

А.С. ЗАЙЦЕВ, К.С. ЯКОВЛЕВ
**ИНТЕРВАЛЬНЫЕ ВРЕМЕННЫЕ ОГРАНИЧЕНИЯ ДЛЯ
ПЛАНИРОВАНИЯ СОГЛАСОВАННЫХ ДВИЖЕНИЙ
МНОГОЗВЕННЫХ МАНИПУЛЯТОРОВ**

Зайцев А.С., Яковлев К.С. Интервальные временные ограничения для планирования согласованных движений многозвенных манипуляторов.

Аннотация. В работе рассматривается задача планирования согласованных движений для системы многозвенных роботов-манипуляторов. Одним из перспективных современных подходов к её решению является конфликтно-ориентированное планирование, позволяющее отказаться от построения комбинированного пространства поиска высокой размерности за счет перехода к последовательному решению совокупности задач более низкой размерности. Этот переход осуществляется путем задания пространственно-временных ограничений при наличии конфликта в индивидуальных планах манипуляторов и последующего перепланирования с учетом наложенных ограничений. К сожалению, известные к настоящему моменту способы наложения ограничений оперируют отдельными моментами времени, что снижает их эффективность на практике. В данной работе представляется новый алгоритм конфликтно-ориентированного планирования, основанный не на точечных, а на интервальных временных ограничениях – GECBS-T. С теоретической точки зрения, предлагаемый алгоритм гарантирует ограниченную суб-оптимальность конструируемых решений, т.е. при любом наперед заданном пользователем ограничении $w > 1$, стоимость решения GECBS-T не будет превышать стоимость оптимального решения более, чем в w раз. На практике же, предлагаемый алгоритм существенно превосходит аналогичные алгоритмы по скорости решения задач планирования, что подтверждается проведенными экспериментами в робототехническом симуляторе *tuJoso*, включающими 2-4 робота-манипулятора *Kuka* с 7 степенями свободы каждый.

Ключевые слова: планирование движения, планирование траекторий, многоагентное планирование, манипулятор, конфликтно-ориентированное планирование, поиск на основе конфликтов, поиск по графу, интервальные ограничения, временные ограничения.

1. Введение. В последние десятилетия робототехника стала одной из наиболее динамично развивающихся областей науки и техники. Особое место в этой сфере занимают многозвенные манипуляторы – механические устройства, состоящие из звеньев и шарниров с рабочим органом на последнем из них.

Манипуляторы активно применяются в промышленности, складской автоматизации, сервисной и медицинской робототехнике и др. При этом в последнее время всё чаще возникает необходимость в адаптивном поведении манипуляторов, а не выполнении одной или нескольких заранее построенных программ движений и действий. Также зачастую возникают задачи предполагающие их выполнение не одним, а системой из нескольких манипуляторов. Таким образом, весьма актуальной становится задача разработки эффективных методов

планирования согласованных траекторий для системы из нескольких робототехнических манипуляторов, действующих в общем рабочем пространстве.

Задача адаптивного планирования движений для отдельного манипулятора уже является вычислительно сложной из-за, во-первых, необходимости решения задачи обратной кинематики и частого вызова достаточно ресурсоемкой процедуры проверки на коллизии и, во-вторых, из-за большого числа степеней свободы робототехнической системы. Если же роботов-манипуляторов несколько, то сложность задачи возрастает ещё больше, т.к. для многоагентного планирования характерен экспоненциальный рост пространства комбинаций действий относительного числа агентов.

Одним из способов борьбы с проклятием размерности при многоагентном планировании является конфликтно-ориентированный поиск. При таком типе поиска комбинированного пространства действий не строится, наоборот – планирование осуществляется лишь для отдельных агентов. При обнаружении конфликтов на вовлеченных агентов накладываются пространственно-временные ограничения, определенные специальным образом так, чтобы их соблюдение устраняло обнаруженный конфликт. Затем происходит индивидуальное ерхнеуровневого дерева поиска возникает планирование с учетом наложенных ограничений и процесс повторяется до тех пор, когда не будут устранены все конфликты и не будет получен набор согласованных индивидуальных планов, одновременное выполнение которых происходит без столкновений. Этот набор планов и является решением исходной задачи.

Изначально конфликтно-ориентированное планирование было предложено для мобильных роботов [1], однако в последние несколько лет появились алгоритмы, адаптирующие этот подход к системам много-звенных робототехнических манипуляторов [2]. К сожалению, подобные алгоритмы используют способы задания конфликтов, характеризующиеся привязкой к конкретному моменту времени. Из-за этого эффективность подхода снижается, т.к. устранение конфликта в момент времени t зачастую приводит к возникновению аналогичного конфликта в момент времени $t + 1$. Именно на решение этой проблемы и направлена данная работа.

Для повышения эффективности планирования согласованных траекторий для системы много-звенных манипуляторов в работе предлагается новый алгоритм поиска, GECBS-T, который использует принцип конфликтно-ориентированного планирования, но вводит в рассмотрение интервальные временные ограничения, которые нацелены

на избегание цепочек конфликтов, возникающих при стандартном подходе. Благодаря использованию дополнительной техники фокусировки поиска, предлагаемый алгоритм GECBS-T позволяет гарантировать отыскание решений, стоимость которых не превосходит стоимость оптимальных более, чем на определенную пользователем величину. Таким образом, с одной стороны алгоритм позволяет контролировать качество отыскиваемых траекторий, с другой – превосходит предшествующие алгоритмы по скорости решения задач планирования. Последнее обстоятельство подтверждается проведенными многочисленными экспериментами с использованием робототехнического симулятора *muJoCo* и реальных моделей симуляторов *Kuka* с 7 степенями свободы.

Программный код разработанного метода и системы тестирования открыт и свободно доступен¹.

2. Обзор работ. Задача планирования движений группы роботов-манипуляторов (здесь и далее именуемая как M-RAMP, от. англ. *multi-robot/agent motion planning*) может быть рассмотрена как более сложная версия задачи многоагентного поиска путей (MAPF, *multi-agent pathfinding*) [3, 4]. В классической формулировке этой задачи на плоскости с препятствиями расположены мобильные роботы, способные перемещаться по неза занятой препятствиями территории. У каждого робота есть начальное и конечное положение и нужно построить для каждого робота путь между ними так, чтобы ни в какой момент времени никакие роботы не столкнулись друг с другом или с препятствиями.

Разница задач M-RAMP и MAPF заключается в том, что в первом случае степень свободы одного робота находится в диапазоне от 6 до 8, тогда как у робота на плоскости она равна 2, из-за чего планирование движений для одного робота-манипулятора намного более сложная задача, чем планирование траектории агента на плоскости. Так, если в задаче MAPF количество агентов может достигать нескольких сотен, то в задаче M-RAMP их число не более 20. Таким образом, решения задачи MAPF, основанные на том, что планирование траектории одного агента происходит быстро, нужно адаптировать к задаче M-RAMP.

Для решения задачи MAPF применяются разные методы: метод приоритизированного планирования [5–7] и метод конфликтно-ориентированного планирования [8], которые могут быть перенесены с роботов на плоскости на роботов-манипуляторов. Так как главное отличие задач в том, что в задаче MAPF путь для одного робота строится относительно быстро, а в M-RAMP достаточно медленно, то для переноса метода конфликтно-ориентированного планирования авторы

¹https://github.com/machine-solution/motion_planning_for_manipulators

алгоритмов стараются оптимизировать их за счёт уменьшения количества вызовов планирования траекторий для одного робота, что показывает алгоритм Generalized ECBS [2], являющийся на данный момент лучшим алгоритмом на основе поиска для решения данной задачи. Это основной метод решения данной задачи на основе поиска [9]. Приоритизированное планирование можно перенести и без модификаций [10], тем не менее такой подход также существенно зависит от выбранных приоритетов и на практике является менее эффективным, чем остальные.

Для задачи M-RAMP также есть и специфичный подход на основе семплирования. Такой метод представляет группу манипуляторов как единую систему и применяет к ней алгоритм RRT [11]. Также для этого алгоритма было разработано множество модификаций [12, 13]. Такой алгоритм не так сильно подвержен замедлению, связанному с ростом размерности системы, за счёт чего является довольно быстрым. Такой подход не гарантирует оптимальности или ограниченной субоптимальности решения и даже не гарантирует нахождения решения, если оно существует, что является его недостатком по сравнению с алгоритмами на основе поиска. Также минусом является и то, что такие алгоритмы на основе семплирования [14] выдают непредсказуемые случайные результаты.

Также к задаче планирования манипуляторов применяется подход на основе машинного обучения [15–17]. Эти методы позволяют проще адаптироваться к новым задачам и легко адаптируются к условиям рабочего пространства, но само обучение является трудоёмким процессом, который требует большое количество времени и данных. Также такие алгоритмы довольно тяжело интерпретировать и они не могут давать каких-либо гарантий относительно решения.

Таким образом, одним из самых перспективных направлений изучения являются методы планирования на основе поиска, в частности конфликтно-ориентированном поиске, так как этот алгоритм имеет теоретические гарантии, выдаёт предсказуемый результат, а лучший алгоритм из этого класса Generalized ECBS показывает сравнимые показатели скорости работы с алгоритмами на основе семплирования, поэтому данный алгоритм является базовым для этой работы.

3. Постановка задачи. Дано рабочее пространство, разбитое на свободное пространство и препятствия $W = W_{free} \cup W_{obs}$, $W \subset \mathbb{R}^3$. Рассмотрим n многосвязных манипуляторов $R_1, \dots, R_n$. Конфигурацией манипулятора называется набор углов поворота k сочленений манипулятора $q = (\alpha_1, \dots, \alpha_k)$, а $B_i(q)$ обозначает множество точек рабочего пространства, которое занимает i -й манипулятор в

конфигурации q . Конфигурации каждого манипулятора являются подмножеством $\mathbb{R}^k$ и образуют конфигурационные пространства $C^i = C_{free}^i \cup C_{obs}^i$, где C_{free}^i – это множество конфигураций q , таких, что $B_i(q) \cap W_{obs} = \emptyset$, а C_{obs}^i наоборот. Так, на рисунке 1 слева изображено рабочее пространство (W-space) и двухзвенный манипулятор, а справа пространство конфигураций (C-space) этого манипулятора. Агенты R_i и R_j считаются находящимися в конфликте в конфигурациях q_i и q_j , если выполняется условие $B_i(q_i) \cap B_j(q_j) \neq \emptyset$.

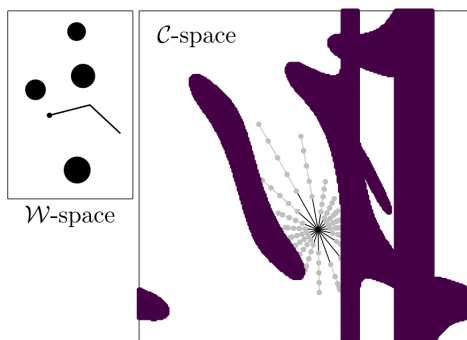


Рис. 1. Пример рабочего и конфигурационного пространств (изображение из [13])

Даны начальные конфигурации $q_{start}^i \in C_{free}^i$ и целевые конфигурации $q_{goal}^i \in C_{free}^i$. Также дано число $w > 1$, отвечающее за ограниченную субоптимальность. Решением задачи M-RAMP в дискретном времени является набор путей для отдельных агентов $\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$. Каждый путь $\pi_i = \{q_i^0, q_i^1, \dots, q_i^T\}$ удовлетворяет следующим условиям: $q_i^0 = q_i^{start}$, $q_i^T = q_i^{goal}$, все конфигурации $q_i^t \in C_{free}^i$, из конфигурации q_i^t в конфигурацию q_i^{t+1} можно добраться за одно действие, которое является поворотом в одном сочленении на угол минимального поворота δ , и между любыми парами агентов R_i , R_j нет конфликтов на всех временных шагах при следовании путям π_i и π_j . Также часто требуется не только найти допустимые пути для всех агентов, но и минимизировать стоимость этих путей. Обозначим за временную стоимость одного пути π_i такое наименьшее целое число $cost(\pi_i)$, что $\forall t \geq cost(\pi_i) : q_i^t = q_i^{goal}$, что соответствует времени, за которое конкретный агент добирается до целевой конфигурации и больше не уходит из неё. Тогда стоимостью решения задачи M-RAMP является следующее значение:

$$cost(\Pi) = \sum_{i=1}^n cost(\pi_i).$$

Набор путей считается корректным решением, если $cost(\Pi) \leq w \cdot cost(\Pi^*)$, где Π^* – это набор путей оптимальной стоимости. Такое решение называется ограниченно субоптимальным.

4. Метод. Предлагаемый в работе алгоритм реализует принципы конфликтно-ориентированного планирования, адаптирует эти принципы к рассматриваемой постановке и отличается от аналогов способом определения ограничений. Рассмотрим эти компоненты алгоритма более подробно далее.

4.1. Конфликтно-ориентированное планирование.

Конфликтно-ориентированное планирование было предложено в работе [18], где описан один из базовых алгоритмов реализующих этот принцип планирования – CBS (от англ. Conflict-Based Search). Основная идея конфликтно-ориентированного планирования заключается в двухуровневой структуре поиска. На верхнем уровне строится дерево разрешения конфликтов или же дерево ограничений (Constraint Tree), где каждая вершина содержит набор ограничений на движения агентов, а также индивидуальные пути для каждого агента, построенные с учётом этих ограничений. На нижнем уровне для каждого агента осуществляется поиск индивидуального пути с помощью стандартных алгоритмов (например, A^* или $weighted A^*$ [19]), учитывая наложенные ограничения.

Алгоритм CBS работает следующим образом:

- На начальном этапе для каждого агента строится оптимальный путь без учёта других агентов.
- Если обнаруживается конфликт (например, два агента занимают одну и ту же вершину в одно и то же время), то создаются две новые вершины дерева СТ, в каждой из которых одному из агентов добавляется ограничение, запрещающее конфликтное действие.
- Для каждой новой вершины СТ пересчитывается путь только для того агента, для которого добавлено новое ограничение.
- Процесс повторяется, пока не будет найдено решение, в котором отсутствуют конфликты между агентами.

На рисунке 2 представлен пример части верхнеуровневого дерева поиска. Пример взят для планирования путей для роботов на плоскости, так как он нагляднее и проще для понимания. На рисунке показано как конфликт в корневой вершине приводит к появлению двух новых вершин с дополнительными ограничениями.

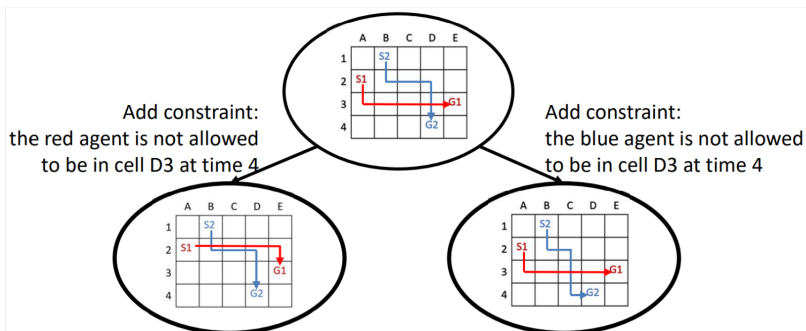


Рис. 2. Верхнеуровневое дерево поиска алгоритма CBS. В корневой вершине произошёл конфликт, из-за чего были созданы две новые вершины. В левой вершине был перепланирован красный путь с учётом ограничения, а в правой синий. Изображение взято из презентации "Model AI Assignment: Introduction to Multi-Agent Path Finding Wolfgang Hoenig, Jiaoyang Li, Sven Koenig

Данный подход позволяет эффективно разрешать конфликты, минимизируя количество перепланирований путей и обеспечивая оптимальность решения по суммарной длине путей агентов.

Алгоритм CBS является оптимальным и его эффективность опирается на предположение о том, что функция FindPath находит путь для одного робота быстро, что было верным для роботов, передвигающихся по плоскости, но не соответствует действительности для роботов-манипуляторов из-за большой размерности пространства конфигураций C^i .

По этой причине для роботов-манипуляторов был придуман алгоритм ECBS (Enhanced CBS) [20], являющийся ограниченно-субоптимальным, но позволяющим значительно быстрее находить решение. Модификация заключается в том, что помимо верхнеуровневой очереди OPEN алгоритм получает ещё и очередь FOCAL:

$$\text{FOCAL} := \{N \in \text{OPEN} \mid N.\text{cost} \leq w \cdot \min_{N' \in \text{OPEN}} \text{lb}(N')\},$$

где $\text{lb}(N) := \sum_{i=1}^n \text{lb}(\pi_i)$, и $\text{lb}(\pi_i)$ является нижней оценкой стоимости пути $N.\Pi[i]$. ECBS на каждой итерации главного цикла выбирает вершину N из FOCAL с наименьшим количеством конфликтов в ней:

$$N = \arg \min_{N' \in \text{FOCAL}} |N' \cdot \Omega|,$$

где $N \cdot \Omega$ – это множество конфликтов, обнаруженных среди набора путей N .П. Такой подход позволяет находить решения с общей стоимостью, не превышающей $w \cdot C^*$, где C^* – оптимальная суммарная стоимость. Как правило, такие решения находятся значительно быстрее, чем при использовании алгоритма CBS.

4.2. Конфликтно-ориентированное планирование для манипуляторов. Для обеспечения полноты в CBS и его вариантах крайне важно, чтобы среди используемых ограничений были полные ограничения

Определение 1 (Полные ограничения). Ограничение является полным, если при использовании для разрешения конфликта между агентами R_i и R_j , путём наложения ограничений c_i и c_j соответственно, не существует двух путей без конфликтов для R_i и R_j , где R_i нарушает c_i , а R_j нарушает c_j .

Использование только лишь полных ограничений зачастую является неэффективным, так как такие ограничения слишком слабо ограничивают движения агентов. Поэтому для планирования движений роботов-манипуляторов в работе [2] были предложены более рестриктивные, но не полные, ограничения, повышающие эффективность алгоритма конфликтно-ориентированного планирования на практике. Ниже описаны все известные на данный момент ограничения с указанием их свойств.

Ограничение на вершины (Vertex Constraints). Ограничения на вершины запрещают агенту R_i занимать определённую конфигурацию q_i^ω в момент времени t . Несложно убедиться, что это ограничение является полным, а вот следующие ограничения уже не будут являться полными

Ограничение на избегание (Avoidance Constraints). Ограничения на избегание запрещают агенту R_i сталкиваться с объёмом, занимаемым другим агентом $R_j(q_j^\omega)$ в момент времени t .

Ограничение на приоритеты (Priority Constraints). Ограничения на приоритеты запрещают агенту R_i сталкиваться с агентом R_j , пока тот движется по своему текущему пути π_j . Это ограничение используется в алгоритмах Prioritized Planning (PP) и PBS, однако оно также не является полным.

Сферическое ограничение (Sphere Constraints). Сферические ограничения запрещают агенту R_i сталкиваться с сферическим препятствием $S^2(p, r)$, имеющим радиус r и центр в точке столкновения

p , в момент времени t . При $r > 0$ это ограничение является неполным, однако при уменьшении радиуса до нуля оно становится полным.

На рисунке 3 изображения разные ограничения, которые могут быть наложены на агентов на основе одного конфликта. Преимуществом неполных ограничений является то, что они отсекают более обширную часть пространства поиска, что приводит к значительному ускорению, но при этом решение может не быть найдено из-за отсечения. Обозначим за k количество ограничений. Ради сохранения полноты алгоритм Generalized ECBS на верхнем уровне планирования рассматривает варианты с накладыванием всех k ограничений, а не только вершинного, в результате чего у каждой вершины верхнеуровневого дерева поиска возникает $2k$ потомков, а не 2, как это было в ECBS.

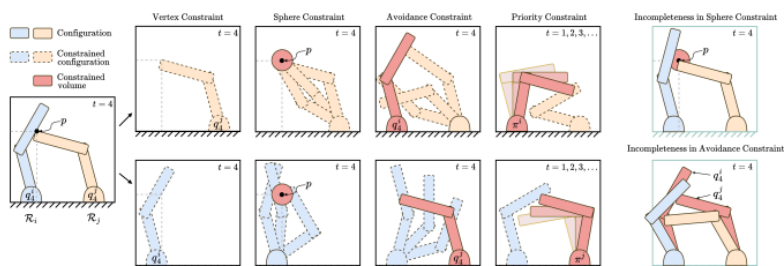


Рис. 3. Ограничения, введенные авторами алгоритма Generalized ECBS (изображение из [2])

4.3. Алгоритм GECBS-T. Идея ограничений алгоритма ECBS заключается в отсекании пространства поиска в вершинах, образованных неполными ограничениями. Но большинство из этих ограничений устанавливается на определённый момент времени, за который манипулятор может успеть выполнить только одно элементарное действие. Ограничения на избегание, приоритет и сферическое ограничение расширяют вершинное ограничение в пространстве конфигураций, то есть запрещают в конкретный момент времени находиться не в одной конфигурации, а в некотором подмножестве близких конфигураций. На практике же требуется и обратное, запрещение находиться в близкой конфигурации в течение некоторого продолжительного времени, так как для того, чтобы развести маршруты двух манипуляторов в пространстве может потребоваться большое количество шагов.

Второе соображение заключается в том, что ограничение, накладываемое на один момент времени, зачастую приводит к столкновению в следующий момент времени, при этом алгоритму

приходится накладывать ещё одно ограничение и перезапустить алгоритм планирования в вершине-потомке, что является дорогостоящей операцией. Так, например, происходит на рисунке 4 слева, при наложении ограничения на конфликтную конфигурацию в конкретный момент времени манипулятор оказывается в той же конфигурации в следующий момент времени, что также часто приводит к конфликту в реальных заданиях. В работе предлагается подход, который уменьшит количество запусков низкоуровневого планирования за счёт постановки ограничения сразу на продолжительное время.

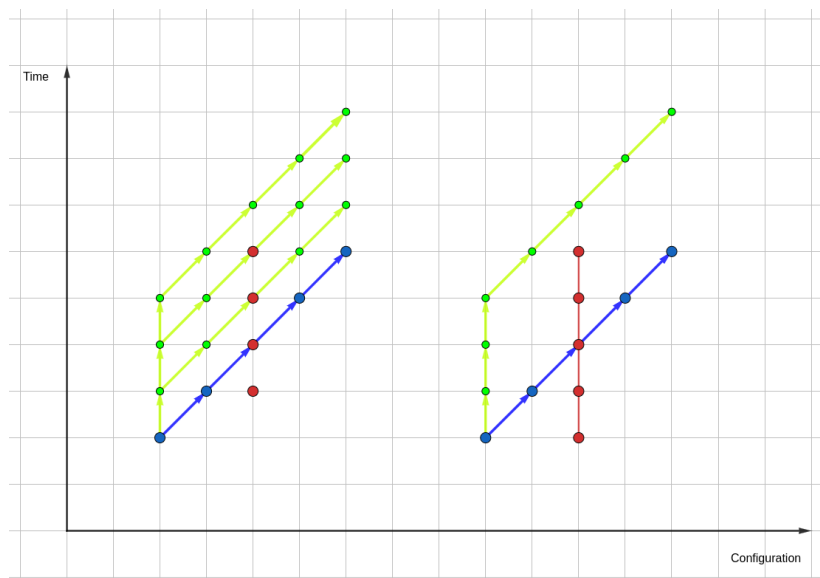


Рис. 4. По вертикали обозначена координата времени, по горизонтали обозначены координаты пространства конфигураций (одним измерением для упрощения). Синим обозначен путь манипулятора в пространстве-времени, красным обозначены конфликтные конфигурации. Синий путь – первый среди найденных планировщиков, в котором есть конфликт. Зеленым выделены пути, найденные планировщиком после синего с учётом конфликтов (в порядке наложения). Слева ограничения на конфликтные конфигурации были наложены по-отдельности, справа все конфигурации покрыты одним ограничением

Разработанный в данной работе алгоритм GECBS-T использует FOCAL очередь для ускорения поиска, а также неполные ограничения. Также алгоритм принимает на вход параметр T – длительность промежутка ограничений в тактах времени и вместо установки ограничения в момент

времени t устанавливать его с времени $t - T$ до $t + T$ включительно. Так, например, на рисунке 4 справа за счёт увеличенного по времени ограничения экономятся промежуточные вершины верхнеуровневого дерева CBS, что приводит к меньшему количеству низкоуровневых запусков планировщика. Такая формулировка ограничения не замедляет низкоуровневого планировщика и, даже наоборот, немного ускоряет за счёт того, что ограничение хранится в сжатом виде, а не как T экземпляров. Псевдокод данного алгоритма представлен ниже.

Алгоритм 1. GECBS-T

Require: agents, start, goal, T

```

1: root.constraints  $\leftarrow \emptyset$ 
2: for all agent  $i$  do
3:   root.path[ $i$ ]  $\leftarrow$  FindPath( $i$ , start[ $i$ ], goal[ $i$ ], root.constraints)
4: end for
5: root.cost  $\leftarrow$  sum of costs of all root.path[ $i$ ]
6: root.conflicts  $\leftarrow$  number of conflicts for all root.path[ $i$ ]
7: OPEN  $\leftarrow$  priority queue ordered by (cost)
8: FOCAL  $\leftarrow$  subset of OPEN with cost  $\leq w \cdot \min \text{cost}(\text{OPEN})$  ordered by
   (conflicts.size())
9: Insert root into OPEN and FOCAL
10: while FOCAL is not empty do
11:   node  $\leftarrow$  FOCAL.pop() ▷ Node with minimum conflicts
12:   if node.paths have no conflicts then
13:     return node.paths
14:   end if
15:   conflict  $\leftarrow$  earliest conflict in node.conflicts
16:   for all constraint type  $c \in$  all constraint types do
17:     for all agent  $i$  involved in  $c$  do
18:       child  $\leftarrow$  copy of node
19:       append new Constraint( $c, T, i, \text{conflict}$ ) to child.constraints
20:       child.path[ $i$ ]  $\leftarrow$  FindPath( $i, \text{start}[i], \text{goal}[i], \text{child.constraints}$ )
21:       if child.path[ $i$ ] exists then
22:         child.cost  $\leftarrow$  sum of costs of all child.path[ $j$ ]
23:         child.h  $\leftarrow$  heuristic estimate for child
24:         Insert child into OPEN
25:       end if
26:     end for
27:   end for
28:   Update FOCAL with all nodes in OPEN where cost  $\leq w \cdot \min(\text{OPEN})$ 
29: end while
30: return failure

```

В данном псевдокоде функция `FindPath` осуществляет поиск индивидуального пути для агента с учётом текущих ограничений (например, с помощью алгоритма A^*). А функция `Constraint` создаёт ограничение для агента, исходя из конфликта, параметра T и типа ограничения. Процесс продолжается до тех пор, пока не будет найдено решение без конфликтов или не будет исчерпан список возможных вариантов.

Приведем теперь утверждение, характеризующее основные теоретические гарантии, предоставляемые предлагаемым алгоритмом.

Утверждение. Алгоритм GECBS- T является полным, то есть если решение задачи планирования существует, то оно будет найдено алгоритмом. Более того, решения, возвращаемые GECBS- T являются ограниченно субоптимальными, то есть для любой заданной наперёд константы w (являющейся параметром алгоритма) верно, что стоимость найденного решения $cost$ не превосходит $w \cdot cost^*$, где $cost^*$ – стоимость оптимального решения.

Доказательство. Во-первых покажем, что в верхнеуровневом дереве ECBS всегда будет нераскрытая вершина, ограничения которой соответствуют одному из оптимальных решений, если решение задачи существует. Докажем это утверждение по индукции. Изначально это верно, так как в корневой вершине нет ограничений, а так как решение существует, то существует и оптимальное. При раскрытии каждой вершины либо раскрывается вершина, ограничения которой удовлетворяют оптимальному решению, либо нет. Если ограничения вершины противоречат оптимальному решению, то так как в дереве есть хотя бы одна вершина без противоречий ограничений оптимальному решению – эти вершины разные и искомая вершина остаётся нераскрытой после данного шага. Если же вершина, ограничения которой удовлетворяют оптимальному решению, раскрывается, то за счёт наличия полного ограничения у неё есть два потомка, объединение решений которых содержит все решения без конкретного разрешённого конфликта по свойству полных ограничений, а значит одна из этих вершин не запрещает своими ограничениями оптимальное решение. Перейдём к доказательству ограниченной субоптимальности алгоритма с константой w . Так как в очереди OPEN всегда есть вершина, ограничения которой не противоречат оптимальному решению со стоимостью $cost^*$, то верно неравенство $\min_{N' \in OPEN} lb(N') \leq cost^*$, так как $lb(N) \leq cost^*$, где N – та самая вершина. Это означает, что в FOCAL стоимость всех вершин будет не больше чем $w \cdot cost^*$ по определению очереди FOCAL, следовательно каждая раскрытая вершина будет иметь решение, не

превосходящее $w \cdot cost^*$, значит и итоговое решение обладает этим же свойством.

5. Экспериментальное исследование. Предлагаемый метод был программно реализован на языке C++² и использован для проведения экспериментов в робототехническом симуляторе mujoco от Google Deep Mind. Все эксперименты проводились на компьютере с процессором Intel Xeon Gold 6338 с двумя ядрами и 16 ГБ оперативной памяти.

5.1. Подготовка тестовых заданий. Для проведения экспериментов использовалась программная модель современного робота-манипулятора iiwa-14 от kuka (рисунок 5), доступная в симуляторе mujoco.

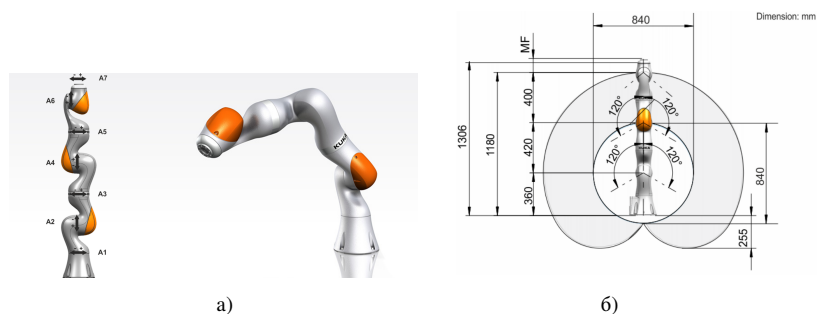


Рис. 5. Робот iiwa-14 от kuka с указанием размеров

Так как датасетов с заданиями для этой задачи очень мало и они являются специфичными для разных роботов и симуляторов, задания в этой работе были сгенерированы самостоятельно. Для подбора заданий было решено генерировать случайным образом начальные и конечные конфигурации для каждого манипулятора, при этом основания манипуляторов расположены в вершинах квадрата со стороной 1 метр.

При случайной генерации для решения многих заданий достаточно простого независимого планирования всех манипуляторов, так как при этом у найденных путей не возникало конфликтов. Такие задания также не попали в датасет, так как эти задания решаются каждым алгоритмом одинаково быстро, а значит они заведомо не позволяют сравнить два алгоритма. Также в результате случайной генерации заданий они получались достаточно простыми и почти независимыми, поэтому был найден способ генерировать в среднем более сложные задания для манипуляторов. Если рабочий орган каждого манипулятора находится

²https://github.com/machine-solution/motion_planning_for_manipulators

внутри куба $1 \times 1 \times 1$ метр, нижние вершины которого совпадают с креплениями четырёх манипуляторов, то независимо построенные траектории роботов-манипуляторов существенно мешают друг другу, что приводит к более сложным заданиям. Итоговый датасет собирался только из тех заданий, которые удовлетворяют данному условию.

Алгоритм GECBS-T сравнивался с алгоритмом Generalized ECBS для разных значений параметра T , из следующего набора: 5, 25, 50, 100, 250, 350, 425, 500 и 750.

Было подготовлено 5 сцен с 2, 3 и 4 манипуляторами, в трёх из которых нет препятствий, а в двух есть препятствие в центре сцены (рисунок 6).

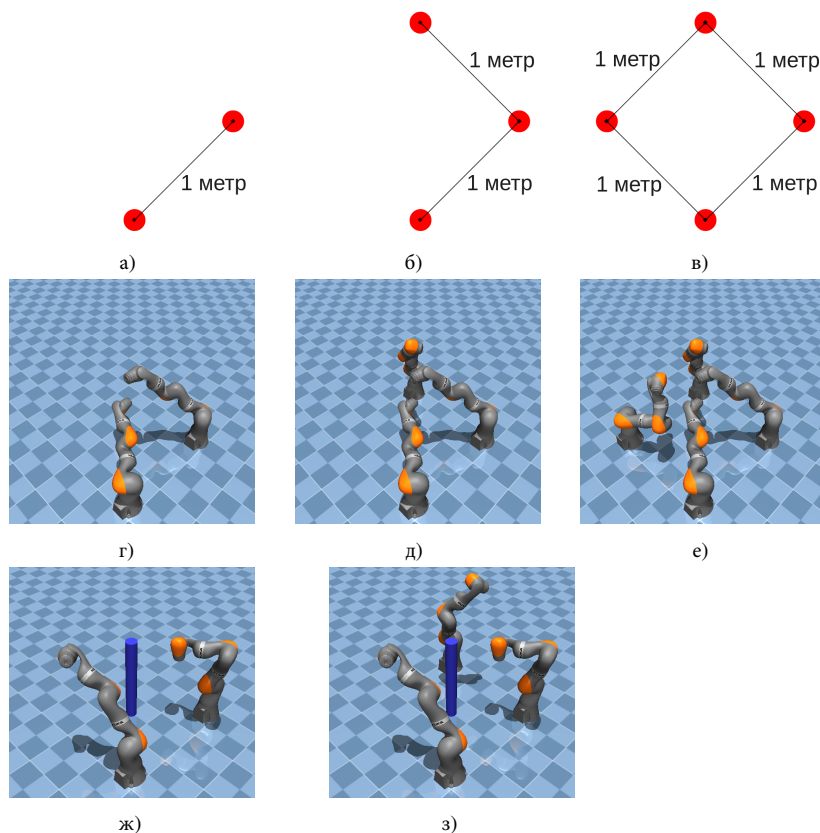


Рис. 6. Пример задания для: г, ж) двух, д, з) трёх и е) четырёх манипуляторов и а, б, в) расположение их оснований

Для каждой сцены генерировалось 100 случайных заданий как описано в предыдущей главе. Для каждого задания отводился бюджет времени 60 секунд. Если алгоритм не успевал найти ответ за это время, его выполнение останавливалось.

Для экспериментов был выбран минимальный угол поворота $\delta = \frac{\pi}{128}$ и любое элементарное действие является поворотом одного из сочленений на угол δ .

5.2. Результаты экспериментов. Алгоритмы GECBS-T и Generalized ECBS являются полными, то есть находят решение задания, если оно есть, но только в предположении бесконечного времени и вычислительных ресурсов. Если имеется ограничение по времени решения, как в проводимых экспериментах, то не для каждого алгоритм успевает найти ответ. Таким образом, возникает такой показатель вычислительной эффективности алгоритма как **доля решённых заданий** за отведенный временной лимит (60 секунд в рассматриваемом случае). Значение этого показателя приведено в таблице 1.

Таблица 1. Доля решённых заданий в зависимости от количества манипуляторов и значения параметра T

	2 ман-ра	3 ман-ра	4 ман-ра	2 ман-ра, преп.	3 ман-ра, преп.
Generalized ECBS	50%	42%	26%	45%	29%
GECBS-T, $T = 5$	55%	42%	28%	47%	30%
GECBS-T, $T = 25$	58%	44%	36%	47%	30%
GECBS-T, $T = 50$	57%	42%	31%	46%	33%
GECBS-T, $T = 100$	59%	49%	33%	49%	32%
GECBS-T, $T = 250$	69%	53%	41%	54%	33%
GECBS-T, $T = 350$	67%	57%	43%	54%	39%
GECBS-T, $T = 425$	66%	56%	31%	54%	39%
GECBS-T, $T = 500$	61%	50%	34%	54%	38%
GECBS-T, $T = 750$	55%	52%	29%	54%	35%

Первая строка в таблице 1 – это доля решенных заданий для алгоритма GECBS-T, остальные строки показывают долю решенных заданий для предлагаемого в работе алгоритма GECBS-T, для различных значений параметра T . Каждый столбец соответствует отдельной сцене с различным числом манипуляторов. Как видно из результатов, предлагаемый в работе алгоритм заметно превосходит предшественника по способности отыскивать решения за ограниченное время. Доля решённых заданий у алгоритма GECBS-T выше на 15% – 65% по сравнению с Generalized ECBS, в зависимости от количества манипуляторов, типа сцены (с препятствием или без) и значения параметра T . Для алгоритма GECBS-T наибольшее доля решенных заданий достигается для значений параметра $T = 250$, и $T = 350$. При

этом до некоторого значения параметра T эта доля растёт, после чего уменьшается за счёт того, что такая длина интервала уже является избыточной.

Второй метрикой для сравнения алгоритмов является скорость работы. На рисунках 7, 8 представлены графики зависимости количества решённых заданий (ось Y) от бюджета времени (ось X) для сцен с тремя манипуляторами.

Количество решённых заданий в зависимости от бюджета времени для 3 манипуляторов

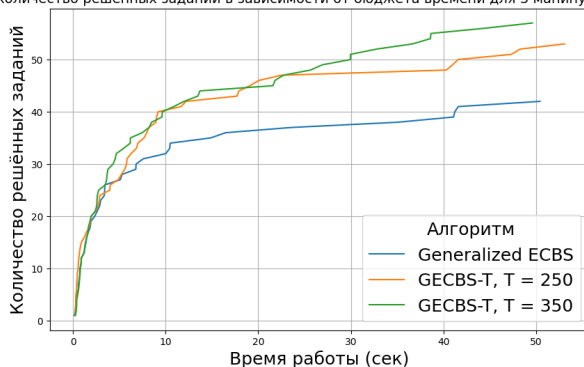


Рис. 7. Количество решённых заданий в зависимости от бюджета времени для трёх манипуляторов для сцены без препятствий

Решённые задания в зависимости от бюджета времени, 3 манипулятора

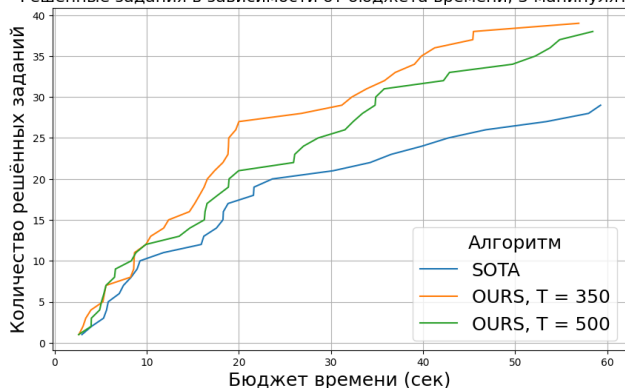


Рис. 8. Количество решённых заданий в зависимости от бюджета времени для трёх манипуляторов для сцены с препятствиями

На графиках отображены алгоритмы Generalized ECBS и GECBS-T с параметрами $T = 250, 350$. По этим результатам видно, что для каждой сцены есть какое-то количество заданий, которые решаются одинаково быстро всеми версиями алгоритма, так как они очень простые, зато для более сложных заданий, которые всё ещё решаются алгоритмами за минуту, которые составляют долю 20% – 35%, представленный в работе алгоритм решает в 2-5 раз быстрее. Именно за счёт большей скорости решения сложных заданий алгоритм GECBS-T показывает большую долю решённых заданий.

Перейдем теперь к сравнению качества решений, полученных алгоритмом Generalized ECBS и GECBS-T (для параметров $T = 250, T = 350$ и $T = 500$). Для каждого задания, которое решено алгоритмами, было рассчитано значение $\frac{cost(\Pi)}{cost(\Pi^*)} - 1$, где Π – решение, выданное алгоритмом, а Π^* – решение, полученное из Generalized ECBS. Посчитанное значение показывает превышение стоимости найденного пути по сравнению с оригинальным алгоритмом. Все эти значения, усреднённые по всем заданиям, показаны в таблице 2.

Таблица 2. Среднее превышение стоимости пути по сравнению с алгоритмом Generalized ECBS в зависимости от параметра T

	2 ман-ра	3 ман-ра	4 ман-ра	2 ман-ра, преп.	3 ман-ра, преп.
Generalized ECBS	0%	0%	0%	0%	0%
GECBS-T, $T = 250$	4.65%	0.63%	0.72%	0.04%	0.04%
GECBS-T, $T = 350$	3.30%	0.82%	0.76%	0.03%	0.01%
GECBS-T, $T = 350$	3.01%	0.71%	0.69%	0.03%	0.02%

Видно, что GECBS-T получает в среднем решения с большей стоимостью, но незначительно. Это показывает, что он практически ни в чём не уступает оригинальному алгоритму, но решает задания гораздо быстрее.

5.3. Выводы. Экспериментальные исследования показывают, что разработанный алгоритм GECBS-T превосходит Generalized ECBS по нескольким ключевым метрикам. Проведённые эксперименты выявили следующие преимущества.

1. Скорость решения заданий. GECBS-T на сложных заданиях показывает увеличение скорости в 2-5 раз по сравнению с Generalized ECBS без потери в скорости на простых заданиях, что делает его значительно более быстрым.

2. Выше доля решённых заданий. По сравнению с Generalized ECBS, GECBS-T решает больше задач в отведённый временной бюджет. Доля решённых заданий зависит от параметра алгоритма T , для которого

эксперименты показывают, что оптимальное значение находится между 250 и 350 для $\delta = \frac{\pi}{128}$, где δ – это минимальный угол поворота. Такой эффект достигается за большей скорости работы на сложных заданиях

3. Сравнимое качество решения. Алгоритм GECBS-T выдаёт решения с превышением стоимости менее чем на 5% относительно решений Generalized ECBS в худшей сцене и с превышением стоимости менее чем на 1% в остальных сценах, что означает практически такое же фактическое качество решения. При этом он сохраняет все теоретические гарантии на стоимость пути.

Исходя из этих данных можно сделать вывод, что GECBS-T из данной работы показывает себя намного быстрее и эффективнее, чем базовый алгоритм Generalized ECBS. Эти результаты достигаются благодаря меньшему количеству обращений к низкоуровневому планировщику за счёт более грубых ограничений. По этой причине GECBS-T незначительно уступает Generalized ECBS по качеству решений, но при этом сохраняет теоретические гарантии, оставаясь полным и ограниченно субоптимальным. Таким образом, предложенный алгоритм получает значительно более эффективны решения задач M-RAMP с гарантией ограниченной субоптимальности.

6. Заключение. В данной работе предложен алгоритм планирования согласованных траекторий системы из нескольких многозвенных манипуляторов GECBS-T, основанный на методе конфликтно-ориентированного планирования и доказана его корректность. Алгоритм был реализован и проведены экспериментальные исследования, в которых алгоритм GECBS-T сравнивался с алгоритмом Generalized ECBS. Эксперименты показали значительно большую эффективность представленного в статье алгоритма. В дальнейшем планируется исследование динамического определения длительности интервала ограничений в алгоритме в зависимости от уже построенных путей, улучшение системы приоритетов для разных ограничений в зависимости от показанной ими эффективности в прошлом и проведение экспериментов на реальных роботах.

Литература

1. Sharon G., Stern R., Felner A., Sturtevant N.R. Conflict-based search for optimal multi-agent pathfinding. Artificial intelligence. 2015. vol. 219. pp. 40–66.
2. Shaoul Y., Veerapaneni R., Likhachev M., Li J. Unconstraining multi-robot manipulation: Enabling arbitrary constraints in ecbs with bounded sub-optimality. Proceedings of the International Symposium on Combinatorial Search. 2024. vol. 17. pp. 109–117.
3. Ma H., Koenig S. Ai buzzwords explained: multi-agent path finding (mapf). AI Matters. 2017. vol. 3(3). pp. 15–19.

4. Stern R., Sturtevant N., Felner A., Koenig S., Ma H., Walker T., Li J., Atzmon D., Cohen L., Kumar T.K., et al. Multi-agent pathfinding: Definitions, variants, and benchmarks. *Proceedings of the International Symposium on Combinatorial Search*. 2019. vol. 10. pp. 151–158.
5. Ma H., Harabor D., Stuckey P.J., Li J., Koenig S. Searching with consistent prioritization for multi-agent path finding. In *Proceedings of the AAAI conference on artificial intelligence*. 2019. vol. 33. pp. 7643–7650.
6. Chan S.-H., Stern R., Felner A., Koenig S. Greedy priority-based search for suboptimal multi-agent path finding. *Proceedings of the International Symposium on Combinatorial Search*. 2023. vol. 16. pp. 11–19.
7. Zhang S., Li J., Huang T., Koenig S., Dilkina B. Learning a priority ordering for prioritized planning in multi-agent path finding. *Proceedings of the International Symposium on Combinatorial Search*. 2022. vol. 15. pp. 208–216.
8. Sharon G., Stern R., Felner A., Sturtevant N.R. Conflict-based search for optimal multi-agent pathfinding. *Artificial intelligence*. 2015. vol. 219. pp. 40–66.
9. Shaoul Y., Mishani I., Likhachev M., Li J. Accelerating search-based planning for multi-robot manipulation by leveraging online-generated experiences. *Proceedings of the International Conference on Automated Planning and Scheduling*. 2024. vol. 34. pp. 523–531.
10. Chen J., Li J., Huang Y., Garrett C., Sun D., Fan C., Hofmann A., Mueller C., Koenig S., Williams B.C. Cooperative task and motion planning for multi-arm assembly systems. *arXiv preprint arXiv:2203.02475*. 2022.
11. LaValle S. Rapidly-exploring random trees: A new tool for path planning. *Research Report 9811*. 1998.
12. Kuffner J.J., LaValle S.M. Rrt-connect: An efficient approach to single-query path planning. *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065). IEEE*, 2000. vol. 2. pp. 995–1001.
13. Lacevic B., Osmankovic D.. Improved c-space exploration and path planning for robotic manipulators using distance information. *2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE*, 2020. pp. 1176–1182.
14. Karaman S., Frazzoli E. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*. 2011. vol. 30(7). pp. 846–894.
15. Zhang D., Zhu G., Zhang Q. Multi-robot motion planning: A learning-based artificial potential field solution. *2nd Conference on Fully Actuated System Theory and Applications (CFASTA). IEEE*, 2023. pp. 233–238.
16. Shaoul Y., Mishani I., Vats S., Li J., Likhachev M. Multi-robot motion planning with diffusion models. *arXiv preprint arXiv:2410.03072*. 2024.
17. Aljalbout E., Karl M., van der Smagt P. Clas: Coordinating multi-robot manipulation with central latent action spaces. *Learning for Dynamics and Control Conference. PMLR*, 2023. pp. 1152–1166.
18. Sharon G., Stern R., Felner A., Sturtevant N.R. Conflict-based search for optimal multi-agent pathfinding. *Artificial intelligence*. 2015. vol. 219. pp. 40–66.
19. Ebendt R., Drechsler R. Weighted a search-unifying view and application. *Artificial Intelligence*. 2009. vol. 173(14). pp. 1310–1342.
20. Li J., Ruml W., Koenig S. Eecbs: A bounded-suboptimal search for multi-agent path finding. In *Proceedings of the AAAI conference on artificial intelligence*. 2021. vol. 35(14). pp. 12353–12362.

Зайцев Андрей Сергеевич — студент, факультет математики и компьютерных наук, Санкт-Петербургский государственный университет. Область научных интересов: компьютерные науки, наука о данных. Число научных публикаций — 3. Dusha.Zaitsev@yandex.ru; Университетская набережная, 7–9, 199034, Санкт-Петербург, Россия; р.т.: +7(812)328–2000.

Яковлев Константин Сергеевич — канд. физ.-мат. наук, доцент, ведущий научный сотрудник, факультет математики и компьютерных наук, Федеральное государственное учреждение «Федеральный исследовательский центр «Информатика и управление» Российской академии наук» (ФИЦ ИУ РАН); заведующий лабораторией, лаборатория искусственного интеллекта, математики и компьютерных наук им А.А. Маркова, Санкт-Петербургский государственный университет (СПбГУ). Область научных интересов: искусственный интеллект, интеллектуальные динамические системы, интеллектуальные системы управления, интеллектуальная робототехника, интеллектуальное планирование, планирование траектории, эвристический поиск, многоагентные системы, когнитивные агенты. Число научных публикаций — 75. yakovlev@isa.ru; улица Вавилова, 44/2, 119333, Москва, Россия; р.т.: +7(926)270-8979.

A. ZAITSEV, K. YAKOVLEV
**TIME RANGE CONSTRAINTS FOR MOTION PLANNING FOR
MANIPULATORS**

Zaitsev A., Yakovlev K. **Time Range Constraints for Motion Planning for Manipulators.**

Abstract. This paper addresses the problem of coordinated motion planning for multi-link robotic manipulator systems. One of the promising modern approaches to solving this problem is conflict-based planning, which avoids constructing a high-dimensional joint search space by sequentially solving a series of lower-dimensional problems. This is achieved by introducing spatio-temporal constraints whenever conflicts arise in individual manipulator plans, followed by replanning with these constraints in place. Unfortunately, existing methods that use constraints operate with individual time points, which reduces their practical efficiency. In this work, we present a novel conflict-based planning algorithm that utilizes interval-based temporal constraints rather than point-based ones – GECBS-T. Theoretically, the proposed algorithm guarantees bounded sub-optimality of the generated solutions; that is, for any user-defined bound $w > 1$, the cost of the GECBS-T solution will not exceed w times the cost of the optimal solution. In practice, the proposed algorithm significantly outperforms analogous algorithms in terms of planning speed, as confirmed by experiments conducted in the Mujoco robotics simulator involving 2–4 KUKA robotic manipulators, each with 7 degrees of freedom.

Keywords: motion planning, manipulators, M-RAMP, CBS, search-based, conflict-based, graph search, constraints, multi-agent, time range.

References

1. Sharon G., Stern R., Felner A., Sturtevant N.R. Conflict-based search for optimal multi-agent pathfinding. *Artificial intelligence*. 2015. vol. 219. pp. 40–66.
2. Shaoul Y., Veerapaneni R., Likhachev M., Li J. Unconstraining multi-robot manipulation: Enabling arbitrary constraints in ecbs with bounded sub-optimality. *Proceedings of the International Symposium on Combinatorial Search*. 2024. vol. 17. pp. 109–117.
3. Ma H., Koenig S. Ai buzzwords explained: multi-agent path finding (mapf). *AI Matters*. 2017. vol. 3(3). pp. 15–19.
4. Stern R., Sturtevant N., Felner A., Koenig S., Ma H., Walker T., Li J., Atzmon D., Cohen L., Kumar T.K., et al. Multi-agent pathfinding: Definitions, variants, and benchmarks. *Proceedings of the International Symposium on Combinatorial Search*. 2019. vol. 10. pp. 151–158.
5. Ma H., Harabor D., Stuckey P.J., Li J., Koenig S. Searching with consistent prioritization for multi-agent path finding. In *Proceedings of the AAAI conference on artificial intelligence*. 2019. vol. 33. pp. 7643–7650.
6. Chan S.-H., Stern R., Felner A., Koenig S. Greedy priority-based search for suboptimal multi-agent path finding. *Proceedings of the International Symposium on Combinatorial Search*. 2023. vol. 16. pp. 11–19.
7. Zhang S., Li J., Huang T., Koenig S., Dilkina B. Learning a priority ordering for prioritized planning in multi-agent path finding. *Proceedings of the International Symposium on Combinatorial Search*. 2022. vol. 15. pp. 208–216.
8. Sharon G., Stern R., Felner A., Sturtevant N.R. Conflict-based search for optimal multi-agent pathfinding. *Artificial intelligence*. 2015. vol. 219. pp. 40–66.

9. Shaoul Y., Mishani I., Likhachev M., Li J. Accelerating search-based planning for multi-robot manipulation by leveraging online-generated experiences. *Proceedings of the International Conference on Automated Planning and Scheduling*. 2024. vol. 34. pp. 523–531.
10. Chen J., Li J., Huang Y., Garrett C., Sun D., Fan C., Hofmann A., Mueller C., Koenig S., Williams B.C. Cooperative task and motion planning for multi-arm assembly systems. *arXiv preprint arXiv:2203.02475*. 2022.
11. LaValle S. Rapidly-exploring random trees: A new tool for path planning. *Research Report 9811*. 1998.
12. Kuffner J.J., LaValle S.M. Rrt-connect: An efficient approach to single-query path planning. *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*. IEEE, 2000. vol. 2. pp. 995–1001.
13. Lacevic B., Osmankovic D. Improved c-space exploration and path planning for robotic manipulators using distance information. *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020. pp. 1176–1182.
14. Karaman S., Frazzoli E. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*. 2011. vol. 30(7). pp. 846–894.
15. Zhang D., Zhu G., Zhang Q. Multi-robot motion planning: A learning-based artificial potential field solution. *2nd Conference on Fully Actuated System Theory and Applications (CFASTA)*. IEEE, 2023. pp. 233–238.
16. Shaoul Y., Mishani I., Vats S., Li J., Likhachev M. Multi-robot motion planning with diffusion models. *arXiv preprint arXiv:2410.03072*. 2024.
17. Aljalbout E., Karl M., van der Smagt P. Clas: Coordinating multi-robot manipulation with central latent action spaces. *Learning for Dynamics and Control Conference*. PMLR, 2023. pp. 1152–1166.
18. Sharon G., Stern R., Felner A., Sturtevant N.R. Conflict-based search for optimal multi-agent pathfinding. *Artificial intelligence*. 2015. vol. 219. pp. 40–66.
19. Ebdend R., Drechsler R. Weighted a search—unifying view and application. *Artificial Intelligence*. 2009. vol. 173(14). pp. 1310–1342.
20. Li J., Ruml W., Koenig S. Eecbs: A bounded-suboptimal search for multi-agent path finding. In *Proceedings of the AAAI conference on artificial intelligence*. 2021. vol. 35(14). pp. 12353–12362.

Zaitsev Andrey – Student, Faculty of mathematics and computer science, St. Petersburg State University. Research interests: computer science, data science. The number of publications — 3. Dusha.Zaitsev@yandex.ru; 7–9, University Emb., 199034, St. Petersburg, Russia; office phone: +7(812)328–2000.

Yakovlev Konstantin – Ph.D., Associate Professor, Leading researcher, Faculty of mathematics and computer science, Federal Research Center for Computer Science and Control of the Russian Academy of Sciences (FRCSC RAS); Head of the laboratory, Markov laboratory, St. Petersburg State University. Research interests: artificial intelligence, intelligent dynamic systems, intelligent control systems, intelligent robotics, intelligent planning, trajectory planning, heuristic search, multi-agent systems, cognitive agents. The number of publications — 75. yakovlev@isa.ru; 44/2, Vavilova St., 119333, Moscow, Russia; office phone: +7(926)270-8979.