

МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА
ВЕЛИКОГО»

Институт компьютерных наук и кибербезопасности

Высшая школа технологий искусственного интеллекта

РЕФЕРАТ

на тему: «Муравьиный алгоритм построения пути»

Дисциплина: Генетические алгоритмы

Студент: _____

Салимли Айзек Мухтар Оглы

Преподаватель: _____

Большаков Александр Афанасьевич

«____»_____ 20___ г.

Санкт-Петербург, 2025

Содержание

Введение	3
1 Постановка задачи	4
1.1 Математическая формулировка	4
1.2 Ограничения и требования	4
1.3 Ключевые подзадачи навигации	4
2 Типы планирования пути	5
2.1 Глобальное планирование (статическая среда)	5
2.2 Локальное планирование (динамическая среда)	5
2.3 Гибридный подход	5
3 Представление окружающей среды и модели	6
3.1 Решётка (grid)	6
3.2 MAKLINK-модель свободного пространства	6
3.3 Преимущества и недостатки моделей	7
4 Муравьиный алгоритм: базовые принципы и интуиция	9
5 Ant Colony System (ACS): формализация	10
5.1 Правило выбора перехода	10
5.2 Локальное обновление феромона	10
5.3 Глобальное обновление феромона	10
5.4 Критерии и останов	10
6 Параметры алгоритма и практическая настройка	11
6.1 Основные параметры	11
6.2 Влияние параметров на поведение	11
6.3 Экспериментальная настройка (пример)	11
7 Применение к решению на дискретных MAKLINK-линиях	12
7.1 Дискретная параметризация	12
7.2 MA как оптимизатор в пространстве смещений	12
7.3 Алгоритмическая схема	12
8 Псевдокод детального алгоритма (шаги)	13
9 Эксперименты и численные примеры	14
10 Анализ влияния числа направлений движения	16
10.1 Стандартные случаи	16
10.2 Расширенные варианты и их эффект	16
11 Гибридные подходы и преимущества интеграции методов	17
11.1 Мотивы для гибрида	17
11.2 Описание примерной схемы	17
11.3 Числовые примеры эффективности	17
11.4 Пример свободного пространства	17
Заключение	20
Список литературы	21

Введение

Планирование пути мобильного робота является фундаментальной задачей автономной робототехники. Под задачей понимается вычисление траектории, которая переводит робота из заданной начальной позиции S в целевую позицию T в некотором рабочем пространстве, содержащем стационарные и/или подвижные препятствия. Критериями качества решения обычно являются: отсутствие столкновений, минимальная длина траектории, минимальное время движения, безопасность движения и соблюдение динамических ограничений робота.

Рассматривается подход к планированию пути, основанный на муравьином алгоритме (Ant Colony System, ACS), а также его гибридные комбинации с классическими алгоритмами графового поиска и методами построения свободного пространства (MAKLINK). Муравьиный подход реализует стохастический многократный поиск, опирающийся на коллективную память в виде искусственного феромона. Этот подход является пригодным для задач с большой размерностью пространства и сложной структурой препятствий, поскольку демонстрирует хорошую способность находить качественные пути и быстро сходиться при корректной параметризации.

Введение в проблематику включает следующие ключевые положения:

- отличие статической (полностью известной) и динамической (частично неизвестной) среды и связанные с этим режимы планирования;
- роль представления окружения (решётка/grid, MAKLINK-граф) для постановки дискретизированной задачи;
- мотивация использования стохастических методов (МА) и комбинированных подходов (гибрид Дейкстра + МА) для повышения качества и надёжности путей.

1 Постановка задачи

1.1 Математическая формулировка

Пусть \mathcal{W} — рабочее пространство робота, $\mathcal{O} = \bigcup_k O_k$ — множество областей препятствий. Требуется построить дискретизированную последовательность точек

$$\mathcal{P} = \{P_0 = S, P_1, \dots, P_n = T\},$$

такую, что для каждого сегмента:

$$\text{seg}(P_{i-1}, P_i) \cap \mathcal{O} = \emptyset.$$

Цель оптимизации может формулироваться, например, как минимизация длины пути:

$$L(\mathcal{P}) = \sum_{i=1}^n \|P_i - P_{i-1}\| \rightarrow \min.$$

1.2 Ограничения и требования

Задача планирования должна учитывать:

- геометрические размеры и формы препятствий;
- динамические ограничения робота (максимальная скорость, радиус разворота и т.п.), если они включены;
- необходимость работы в статических и динамических условиях (т.е. возможность обработки новых препятствий, появляющихся в процессе движения);
- требования к вычислительным затратам и числу итераций для получения решения в заданном времени.

1.3 Ключевые подзадачи навигации

При решении общей задачи принято выделять четыре взаимосвязанные подзадачи:

1. локализация робота в пространстве;
2. построение и обновление карты окружения;
3. планирование пути (поиск безопасной траектории);
4. управление движением (следование по траектории).

В данной работе основной акцент делается на задаче (3) — планировании пути.

2 Типы планирования пути

2.1 Глобальное планирование (статическая среда)

Глобальное планирование предполагает, что всю информацию о среде (координаты и формы препятствий) можно задать заранее. Тогда задача решается следующим образом: строится полная модель пространства и на ней выполняется алгоритм поиска пути (например, на графе, полученном из сеточной дискретизации или из MAKLINK-графа). Глобальное планирование позволяет гарантировать отсутствие столкновений при условии корректности карты, и позволяет оптимизировать выбранный критерий (длина, время, и т.д.).

2.2 Локальное планирование (динамическая среда)

Локальное планирование применяется, когда среда частично неизвестна или динамична: новые препятствия могут появляться во время движения робота. В этом случае необходим режим online, в котором сенсоры робота периодически сообщают об обнаруженных препятствиях, и планировщик корректирует маршрут в реальном времени, сохраняя приоритет безопасности и корректности траектории.

2.3 Гибридный подход

Практически часто используется гибрид: глобальный план предоставляет ориентировочную траекторию, а локальный планировщик выполняет обработку неожиданных помех и корректирует движения. Гибридная схема сочетает надёжность глобального поиска и адаптивность локального управления.

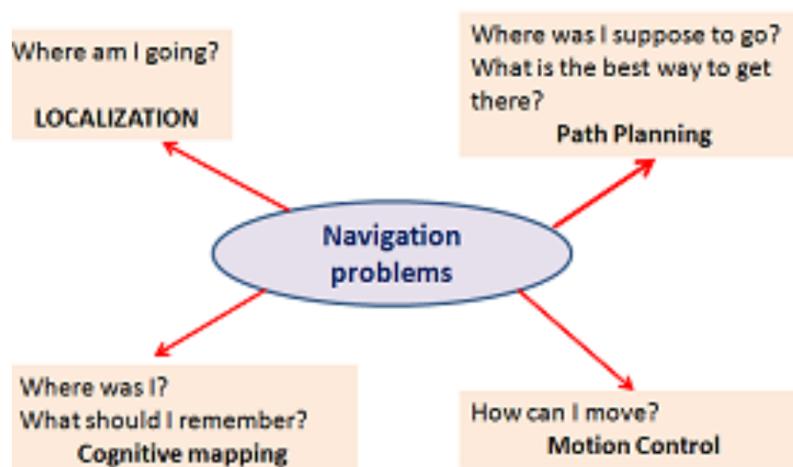


Рис. 1: Проблемы навигации робота

3 Представление окружающей среды и модели

3.1 Решётка (grid)

Один из простых способов моделирования — разбить пространство на равномерную решётку клеток. Каждая клетка помечается как свободная или занятая. Путь затем рассматривается как последовательность переходов между центрами свободных клеток. Выбор разрешённого набора направлений движения определяет структуру графа:

- 4-соседство: движения по горизонтали и вертикали;
- 8-соседство: добавление диагональных переходов;
- расширенные варианты: 16, 24, 30 направлений (увеличение степени свободы при выборе направления).

3.2 MAKLINK-модель свободного пространства

MAKLINK-подход заключается в представлении свободного пространства через набор свободных линий, соединяющих вершины окаймлённых препятствий и границы среды. Основные этапы формирования модели:

1. окаймление (расширение) границ препятствий для обеспечения безопасного буфера;
2. построение всех возможных линий, соединяющих углы окаймлённых препятствий;
3. удаление пересекающих препятствия или избыточных линий для формирования выпуклых областей свободного пространства;
4. взятие средних точек оставшихся свободных линий как вершин (узлов) MAKLINK-графа.

Далее строится граф из средних точек; путём по этому графу (например, алгоритмом Дейкстры) находится субоптимальное безопасное решение, которое может служить начальным приближением для дальнейшей оптимизации.

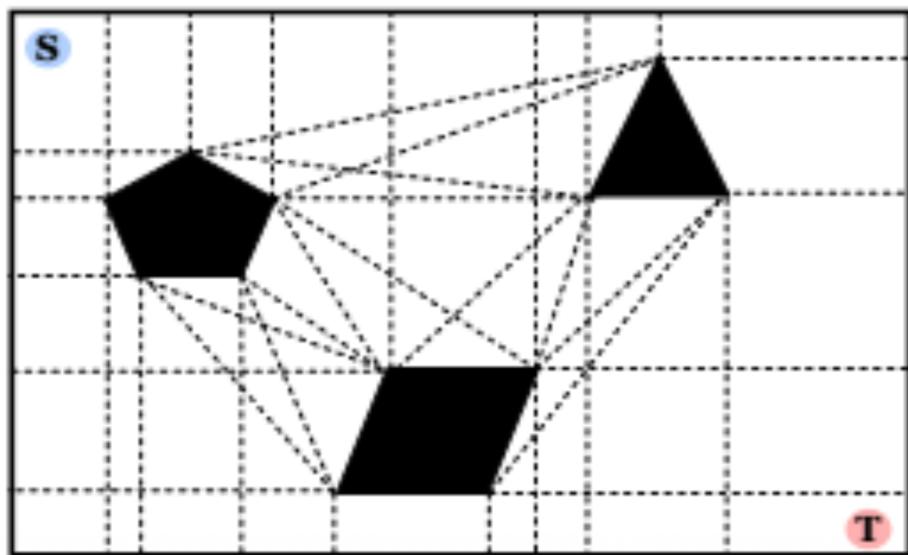


Рис. 2: MAKLINK граф: без избыточных свободных линий

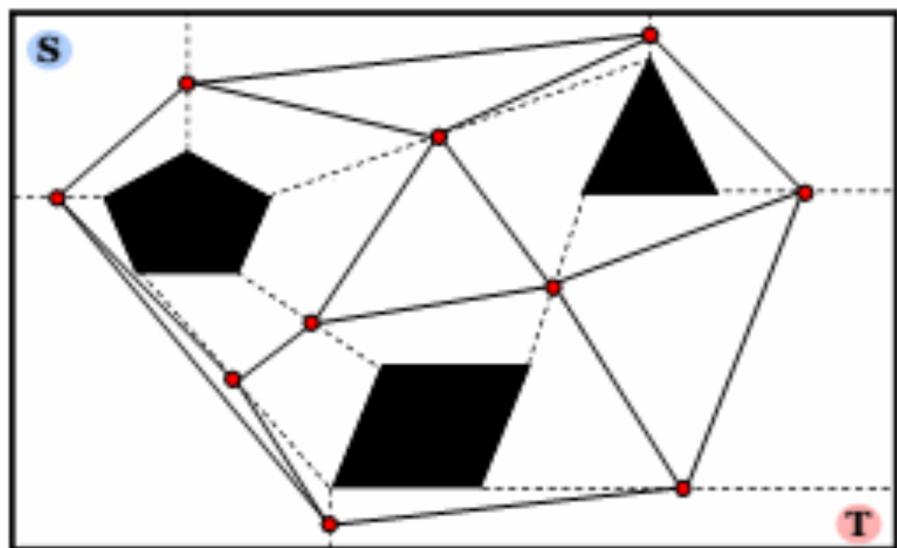


Рис. 3: MAKLINK граф: соединенные средние точки

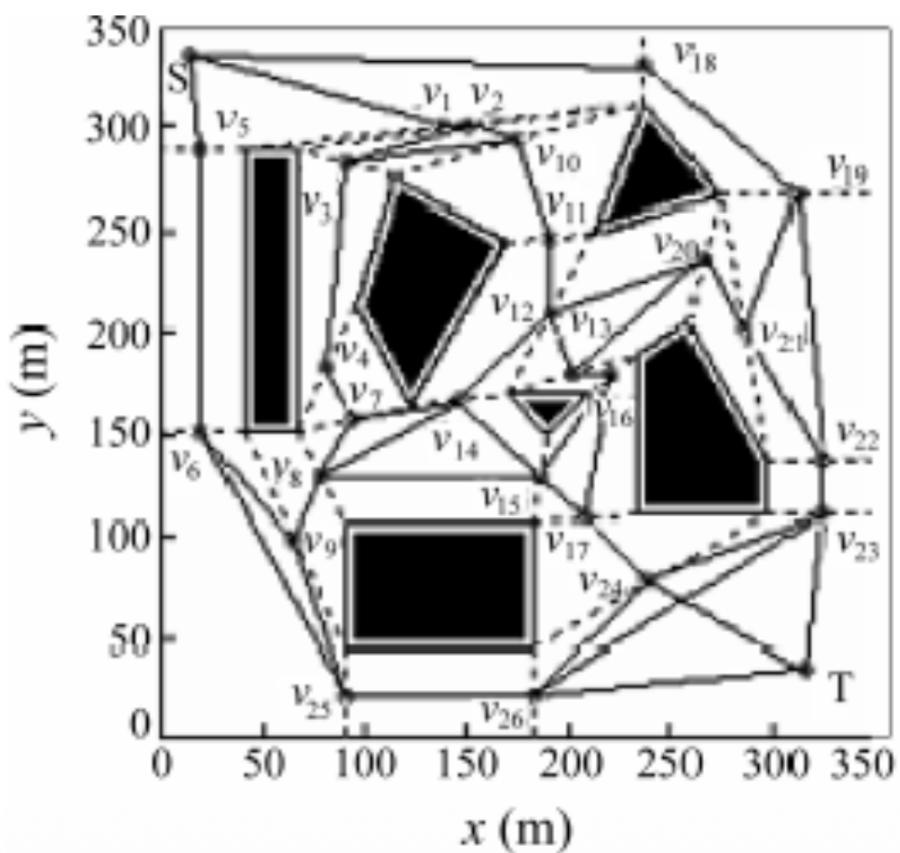


Рис. 4: MAKLINK граф: пути

3.3 Преимущества и недостатки моделей

- Grid: простота реализации, гибкость, но возможна большая размерность и чувствительность к разрешению (размеру клетки).
- MAKLINK: более компактное представление свободного пространства, гарантии отсутствия

пересечений с препятствиями, подходящее для построения субоптимальных маршрутов, пригодных для последующей оптимизации.

4 Муравьиный алгоритм: базовые принципы и интуиция

Муравьиный алгоритм моделирует коллективное поведение агентов, способных оставлять и считывать след в окружающей среде. В контексте планирования пути агент — это виртуальный муравей, проходящий по дискретизированному графу. Каждый агент независимо генерирует путь от S к T , причем вероятность выбора очередного шага зависит от накопленной информации (феромона) и локальной эвристики (видимости перехода). Коллективное поведение реализуется через два ключевых механизма:

1. **испарение феромона** — снижение значений τ во времени, что предотвращает преждевременную конвергенцию и способствует исследованию;
2. **подкрепление лучших решений** — увеличение τ на ребрах/узлах, входящих в хорошие по качеству пути.

В результате алгоритм способен балансировать между исследованием новых областей пространства и эксплуатацией накопленного знания о хороших траекториях.

5 Ant Colony System (ACS): формализация

5.1 Правило выбора перехода

Пусть муравей в текущей вершине i рассматривает множество соседних вершин \mathcal{N}_i , исключая временно запрещённые вершины из “BlackList”. Для каждой допустимой вершины j задаются:

- феромоновая метка $\tau_{ij}(t)$;
- эвристическая характеристика $\eta_{ij}(t)$ (например, обратная длине перехода или функция, учитывающая изменение направления);
- параметры α и β , задающие относительную важность τ и η .

Тогда вероятность перехода:

$$P_{ij}(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{k \in \mathcal{N}_i \setminus \text{BlackList}} [\tau_{ik}(t)]^\alpha [\eta_{ik}(t)]^\beta}.$$

Практическая реализация иногда включает жадный выбор с вероятностью q_0 (если случайное $q < q_0$, используется наиболее вероятный переход; иначе — случайный по распределению P_{ij}).

5.2 Локальное обновление феромона

После того как муравей проходит ребро (i, j) , выполняется локальная коррекция:

$$\tau_{ij} \leftarrow (1 - \rho) \tau_{ij} + \rho \tau_0,$$

где τ_0 — базовый уровень феромона, ρ — локальный коэффициент. Локальное уменьшение феромона способствует расширению поиска, уменьшая склонность других агентов к немедленному следованию тем же связям.

5.3 Глобальное обновление феромона

По завершении всех турок текущей итерации определяется лучший путь длины L^+ . Для ребёр (i, j) принадлежащих этому пути производится глобальное обновление:

$$\tau_{ij}(t+1) = (1 - \rho) \tau_{ij}(t) + \frac{Q}{L^+},$$

где Q — константа, масштабирующая вклад феромона от лучшего пути. Таким образом, лучшие решения получают пропорциональное вознаграждение, обратное длине пути.

5.4 Критерии и останов

Алгоритм повторяет итерации до выполнения одного из критериев:

- достижение предельного числа итераций NC ;
- все муравьи построили один и тот же путь (сходимость);
- достижение требуемого уровня качества решения.

6 Параметры алгоритма и практическая настройка

6.1 Основные параметры

α степень влияния феромона в правиле перехода;

β степень влияния эвристики η ;

ρ коэффициент испарения/локальной коррекции феромона;

Q количество феромона, вносимого лучшим муравьём;

τ_0 начальная концентрация феромона;

m число муравьёв (мощность популяции);

NC максимальное число итераций.

6.2 Влияние параметров на поведение

- Повышение α увеличивает влияние накопленного феромона и ускоряет эксплуатацию уже найденных путей. Чрезмерно большое α может привести к преждевременной сходимости.
- Повышение β увеличивает роль эвристики (локальная видимость), что стимулирует локально оптимальные решения.
- Большое ρ ускоряет испарение феромона; при больших значениях поддерживается более сильное исследование.
- Размер популяции m влияет на разнообразие решений: малое m может ограничить поиск, слишком большое — увеличить вычислительные затраты.

6.3 Экспериментальная настройка (пример)

Практический выбор параметров обычно производится эмпирически. В одном из примеров эффективными оказались значения (порядочный пример): $NC = 200$, $m = 10$, $\tau_0 = 0.2$, $\rho = 0.85$, $\beta = 2$, $q_0 = 0.1$, $\alpha = 1.0$. (Значения приведены как ориентир и зависят от конкретной постановки).

7 Применение к решению на дискретных MAKLINK-линиях

7.1 Дискретная параметризация

Построение пути через средние точки MAKLINK-линий сводит геометрическую задачу к параметрической оптимизации. На каждой линии i средняя точка может быть смещена вдоль перпендикулярного направления в одном из дискретных шагов $\{h_{i,0}, \dots, h_{i,r_i}\}$. Тогда путь описывается вектором дискретных смещений $\mathbf{h} = (h_1, \dots, h_d)$. Задача оптимизации — найти \mathbf{h}^* минимизирующий длину пути $L(\mathbf{h})$.

7.2 MA как оптимизатор в пространстве смещений

Варианты смещений трактуются как “узлы” для муравьёв: муравей в позиции i выбирает вариант $u \in A_i$ с вероятностью

$$P_{iu}(t) = \frac{[\tau_{iu}(t)]^\alpha [\eta_{iu}(t)]^\beta}{\sum_{v \in A_i} [\tau_{iv}(t)]^\alpha [\eta_{iv}(t)]^\beta}.$$

Здесь η_{iu} может задаваться через отклонение координат по y (или другой оси), например:

$$\eta_{iu} = \frac{1}{1 + |y_{iu} - y_{iu}^*|},$$

где y_{iu}^* — эталонное значение (например, по предыдущей итерации). Такой выбор η стремится к смещениям, приближающимся к ранее найденным хорошим решениям.

7.3 Алгоритмическая схема

В данной постановке один тур муравьиного алгоритма — это продвижение всех m муравьёв от S по всем MAKLINK-линиям до T , с записью выбранных смещений в массивы Path_k и локальными обновлениями τ . Затем вычисляются длины полученных путей L_k , выбирается лучший L^+ и выполняется глобальное обновление феромона. Далее возможна проверка на сходимость и повтор итераций.

8 Псевдокод детального алгоритма (шаги)

Ниже приведена формальная последовательность шагов алгоритма (развернутая версия):

1. Построить MAKLINK-граф и найти субоптимальный путь $S \rightarrow P_1 \rightarrow \dots \rightarrow P_d \rightarrow T$ методом Дейкстры.
2. Задать параметры: $m, \tau_0, \rho, \alpha, \beta, q_0, NC$.
3. Инициализировать $t \leftarrow 1$; для всех узлов установите $\tau_{iu} = \tau_0$.
4. Поместить m муравьёв в начальную точку S .
5. Для $i = 1$ до d : для каждого муравья $k = 1 \dots m$:
 - a) выбрать узел $u \in A_i$ по правилу $P_{iu}(t)$ (включая возможный жадный выбор при $q < q_0$);
 - b) запомнить выбор в $\text{Path}_k[i]$;
 - c) выполнить локальную коррекцию $\tau_{iu} \leftarrow (1 - \rho)\tau_{iu} + \rho\tau_0$.
6. После прохождения всех линий получить у каждого муравья геометрию пути P_1^k, \dots, P_d^k и вычислить длину L_k .
7. Определить $L^+ = \min_k L_k$ и соответствующий путь T_t . Обновить глобальную память: для всех рёбер/узлов, входящих в T_t , выполнить $\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \frac{Q}{L^+}$.
8. Проверить условие остановки: если $t \geq NC$ или все муравьи построили одинаковый путь, либо иное условие — завершить; иначе $t \leftarrow t + 1$ и вернуться к шагу 5.

9 Эксперименты и численные примеры

Эксперимент был взят из методического материала^[1]. Тестирование проводилось на наборах сред разной сложности: простая, относительно сложная, сложная. Проводились эксперименты с различными начальными и конечными точками, а также с разной конфигурацией препятствий.

Примеры результатов

Пример 1 (сложная среда): Начальная точка $S = (5, 50)$, целевая $T = (400, 450)$. Полученные длины маршрутов:

$$L_{\text{MA}} = 913.82 \text{ см.}$$

Для сравнения, генетический алгоритм на тех же данных дал:

$$L_{\text{GA}} = 921.03 \text{ см.}$$

Таким образом, МА показал лучшее качество пути в представленном тесте.

Гибридный пример (MAKLINK + Дейкстра + МА):

- Субоптимальный путь (Дейкстра): $L = 507.692 \text{ м.}$
- После оптимизации МА: $L = 440.233 \text{ м.}$
- При увеличении точности дискретизации (более мелкий шаг h_i): $L = 439.372 \text{ м.}$

Данные демонстрируют значительное улучшение качества пути после применения МА к субоптимальному начальному решению.

Сходимость и сравнение по времени В приведённых экспериментах:

- МА достигает оптимума примерно на 38-й итерации;
- ГА даёт оптимум примерно на 240-й итерации;
- МА требует меньшее время на построение пути при равной мощности популяции.

	ACS Algorithm	Real-coded GA
Average CPU time per iteration (sec.)	0.00059	0.00067
Average number of iterations needed for convergence	175	912
Average CPU time needed for obtaining optimal solution (sec.)	0.1033	0.6110

Рис. 5: Результаты проведения анализа (с методического материала)

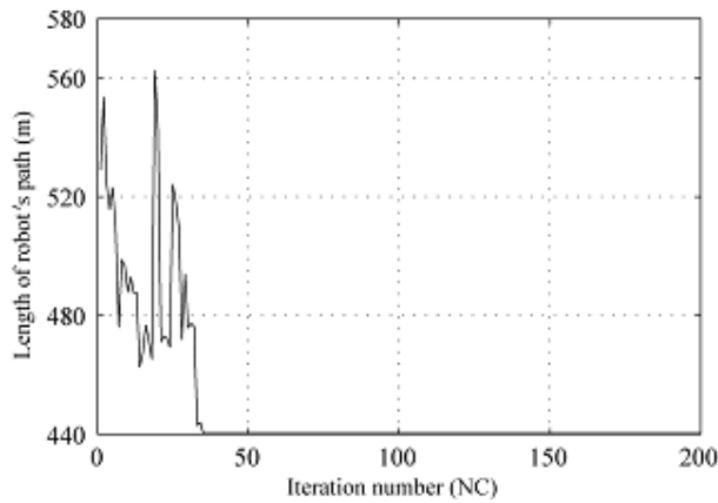


Рис. 6: Муравьиный алгоритм - дает оптимум на 38 итерации

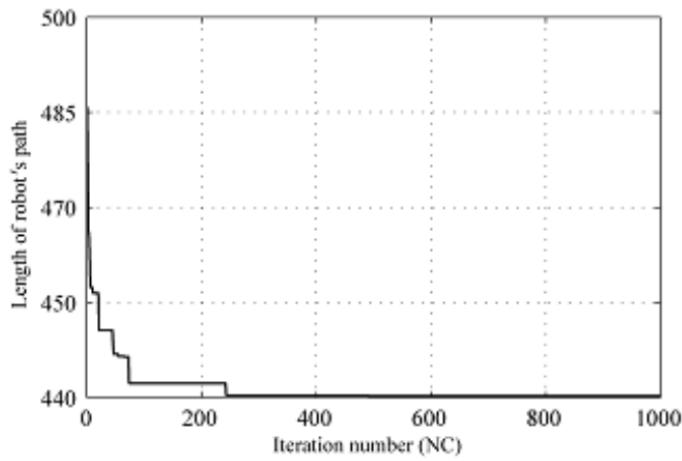


Рис. 7: Генетический алгоритм - дает оптимум на 240 итерации

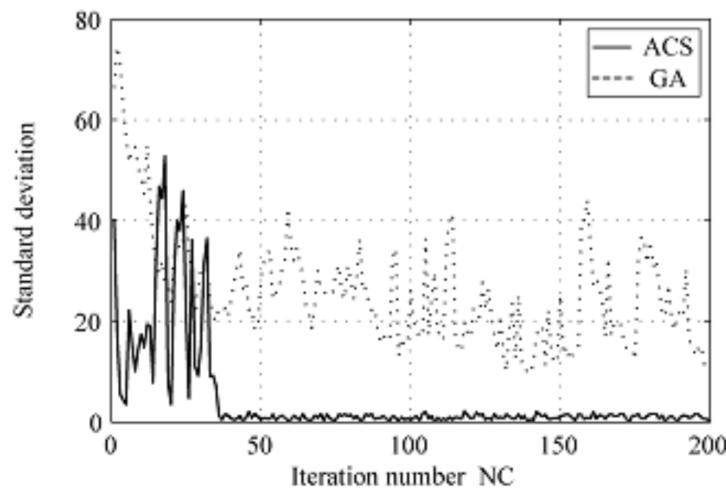


Рис. 8: Сравнения отклонений

10 Анализ влияния числа направлений движения

10.1 Стандартные случаи

При 4 направлениях движение ограничено осями, при 8 направлениях добавляются диагонали, что обычно даёт лучшие пути и большую гибкость выбора.

10.2 Расширенные варианты и их эффект

Увеличение числа допустимых направлений до 16, 24 или 30 позволяет работе выбирать более плавные и короткие траектории, но:

- увеличивается количество возможных переходов из узла, что повышает вычислительную сложность расчёта вероятностей;
- для МА это означает расширение множества альтернатив, что может увеличивать время на сходимость, но при правильно настроенных параметрах повышает шансы найти более качественное решение.

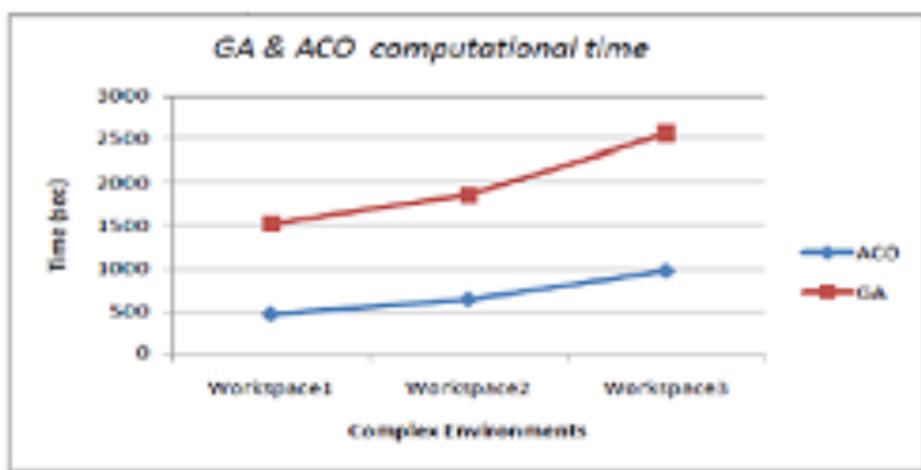


Рис. 9: Сравнение по времени

11 Гибридные подходы и преимущества интеграции методов

11.1 Мотивы для гибрида

Комбинация детерминированных методов поиска (например, Дейкстры) с стохастическими оптимизаторами (МА) даёт:

- гарантированный начальный путь без пересечений с препятствиями;
- возможность улучшить начальное решение по качеству (длина, гладкость) за счёт стохастического поиска в локальном окне;
- снижение риска попадания в локальные минимумы благодаря стохастическому исследованию.

11.2 Описание примерной схемы

1. Построить MAKLINK-граф и найти субоптимальный путь алгоритмом Дейкстры.
2. Дискретизовать окрестности средних точек линий (возможные смещения).
3. Запустить МА для оптимизации вектора смещений.
4. По завершении получить оптимизированный путь и при необходимости выполнить сглаживание.

11.3 Числовые примеры эффективности

Ранее приведённые численные данные (спад длины с 507.692 м до 440.233 м) показывают, что гибридный подход может давать значительное улучшение качества решения.

11.4 Пример свободного пространства

Рассматривается дискретная сетка, которая представляет окружение робота, где Черным отмечены препятствия, белым свободные клетки, желтым отмечены вершины, которые может пересекать робот.

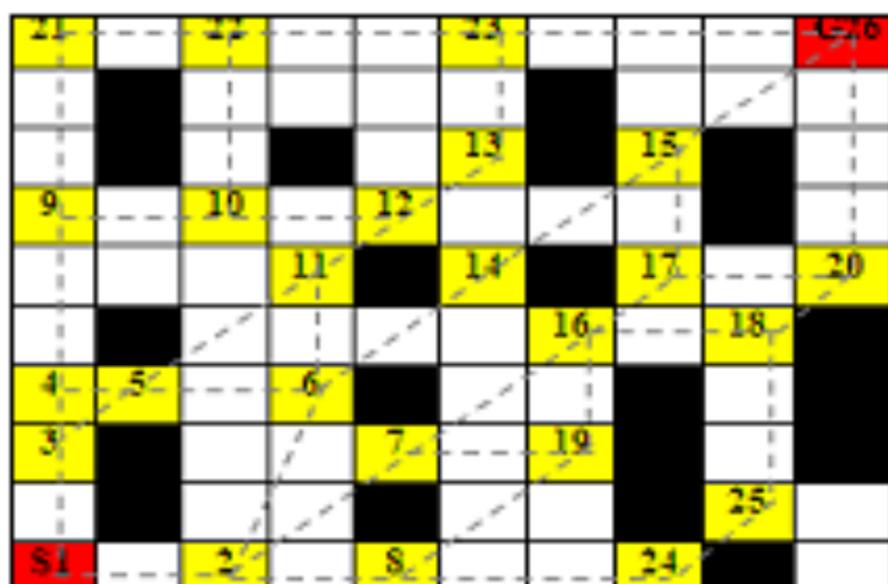


Рис. 10: Дискретная сетка

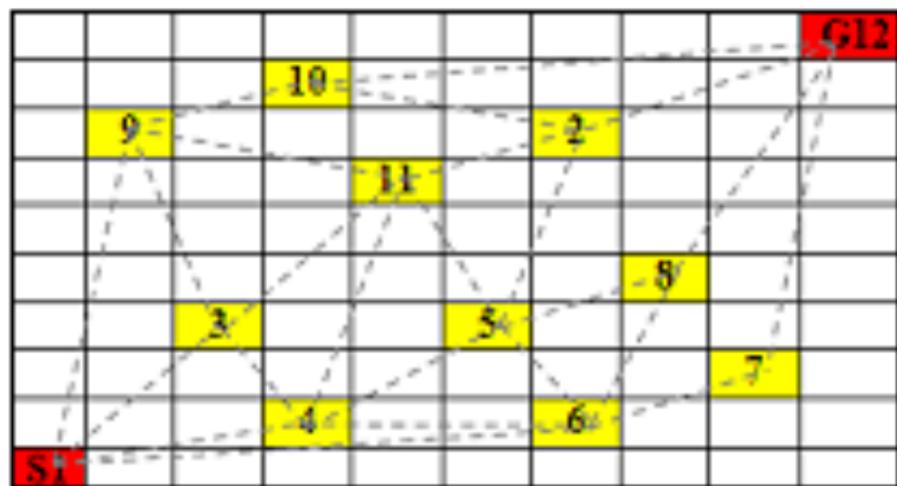


Рис. 11: Простая карта свободного пространства

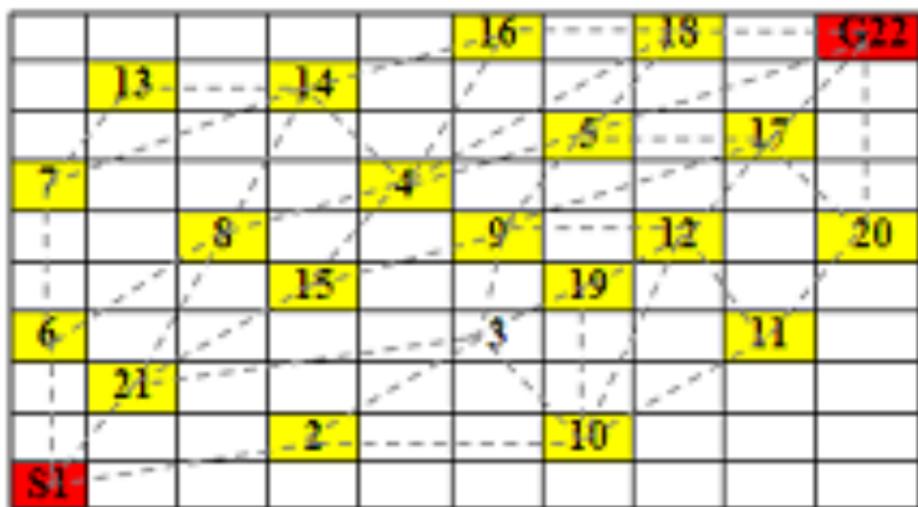


Рис. 12: Средняя карта свободного пространства

	51	57	52	53	54	55		56	G63
44	58	48	46		47	60	48	49	50
38	39	40		41	59	42		43	
31		32	33		34		35	36	37
	27	28		29		30		25	
20		21	22		23		24		26
	14			13		16	17		19
10		11	12		15			13	
	61	62	3	4		6	7		9
91		2			5			8	

Рис. 13: Сложная карта свободного пространства

Заключение

- муравьиный алгоритм (ACS) является эффективным стохастическим методом для задачи планирования пути, демонстрирующим хорошую сходимость и качество решений на тестовых средах;
- гибридные схемы, комбинирующие детерминированные алгоритмы (Дейкстра, MAKLINK) и MA, дают значительное улучшение начальных субоптимальных решений и обеспечивают безопасность (отсутствие столкновений);
- корректная настройка параметров ($\alpha, \beta, \rho, Q, m, NC$) и выбор представления пространства (grid/MAKLINK) критично влияют на производительность и качество;
- практические реализации должны учитывать динамику среды, интеграцию с сенсорными данными и требования к сглаживанию траекторий.

Рекомендуется использовать гибридный подход в задачах, где требуется сочетание гарантии отсутствия столкновений и высокого качества пути: построить субоптимальный безопасный маршрут детерминированно и затем улучшить его стохастическим оптимизатором (MA), при этом применяя адаптивную дискретизацию и параллельные вычисления для ускорения работы.

Список литературы

- [1] Скобцов Ю. А. ГА: Планирование пути МА. СПбПУ. 1-59 с.
- [2] DORIGO, M., BIRATTARI, M., STÜTZLE, T. Ant Colony Optimization. *IEEE Computational Intelligence Magazine*. 2006. Vol. 1, no. 4. pp. 28–39.
- [3] МАЛЬГИНА, Майя. Графы и муравьиные алгоритмы. *Skillbox Media*. Доступно по ссылке: <https://skillbox.ru/media>.
- [4] ЦУРИКОВ, А. Построение маршрутов с использованием муравьиных алгоритмов. Доступно по ссылке: <https://example.com>.
- [5] Amazon Robotics. Применение муравьиных алгоритмов в логистике. [Электронный ресурс]. Доступно по ссылке: <https://example.com/amazon>.
- [6] ВАСИЛЬЕВ, С. В., ПЕТРОВ, А. А. Моделирование свободного пространства с использованием теории MAKLINK. *Известия вузов. Технические науки*. 2015. №3. С. 56–64.
- [7] ACS Ant Colony System: A new heuristic optimization approach for solving TSP. *Elsevier Science*. 1997. Vol. 6, no. 4. pp. 321–333.
- [8] FedEx Logistics. Оптимизация маршрутов на основе муравьиных алгоритмов. [Электронный ресурс]. Доступно по ссылке: <https://example.com/fedex>.
- [9] BONABEAU, E., DORIGO, M., THERAULAZ, G. Swarm Intelligence: From Natural to Artificial Systems. *Oxford University Press*. 1999.