



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И
УПРАВЛЕНИЯ

ЛАБОРАТОРНАЯ РАБОТА № 8

«Алгоритм А*»

ДИСЦИПЛИНА: «Основы программирования»

Выполнил: студент гр. ИУ5-14Б

Корнеев Г. И.
(Подпись) (Ф.И.О.)

Проверил:

Колосов М. И.
(Подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

2025 г.

Цель работы.

Разработать и исследовать алгоритмы поиска безопасного пути в лесной местности во время пожара, обеспечивая обход очагов возгорания и непроходимых зарослей, а также оценить эффективность различных методов поиска пути в условиях сложной среды.

Постановка задачи.

На вход программе подаётся карта лесной местности, представленная в виде двумерной сетки, содержащей свободные клетки, препятствия (очаги пожара и заросли), стартовую и целевую точки. Требуется построить безопасный маршрут от старта к цели, минимизируя длину пути и обходя непроходимые клетки.

Тема работы.

В данной лабораторной работе исследуется задача поиска безопасного пути через лесную местность во время пожара с учётом динамических препятствий. Основная цель — реализовать и сравнить алгоритмы поиска пути (A^* , A^* с пост-сглаживанием, Θ^*) в условиях наличия очагов возгорания и непроходимых зарослей. В работе проводится анализ эффективности алгоритмов и визуализация найденных маршрутов.

Выполнение ЛБ.

В данной работе сравниваются три алгоритма поиска пути: A^* , A^*PS и Θ^* .

A^* — классический информированный алгоритм поиска, основанный на оценке функции стоимости $f(n) = g(n) + h(n)$, где $g(n)$ — длина пути от старта, а $h(n)$ — эвристическая оценка расстояния до цели. Алгоритм гарантирует нахождение оптимального пути при допустимой эвристике.

A^*PS — модифицированная версия A^* , в которой после построения маршрута выполняется пост-сглаживание (post-smoothing). Этот этап уменьшает количество поворотов, делая путь более прямолинейным без изменения основной логики поиска.

Θ^* — алгоритм любой-угловой навигации (any-angle pathfinding), который использует проверку прямой видимости между вершинами. Благодаря этому Θ^* строит более короткие и естественные маршруты, допускающие произвольные углы движения.

Структура проекта в VS Code представлена на рисунке 1.

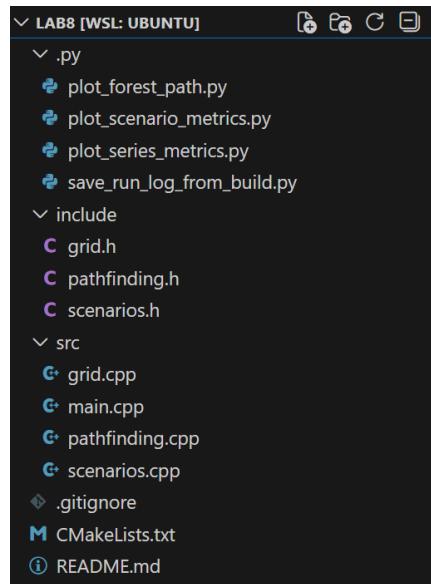


Рисунок 1 – Структура проекта.

В программе предусмотрены три режима работы, обеспечивающие комплексную оценку алгоритмов поиска пути.

1. Интерактивный режим позволяет продемонстрировать работу алгоритма на одной карте и провести первичный анализ результата.
2. Серийный режим обеспечивает выполнение множества запусков на случайно генерируемых картах, что необходимо для получения статистически значимых показателей эффективности.
3. Режим фиксированных сценариев предоставляет возможность сравнить алгоритмы в строго определённых, повторяемых условиях, моделирующих типовые структуры местности.

Совокупность этих режимов обеспечивает полноту исследования и позволяет объективно оценить поведение алгоритмов при различных конфигурациях среды. На рисунке 2 представлен главный «экран» вывода программы в терминале VS Code.

```

korneev@LENOVO:~/lab8$ rm -rf build && rm -rf data && cmake -B build && cmake
hon3 .py/save_run_log_from_build.py && python3 .py/plot_forest_path.py
Лабораторная работа 8.
Поиск безопасного пути через лес во время пожара.

Режим работы:
1 – Один запуск (интерактивный)
2 – Серия запусков на случайных картах (статистика)
3 – Тестирование на заранее заданных картах (open field, лабиринт, и др.)
Выберите режим: 1

```

Рисунок 2 – Вывод в терминале, часть 1 (главный экран).

В режимах работы пользователь может выбирать параметры карты.

Ведите высоту и ширину карты (например, 10 15):

```

korneev@LENOVO:~/lab8$ rm -rf build && rm -rf data && cmake -B build && cmake
Режим работы:
1 – Один запуск (интерактивный)
2 – Серия запусков на случайных картах (статистика)
3 – Тестирование на заранее заданных картах (open field, лабиринт, и др.)
Выберите режим: 1
Ведите высоту и ширину карты (например, 10 15): 15 15

```

Сгенерированная карта:

```

S.....##.....
.....###.....
....#.##....#...
..#.....#....
.#.##...#####...
.....#....#.
.....##....#.
#....#....##..#
##.##....#.
....##....#.
.....##....#.
.##.##....#.
.....##....#.
#....#....#...
.....##....#.
.#.....#...
##...#....#...G

```

Выберите алгоритм:

```

1 – A*
2 – A*PS
3 – Theta*
1

```

Карта с путём:

```

S.*.....##.....
.*.*##.....
...###....#...
..#..*....#...
.#.##.*.#####..
.....##*....#
.....*....#.
#....#...*##..#
##.##...*....#
....##...*.....
.#..##.####..*..
#....#.##.*..
.....##...*#.
.#.....*.
##...#.##...G

```

==== Метрики ===

```

L_opt=21.799
L_found=21.799
K0=1
00%=0
Closed=67
EP=3.07354
Лог из build/run_output.txt добавлен в data/run_log.md
Картина сохранена в data/png/forest_path.png

```

Рисунок 4 – Демонстрация работы режима 1 на карте
15*15 (часть1)

Рисунок 3 – Настройка параметров карты

Рисунок 5 – Демонстрация работы режима 1
на карте 15*15 (часть2)

Режим 1 — интерактивный запуск: в интерактивном режиме выполняется один запуск алгоритма на случайно сгенерированной карте. Пользователь сразу видит структуру лесной местности, расположение препятствий и итоговый найденный путь. Также с помощью Python скрипта создается визуализация карты. Данный режим позволяет наглядно продемонстрировать работу алгоритма и провести первичный анализ его поведения.

Также с помощью Python-скрипта создается визуализация карты, которая представлена на рисунке 6.

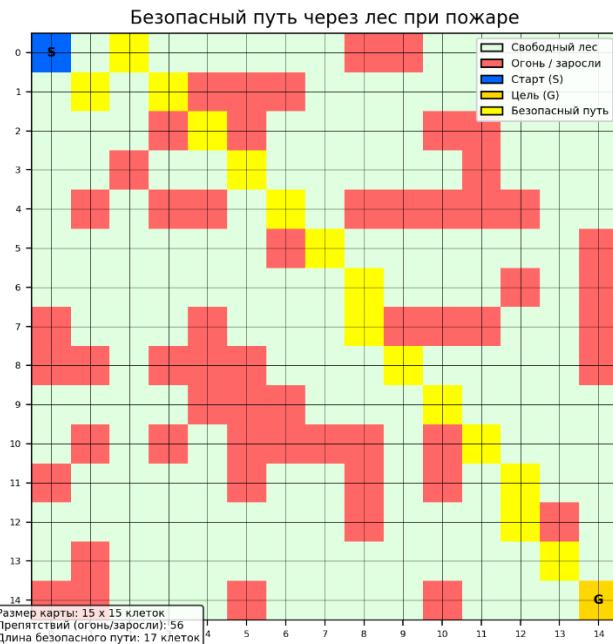


Рисунок 6 – Визуализация карты работы режима 1.

Режим 2 — серия запусков (статистика): В этом режиме выполняется множество последовательных запусков на случайных картах, результаты которых автоматически собираются в таблицу. Также в этом режиме программа автоматически строит набор визуализаций, отражающих статистику работы алгоритмов на множестве случайных карт:

1. **series_L_found.png** — длина найденного пути для каждого алгоритма по всем картам.
2. **series_EP.png** — показатель эффективности (EP) по каждой карте.
3. **series_closed.png** — количество раскрытых узлов (число посещённых клеток) по всем запускам.
4. **series_success_counts.png** — сколько раз каждый алгоритм успешно находил путь.

Данный режим обеспечивает получение статистически значимых данных (длина пути, количество раскрытых узлов, успешность), что необходимо для объективного сравнения алгоритмов. Далее представлены некоторые из генерируемых визуализаций.

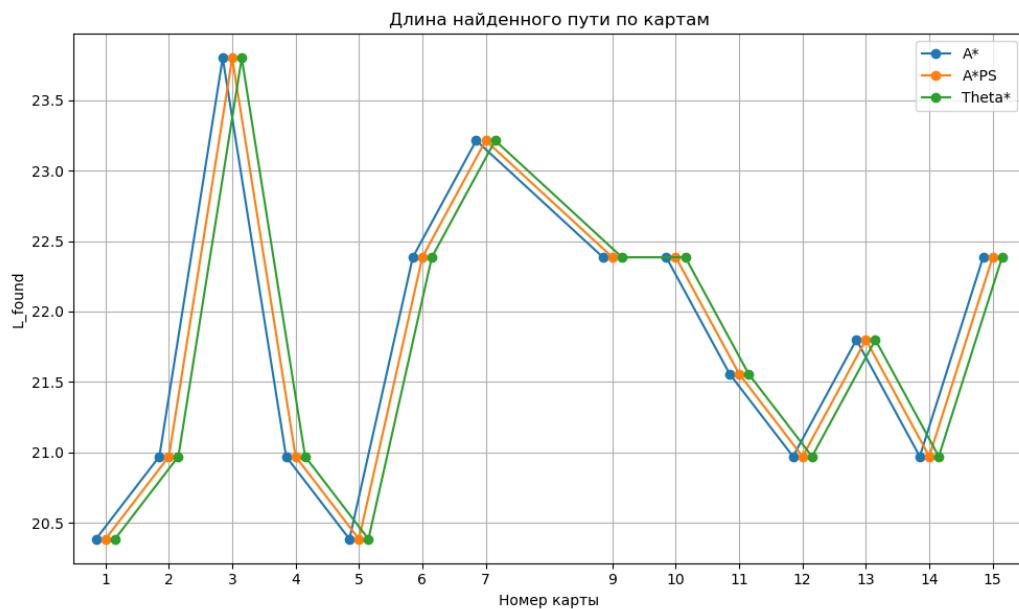


Рисунок 7 – График длины найденного пути (series_L_found.png)

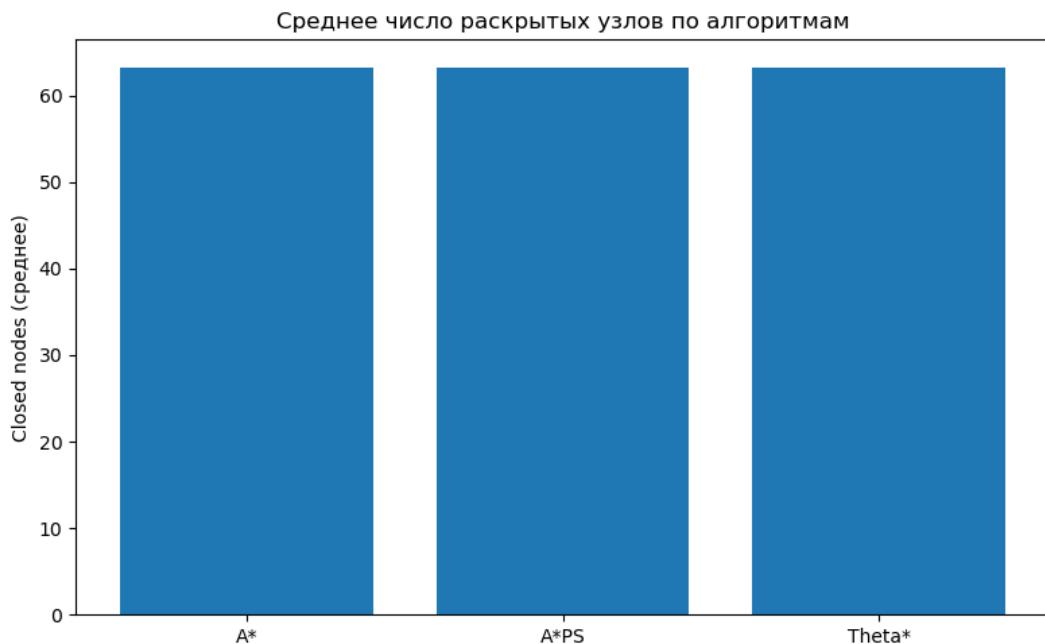


Рисунок 8 – График среднего количества раскрытых узлов (series_closed_mean.png)

Режим 3 — заранее заданные сценарии: Алгоритмы тестируются на фиксированных, заранее подготовленных картах (open field, лабиринт, плотные препятствия и т.д.). В данном режиме работы программы средствами Python генерируется несколько визуализаций:

1. **scenarios_L_found.png** — длина найденного пути для каждого алгоритма на фиксированных картах.
2. **scenarios_EP.png** — эффективность поиска (EP) по сценариям.
3. **scenarios_closed.png** — число раскрытых узлов в каждом сценарии.

4. **scenarios_maps.png** — изображения самих сценариев (открытое поле, лабиринт, плотные застройки, узкие коридоры).

Данный режим позволяет сравнить работу алгоритмов в типовых, повторяемых конфигурациях среды и выявить особенности поведения в разных структурах местности. Далее на рисунках представлены некоторые из генерируемых визуализаций.

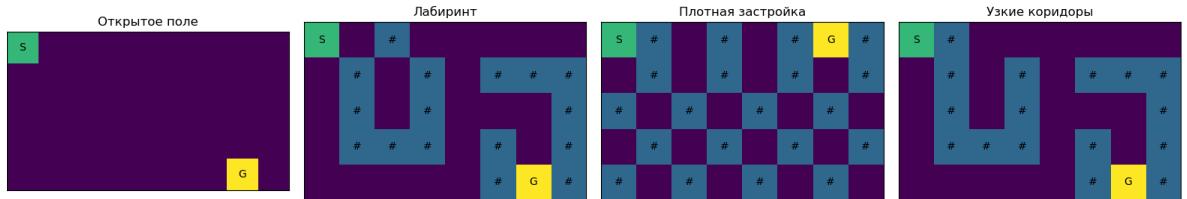


Рисунок 9 – Сценарные карты (scenarios_maps.png).

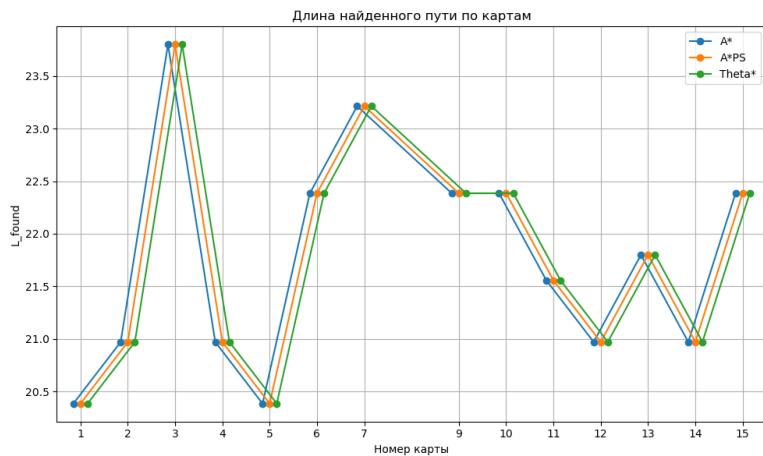


Рисунок 10 — Длина найденного пути по сценариям (scenarios_L_found.png).

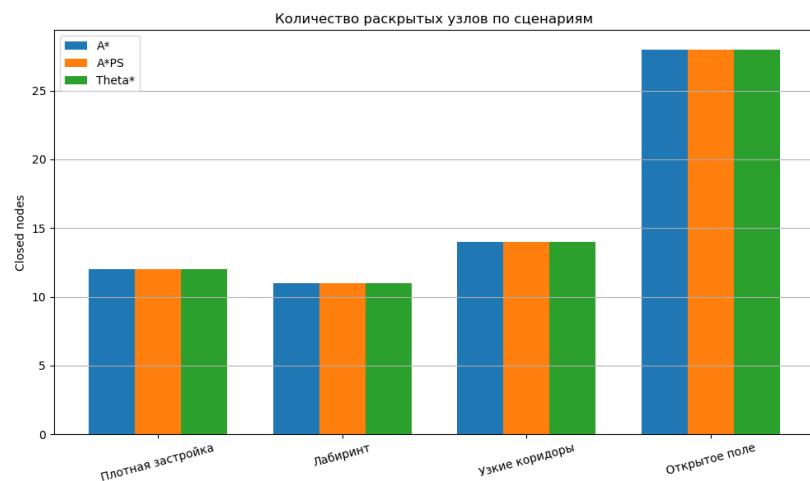


Рисунок 11 — Число раскрытых узлов по сценариям (scenarios_closed.png).

На основе статистических данных второго режима работы программы была сформирована сводная таблица средних метрик для алгоритмов A*, A*PS и Theta*. Таблица включает среднюю длину найденного пути, среднее количество раскрытых узлов, показатель эффективности EP и число успешных запусков.

	A	B	C	D	E
1	Algorithm	L_found_mean	Closed_mean	EP_mean	Success_count
2	A*	22,04	52,20	2,45	15
3	A*PS	22,04	52,20	2,45	15
4	Theta*	22,04	52,20	2,45	15

Таблица 12 — Сравнение средних метрик алгоритмов поиска пути по результатам серии запусков.

Вывод

В ходе выполнения работы были реализованы и исследованы три алгоритма поиска пути: A*, A*PS и Theta*. Проведённые эксперименты в трёх режимах — интерактивном, статистическом и сценарном — позволили всесторонне оценить их поведение в условиях лесной местности с препятствиями. Результаты серии запусков показали, что алгоритмы демонстрируют схожие показатели длины пути и количества раскрытых узлов на случайно генерируемых картах. В заранее заданных сценариях Theta* формировал более прямые маршруты, что соответствует его природе any-angle навигации. Пост-сглаживание в A*PS улучшало плавность пути без значительного влияния на вычислительную сложность. Полученные визуализации и таблицы метрик подтверждают корректность работы алгоритмов и позволяют оценить их эффективность в различных конфигурациях среды.

Работа на GitHub - <https://github.com/gregkorneev/lab8.git>



СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Быков, А. Ю. Решение задач на языках программирования Си и Си++ : методические указания / А. Ю. Быков. — Москва : МГТУ им. Н.Э. Баумана, 2017. — 248 с. — ISBN 978-5-7038-4577-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/103505>
2. Каширин, И. Ю. От Си к Си++ : учебное пособие / И. Ю. Каширин, В. С. Новичков. — 2-е изд., стер. — Москва : Горячая линия-Телеком, 2012. — 334 с. — ISBN 978-5-9912-0259-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/5161>
3. Быков А. Ю. Решение задач на языках программирования Си и Си++ : метод. указания к выполнению лаб. работ / Быков А. Ю. ; МГТУ им. Н. Э. Баумана. - М. : Изд-во МГТУ им. Н. Э. Баумана, 2017. - 244 с. : ил. - ISBN 978-5-7038-4577-6.
4. Иванова Г. С., Ничушкина Т. Н. Объектно-ориентированное программирование : учебник для вузов / Иванова Г. С., Ничушкина Т. Н. ; общ. ред. Иванова Г. С. - М. : Изд-во МГТУ Н.Э.Баумана, 2002. <http://progbook.ru/technologiya-programmirovaniya/582-ivanova-tehnologiya-programmirovaniya.html>
5. Иванова Г. С., Ничушкина Т. Н. Объектно-ориентированное программирование : учебник для вузов / Иванова Г. С., Ничушкина Т. Н. ; общ. ред. Иванова Г. С. - М. : Изд-во МГТУ им. Н. Э. Баумана, 2014. - 455 с. : ил. - Библиог.: с. 450. - ISBN 978-5-7038-3921-8.
6. Подбельский В.В. Язык Си++: Учебное пособие. – М.: Финансы и статистика, 2003.
<http://progbook.ru/c/737-podbelskii-programmiovanie-na-yazyke-si.html>.
7. Акулов О.А., Медведев Н.В. Информатика. Базовый курс. – М.: Омега-Л, 2006. <http://razym.ru/naukaobraz/obrazov/151874-akulov-oa-medvedev-nv-informatika-bazovyy-kurs.html>

8. Вычислительные методы и программирование. МГУ им. М.В. Ломоносова. ISSN 1726-3522. Журнал входит в 1-й уровень Белого списка научных журналов Минобрнауки России. <https://num-meth.ru/index.php/journal/index>

Дополнительные материалы

1. Иванова Г.С. Технология программирования: Учебник для вузов. – М.: Изд. МГТУ им. Ахо А.В., Хопкрофт Д.Э., Ульман Д.Д. Структуры данных и алгоритмы. – М., Вильямс, 2003. <http://razym.ru/naukaobraz/obrazov/181547-aho-a-ulman-d-hopcroft-d-struktury-dannyh-i-algoritmy.html>
2. Дейтел Х.М., Дейтел П.Дж. Как программировать на C++. – М.: Бином, 2001.
3. Т. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. – М. МЦНМО, 2005. <http://rutracker.org/forum/viewtopic.php?t=533181>
4. Джосьютис Н. С++ Стандартная библиотека для профессионалов. – СПб.: Питер, 2004. http://progbook.ru/c/178-dzhosyutis_c_standartnaya_biblioteka.html
5. Подбельский В.В. Стандартный Си++: Учебное пособие. – М.: Финансы и статистика, 2008.
6. Объектно-ориентированное программирование в С++: пер. с англ. / Лафоре Р. - 4-е изд. - СПб.: Питер, 2004. - 923 с. - (Классика computer science). - ISBN 5-94723-302-9.
7. Т. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. – М. МЦНМО, 2005.
8. Г. Шилдт. С++. Базовый курс, 3-е издание: Пер с англ. – М.: Издательский дом «Вильямс», 2011. – 624 с.
9. Павловская Т. А. С/C++. Программирование на языке высокого уровня: учебник для вузов / Павловская Т. А. - СПб.: Питер, 2003. - 460 с. - (Учебник для вузов). - ISBN 5-94723-568-4.
10. Бесплатные образовательные программы партнера (VK):
<https://education.vk.company/students>