

УДК 004.896

doi: 10.15622/rcai.2025.048

НАВИГАЦИЯ ГРУППЫ ВЗАИМОЗАМЕНЯЕМЫХ АГЕНТОВ В НЕПРЕРЫВНОЙ СРЕДЕ

А.В. Микрюкова (*avmikryukova@edu.hse.ru*)^A

С.А. Дергачев (*sadergachev@hse.ru*)^{A,B}

^A Национальный исследовательский университет

«Высшая школа экономики», Москва

^B Федеральный исследовательский центр

«Информатика и управление» РАН, Москва

В данной работе рассматривается задача планирования путей для группы взаимозаменяемых агентов в непрерывной среде. В отличие от классического случая, в рассматриваемой постановке целевые позиции не закреплены за конкретными агентами. Проведён обзор современных методов много-агентного планирования, в ходе которого выделен алгоритм *GAP*, который использует непрерывное представление пространства и предъявляет минимальные требования к входным данным. Алгоритм был экспериментально сравнен с актуальными дискретными подходами. Результаты показали, что переход к непрерывному представлению существенно повышает качество планирования. Вместе с тем, ограничения существующих алгоритмов могут снижать долю успешно решённых задач при увеличении плотности агентов.

Ключевые слова: много-агентная навигация, много-агентный поиск путей, навигация групп взаимозаменяемых агентов, немаркированная навигация.

Введение

Проблема координации действий групп мобильных агентов является одной из важных исследовательских задач, встречающихся во многих практических областях, среди которых автоматизация логистики и доставки грузов [Wurman et al., 2008], создание мобильных робототехнических платформ для экстренного реагирования, а также разработкой видеоигр и симуляторов транспортных потоков. Ключевое направление исследований заключается в создании алгоритмов, способных эффективно планировать безопасные перемещения десятков или сотен роботов/агентов в динамических средах с учетом ограниченных ресурсов.

Задачу координации движения зачастую формулируют как задачу много-агентного планирования путей (*Multi-Agent Path Finding, MAPF*). Она заключается в поиске совокупности траекторий, обеспечивающих перемещение каждого робота/агента из начальной позиции в целевую без столкновений. Вариативность постановок задачи *MAPF* порождает множество исследовательских направлений. Например, могут рассматриваться различные представления рабочего пространства: перемещения могут ограничиваться дискретной сеткой или происходить в непрерывном пространстве [Stern et al., 2019].

В классической постановке задачи *MAPF* каждому агенту назначена конкретная цель, однако, существуют и альтернативные формулировки. Например, задача немаркированного/анонимного много-агентного поиска путей (*Unlabeled/Anonymous MAPF, AMAPF*), в котором агенты взаимозаменяемы и цели не закреплены за конкретными агентами, главное, чтобы все целевые позиции были достигнуты.

Аналогично классической задаче, при планировании для группы взаимозаменяемых агентов так же может быть использовано как дискретное представление пространства, так и непрерывное. При этом существует лишь небольшое количество работ, посвященных этому направлению, причем все полные методы, работающие в непрерывном пространстве, опираются на различные ограничения, накладываемые на входные данные. Данная работа сфокусирована на задаче с непрерывным представлением среды, что более точно отражает реальные условия применения, например, в роботизированных складах или при координации автономных транспортных средств. Для исследования был выбран алгоритм *GAP* [Turpin et al., 2014], работающий в непрерывном пространстве и накладывающий наименьший набор ограничений на входные данные. В рамках работы впервые приводятся результаты экспериментального сравнения непрерывного метода *GAP* с дискретными алгоритмами *TSWAP* и *FlowNetwork*, что позволяет выявить практические преимущества и ограничения методов, работающих в непрерывном пространстве.

1. Обзор литературы

Нахождение оптимального решения в задаче много-агентного поиска пути (*MAPF*) в её классической формулировке относится к классу NP-трудных проблем, что существенно затрудняет разработку эффективных алгоритмов и их применение в системах с большим количеством агентов [Okumura et al., 2022]. В случае немаркированного планирования путей (*Unlabeled/Anonymous MAPF, AMAPF*) задача (с точки зрения времени достижения всеми агентами целевых позиций) становится разрешимой за полиномиальное время [Yu et al., 2013].

Рассмотрим подробнее методы решения задачи немаркированного много-агентного планирования траектории. Большинство существующих методов предполагает наличие заранее заданного дискретного представления среды в виде сетки или графа специального вида, что не всегда возможно на практике и может приводить к снижению качества решений. Например, в работе [Yu et al., 2013] представлен алгоритм *FlowNetwork*, работающий на безопасных графах единичных расстояний, гарантирующий нахождение *makespan*-оптимальных решений за полиномиальное время, благодаря сведению исходной задачи к проблеме поиска максимального потока в сети. На практике, при использовании графов больших размеров, указанных алгоритм требует существенных вычислительных ресурсов. Альтернативой являются эвристические подходы, такие как алгоритм *TSWAP* [Okumura et al., 2022], который жертвует оптимальностью ради снижения вычислительных затрат. *TSWAP* комбинирует локальные перестановки целей с поэтапным планированием, демонстрируя гораздо большую скорость работы по сравнению с алгоритмом *FlowNetwork*.

С другой стороны, существует ограниченное число подходов, позволяющих находить решение в непрерывном пространстве. Алгоритмы, подходящие для агентов, движение которых происходит в непрерывной среде, представлены в статьях [Turpin et al., 2014], [Solovey et al., 2015], [Kloder et al., 2006], [Banyassady et al., 2022], [Le et al., 2021]. Алгоритм в работе [Banyassady et al., 2022] является полным, он предназначен для среды сложной формы, однако не предусматривает наличие отдельных препятствий, что делает его непригодным для большинства практических сценариев. Подход, представленный в статье [Solovey et al., 2015], так же является полным, авторы доказывают, что сумма длин всех путей их решения не превышает значения $\frac{1}{2} \sum_{i=1}^n d_i$, где d_i – стоимость оптимального решения, а n – число агентов. Однако, для корректной работы алгоритма необходимо ввести сильные ограничения: минимальное расстояние между стартами/целями и препятствиями $\geq r$, а минимальное расстояние между позициями стартов и между позициями целей равно $\geq 2r$, где r – радиус агента. Алгоритм GAP, предложенный в [Turpin et al., 2014], гарантирует нахождение решения, не накладывая при этом условий на расстояние от агентов до препятствий, ограничивая только расстояния между стартами и между целями ($\geq r$). Авторы статьи [Kloder et al., 2006] используют комплексные многочлены для решения задачи, что позволяет им строить путь между целью и агентом максимально приближенным к кратчайшему, однако данный алгоритм не учитывает возможные столкновения между агентами, так как они представляются точками. Стоит отметить, что существует алгоритм, учитывающий различные особенности агентов (их форму, модель движения) [Le et al., 2021], однако он не гарантирует нахождение решений.

Ещё одним активно развивающимся подходом к решению задачи является использование нейронных сетей и обучения с подкреплением [Khan et al., 2021], [Chu et al., 2024]. Однако эти не обеспечивают теоретических гарантий безопасности и полноты, а также не рассматривают препятствия сложной структуры.

На основе проведенного анализа были выбраны следующие критерии сравнения алгоритмов: возможность работы в непрерывном пространстве, ограничения, накладываемые на входные данные, теоретические гарантии полноты и безопасности, оптимальность с точки зрения одной из метрик. Указанная информация для ключевых алгоритмов представлена в табл. 1. В дополнение к выбранным критериям для всех методов также был указан год публикации работы и доступность открытой реализации.

Таблица 1

Сравнительный анализ ключевых методов решения задачи *AMAPF*

Метод	Непрерывная среда	Полнота и безопасность	Ограничения (старты/цели)	Ограничения (позиции-препятствия)	Оптимальность	Реализация
<i>FlowNetwork</i> [Yu et al., 2013]	—	+	Н/П	Н/П	+	+
<i>TSWAP</i> [Okumura et al., 2022]	—	+	Н/П	Н/П	—	+
Метод из [Banyassady et al., 2022]	+	+	—	—	—	—
Метод из [Kloder et al., 2006]	+	—	+	+	—	—
Метод из [Solovey et al., 2015]	+	+	—	—	—	—
<i>GAP</i> [Turpin et al., 2014]	+	+	—	+	—	—
Метод из [Le et al., 2021]	+	—	+	+	—	+
<i>GPG</i> [Khan et al., 2021]	+	—	+	—	—	—
<i>PROFILE</i> [Chu et al., 2024]	+	—	+	—	—	—

Как можно видеть из результатов анализа, ни один из алгоритмов, работающих в непрерывном пространстве, не гарантирует оптимальность найденного решения. При этом алгоритмы [Turpin et al., 2014], [Solovey et al., 2015], [Banyassady et al., 2022] гарантируют нахождение безопасного решения при выполнении ряда условий. Среди них можно выделить алгоритм *GAP* [Turpin et al., 2014], накладывающий ограничения только на расстояния между стартами/целями. Отдельно отметим, что ни для одного алгоритма планирования в непрерывной среде нет открытой реализации для проведения экспериментов.

Таким образом, по результатам анализа был выбран алгоритм [Turpin et al., 2014], так как он гарантирует нахождение решений и накладывает наименьшее число ограничений на среду.

2. Формальная постановка задачи

Обозначим агентами натуральными числами от 1 до n , а множество агентов как A . Дано рабочее пространство R – непрерывная среда с препятствиями, заданными как замкнутые множества. Обозначим P пространство препятствий, тогда $R \setminus P$ – область, доступная для движения агентов. Каждый из агентов моделируется диском радиуса r и может двигаться с постоянной скоростью v . За t обозначим состояние агента в момент t , а за S – множество всех точек, занимаемых агентом.

Для каждого агента задана начальная позиция p_i . Также заданы позиции целей: g_1, g_2, \dots, g_m . Позиции стартов и целей задаются так, чтобы не возникало пересечений агентов с препятствиями и между собой:

За множество всех точек, занимаемых агентом, который движется по траектории/пути τ_i обозначим τ_i , при этом для корректности траектории также должно выполняться $\tau_i \cap P = \emptyset$.

Таким образом, задача немаркированного много-агентного планирования путей заключается в нахождении:

1. Функции назначения целей всем агентам σ , где $\sigma(i) = g_j$ означает, что агент i назначен к цели g_j .
2. Набора траекторий τ_i для каждого агента от его стартовой позиции до соответствующей цели т.ч. не возникает столкновений с препятствиями и между агентами:

где t_i – время, к которому агент i достигает цель.

Для оценки качества решения введём метрики *makespan* – максимальное время достижения целевой позиции среди всех агентов и *flowtime* – сумма времен достижения целевых позиций всеми агентами.

3. Алгоритм немаркированного много-агентного планирования пути в непрерывном пространстве

3.1. Общее описание алгоритма

В работе рассматривается алгоритм *GAP* [Turpin et al., 2014], который решает задачу планирования траекторий для групп взаимозаменяемых агентов в непрерывных средах с препятствиями. Основная идея заключается в декомпозиции проблемы на три ключевых этапа:

1. Поиск кратчайших путей: для каждого агента к каждой цели независимо от других агентов находится оптимальный маршрут.
2. Оптимальное назначение целей: решается задача лексикографического назначения, минимизирующая максимальную стоимость траекторий.
3. Приоритизация и параметризация:
 - Агенты упорядочиваются по приоритету на основе пересечений их траекторий.
 - Вводятся временные смещения для гарантии отсутствия столкновений между агентами.

Далее будут более подробно рассмотрены отдельные компоненты алгоритма.

3.2. Планирование траекторий и построение графа

Для работы алгоритма нам необходимо построить путь для каждого агента от его начальной позиции до всех целевых. Одним из наиболее распространенных и эффективных методов заключается в решении этой задачи, как задачи поиска пути на графе.

Итак, вводится граф общего вида $G = (V, E)$. Граф может формироваться путём дискретизации пространства \mathbb{R}^2 с использованием детерминированных или вероятностных методов [Kavraki L.E. et al., 1996]. Каждая вершина $v \in V$ представляет допустимое, не допускающее столкновений состояние в \mathbb{R}^2 . Кроме того, множество вершин V должно содержать все начальные и целевые состояния:

Ребра $E \subseteq V \times V$ соответствуют безопасным траекториям γ_{ij} между соответствующими вершинами: $\gamma_{ij} : [0, 1] \rightarrow \mathbb{R}^2$, $\gamma_{ij}(0) = v_i$, $\gamma_{ij}(1) = v_j$. Каждому ребру ставится в соответствие стоимость перехода c_{ij} .

Для гарантии полноты должны выполняться следующие свойства:

Также для всех стартов и целей должны выполняться следующие условия:

После построения графа для каждого вычисляется оптимальная траектория между позициями и , минимизирующая суммарную стоимость ребер. Этот этап реализуется с помощью алгоритма Дейкстры [Dijkstra, 1959]. В результате формируется матрица , где элемент соответствует стоимости пути в графе:

Если цель недостижима из , полагаем .

Стоит отметить, что в отличие от дискретных методов, в данном алгоритме граф может иметь свободную форму и строиться на основе различных подходов, например, на основе графа видимости или методом PRM [Kavraki et al., 1996].

3.3. Назначение целей

Назначение целей определяется путем решения задачи *LexBAP* (*Lexicographic Bottleneck Assignment Problem*). Решение гарантирует, что:

где — множество допустимых биекций.

Для снижения вычислительной сложности *LexBAP* аппроксимируется задачей:

где (например, в текущей реализации используется). При решении совпадает с *LexBAP*.

Модифицированная задача решается за с помощью Венгерского алгоритма и дополнительной проверкой условия:

3.4. Приоритизация и предотвращение столкновения

Для обеспечения предотвращения столкновений вводятся следующие правила частичного упорядочивания между агентами, основанный на их траекториях и взаимном расположении. Условия приоритизации задаются следующим образом:

1. Правило стартовых пересечений – если стартовая позиция агента лежит на оптимальном пути агента, последний получает более высокий приоритет

2. Правило целевых пересечений – приоритет агента больше, чем приоритет агента, если его траектория проходит через цель агента

На основе установленных правил вычисляется полный порядок и выполняется поиск смещений времени начала движения для каждого агента. Далее представлен алгоритм определения временных смещений:

1. Агенту с наивысшим приоритетом назначается .
2. Для каждого последующего агента вычисляется минимальное смещение:

В результате получается набор траекторий, гарантирующий:

где – параметр алгоритма, задающий дополнительный безопасный зазор.

4. Экспериментальное исследование

Для экспериментов были использованы три карты из широко используемого в *MAPF* сообществе набора бенчмарков *MovingAI* [Stern et al., 2019]: *random-32-32-20* (), *maze-32-32-2* (), *den312d* (). Иллюстрация карт представлена на рис. 1.

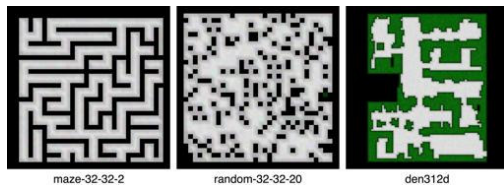


Рис. 1. Карты, используемые в экспериментах

Для каждой карты было случайно сгенерировано по 50 сценариев, содержащих по 100 пар стартовых и целевых позиций, удовлетворяющих ограничениям алгоритма *GAP*. Число агентов варьировалось от 10 до 100 с шагом 10; для запуска с k агентами выбирались первые k пар из сценария.

Для проведения экспериментальной оценки были реализованы несколько вариантов алгоритма *GAP*¹. Так, были реализованы варианты алгоритма, использующие начальное назначение, минимизирующее сумму длин путей () и назначение, полученное на основе аппроксимации решения задачи *LexBAP* (). Кроме того, были реализованы два варианта построения графа. В первом варианте использовалось более близкое к дискретному представление в виде 8-связной сетки. Во втором варианте использовался стохастический алгоритм *PRM* [Kavraki et al., 1996] с числом вершин равным 2000 и радиусом соединения 5.

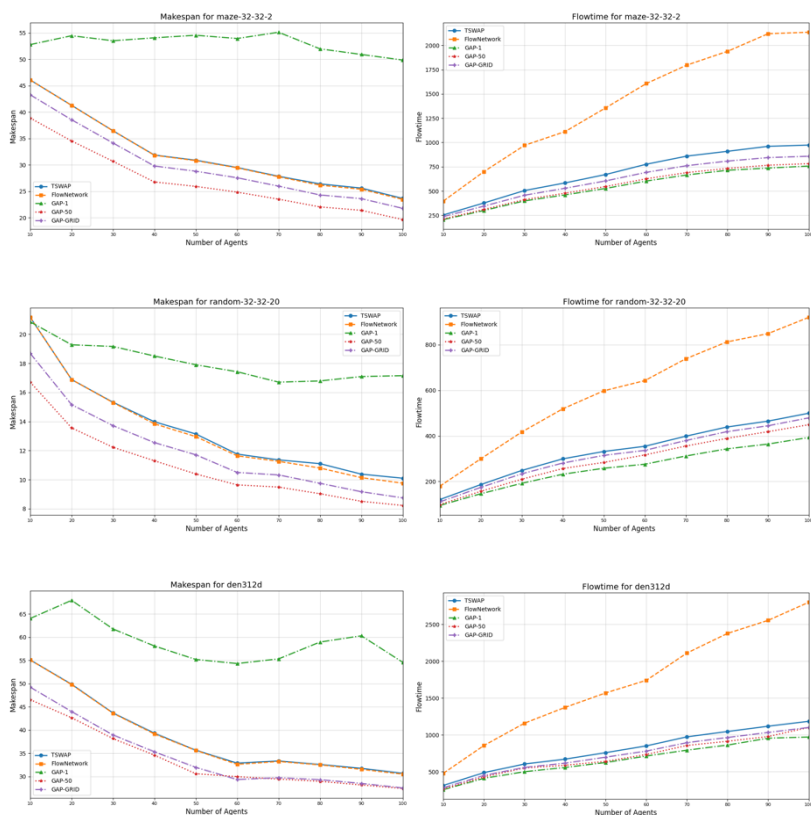


Рис. 2. Средние показатели метрик *flowtime* и *makespan* для различного числа агентов. Меньше – лучше

¹ Исходный код: <https://github.com/a-v-mikryukova/course-project-2>.

В экспериментальном исследовании участвовали следующие варианты алгоритма: два варианта использующих для формирования графа метод *PRM*, один из которых в задаче назначения использует (обозначен как ***GAP-1***), другой использует (обозначен как ***GAP-50***), третий вариант, обозначенный как ***GAP-GRID*** использует 8-связную сетку и . Кроме того, в рамках экспериментов для проведения сравнения были проведены запуски дискретных алгоритмов *TSWAP* [Okumura et al., 2022] и *FlowNetwork* [Yu et al., 2013].

Алгоритмы оценивались по средним значениям метрик *makespan* и *flowtime*. Результаты, представленные на рис. 2, оказались схожими для всех карт. Алгоритм *GAP-1* показал лучшие значения по *flowtime*, однако уступает другим методам по *makespan*. При этом, результаты *GAP-50* превосходят показатели дискретных методов (включая оптимальный по *makespan* алгоритм *FlowNetwork*) по обоим метрикам, приближаясь по *flowtime* к *GAP-1*. *GAP-GRID* занял промежуточное положение между дискретным алгоритмом *TSWAP* и непрерывным *GAP-50*, а на карте *den312d* его результаты почти совпадают с *GAP-50*.

Таким образом, алгоритмы, работающие в непрерывной среде, могут обеспечивать более высокое качество планирования даже при использовании представления близкого к полностью дискретному.

Также было проведено исследование о влиянии ограничений, накладываемых алгоритмом *GAP*, на процент успешно решенных заданий на карте *maze-32-32-2*. Стартовые и целевые позиции выбирались аналогично предыдущим экспериментам, однако при построении сценариев не учитывали ограничения алгоритма *GAP*. Результаты представлены в табл. 2. Так, можно видеть, что при числе агентов свыше 70 доля успешных решений упала ниже 80%, а при 100 агентах – до 65%. Такие результаты в большой степени связаны со структурой карты, где небольшое число свободных клеток образуют узкие проходы, что не позволяет выполнить ограничения алгоритма *GAP*. При этом отметим, что указанные задания могут быть решены рассмотренными дискретными алгоритмами в 100% случаев.

Таблица 2

Доля успешно решенных заданий алгоритмом *GAP-50* при полностью случайном расположении агентов

Число агентов	10	20	30	40	50	60	70	80	90	100
Процент успеха	99%	96%	92%	89%	85%	83%	79%	74%	70%	65%

Заключение

Таким образом, в работе была рассмотрена задача немаркированного много-агентного планирования путей в непрерывной среде. Проведённый анализ существующих подходов позволил выделить алгоритм *GAP* как наиболее универсальный благодаря минимальным ограничениям на входные данные. Экспериментальное сравнение с дискретными алгоритмами показало значительное преимущество *GAP* по качеству получаемых решений. Однако было установлено, что ограничения, накладываемые на входные данные, могут существенно снижать долю успешно решённых задач при увеличении плотности агентов. Таким образом, важным направлением будущих исследований является модификация алгоритма *GAP* для устранения или смягчения ограничений на входные данные при сохранении теоретических гарантий. Другим важным направлением работ является сравнение непрерывных алгоритмов с дискретными аналогами с точки зрения времени решения.

Список литературы

- [Banyassady et al., 2022] Banyassady B. et al. Unlabeled multi-robot motion planning with tighter separation bounds // 38th international symposium on computational geometry (SoCG 2022). – 2022.
- [Chu et al., 2024] Chu W. et al. Perception field based imitation learning for unlabeled multi-agent pathfinding // Science China Information Sciences. – Springer, 2024. – Vol. 67(5). – P. 152107.
- [Dijkstra, 1959] Dijkstra E.W. A note on two problems in connexion with graphs // Numerische mathematik. – 1959. – Vol. 1(1). – P. 269-271.
- [Kavraki et al., 1996] Kavraki L.E. et al. Probabilistic roadmaps for path planning in high-dimensional configuration spaces // IEEE transactions on Robotics and Automation. – IEEE, 1996. – Vol. 12(4). – P. 566-580.
- [Khan et al., 2021] Khan A., Kumar V., Ribeiro A. Large scale distributed collaborative unlabeled motion planning with graph policy gradients // IEEE Robotics and Automation Letters. – 2021. – Vol. 6(3). – P. 5340-5347.
- [Kloder et al., 2006] Kloder S., Hutchinson S. Path planning for permutation-invariant multirobot formations // IEEE Transactions on Robotics. – IEEE, 2006. – Vol. 22(4). – P. 650-665.
- [Le et al., 2021] Le D., Plaku E. Multi-robot motion planning with unlabeled goals for mobile robots with differential constraints // 2021 IEEE international conference on robotics and automation. – IEEE, 2021.
- [Okumura et al., 2022] Okumura K., Défago X. Solving simultaneous target assignment and path planning efficiently with time-independent execution // Proceedings of the international conference on automated planning and scheduling (ICAPS 2022). – 2022. – Vol. 32. – P. 270-278.
- [Solovey et al., 2015] Solovey K. et al. Motion planning for unlabeled discs with optimality guarantees // Proceedings of robotics: Science and systems (RSS 2015). – 2015.

- [**Stern et al., 2019**] Stern R. et al. Multi-agent pathfinding: Definitions, variants, and benchmarks // Proceedings of the 12th annual symposium on combinatorial search (SoCS 2019). – 2019. – P. 151-158.
- [**Turpin et al., 2014**] Turpin M. et al. Goal assignment and trajectory planning for large teams of interchangeable robots // Autonomous Robots. – 2014. – Vol. 37(4). – P. 401-415.
- [**Wurman et al., 2008**] Wurman P.R., D'Andrea R., Mountz M. Coordinating hundreds of cooperative, autonomous vehicles in warehouses // AI magazine. – 2008.
- [**Yu et al., 2013**] Yu J., LaValle S.M. Multi-agent path planning and network flow // Proceedings of the workshop on the algorithmic foundations of robotics (WAFR 2013). – 2013. – P. 157-173.