

Методы эвристического поиска на графах особой структуры и их применение в задачах планирования траекторий

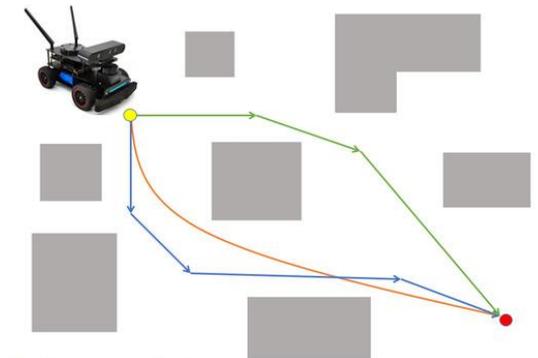
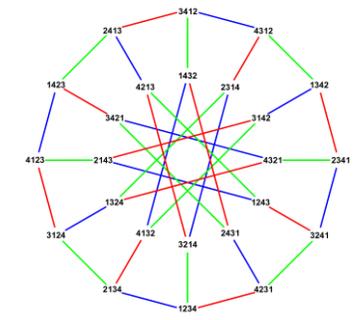
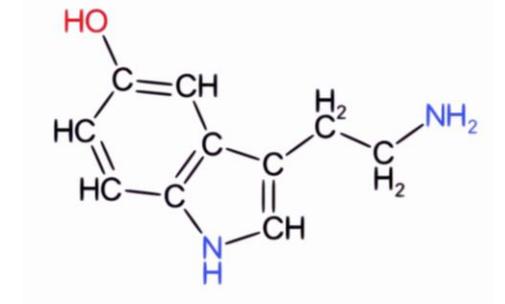
Яковлев Константин Сергеевич

к.ф.-м.н., в.н.с. Отдела 71 ФИЦ ИУ РАН

Диссертация на соискание степени доктора физико-математических наук по специальности 1.2.1 – Искусственный интеллект и машинное обучение

Актуальность

- **Химия** – граф химических реакций
 - Вершины – молекулы, ребра – реакции. Синтез молекул с заданными свойствами = задача поиска пути.
- **Математика** – граф Кэли
 - Вершины – элементы группы перестановок, ребра – образующие. Получение единичного элемента = задача поиска пути.
- **Робототехника** – граф окружающего робота пространства
 - Вершины – локация, ребра – перемещения между локациями. Планирование траектории = задача поиска пути.

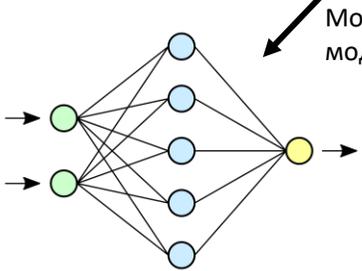


Планирование траектории



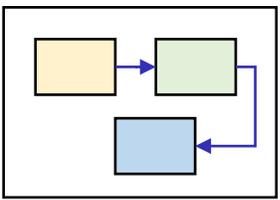
- **Цель – повышение степени автономности мобильного агента (робота)**
 - за счет применения моделей, методов, алгоритмов искусственного интеллекта в системах управления,
 - в том числе методов эвристического поиска на графах особой структуры для решения задач (автоматического) планирования траектории

Интеллектуальная система управления

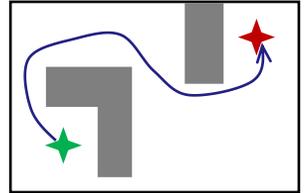


Монолитная модель МО

Модульная система



Модуль навигации

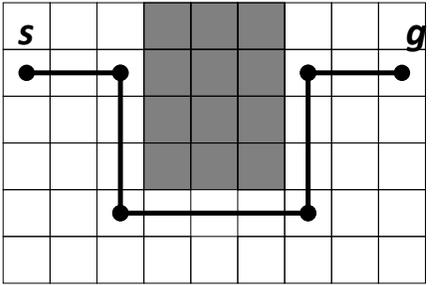


Последовательное принятие решений

- Правила
- Обучение с подкреплением
- Имитационное обучение
- И др.

Планирование пути + Следование вдоль пути

- Теория автоматического управления
- Методы оптимизации
- И др.



Практико-ориентированные ограничения

- **Динамика окружающей среды**
- **Кинематические ограничения**
- **Выч.эффективность**
- **Много-агентная постановка**

Построение	Пост-обработка
<ul style="list-style-type: none"> • Поиск пути на графах • Случайное семплирование • ... 	<ul style="list-style-type: none"> • Методы оптимизации • Методы сглаживания • ...

Графы регулярной декомпозиции + Эвристический поиск

- Поиск пути на динамических ГРД
- Поиск пути с геометрическими ограничениями
- Адаптивные обучаемые эвристические функции
- Поиск совокупности неконфликтных путей на ГРД



Новые методы, алгоритмы, модели

Цель и задачи исследования

Основной **целью** работы является получение новых теоретических и экспериментальных результатов в области **методов поиска пути** (совокупности путей) **на графах особой структуры** вложенных в метрическое пространство, и **применимых для** решения задач автоматического **планирования траектории** (совокупности неконфликтных траекторий).

Задачи

- Разработка новых методов поиска пути на динамических графах регулярной декомпозиции, применимых для планирования траектории в среде со статическими и динамическими препятствиями (траектории движения которых известны); проведение теоретических исследований предложенных методов; разработка алгоритмов их реализующих; проведение экспериментальных исследований разработанных алгоритмов.
- Разработка новых методов поиска совокупности неконфликтных путей на графах регулярной декомпозиции, применимых для решений задачи централизованного много-агентного планирования траекторий для группы агентов; проведение теоретических исследований предложенных методов; разработка алгоритмов их реализующих; проведение экспериментальных исследований разработанных алгоритмов, в том числе на реальных робототехнических платформах.
- Разработка новых методов поиска пути на графах регулярной декомпозиции, косвенно учитывающих кинематические ограничения мобильного агента; проведение теоретических исследований предложенных методов; разработка алгоритмов их реализующих; проведение экспериментальных исследований разработанных алгоритмов.
- Разработка новых методов автоматического конструирования эвристических функций для решения задач поиска пути на графе регулярной декомпозиции с помощью методов машинного обучения, в частности -- глубокого обучения с применением современных нейросетевых архитектур, а также способов интеграции этих функций в алгоритмы эвристического поиска; программная реализация предлагаемых подходов и проведение экспериментальных исследований.

Основной объект исследования

Граф, вложенный в метрическое пространство

$$(\mathcal{M}, \mathcal{G}, coord)$$

\mathcal{M} – метрическое пространство ($\mathcal{M} = \mathbb{R}^2$)

$\mathcal{G} = (V, E, w)$ – неориентированный взвешенный граф

- V – множество вершин
- E – множество пар вершин (ребер). $E = \{e = (u, v) | u \in V, v \in V, u \neq v\}$
- $w: E \rightarrow \mathbb{R}^+$ – (неотрицательный) вес ребра. $w(e) > 0$

$$coord: V \rightarrow \mathcal{M}$$

$$w(e) = \|coord(u), coord(v)\|$$

Граф регулярной декомпозиции (ГРД)

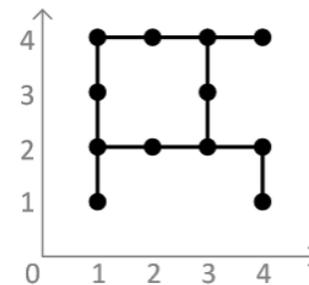
ГРД – это граф, вложенный в \mathbb{R}^2 и содержащий вершины, которым соответствуют точки на плоскости, образующие регулярную сетку.

$\forall v, u \in V$:

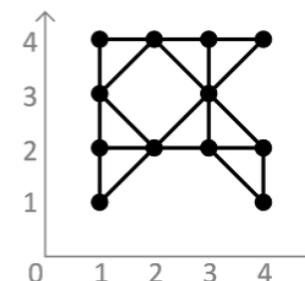
$$coord_x(u) = coord_x(v) + k \cdot \Delta,$$

$$coord_y(u) = coord_y(v) + k \cdot \Delta,$$

где $k \in \mathbb{N} \cup 0, \Delta = const > 0$.



а) 4-связный ГРД



б) 8-связный ГРД

Поиск пути на ГРД и планирование траектории

Кусочно-линейная траектория

$$\pi : [\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_{n-1}, \alpha_n] \rightarrow C,$$

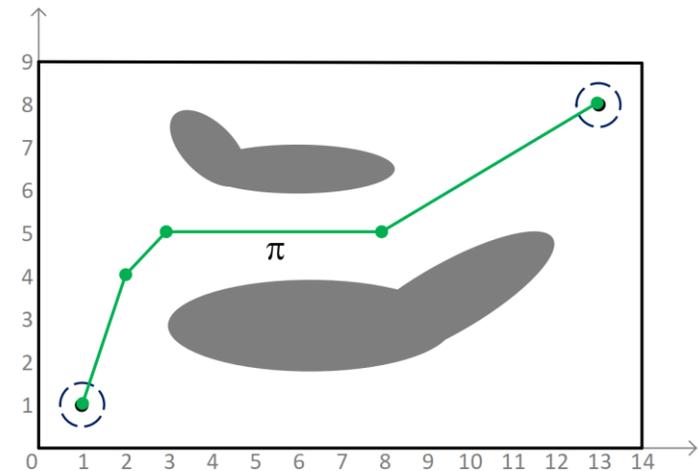
т.ч.:

$$0 = \alpha_0 < \alpha_1 < \dots < \alpha_{n-1} < \alpha_n = 1,$$

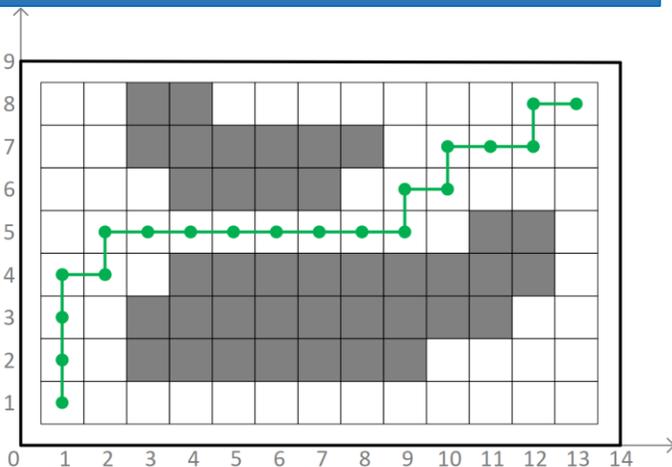
$$\pi(0) = \pi(\alpha_0) = \mathbf{x}_{start}, \pi(1) = \pi(\alpha_n) = \mathbf{x}_{goal},$$

$$\forall i \in \{0, 1, \dots, n-1\}, \forall \alpha \in [\alpha_i, \alpha_{i+1}] :$$

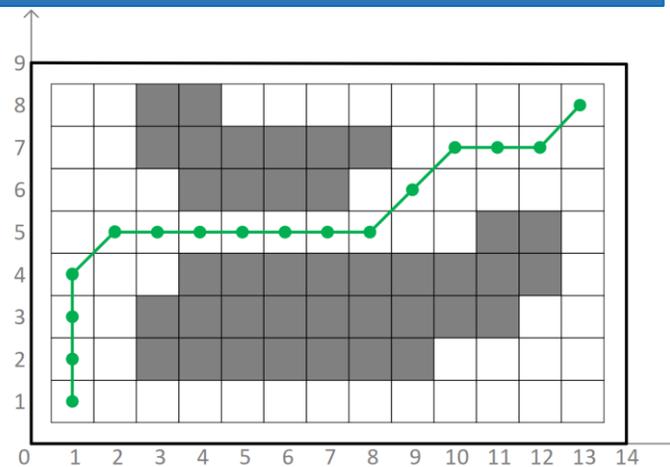
$$coord(\pi(\alpha)) = (1 - \alpha) \cdot coord(\pi(\alpha_i)) + \alpha \cdot coord(\pi(\alpha_{i+1})).$$



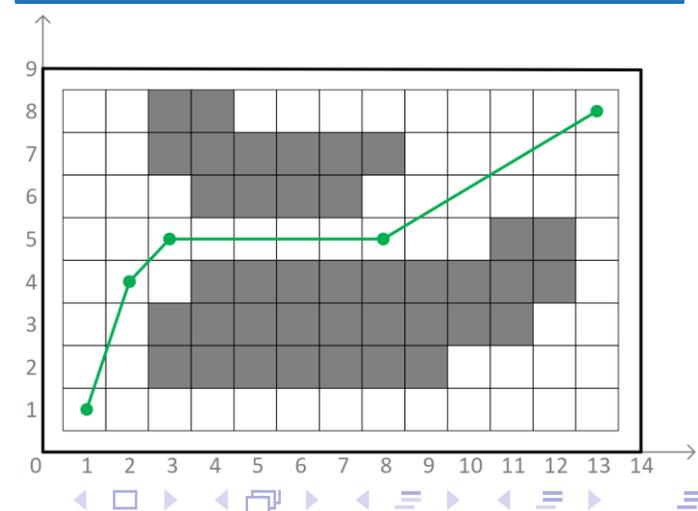
Путь на 4-связном ГРД



Путь на 8-связном ГРД

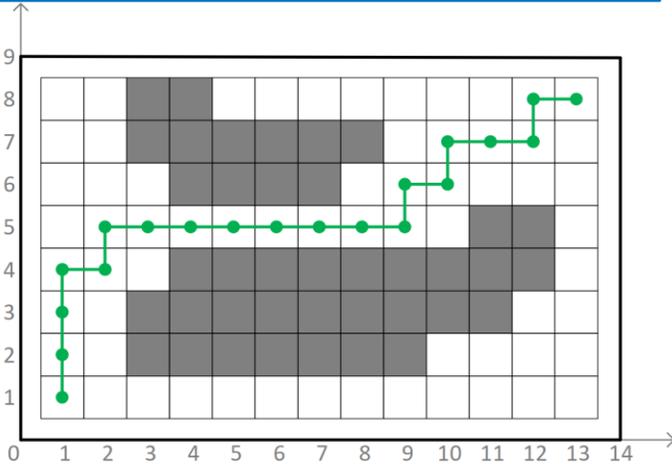


ПН-путь (any-angle)

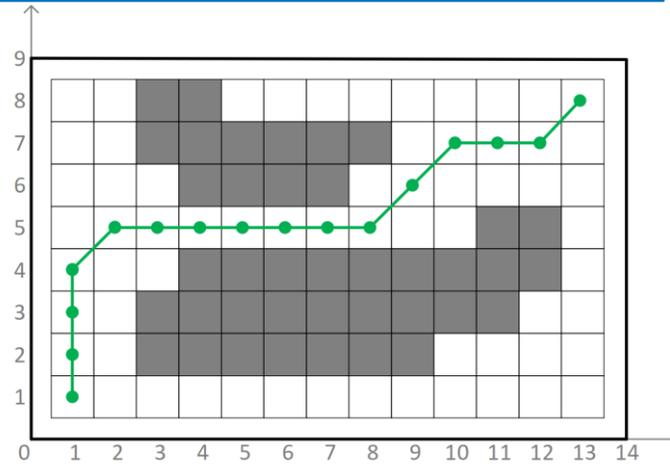


Поиск ПН-пути (пути) на ГРД

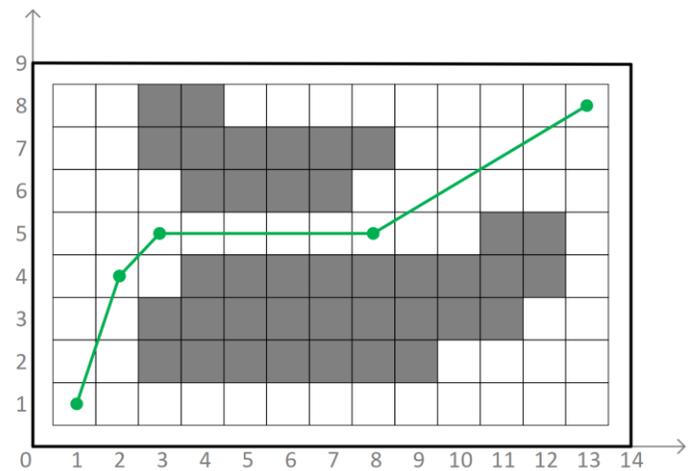
Путь на 4-связном ГРД



Путь на 8-связном ГРД



ПН-путь (any-angle)



Неинформированный поиск

- Алгоритм Белмана-Форда (1958, 1962)
- Алгоритм Дейкстры (1959)
- И др.

Эвристический поиск

- A* (1968)
- WA* (1970)
- R* (2008)
- JPS (2011)
- EPEA* (2015)
- JPSW (2024)
- И др.

$$los: V \times V \rightarrow \{\text{True}, \text{False}\}$$

- Вход алгоритма
- Выч.сложность вызова ($O(V)$)

Субоптимальные решения

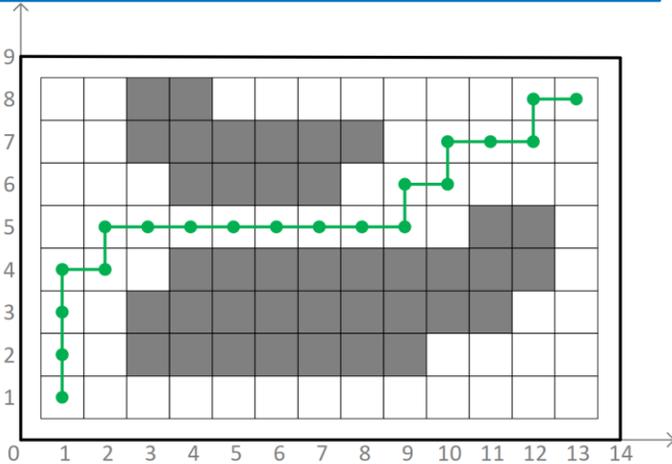
- Theta* (2007)

Оптимальные решения

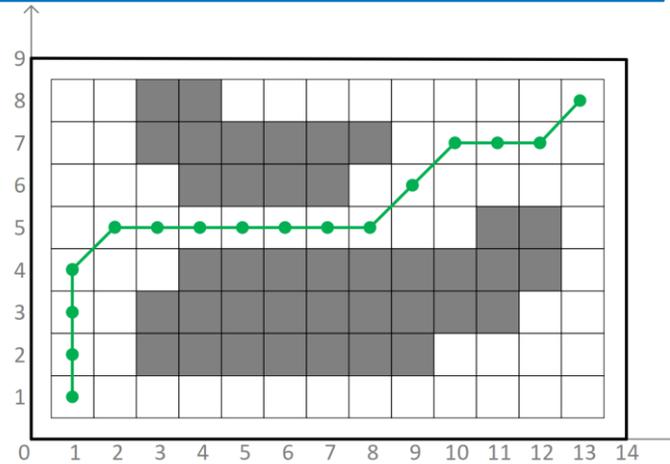
- ANYA (2015)

Поиск ПН-пути (пути) на ГРД

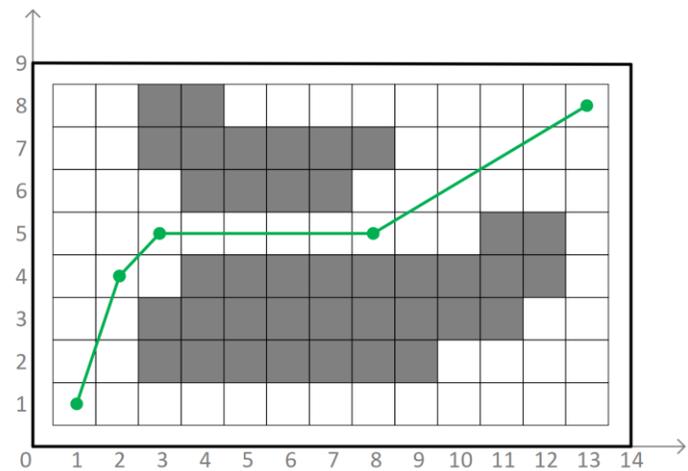
Путь на 4-связном ГРД



Путь на 8-связном ГРД



ПН-путь (any-angle)



Задачи не имеющие решения

- Поиск ПН-пути на динамическом (темпорально-аннотированном) ГРД
 - Планирование траектории в среде с движущимися препятствиями
- Поиск множества неконфликтных ПН-путей
 - Много-агентное планирование траекторий
- Поиск пути с геометрическими ограничениями
 - Планирование кинематически-согласованной траектории

Повышение эффективности алгоритмов (эвристического)

поиска

Новые методы и алгоритмы

- TO-AA-SIPP (опт.)
- AA-SIPP, WSIPP (суб., выч.эфф.)
- AA-CCBS (опт.)
- AA-SIPP(m) (опт.*)
- LIAN, D-LIAN (опт.*)
- WA*-CF (суб., выч.эфф.)
- FS-PPM (суб., выч.эфф.)

* в определенном классе задач

Поиск пути на динамическом ГРД

$$\text{AA - PFD} = (\mathcal{G}, T, \text{los}, T_{\text{safe}}(v), T_{\text{safe}}(u, v), v_{\text{start}}, v_{\text{goal}})$$

\mathcal{G} – ГРД $T = [0, \infty)$ – время $\text{los}: V \times V \rightarrow \{\text{True}, \text{False}\}$ – допустимость перехода

$v_{\text{start}}, v_{\text{goal}}$ – начальная и целевая вершины $T_{\text{safe}}(v), T_{\text{safe}}(u, v)$ – множества безопасных интервалов

Безопасные интервалы

$$T_{\text{safe}}(v) = (SI_1(v), SI_2(v), \dots, SI_n(v)),$$

$$SI_i(v) = [t_l^i, t_u^i],$$

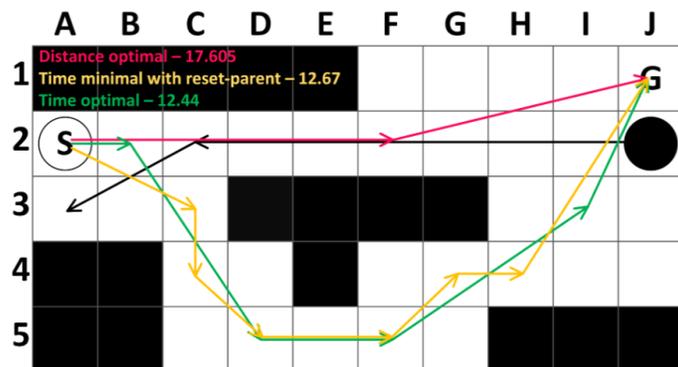
$$t_l^i, t_u^i \in T_{\text{safe}}(v),$$

$$t_l^i \leq t_u^i.$$

$$\forall SI = [t_l, t_u] \in T_{\text{safe}}(v), \forall \varepsilon > 0:$$

$$t_l - \varepsilon \notin T_{\text{safe}}(v),$$

$$t_u + \varepsilon \notin T_{\text{safe}}(v).$$



Путь на Д-ГРД

$$\pi(v_{\text{start}}, v_{\text{goal}}) = ((u_1, v_1, t_1), (u_2, v_2, t_2), \dots, (u_n, v_n, t_n)),$$

т.ч.:

$$u_1 = v_{\text{start}},$$

$$v_n = v_{\text{goal}},$$

$$\forall i = \overline{1, n-1}: v_i = u_{i+1},$$

$$\forall i = \overline{1, n}: t_i \in T_{\text{safe}}(u_i, v_i),$$

$$\forall i = \overline{1, n-1}: t_{i+1} \geq t_i + w(u_i, v_i)$$

$$\forall i = \overline{1, n-1}: [t_i + w(u_i, v_i), t_{i+1}] \subseteq T_{\text{safe}}(u_{i+1}),$$

$$t_1 \in SI_1(v_{\text{start}}),$$

$$t_n + w(u_n, v_n) \in SI_{\text{last}}(v_{\text{goal}})$$

$$\forall i = \overline{1, n}: \text{los}(u_i, v_i) = \text{true}$$

$$\text{cost}(\pi) = t_n + w(e_n)$$

Наивный подход

Безопасно-интервальное планирование

Поиск в пространстве состояний (вершина, интервал)

- Фокусировка с помощью эвристической функции

+

- Рассмотрение всех возможных потомков

Состояние

Интервал

$$n = (v, SI = [t_l, t_u])$$

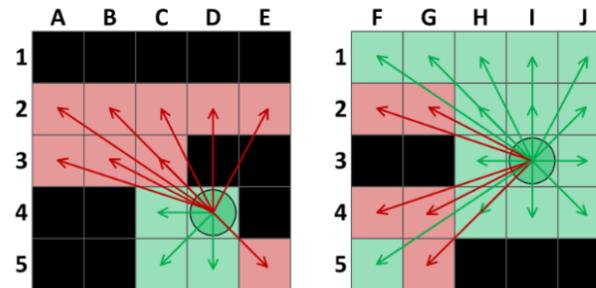
Вершина

$g(n), h(n), parent(n)$

$parent(n)$ – состояние-предок

$g(n)$ – время достижения

$h(n)$ – эвр. оценка стоимости пути до цели

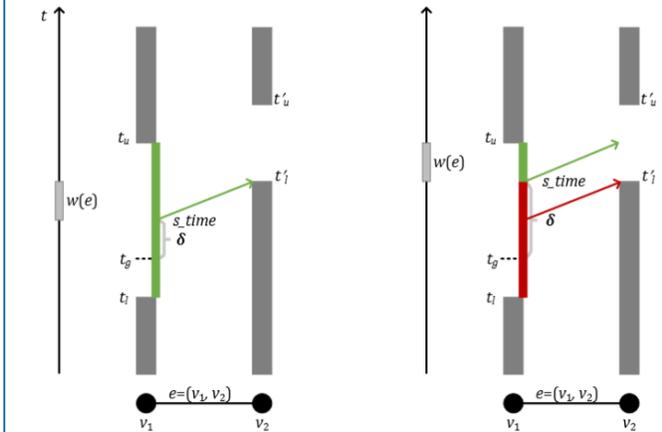


Высокая трудоемкость поиска

- генерация потомка
 - $[los(O(V)) + ValidateTransition]$
- Высокий коэффициент ветвления дерева поиска
 - $O(V)$



$$n_1 = (v_1, SI) \rightarrow n_2 = (v_2, SI')$$



ValidateTransition

Предлагаемый подход

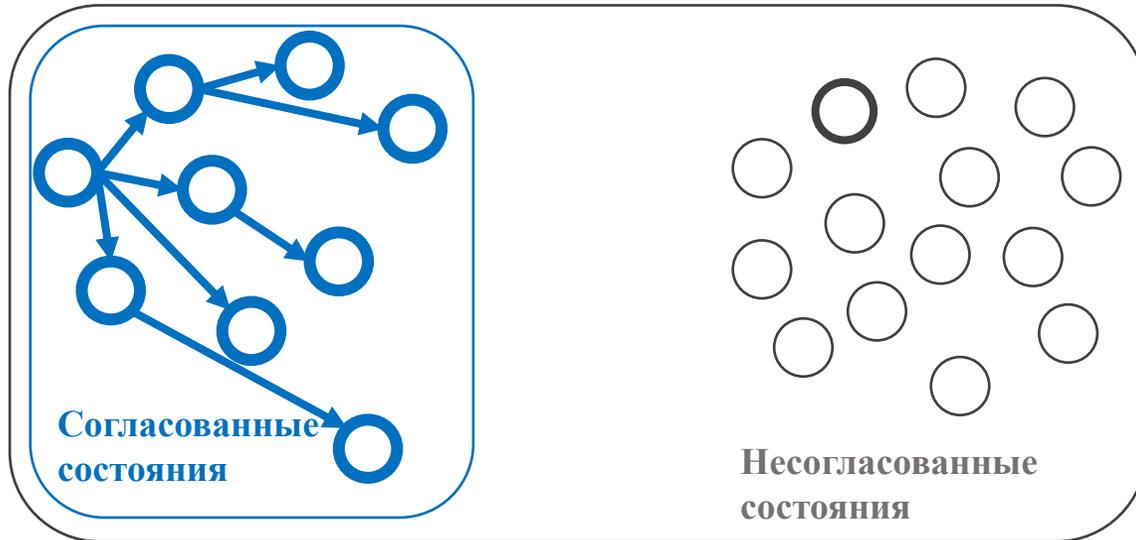
- Инверсивный поиск
 - Выбираем родителя, а не создаем потомка
- Ленивые вычисления
 - $los + ValidateTransition$

Предлагаемый алгоритм (TO-AA-SIPP)

Наилучший путь известен

$$g(n) = g^*(n)$$

Родительские указания валидны



Путь минимальной стоимости не известен
Родительские указатели не валидны

Цель

Согласованное (consistent) целевое состояние

За счет

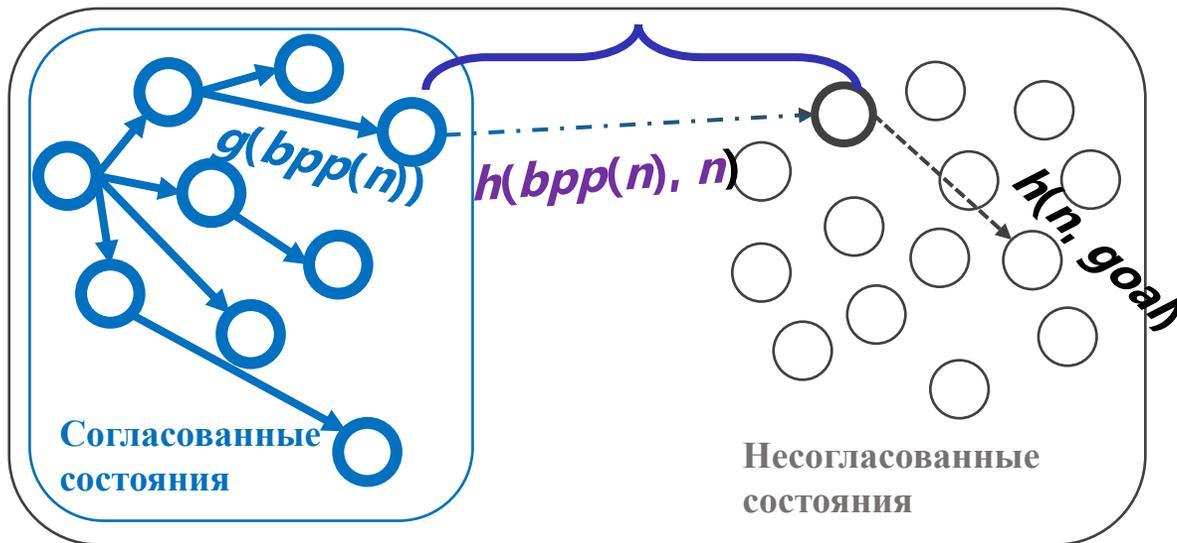
Систематического рассмотрения переходов из потенциальных родителей

1. Выбрать наиболее перспективное несогласованное состояние n и его лучшего потенциального родителя $bpp(n)$
2. Попытаться уменьшить $g(n)$ за счет рассмотрения перемещения $bpp(n) \rightarrow n$
3. Определить согласовано ли n . Если да, то перенести его в список потенциальных родителей

Предлагаемый алгоритм (TO-AA-SIPP)

1. Выбрать наиболее перспективное несогласованное состояние n и его лучшего потенциального родителя $bpp(n)$
2. Попытаться уменьшить $g(n)$ за счет рассмотрения перемещения $bpp(n) \rightarrow n$
3. Определить согласовано ли n . Если да, то перенести его в список потенциальных родителей

$g_{low}(n)$



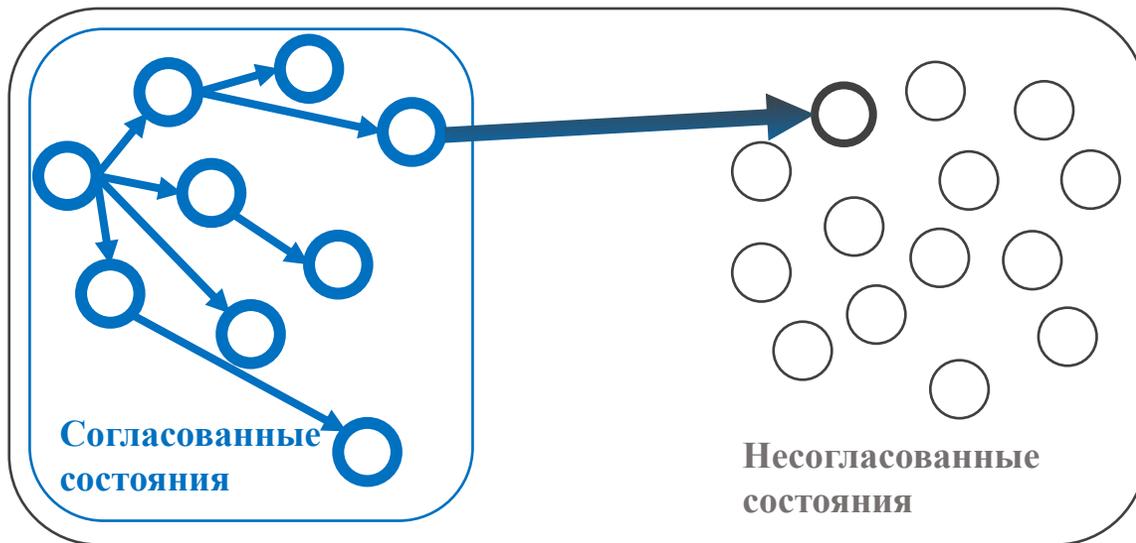
$$f(n) = g(bpp(n)) + h(bpp(n), n) + h(n, goal) \quad \} \quad g_{low}(n)$$

$g(n)$ – реальная стоимость
все переходы валидны

$g_{low}(n)$ – оптимистичная оценка
последний переход не валидирован

Предлагаемый алгоритм (TO-AA-SIPP)

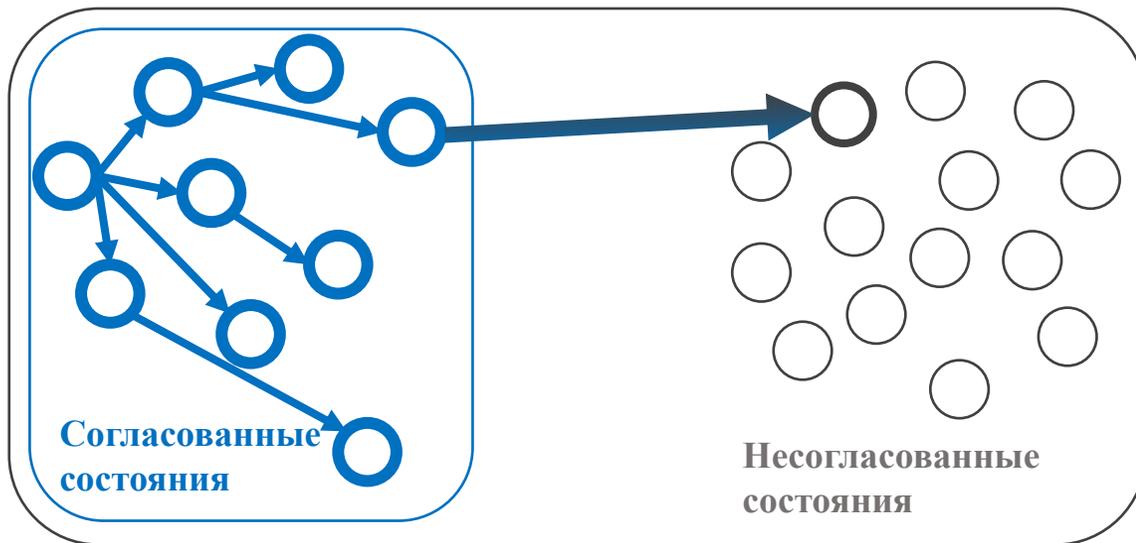
1. Выбрать наиболее перспективное несогласованное состояние n и его лучшего потенциального родителя $bpp(n)$
2. Попробовать уменьшить $g(n)$ за счет рассмотрения перемещения $bpp(n) \rightarrow n$
3. Определить согласовано ли n . Если да, то перенести его в список потенциальных родителей



- ValidateTransition
- g_{new} – время достижения n за счет перемещения $bpp(n) \rightarrow n$
- if $g_{new} < g(n)$
 $g(n) \leftarrow g_{new}$

Предлагаемый алгоритм (TO-AA-SIPP)

1. Выбрать наиболее перспективное несогласованное состояние n и его лучшего потенциального родителя $bpp(n)$
2. Попытаться уменьшить $g(n)$ за счет рассмотрения перемещения $bpp(n) \rightarrow n$
3. **Определить согласовано ли n . Если да, то перенести его в список потенциальных родителей**



Идея

- Нет состояний среди согласованных, позволяющих уменьшить $g(n)$
- Нет состояний среди несогласованных, позволяющих уменьшить $g(n)$

$$\Rightarrow g(n) = g^*(n)$$

TO-AA-SIPP: Псевдокод

Инициализация

Процедура TO-AA-SIPP-Init(\mathcal{G} , v_{start} , los , $\{T_{safe}(v)\}$, h):

```
1 OPEN ← ∅; CLOSED ← ∅
2 foreach v ∈ G do
3   foreach [tl, tu] ∈ Tsafe(v) do
4     n ← GenerateSearchNode(v, [tl, tu])
5     OPEN ← OPEN ∪ {n}
6 nstart ← FindNode(OPEN, vstart, SIfirst(vstart))
7 g(nstart) ← 0; glow(nstart) ← 0
8 OPEN ← OPEN \ {nstart}
9 CLOSED ← CLOSED ∪ {nstart}
10 foreach n = (v, [tl, tu]) ∈ OPEN do
11   if los(n.v, nstart.v) = true then
12     glow(n) ← h(nstart, n)
13     bpp(n) ← nstart
14     PARENTS(n) ← {nstart}
15   else
16     glow(n) ← ∞; bpp(n) ← null
17     PARENTS(n) ← ∅
18   f(n) ← glow(n) + h(n)
19   g(n) ← ∞
20   parent(n) ← null
21 return OPEN, CLOSED
```

Основной цикл

Процедура TO-AA-SIPP-Main:

```
1 while minn∈OPEN f(n) < ∞ do
2   n ← arg minn∈OPEN f(n); OPEN ← OPEN \ {n}
3   if bpp(n) ∈ PARENTS(n) then
4     PARENTS(n) ← PARENTS(n) \ {bpp(n)}
5   gnew ← ValidateTransition(n, bpp(n))
6   if gnew < g(n) then
7     g(n) ← gnew; parent(n) ← bpp(n)
8   if SetBestPotentialParent(n) then
9     OPEN ← OPEN ∪ {n}
10    continue
11   if g(n) + h(n) ≤ minn∈OPEN f(n) then
12     CLOSED ← CLOSED ∪ {n}
13     if n.v = vgoal and g(n) ∈ SIlast(vgoal) then
14       return ReconstructPath(n)
15     foreach n' ∈ OPEN do
16       if los(n', n) = true then
17         PARENTS(n') ← PARENTS(n') ∪ {n}
18         if g(n) + h(n, n') < glow(n') then
19           glow(n') ← g(n) + h(n, n')
20           bpp(n') ← n
21           f(n') ← glow(n') + h(n')
22           Update n' in OPEN
23   else
24     OPEN ← OPEN ∪ {n}
25 return failure
```

Поиск лучшего родителя

Процедура SetBestPotentialParent(n):

```
1 glow(n) ← g(n); bpp(n) ← parent(n)
2 f(n) ← glow(n) + h(n)
3 BPPFound ← false
4 foreach n' ∈ PARENTS(n) do
5   if g(n') + h(n', n) < glow(n) then
6     bpp(n) ← n'
7     glow(n) ← g(n') + h(n', n)
8     f(n) ← glow(n) + h(n)
9     BPPFound ← true
10 return BPPFound
```

TO-AA-SIPP: Теоретические свойства (1/2)

Лемма 1. Для любого состояния поиска n либо $g_{low}(n) = g(bpp(n)) + h(bpp(n), n)$, либо $g_{low}(n) = g(n)$

Идея доказательства: по индукции (перед каждой итерацией цикла выполняется).

Лемма 2. Для любого состояния поиска n всегда выполняется $g_{low}(n) \leq g(n)$

Идея доказательства: рассмотрение всех возможных случаев изменения значения $g_{low}(n)$.

Лемма 3. Для любого i выполняется $g_{low}^i(n) \leq g_{low}(n)$, где $g_{low}^i(n)$ – это g_{low} -значение состояния n , извлеченного из списка OPEN в начале i -й итерации основного цикла, а $g_{low}(n)$ – это g_{low} -значение состояния n в конце i -й итерации.

Идея доказательства: от противного.

Лемма 4. Последовательность $\{f_{min}^1, f_{min}^2, \dots, f_{min}^K\}$, где f_{min}^i – это f -значение состояния поиска, извлеченного из OPEN на i -й итерации, является невозрастающей последовательностью.

Идея доказательства: рассмотрение всех возможных случаев обновления f -значений состояний.

Лемма 5. Список CLOSED содержит состояния поиска, g -значение которых не может быть уменьшено.

Идея доказательства: по индукции.

Теорема 1. Решение задачи AA-PFD, возвращаемое алгоритмом TO-AA-SIPP является оптимальным.

Идея доказательства: Следствие Леммы 5.

TO-AA-SIPP: Теоретические свойства (2/2)

Лемма 6. Когда состояние поиска n , т.ч. $\text{bpr}(n)=\text{parent}(n)$, извлекается из OPEN на очередной итерации основного цикла, то это состояние будет добавлено в список CLOSED на этой же итерации.

Идея доказательства: анализ всех возможных случаев.

Лемма 7. Если в основном цикле опустить проверку на достижение целевого состояния, то алгоритм завершит свою работу за конечное число итераций.

Идея доказательства: рассмотрение всех возможных случаев (с опорой на Лемму 6).

Теорема 2. Число итераций алгоритма ограничено сверху величиной $M(M-1)$, где M – число состояний поиска.

Идея доказательства: Следствие Леммы 7.

Лемма 8. Если в основном цикле опустить проверку на достижение целевого состояния, то за конечное число итераций всякое достижимое состояние будет добавлено в список CLOSED.

Идея доказательства: от противного.

Теорема 3. Алгоритм TO-AA-SIPP гарантирует отыскание решения задачи AA-PFD, если оно существует, и гарантирует корректное завершение работы, если решения нет.

Идея доказательства: Следствие Лемм 7 и 8.

Поиск субоптимальных решений задачи AA-PFD

TO-AA-SIPP

Достоинства

- Полнота
- Гарантия отыскания оптимальных решений
- Эффективней, чем прямой перебор

Недостатки

- Высокое время работы при решении практических задач
- Сложность механизма инверсивного поиска

Размен оптимальности решения
на повышенную эффективность

+

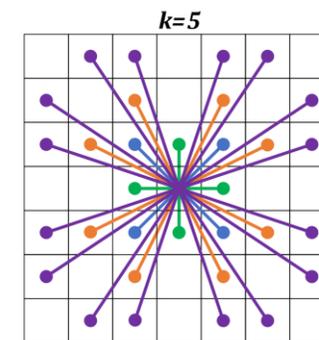
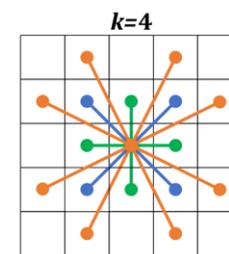
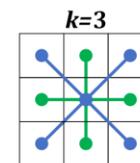
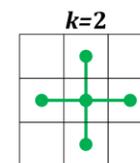
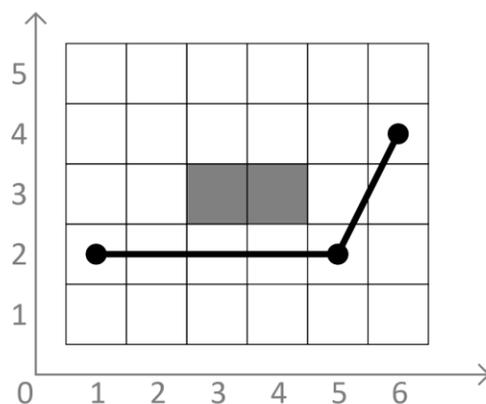
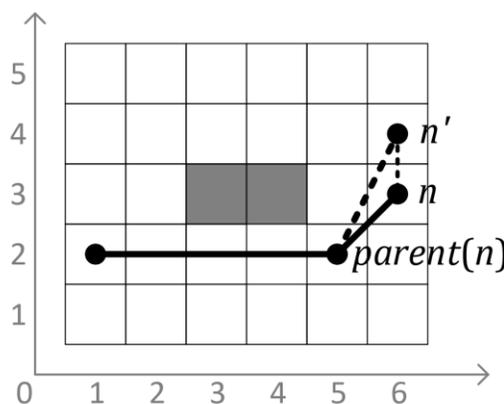
Стандартная (прямая) схема
поиска

AA-SIPP

(SIPP + ResetParent)

WSIPP

(SIPP + Lattice
+ WeightedH)



$$f(n) = g(n) + w \cdot h(n)$$

w – вес эвристики

Алгоритмы AA-SIPP и WSIPP

Теорема 4. Для любой решаемой задачи PFD алгоритм AA-SIPP найдёт решение, при этом стоимость решения не будет превосходить стоимость решения, найденного алгоритмом SIPP.

Идея доказательства: Любое состояние поиска, достижимое алгоритмом SIPP, достижимо также алгоритмом AA-SIPP с меньшей либо равной стоимостью.

Утверждение 1. Алгоритм WSIPP, использующий взвешенную монотонную эвристику, не гарантирует отыскания решения в случае, если процедура повторного рассмотрения состояний не предусмотрена

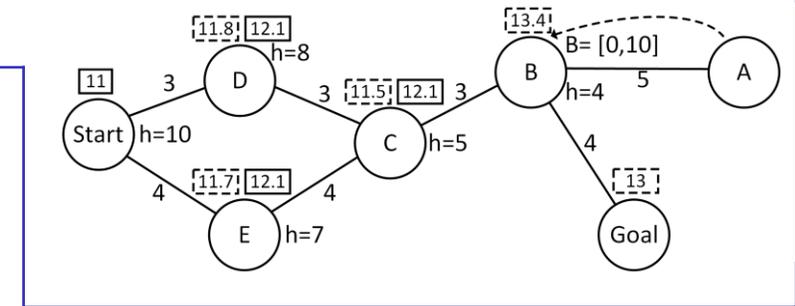
Идея доказательства: Конструктивно, через предъявление контрпримера.

AA-SIPP

- Простота реализации
- нет необходимости настройки параметров
- необходимость использования **los-функции**

WSIPP

- простота реализации
- возможна реализация без **los-функции**
- необходимость выбора (настройки) входных параметров
- необходимость реализации процедуры



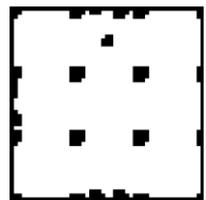
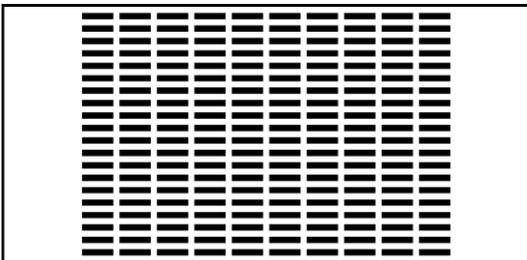
повторного рассмотрения состояний

Экспериментальные исследования

Практико-ориентированный сценарий: мобильный робот движется в среде с динамическими препятствиями (напр., другими роботами)

Входные данные

- ГРД (карта) random (32×32)
- ГРД (карта) arena (49×49)
- ГРД (карта) warehouse (170×84)
- безопасные интервалы \leftarrow динамические препятствия (*dobs*)
- $|dobs| = [32, 64, 96, 128]$
- Задание = старт + финиш + *dobs*
 - 500 заданий $\times |dobs|$
 - 2 000 заданий на ГРД
 - 6 000 заданий всего



Алгоритмы

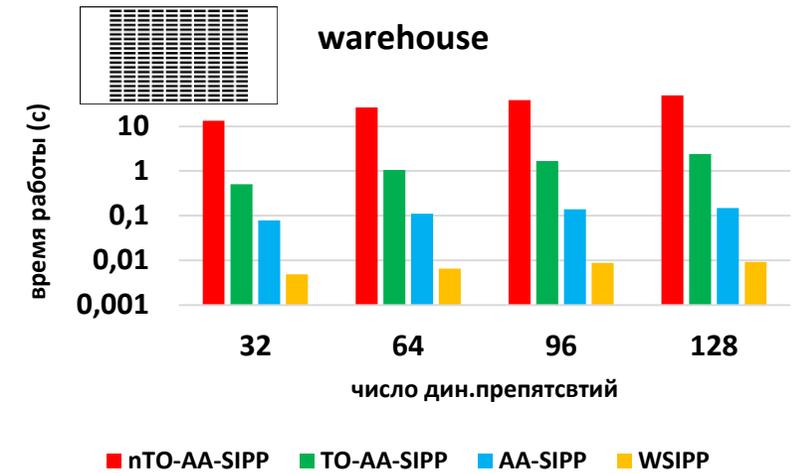
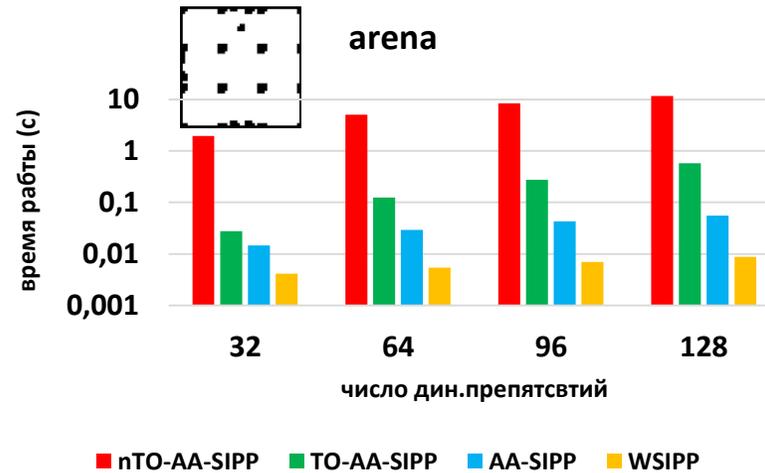
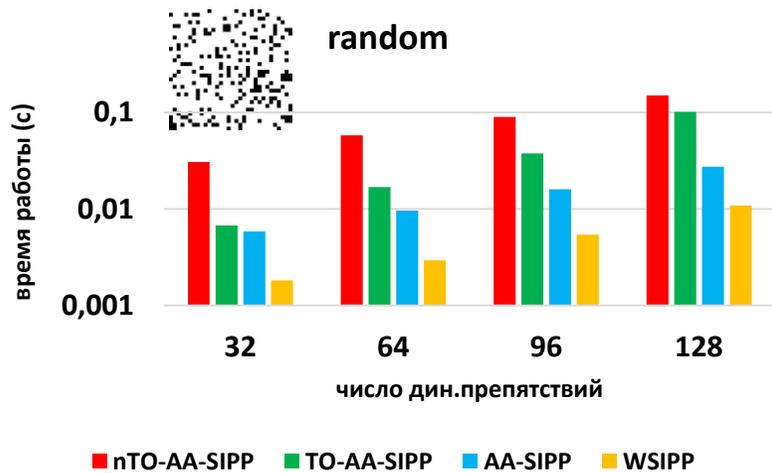
- nTO-AA-SIPP
- TO-AA-SIPP
- AA-SIPP
- WSIPP
 - $k=5, w=2$ (32 перехода в решетке, вес эвристики = 2)
- Все алгоритмы реализованы на C++ с использованием одинаковых структур данных и приемов программирования

По результатам предварительной серии экспериментов по настройке параметров

Основные вопросы для исследования

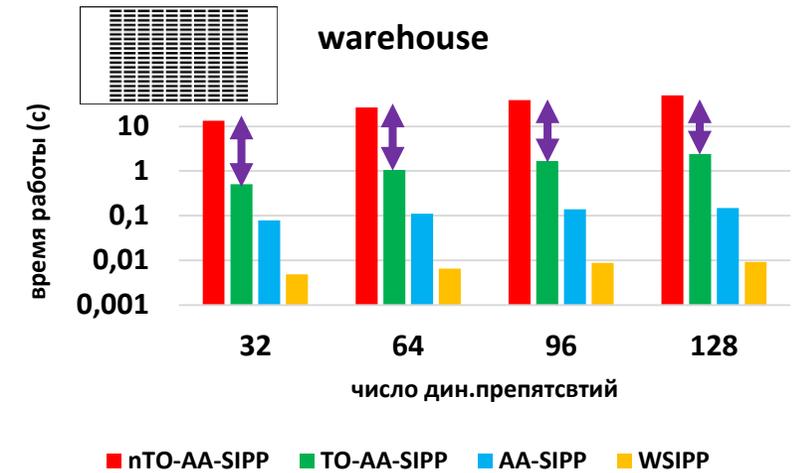
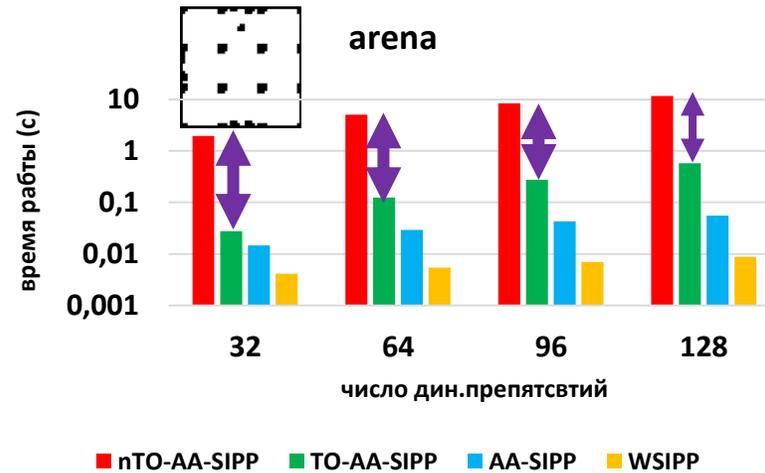
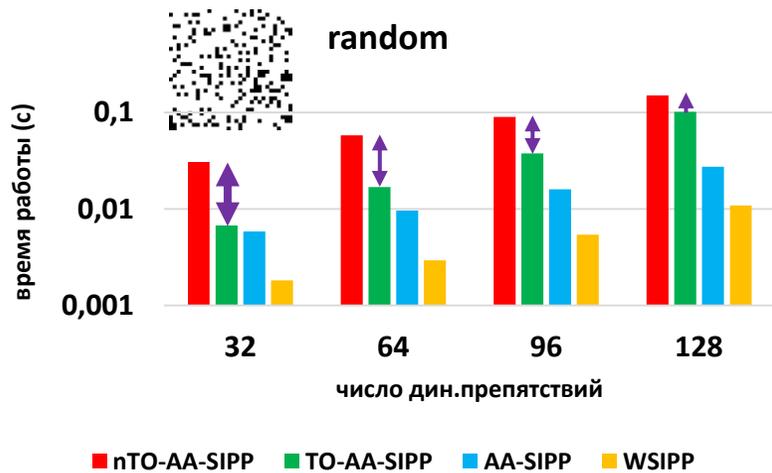
- Насколько TO-AA-SIPP эффективней (быстрее) nTO-AA-SIPP (единственного конкурента, способного оптимально решать задачи AA-PFD)
- Насколько хорошо алгоритмы AA-SIPP и WSIPP разменивают качество решения на скорость работы

Экспериментальные исследования – результаты



- OX – число динамических препятствий
- OY – время работы алгоритма (медиана по всем заданиям)
- Алгоритм = цвет

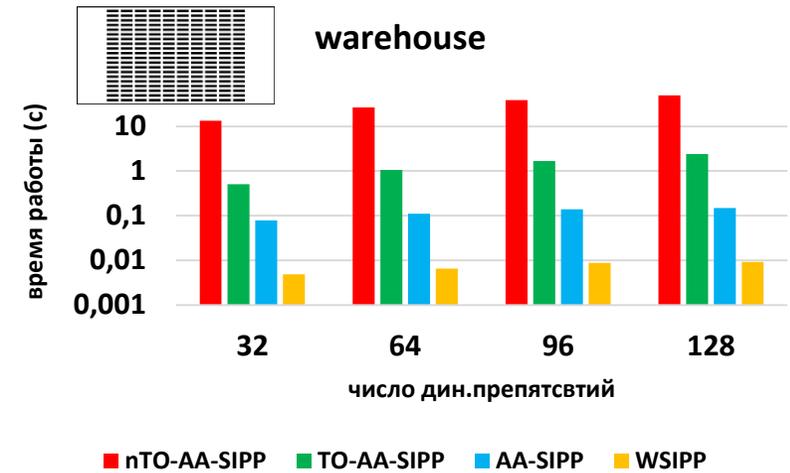
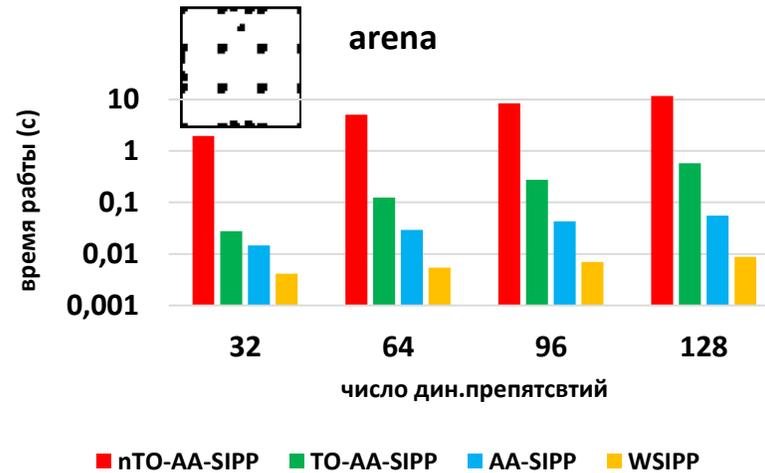
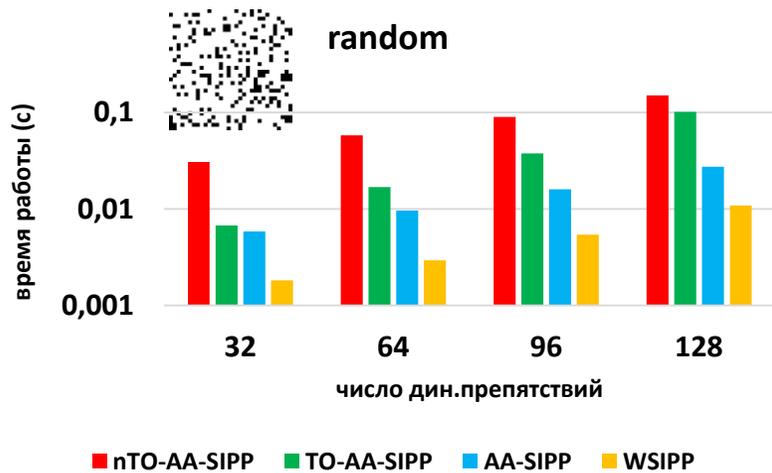
Экспериментальные исследования – результаты



TO-AA-SIPP быстрее nTO-AA-SIPP в 3-40 раз

- Наибольший прирост эффективности, до 40х, на ГРД arena (много свободного пространства)
- Наименьший прирост эффективности, до 3х, на ГРД random (мало свободного пространства)
- Прирост эффективности на ГРД warehouse до 20х (топология, максимально близкая к реальным применениям)

Экспериментальные исследования – результаты



TO-AA-SIPP быстрее nTO-AA-SIPP в 3-40 раз

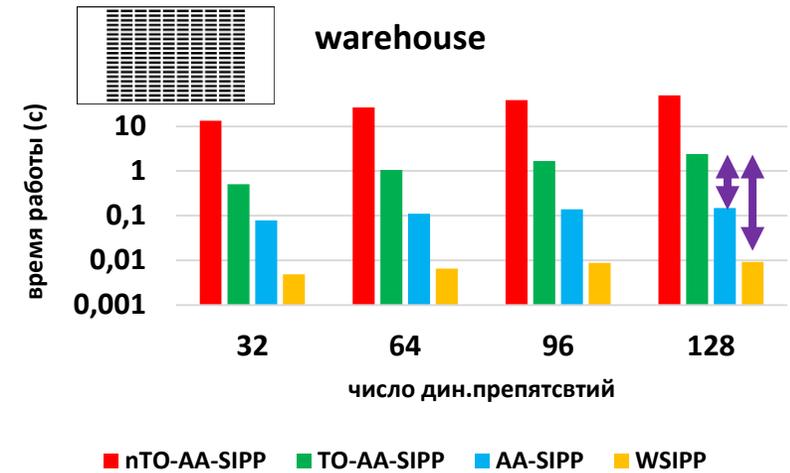
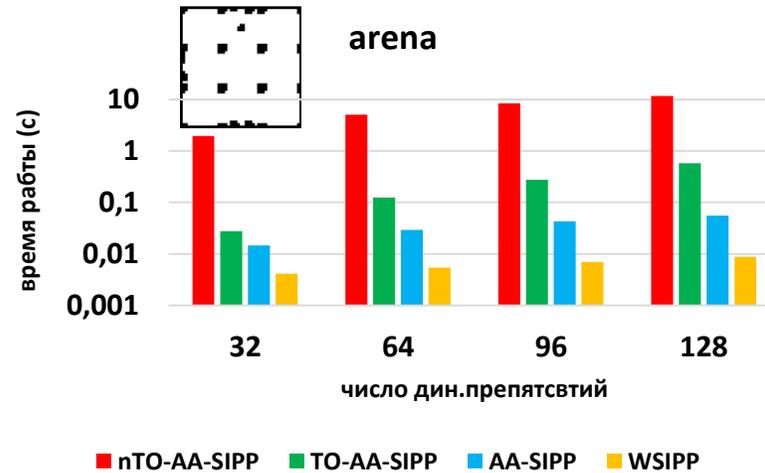
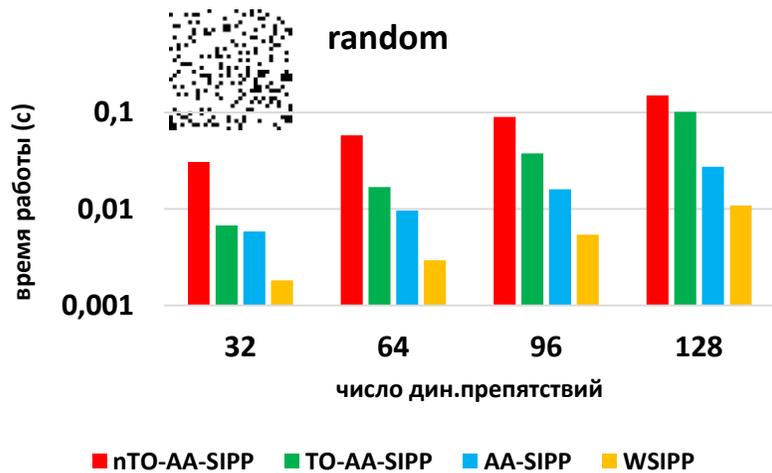
- Наибольший прирост эффективности, до 40x, на ГРД arena (много свободного пространства)
- Наименьший прирост эффективности, до 3x, на ГРД random (мало свободного пространства)
- Прирост эффективности на ГРД warehouse до 20x (топология, максимально близкая к реальным применениям)

Медианное число итераций, It , и вызовов (наиболее дорогой) процедуры алгоритма `ValidateTransition`, VT

#	TO-AA-SIPP		nTO-AA-SIPP	
	It	VT	It	VT
32	7 348	7 348	856	391 845
64	11 953	11 953	871	461 357
96	16 884	16 884	908	484 819
128	24 088	24 088	931	535 076

Снижение до 50x

Экспериментальные исследования – результаты

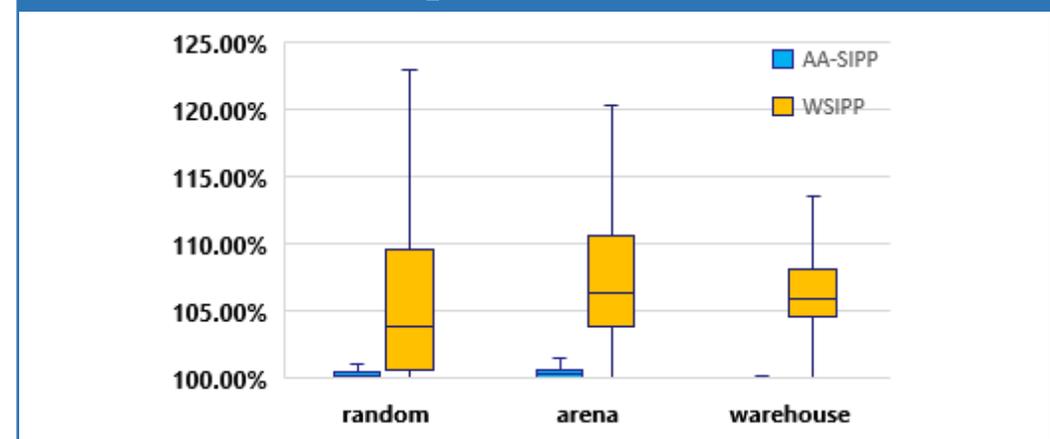


AA-SIPP и WSIPP быстрее TO-AA-SIPP на 1-2 порядка

#	random		arena		warehouse	
	AA-SIPP	WSIPP	AA-SIPP	WSIPP	AA-SIPP	WSIPP
32	1.2	3.7	1.9	6.7	6.5	104.0
64	1.8	5.8	4.3	23.0	9.6	160.7
96	2.4	7.0	6.4	39.2	11.9	188.0
128	3.7	9.3	10.4	66.3	16.0	258.5

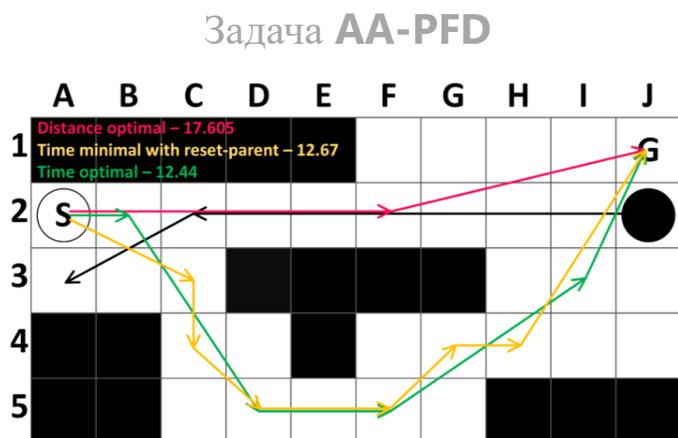
Ускорение по времени работы (раз)

Стоимость решений AA-SIPP заметно ниже стоимости решений WSIPP



Выводы (задача AA-PFD)

- Впервые предложен эффективный (в вычислительном смысле) алгоритм решения задачи **AA-PFD**, обладающий строгими теоретическими гарантиями – **TO-AA-SIPP**
 - Ускорение по сравнению с наивным поиском – до 50 раз.
- Предложены алгоритмы решения задачи **AA-PFD**, существенно повышающие эффективность решения задач на практике (за счет отказа от поиска оптимального решения), – **AA-SIPP** и **WSIPP**
 - Ускорение по сравнению с поиском оптимального решения – на 1-2 порядка
 - Стоимость решения повышается незначительно (на 1-2% в среднем в случае AA-SIPP)



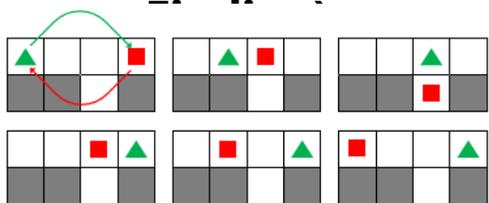
- * Нет необходимости подбирать входные параметры
- * Качество решений лучше (чем WSIPP)
- Простота реализации
- Теоретические гарантии

* Преимущества перед WSIPP

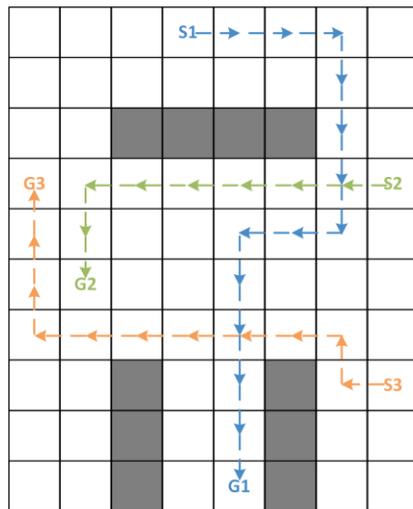
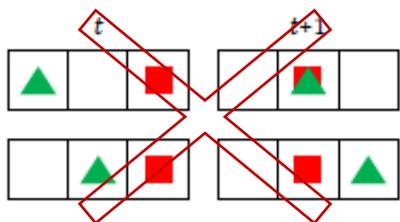
Поиск совокупности неконфликтных путей на ГРД

Практико-ориентированный сценарий: несколько мобильных роботов функционируют в общем рабочем пространстве и управляются центральным контроллером

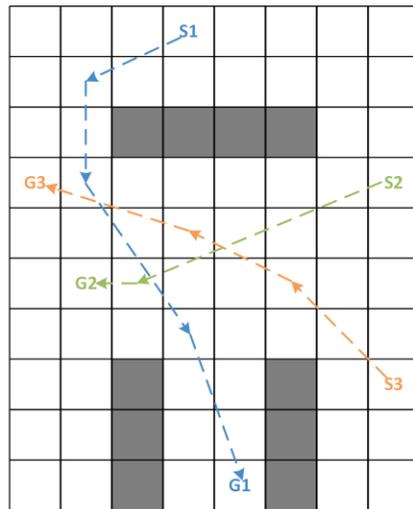
MAPF (Multi-Agent Path)



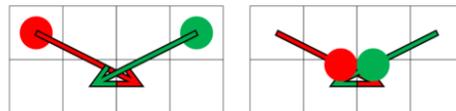
Конфликт:



AA-MAPF

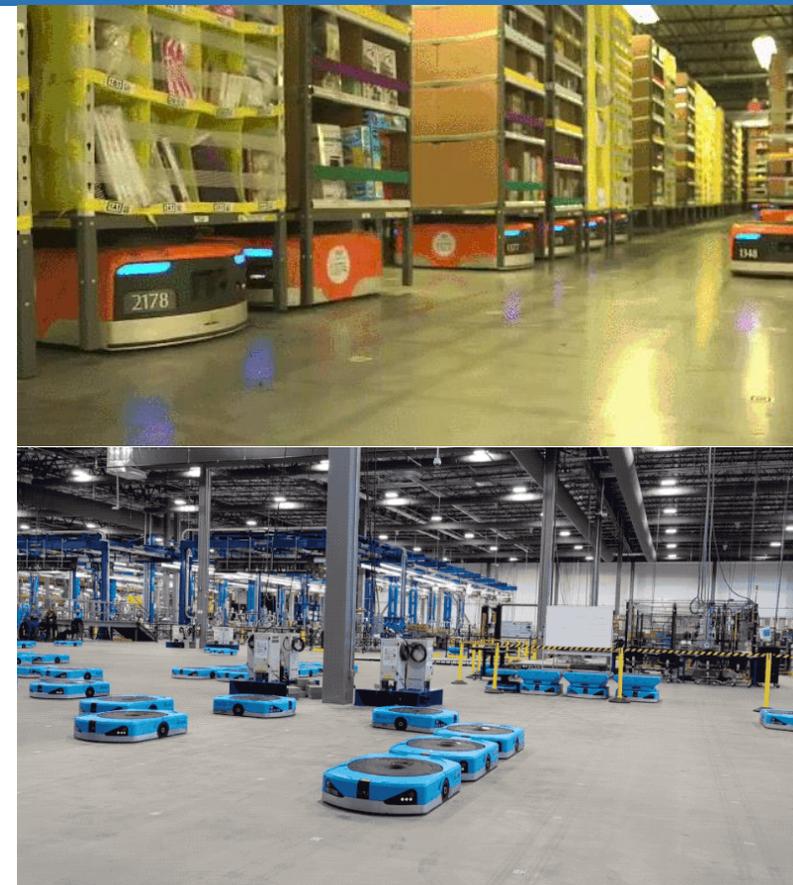


Конфликт:



AA – MAPF = (G, T, S, G, los, con)

- S, G – мн-во начальных/целевых вершин
- con – функция, определения конфликта



Задача АА-МАРФ

$$\text{AA} - \text{MAPF} = (\mathcal{G}, T, S, G, \text{los}, \text{con})$$

$\mathcal{G} = (V, E)$ – ГРД

$T = [0, \infty)$ – время

$S = \{s^1, s^2, \dots, s^n \mid s^i \in V\}$ – начальные вершины

$\text{los}: V \times V \rightarrow \{\text{True}, \text{False}\}$ – допустимость перехода

$G = \{g^1, g^2, \dots, g^n \mid g^i \in V\}$ – целевые вершины

Действие

$a = (u, v, \Delta) \mid u, v \in V, \Delta \in T$

$A = \{A_{\text{wait}}, A_{\text{move}}\}$

Ожидание

Перемещение

Множество всех действий

$u = v$

$u \neq v$

$\Delta \in T$

$\text{los}(u, v) = \text{true}$

Прямолинейное движение с единичной скоростью

$\Delta = \|u, v\|$

Путь (индивидуальный)

$\pi^i = (a_1^i, t_1^i), (a_2^i, t_2^i), \dots, (a_k^i, t_k^i)$

т.ч.

$\text{source}(a_1^i) = s^i, \text{target}(a_k^i) = g^i$

$\forall j \in [1, k-1]: \text{source}(a_{j+1}^i) = \text{target}(a_j^i)$

$t_j^i + \Delta(a_j^i) = t_{j+1}^i$

Стоимость пути: $\text{cost}(\pi^i) = t_k^i + \Delta(a_k^i)$

Конфликт (между действиями)

$\text{con}: A \times T \times A \times T \rightarrow \{\text{true}, \text{false}\}$

$\text{con}(a_k^i, t_k^i, a_l^j, t_l^j) = \text{true} \iff$

k -е действие i -го агента

l -е действие j -го агента

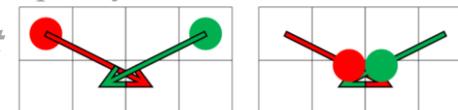
$\exists t \in [0, \min(\Delta(a_k^i), \Delta(a_l^j))]:$

$\text{dist}(\text{pos}(t, a_k^i, t_k^i), \text{pos}(t, a_l^j, t_l^j)) < 2 \cdot r$

r – радиус безопасности агента

pos – позиция агента в момент времени t

dist – расстояние между позициями



Конфликт (между путями π^i и π^j)

$\exists (a_k^i, t_k^i) \in \pi^i, (a_l^j, t_l^j) \in \pi^j: \text{con}(a_k^i, t_k^i, a_l^j, t_l^j) = \text{true}$

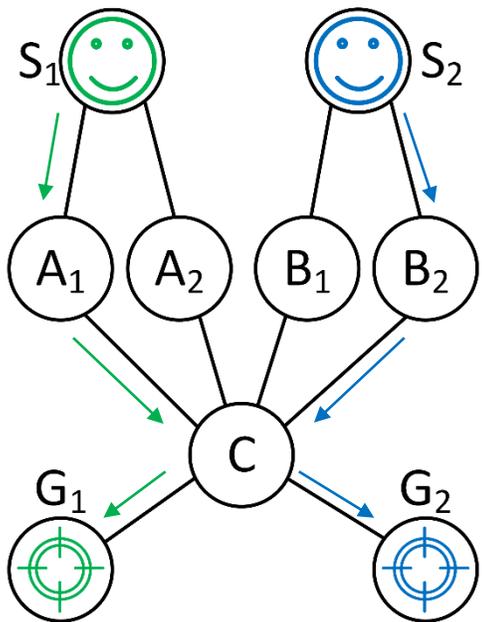
Задача: найти n неконфликтных путей

$\Pi = \{\pi^1, \pi^2, \dots, \pi^n\}$

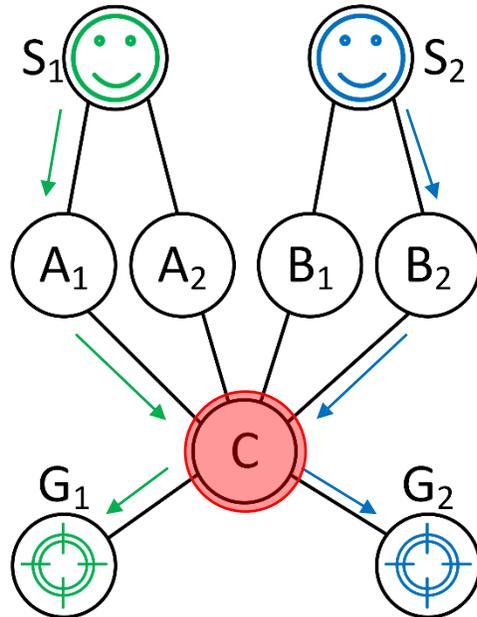
$\text{cost}(\Pi) = \sum_{i=1}^n \text{cost}(\pi^i)$ Стоимость решения (чем ниже, тем лучше)

Конфликтно-ориентированное планирование

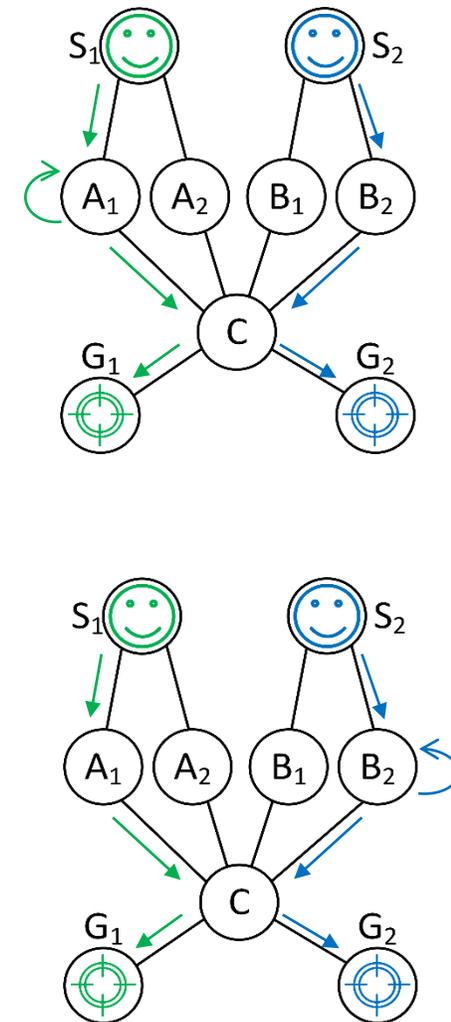
Поиск индивидуальных путей



Поиск и выбор конфликта



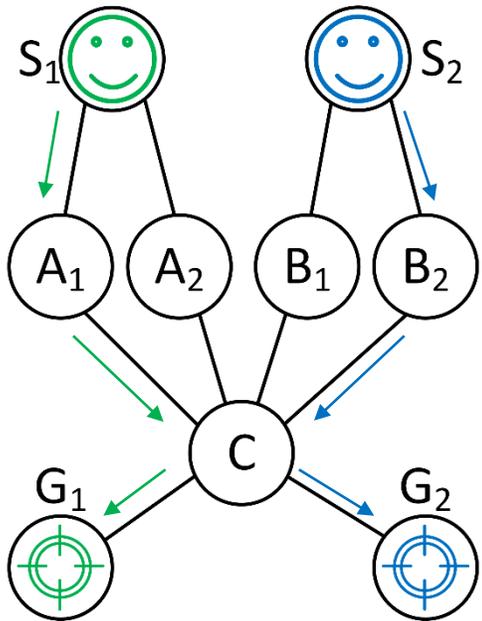
Наложение ограничения и перепланирование



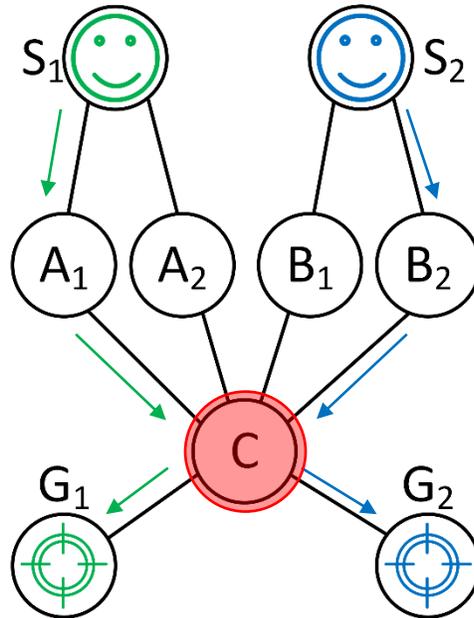
Sharon G., Stern R., Felner A., Sturtevant N.R. Conflict-based search for optimal multi-agent pathfinding. Artificial Intelligence, 219, 2015. pp. 40-66.

Конфликтно-ориентированное планирование (КОП)

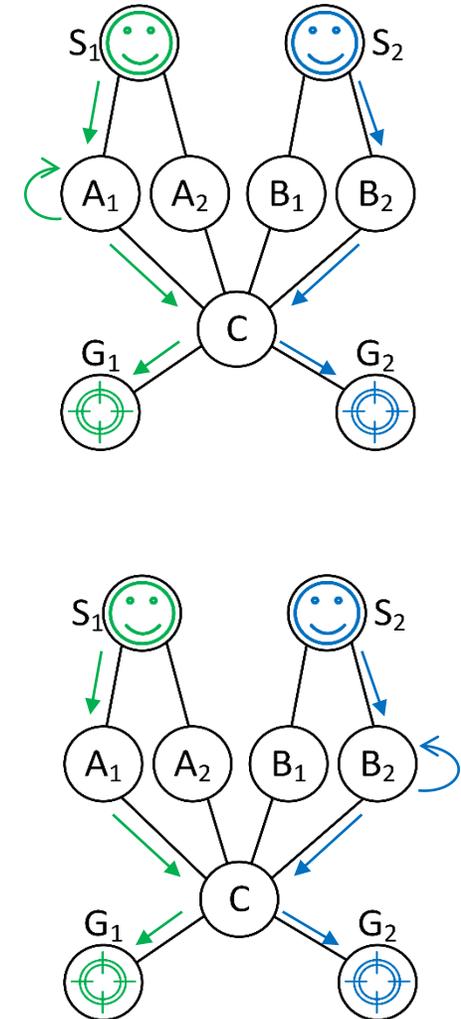
Поиск индивидуальных путей



Поиск и выбор конфликта



Наложение ограничений и перепланирование



Ключевая процедура КОП

Планирование с учетом ограничений
 $\text{FindPath}(G, s, g, \Psi)$

$\Psi = \{\psi_1, \dots, \psi_n\}$
 ψ_i – ограничение

- Необходимо гарантировать отыскание кратчайшего пути

Ограничения КОП

Задача MAPF

- **точечные ограничения** [1]
 - $\psi = (i, v, t)$ (агент i не должен быть в вершине v на шаге t)
 - $\psi = (i, e, t)$ (агент i не должен переходить по ребру e на шаге t)

Задача AA-MAPF

- **интервальные ограничения** [2, 3]
 $c = (a_k^i, t_k^i, a_l^j, t_l^j)$, т.ч. $con(c) = true$

$$a_k^i \in A_{move}, a_l^j \in A_{move}$$

- $\psi(i, c) = (a_k^i, [t_k^i, t_k^i + \alpha))$
- $\psi(j, c) = (a_l^j, [t_l^j, t_l^j + \beta))$

Первый момент времени, когда возможно перемещение без конфликта

Определяется по формуле

$$a_k^i \in A_{wait}, a_l^j \in A_{move}$$

- $\psi(i, c) = (v_{wait}(a_k^i), (t_1 + \delta, t_2))$
- $\psi(i, c) = (a_l^j, [t_l^j, t_l^j + \delta))$

Моменты времени, определяемые исходя из геометрических соображений

[1] Sharon G., Stern R., Felner A., Sturtevant N.R. Conflict-based search for optimal multi-agent pathfinding. Artificial Intelligence, 219, 2015. pp. 40-66.

[2] Andreychuk A., Yakovlev K, Surynek P., Atzmon D., Stern R. Multi-agent pathfinding with continuous time. Artificial Intelligence, 305, 2022. p.103662.

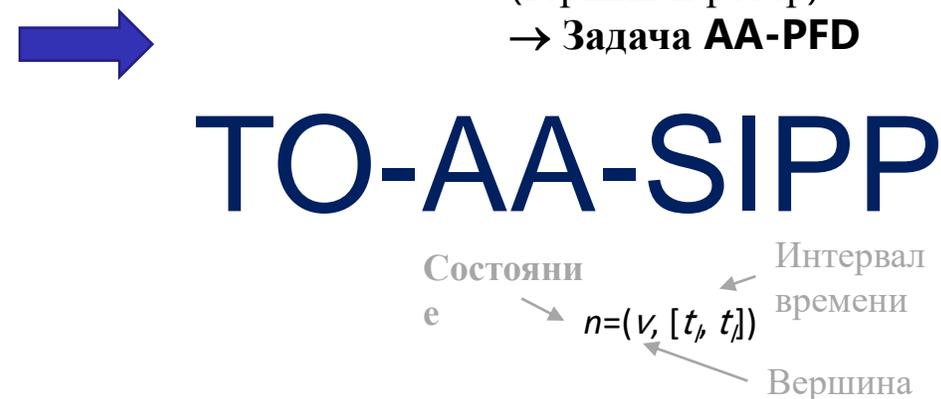
[3] Combrink A., Roselli S.F., Fabian M. Optimal Multi-agent Path Finding in Continuous Time. arXiv preprint arXiv:2508.16410, 2025.



Ограничения КОП → Безопасные интервалы

(вершин и ребер)

→ Задача AA-PFD



Алгоритм AA-CCBS

Алгоритм AA-CCBS($\mathcal{G}, s^1, s^2, \dots, s^n, g^1, g^2, \dots, g^n$):

Входные данные: Граф \mathcal{G} , множество начальных вершин s^1, s^2, \dots, s^n , множество целевых вершин g^1, g^2, \dots, g^n

Выходные данные: Множество неконфликтных путей $\Pi = \{\pi^1, \dots, \pi^n\}$

```
1  foreach  $i \in \{1, \dots, n\}$  do
2  |  $\pi_i \leftarrow \text{TO-AA-SIPP}(\mathcal{G}, s^i, g^i, \emptyset)$ 
3   $\Pi_0 \leftarrow \{\pi_1, \dots, \pi_n\}; \Psi_0 \leftarrow \emptyset; N_{root} \leftarrow (\Pi_0, \Psi_0)$ 
4   $OPEN \leftarrow \{N_{root}\}$ 
5  while  $OPEN \neq \emptyset$  do
6  |  $N_{best} \leftarrow \arg \min_{N \in OPEN} \text{cost}(N.\Pi)$ 
7  |  $c = (a_k^i, t_k^i, a_l^j, t_l^j) \leftarrow \text{GetConflict}(N_{best}.\Pi)$ 
8  | if  $c = \emptyset$  then
9  | | return  $N_{best}.\Pi$ 
10 | foreach  $z \in \{i, j\}$  do
11 | |  $\psi_{new} \leftarrow \text{GetConstraint}(z, c)$ 
12 | |  $\Psi_{new} \leftarrow N_{best}.\Psi \cup \psi_{new}$ 
13 | |  $\pi'_z \leftarrow \text{TO-AA-SIPP}(\mathcal{G}, s^z, g^z, \Psi_{new})$ 
14 | | if  $\pi'_z \neq \emptyset$  then
15 | | |  $\Pi_{new} \leftarrow N_{best}.\Pi \setminus \{\pi_z\} \cup \{\pi'_z\}$ 
16 | | |  $N_{new} \leftarrow (\Pi_{new}, \Psi_{new})$ 
17 | | |  $OPEN \leftarrow OPEN \cup \{N_{new}\}$ 
```

Утверждение 2. Алгоритм AA-CCBS гарантирует отыскание решения задачи AA-MAPF, если оно существует, при этом данное решение является оптимальным (минимальным по стоимости).

Идея доказательства: Следствие того, что алгоритм TO-AA-SIPP является полным и гарантирует отыскание оптимальных (индивидуальных) путей.

Алгоритм AA-CCBS

Алгоритм AA-CCBS($\mathcal{G}, s^1, s^2, \dots, s^n, g^1, g^2, \dots, g^n$):

Входные данные: Граф \mathcal{G} , множество начальных вершин s^1, s^2, \dots, s^n , множество целевых вершин g^1, g^2, \dots, g^n

Выходные данные: Множество неконфликтных путей $\Pi = \{\pi^1, \dots, \pi^n\}$

```
1  foreach  $i \in \{1, \dots, n\}$  do
2     $\pi_i \leftarrow \text{TO-AA-SIPP}(\mathcal{G}, s^i, g^i, \emptyset)$ 
3   $\Pi_0 \leftarrow \{\pi_1, \dots, \pi_n\}; \Psi_0 \leftarrow \emptyset; N_{root} \leftarrow (\Pi_0, \Psi_0)$ 
4   $OPEN \leftarrow \{N_{root}\}$ 
5  while  $OPEN \neq \emptyset$  do
6     $N_{best} \leftarrow \arg \min_{N \in OPEN} \text{cost}(N.\Pi)$ 
7     $c = (a_k^i, t_k^i, a_l^j, t_l^j) \leftarrow \text{GetConflict}(N_{best}.\Pi)$ 
8    if  $c = \emptyset$  then
9      return  $N_{best}.\Pi$ 
10   foreach  $z \in \{i, j\}$  do
11      $\psi_{new} \leftarrow \text{GetConstraint}(z, c)$ 
12      $\Psi_{new} \leftarrow N_{best}.\Psi \cup \psi_{new}$ 
13      $\pi'_z \leftarrow \text{TO-AA-SIPP}(\mathcal{G}, s^z, g^z, \Psi_{new})$ 
14     if  $\pi'_z \neq \emptyset$  then
15        $\Pi_{new} \leftarrow N_{best}.\Pi \setminus \{\pi_z\} \cup \{\pi'_z\}$ 
16        $N_{new} \leftarrow (\Pi_{new}, \Psi_{new})$ 
17        $OPEN \leftarrow OPEN \cup \{N_{new}\}$ 
```

Утверждение 2. Алгоритм AA-CCBS гарантирует отыскание решения задачи AA-MAPF, если оно существует, при этом данное решение является оптимальным (минимальным по стоимости).

Идея доказательства: Следствие того, что алгоритм TO-AA-SIPP является полным и гарантирует отыскание оптимальных (индивидуальных) путей.

Основная проблема

- Большое число вызовов трудоемкого алгоритма индивидуального планирования TO-AA-SIPP



Предлагаемое решение

- Мульти-ограничения

Мульти-ограничения (на действия перемещения)

Мульти-ограничение (МС)

$$\Psi(c, i) = \{\psi_1(i, c), \psi_2(i, c), \dots, \psi_M(i, c)\},$$

$$\psi_m(i, c) = (a_m^i, [t_m^i, t_m^i + \alpha_m)), m \in \overline{1, M},$$

$$M \in \mathbb{N}.$$

Запрет выполнять сразу несколько переходов

Мульти-ограничение типа 1 (МС1)

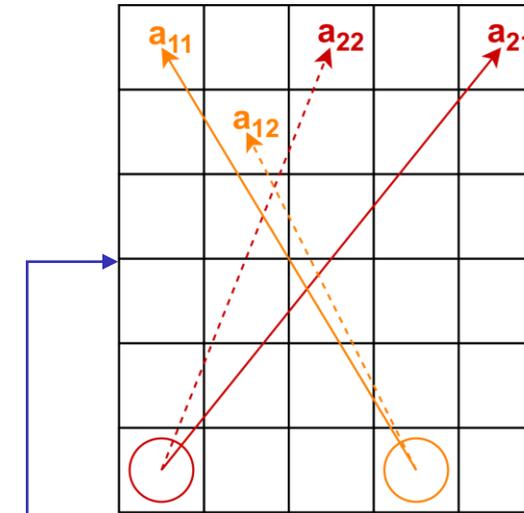
$(a_k^i, t_k^i), (a_l^j, t_l^j)$ – конфликтные действия (перемещения)

A_i, A_j – множества взаимно-конфликтных действий:

- $\forall a_m^i \in A_i, \forall a' \in A_j: \text{src}(a_m^i) = \text{src}(a')$
 $\text{con}(a_m^i, t_k^i, a', t_l^j) = \text{true}$

$$\psi_m(i, c) = (a_m^i, [t_k^i, t_k^i + \alpha'))$$

$[t_k^i, t_k^i + \alpha')$ – максимальный по продолжительности интервал, такой что полностью покрывает все конфликтные интервалы индуцированные действием a_m и любым действием из A_j .



Конфликтные интервалы

a_{11} vs a_{21} : [0, 1.106)	a_{21} vs a_{11} : [0, 0.616)
a_{11} vs a_{22} : [0, 0.522)	a_{21} vs a_{12} : [0, 0.835)
a_{12} vs a_{21} : [0, 1.215)	a_{22} vs a_{11} : [0, 1.057)
a_{12} vs a_{22} : [0, 0.638)	a_{22} vs a_{12} : [0, 0.912)

Мульти-ограничения (типа 1)

$$\Psi_1 \{(a_{11}, [0, 0.522)), (a_{12}, [0, 0.638))\} \quad \Psi_2 \{(a_{21}, [0, 0.616)), (a_{22}, [0, 0.912))\}$$

- Ограничивается сразу несколько переходов

Мульти-ограничения (на действия перемещения)

Мульти-ограничение (МС)

$$\Psi(c, i) = \{\psi_1(i, c), \psi_2(i, c), \dots, \psi_M(i, c)\},$$

$$\psi_m(i, c) = (a_m^i, [t_m^i, t_m^i + \alpha_m)), m \in \overline{1, M},$$

$$M \in \mathbb{N}.$$

Запрет выполнять сразу несколько переходов

Мульти-ограничение типа 1 (МС1)

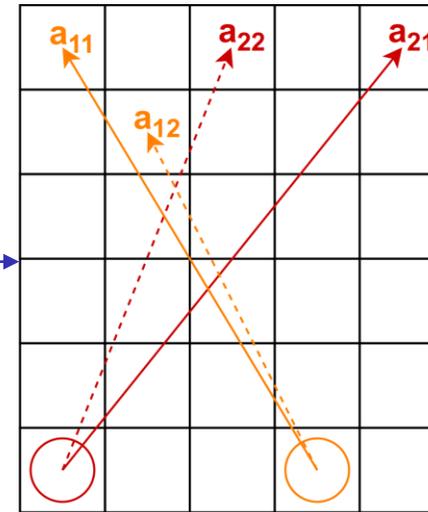
$(a_k^i, t_k^i), (a_l^j, t_l^j)$ – конфликтные действия (перемещения)

A_i, A_j – множества взаимно-конфликтных действий:

- $\forall a_m^i \in A_i, \forall a' \in A_j: \text{src}(a_m^i) = \text{src}(a')$
 $\text{con}(a_m^i, t_k^i, a', t_l^j) = \text{true}$

$$\psi_m(i, c) = (a_m^i, [t_k^i, t_k^i + \alpha'))$$

$[t_k^i, t_k^i + \alpha')$ – максимальный по продолжительности интервал, такой что полностью покрывает все конфликтные интервалы индуцированные действием a_m^i и любым действием из A_j .



Конфликтные интервалы

a_{11} vs a_{21} : [0, 1.106)	a_{21} vs a_{11} : [0, 0.616)
a_{11} vs a_{22} : [0, 0.522)	a_{21} vs a_{12} : [0, 0.835)
a_{12} vs a_{21} : [0, 1.215)	a_{22} vs a_{11} : [0, 1.057)
a_{12} vs a_{22} : [0, 0.638)	a_{22} vs a_{12} : [0, 0.912)

Мульти-ограничения (типа 1)

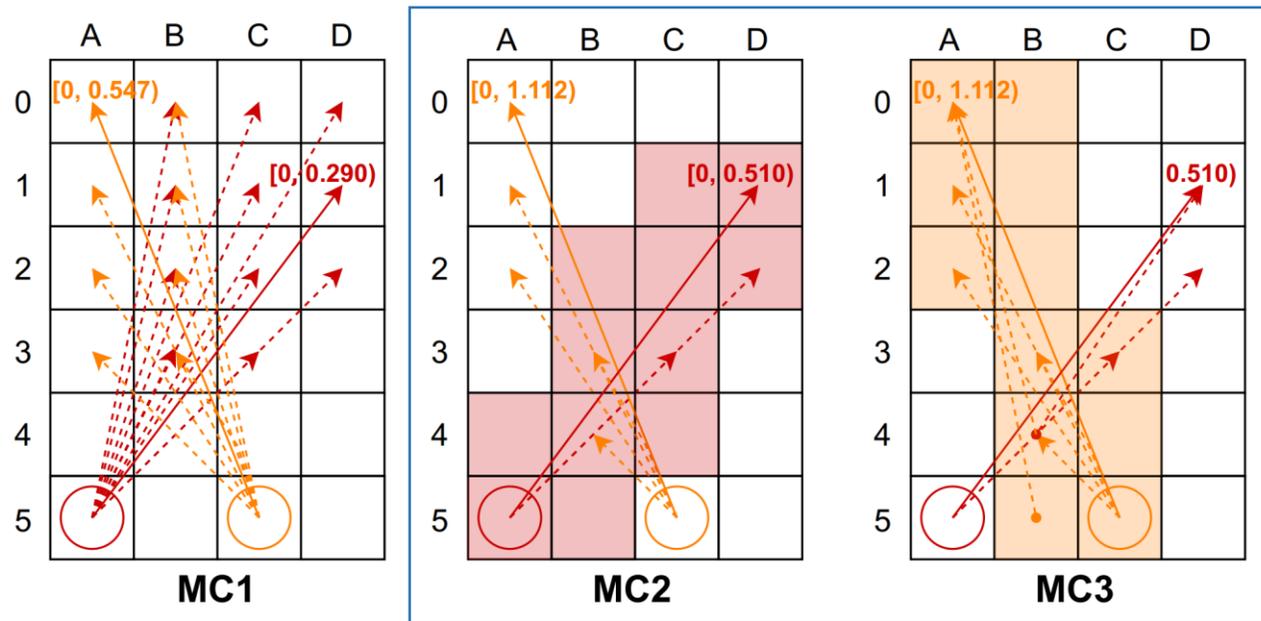
$$\Psi_1 \{(a_{11}, [0, 0.522)), (a_{12}, [0, 0.638))\}, \quad \Psi_2 \{(a_{21}, [0, 0.616)), (a_{22}, [0, 0.912))\}$$

- Ограничивается сразу несколько переходов
- Конфликтный интервал сужается

эффект от наложения мульти-ограничения может нивелироваться – сокращения пространства поиска может не произойти (из-за уменьшения продолжительности конфликтных интервалов)

Мульти-ограничения типа 2 и 3 (МС2, МС3)

Идея – дополнять исходное ограничение ограничениями на геометрически схожие действия, не уменьшая (принудительно) конфликтный интервал.



устранение исходного конфликта

Геометрически схожие перемещения

Исходный конфликтный интервал не уменьшается

```

Процедура GetMultiConstraints( $a^i, t^i, a^j, t^j, r$ ):
Входные данные: Действия и моменты времени, приводящие к
конфликту:  $a^i, t^i, a^j, t^j$ , настраиваемый
параметр  $r$  (радиус безопасности)
Выходные данные: Множества действий, формирующих
мульти-ограничение типа 2:  $A_i, A_j$ 
1  $[t^i, t^j] \leftarrow$  конфликтный интервал ( $a^i, t^i$ ) относительно ( $a^j, t^j$ )
2  $[t^j, t^i] \leftarrow$  конфликтный интервал ( $a^j, t^j$ ) относительно ( $a^i, t^i$ )
3  $A_i \leftarrow \emptyset; A_j \leftarrow \emptyset$ 
4  $\hat{V}_{\{i,j\}} \leftarrow$  вершины графа, удаленные не более чем на расстояние  $r$  от
отрезка  $[source(a_{\{i,j\}}), target(a_{\{i,j\}})]$ 
5  $\hat{A}_{\{i,j\}} \leftarrow$  действия, начинающиеся в  $source(a_{\{i,j\}})$  и заканчивающиеся
в  $\hat{V}_{\{i,j\}}$ 
6  $\hat{A} \leftarrow zip(\hat{A}_i, \hat{A}_j)$ 
7 for each  $\hat{a} \in \hat{A}$  do
8   if  $source(\hat{a}) = source(a_i)$  then
9     if  $IsConflict(\hat{a}, \hat{A}_j) = true$  then
10       $[t^j, \hat{t}] \leftarrow$  конфликтный интервал ( $a^j, t^j$ ) отн. ( $\hat{a}, t^i$ )
11      if  $t^j > \hat{t}$  then
12         $\hat{A}_i \leftarrow \hat{A}_i \setminus \{\hat{a}\}$ 
13     else
14       Обработка  $a^j$  аналогично (Строки 9-12)
15 for each  $a \in \hat{A}_i$  do
16    $t_{min} \leftarrow \infty$ 
17   for each  $a' \in \hat{A}_j$  do
18      $[t^i, t_{cur}] \leftarrow$  конфликтный интервал ( $a, t^i$ ) отн. ( $a', t^j$ )
19      $t_{min} \leftarrow \min\{t_{min}, t_{cur}\}$ 
20    $A_i \leftarrow A_i \cup \{(a, [t^i, t_{min}])\}$ 
21 Обработка  $\hat{A}_j$  аналогично (Строки 15-20)
22 return  $A_i, A_j$ 
    
```

Утверждение 3. Решения задачи AA-MARF, возвращаемое алгоритмом AA-CCBS+MC является оптимальным.

Идея доказательства: Мульти-ограничения по построению включают лишь элементы из множества взаимно-конфликтных действий.



Дальнейшее повышение эффективности AA-CCBS

Поиск оптимальных решений

- Мульти-ограничения (MC)
- Взаимоисключающие разделения [1, 2]
 - disjoint splitting (DS) Включают ограничения типа «перемещение должно быть совершенно»
- Комбинация MC и DS

Поиск ограниченно-субоптимальных решений

- Фокусировка поиска (Focal)

$$w > 1 \rightarrow \Pi_w: \\ \text{cost}(\Pi_w) \leq w \cdot \\ \text{cost}(\Pi^*)$$

Утверждение 4. Для любой задачи AA-MAPF, имеющей решение: $\text{cost}(\Pi_{\text{Focal-AA-SIPP}, w}) \leq w \cdot \text{cost}(\Pi^*)$

Идея доказательства: При использовании фокусировки меняется лишь порядок рассмотрения частичных решений, при этом каждое из них ограничено по стоимости должным образом.

[1] Li J., Harabor D., Stuckey P.J., Felner A., Ma H., Koenig S. Disjoint splitting for multi-agent path finding with conflict-based search // ICAPS 2019. pp. 279-283.

[2] Andreychuk A., Yakovlev K., Boyarski E., Stern R. Improving continuous-time conflict-based search // AAAI 2021. pp. 11220-11227.

Алгоритм Focal-AA-CCBS($\mathcal{G}, s^1, \dots, s^n, g^1, \dots, g^n, w, h_{\text{FOCAL}}$):

Входные данные: Граф \mathcal{G} , множество начальных (s^1, \dots, s^n), и целевых (g^1, \dots, g^n) вершин, фактор субоптимальности $w \geq 1$, эвристическая функция h_{FOCAL}

Выходные данные: Множество неконфликтных путей $\Pi = \{\pi^1, \dots, \pi^n\}$

```
1 foreach  $i \in \{1, \dots, n\}$  do
2    $\pi_i \leftarrow \text{TO-AA-SIPP}(\mathcal{G}, s^i, g^i, \emptyset)$ 
3  $\Pi_0 \leftarrow \{\pi_1, \dots, \pi_n\}; \Psi_0 \leftarrow \emptyset; N_{\text{root}} \leftarrow (\Pi_0, \Psi_0)$ 
4  $\text{OPEN} \leftarrow \{N_{\text{root}}\}; \text{FOCAL} \leftarrow \emptyset$ 
5 while  $\text{OPEN} \neq \emptyset$  do
6    $\text{cmin} \leftarrow \min_{N \in \text{OPEN}} \text{cost}(N, \Pi)$ 
7    $\text{FOCAL} \leftarrow \{N \mid N \in \text{OPEN}, \text{cost}(N) \leq w \cdot \text{cmin}\}$ 
8    $N_{\text{best}} \leftarrow \arg \min_{N \in \text{FOCAL}} h_{\text{FOCAL}}(N)$ 
   /* Стандартные шаги конфликтно-ориентированного
   планирования */
9
```

$$h_{\text{FOCAL}} = - \sum_{i=1}^n |N \cdot \Psi_i|$$

FOCAL – список состояний, каждое из которых удовлетворяет ограничению на заданную субоптимальность

h_{FOCAL} – вторичная эвристика

- наибольшее число ограничений

Приоритизированное планирование для поиска субоптимальных решений задачи AA-MARF

Общая схема ПП

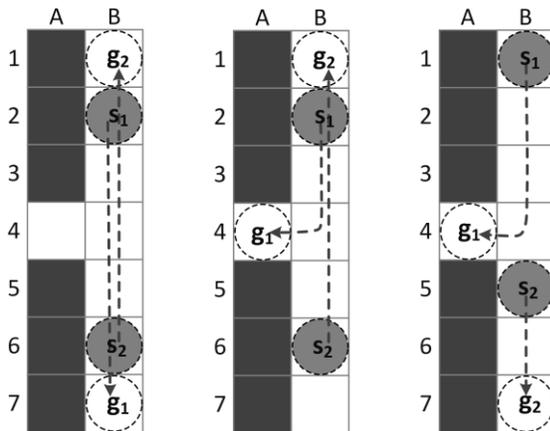
1. Назначение приоритетов агентам (парам (s_i, g_i))
2. Последовательный поиск индивидуальных путей
 - Необходимо избегать конфликтов с ранее построенными путями
 - Нет необходимости использовать (медленный) оптимальный планировщик

AA-
SIPP

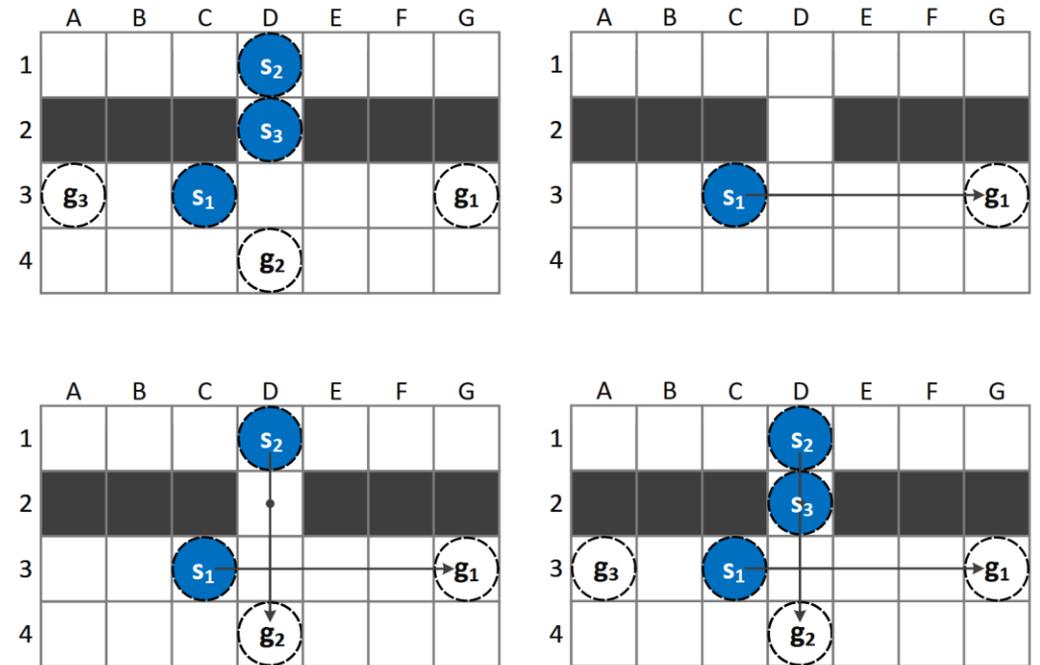
3 класса задач для ПП

- Решается вне зависимости от назначенных приоритетов
- Решается при определенной приоритизации
- Не решается при любой приоритизации

Важность процедур назначение и переназначения приоритетов



Начальные положение низкоприоритизированных



Проблема

Высокоприоритизированный агент (2) «сбивает» низкоприоритизированного агента (3) на старте

Предлагаемое решение

$[0, t_{ssj}]$ – безопасный интервал для любого s_j (подбирается эмпирически)

Алгоритм AA-SIPP(m)

Алгоритм AA-SIPP(m) ($\mathcal{G}, s^1, \dots, s^n, g^1, \dots, g^n, t_{ssi}$):

Входные данные: Граф $\mathcal{G} = (V, E)$, множество начальных (s^1, \dots, s^n), и целевых (g^1, \dots, g^n) вершин, продолжительность начального безопасного интервала t_{ssi}

Выходные данные: Множество неконфликтных путей
 $\Pi = \{\pi^1, \dots, \pi^n\}$

1 $ps = \{i_1, i_2, \dots, i_n\} \leftarrow \text{SetInitialPriorities}()$

2 **foreach** $v \in V$ **do**
 3 $T_{safe}(v) \leftarrow \{[0, t_{ssi}]\}$

4 $PS \leftarrow \emptyset$

5 **while** *true* **do**

6 $PS \leftarrow PS \cup \{ps\}$

7 $\Pi \leftarrow \emptyset$

8 **for** j *from* 1 *to* n **do**

9 $\pi_j \leftarrow \text{AA-SIPP}(\mathcal{G}, s^{i_j}, g^{i_j}, \{T_{safe}(v)\})$

10 **if** π_j *is found* **then**

11 $\Pi \leftarrow \Pi \cup \{\pi_j\}$

12 UpdateSafeIntervals(\mathcal{G}, π_j)

13 **if** $j = n$ **then**

14 **return** Π

15 **else**

16 $ps \leftarrow \{i_1 = i_j, i_2 = i_1, \dots, i_n = i_{n-1}\}$

17 **if** $ps \in PS$ **then**

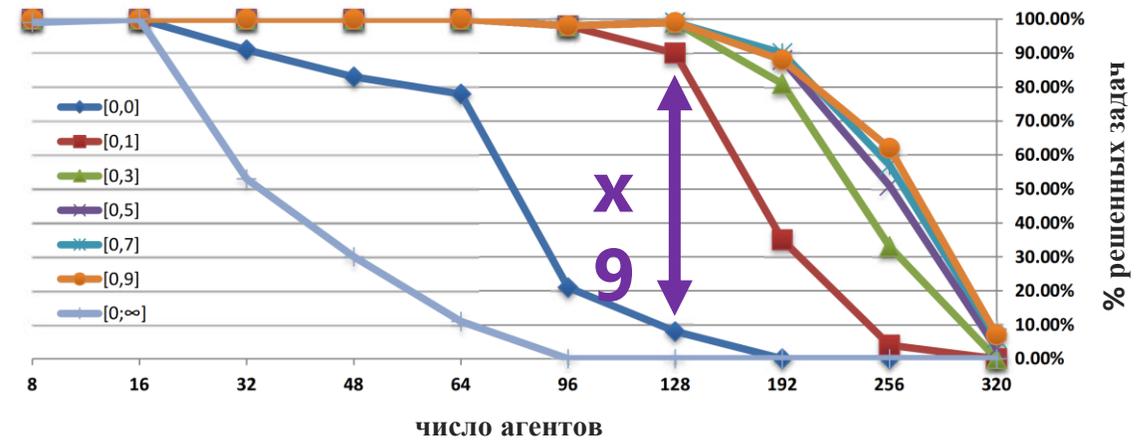
18 **return** *failure*

19 **else**

20 **break**

Агент, для которого не удалось найти путь, получает наивысший приоритет

Предлагаемые техники ПП эффективны на



число агентов	Random Restarts			Rule-Based Restarts			SR – % решенных задач
	SR	time	cost	SR	time	cost	
16	100	0.01	285.15	100	0.01	285.15	
32	100	0.04	591.3	100	0.04	591.3	
64	100	0.22	1 274.94	100	0.17	1 252.82	
96	100	0.68	2 212.14	100	0.49	2 149.31	
128	100	5.73	3 543.92	100	1.4	3 322.52	
160	42	117.82	5 220.9	99	28.28	5 298.85	

time – время работы алгоритма
 cost – стоимость найденного решения

x2 (from 128 to 160 agents)
 x4 (from 128 to 160 agents)

Экспериментальные исследования (серия 1)

Входные данные [1]

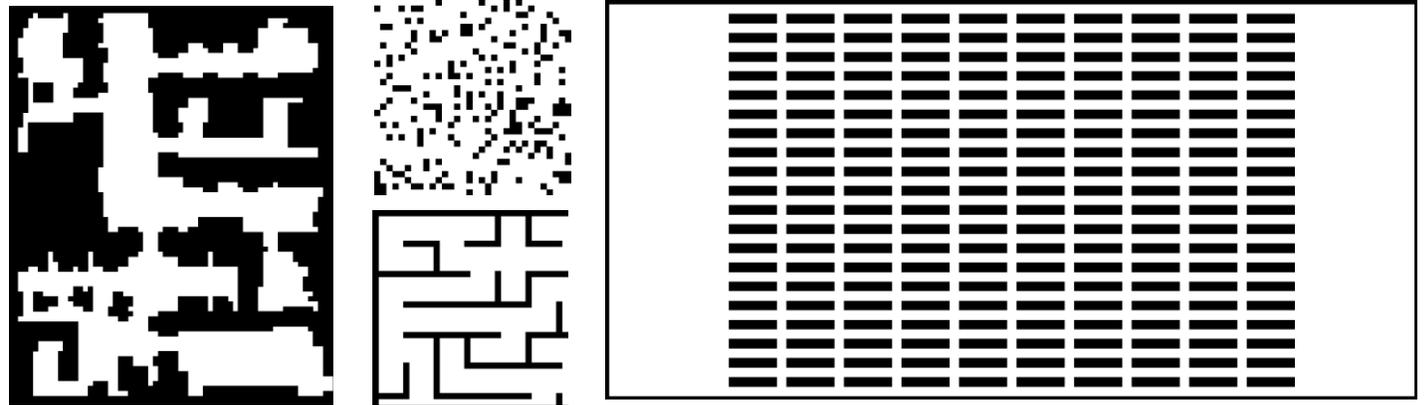
- ГРД (карта) random (32 × 32)
- ГРД (карта) maze (32 × 32)
- ГРД (карта) den312 (65 × 81)
- ГРД (карта) warehouse (170 × 84)
- 25 сценариев на карту
- сценарий = набор (s, g) для > 100 агентов

Методология [1]

- задание = первые x пар (s, g) из сценария
- последовательное увеличение сложности заданий (+1 агент к предыдущему)
- ограничение по времени 300 с.

Алгоритмы (*)

- AA-CCBS (+MC,+DS,+MC+DS)
- Focal-AA-CCBS (+MC,+DS,+MC+DS)
- AA-SIPP(m)



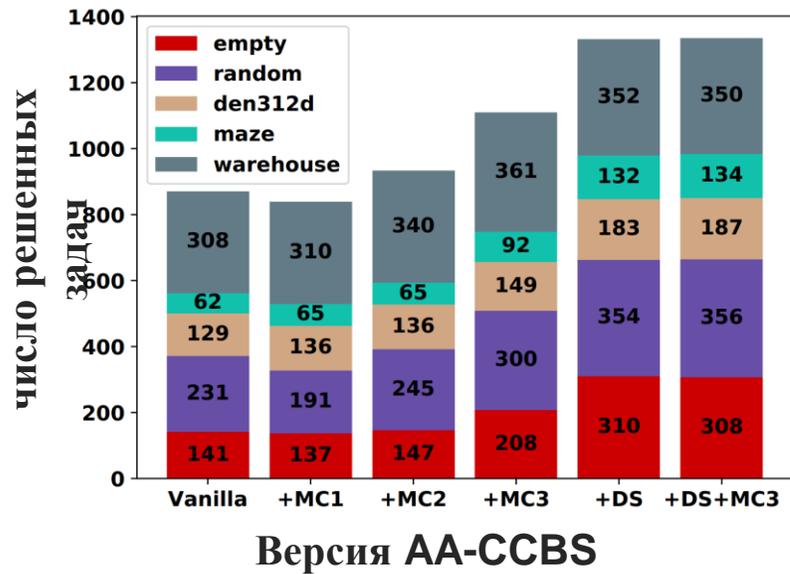
Основные вопросы для исследования

- Насколько предлагаемый в работе подход наложения мультиограничений повышает эффективность AA-CCBS
- Насколько фокусировка поиска (Focal-A-SIPP) ускоряет алгоритм AA-CCBS (и насколько при этом снижается качество решений)
- Как себя показывает приоритизированное планирование (AA-SIPP(m)) в сравнении с конфликтно-ориентированным (AA-CCBS)

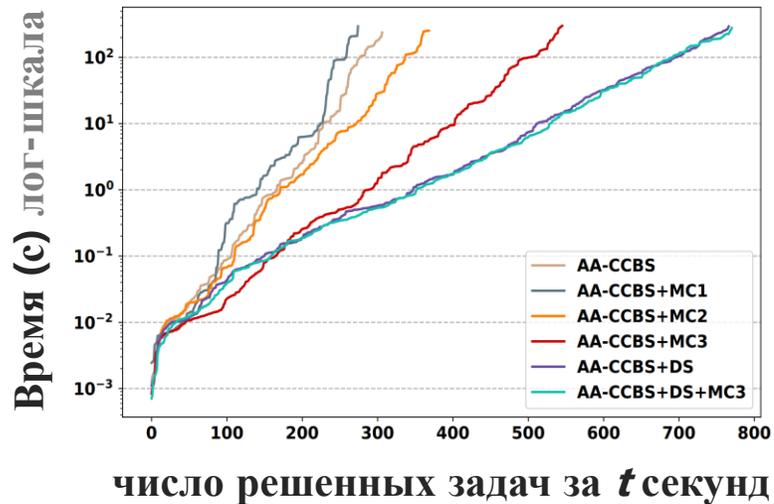
* Не существует прямых аналогов (алгоритмов решения задачи AA-MAPF)

[1] Stern R., Sturtevant N., Felner A., Koenig S., Ma H., Walker T., Li J., Atzmon D., Cohen L., Kumar T.K., Boyarski E., Barták R. Multi-agent pathfinding: Definitions, variants, and benchmarks // SoCS 2019. pp. 151-158.

Результаты экспериментов (AA-CCBS)

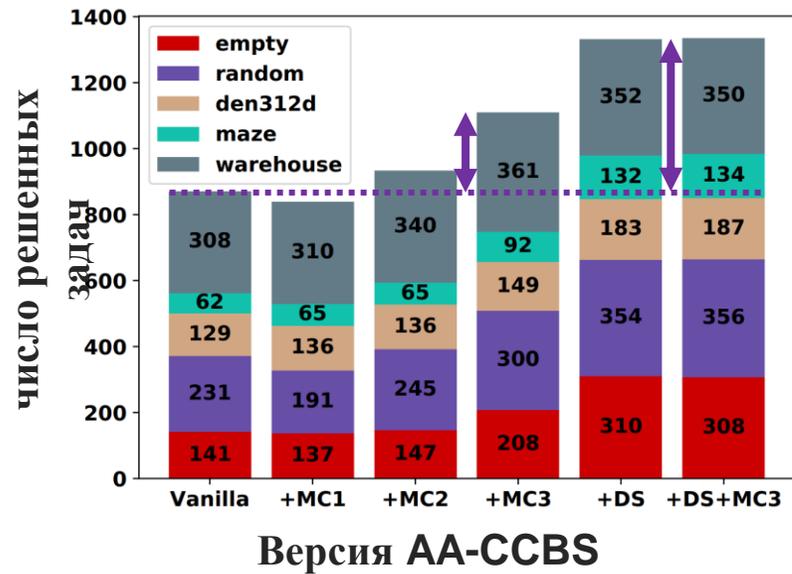


Разные цвета = разные карты

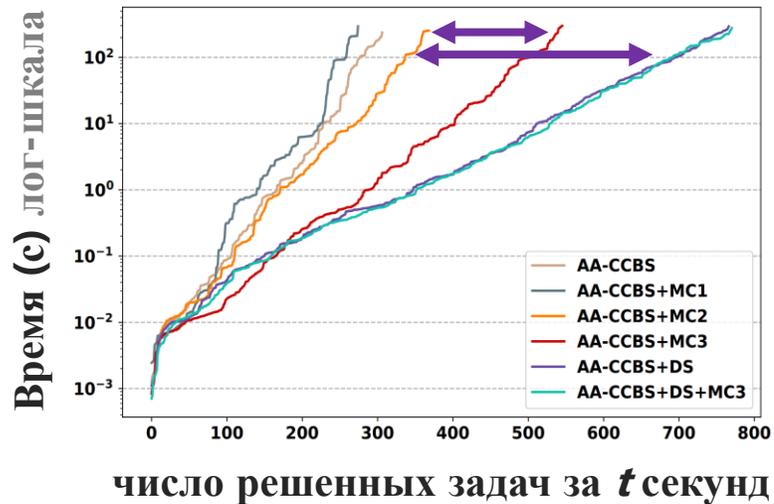


Разные цвета = разные алгоритмы

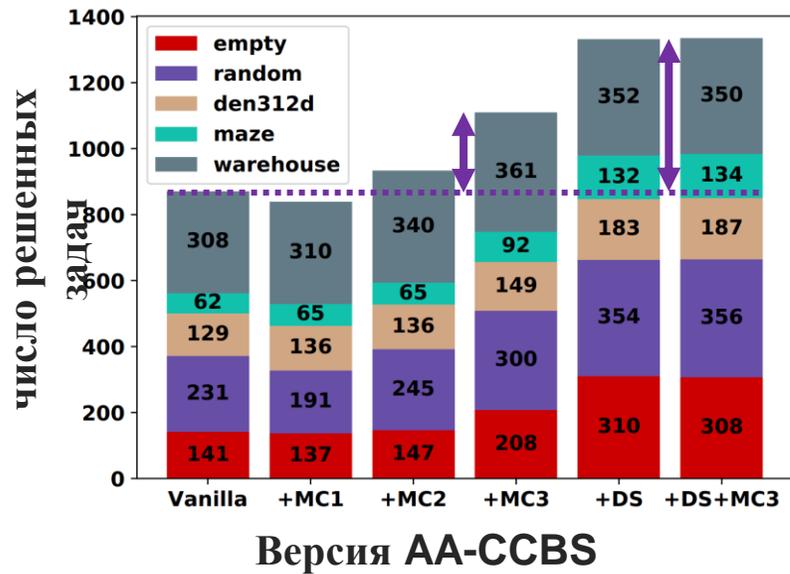
Результаты экспериментов (AA-CCBS)



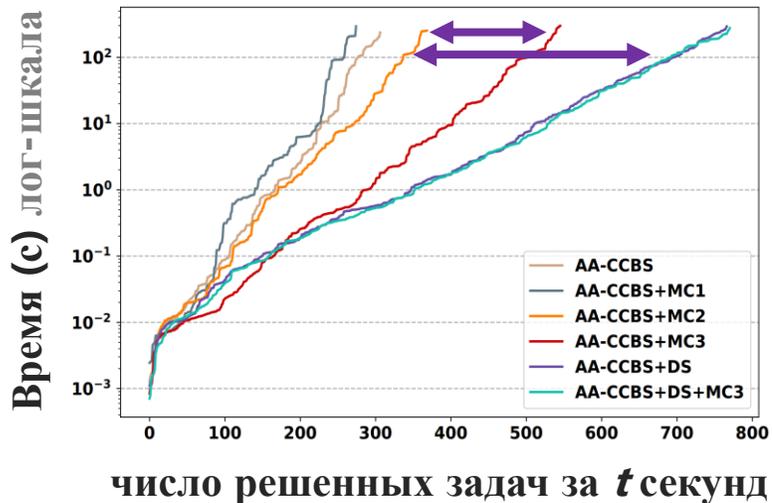
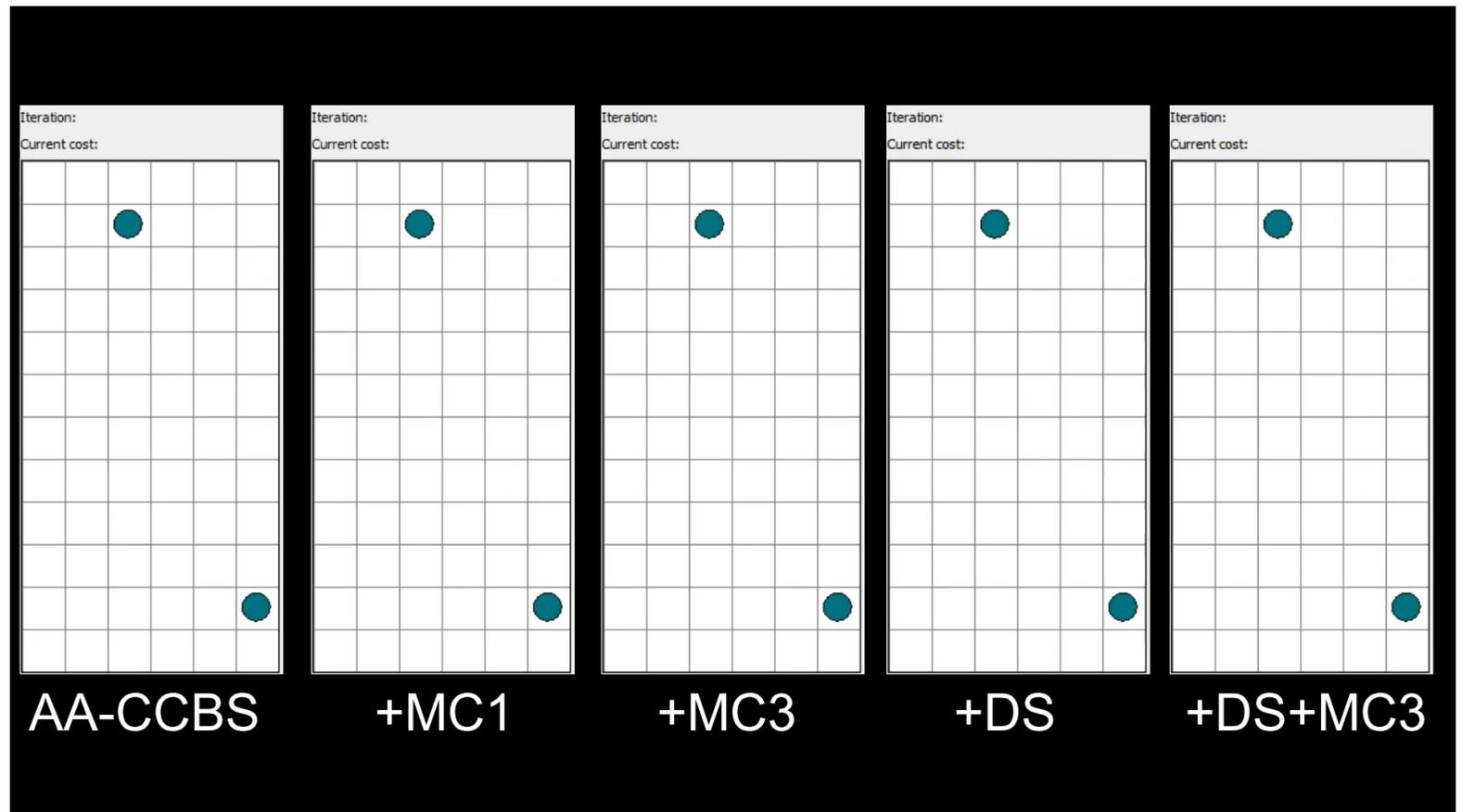
Предложенная в работе техника MC3 заметно повышает эффективность алгоритма (также как и DS)



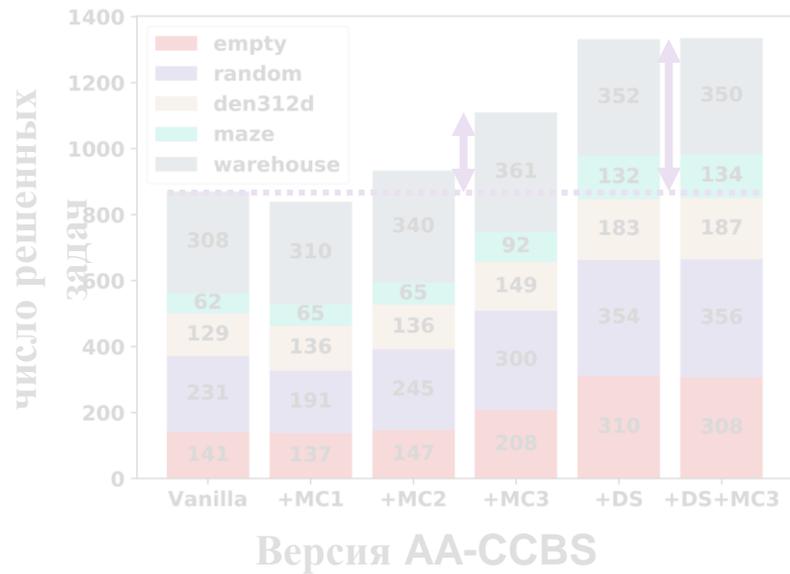
Результаты экспериментов (AA-CCBS)



Предложенная в работе техника MC3 заметно повышает эффективность алгоритма (также как и DS)



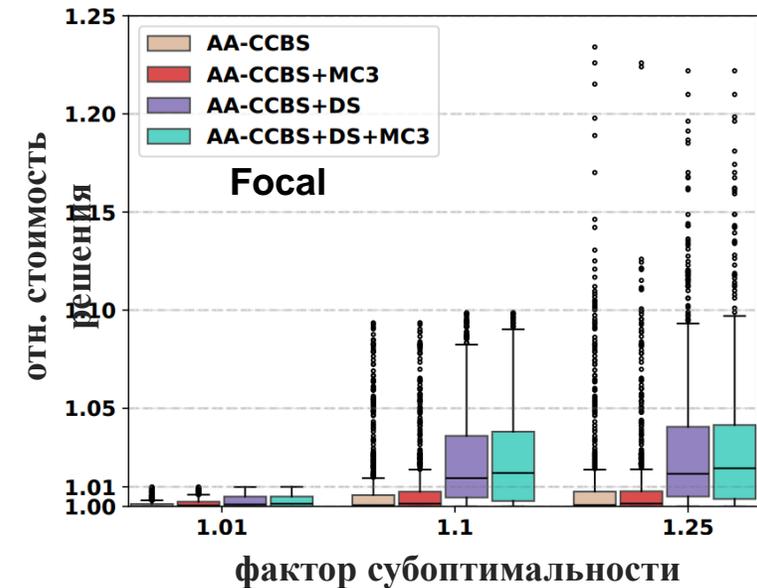
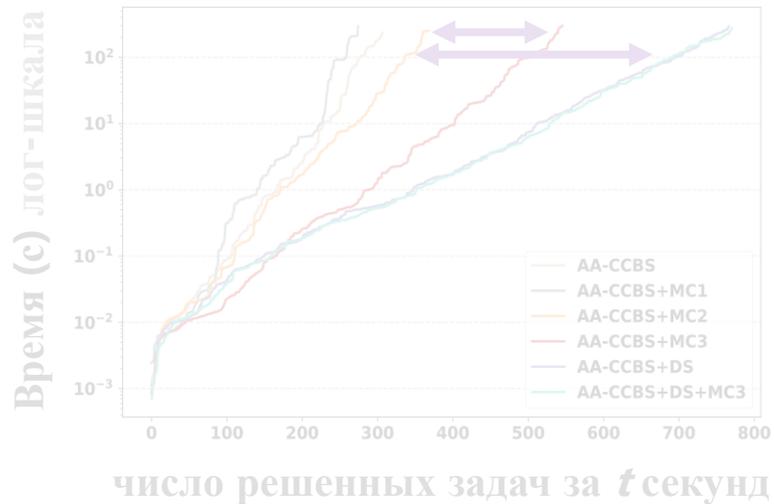
Результаты экспериментов (Focal-AA-CCBS)



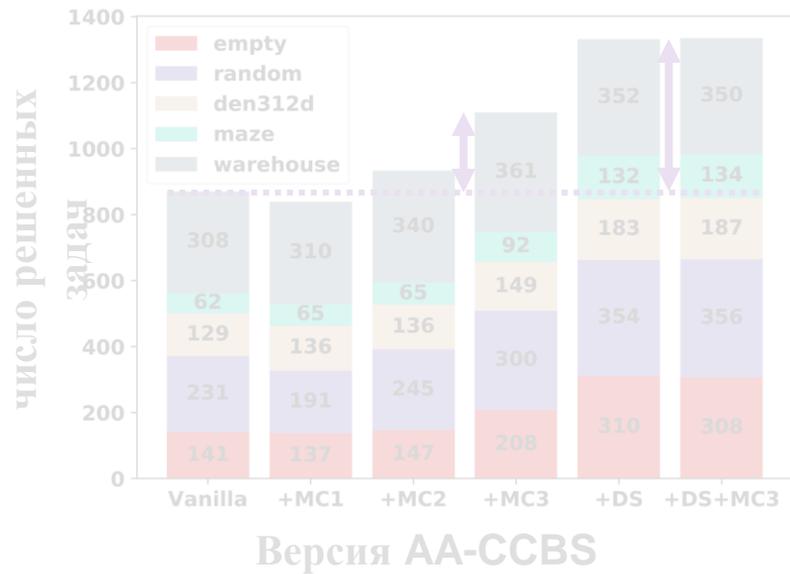
Предложенная в работе техника MC3 заметно повышает эффективность алгоритма (также как и DS)

Focal	фактор субоптимальности			
	$w = 1.0$	$w = 1.01$	$w = 1.1$	$w = 1.25$
AA-CCBS	871	1653	1929	1950
AA-CCBS+MC3	1110	2002	2437	2440
AA-CCBS+DS	1331	1661	1583	1564
AA-CCBS+DS+MC3	1335	1626	1561	1543

число решенных задач



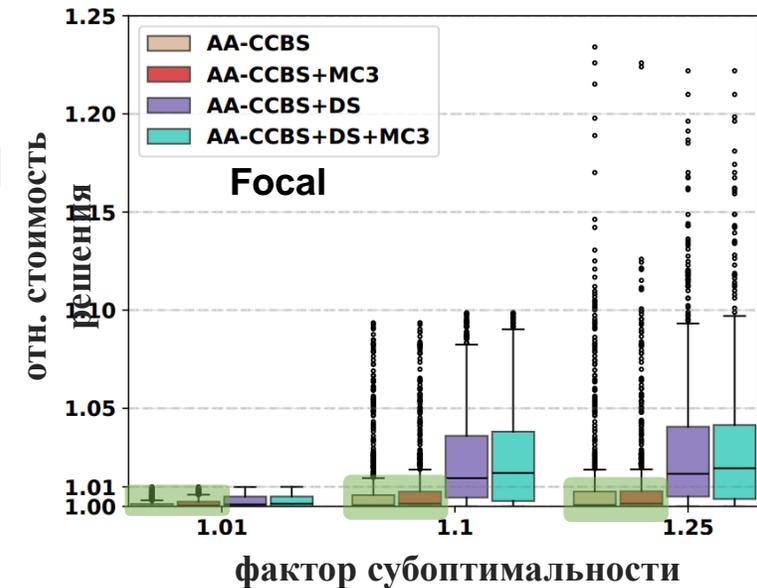
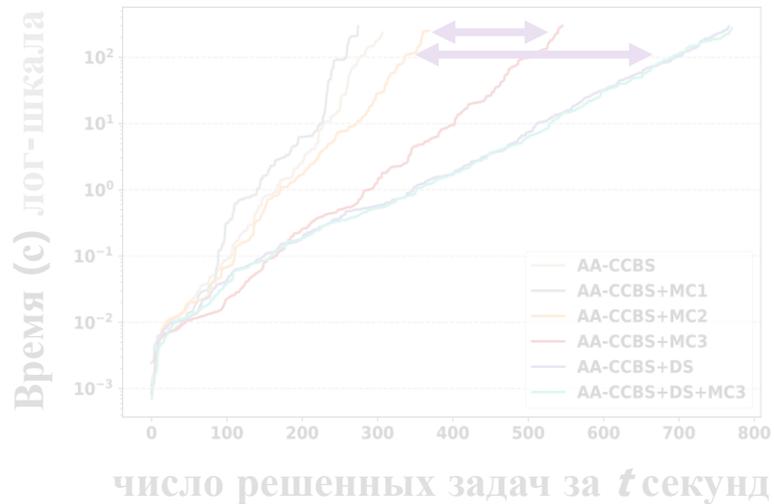
Результаты экспериментов (Focal-AA-CCBS)



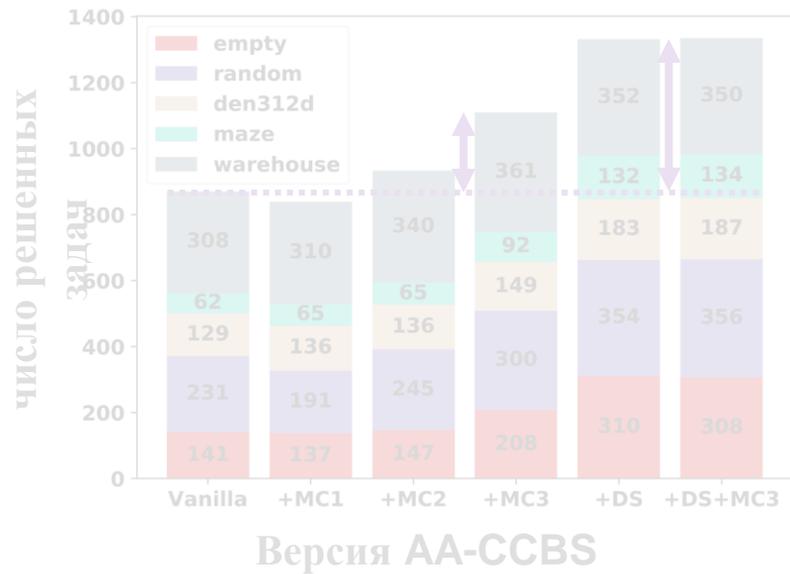
Предложенная в работе техника MC3 заметно повышает эффективность алгоритма (также как и DS)

Focal	фактор субоптимальности			
	$w = 1.0$	$w = 1.01$	$w = 1.1$	$w = 1.25$
AA-CCBS	871	1653	1929	1950
AA-CCBS+MC3	1110	2002	2437	2440
AA-CCBS+DS	1331	1661	1583	1564
AA-CCBS+DS+MC3	1335	1626	1561	1543

число решенных задач



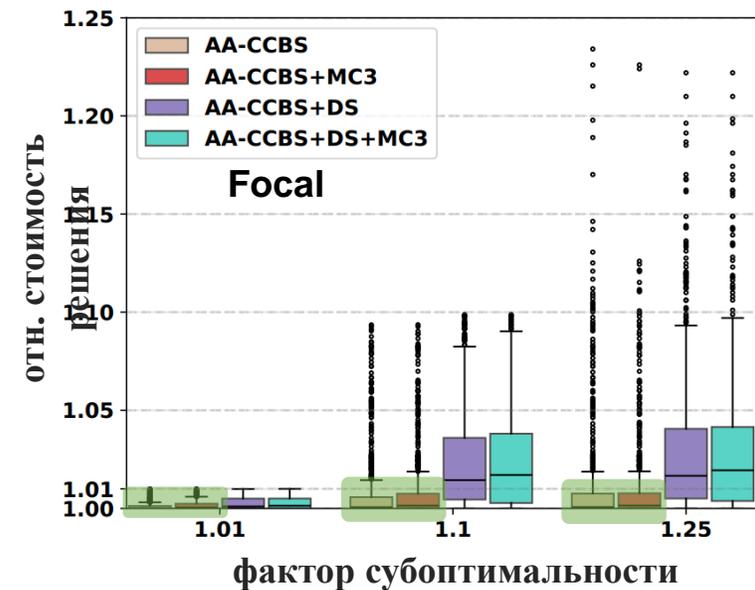
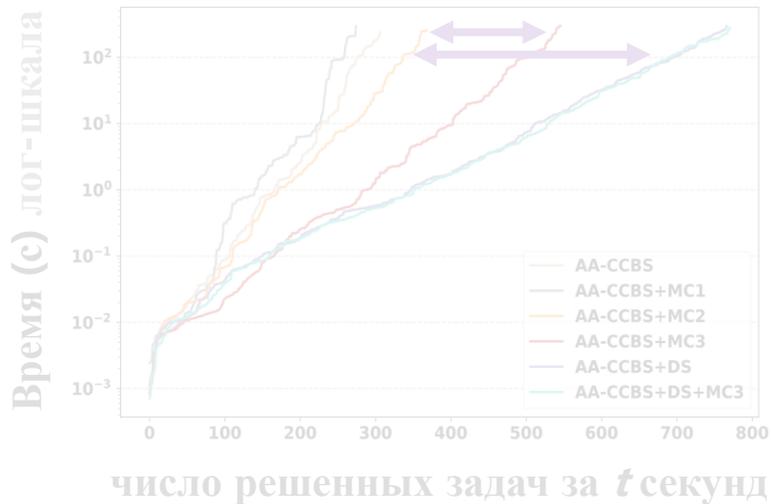
Результаты экспериментов (Focal-AA-CCBS)



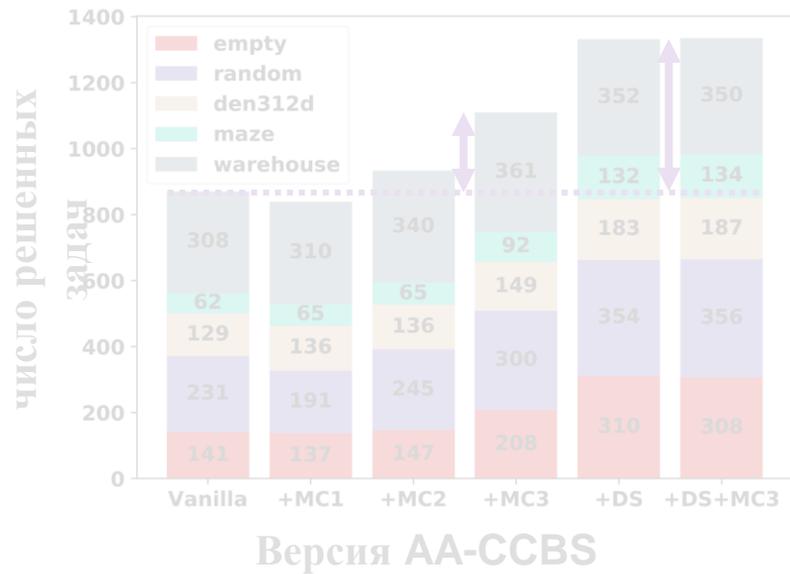
Предложенная в работе техника MC3 заметно повышает эффективность алгоритма (также как и DS)

Focal	фактор субоптимальности			
	$w = 1.0$	$w = 1.01$	$w = 1.1$	$w = 1.25$
AA-CCBS	871	1653	1929	1950
AA-CCBS+MC3	1110	2002	2437	2440
AA-CCBS+DS	1331	1661	1583	1564
AA-CCBS+DS+MC3	1335	1626	1561	1543

x3 (рост числа успешно решенных заданий)



Результаты экспериментов (Focal-AA-CCBS)

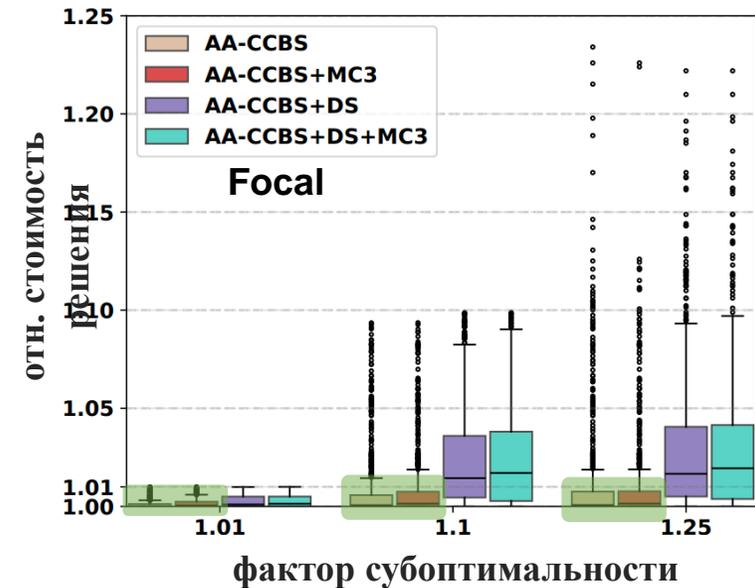
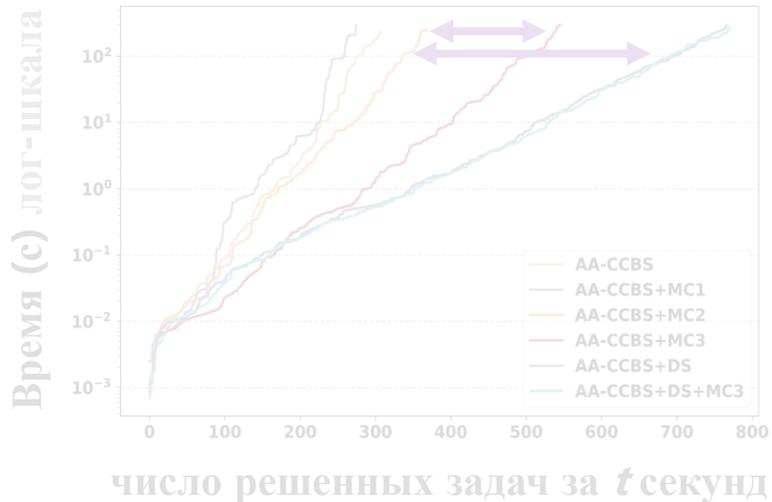


Предложенная в работе техника MC3 заметно повышает эффективность алгоритма (также как и DS)

Focal	фактор субоптимальности			
	$w = 1.0$	$w = 1.01$	$w = 1.1$	$w = 1.25$
AA-CCBS	871	1653	1929	1950
AA-CCBS+MC3	1110	2002	2437	2440
AA-CCBS+DS	1331	1661	1583	1564
AA-CCBS+DS+MC3	1335	1626	1561	1543

число решенных задач

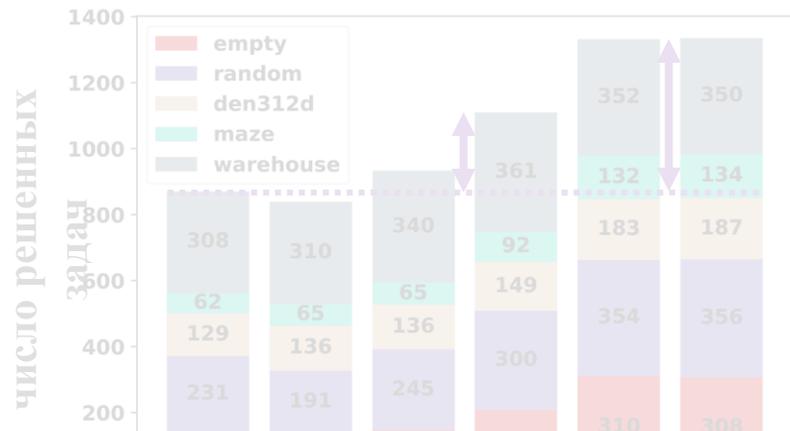
x3 (рост числа успешно решенных заданий)



Предложенная в работе техника MC3 заметно повышает эффективность алгоритма при фокусировке поиска

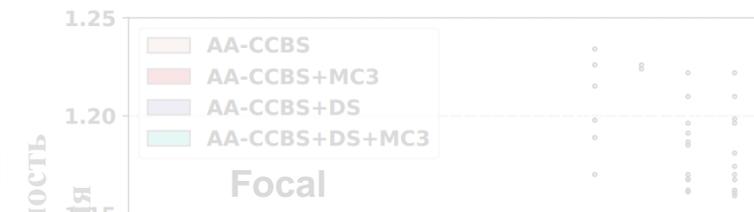
Чего нельзя сказать о DS (очень низкая эффективность)

Результаты экспериментов (Focal-AA-CCBS)

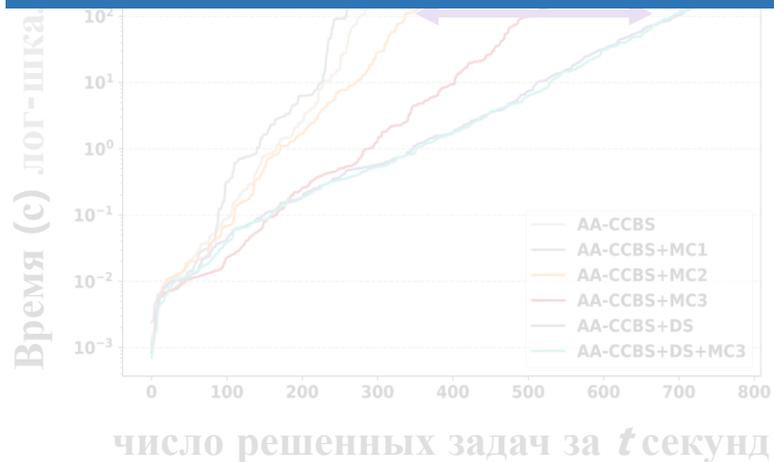


Предложенная в работе техника МСЗ заметно повышает эффективность алгоритма (также как и DS)

Focal	фактор субоптимальности			
	$w = 1.0$	$w = 1.01$	$w = 1.1$	$w = 1.25$
AA-CCBS	871	1653	1929	1950
AA-CCBS+MC3	1110	2002	2437	2440
AA-CCBS+DS	1221	1661	1582	1564



Предложенная в работе техника МСЗ заметно повышает эффективность алгоритма поиска как оптимальных, так и (в особенности) ограниченно-субоптимальных решений задачи **AA-MARF**



Предложенная в работе техника МСЗ заметно повышает эффективность алгоритма при фокусировке поиска
Чего нельзя сказать о DS (очень низкая эффективность)

Результаты экспериментов (AA-SIPP(m))

- Лучшая версия **AA-CCBS**
 - AA-CCBS+DS+MC3
- Лучшая версия **Focal-AA-CCBS**
 - Focal-AA-CCBS+MC3
- **AA-SIPP(m)**

Результаты экспериментов (AA-SIPP(m))

- Лучшая версия **AA-CCBS**
 - AA-CCBS+DS+MC3
- Лучшая версия **Focal-AA-CCBS**
 - Focal-AA-CCBS+MC3
- **AA-SIPP(m)**

	AA-CCBS	Focal-AA-CCBS	AA-SIPP(m)
empty	267	455	3 640
random	340	623	1 922
maze	120	199	1 968
den312	189	366	7 512
Total	916	1 643	15 042

число решенных задач

Результаты экспериментов (AA-SIPP(m))

- Лучшая версия **AA-CCBS**
 - AA-CCBS+DS+MC3
- Лучшая версия **Focal-AA-CCBS**
 - Focal-AA-CCBS+MC3
- **AA-SIPP(m)**

	AA-CCBS	Focal-AA-CCBS	AA-SIPP(m)
empty	267	455	3 640
random	340	623	1 922
maze	120	199	1 968
den312	189	366	7 512
Total	916	1 643	15 042

число решенных задач

x9-15

Результаты экспериментов (AA-SIPP(m))

- Лучшая версия **AA-CCBS**
 - AA-CCBS+DS+MC3
- Лучшая версия **Focal-AA-CCBS**
 - Focal-AA-CCBS+MC3
- **AA-SIPP(m)**

	AA-CCBS	Focal-AA-CCBS	AA-SIPP(m)
empty	267	455	3 640
random	340	623	1 922
maze	120	199	1 968
den312	189	366	7 512
Total	916	1 643	15 042

число решенных задач

x9-15

	AA-CCBS	Focal-AA-CCBS	AA-SIPP(m)
empty	100.0%	101.4%	104.3%
random	100.0%	101.1%	104.4%
maze	100.0%	100.4%	104.2%
den312	100.0%	100.2%	102.3%
AVG	100.0%	100.8%	103.8%

отн. стоимость
решения

Результаты экспериментов (AA-SIPP(m))

- Лучшая версия **AA-CCBS**
 - AA-CCBS+DS+MC3
- Лучшая версия **Focal-AA-CCBS**
 - Focal-AA-CCBS+MC3
- **AA-SIPP(m)**

	AA-CCBS	Focal-AA-CCBS	AA-SIPP(m)
empty	267	455	3 640
random	340	623	1 922
maze	120	199	1 968
den312	189	366	7 512
Total	916	1 643	15 042

число решенных задач

x9-15

	AA-CCBS	Focal-AA-CCBS	AA-SIPP(m)
empty	100.0%	101.4%	104.3%
random	100.0%	101.1%	104.4%
maze	100.0%	100.4%	104.2%
den312	100.0%	100.2%	102.3%
AVG	100.0%	100.8%	103.8%

отн. стоимость

≤5% (превышение)

Результаты экспериментов (AA-SIPP(m))

- Лучшая версия **AA-CCBS**
 - AA-CCBS+DS+MC3
- Лучшая версия **Focal-AA-CCBS**
 - Focal-AA-CCBS+MC3
- **AA-SIPP(m)**

Предложенный в работе алгоритм AA-SIPP(m) на порядок повышает эффективность поиска решений задачи **AA-MAPF**. При этом стоимость решений возрастает незначительно (менее, чем на 5% по сравнению с AA-CCBS).

	AA-CCBS	Focal-AA-CCBS	AA-SIPP(m)
empty	267	455	3 640
random	340	623	1 922
maze	120	199	1 968
den312	189	366	7 512
Total	916	1 643	15 042

число решенных задач

x9-15

	AA-CCBS	Focal-AA-CCBS	AA-SIPP(m)
empty	100.0%	101.4%	104.3%
random	100.0%	101.1%	104.4%
maze	100.0%	100.4%	104.2%
den312	100.0%	100.2%	102.3%
AVG	100.0%	100.8%	103.8%

отн. стоимость

≤5% (превышение)

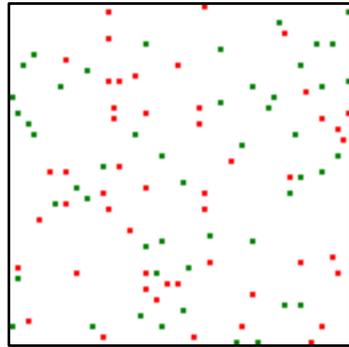
Экспериментальные исследования (серия 2)

AA-MAPF

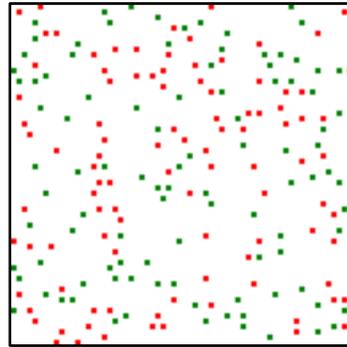
- AA-SIPP(m)

MAPF (4-х связный ГРД)

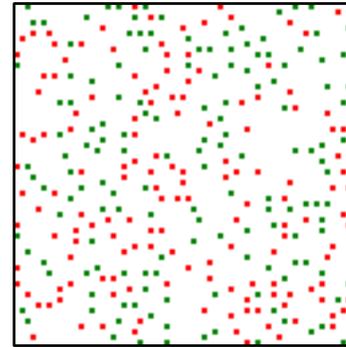
- SIPP(m) (субопт.)
- ICBS (опт.) [1]
- ECBS (субопт.) [2]



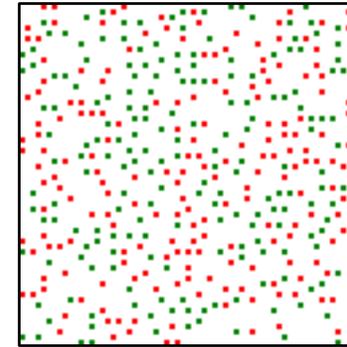
50 агентов



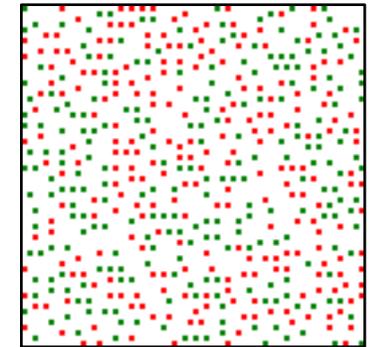
100 агентов



150 агентов



200 агентов



250 агентов

- ГРД empty (64 × 64)
- число агентов [50, 250]
 - 500 заданий
 - Случайная расстановка (s, g)
- ограничение по времени
300 с.

[1] Boyarski E., Felner A., Stern R., Sharon G., Tolpin D., Betzalel O., Shimony E. ICBS: Improved conflict-based search algorithm for multi-agent pathfinding // IJCAI 2015. pp. 740-746.

[2] Barer M., Sharon G., Stern R., Felner A. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem // SoCS 2014 pp. 19-27.

Результаты

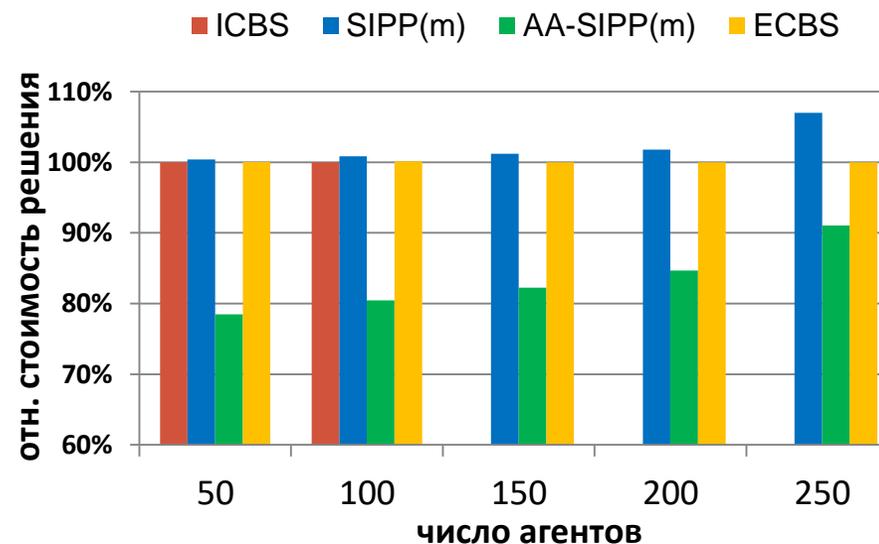
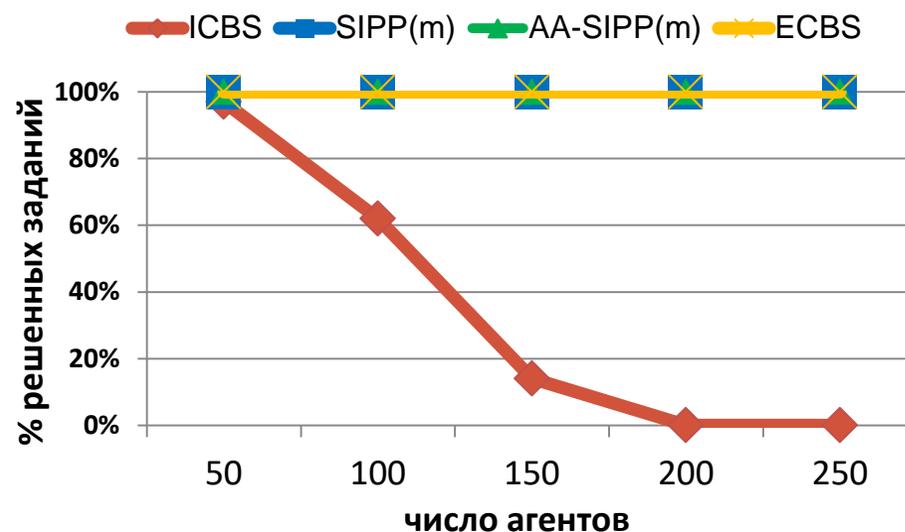
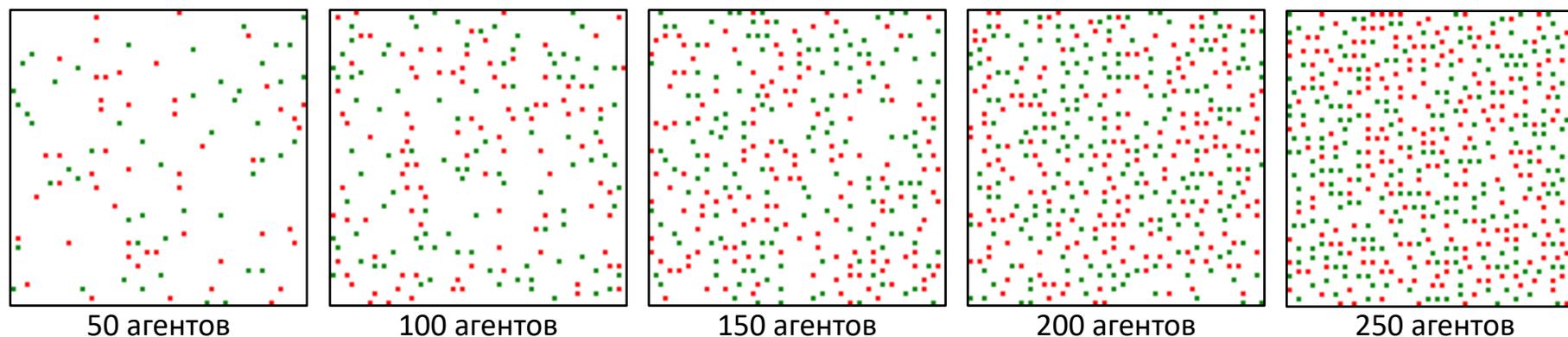
AA-MAPF

- AA-SIPP(m)

MAPF (4-х связный ГРД)

- SIPP(m) (субопт.)
- ICBS (опт.) [1]
- ECBS (субопт.) [2]

- ГРД empty (64 × 64)
- число агентов [50, 250]
 - 500 заданий
 - Случайная расстановка (s, g)
- ограничение по времени 300 с.



[1] Boyarski E., Felner A., Stern R., Sharon G., Tolpin D., Betzalel O., Shimony E. ICBS: Improved conflict-based search algorithm for multi-agent pathfinding // IJCAI 2015. pp. 740-746.

[2] Barer M., Sharon G., Stern R., Felner A. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem // SoCS 2014 pp. 19-27.

Результаты

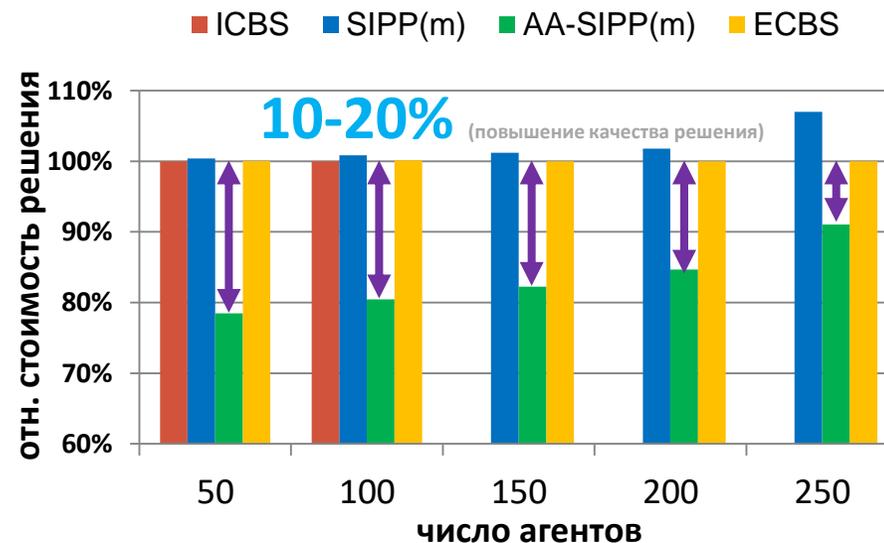
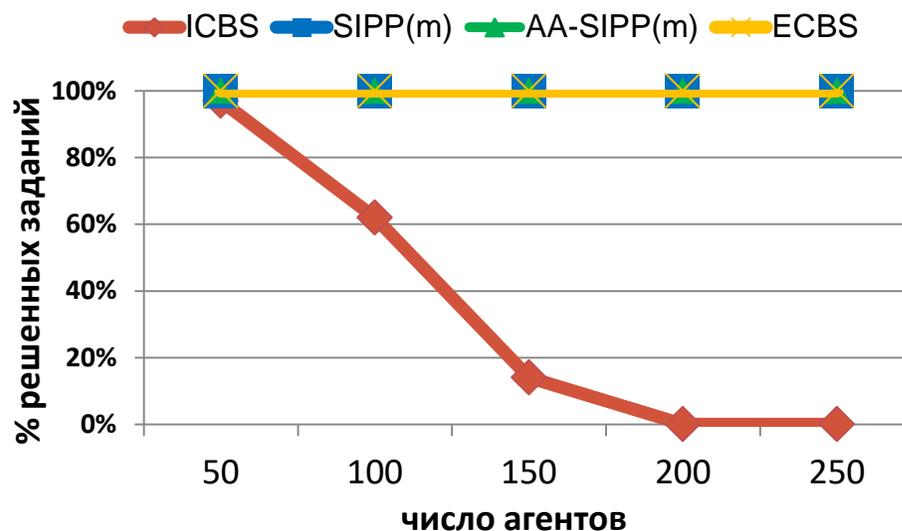
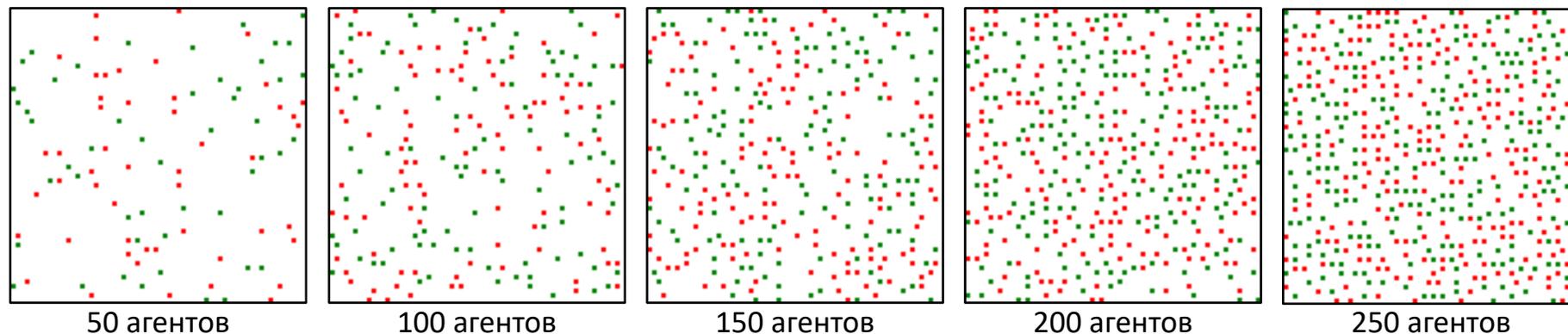
AA-MAPF

- AA-SIPP(m)

MAPF (4-х связный ГРД)

- SIPP(m) (субопт.)
- ICBS (опт.) [1]
- ECBS (субопт.) [2]

- ГРД empty (64 × 64)
- число агентов [50, 250]
 - 500 заданий
 - Случайная расстановка (s, g)
- ограничение по времени 300 с.



[1] Boyarski E., Felner A., Stern R., Sharon G., Tolpin D., Betzalel O., Shimony E. ICBS: Improved conflict-based search algorithm for multi-agent pathfinding // IJCAI 2015. pp. 740-746.

[2] Barer M., Sharon G., Stern R., Felner A. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem // SoCS 2014 pp. 19-27.

Результаты

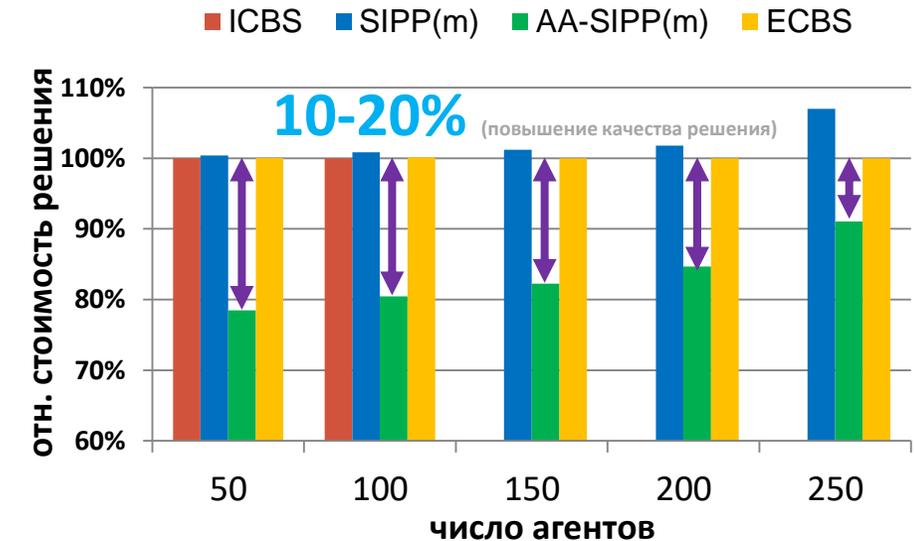
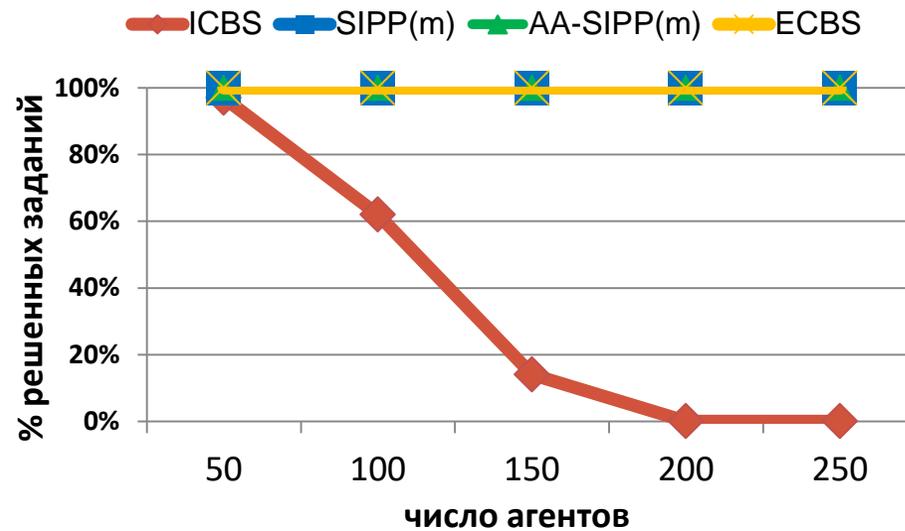
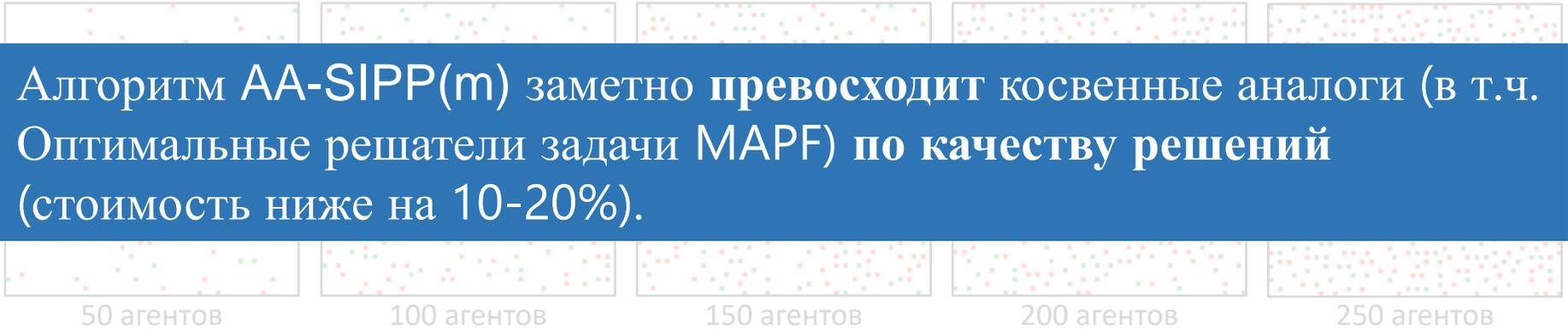
AA-MAPF

- **AA-SIPP(m)**

MAPF (4-х связный ГРД)

- **SIPP(m)** (субопт.)
- **ICBS** (опт.) [1]
- **ECBS** (субопт.) [2]

- ГРД empty (64 × 64)
- число агентов [50, 250]
 - 500 заданий
 - Случайная расстановка (s, g)
- ограничение по времени 300 с.



[1] Boyarski E., Felner A., Stern R., Sharon G., Tolpin D., Betzalel O., Shimony E. ICBS: Improved conflict-based search algorithm for multi-agent pathfinding // IJCAI 2015. pp. 740-746.

[2] Barer M., Sharon G., Stern R., Felner A. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem // SoCS 2014 pp. 19-27.

Результаты

AA-MAPF

- **AA-SIPP(m)**

MAPF (4-х связный ГРД)

- **SIPP(m)** (субопт.)
- **ICBS** (опт.) [1]
- **ECBS** (субопт.) [2]

- ГРД empty (64 × 64)
- число агентов [50, 250]
 - 500 заданий
 - Случайная расстановка (s, g)
- ограничение по времени 300 с.

Алгоритм AA-SIPP(m) заметно превосходит косвенные аналоги (в т.ч. Оптимальные решатели задачи MAPF) по качеству решений (стоимость ниже на 10-20%).

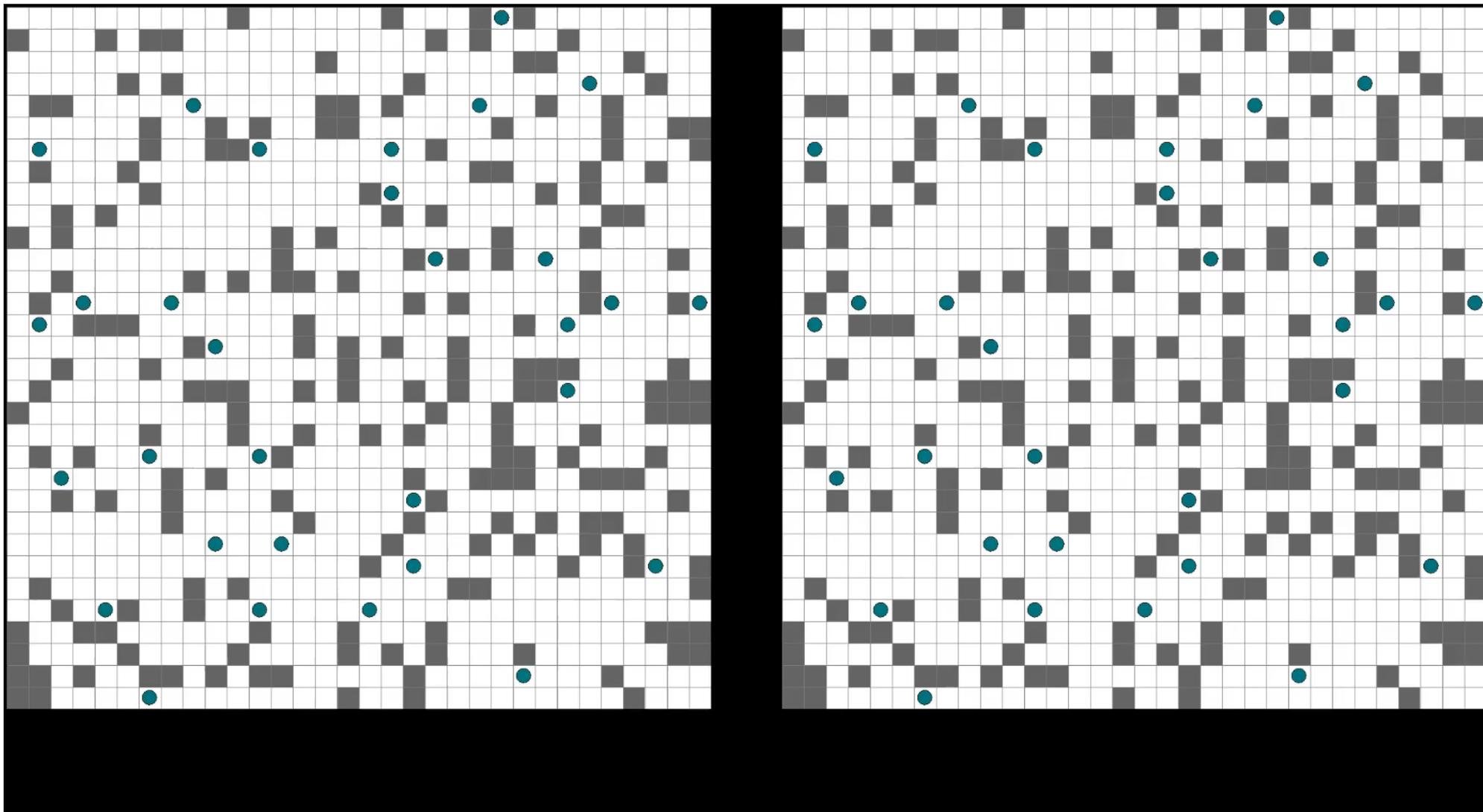


Аналогичные результаты получены при экспериментах на картах с препятствиями

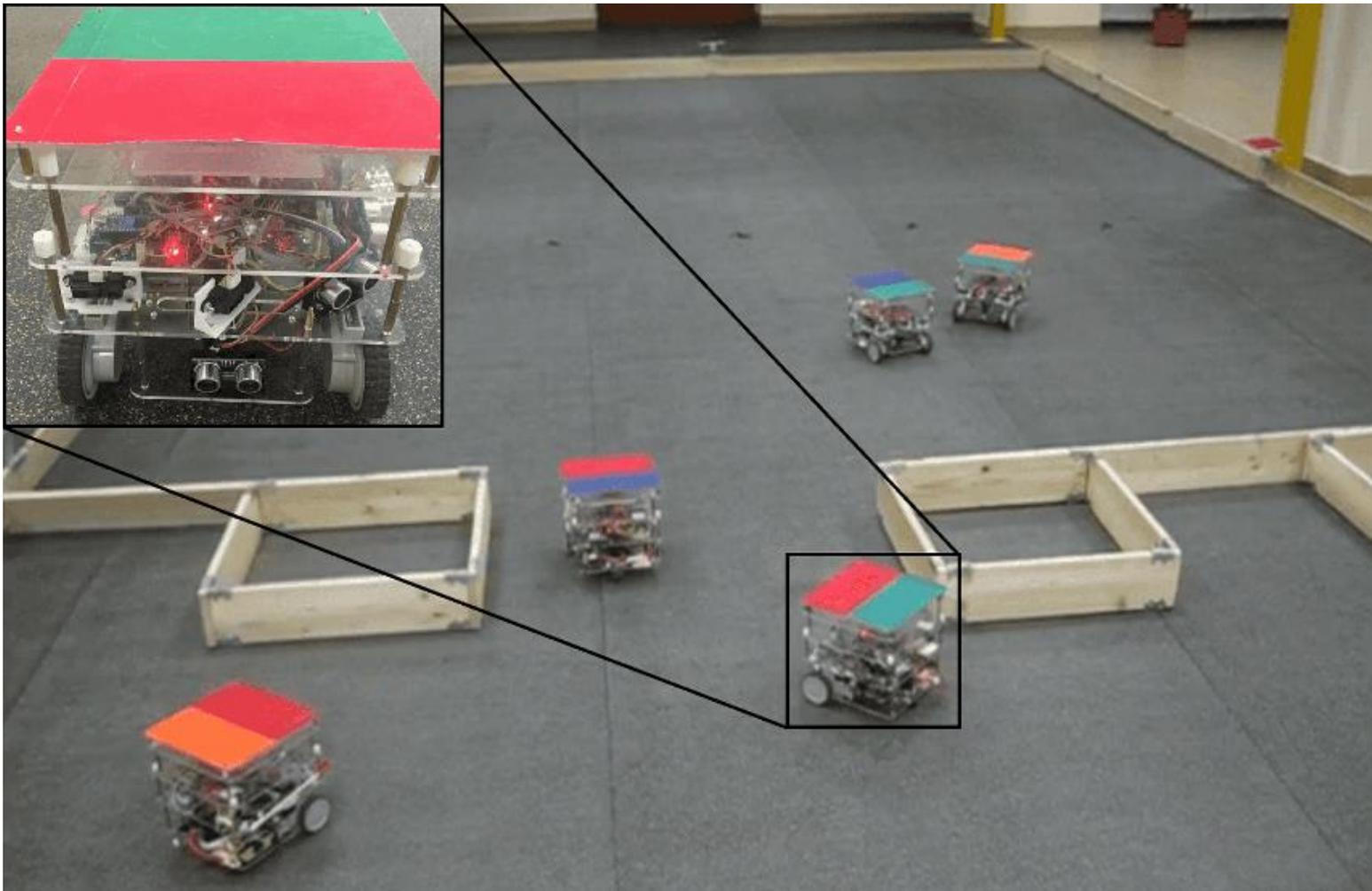
[1] Boyarski E., Felner A., Stern R., Sharon G., Tolpin D., Betzalel O., Shimony E. ICBS: Improved conflict-based search algorithm for multi-agent pathfinding // IJCAI 2015. pp. 740-746.

[2] Barer M., Sharon G., Stern R., Felner A. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem // SoCS 2014 pp. 19-27.

Демонстрация решений задач MAPF и AA-MAPF



Эксперименты на реальных роботах



Робототехнический полигон НИЦ «Курчатовский институт»

Полигон

- 4.8×6 м.
- глобальная система локализации

Роботы YARP-2

- Колесная платформа
- Дифференциальный привод
- Размер 21×21 см.
- 0.1 м/с.
- локальная система управления

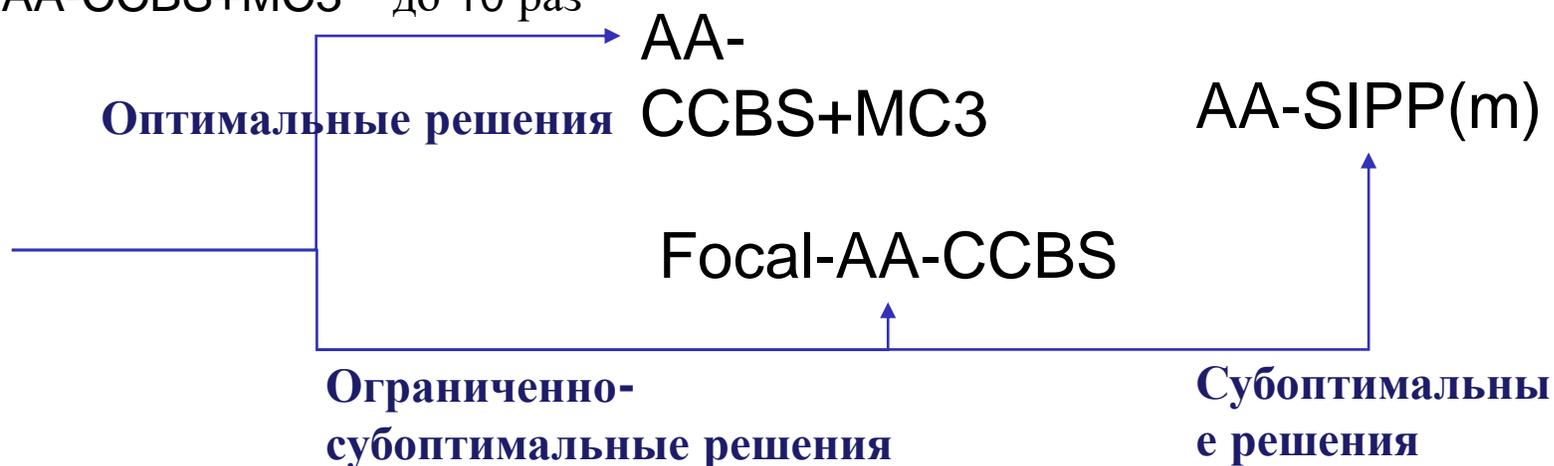
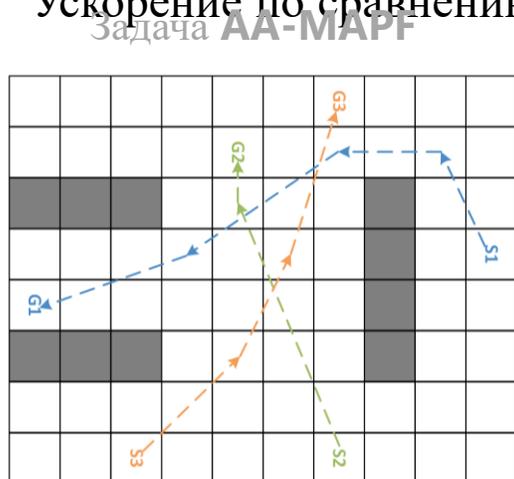
Все эксперименты
завершены успешно

Эксперименты на реальных роботах

AAt-SIPP(m) vs ORCA
Warehouse 27x35 grid
with 64 agents

Выводы (задача AA-MAPF)

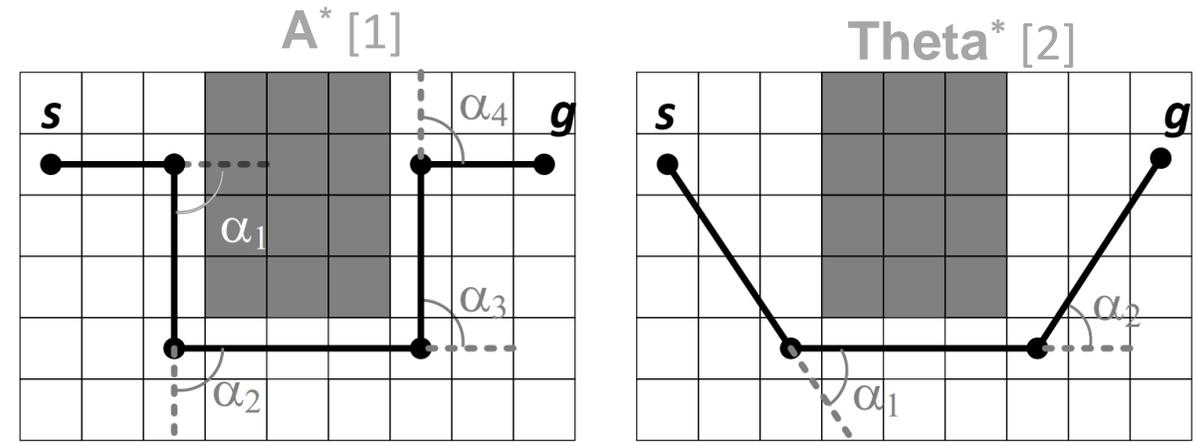
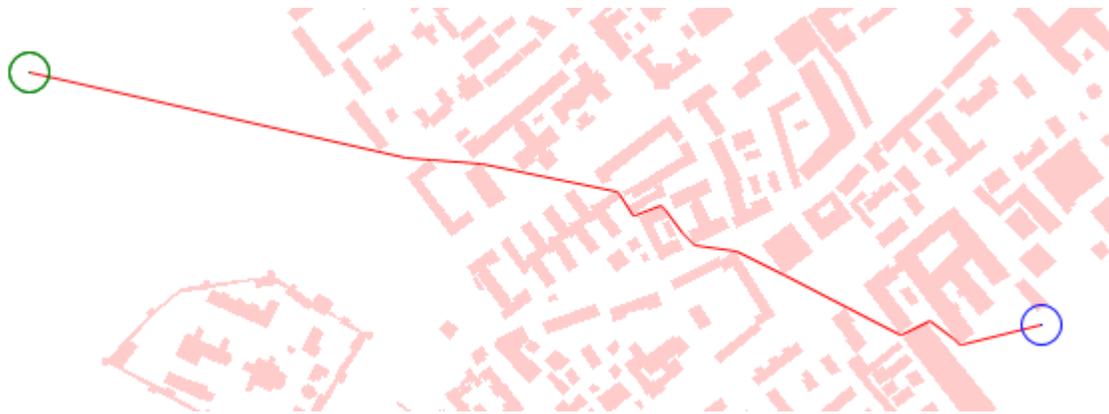
- Впервые предложен эффективный (в вычислительном смысле) алгоритм получения оптимальных решений задачи **AA-MAPF** – AA-CCBS+MC3
 - Ускорение по сравнению с базовым вариантом конфликтно-ориентированного поиска – до 2 раз.
- Предложен эффективный алгоритм получения ограниченно-субоптимальных решений задачи **AA-MAPF** – AA-Focal-SIPP
 - Ускорение по сравнению с AA-CCBS+MC3 – в 2 раза (в среднем)
- Предложен эффективный алгоритм получения субоптимальных решения задачи **AA-MAPF** – AA-SIPP(m)
 - Ускорение по сравнению с AA-CCBS+MC3 – до 10 раз



Поиск путей с геометрическими ограничениями на ГРД

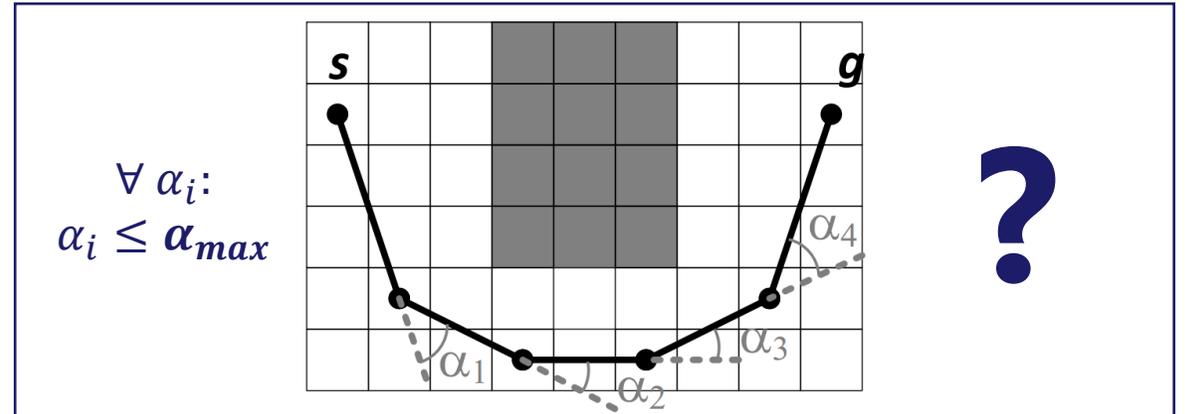
Практико-ориентированный сценарий: косвенный учет кинематических ограничений при движении мобильного робота (например, маловысотный полет беспилотного летательного аппарата с постоянной скоростью)

Опасный (опорный) путь, содержащий резкие повороты



[1] 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, 4(2), 1968. pp. 100-107.

[2] Nash, A., Daniel, K., Koenig, S., Felner, A. Theta*: Any-Angle Path Planning on Grids // AAAI 2007. p. 1177-1186.



Задача AC-PF (angle-constrained pathfinding)

$$\text{AC-PF} = (\mathcal{G}, v_{start}, v_{goal}, los, \alpha_{max})$$

$\mathcal{G} = (V, E)$ – ГРД

$v_{start} \in V$ – начальная вершина

$v_{goal} \in V$ – целевая вершина

$los: V \times V \rightarrow \{\text{True}, \text{False}\}$ – допустимость перехода

$\alpha_{max} \in [0^\circ, 180^\circ]$ – макс. угол отклонения

Секция

$e = (u, v)$ – упорядоченная пара вершин не обязательно смежных

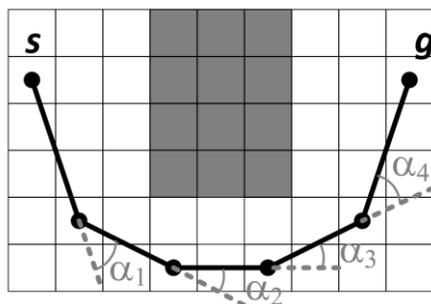
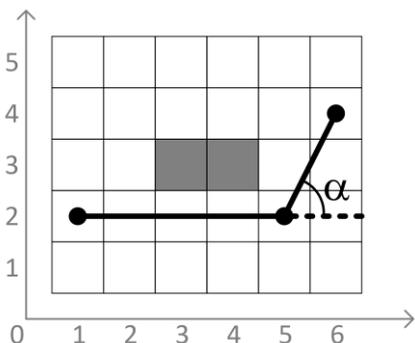
$los(e) = true$ → допустимая секция

$e = (u, v), e' = (u', v'), v = u' \rightarrow$ смежные секции

$\alpha(e, e')$ – угол отклонения (поворота) e' от e

модуль угла между векторами

$(v_x - u_x, v_y - u_y)$ и $(v'_x - v_x, v'_y - v_y)$



Путь

$$\pi(v_{start}, v_{goal}) = (e_1, e_2, \dots, e_n)$$

т.ч.:

$$source(e_1) = v_{start}$$

$$target(e_n) = v_{goal}$$

$$\forall i \in [1, n-1] : target(e_i) = source(e_{i+1})$$

$$\forall i \in [1, n] : los(source(e_i), target(e_i)) = true$$

СТОИМОСТЬ = ДЛИНА

$$cost(\pi) = \sum_{i=1}^n len(e_i)$$

Максимальный угол поворота в пути

$$\alpha_m(\pi) = \max_{i=1..n} \{\alpha(e_1, e_2), \dots, \alpha(e_{n-1}, e_n)\}$$

Задача (найти путь, удовлетворяющий ограничению)

$$(\mathcal{G}, v_s, v_g, los, \alpha_{max}) \rightarrow \pi(v_s, v_g) : \alpha_m(\pi) \leq \alpha_{max}$$

Сужение класса рассматриваемых решений

- У каждой вершины – $O(N)$ потенциальных потомков
 - для проверки каждого требуется вызов los
- Состояние поиска $n=(v, parent(n))$
 - Дубликатов (почти) нет

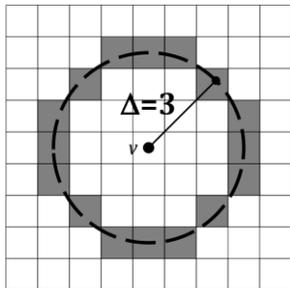


Пространство поиска чрезмерно велико

Фокусировка на отдельном классе решений

Δ -решения

$CIRCLE(v, \Delta)$ – множество вершин ГРД, аппроксимирующих окружность радиуса Δ с центром в v

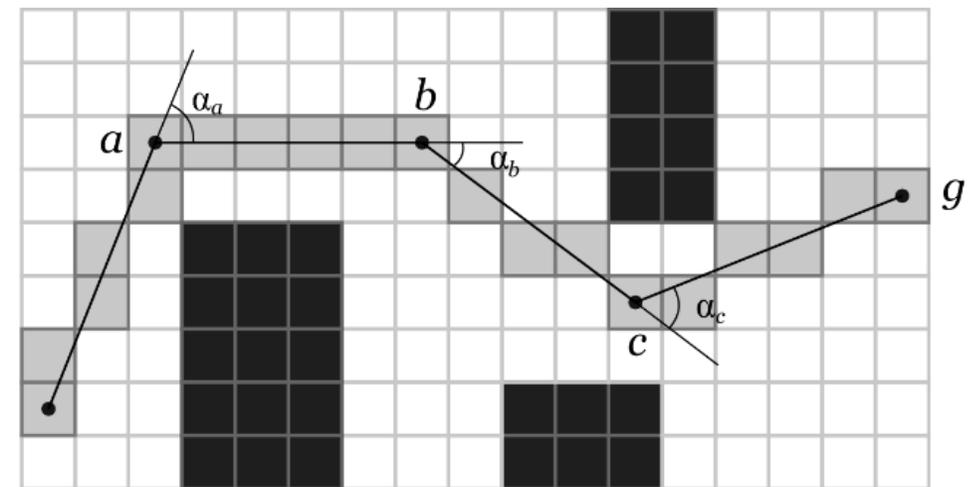


Построение с помощью алгоритма средней точки (midpoint algorithm) [1]

[1] Pitteway M.L. Algorithms of conic generation // Fundamental Algorithms for Computer Graphics, 1985. pp. 219-237.

Δ -секция – секция $e = (u, v)$, т.ч. $v \in CIRCLE(u, \Delta)$, $\Delta \in \mathbb{N}$

Δ -путь – путь, все секции которого, за исключением лишь может быть последней являются Δ -секциями



Δ -путь ($\Delta=5$)

Алгоритм LIAN (limited angle)

Алгоритм LIAN($\mathcal{G}, v_{start}, v_{goal}, \alpha_{max}, \Delta, h, los$):

Входные данные: Граф \mathcal{G} , начальная и целевая вершины v_{start} , v_{goal} , ограничение на угол отклонения α_{max} , длина секции Δ , монотонная эвристическая функция h , функция, определяющая возможность перехода, los

Выходные данные: Путь π

```
1  $n_{start} \leftarrow \text{GenerateSearchNode}(v_{start}, \text{null})$ 
2  $g(n_{start}) \leftarrow 0; \text{parent}(n_{start}) \leftarrow \text{null}$ 
3  $OPEN \leftarrow \{n_{start}\}; CLOSED \leftarrow \emptyset$ 
4 while  $OPEN \neq \emptyset$  do
5    $n \leftarrow \arg \min_{n \in OPEN} (f(n) = g(n) + h(n))$ 
6    $OPEN \leftarrow OPEN \setminus \{n\}$ 
7    $CLOSED \leftarrow CLOSED \cup \{n\}$ 
8   if  $n.v = v_{goal}$  then
9     return  $\text{ReconstructPath}(n)$ 
10   $SUCC \leftarrow \text{GenerateLIANSuccesors}(n, \alpha_{max}, \Delta, v_{goal}, los)$ 
11  foreach  $n_{succ} \in SUCC$  do
12    if  $\text{InClosed}(n_{succ})$  then
13      continue
14     $n_{dubl} \leftarrow \text{FindNode}(n_{succ}, OPEN)$ 
15    if  $n_{dubl} = \text{null}$  then
16       $OPEN \leftarrow OPEN \cup \{n_{succ}\}$ 
17    else if  $g(n_{dubl}) > g(n_{succ})$  then
18       $OPEN \leftarrow (OPEN \setminus \{n_{dubl}\}) \cup \{n_{succ}\}$ 
19 return failure
```

- Систематический поиск от старта к цели
- Эвристика – эвклидово расстояние
- Состояние поиска $n = (v, \text{parent}(n))$

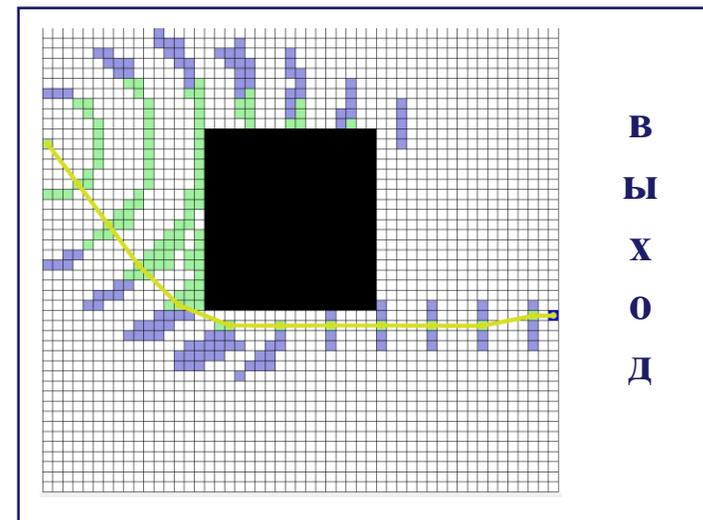
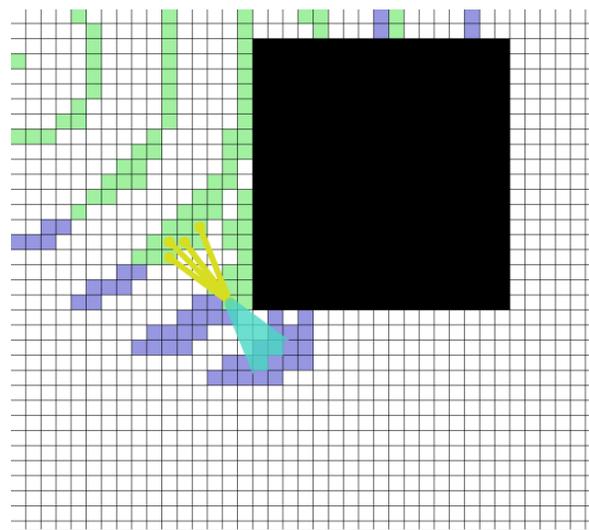
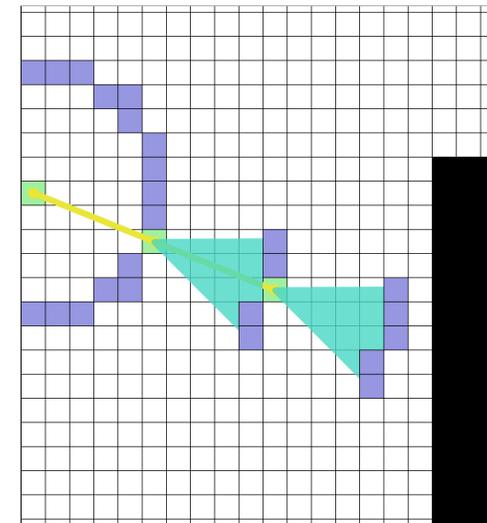
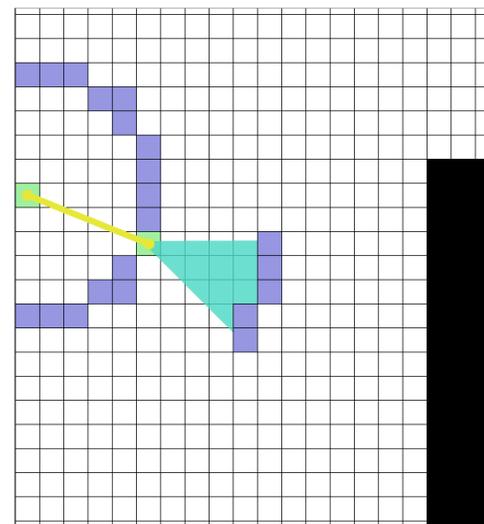
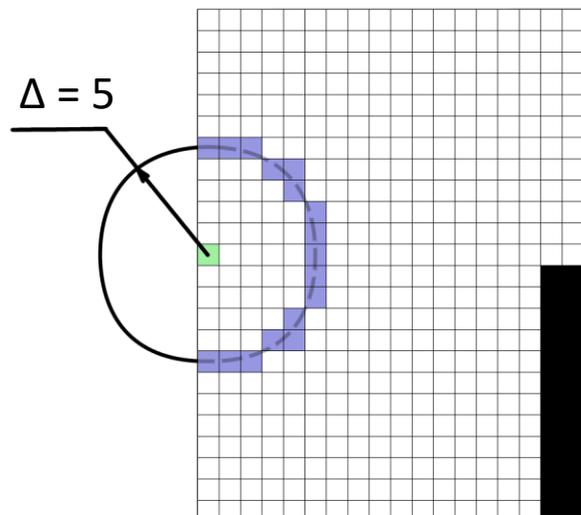
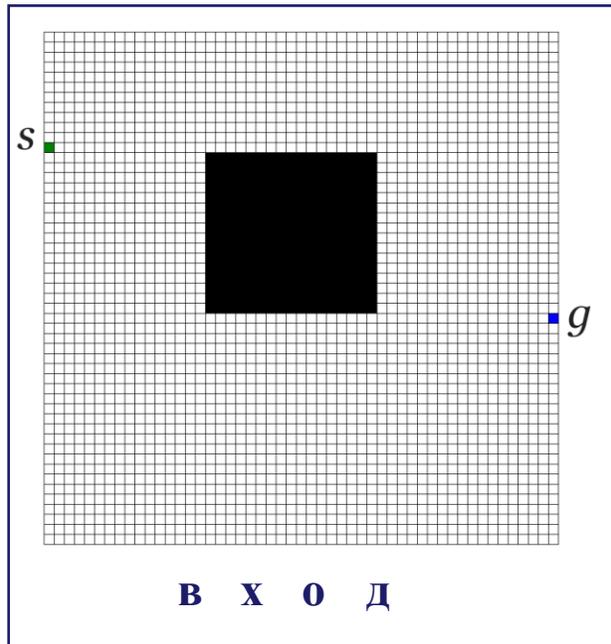
Процедура GenerateLIANSuccesors($n, \alpha_{max}, \Delta, v_{goal}, los$):

Входные данные: Состояние поиска $n = (v, \text{parent}(n))$, ограничение на угол отклонения α_{max} , длина секции Δ , целевая вершина v_{goal} , функция, определяющая возможность перехода, los

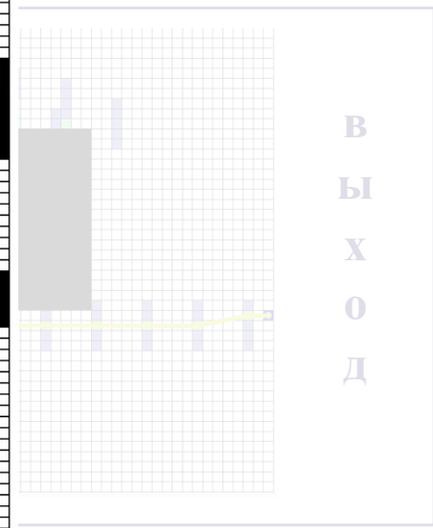
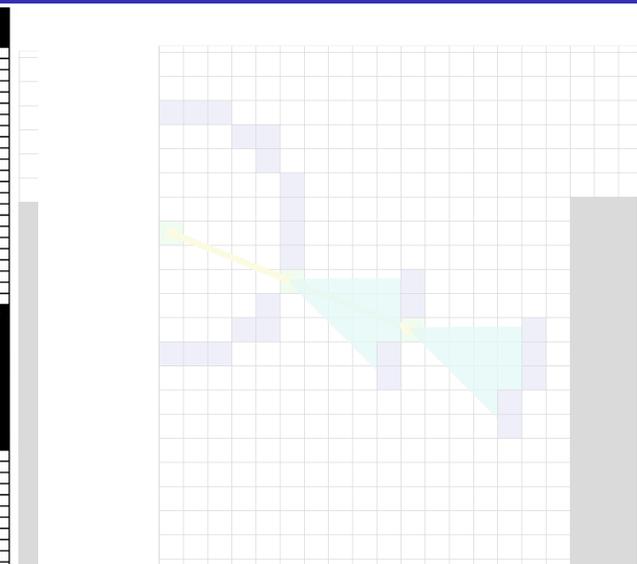
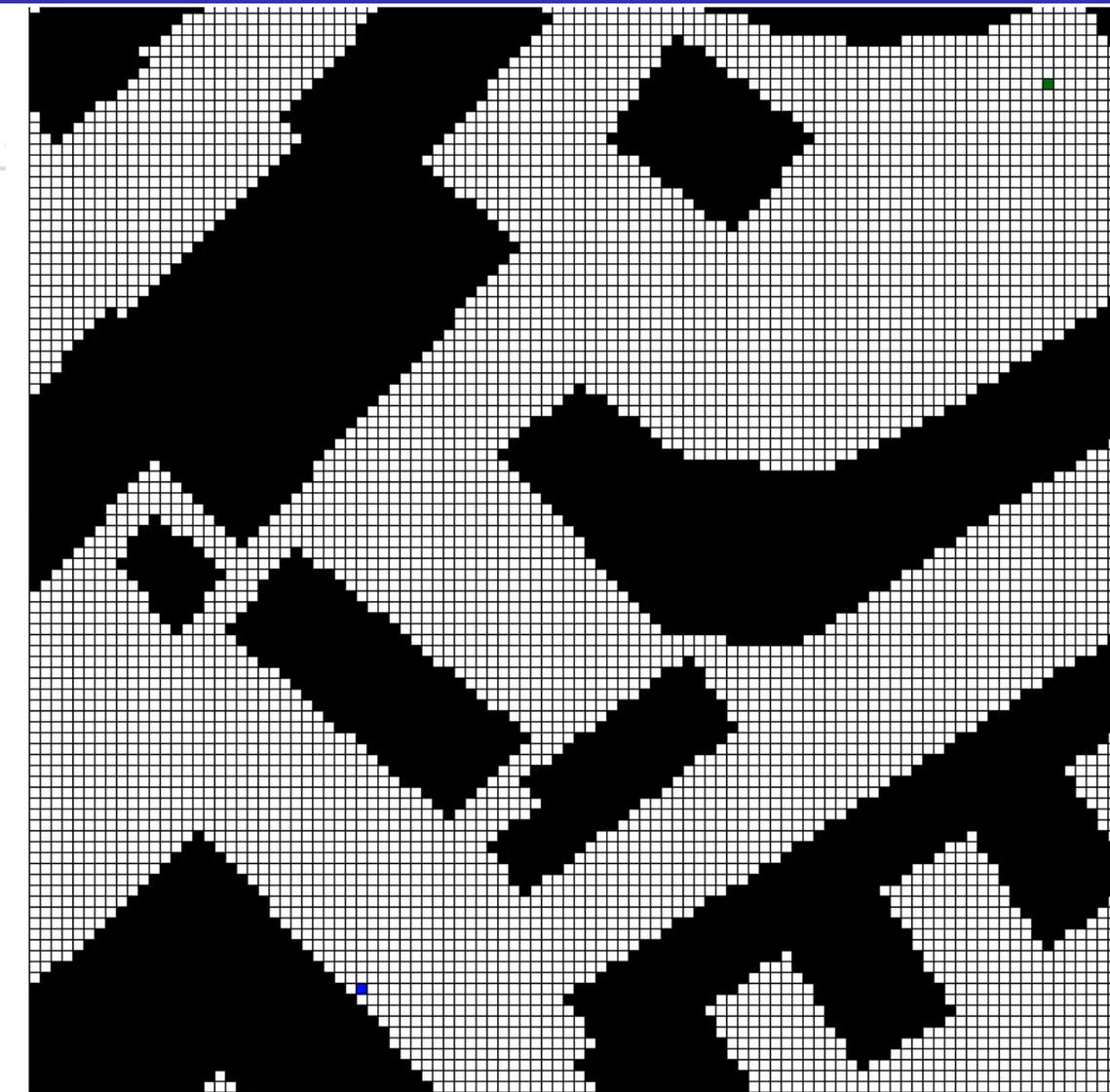
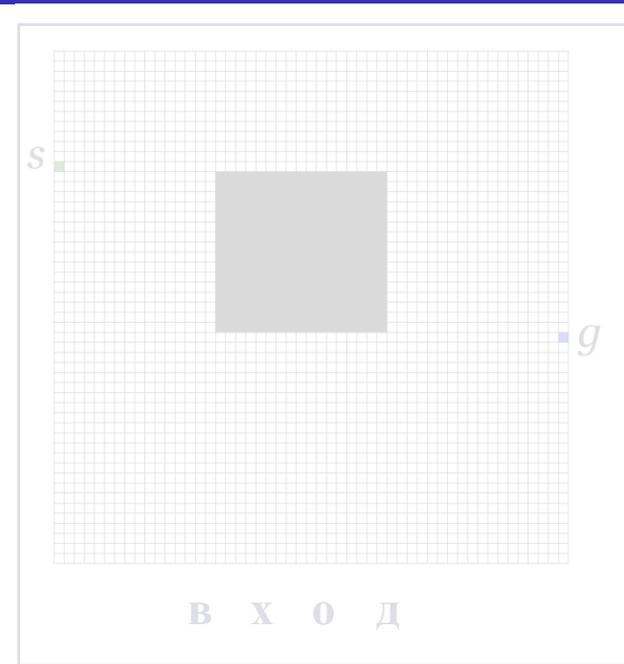
Выходные данные: Множество состояний-потомков $SUCC$ для состояния n

```
1  $SUCC \leftarrow \emptyset; SUCC_V \leftarrow \text{CIRCLE}(n.v, \Delta)$ 
2 if  $\|(n.v, v_{goal})\| < \Delta$  then
3    $SUCC_V \leftarrow SUCC_V \cup \{v_{goal}\}$ 
4 foreach  $v_{circ} \in SUCC_V$  do
5   if  $los(n.v, v_{circ}) = \text{false}$  then
6     continue
7    $e_1 = (\text{parent}(n).v, n.v); e_2 = (n.v, v_{circ})$ 
8   if  $\alpha(e_1, e_2) > \alpha_{max}$  then
9     continue
10   $n_{succ} \leftarrow \text{GenerateSearchNode}(v_{circ}, n)$ 
11   $g(n_{succ}) \leftarrow g(n) + \|e_2\|$ 
12   $\text{parent}(n_{succ}) \leftarrow n$ 
13   $SUCC \leftarrow SUCC \cup \{n_{succ}\}$ 
14 return  $SUCC$ 
```

Алгоритм LIAN (Пример работы. $\alpha=25, \Delta=5$)



Алгоритм LIAN (Пример работы. $\alpha=25, \Delta=5$)



В
Ы
ХО
Д

Алгоритм LIAN (теоретические гарантии)

P^+ – множество задач AC-PF, имеющих решение

P^Δ – задачи AC-PF, т.ч. хотя бы одно их решение есть Δ -решение

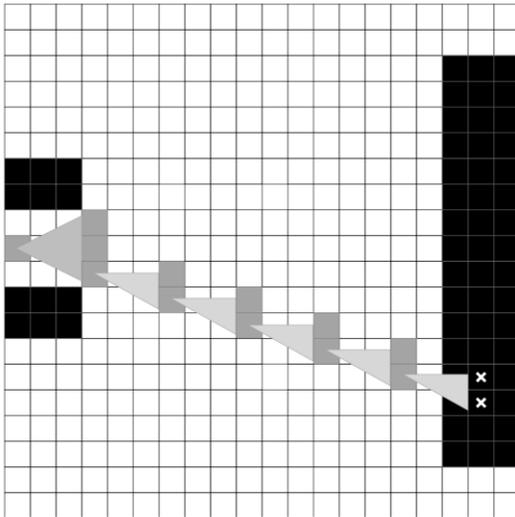
Теорема 5. Алгоритм LIAN является полным и оптимальным относительно класса задач P^Δ , т.е. гарантирует отыскание Δ -решения, если оно существует, и гарантирует, что стоимость найденного решения не превосходит стоимость любого другого Δ -решения (т.е. гарантирует отыскание Δ -пути минимальной длины).

Идея доказательства: Следствие из того, что на каждом шаге генерируются все допустимые состояния-потомки, и для исследования пространства состояний используется классическая стратегия алгоритма A^* (обладающего всеми нужными гарантиями).

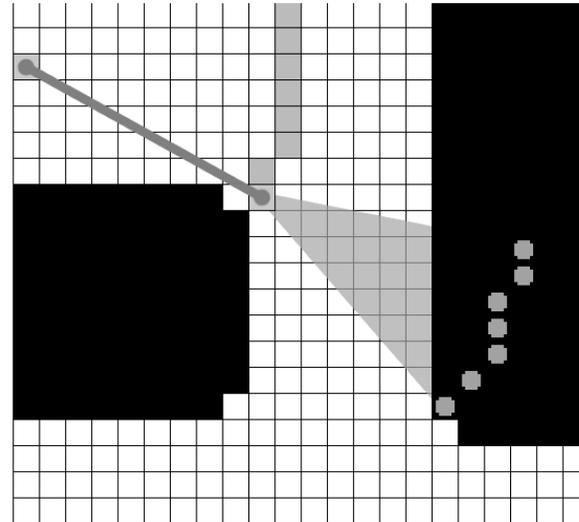
Алгоритм LIAN (недостатки и пути их решения)

Необходимость фиксации параметра Δ

Низкое значение Δ

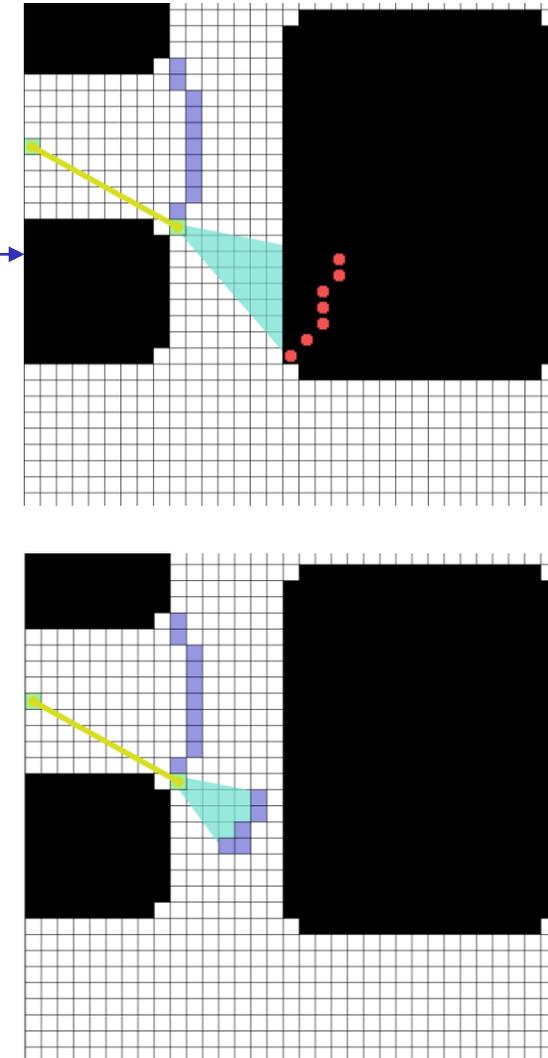


Высокое значение Δ



Решение – динамическая подстройка Δ по ходу работы

- Инициализация Δ максимальным (разумным) значением
 - Быстрое продвижение к цели
- Уменьшение Δ в два раза, если не удастся создать потомков
 - маневрирование в узких местах
- Если при текущем Δ совершено 2 перехода подряд – увеличение Δ в два раза
 - Быстрое продвижение к цели



Алгоритм D-LIAN

Алгоритм D-LIAN($\mathcal{G}, v_{start}, v_{goal}, \alpha_{max}, \Delta_{min}, \Delta_{max}, h, los$):

Входные данные: Граф \mathcal{G} , начальная и целевая вершины v_{start}, v_{goal} , ограничение на угол отклонения α_{max} , минимальная и максимальная длина секции $\Delta_{min}, \Delta_{max}$, монотонная эвристическая функция h , функция, определяющая возможность перехода, los

Выходные данные: Путь π

```
1  $n_{start} \leftarrow \text{GenerateSearchNode}(v_{start}, \text{null})$ 
2  $\Delta(n_{start}) \leftarrow \Delta_{max}; g(n_{start}) \leftarrow 0; \text{parent}(n_{start}) \leftarrow \text{null}$ 
3  $OPEN \leftarrow \{n_{start}\}; CLOSED \leftarrow \emptyset$ 
4 while  $OPEN \neq \emptyset$  do
5    $n \leftarrow \text{arg min}_{n \in OPEN}(f(n) = g(n) + h(n))$ 
6    $OPEN \leftarrow OPEN \setminus \{n\}$ 
7   if  $n.v = v_{goal}$  then
8     return  $\text{ReconstructPath}(n)$ 
9    $SUCC \leftarrow \text{GenerateDLIANSuccesors}(n, \alpha_{max}, \Delta_{max}, v_{goal}, los)$ 
10  if  $SUCC = \emptyset$  and  $\Delta(n) > \Delta_{min}$  then
11     $\Delta(n) \leftarrow \Delta(n)/2$ 
12     $OPEN \leftarrow OPEN \cup \{n\}$ 
13    continue
14   $\text{UpdateOPENandCLOSED}()$ 
15   $CLOSED \leftarrow CLOSED \cup \{n\}$ 
16 return  $\text{failure}$ 
```

Уменьшение Δ

Процедура $\text{GenerateDLIANSuccesors}(n, \alpha_{max}, \Delta_{max}, v_{goal}, los)$:

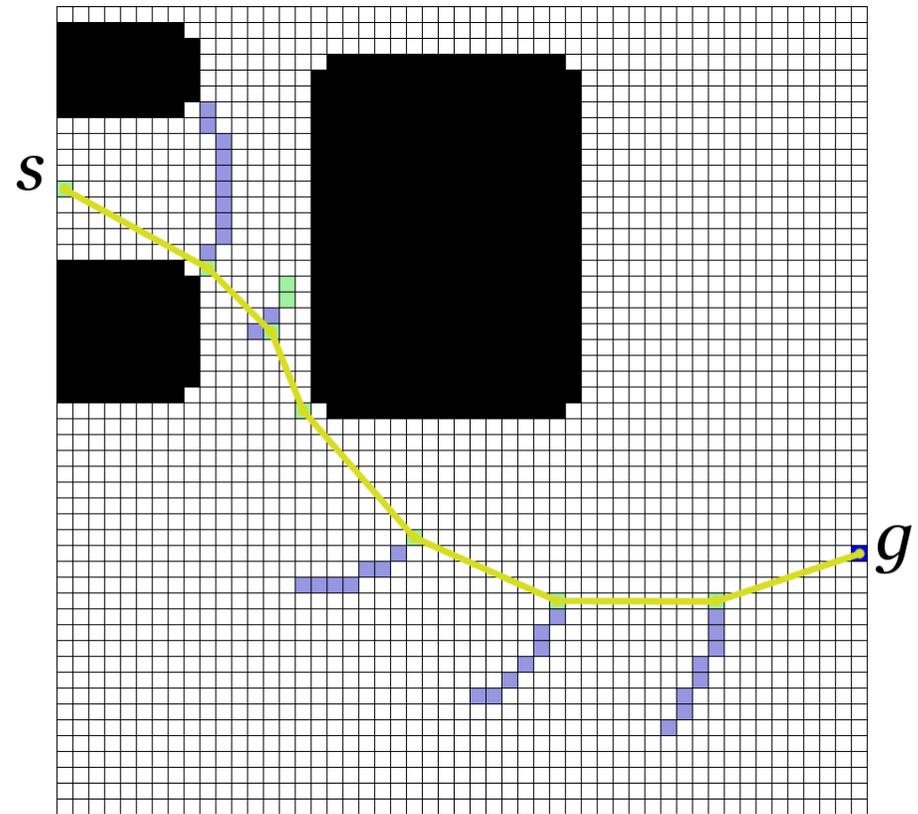
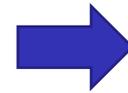
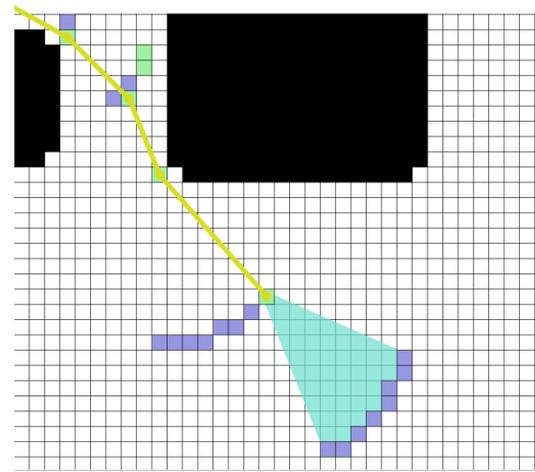
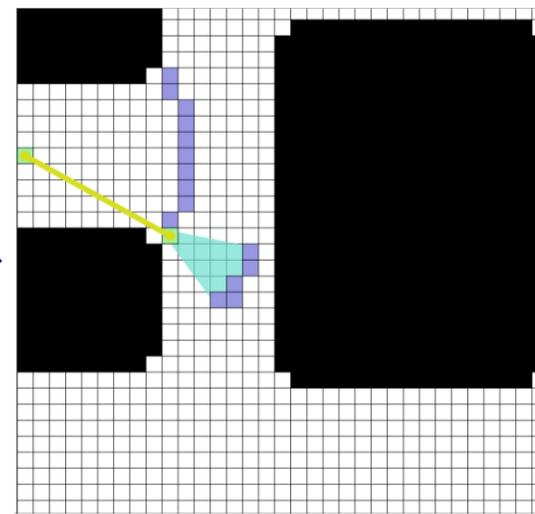
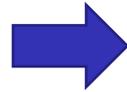
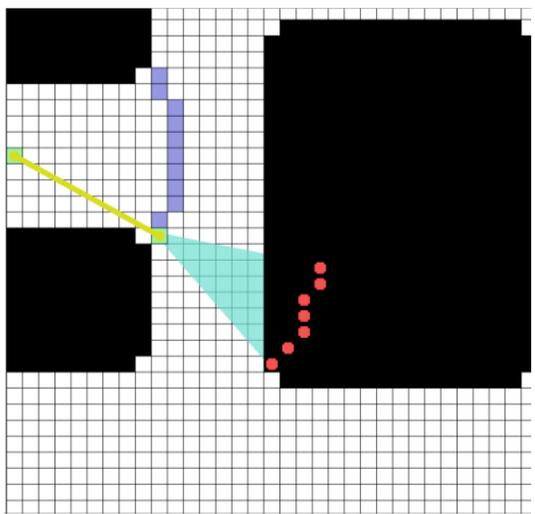
Входные данные: Состояние поиска $n = (v, \text{parent}(n))$, ограничение на угол отклонения α_{max} , максимальная длина секции Δ_{max} , целевая вершина v_{goal} , функция, определяющая возможность перехода, los

Выходные данные: Множество состояний-потомков $SUCC$ для состояния n

```
1  $SUCC \leftarrow \emptyset; SUCC_V \leftarrow \text{CIRCLE}(n.v, \Delta(n))$ 
2 if  $\|(n.v, v_{goal})\| < \Delta(n)$  then
3    $SUCC_V \leftarrow SUCC_V \cup \{v_{goal}\}$ 
4 foreach  $v_{circ} \in SUCC_V$  do
5    $e_1 = (\text{parent}(n).v, n.v); e_2 = (n.v, v_{circ})$ 
6   if  $los(n.v, v_{circ}) = \text{false}$  or  $\alpha(e_1, e_2) > \alpha_{max}$  then
7     continue
8    $n_{succ} \leftarrow \text{GenerateSearchNode}(v_{circ}, n)$ 
9    $g(n_{succ}) \leftarrow g(n) + \|e_2\|$ 
10  if  $\Delta(n) = \Delta(\text{parent}(n))$  and  $\Delta(n) < \Delta_{max}$  then
11     $\Delta(n_{succ}) \leftarrow 2 \cdot \Delta(n)$ 
12  else
13     $\Delta(n_{succ}) \leftarrow \Delta(n)$ 
14     $\text{parent}(n_{succ}) \leftarrow n$ 
15     $SUCC \leftarrow SUCC \cup \{n_{succ}\}$ 
16 return  $SUCC$ 
```

Увеличение Δ

Алгоритм D-LIAN



Алгоритм D-LIAN (теоретические свойства)

Утверждение 5. Алгоритм D-LIAN корректно завершается за конечное число итераций.

Идея доказательства: Конечное число пар (состояние, родитель) + проверка на дубликаты + гарантия того, что состояние попавшее в список OPEN рано или поздно будет из него удалено.

Утверждение 6. Алгоритм D-LIAN является полным относительно класса задач $P^{\Delta_{\max}}$, т.е. гарантирует отыскание решения задачи, если для неё существует Δ_{\max} -решение и гарантирует корректное завершение в противном случае.

Идея доказательства: от противного (с опорой на Теорему 5).

Утверждение 7. Длина пути, отыскиваемого алгоритмом D-LIAN, не превосходит длину кратчайшего Δ_{\max} -пути.

Идея доказательства: использование классической стратегия алгоритма A^* для исследование пространства состояний.

Теорема 5. Алгоритм D-LIAN является полным и оптимальным относительно класса задач $P^{\Delta_{\max}}$, т.е. гарантирует отыскание Δ_{\max} -решение, если оно существует, и гарантирует, что стоимость найденного решения не превосходит стоимость любого другого Δ_{\max} -решения.

Экспериментальные исследования

Входные данные

- Коллекция карт Moscow Maps (500 × 500)
 - 100 фрагментов реальных карт
- Коллекция карт Warcraft III (512 × 512) [1]
 - 36 карт
- Коллекция карт Baldurs Gate (512 × 512) [1]
 - 75 карт

Задания

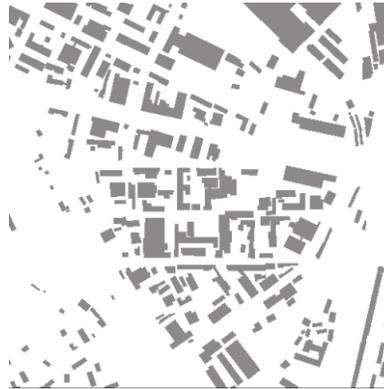
- 5 заданий на карту
 - случайная генерация
 - Расстояния между s и $g > 400$ клеток
- $\alpha_{max} = \{20^\circ, 25^\circ, 30^\circ\}$ [2]

Алгоритмы

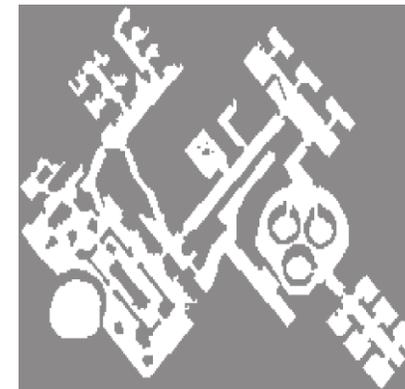
- LIAN
- D-LIAN
- Theta*-LA [3]
- wTheta*-LA [4]

[1] Sturtevant N.R. Benchmarks for grid-based pathfinding // IEEE Transactions on Computational Intelligence and AI in Games, 4(2), 2012, pp. 144-148.

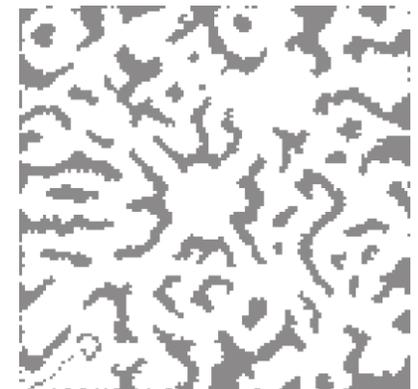
Moscow Maps



Baldurs Gate



Warcraft III



Основные вопросы для исследования

- Насколько предлагаемые в работе алгоритмы лучше аналогов
- Насколько техника адаптивного изменения параметра Δ влияет на эффективность

[2] Яковлев К.С., Макаров Д.А., Баскин Е.С. Метод автоматического планирования траектории беспилотного летательного аппарата в условиях ограничений на динамику полета // Искусственный интеллект и принятие решений, 4, 2014.

[3] Kim H., Kim D., Shin J.U., Kim H., Myung H. Angular rate-constrained path planning algorithm for unmanned surface vehicles // Ocean Engineering, 84, 2014. pp. 37-44.

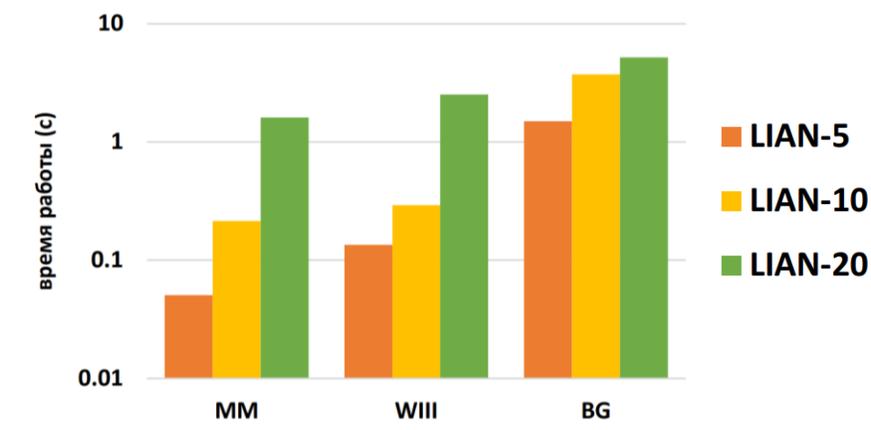
[4] Yakovlev K., Baskin E., Hramoin I. Grid-based angle-constrained path planning // KI 2015. pp. 208-221.

Результаты

Процент успешно решенных заданий

	Moscow Maps			Baldur's Gate			Warcraft III		
	20°	25°	30°	20°	25°	30°	20°	25°	30°
LIAN-5	99.4%	100.0%	100.0%	100.0%	100.0%	100.0%	99.4%	100.0%	100.0%
LIAN-10	96.8%	99.0%	99.6%	87.7%	96.2%	98.4%	92.7%	96.1%	98.9%
LIAN-20	87.2%	91.4%	94.0%	61.6%	72.8%	81.3%	76.6%	80.5%	84.4%
wTheta*-LA	56.4%	86.8%	93.8%	33.8%	76.2%	91.4%	49.4%	80.0%	87.7%
Theta*-LA	4.4%	6.2%	14.0%	10.6%	10.9%	11.7%	4.0%	5.5%	10.0%

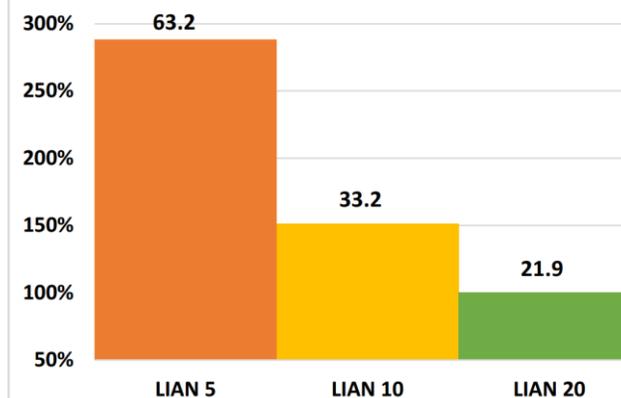
Время работы (лог-шкала)



Длина пути (Moscow Maps, $\alpha_{\max}=25^\circ$)



Число поворотов (Moscow Maps, $\alpha_{\max}=25^\circ$)



Нормировка на результаты лучшего алгоритма

Результаты

Процент успешно решенных заданий

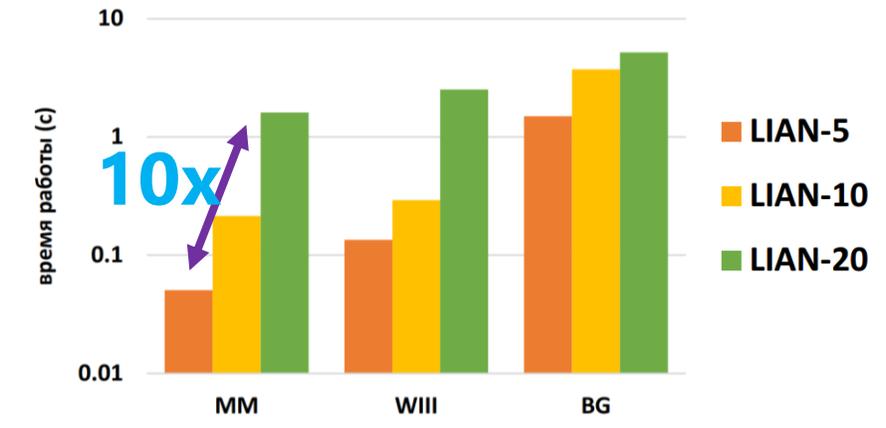
	Moscow Maps			Baldur's Gate			Warcraft III		
	20°	25°	30°	20°	25°	30°	20°	25°	30°
LIAN-5	99.4%	100.0%	100.0%	100.0%	100.0%	100.0%	99.4%	100.0%	100.0%
LIAN-10	96.8%	99.0%	99.6%	87.7%	96.2%	98.4%	92.7%	96.1%	98.9%
LIAN-20	87.2%	91.4%	94.0%	61.6%	72.8%	81.3%	76.6%	80.5%	84.4%
wTheta*-LA	56.4%	86.8%	93.8%	33.8%	76.2%	91.4%	49.4%	80.0%	87.7%
Theta*-LA	4.4%	6.2%	14.0%	10.6%	10.9%	11.7%	4.0%	5.5%	10.0%

До
2x

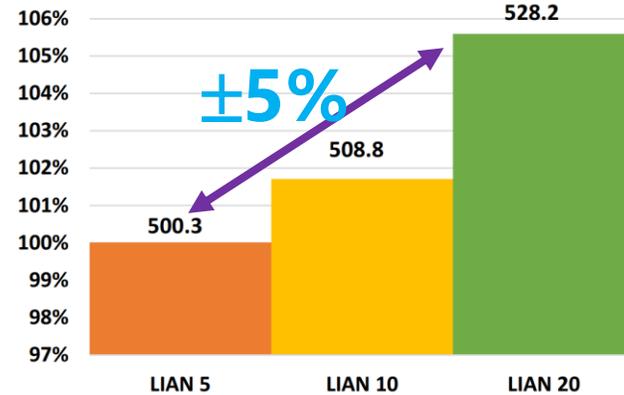
Разница в числе решенных заданий

- Процент решенных заданий алгоритмом LIAN существенно выше, чем у аналогов (разница до 2x раз)
- Низкие значения Δ влекут наибольшую скорость работы и наименьшую длину пути
- При высокие значениях Δ результирующие пути содержат наименьшее число

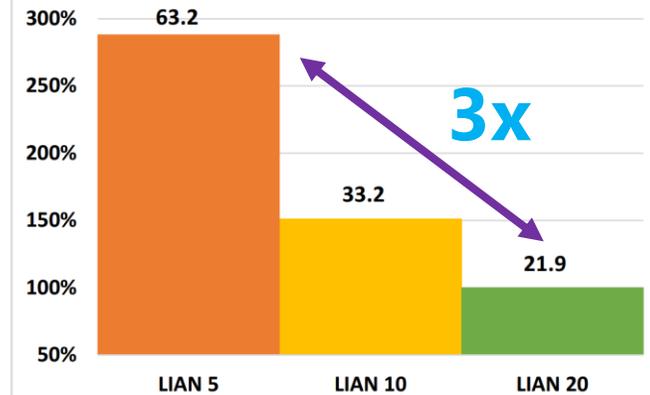
Время работы (лог-шкала)



Длина пути (Moscow Maps, $\alpha_{\max}=25^\circ$)



Число поворотов (Moscow Maps, $\alpha_{\max}=25^\circ$)



Нормировка на результаты лучшего алгоритма

Результаты

Процент успешно решенных заданий

	Moscow Maps			Baldur's Gate			Warcraft III		
	20°	25°	30°	20°	25°	30°	20°	25°	30°
LIAN-5	99.4%	100.0%	100.0%	100.0%	100.0%	100.0%	99.4%	100.0%	100.0%
LIAN-10	96.8%	99.0%	99.0%	87.7%	90.2%	88.4%	92.7%	96.1%	98.9%
LIAN-20	87.4%	87.4%	87.4%	81.3%	76.6%	80.5%	84.4%		
wTheta*-LA	56.4%	86.8%	93.8%	33.8%	76.2%	91.4%	49.4%	80.0%	87.7%
Theta*-LA	4.4%	6.2%	14.0%	10.6%	10.9%	11.7%	4.0%	5.5%	10.0%

Способность успешно решать задачи ($\Delta=5$) +
 высокая скорость работы ($\Delta=5$) +
 Низкая длина пути ($\Delta=5$) +

Разница в числе решенных заданий
 Низкое число поворотов ($\Delta=20$)

до
2x

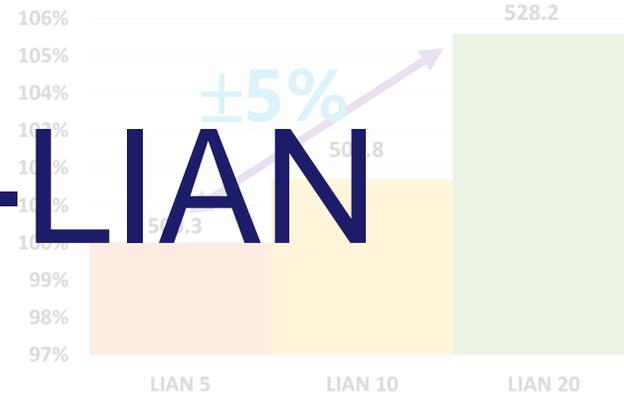
- Процент решенных заданий алгоритмом LIAN существенно выше, чем у аналогов (разница до 2x раз)
- Низкие значения Δ влекут наибольшую скорость работы и наименьшую длину пути
- При высокие значениях Δ результирующие пути содержат наименьшее число поворотов

D-LIAN

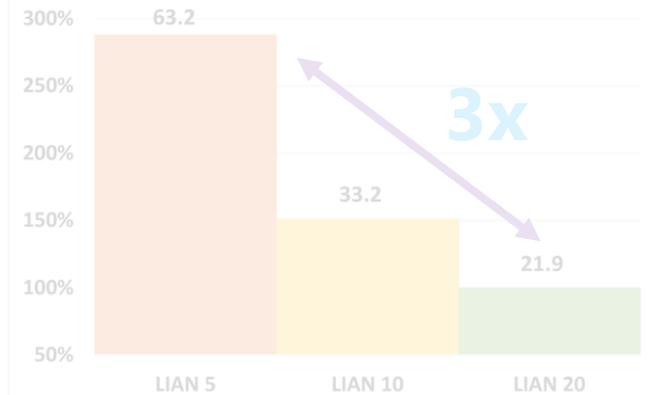
Время работы (лог-шкала)



Длина пути (Moscow Maps, $\alpha_{\max}=25^\circ$)



Число поворотов (Moscow Maps, $\alpha_{\max}=25^\circ$)

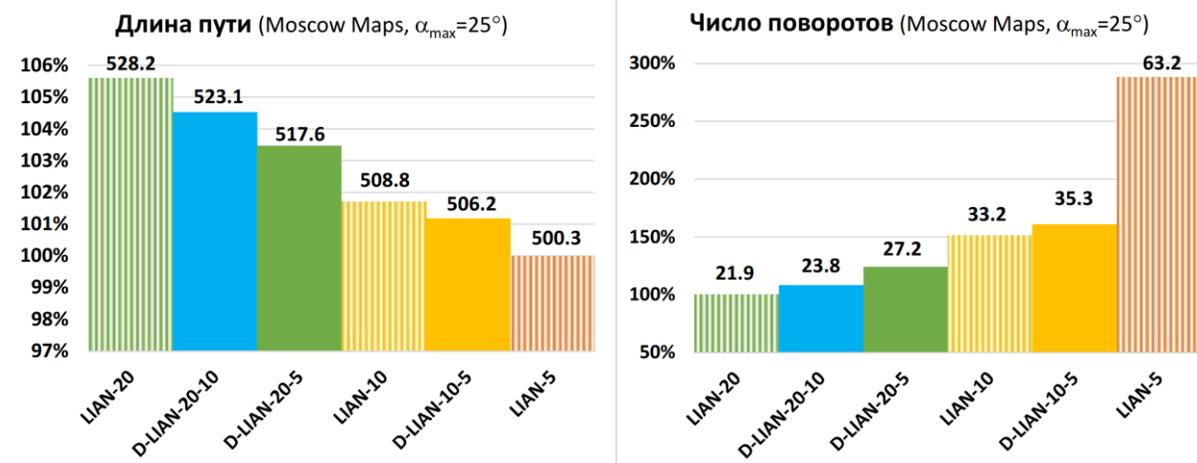
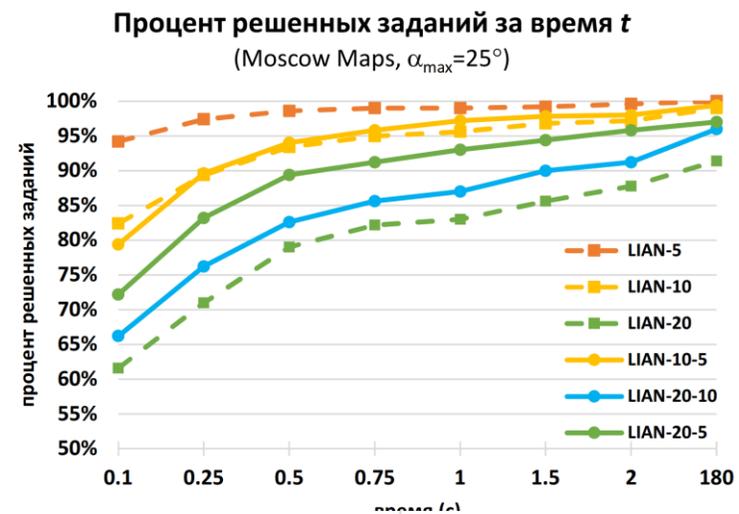


Нормировка на результаты лучшего алгоритма

Результаты (D-LIAN)

Увеличения числа успешно решенных заданий (в %)

	Moscow Maps			Baldur's Gate			Warcraft III		
	20°	25°	30°	20°	25°	30°	20°	25°	30°
D-LIAN-10-5	43.8%	40.0%	50.0%	21.7%	50.0%	12.5%	53.8%	85.7%	100.0%
D-LIAN-20-10	40.6%	46.5%	43.3%	34.0%	56.9%	64.3%	28.6%	68.6%	39.3%
D-LIAN-20-5	59.4%	65.1%	70.0%	57.6%	59.8%	45.7%	50.0%	80.0%	92.9%



Алгоритм становится строго лучше

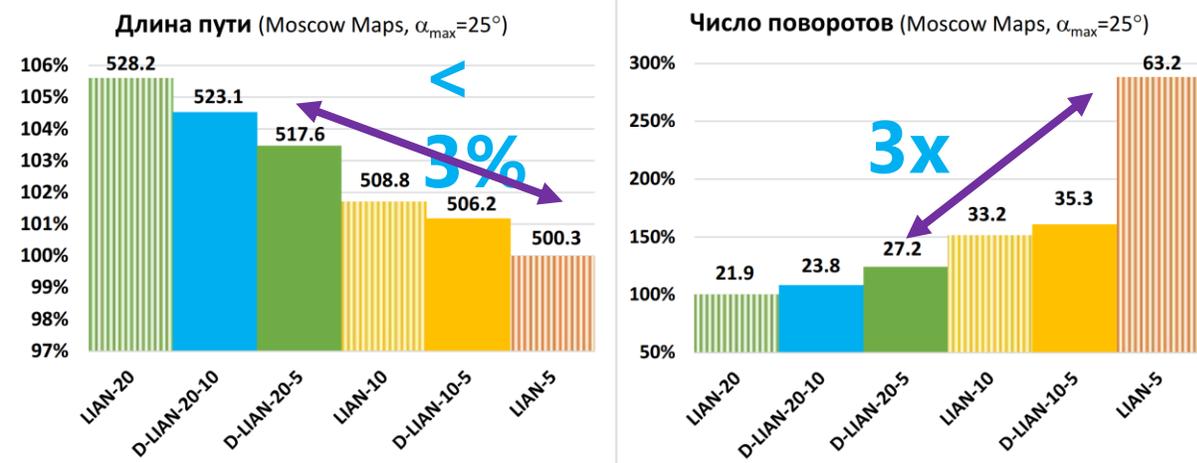
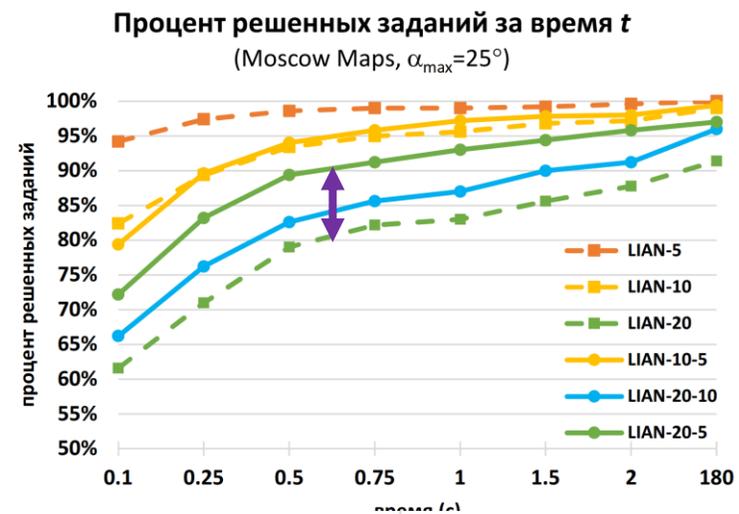
- Существенный рост числа решенных задач
- Можно сочетать низкую длину и невысокое число поворотов

Нормировка на результаты лучшего алгоритма

Результаты (D-LIAN)

Увеличения числа успешно решенных заданий (в %)

	Moscow Maps			Baldur's Gate			Warcraft III		
	20°	25°	30°	20°	25°	30°	20°	25°	30°
D-LIAN-10-5	43.8%	40.0%	50.0%	21.7%	50.0%	12.5%	53.8%	85.7%	100.0%
D-LIAN-20-10	40.6%	46.5%	43.3%	34.0%	56.9%	64.3%	28.6%	68.6%	39.3%
D-LIAN-20-5	59.4%	65.1%	70.0%	57.6%	59.8%	45.7%	50.0%	80.0%	92.9%



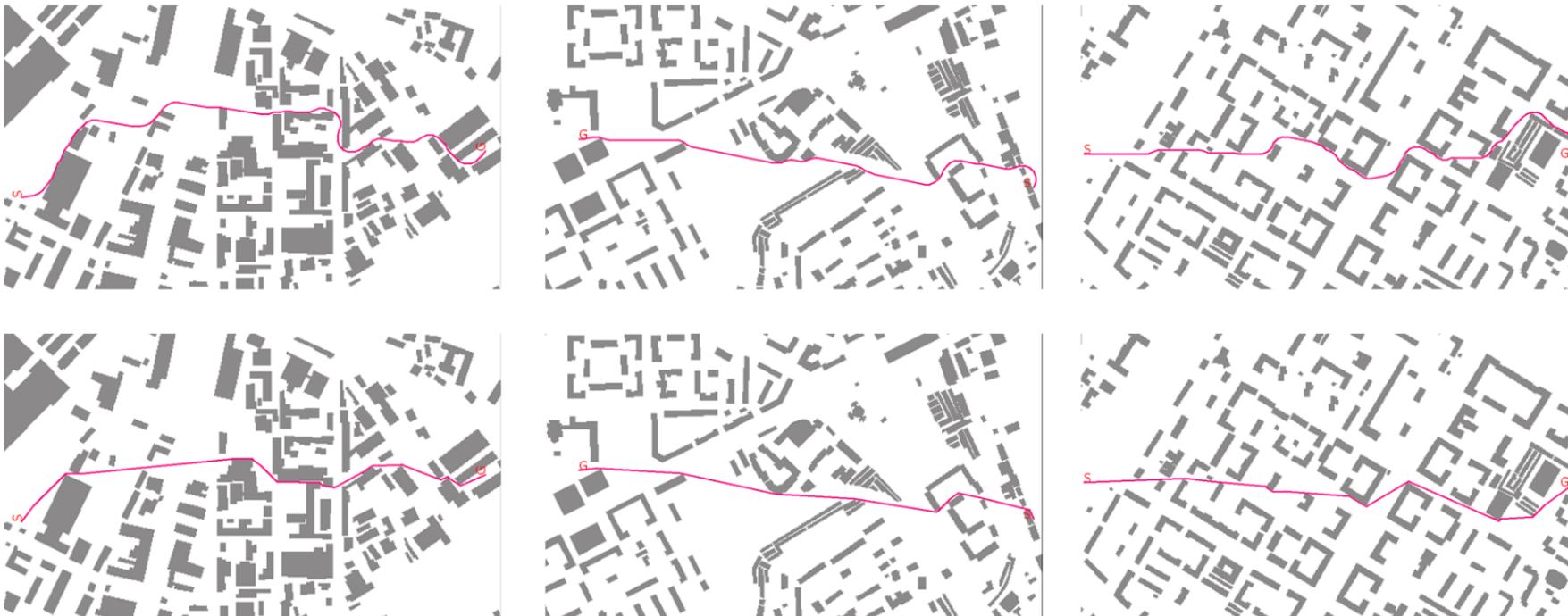
Алгоритм становится строго лучше

- Существенный рост числа решенных задач
- Можно сочетать низкую длину и невысокое число поворотов

Нормировка на результаты лучшего алгоритма

Выводы (задача AC-PF)

- Предложены эффективные алгоритмы решения задачи **AC-PF** – LIAN, D-LIAN
- Установлена полнота и оптимальность алгоритмов относительно вполне определенного класса задач **AC-PF**
- Проведено экспериментальное исследование, показавшее существенное преимущество перед аналогами



Методы машинного обучения для поиска пути на ГРД

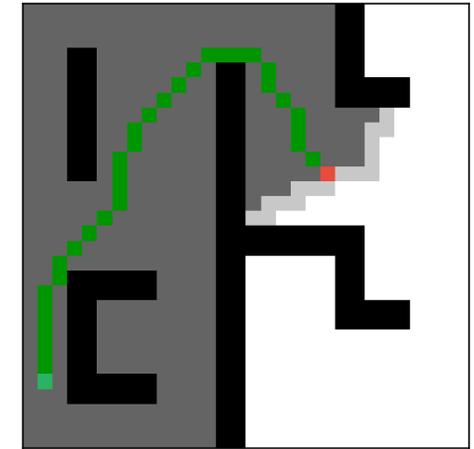
Классический эвристический поиск (A^*)

- Лучше*, чем неинформированный поиск (Dijkstra)
 - За счет использования эвристической функции, которая оценивает расстояние от любой вершины до целевой
 - Но на практике эта оценка расстояния не точна, т.к. не учитывает топологию ГРД
- =>
- В практических задачах (при наличии препятствий) **сокращение пространства поиска (числа итераций) незначительное**

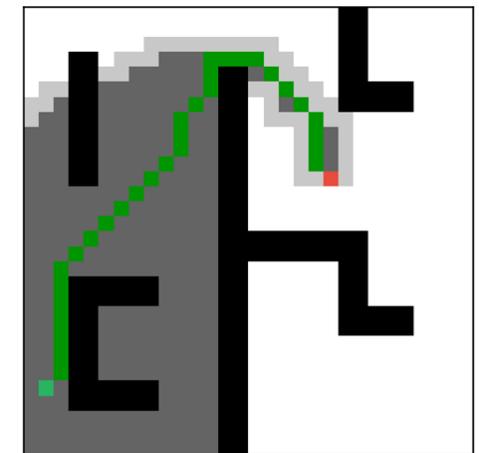
Можно ли повысить эффективность поиска (снизить число итераций)?

* Как минимум не хуже по числу итераций, во многих практически задачах –лучше

Dijkstra



A^*



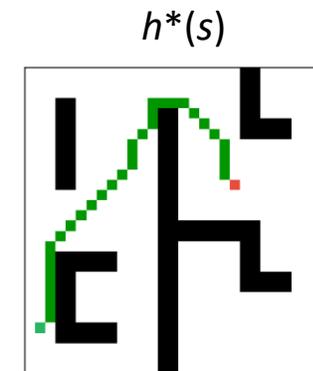
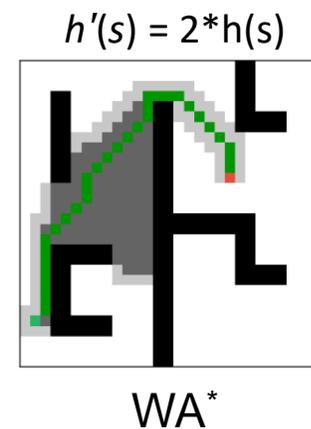
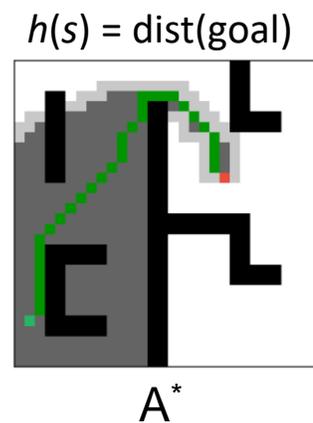
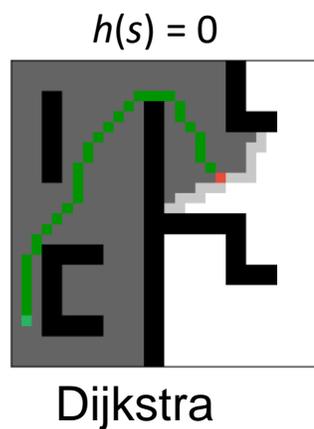
Информированность эвристических функций

A^* – итеративный алгоритм поиска, исследующий пространство состояний и обрабатывающий состояния в соответствии с их f -значениями

$$f(s) = g(s) + h(s)$$

$g(s)$ – стоимость (длина) пути от старта до состояния (вершины) s (вычисляется и обновляется алгоритмом в ходе работы)

$h(s)$ – эвристическая оценка стоимости пути от s до цели (передается на вход алгоритму)



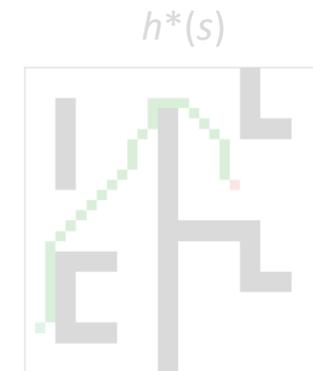
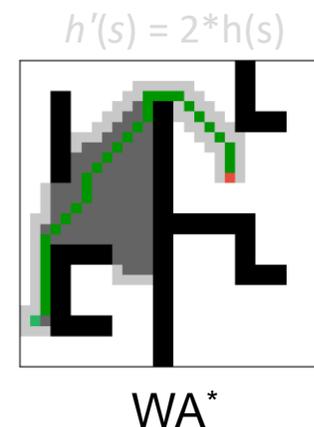
Информированность эвристических функций

A^* – итеративный алгоритм поиска, исследующий пространство состояний и обрабатывающий состояния в соответствии с их f -значениями

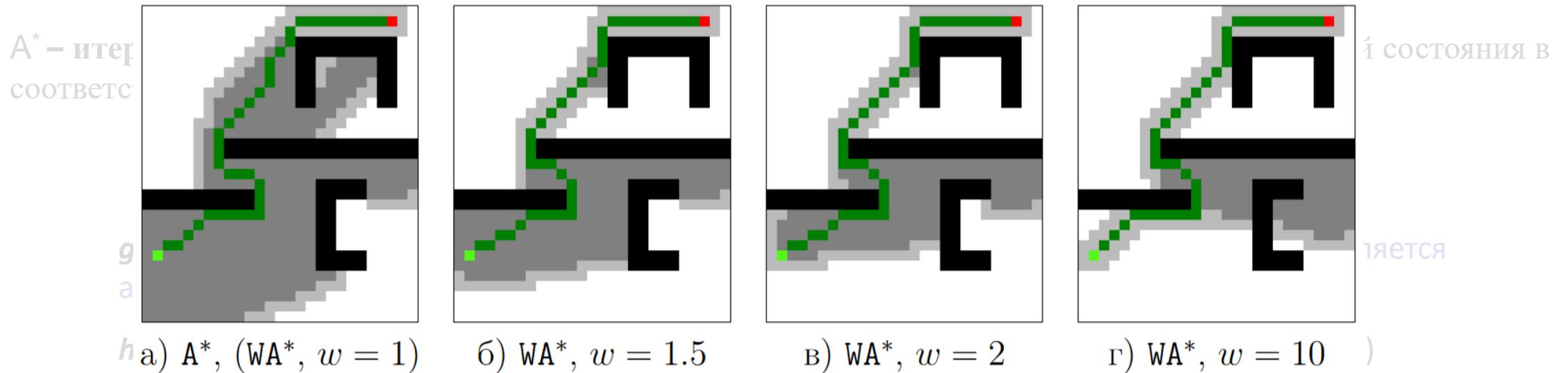
$$f(s) = g(s) + h(s)$$

$g(s)$ – стоимость (длина) пути от старта до состояния (вершины) s (вычисляется и обновляется алгоритмом в ходе работы)

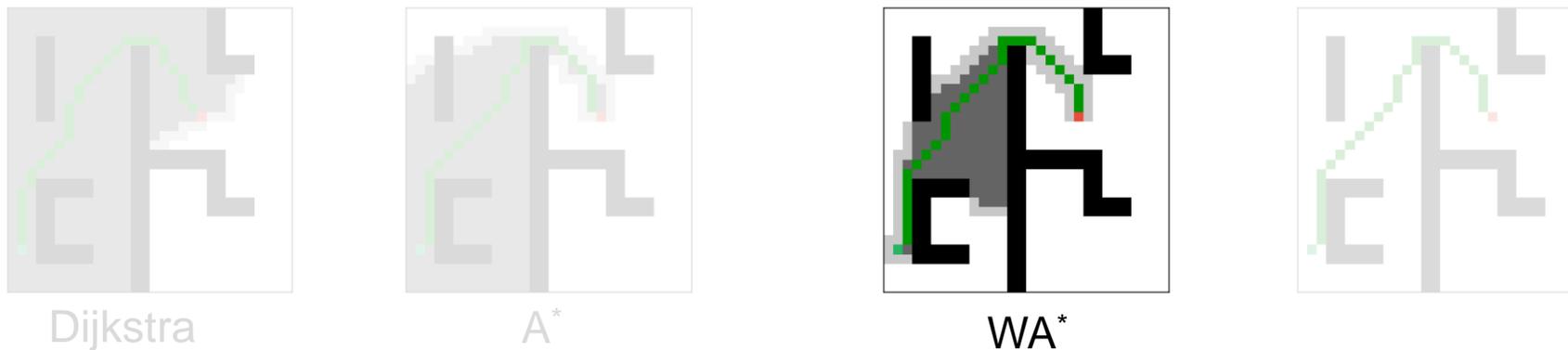
$h(s)$ – эвристическая оценка стоимости пути от s до цели (передается на вход алгоритму)



Информированность эвристических функций



Эффект от взвешивания эвристики зачастую ограничен



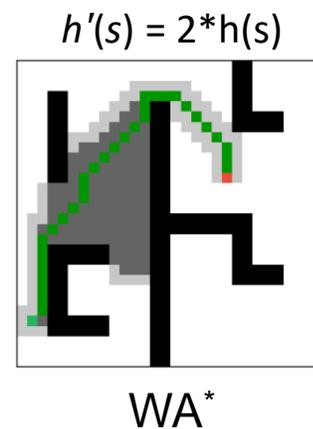
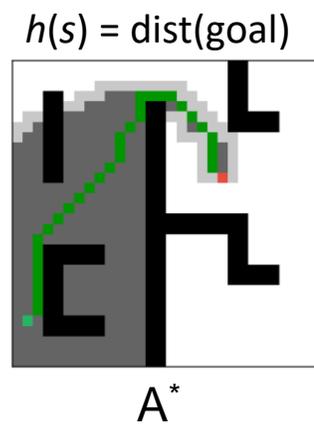
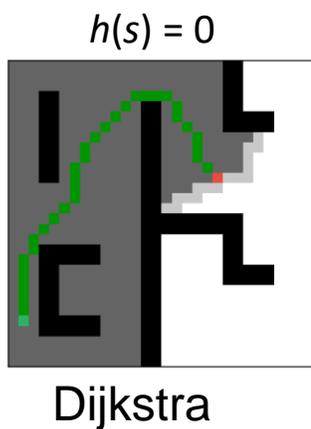
Информированность эвристических функций

A^* – итеративный алгоритм поиска, исследующий пространство состояний и обрабатывающий состояния в соответствии с их f -значениями

$$f(s) = g(s) + h(s)$$

$g(s)$ – стоимость (длина) пути от старта до состояния (вершины) s (вычисляется и обновляется алгоритмом в ходе работы)

$h(s)$ – эвристическая оценка стоимости пути от s до цели (передается на вход алгоритму)

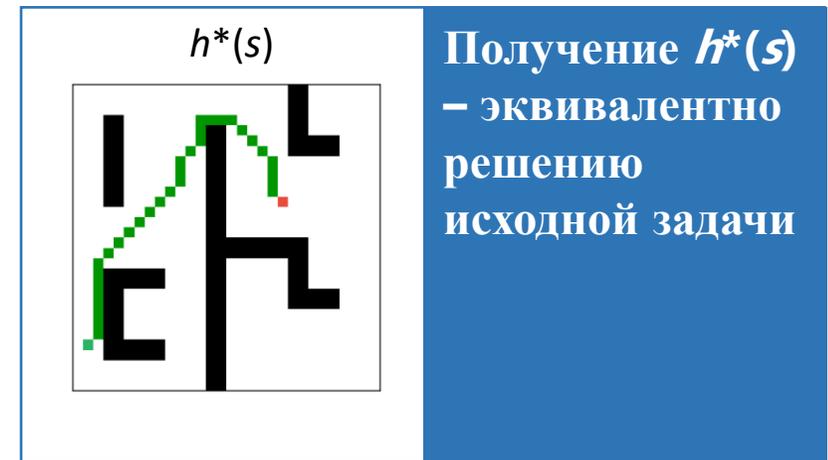
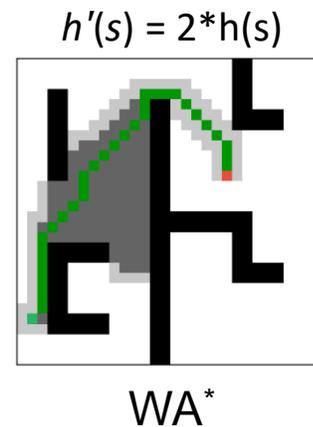
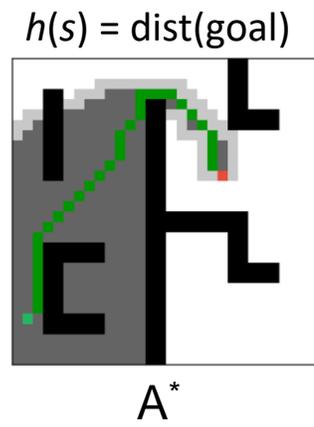
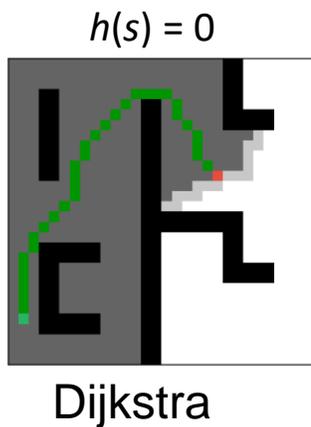
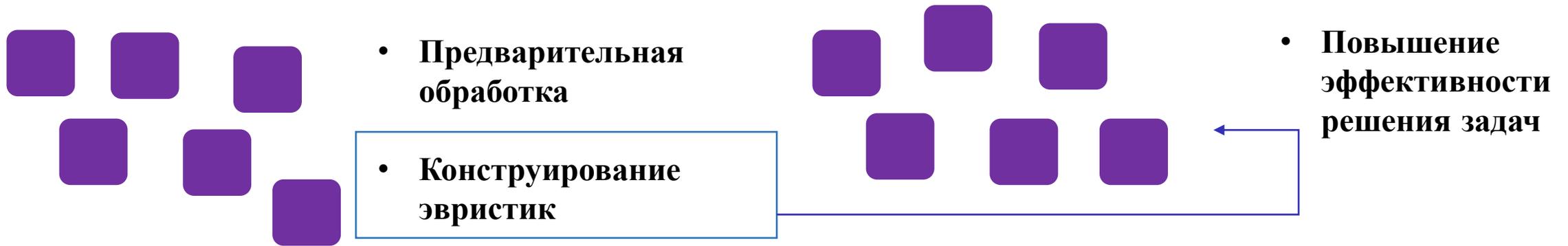


$h^*(s)$

Получение $h^*(s)$
– эквивалентно
решению
исходной задачи

Предположение о выборке задач

Практически ориентированный сценарий: решается множество однотипных задач планирования траектории мобильного робота (робот перемещается в городской/офисной среде)



Продолжение следует

....

Результаты

- Метод построения пути на динамических графах регулярной декомпозиции, допускающий возможность перемещения между произвольными вершинами графа и гарантирующий полноту и оптимальность отыскиваемых решений.
- Модификация указанного метода, существенно повышающая его вычислительную эффективность за счет техники сброса родительского указателя в дереве поиска и дающая ряд гарантий о качестве отыскиваемых решений.
- Метод конфликтно-ориентированного планирования для построения совокупности неконфликтных путей на графе регулярной декомпозиции, допускающий возможность перемещения между произвольными вершинами графа и гарантирующий оптимальность отыскиваемых решений.
- Метод приоритизированного поиска совокупности неконфликтных путей на графе регулярной декомпозиции, опирающийся на модифицированный метод поиска путей на динамическом графе и использующий ряд оригинальных эвристик для повышения эффективности своей работы.

Результаты

- Семейство алгоритмов поиска пути на графе регулярной декомпозиции, учитывающих геометрические ограничения, и гарантирующее отыскание оптимальных решений в определенном классе.
- Оригинальные эвристические функции для решения задачи поиска пути на графе регулярной декомпозиции и способы их интеграции с алгоритмами систематического поиска.
- Способ автоматического построения (обучения) указанных эвристических функций на основе глубоких нейронных сетей, построенных на современных архитектурных принципах (с использованием блоков внимания).
- Гибридные методы поиска пути на графе регулярной декомпозиции, использующие предлагаемые обучаемые эвристики и алгоритмы систематического поиска, гарантирующие корректность отыскиваемых решений и, в отдельных случаях, их ограниченную суб-оптимальность.

Апробация результатов (конференции, семинары)

- Международная конференция "Системный анализ и информационные технологии" (**САИТ**) 2015 (Светлогорск, Россия)
- Всероссийский научно-практический семинар "Беспилотные транспортные средства с элементами искусственного интеллекта" (**БТС-ИИ**), 2016 (Иннополис, Россия), 2017 (Казань, Россия), 2019 (Санкт-Петербург, Россия), 2021 (Москва, Россия)
- Национальная конференция по искусственному интеллекту с международным участием (**КИИ**) 2016 (Смоленск, Россия), 2018 (Москва, Россия), 2021 (Таганрог, Россия)
- Международная научно-техническая конференция "Экстремальная робототехника" (**ЭР**) 2016, 2025 (Санкт-Петербург, Россия)
- Совместный семинар Российской ассоциации искусственного интеллекта и Федерального исследовательского центра "Информатика и управление" Российской академии наук (**семинар РАИИ-ФИЦ ИУ РАН**) 2023 (Москва, Россия)
- German Conference on Artificial Intelligence (**КИ**) 2015 (Дрезден, Германия), 2019 (Касеель, Германия)
- Workshop on Multi-agent Pathfinding (**WoMAPF**) 2016 (Нью-Йорк, США), 2019 (Макао, КНР)

Апробация результатов (конференции, семинары)

- International Conference on Automated Planning and Scheduling (**ICAPS**) 2017 (Питтсбург, США), 2020 (Франция, онлайн), 2021 (КНР, онлайн)
- International Conference on Autonomous Agents and Multi-agent Systems (**AAMAS**) 2018 (Стокгольм, Швеция)
- International Conference on Informatics in Control, Automation and Robotics (**ICINCO**) 2020 (Франция, онлайн)
- European Conference on Mobile Robotics (**ECMR**) 2019 (Прага, Чехия)
- Conference on Interactive Collaborative Robotics (**ICR**) 2019 (Стамбул, Турция), 2020 (Россия, онлайн)
- IFAC Symposium on Robot Control (**SyRoCo**) 2022 (Япония, онлайн)
- AAI Conference on Artificial Intelligence (**AAAI**) 2023 (США, онлайн)
- International Symposium on Combinatorial Search (**SoCS**) 2024 (Канада, 2024)
- IEEE Conference on Intelligent Robots and Systems (**IROS**) 2024 (ОАЭ, 2024), 2025 (Ханчжоу, КНР)

Апробация результатов (гранты)

- 075-15-2024-544, "Математические модели и численные методы как основа для разработки робототехнических комплексов, новых материалов и интеллектуальных технологий конструирования" (**МНФО РФ**, 2024-2026 гг., исполнитель).
- 075-15-2020-799, "Методы построения и моделирования сложных систем на основе интеллектуальных и суперкомпьютерных технологий, направленные на преодоление больших вызовов" (**МНФО РФ**, 2020-2023 гг., исполнитель).
- 16-11-00048 "Создание теории, методов и моделей децентрализованного управления поведением коллективов когнитивных робототехнических систем в недетерминированной среде" (**РНФ**, 2016-2018 гг., исполнитель; 2019-2020 гг., **руководитель**);
- 20-57-00011 Бел_а "Вычислительно эффективные методы навигации группы автономных колесных роботов в динамической среде" (**РФФИ**, 2020-2021 гг., исполнитель);
- 18-37-20032 мол_а_вед "Методы управления автономными техническими объектами на основе планирования траектории в среде с динамическими препятствиями" (**РФФИ**, 2019-2020 гг., **руководитель**);
- 17-29-07053 офи_м "Гетерогенные иерархические системы динамического планирования и управления поведением интеллектуального агента" (**РФФИ**, 2017-2019 гг., **исполнитель**);

Апробация результатов (гранты)

- 17-07-00281-а "Модели и методы решения задач интеллектуального управления коалицией сложных технических объектов" (**РФФИ**, 2017-2019 гг., **исполнитель**);
- 15-07-07483-а "Исследование методов и разработка алгоритмов картирования, локализации и автоматического планирования траектории сложных технических объектов, обладающих многими степенями свободы" (**РФФИ**, 2015-2017 гг., **руководитель**);
- 15-37-20893 мол_а_вед " Исследование и разработка методов и алгоритмов планирования траекторий в условиях коллективного решения задач интеллектуальными агентами" (**РФФИ**, 2015-2016 гг., **руководитель**);
- 14.514.11.4081 "Исследование методов и разработка алгоритмов и программных средств управления современными беспилотными транспортными системами на базе свободно-распространяемого программного обеспечения" (**МНФО РФ**, 2014 г., **ответственный исполнитель**).
- 14-11-00692 " Создание основ теории и технологии построения многоуровневых систем управления коалициями сложных технических объектов в динамической среде на основе взаимодействия методов искусственного интеллекта и теории динамических систем" (**РНФ**, 2014-2016 гг., **исполнитель**).

- **15** статей в журналах категории **К1**
- **6** публикаций в сборниках трудов конференций, имеющих рейтинг **A/A*** по CORE
- **20** публикаций в сборниках трудов ведущих конференций, индексируемые в международных базах данных (**Scopus/WoS**)
- **13** публикации в сборниках трудов Российских профильных конференций, индексируемые в **РИНЦ**

Публикации (журналы)

- Яковлев К.С., Макаров Д.А., Баскин Е.С. Метод автоматического планирования траектории беспилотного летательного аппарата в условиях ограничений на динамику полета // **Искусственный интеллект и принятие решений**, 4, 2014. С.3-17. **К1**.
- Яковлев К.С., Хитьков В.В., Логинов М.И., Петров А.В. Система навигации группы БЛА на основе маркеров // **Робототехника и техническая кибернетика**, 4(5), 2014. С. 44-48. **К1**.
- Макаров Д.А., Панов А.И., Яковлев К.С. Архитектура многоуровневой интеллектуальной системы управления беспилотными летательными аппаратами // **Искусственный интеллект и принятие решений**, 3, 2015. С.18-32. **К1**.
- Яковлев К.С., Баскин Е.С., Андрейчук А.А. Метод автоматического планирования совокупности траекторий для навигации беспилотных транспортных средств // **Управление большими системами**, 58, 2015. С. 306-342. ВАК **К1**.
- Emel'yanov S., Makarov D., Panov A., Yakovlev K. Multilayer cognitive architecture for UAV control // **Cognitive Systems Research**, 39, 2016. P. 58-72. **К1**.
- Панов А.И, Яковлев К.С. Взаимодействие стратегического и тактического планирования поведения коалиции агентов в динамической среде // **Искусственный интеллект и принятие решений**, 4, 2016. С.68-78. **К1**.
- Андрейчук А.А., Яковлев К.С. Методы планирования траектории на плоскости с учетом геометрических ограничений // **Известия РАН. Теория и системы управления**, 6, 2017. С. 125-140. **К1**.
- Боковой А.В., Муравьев К.Ф., Яковлев К.С. Система одновременного картирования, локализации и исследования неизвестной местности по видеопотоку // **Информационные технологии и вычислительные системы**, 2, 2020. С. 51-61. **К1**

Публикации (журналы)

- Яковлев К.С. AA-SIPP: Алгоритм планирования в среде с динамическими препятствиями // **Искусственный интеллект и принятие решений**, 1, 2020. С. 80-82. **К1**.
- Яковлев К. С., Белинская Ю. С., Макаров Д. А., Андрейчук А. А. Безопасно-интервальное планирование и метод накрытий для управления движением мобильного робота в среде со статическими и динамическими препятствиями // **Автоматика и телемеханика**, 6, 2022. С. 96-117. **К1**.
- Яковлев К.С., Андрейчук А.А., Скрынник А.А., Панов А.И. Методы планирования и обучения в задачах многоагентной навигации // **Доклады Российской академии наук. Математика, информатика, процессы управления**, 508, 1, 2022. С. 88-93. **К1**.
- Staroverov A., Muravyev K., Yakovlev K., Panov A.I. Skill Fusion in Hybrid Robotic Framework for Visual Object Goal Navigation // **Robotics**, 12(4), 2023. pp. 104. **К1**.
- Миронов К.В., Юдин Д.А., Алхаддад М., Макаров Д.А., Пушкарев Д.С., Ленок С.А., Белкин И.В., Криштопик А.С., Головин В.А., Яковлев К.С., Панов А.И. STRL-Robotics: Интеллектуальное управление поведением робототехнической платформы в человеко-ориентированной среде // **Искусственный интеллект и принятие решений**, 2, 2023. С. 45-63.
- Головин В.А., Яковлев К.С. Примитивы движения робота в задаче планирования траектории с кинематическими ограничениями // **Информатика и автоматизация**, 22(6), 2023. С. 1354-1386. **К1**.
- Kirilenko D., Andreychuk A., Panov A.I., Yakovlev K. Generative Models for Grid-Based and Image-Based Pathfinding // **Artificial Intelligence**, 338, 2025. pp. 104238. **К1**.

Публикации (конференции A/A*)

- Yakovlev, K., Andreychuk, A. Any-Angle Pathfinding for Multiple Agents Based on SIPP Algorithm. In Proceedings of the 27th International Conference on Automated Planning and Scheduling (**ICAPS 2017**). pp. 586-593.
- Andreychuk A., Yakovlev K. Two Techniques That Enhance the Performance of Multi-robot Prioritized Path Planning // In Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (**AAMAS 2018**). pp. 2177-2179.
- Yakovlev K., Andreychuk A., Stern R. Revisiting Bounded-Suboptimal Safe Interval Path Planning // In Proceedings of the 30th International Conference on Automated Planning and Scheduling (**ICAPS 2020**). pp. 300-304.
- Yakovlev K., Andreychuk A. Towards Time-Optimal Any-Angle Path Planning With Dynamic Obstacles // In Proceedings of the 31st International Conference on Automated Planning and Scheduling (**ICAPS 2021**). pp. 405-414.
- Kirilenko D., Andreychuk A., Panov A., Yakovlev K. TransPath: Learning Heuristics For Grid-Based Pathfinding via Transformers // In Proceedings of the 37th AAAI Conference on Artificial Intelligence (**AAAI 2023**). pp. 12436-12443.
- Yakovlev K., Andreychuk A., Stern R. Optimal and Bounded Suboptimal Any-Angle Multi-agent Pathfinding // In Proceedings of the IEEE/RSJ International Conference on Intelligent Robotic Systems (**IROS 2024**). pp. 7996-8001.