# Задание 1

## Abstract

Структурные части:

**Purpose and motivation + Problem**

Rapidly-exploring Random Tree Star (RRT\*) algorithm and its variants based on random sampling can provide a collision-free and asymptotic optimal solution for many path planning problems. However, many RRT\* based variants have low sampling efficiency and slow convergence rate in the environment which consists of long corridors, due to a large number of iterations are required in sampling critical nodes.

**Approach**

To overcome this problem, the paper proposes the Expanding Path RRT\* (EP-RRT\*) based on heuristic sampling in path expansion area. By combining the greedy heuristic of Rapidly exploring Random Tree (RRT)-Connect, EP-RRT\* quickly explores the environment in order to find a feasible path, and then expands it to obtain the heuristic sampling area. It iteratively searches in the heuristic sampling area which also changes with the continuous optimization of the path, and finally obtains an optimal or suboptimal path connecting starting point and target point.

**Results + Conclusion**

Comparisons of EPRRT\* with RRT\* and Informed RRT\* in four simulation environments verify that EP-RRT\* improves the node utilization, accelerates the convergence rate, and obtains a better path for the same number of iterations.

***Перевод:***

Алгоритм Rapidly-exploring Random Tree Star (RRT\*) и его вариации, основанные на случайном сэмплировании, может обеспечить нахождение асимптотически оптимального решения без коллизий для различных задач планирования движения. Однако многие методы, основанные на RRT\*, имеют малую эффективность сэмплирования и низкую скорость сходимости в среде, состоящей из длинных коридоров, поскольку при сэмплировании критических узлов требуется большое количество итераций. Чтобы решить эту проблему, в работе предлагается метод Expanding Path RRT\* (RRT\* с расширением пути, EP-RRT\*), основанный на эвристическом сэмплировании в области расширения пути. Применяя эвристику жадного алгоритма RRT-Connect, EP-RRT\* эффективно осматривает окрестность и находит допустимый путь, а затем расширяет его, чтобы получить область для выполнения эвристического сэмплирования. Метод итеративно выполняет поиск в динамически изменяющейся области эвристического сэмплирования, после чего находит оптимальный или субоптимальный путь, соединяющий начальную и конечную точку. Сравнение EPRRT\* с RRT\* и Informed RRT\* в четырех средах моделирования показывает, что EP-RRT\* улучшает использование узлов пути, повышает скорость сходимости и обеспечивает лучший путь при равном числе итераций.

## 1.2 Introduction + Related work

***Конспект:***

Мобильные роботы, широко применяющиеся в различных областях, привлекают к себе значительное внимание. Одной из технологий в этой области является задача планирования пути, связанная с поиском маршрута между начальным и конечным состоянием с учетом заданного критерия оптимизации и ограничений.

На сегодняшний день данная задача активно изучается, и для ее решения предложено множество подходов. Так, алгоритм потенциальных полей (artificial potential field algorithm, APF) основан на моделировании движущегося агента как частицы в потенциальном поле; конечная точка обладает минимальным потенциалом, которого и стремится достичь частица, в то же время вокруг каждого препятствия потенциал повышается и убывает с удалением от препятствий, в результате чего частица стремится их облетать, двигаясь в сторону цели. Метод дает плавные траектории, однако может попасть в точку локального минимума потенциала.

Бионические алгоритмы, вдохновленные самоорганизующимися природными системами, такие как генетические или муравьиные алгоритмы, хорошо применимы, но ограничены низкой скоростью работы и могут привести к локально оптимальному решению.

Алгоритмы поиска по решетке (обычно, A\*) требуют дискретизируемого пространство небольшой размерности.

Алгоритмы, основанные на сэмплировании, такие как RRT, хорошо применимы для пространств большой размерности и не требуют явного моделирования пространства. Однако RRT лишь вероятностно полон, имеет низкую скорость сходимости и может дать неоптимальный результат. Существуют различные модификации RRT, которые будут рассмотрены далее. Предлагаемое решение основано на нескольких модификациях RRT: быстром RRT-Connect и RRT\*, который значительно оптимизирует путь. В решении используется стратегия эвристического сэмплирования из определенных областей, что повышает скорость сходимости и использование взятых узлов.

В работе используется следующая постановка задачи. Пусть – конфигурационное пространство размерности , ; – множество препятствий, – пустое пространство. Начальная и конечная точки . Путь отражает непрерывная функция .

Если , путь не пересекает препятствия, и задача состоит в нахождении такого пути между точками , называемого достижимым. Множество достижимых путей . Вводится функция оценки стоимости пути .

Задача планирования оптимального пути состоит в нахождении достижимого пути , имеющего минимальную стоимость:

***Важные статьи из списка литературы:***

В каждой из работ рассматривается задача поиска оптимального пути.

1. LaValle, Steven M., Rapidly-exploring random trees: A new tool for path planning

В работе предлагается последовательно строить дерево с корнем в начальной точке . Каждый узел дерева – точка в пространстве. Алгоритм состоит в том, что до тех пор, пока ни один из узлов дерева нельзя соединить с конечной точкой, в дерево добавляется новый лист:

1. в пространстве выбирается случайная точка ;
2. в дереве выбирается узел , ближайший к ;
3. от в сторону строится отрезок: происходит постепенное движение из точки *K* с малым сдвигом, до тех пор, пока не достигнута (в этом случае устанавливается )или не произошло коллизии с препятствием в точке ; в результате в дерево добавляется как потомок узла .

Метод является вероятностно полным, но он неоптимален. Для многомерных пространств метод оказывается быстрым по сравнению с другими подходами.

2. J.J. Kuffner, S.M. LaValle, RRT-connect: An efficient approach to single-query path planning

Метод предполагает для повышения эффективности классического алгоритма RRT использовать двунаправленный поиск. В алгоритме используется пара деревьев , корням которых соответствуют начальная конфигурация и конечная .

На каждом шаге одно из деревьев дополняется новыми бесконфликтными конфигурациями в соответствии с алгоритмом RRT. Однако вместо периодических попыток включения выполняется операция сращивания деревьев. При каждом включении новой вершины в предпринимается аналогичная попытка включить ее и в .

При выполнении некоторого количества шагов деревья меняются ролями, что обеспечивает их более сбалансированный рост. Алгоритм построен таким образом, что предпочтение отдается тому дереву, распространение которого затруднено (например, в связи с наличием большого количества препятствий в его области). Количественным критерием в данном случае может служить общее число вершин дерева или суммарная длина ребер.

3. Karaman S., Frazzoli E. Sampling-based algorithms for optimal motion planning

В работе предлагается эвристическая модификация RRT\* базового алгоритма RRT. В ходе построения дерево поиска дополняется значениями стоимости пути, ведущего в каждую из его вершин. Как и в базовом алгоритме, на каждом шаге для случайным образом выбранной точки ищется ближайшая вершина дерева , а также точка , полученная в результате распространения из в .

Далее определяется подмножество вершин дерева, лежащих внутри шара радиуса с центром в . Среди них находят вершину такую, что стоимость пути из начальной конфигурации в была минимальной.

После того, как включается в дерево в качестве дочернего узла , проводится локальная оптимизация путей внутри шара. Для каждой конфигурации в нем предпринимается попытка построения пути, ведущего из в через , и в случае, если он имеет меньшую стоимость, то конфигурация становится дочерней вершиной .

Правильный подбор параметра позволяет получить асимптотически оптимальный алгоритм.

4. J.D. Gammell, S.S. Srinivasa, T.D. Barfoot, Informed RRT: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic

В работе предлагается модификация алгоритма RRT\*, которая при сохранении свойств RRT\* ограничивает область сэмплирования новых точек, что приводит к ускорению алгоритма.

Метод предполагает, что было получено некоторое (вообще говоря, неоптимальное) решение, которое необходимо оптимизировать. В то время, как RRT\* добавляет новые точки в дерево с помощью случайного сэмплирования, в Informed RRT\* предлагается ограничить область сэмплирования многомерным эллипсоидом, в котором одна из осей (главная) проходит через точки , а другие оси равны, и их величина уменьшается с уменьшением стоимости лучшего найденного пути. С уменьшением этой стоимости уменьшается и длина главной оси.

Из геометрических соображений очевидно, что предложенный метод отсекает лишь более длинные варианты путей.

# Задание 2

## 2.1 Конспект

Недостатком RRT\* является тот факт, что он требует большое число операций, чтобы найти оптимальный путь. Проблема отчасти решается в Informed RRT\*, однако используемый им многомерный эллипсоид не дает желаемого эффекта в случае, если имеется узкий коридор, через который должен пройти оптимальный маршрут.

Для дальнейшего увеличения скорости сходимости алгоритма в работе предложен метод EP-RRT\*. Перед тем, как найти начальное (неоптимальное) решение, алгоритм работает схожим с RRT-Connect образом. Как только найдено начальное решение (неоптимальный путь), сэмплирование происходит эвристически в определенной области. В листинге 1 приведен псевдокод алгоритма EP-RRT\*.

Листинг 1 — Алгоритм EP-RRT\*

1. ;
2. ;
3. ;
4. for i = 1 to do
6. ;
8. if then





15. end if
16. end for
17. return ;

Алгоритм имеет два основных улучшения по сравнению с RRT\*. Первое состоит в использовании жадной стратегии RRT-Connect для быстрого нахождения пути без коллизий. Второе вдохновлено идеей области многомерного эллипсоида из алгоритма Informed RRT\* и также состоит в построении области сэмплирования для повышения скорости сходимости. В сложных средах, таких как узкие коридоры или лабиринты, лучшие варианты сэмплирования распределены вокруг углов или краев, что учитывается алгоритмом при выборе области сэмплирования.

Главным аспектом эвристического сэмплирования является определение области для него. Поскольку путь представляет собой набор вершин, последовательно соединенных отрезками, область для расширения представляется как набор соединенных четырехугольников. На рисунке 1 приведен пример пути из четырех вершин и трех соединяющих их ребер, а область расширения составлена из восьми вершин, полученных из имеющихся.

Изображение выглядит как диаграмма, линия

Автоматически созданное описание

Рисунок 1. Расширение пути

Для вершины вектор биссектрисы между двумя смежными зонами можно получить как:

Две новые вершины для области расширения пути получаются на основе параметра дистанции расширения как:

где – половина угла поворота пути вокруг вершины .

В листинге 2 приведен псевдокод алгоритма процесса расширения.

Листинг 2 — Процесс расширения

1. for i = 2 to do


5. end for
6. return

В процессе расширения для оптимизации ранее найденного пути важным аспектом является выбор параметра дистанции расширения , который предлагается изменять со временем. Начальная величина параметра основана на размере исследуемой области пространства конфигураций.

где – начальный коэффициент расширения (настраиваемый, в работе используется 8), – соответственно, длина и ширина области. Вводится динамически изменяемый коэффициент расширения :

Здесь – номер текущей итерации, – номер итерации, на которой алгоритм нашел начальный путь с помощью RRT-Connect, – максимальное число итераций. Таким образом, итоговый параметр расширения вычисляется как:

## 2.2 Список обозначений

– конфигурационное пространство размерности ,

– множество препятствий

– пустая область пространства

– начальная и конечная точка,

– путь в конфигурационном пространстве,

– оптимальный путь в конфигурационном пространстве

– множество достижимых путей

– функция стоимости пути

– дистанция расширения пути, скаляр

– начальная дистанция расширения пути, скаляр

– коэффициент расширения, скаляр

– начальный коэффициент расширения, скаляр

– точка в кривой пути,

, – новые вершины для области расширения,

– вектор биссектрисы между двумя смежными зонами,

– половина угла поворота пути вокруг вершины , скаляр

– длина исследуемой области, скаляр

–ширина исследуемой области, скаляр

– номер текущей итерации, скаляр

– номер итерации, на которой алгоритм нашел начальный путь с помощью RRT-Connect, скаляр

– максимальное число итераций, скаляр.

## 2.3 Блок-схемы

На рисунках 2, 3 приведены блок-схемы алгоритмов EP-RRT\* и процесса расширения соответственно.

Изображение выглядит как текст, диаграмма, зарисовка, рисунок

Автоматически созданное описание

Рисунок 2. Блок-схема алгоритма EP-RRT\*

Изображение выглядит как текст, диаграмма, зарисовка, снимок экрана

Автоматически созданное описание

Рисунок 3. Блок-схема алгоритма расширения

# Задание 3

## 3.1 Конспект постановки эксперимента, исходные данные

Для проведения экспериментов были смоделированы четыре среды 1000\*1000 пикселей, приведенные на рисунках 4(а) – 4(г). На рисунках розовым цветом обозначена начальная точка, зеленым – конечная. Для каждой из сред исследовался каждый из алгоритмов: RRT\*, Informed RRT\*, EP-RRT\*.

Изображение выглядит как текст, снимок экрана, диаграмма, дизайн

Автоматически созданное описание

Рисунок 4. Среды для поиска пути

Каждый алгоритм запускался независимо 100 раз с максимальным числом итераций (сэмплирований) 2000. Для каждого запуска вычислялись величины: – число итераций, необходимое для получения субоптимального пути стоимостью , где – стоимость оптимального пути; кроме того, вычислялась величина – стоимость итогового пути, полученного алгоритмом.

Для запуска использовалась конфигурация: Matlab R2018b (9.5.0.944444); 64 bit Windows 10; процессор Intel(R) Core(TM) i5–2310 с частотой 2.9 GHz; 12 GB ОЗУ.

## 3.2 Описание графиков

***Обычная среда***

На рисунке 5 приведен график зависимости стоимости найденного оптимального пути от числа итераций для каждого алгоритма.

Изображение выглядит как текст, линия, График, диаграмма

Автоматически созданное описание

Рисунок 5. Зависимость стоимости оптимального пути от числа итераций в обычной среде

Для каждой кривой можно видеть тенденцию к снижению стоимости оптимального пути с увеличением числа итераций, по своему характеру это изменение выглядит как экспоненциальное затухание. График EP-RRT\* в конце оказывается близок к оптимальному значению, график Informed RRT\* оказывается дальше, и RRT\* наиболее удален от оптимума.

Можно видеть, что и по скорости сходимости, и по стоимости найденного оптимального пути лучший результат дает EP-RRT\*, хуже работает Informed RRT\*, и еще хуже RRT\*. Примечательно, что для кривой каждого из алгоритмов первая производная становится меньше 1 при приблизительно одном и том же значении Path cost: 1050 – 1060, однако EP-RRT\* достигает этой точки приблизительно на 500 итерации, Informed RRT\* и RRT\* – около 1000.

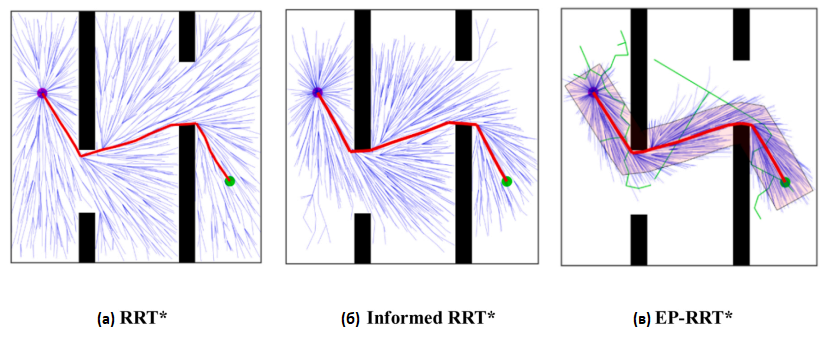


Рисунок 6. Поиск пути в обычной среде

На рисунках 6(а – в) приведены результаты запуска алгоритмов в обычной среде. Можно видеть, что как и предполагает эвристика каждого из методов, RRT\* развивает дерево во всех направлениях, Informed RRT\* сэмплирует точки из эллиптической области, которая, впрочем, слишком широка, и в дереве присутствует много лишних ветвей, а EP-RRT\* сэмплирует точки вокруг оптимального пути.

На рисунке 7 приведены боксплоты для величин  *и* . Боксплоты подтверждают выводы, сделанные из графиков на рисунке 5. Примечательно также то, что у Informed RRT\* и RRT\* с ухудшением качества работы растет так же и дисперсия величин и.

Изображение выглядит как текст, диаграмма, линия, снимок экрана

Автоматически созданное описание

Рисунок 7. Боксплоты для и в обычной среде

***Хаотичная среда***

На рисунке 8 приведен график зависимости стоимости найденного оптимального пути от числа итераций для каждого алгоритма.

Для каждой кривой можно видеть тенденцию к снижению стоимости оптимального пути с увеличением числа итераций, по своему характеру это изменение выглядит как экспоненциальное затухание, однако в RRT\* наблюдается резкое снижение скорости сходимости около 1000 итерации, а затем резкое увеличение около 2700. График EP-RRT\* в конце оказывается близок к оптимальному значению, график Informed RRT\* оказывается дальше, и RRT\* наиболее удален от оптимума

Изображение выглядит как текст, линия, диаграмма, График

Автоматически созданное описание

Рисунок 8. Зависимость стоимости оптимального пути от числа итераций в хаотичной среде

Можно видеть, что и по скорости сходимости, и по стоимости найденного оптимального пути лучший результат дает EP-RRT\*, хуже работает Informed RRT\*, и еще хуже RRT\*.

На рисунках 9 (а – в) приведены результаты запуска алгоритмов в хаотичной среде. Можно видеть, что как и предполагает эвристика каждого из методов, RRT\* развивает дерево во всех направлениях, Informed RRT\* сэмплирует точки из эллиптической области, а EP-RRT\* сэмплирует точки в более узкой зоне вокруг оптимального пути.

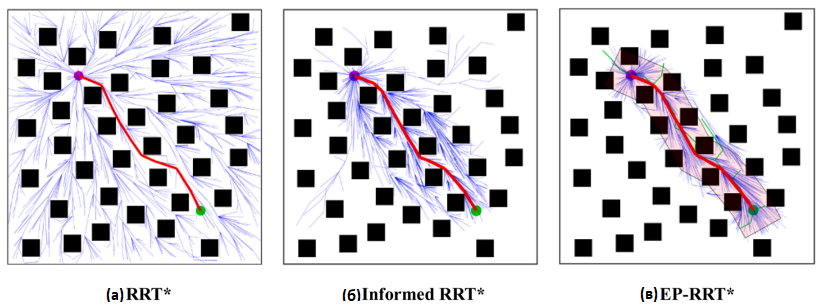


Рисунок 9. Поиск пути в хаотичной среде

На рисунке 10 приведены боксплоты для величин  *и* . Боксплоты подтверждают выводы, сделанные из графиков на рисунке 5. Примечательно также то, что у Informed RRT\* и RRT\* с ухудшением качества работы растет так же и дисперсия величин и.

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

Рисунок 10. Боксплоты для и в хаотичной среде

***Узкий коридор***

На рисунке 11 приведен график зависимости стоимости найденного оптимального пути от числа итераций для каждого алгоритма.

Для каждой кривой можно видеть тенденцию к снижению стоимости оптимального пути с увеличением числа итераций, по своему характеру это изменение выглядит как экспоненциальное затухание, однако в RRT\* наблюдается резкое снижение скорости сходимости около 1000 итерации. График EP-RRT\* в конце оказывается близок к оптимальному значению, графики Informed RRT\* и RRT\* оказываются дальше и соответствуют одинаковому уровню .

Изображение выглядит как текст, линия, диаграмма, График

Автоматически созданное описание

Рисунок 11. Зависимость стоимости оптимального пути от числа итераций в узком коридоре

Можно видеть, что и по скорости сходимости, и по стоимости найденного оптимального пути лучший результат дает EP-RRT\*, хуже работает Informed RRT\*, и еще хуже RRT\*, однако в конце Informed RRT\* и RRT\* дают приблизительно одинаковый результат, а скорость сходимости RRT\* на 5000 итерации оказывается выше.

На рисунках 12 (а – в) приведены результаты запуска алгоритмов в узком коридоре. Можно видеть, что как и предполагает эвристика каждого из методов, RRT\* развивает дерево во всех направлениях, Informed RRT\* сэмплирует точки из эллиптической области, а EP-RRT\* сэмплирует точки в более узкой зоне вокруг оптимального пути.

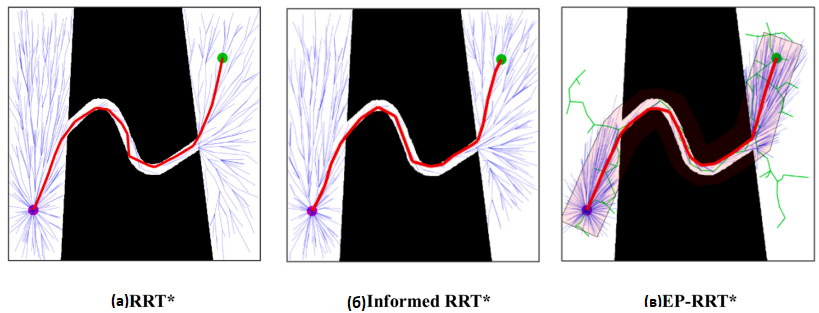


Рисунок 12. Поиск пути в узком коридоре

На рисунке 13 приведены боксплоты для величин  *и* . Боксплоты подтверждают выводы, сделанные из графиков на рисунке 5. Примечательно также то, что у Informed RRT\* и RRT\* с ухудшением качества работы растет так же и дисперсия величин и.

Изображение выглядит как текст, диаграмма, линия, снимок экрана

Автоматически созданное описание

Рисунок 13. Боксплоты для и в узком коридоре

***Лабиринт***

На рисунке 14 приведен график зависимости стоимости найденного оптимального пути от числа итераций для каждого алгоритма.

Для каждой кривой можно видеть тенденцию к снижению стоимости оптимального пути с увеличением числа итераций, по своему характеру это изменение выглядит как экспоненциальное затухание, однако в RRT\* наблюдается резкое снижение скорости сходимости около 1500 итерации, а затем ее увеличение около 4000. График EP-RRT\* в конце оказывается близок к оптимальному значению, график Informed RRT\* оказывается дальше, и еще дальше – график RRT\*.

Изображение выглядит как текст, линия, диаграмма, График

Автоматически созданное описание

Рисунок 14. Зависимость стоимости оптимального пути от числа итераций в лабиринте

Можно видеть, что и по скорости сходимости, и по стоимости найденного оптимального пути лучший результат дает EP-RRT\*, хуже работает Informed RRT\*, и еще хуже RRT\*, но скорость сходимости RRT\* на 5000 итерации оказывается выше.

На рисунках 15 (а – в) приведены результаты запуска алгоритмов в лабиринте. Можно видеть, что как и предполагает эвристика каждого из методов, RRT\* развивает дерево во всех направлениях, Informed RRT\* сэмплирует точки из эллиптической области, а EP-RRT\* сэмплирует точки в более узкой зоне вокруг оптимального пути.

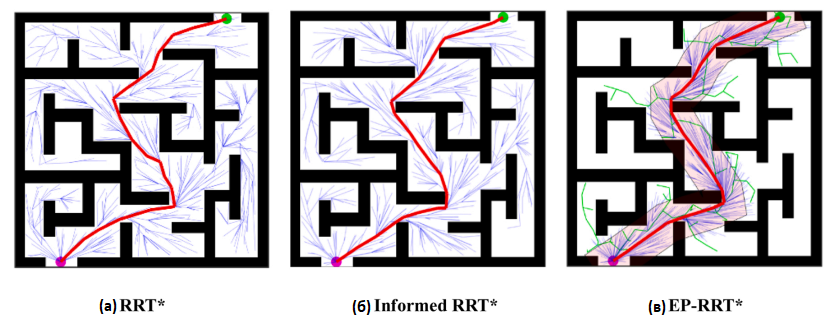


Рисунок 15. Поиск пути в лабиринте

На рисунке 16 приведены боксплоты для величин  *и* . Боксплоты подтверждают выводы, сделанные из графиков на рисунке 5. Примечательно также то, что у Informed RRT\* и RRT\* с ухудшением качества работы растет так же и дисперсия величины .

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

Рисунок 16. Боксплоты для и в лабиринте

# Задание 4

Авторы указывают, что алгоритм EP-RRT\* предложен для того, чтобы устранить недостатки RRT\*, такие как низкая скорость сходимости и лишние ветви в особых средах. Основной идеей алгоритма является использование жадного поиска неоптимального пути с помощью RRT-Connect и дальнейшая оптимизация найденного решения путем рассмотрения определенной окрестности найденного пути.

Авторы указывают, что предложенный метод значительно увеличивает вероятность сэмплирования наиболее важных точек, что повышает скорость сходимости. Метод может быть совмещен с любым алгоритмом, основанном на сэмплировании.

Результаты экспериментов показывают, что предложенный метод по сравнению с прочими быстрее сходится и дает более оптимальное решение, в особенности, в случае узких коридоров и лабиринтов.

Однако несмотря на преимущества метода, он имеет свои ограничения в производительности. В средах, где есть множество альтернативных путей различной стоимости, алгоритм требует области расширения больших размеров, чтобы избежать попадания в локальный оптимум.

Дальнейшие исследования могут быть направлены на изучение возможности адаптивной настройки параметров предложенного алгоритма.