., Vonasek V., Penicka R. Multi-Goal Path Planning Using Multiple Random Trees, *IEEE Robotics and Automation Letter*, 2021, Vol. 6 (2), pp. 4201-4208.
53. Devaurs D., Siméon T., Cortés J. A multi-tree extension of the transition-based RRT: Application to ordering-and-pathfinding problems in continuous cost spaces, *In Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 2991-2996.
54. Vonásek V., Pěnička R. Space-filling forest for multi-goal path planning, *In Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Automat.*, 2019, pp. 1587-1590.
55. Jeong I.-B., Lee S.-J., Kim J.-H. Quick-RRT\*: Triangular inequality-based implementation of RRT\* with improved initial solution and convergence rate, *Expert Systems with Applications*, 2019, Vol. 123, pp. 82-90.
56. Yang K. Anytime synchronized-biased-greedy rapidly-exploring random tree path planning in two dimensional complex environments, *International Journal of Control, Autom. Syst.*, 2011, Vol. 9, No. 4, pp. 750.
57. Wang J., Chi W., Shao M., Meng M. Q.-H. Finding a high-quality initial solution for the RRTs algorithms in 2D environments, *Robotica*, 2019, Vol. 37, No. 10, pp. 1677-1694.
58. Medvedev M., Pshikhopov V., Gurenko B., Hamdan N. Path planning method for mobile robot with maneuver restrictions, *Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME) 7-8 October 2021, Mauritius*. 10.1109/ICECCME52200.2021.9591090.
59. Medvedev M.Yu., Kostyukov V.A., Pshikhopov V.Kh. Optimizatsiya dvizheniya mobil'nogo robota na ploskosti v pole konechnogo chisla istochnikov-repellerov [Optimization of the movement of a mobile robot on a plane in the field of a finite number of repeller sources], *Tr. ISP RAN [Proceedings of ISP RAS]*, 2020, Vol. 19 (1), pp. 43-78.
60. Kostyukov V., Medvedev M., Pshikhopov V. Method for Optimizing of Mobile Robot Trajectory in Repeller Sources Field, *Informatics and Automation*, 2021, Vol. 20 (3), pp. 690-726.
61. Pshikhopov V.Kh., Medvedev M.Yu., Gayduk A.R., Neydorf R.A., Belyaev V.E., Fedorenko R.V., Kostyukov V.A., Krukhmalev V.A. Sistema pozitsionno-traektorного upravleniya robotizirovannoy vozdukhoplavatel'noy platformoy: matematicheskaya model' [Positional trajectory control system of a robotic aeronautical platform: mathematical model], *Mekhatronika, avtomatizatsiya, upravlenie [Mechatronics, automation, control]*, 2013, No. 6, pp. 14-21.
62. Pshikhopov V., Medvedev M. Multi-Loop Adaptive Control of Mobile Objects in Solving Trajectory Tracking Tasks, *Automation and Remote Control*, 2020, Vol. 81 (11), pp. 2078-2093. Available at: <https://doi.org/10.1134/S0005117920110090>.
63. Pshikhopov V.Kh., Medvedev M.Yu. Sravnitel'nyy analiz tsentralizovannogo i detsentralizovannogo algoritmov dvizheniya stroem BLA mul'tikopternogo tipa [Comparative analysis of centralized and decentralized algorithms for the movement of a multicopter-type UAV system], *Izvestiya YuFU. Tekhnicheskie nauki [Izvestiya SFedU. Engineering Sciences]*, 2022, No. 1 (225), pp. 121-139.

64. Pshikhopov V.Kh., Medvedev M.Yu., Gaiduk A.R., Fedorenko R.V., Krukhmalev V.A., Gurenko B.V. Position-Trajectory Control System for Unmanned Robotic Airship, *IFAC Proceedings Volumes (IFAC-PapersOnline). The 19th World Congress the International Federation of Automatic Control. Cape Town, South Africa. August 24-29, 2014*, pp. 8953-8958.
65. Pshikhopov V.Kh., Medvedev M.Yu. Sintez sistem upravleniya podvodnymi apparatami s nelineynymi kharakteristikami ispolnitel'nykh organov [Synthesis of control systems for underwater vehicles with nonlinear characteristics of executive bodies], *Izvestiya YuFU. Tekhnicheskie nauki [Izvestiya SFedU. Engineering Sciences]*, 2011, No. 3 (116), pp. 147-156.

Статью рекомендовал к опубликованию к.т.н., с.н.с. В.А. Шевченко.

**Медведев Михаил Юрьевич** – НИИ робототехники и процессов управления Южного федерального университета; e-mail: medvmihal@sfedu.ru; г. Таганрог, Россия; тел.: 88634371694; д.т.н.; в.н.с.

**Пшихопов Вячеслав Хасанович** – e-mail: pshichop@rambler.ru; д.т.н.; профессор; директор.

**Бросалин Дмитрий Олегович** – e-mail: brosalin@sfedu.ru; инженер.

**Гуренко Борис Викторович** – e-mail: borisgurenko@sfedu.ru; к.т.н.; с.н.с.

**Васильева Мария Александровна** – Южный федеральный университет; e-mail: marv@sfedu.ru; г. Таганрог, Россия; тел.: 88634371694; магистрант.

**Хамдан Низар** – e-mail: dr.nizar.abou.hamdane@gmail.com; аспирант.

**Medvedev Mikhail Yurjevich** – R&D Institute of Robotics and Control Systems; e-mail: medvmihal@sfedu.ru; Taganrog, Russia; phone: +78634371694; dr. of eng. sc.; leading researcher.

**Pshikhopov Viacheslav Khasanovich** – e-mail: pshichop@rambler.ru; dr. of eng. sc.; professor; director.

**Brosalin Dmitry Olegovich** – e-mail: brosalin@sfedu.ru; engineer.

**Gurenko Boris Viktorovich** – e-mail: borisgurenko@sfedu.ru; cand. of eng. sc.; senior researcher.

**Vasileva Maria Aleksandrovna** – Southern Federal University; e-mail: marv@sfedu.ru; Taganrog, Russia; phone: +78634371694; student.

**Hamdan Nizar** – e-mail: dr.nizar.abou.hamdane@gmail.com; post-graduate student.

УДК 004.021

DOI 10.18522/2311-3103-2022-3-192-201

**И.О. Шепель**

## **ОБНАРУЖЕНИЕ ДИНАМИЧЕСКИХ ОБЪЕКТОВ НА КАРТЕ ЗАНЯТОСТИ С НАКОПЛЕНИЕМ НА ОСНОВЕ ФИЛЬТРА ЧАСТИЦ**

*Рассматривается проблема обнаружения динамических препятствий на карте занятости, полученной по данным системы технического зрения мобильной робототехнической платформы. Целью работы является качественное улучшение алгоритма обнаружения препятствий с помощью добавления фильтра частиц для выделения движущихся объектов по данным карты. В исследовании решается задача корректного накопления данных в карте занятости и уменьшения задержки обновления информации в ячейках карты, по которым двигаются динамические объекты. Представленная в статье модификация фильтра частиц способна корректно работать с динамическими препятствиями в широком диапазоне скоростей, устойчива к выбросам, вызванным в результате случайной генерации начальной скорости частиц, и предназначена для работы в реальных условиях в среде с большим количеством движущихся препятствий и в реальном масштабе времени. Разработана эвристика, которая уменьшает количество неправильных классификаций в окклюзированных зонах. Показано, что алгоритм обнаружения динамических объектов в*