ДОКЛАДЫ РОССИЙСКОЙ АКАДЕМИИ НАУК. МАТЕМАТИКА, ИНФОРМАТИКА, ПРОЦЕССЫ УПРАВЛЕНИЯ, 2022, том 508, с. 88–93

ПЕРЕДОВЫЕ ИССЛЕДОВАНИЯ В ОБЛАСТИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И МАШИННОГО ОБУЧЕНИЯ

УДК 004.8

МЕТОДЫ ПЛАНИРОВАНИЯ И ОБУЧЕНИЯ В ЗАДАЧАХ МНОГОАГЕНТНОЙ НАВИГАЦИИ

© 2022 г. К. С. Яковлев[1,2,*], А. А. Андрейчук[1], А. А. Скрынник[1], А. И. Панов[1,2]

Представлено академиком РАН В.Б. Бетелиным
Поступило 28.10.2022 г.
После доработки 31.10.2022 г.
Принято к публикации 03.11.2022 г.

Задача многоагентной навигации возникает, с одной стороны, во множестве прикладных областей. Классический пример – автоматизированные склады, на которых одновременно функционирует большое число мобильных роботов-сортировщиков товаров. С другой стороны, эта задача характеризуется отсутствием универсальных методов решения, удовлетворяющих одновременно многим (зачастую – противоречивым) требованиям. Примером таких критериев могут служить гарантия отыскания оптимальных решений, высокое быстродействие, возможность работы в частично-наблюдаемым средах и т.д. В настоящей работе приведен обзор современных методов решения задачи многоагентной навигации. Особое внимание уделяется различным постановкам задачи. Рассматриваются различия и вопросы применимости обучаемых и необучаемых методов решения. Отдельно приводится анализ экспериментальных программных сред, необходимых для реализации обучаемых подходов.

Ключевые слова: планирование пути, эвристический поиск, обучение с подкреплением, многоагентные системы

DOI: 10.31857/S2686954322070220

1. ВВЕДЕНИЕ

Задача многоагентной навигации в общем виде формулируется следующим образом. Группа мобильных агентов (например, мобильных роботов или персонажей в виртуальной среде) функционирует в общем пространстве, при этом каждому агенту необходимо переместиться в известное ему целевое положение, избегая столкновений как с другими агентами, так и со статическими и стохастическими препятствиями. В последнее время интерес к методам решения этой задачи существенно возрос, в основном в связи с их востребованностью в области складской и сервисной робототехники [1] и в интеллектуальных транспортных системах [2].

Различные допущения, принимаемые на этапе формализации задачи, оказывают существенное влияние на выбор методов решения. Так, одной из наиболее распространенных и активно изучаемых формализаций является так называемая классическая задача многоагентного планирования (Classical MAPF). В этой задаче подразумевается существование централизованного контроллера, который обладает полной информацией о состоянии среды и всех агентов (полная наблюдаемость). При этом время считается дискретным, т.е. за один такт каждый агент может совершить действие перемещения либо ожидания. Пространство также дискретизируется в виде графа, т.е. считается, что агенты могут перемещаться лишь вдоль ребер априори заданного графа, а совершать действия ожидания только в его узлах. На практике обычно используются четырехсвязные графы – решетки (grids) [3]. Известно множество вариаций такой графовой, централизованной постановки задачи. Например, в [4] рассматривается вариант, когда цели агентов не фиксированы, т.е. распределение агентов по целевым положения является частью решения задачи. В работе [5] предполагается, что целей у каждого агента может быть несколько и агенты должны последовательно их посетить. В [6] рассматривается непрерывная задача, когда после достижения одной цели агентов ему сразу же назначается другая (заранее не известная). В целом, несмотря на различия в деталях постановок, централизованные варианты задачи многоагентной навигации обычно решаются классически-

[1] *Институт искусственного интеллекта AIRI, Москва, Россия*
[2] *Федеральный исследовательский центр "Информатика и управление" Российской академии наук, Москва, Россия*
*****E-mail: Yakovlev@airi.net*

ми, необучаемыми алгоритмами, основанными либо на эвристическом поиске в пространстве состояний (в том или ином виде) [8–10], либо на сведении исходной задачи к классическим задачам компьютерных наук, например к задаче о выполнимости булевых формул (SAT) [11], либо задаче о потоках [12].

Помимо задач многоагентной навигации, в которых предполагается полная наблюдаемость и централизованное управление, интерес, в том числе с практической точки зрения, вызывает альтернативная постановка, когда централизованный планировщик отсутствует, а агенты могут наблюдать среду (в том числе других агентов) лишь в определенном радиусе вокруг себя, т.н. частичная наблюдаемость. Такая задача логично формализуется в виде задачи последовательного принятия решений, когда в каждый такт времени каждый агент выбирает для исполнения одно действие, опираясь на текущее наблюдение (и, возможно, на историю наблюдений и взаимодействий со средой). Так же логичным является тот факт, что для решения задачи в такой постановке активно используются методы обучения с подкреплением [13].

Рассмотрим далее методы решения обоих классов задач многоагентной навигации более подробно.

2. НЕОБУЧАЕМЫЕ (КЛАССИЧЕСКИЕ) МЕТОДЫ

Необучаемые методы решения задачи многоагентной навигации обычно применяются, когда рассматривается вариант с полной наблюдаемостью, централизованным контроллером и графовой дискретизацией рабочей области агентов. Задача состоит в том, чтобы построить совокупность неконфликтных траекторий, а именно путей на графе, включающих возможные действия-ожидания в вершинах. Известно, что, с одной стороны, решить подобную задачу, в случае неориентированного графа, можно за полиномиальное время [14], с другой, получение оптимальных решений относится к классу NP-Hard [15]. Если же граф направленный, то и получение неоптимального решения – это NP-трудная задача [16].

Известны способы решения этой задачи, основывающиеся на ее сведении к другим известным задачам компьютерных наук. Так, в [11] задача многоагентного планирования (ЗМАП) сводится к SAT-задаче, в [17] к задаче целочисленного программирования, в [12] к задаче о потоках. Среди подобных методов, более других распространены те, что сводят ЗМАП к SAT. Вероятная причина состоит в том, что для SAT-задачи известно множество эффективных решателей, в результате чего скорость решения исходной задачи также достаточно высока. Также стоит упомянуть о следующей аналогии. ЗМАП при определенных допущениях можно рассматривать как задачу об игре в пятнашки (15 puzzle game). Такой подход используется современными алгоритмами, например, Push and Rotate [18], нацеленными на быстрое получение неоптимальных решений.

Другим способом решения ЗМАП является применение алгоритмов, осуществляющих непосредственно поиск на графе. Очевидно, что для повышения эффективности используются эвристические версии поиска. Классическим алгоритмом эвристического поиска является алгоритм A* [7], с определенными модификациями он может применяться для нахождения оптимальных решений ЗМАП [8], однако в целом этот подход не слишком эффективен, т.к. он по сути рассматривает всех агентов как единого мета агента и осуществляет поиск в комбинированном пространстве с коэффициентом ветвления, который экспоненциально зависит от числа агентов. Во избежание комбинаторного взрыва, применяются различные техники разъединения (decoupled search). К примерам подобных алгоритмов можно отнести CBS [9] и M* [10]. Оба алгоритма гарантируют оптимальность разыскиваемых решений и имеют множество модификаций, среди которых можно выделить модификации, направленные на повышение вычислительной эффективности при сохранении оптимальности решения [19], [20], модификации, позволяющие пренебречь (trade-off) оптимальностью в пользу вычислительной эффективности [21], а также модификации, позволяющие решать ЗМАП в менее строгих ограничениях, например, в непрерывном времени [22].

Еще одним подходом, к решению ЗМАП, базирующемся на эвристическом поиске, является так называемое приоритизированное планирование [23]. Здесь каждому агенту назначается приоритет, а затем ищутся лишь индивидуальные пути, при этом все ранее спланированные траектории считаются неизменяемыми (другими словами – динамическими препятствиями для очередного агента). С теоретической точки зрения такой подход не только не гарантирует оптимальности, но даже не дает гарантий, что решение задачи будет найдено, если оно существует. Тем не менее для определенного класса задач такую гарантию можно дать [24]. Более того, на практике приоритизированные алгоритмы находят решения, очень близкие к оптимальным в очень большом числе случаев, расходуя при этом кратно меньше вычислительных ресурсов. Именно поэтому алгоритмы этого класса очень часто применяются в робототехнике [25].

3. ОБУЧАЕМЫЕ МЕТОДЫ

Известно несколько вариантов применения методов машинного обучения в контексте ЗМАП с полной наблюдаемостью и централизованным контроллером. Во-первых, эти методы могут использоваться для выбора алгоритма многоагентного планирования, наиболее подходящего под конкретную задачу (карту, расположение агентов) [26, 27]. Во-вторых, методы машинного обучения могут использоваться для выучивания различных эвристических правил выбора, присутствующих в классических алгоритмах решения ЗМАП [28, 29]. Также в последние несколько лет широкое распространение получили методы обучения с подкреплением, которые позволяют решать задачу многоагентного планирования в децентрализованной и в частично-наблюдаемых постановках. В одной из первых работ на эту тему [30] была представлена обучаемая стратегия, называемая PRIMAL, которая позже была улучшена и обобщена на случай непрерывного поиска [31], когда после достижения цели агент не завершает эпизод, а получает новую задачу. Оба алгоритма использовали демонстрационные траектории, сгенерированные поисковым алгоритмом ODrM* [32]. Алгоритмы семейства PRIMAL используют сложную функцию вознаграждения и значительное число частных допущений, касающихся конкретных условий и карт (domain knowledge). Например, дополнительный штраф за конфликты или предположение о том, что в локальное наблюдение входят не только позиции других агентов, но и их цели. Похожие допущения были использованы в работе [33], которая предлагает еще один обучаемый алгоритм для решения ЗМАП, но уже для более сложных динамических моделей агентов (например, таких как квадрокоптеры). Обучаемые методы, которые используют полную информацию о статических элементах среды (глобальная информация о положениях других агентов им не доступна) были предложены в работах [34, 35].

Помимо алгоритмов, которые были разработаны специально для задач многоагентного планирования существует ряд универсальных подходов многоагентного обучения с подкреплением, которые могут быть использованы при решении ЗМАП. Из большого набора классических алгоритмов одноагентного обучения (такое обучение еще называют независимым), хорошо себя зарекомендовал для частично-наблюдаемых и многоагентных задач подход градиента стратегии, например, одна из популярных реализаций – алгоритм оптимизации ближайшей стратегии (PPO) [36–38]. Вторым направлением является использование централизованного обучения при обучении кооперативных стратегий. Алгоритмы из этого направления обычно обучаются централизованно, используя глобальную информацию о среде, а тестируются децентрализованно. Так, QMIX [39] использует гиперсети для обучения отдельных стратегий через смешивающую сеть полезности. Каждая отдельная сеть при обучении получает только локальное наблюдение, и оптимизируется гиперсетью, которая использует доступ к глобальному состоянию. Алгоритмы обучения MADDPG [40] (обучение по отложенному опыту) и MAPPO [41] (обучение по актуальному опыту) используют централизованную сеть критика. Это общий подход при обучении, когда критик использует глобальное состояние среды для более качественного обучения аппроксиматора функции полезности. Актор, который определяет стратегию выбора действий, получает на вход только частичное наблюдение, но неявно использует общее наблюдение, пользуясь оценками критика. Алгоритм FACMAC [42] является комбинацией алгоритмов MADDPG и QMIX, что позволяет применять его как для дискретного пространства действий, используя преобразования Гюмбеля, так и для непрерывного.

Алгоритмы направления MARL достаточно сильно оптимизированы под ряд сред, которые уже стали классикой для их тестирования, например SMAC [43], использующий игру Starcraft 2. В отличие от одноагентного обучения с подкреплением в открытом доступе намного меньше готовых реализаций, а те, что существуют, подходят лишь для решения довольно простых задач. Основными причинами этого являются медленные реализации, не использующие распараллеливание и рассчитанные лишь на несколько миллионов шагов в среде, использование простых полносвязных архитектур в качестве аппроксиматоров. Из-за этого большинство исследователей отдают предпочтение использованию известных децентрализованных подходов. Но это приводит к другой крайности – предложенные алгоритмы эксплуатируют знания предметной области, что ограничивает их применимость для широкого класса задач навигации.

Одним из перспективных направлений в создании более продвинутых методов решения ЗАМП могут служить методы обучения с подкреплением на основе модели [44]. Предсказание поведения других агентов, учет этой модели динамики в построении собственной стратегии агента могут оказаться особенно полезными в гетерогенной группе агентов, где у каждого из них может быть разная собственная стратегия [45]. Еще одной незакрытой нишей в MARL является использование демонстраций при обучении. Действительно, использование демонстраций может существенно ускорить обучение, а также позволить использовать современный трансформенные и диффузионные модели. Это особенно

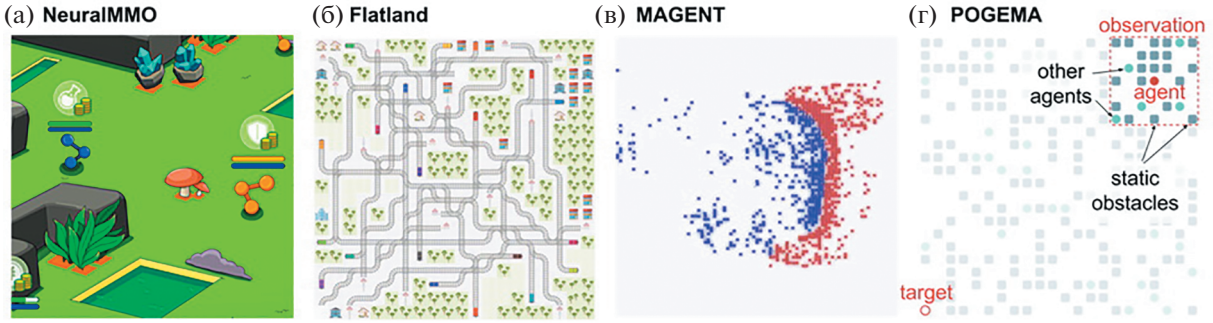

Рис. 1. Примеры сред, используемых для работы с обучаемыми методами решения мультиагентных задач.

актуально для задач навигации, для которых существуют сильные планировочные алгоритмы.

4. ЭКСПЕРИМЕНТАЛЬНЫЕ СРЕДЫ ДЛЯ ТЕСТИРОВАНИЯ АЛГОРИТМОВ

Экспериментальные онлайн среды почти не используются исследователями необучаемых подходов решения ЗМАП, т.к. этим подходы не предполагают обучение путем взаимодействия со средой. Обратная картина наблюдается в сообществе обучения с подкреплением, где существует большое количество различных сред, однако большинство из них являются игровыми и предполагают большое количество дополнительных особенностей, не связанных с задачами многоагентной навигации (например, инвентарь, наличие противодействующих оппонентов и т.п.) (см. рис. 1).

Примером игровой среды может служить NeuralMMO [46], которая является упрощенной версией многопользовательской сетевой игры, в которой группа агентов решает задачу выживания и накопление ресурсов. Команда из 8 агентов соревнуется с другими 15 командами на процедурно генерируемой карте 128 на 128 клеток. Среда является частично наблюдаемой, но агенты могут коммуницировать между собой. Несмотря на то что указанная задача является достаточно сложной, она далека от практического применения, и требует преимущественно реактивного выбора действий на основе системы правил, нежели планирования и навигации.

Одной из наиболее известных сред именно для задачи MAPF является среда Flatland [48] – упрощенная, однако, реалистичная среда для решения задач составления расписания сети железных дорог. Здесь агенты – поезда – должны двигаться от одной станции к другой по путям с односторонним движением, избегая конфликтов друг с другом. В рамках данной задачи было проведено несколько соревнований, целями которых было исследование алгоритмов обучения с подкреплением. Однако оказалось, что доступ к полному состоянию среды дает существенное преимущество подходам планирования и перепланирования [49]. Еще одним недостатком данной среды является то, что она работает очень медленно (около 200 шагов в секунду для небольших карт) при использовании режима наблюдений, предназначенного для обучаемых алгоритмов.

Среда MAGENT – одна из наборов сред библиотеки PettingZoo [47]. Среда предназначена для моделирования роевого поведения агентов, которые могут не только перемещаться из одного места в другое, но и взаимодействовать друг с другом различными способами. Реализация на C++ существенно ускоряет процесс взаимодействия, однако данная среда имеет ограниченный набор сценариев (типов карт) и не имеет интерфейса для тестирования решений, не основанных на обучении с подкреплением.

Наиболее подходящая для задач MAPF среда POGEMA [50] специально создана для задач в частично-наблюдаемой постановке для клеточных карт. Авторы делают особый упор на то, что агенты получают информацию только из ограниченного пространства вокруг себя и не могут передавать какую-либо информацию друг другу, что существенно усложняет задачу как для планировочных алгоритмов, так и для обучаемых. Главными достоинствами данной среды являются ее гибкость и скорость работы. POGEMA позволяет использовать любые созданные пользователем карты препятствий, поддерживает три режима, которые определяются тем, что происходит после достижения агентом цели: агент получает новую цель (непрерывный поиск пути), исчезающие (после достижения цели) агенты и неисчезающие до конца эпизода агенты.

5. ЗАКЛЮЧЕНИЕ

Методы решения задачи многоагентной навигации активно развиваются в последнее время в связи с их востребованностью в различных прак-

тических областях (складская робототехника, транспортные системы и пр.). В условиях централизованного управления и полной наблюдаемости обычно применяются необучаемые методы, основанные либо на эвристическом поиске, либо на сведении задачи многоагентного планирования к другим классическим задачам в области компьютерных наук (SAT-задача, задача о потоках и пр.). В случае, когда централизованный контроллер отсутствует и/или агентам недоступна полная информация об окружающей среде, то зачастую применяются методы обучения с подкреплением, основанные либо на адаптации известных методов поиска стратегии для индивидуальных агентов, либо на парадигме "централизованное обучение – децентрализованное исполнение". Наиболее перспективным (и наименее исследованным) по мнению авторов может являться комбинированный подход, использующий как методы обучения с подкреплением, так и классические методы планирования (эвристический поиск и др.).

СПИСОК ЛИТЕРАТУРЫ

1. *Ma H., Koenig S.* AI buzzwords explained: Multi-agent path finding (MAPF). *AI Matters*. 2017. V. 3 (3). P. 15–19.
2. *Morris R., Pasareanu C.S., Luckow K., Malik W., Ma H., Kumar T.S., Koenig S.* Planning, scheduling and monitoring for airport surface operations. Workshops at the 30th AAAI Conference on Artificial Intelligence, 2016.
3. *Yap P.* Grid-based path-finding. Conference of the canadian society for computational studies of intelligence, Springer, Berlin, Heidelberg, 2002. P. 44–55.
4. *Ma H., Koenig S.* Optimal Target Assignment and Path Finding for Teams of Agents. Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, 2016. P. 1144–1152.
5. *Liu M., Ma H., Li J., Koenig S.* Task and Path Planning for Multi-Agent Pickup and Delivery. Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, 2019. P. 1152–1160.
6. *Li J., Tinka A., Kiesel S., Durham J.W., Kumar T.S., Koenig S.* Lifelong multi-agent path finding in large-scale warehouses. Proceedings of the 30th AAAI Conference on Artificial Intelligence, 2021. P. 11272–11281.
7. *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, 1968. V. 4(2). P. 100–107.
8. Finding optimal solutions to cooperative pathfinding problems. Proceedings of the 24th AAAI Conference on Artificial Intelligence, 2010. P. 173–178.
9. *Sharon G., Stern R., Felner A., Sturtevant N.R.* Conflict-based search for optimal multi-agent pathfinding. Artificial Intelligence, 2015, 219. P. 40–66.
10. *Wagner G., Choset H.* M*: A complete multirobot path planning algorithm with performance bounds. Proceedings of the 2011 IEEE/RSJ international conference on intelligent robots and systems, 2011. P. 3260–3267.
11. *Surynek P., Felner A., Stern R., Boyarski E.* Efficient SAT approach to multi-agent path finding under the sum of costs objective. Proceedings of the 22nd European conference on artificial intelligence, 2016. P. 810–818.
12. *Yu J., LaValle S.M.* Multi-agent path planning and network flow. Algorithmic foundations of robotics X, 2013. P. 157–173.
13. *Sutton R.S., Barto A.G.* Reinforcement learning: An introduction. 2nd edn. Bradford Books, 2018.
14. *Kornhauser D., Miller G., Spirakis P.* Coordinating Pebble Motion on Graphs, The Diameter of Permutation Groups, And Applications. The 25th Annual Symposium on Foundations of Computer Science, 1984. P. 241–250.
15. *Ratner D., Warmuth M.* The (n2−1)-puzzle and related relocation problems. Journal of Symbolic Computation, 1990. V. 10 (2). P. 111–137.
16. *Nebel B.* On the computational complexity of multi-agent pathfinding on directed graphs. Proceedings of the 20th International Conference on Automated Planning and Scheduling, 2020. P. 212–216.
17. *Yu J., LaValle S.M.* Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics. IEEE Transactions on Robotics, 2016. V. 32 (5). P. 1163–1177.
18. *De Wilde B., Ter Mors A.W., Witteveen C.* Push and rotate: a complete multi-agent pathfinding algorithm. Journal of Artificial Intelligence Research, 2014. V. 51. P. 443–492.
19. *Boyarski E., Felner A., Stern R., Sharon G., Tolpin D., Betzalel O., Shimony E.* ICBS: improved conflict-based search algorithm for multi-agent pathfinding. Proceedings of the 24th International Conference on Artificial Intelligence, 2015. P. 740–746.
20. *Li J., Harabor D., Stuckey P. J., Ma H., Koenig S.* Symmetry-breaking constraints for grid-based multi-agent path finding. Proceedings of the 33rd AAAI Conference on Artificial Intelligence, 2019. P. 6087–6095.
21. *Barer M., Sharon G., Stern R., Felner A.* Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem. Proceedings of the 7th Annual Symposium on Combinatorial Search, 2014.
22. *Andreychuk A., Yakovlev K., Surynek P., Atzmon D., Stern R.* Multi-agent pathfinding with continuous time. Artificial Intelligence, 2022. V. 305. P. 103662.
23. *Erdmann M., Lozano-Perez T.* On multiple moving objects. Algorithmica, 1987. V. 2 (1). P. 477–521.
24. *Cap M., Vokrinek J., Kleiner A.* Complete decentralized method for on-line multi-robot trajectory planning in well-formed infrastructures. Proceedings of the 25th International conference on automated planning and scheduling, 2015. P. 324–332.
25. *Yakovlev K., Andreychuk A.* Any-Angle Pathfinding for Multiple Agents Based on SIPP Algorithm. Proceedings of the 17th International Conference on Automated Planning and Scheduling, 2017. P. 586–594.
26. *Kaduri O., Boyarski E. Stern R.* Algorithm selection for optimal multi-agent pathfinding. *Proceedings of the In-*

ternational Conference on Automated Planning and Scheduling, 2020, June, 30. P. 161–165.

27. *Ren J., Sathiyanarayanan V., Ewing E., Senbaslar B., Ayanian N.* MAPFAST: A Deep Algorithm Selector for Multi Agent Path Finding using Shortest Path Embeddings. *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, 2021, May. P.* 1055–1063.

28. *Li J., Chen Z., Harabor D., Stuckey P.J. Koenig S.* MAPF-LNS2: Fast Repairing for Multi-Agent Path Finding via Large Neighborhood Search. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

29. *Huang T., Koenig S., Dilkina B.* Learning to resolve conflicts for multi-agent path finding with conflict-based search. *Proceedings of the AAAI Conference on Artificial Intelligence,* 2021*, May,* V. 35, № 13. P. 11246–11253.

30. *Sartoretti G., Kerr J., Shi Y., Wagner G., Kumar T.S., Koenig S., Choset H.* Primal: Pathfinding via reinforcement and imitation multi-agent learning. IEEE Robotics and Automation Letters, 2019. V. 4 (3). P. 2378–2385.

31. *Damani M., Luo Z., Wenzel E., Sartoretti G.* PRIMAL $ _2 $: Pathfinding via reinforcement and imitation multi-agent learning-lifelong. IEEE Robotics and Automation Letters, 2021. V. 6 (2). P. 2666–2673.

32. *Ferner C., Wagner G., Choset H.* ODrM* optimal multirobot path planning in low dimensional search spaces. 2013 IEEE International Conference on Robotics and Automation, 2013, May. P. 3854–3859.

33. *Riviere B., Hönig W., Yue Y., Chung S.J.* Glas: Global-to-local safe autonomy synthesis for multi-robot motion planning with end-to-end learning. IEEE Robotics and Automation Letters, 2020. V. 5 (3). P. 4249–4256.

34. *Liu Z., Chen B., Zhou H., Koushik G., Hebert M., Zhao D.* Mapper: Multi-agent path planning with evolutionary reinforcement learning in mixed dynamic environments. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2020, October. P. 11748–11754.

35. *Wang B., Liu Z., Li Q., Prorok A.* Mobile robot path planning in dynamic environments through globally guided reinforcement learning. IEEE Robotics and Automation Letters, 2020. V. 5 (4). P. 6932–6939.

36. *Schulman J., Wolski F., Dhariwal P., Radford A., Klimov O.* Proximal policy optimization algorithms, 2017, arXiv preprint arXiv:1707.06347.

37. *Skrynnik A., Andreychuk A., Yakovlev K., Panov A.* Pathfinding in stochastic environments: learning vs planning. PeerJ Computer Science, 2022. V. 8. P. e1056.

38. *Berner C., Brockman G., Chan B., Cheung V., Dębiak P., Dennison C., Farhi D., Fischer Q., Hashme S., Hesse C, Józefowicz R.* Dota 2 with large scale deep reinforcement learning, 2019, arXiv preprint arXiv:1912.06680.

39. *Rashid T., Samvelyan M., Schroeder C., Farquhar G., Foerster J., Whiteson S.* Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. International conference on machine learning, 2018, July. P. 4295–4304. PMLR.

40. *Lowe R., Wu Y.I., Tamar A., Harb J., Pieter Abbeel O., Mordatch I.* Multi-agent actor-critic for mixed cooperative-competitive environments. Advances in neural information processing systems, 2017, 30.

41. *Yu C., Velu A., Vinitsky E., Wang Y., Bayen A., Wu Y.* The surprising effectiveness of ppo in cooperative, multi-agent games, 2021, arXiv preprint arXiv:2103.01955.

42. *Peng B., Rashid T., Schroeder de Witt C., Kamienny P.A., Torr P., Böhmer W., Whiteson S.* Facmac: Factored multi-agent centralised policy gradients. Advances in Neural Information Processing Systems, 2021. V. 34. P. 12208–12221.

43. *Samvelyan M., Rashid T., De Witt C.S., Farquhar G., Nardelli N., Rudner T.G., Hung C.M., Torr P.H., Foerster J., Whiteson S.* The starcraft multi-agent challenge, 2019, arXiv preprint arXiv:1902.04043.

44. *Moerland T.M., Broekens J., Jonker C.M.* Model-based Reinforcement Learning: A Survey," pp. 421–429, Jun. 2020, [Online]. Available: http://arxiv.org/abs/2006.16712.

45. *Skrynnik A., Yakovleva Y., Davydov D., Yakovlev K., Panov A.I.* Hybrid Policy Learning for Multi-Agent Pathfinding, IEEE Access, 2021. V. 9. P. 126034–126047, https://doi.org/10.1109/ACCESS.2021.3111321

46. *Suarez J., Du Y., Isola P., Mordatch I.* Neural MMO: A massively multiagent game environment for training and evaluating intelligent agents, 2019, arXiv preprint arXiv:1903.00784.

47. *Terry J., Black B., Grammel N., Jayakumar M., Hari A., Sulliva, R., Santos L.S., Dieffendahl C., Horsch C., Perez-Vicente R., Williams N.* Pettingzoo: Gym for multi-agent reinforcement learning. Advances in Neural Information Processing Systems, 2021. V. 34. P. 15032–15043.

48. *Laurent F., Schneider M., Scheller C., Watson J., Li J., Chen Z., Zheng Y., Chan S.H., Makhnev K., Svidchenko O., Egorov V.* Flatland competition 2020: MAPF and MARL for efficient train coordination on a grid world. NeurIPS 2020 Competition and Demonstration Track, 2021, August. P. 275–301. PMLR.

49. *Li J., Chen Z., Zheng Y., Chan S.H., Harabor D., Stuckey P.J., Ma H., Koenig S.* Scalable rail planning and replanning: Winning the 2020 flatland challenge. Proceedings of the International Conference on Automated Planning and Scheduling, 2021, May. V. 31. P. 477–485.

50. *Skrynnik A., Andreychuk A., Yakovlev K., Panov A.I.* POGEMA: Partially Observable Grid Environment for Multiple Agents, 2022, arXiv preprint arXiv:2206.10944.