Министерство науки и высшего образования Российской Федерации
Московский физико-технический институт
(национальный исследовательский университет)
Физтех-школа прикладной математики и информатики
Магистерская программа
«Методы и технологии искусственного интеллекта»

Заварзин Никита Юрьевич

Методы многоагентного планирования траекторий с учётом кинодинамических ограничений

09.04.01 — Информатика и вычислительная техника

Литературный обзор

Научный руководитель:
доцент, к.ф.-м.н.
Яковлев Константин Сергеевич

Москва
2024 г.

Содержание

1 Введение **3**

2 Постановка задачи **4**

3 Обзор существующих подходов **5**

3.1 Методы одноагентного планирования траектории 5

3.2 Переход от одноагентной к многоагентной постановке задачи 7

3.3 Методы многоагентного планирования с учётом кинодинамических ограничений 9

1 Введение

Планирование движения — фундаментальная задача робототехники, методы решения которой обеспечивают работоспособность роботизированных систем в огромном числе отраслей: от сельского хозяйства до промышленного производства, от автономной городской навигации до здравоохранения.

Неформально говоря, задача планирования состоит в том, чтобы найти траекторию, которая переводила бы нашу систему из начальной точки в конечное множество, избегая при этом препятствий и минимизируя возможные затраты.

Ограничения, накладываемые на систему, как правило, разделяют на три типа.

1. Геометрические: обязывают нас искать решение, лежащее в некотором заданном множестве. Например, решение, не выходящее за пределы куба с центром в начальной точке.

2. Кинематические: при построении пути требуют, чтобы тот удовлетворял кинематическим связям рассматриваемой физической системы. Сюда относятся движение без проскальзывания, так называемое условие нерастяжимости твёрдого тела и т. п.

3. Кинодинамические: принуждают траектории соответствовать не только кинематическим ограничениям, но и уравнениям выбранной модели динамики системы (из-за того, что мы больше не можем пренебрегать инерцией объекта, добавляются условия на ускорение). Яркими представителями являются второй закон Ньютона и уравнение динамики вращательного движения.

По сложности среды задача планирования классифицируется следующим образом.

1. Имеются только статические препятствия.
   Обычно форма и расположение известны заранее и нам достаточно один раз их промоделировать, а дальше сконцентрироваться исключительно на построении траектории.

2. Кроме статических ещё возможны динамические препятствия.
   Форма и расположение таких препятствий считываются с датчиков системы и постоянно обновляются. Для успешного маневрирования в условиях меняющейся среды агенту помимо планирования собственного движения необходимо прогнозировать действия других участников и корректировать траекторию с целью избегания столкновений.

3. Многоагентная постановка.
   Предполагает координацию действий нескольких агентов для достижения общей цели, в ходе выполнения которой для получения оптимальных результатов агентам требуется постоянно согласовывать планы и синхронизировать свою работу.

В данном обзоре исследуем методы многоагентного планирования для задач с кинодинамическими ограничениями.

2 Постановка задачи

Пусть мы имеем n агентов и ряд их кинодинамических ограничений, представляющих собой систему дифференциальных уравнений

$$\begin{cases} \dot{x}(t) = F(x(t), u(t)), \\ x(t_0) = x_0, \\ x(t_1) \in X_{\text{goal}}. \end{cases} \tag{1}$$

Где $x(t) = \Big(x_1(t), \ldots, x_n(t)\Big)^T$, $F(x,u) = \Big(f_1(x,u), \ldots, f_n(x,u)\Big)^T$, $u(t) = \Big(u_1(t), \ldots u_n(t)\Big)^T$. Здесь вектор-функция $x_i(t) \in X_i$ — состояние i-го агента в момент времени t, $u_i(t) \in \mathcal{U}_i$ — управляющее воздействие на i-го агента в момент времени t, $f_i(x,u) \in C^1$ — функция перехода, t_0 — начальный момент времени, t_1 — конечный момент времени. Кроме того, нам заданы начальные позиции x_0, целевое множество значений вектора состояний X_{goal}, множество допустимых управлений $\mathcal{U} = \mathcal{U}_1 \times \cdots \times \mathcal{U}_n$. За X_{obs} обозначим открытое подмножество $X = X_1 \times \cdots \times X_n$, определяющее область препятствий. X_{free} же, задающее доступную для манёвров область, примем равной $X \setminus X_{\text{obs}}$.

Задача состоит в том, чтобы при заданной области X, известном пространстве препятствий X_{obs}, начальном состоянии x_0 и обозначенной целевой области X_{goal} найти программное управление $u(\cdot) \in \mathcal{U}$, которое минимизирует функционал

$$\mathcal{J}(u(\cdot)) = \int_{t_0}^{t_1} h(x(t), u(t), t)\,\mathrm{d}t, \tag{2}$$

на беспрепятственных траекториях системы (1). Беспрепятственными будем называть решения $x(t) \in X_{\text{free}}, \forall t \in [t_0, t_1]$. Также возможны ограничения в виде неравенств:

$$g_k(x(t), u(t), t) \leqslant 0, \quad k = \overline{1, m}, \tag{3}$$

с $g_k(x(t), u(t), t) \in C^1$.

3 Обзор существующих подходов

3.1 Методы одноагентного планирования траектории

Для начала рассмотрим существующие подходы к задаче планирования траектории для одного агента, возможно представленные идеи удастся перенести на многоагентную постановку задачи. Традиционно, относительно используемого принципа принято выделять следующие классы алгоритмов одноагентного планирования[1]:

1. Эвристический поиск (Dijkstra, A*, Theta*).
   Заключается в дискретизации окружающего пространства графом с применением в дальнейшем алгоритмов поиска наилучшего пути в нём на основе функции стоимости и эвристической функции. Данный метод эффективен в средах с заранее известными препятствиями, в них он гарантированно находит кратчайший путь, но в случаях, когда агенту необходимо объехать преграду, которая не учитывалась при проектировании дорожной карты или проехать по участку, не охваченному графом дорожной сети, возникают трудности.

2. Методы на основе сэмплирования (RRG, RRT*[2]).
   Алгоритмы, основанные на этом принципе, постепенно строят дерево по случайным выборкам из пространства поиска, тем самым последовательно заполняя его. При этом на каждой итерации предпринимается попытка соединить имеющееся дерево с конечным множеством. Данный подход характеризуется высокой скоростью работы и приспосабливанием к изменяющейся обстановке, однако в то же время требует постобработки в виде сглаживания решения.

3. Оптимизационные алгоритмы (Potential Fields, Elastic Band[3]).
   Обычно для методов такого типа изначально задан глобальный план (к примеру функция распределения потенциала в пространстве или ломанная, соединяющая начальную и конечную точки и т. п.) и задача состоит в том, чтобы построить гладкий путь, оптимизируя некоторый функционал по исходной концепции. Основными минусами являются необходимая предобработка и сходимость к локальным минимумам, выигрыш же в непрерывности получаемых управлений, качестве решений.

4. Эволюционные подходы (Genetic algorithm[4], Evolutionary algorithm[5]).
   Эволюционные алгоритмы моделируют процесс биологической эволюции, включая такие операции, как отбор, скрещивание и мутация, для поиска оптимальных траекторий в пространстве решений. Хорошо справляются с локальными минимумами, но требуют больших вычислительных затрат.

5. Принцип общего интеллекта (Particle Swarm Optimization[6], Ant Colony Optimization[7]).

Основывается на моделировании коллективного поведения групп агентов или частиц, которые сотрудничают и обмениваются информацией для поиска оптимального решения. Эти алгоритмы черпают вдохновение из природы, где стаи птиц, муравьиные колонии демонстрируют высокоэффективное коллективное поведение при решении сложных задач.

6. Бионические стратегии (Immunization algorithm, Artificial Fish Swarm).
   Сюда относят алгоритмы, которые имитируют более общие биологические процессы или поведение отдельных организмов. Крайне чувствительны к настройке, но способны находить решение в условиях неопределенности.

7. Методы на основе машинного обучения (DQN, PPO[8, 9]).
   Данные подходы весьма разнообразны по архитектуре нейросетей, входным данным и параметрам. Часто используются алгоритмы обучения с подкреплением, суть которых заключается во взаимодействии агента со средой и зависимости награды от действий агента. Отдельным классом являются так называемые end-to-end методы, которые в качестве входных данных принимают наблюдения среды, а в качестве выходных данных выдают траекторию, или сразу управляющие воздействия. Минусами таких подходов являются обучение моделей и сложность в интерпретации решения, но зато агент способен быстро адаптироваться к новой среде.

3.2 Переход от одноагентной к многоагентной постановке задачи

Давайте попробуем применить методы решения задачи одноагентного планирования для её многоагентного аналога. Начнём с тривиальных подходов, которые первыми приходят в голову:

1. "Независимое планирование" с последующей координацией.
   Все агенты, независимо друг от друга, используют один и тот же метод планирования для построения возможных решений. Далее, для избежания коллизий на исполнение траекторий накладываются временные ограничения.

2. "Приоритетное планирование".
   Каждому агенту выставляется приоритет. Далее в порядке приоритета агенты планируют путь выбранной стратегией, воспринимая уже построенные траектории как динамические препятствия.

3. "Системное планирование".
   Все агенты рассматриваются как единая система, а алгоритм планирования адаптируется для работы в высокоразмерном пространстве (где каждая размерность соответствует координатам отдельного агента).

К сожалению, используя первые два подхода мы можем прийти к ситуации, когда в отсутствии координации один из агентов выбирает путь, создающий "заторы" для остальных. То есть теряется оптимальность, а в некоторых случаях даже полнота (например взаимная блокировка). Третий же подход подразумевает экспоненциальный рост пространства состояний в зависимости от числа агентов, то есть избыточную для данной задачи вычислительную сложность. Поэтому вместо универсальной адаптации стратегий одноагентного планирования проанализируем уникальные.

Если не указано обратное, то представленные алгоритмы не только полны, но и даже оптимальны.

1. Многоагентный эвристический поиск (OD[10], HCA*).
   Характерным является модификация функции стоимости при пересечении с другим агентом, или же разделение пространства на области ответственности для каждого агента, чтобы избежать пересечений и оптимизировать процесс планирования.

2. Наиболее часто используемые обобщения методов сэмплирования, оптимизационных алгоритмов и эволюционных подходов на многоагентную постановку (так называемый Priority Planning [11], [12]).
   Суть заключается в разделении шага алгоритма на два этапа: этап динамического выставления приоритетов и этап локального перепланирования траекторий. В качестве

планировщика как раз берётся один из имеющихся алгоритмов одноагентного планирования, единственным требованием к которому является умение избегать динамические препятствия, движущиеся по известным траекториям. Отметим, что характеристики методов, их плюсы и минусы полностью совпадают с таковыми у используемых одноагентных планировщиков.

3. Принцип общего интеллекта ([13]).
   В данном случае подход адаптируется следующим образом: сначала глобально строятся траектории для каждого агента, затем по мере их продвижения местные конфликты решаются по праву первенства. Полнота и оптимальность в данном случае зависят от среды и настройки параметров алгоритма.

4. Методы на основе машинного обучения в задаче многоагентного планирования (MADQN, MAPPO).
   Каждая политика тренируется с учётом состояний и действий других агентов (посредством центрального критика). Центральный критик агрегирует состояния всех агентов, помогая моделировать взаимодействие между ними. Полнота зависит от награды и корректности модели среды.

К уже имеющемуся списку можно также добавить две стратегии, аналогов которым ранее не встречалось.

1. Поиск основанный на конфликтах (CBS[14]).
   Разделён на два уровня: первый рассматривает конфликт между агентами, определяет ограничения для их путей, создаёт дерево ограничений и разветвляет решение, чтобы устранить конфликты, второй — находит оптимальные индивидуальные пути для каждого агента, соблюдая ограничения, установленные на верхнем уровне.

2. Сведение к уже имеющимся задачам компьютерных наук (SAT, integer programming).
   Название говорит само за себя, в [15] задача многоагентного планирования сводится к SAT-задаче, в [16] к задаче целочисленного программирования, в [17] к задаче о потоках.

3.3 Методы многоагентного планирования с учётом кинодинамических ограничений

1. db-CBS (database-enhanced Conflict-Based Search) db-CBS является улучшением базового алгоритма CBS (Conflict-Based Search), который используется для поиска оптимальных путей для нескольких агентов, минимизируя общее количество конфликтов в дискретных средах. Основное отличие db-CBS заключается в использовании базы данных (database) для хранения и повторного использования ранее найденных решений для конфликтующих агентов. Это позволяет уменьшить количество операций поиска и повысить производительность при планировании.

   Основные этапы работы db-CBS:

   Разделение на высокоуровневую и низкоуровневую части: Высокоуровневая часть находит и разрешает конфликты между агентами (например, агент A и агент B пытаются занять одну клетку в одно время). Низкоуровневая часть планирует индивидуальные пути для каждого агента, учитывая ограничения, наложенные высокоуровневым планировщиком. Использование базы данных: Вместо пересчета новых путей для конфликтующих агентов, db-CBS проверяет, можно ли применить ранее найденное решение из базы данных. Если подходящего решения нет, оно добавляется в базу. Конфликты решаются за счет добавления ограничений, которые предотвращают их повторение. Полный. Может быть оптимальным, если используется правильная эвристика и ограничения. Ускоряет процесс планирования за счет использования повторного опыта.

2. SIPP-IP (Safe Interval Path Planning with Intervals of Probability) SIPP-IP является расширением классического алгоритма SIPP (Safe Interval Path Planning), который фокусируется на безопасном планировании пути для одного агента. В SIPP-IP добавляется учет вероятностных параметров для планирования в условиях неопределенности.

   Основные этапы работы SIPP-IP:

   Модель безопасных интервалов: SIPP разделяет временную шкалу на интервалы, в которых агент может безопасно перемещаться без конфликтов. Каждое состояние хранит информацию о допустимых временных интервалах. В SIPP-IP вводятся вероятностные модели, описывающие динамические препятствия или неопределенности в движении агентов. Каждый интервал ассоциируется с вероятностью успеха прохождения. Планировщик использует эвристические оценки, чтобы выбрать путь, который максимизирует вероятность успеха, минимизируя время. Полный для постановок с дискретным временем и заранее известной динамикой. Подходит для задач с динамическими препятствиями, где нужно учитывать вероятность столкновения.

3. Bezier Curve Optimization Bezier Curve Optimization применяется для плавного планирования траекторий, особенно в задачах, где важна непрерывность и кинодинамические

ограничения.

Основные этапы работы:

Кривая Безье задается контрольными точками, которые определяют форму траектории. Например, для квадратичной кривой используются три точки: начальная, конечная и одна промежуточная. Оптимизация контрольных точек: Цель состоит в выборе таких контрольных точек, чтобы траектория минимизировала определенный критерий (например, длину пути, время или отклонение от оптимального пути). Также учитываются ограничения: максимальная скорость, ускорение, и избегание препятствий. Траектория проверяется на соблюдение кинематических ограничений (например, радиуса поворота) и избегание столкновений. Не является полным в общем случае, так как оптимизация зависит от качества первоначального предположения. Часто используется в сочетании с глобальными планировщиками (например, RRT*) для поиска начальной траектории. Обеспечивает высокую плавность и соблюдение ограничений.

Список литературы

[1] L. Yang, P. Li, S. Qian, H. Quan, J. Miao, M. Liu, Y. Hu, and E. Memetimin, "Path Planning Technique for Mobile Robots: A Review."

[2] S. Karaman and E. Frazzoli, "Sampling-based Algorithms for Optimal Motion Planning," in Robotics: Science and Systems, 2011.

[3] S. Quinlan and O. Khatib, "Elastic Bands: Connecting Path Planning and Control," in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 1993.

[4] Wikipedia, "Genetic Algorithm." Available: `https://en.wikipedia.org/wiki/Genetic_algorithm`.

[5] Wikipedia, "Evolutionary Algorithm." Available: `https://en.wikipedia.org/wiki/Evolutionary_algorithm`.

[6] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in Proceedings of IEEE International Conference on Neural Networks (ICNN), 1995.

[7] M. Dorigo, M. Birattari, and T. Stützle, "Ant Colony Optimization," in IEEE Computational Intelligence Magazine, 2006.

[8] Habr, "Proximal Policy Optimization Explained." Available: `https://habr.com/ru/articles/561746/`.

[9] Wikipedia, "Proximal Policy Optimization." Available: `https://en.wikipedia.org/wiki/Proximal_policy_optimization`.

[10] T. Standley, "Finding Optimal Solutions to Cooperative Pathfinding Problems," in Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2010.

[11] H. Wang, "Multi-Robot Path Deconfliction through Prioritization by Path Prospects."

[12] H. Wang, "Combined Priority and Path Planning through a Double-layer Structure for Multiple Robots."

[13] "Improved Ant Colony Algorithms for Multi-agent Path Planning in Web3D Environment."

[14] G. Sharon, R. Stern, A. Felner, and N. Sturtevant, "Conflict-Based Search For Optimal Multi-Agent Path Finding," in Artificial Intelligence Journal, 2015.

[15] P. Surynek, A. Felner, R. Stern, and E. Boyarski, "Efficient SAT Approach to Multi-agent Path Finding Under the Sum of Costs Objective," in Proceedings of the European Conference on Artificial Intelligence (ECAI), 2016.

[16] J. Yu and S. M. LaValle, "Optimal Multirobot Path Planning on Graphs: Complete Algorithms and Effective Heuristics," in IEEE Transactions on Robotics, 2013.

[17] J. Yu and S. M. LaValle, "Multi-agent Path Planning and Network Flow," in Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2016.