Тема сегодняшнего доклада – «An improved RRT\* algorithm for robot path planning based on path expansion heuristic sampling», это работа группы 6 исследователей из Чунцинского технологического института в Китае.

В работе исследуется задача планирования пути. Существует множество подходов к ее решению – есть метод потенциальных полей, бионические алгоритмы, поиск по решетке и многие другие. Частыми проблемами является работа с пространствами малой размерности, требования к дискретизуемости пространства, попадание в локальные оптимумы, кроме того, препятствия в пространстве могут образовывать сложные рельефы, такие как узкие коридоры или лабиринты.

Отдельно нужно сказать про алгоритмы, основанные на сэмплировании. В первую очередь, надо упомянуть RRT. Он предлагает последовательно строить дерево с корнем в начальной точке. Каждый узел дерева – точка в пространстве. Алгоритм состоит в том, что до тех пор, пока ни один из узлов дерева нельзя соединить с конечной точкой, в дерево добавляется новый лист: выбирается случайная точка в пространстве, в дереве находится ближайшая к ней точка, после чего из ближайшей в случайную точку проводится луч. Если луч достигает выбранной случайной точки, не сталкиваясь с препятствиями, то в дерево добавляется эта точка как потомок ближайшей, в противном случае в дерево добавляется точка коллизии луча с препятствием.

У RRT есть множество модификаций, например RRT\*. Метод предлагает при расширении дерева рассмотреть окрестность новой точки, и 1) новая точка становится потомком той точки дерева, путь через которую от начальной точки минимален; 2) предпринимается попытка сделать точки окрестности новой точки ее потомками, если в таком случае путь в них из начальной точки станет меньше.

Другой важный вариант RRT – Informed RRT\*, который после построения какого-либо неоптимального пути предлагает оптимизировать этот путь, ограничив область сэмплирования эллипсом, размеры осей которого уменьшаются с уменьшением длины найденного оптимального пути.

Наконец, стоит упомянуть RRT-Connect, который предлагает строить два дерева: из начальной точки в конечную, и из конечной в начальную. Этот алгоритм работает быстрее всего среди всех перечисленных.

На слайде приведена постановка задачи. Имеем пространство X, множество препятствий и начальную и конечную точки. В пространстве можно строить пути, вводится функция стоимости пути. Необходимо найти путь из начальной точки в конечную, не пересекающий препятствия и имеющий минимальную стоимость.

В работе предлагается алгоритм, названный EP-RRT\*. Он включает в себя два этапа:

1. Найти неоптимальный путь с помощью быстрого RRT-Connect
2. Оптимизировать этот путь

При оптимизации предлагается так же как и в Informed RRT\*, на каждой итерации ограничить область сэмплирования, в данном случае – объединением четырехугольников вокруг оптимизируемого пути. Область имеет определенную ширину, которая динамически уменьшается с уменьшением стоимости найденного оптимального пути.

На слайде приведены математические выкладки для получения координат точек многоугольника, ограничивающего область сэмплирования, из координат точек построенного пути.

Эксперимент ставился следующим образом. Было смоделировано четыре среды, для каждой среды исследовался каждый из алгоритмов: RRT\*, Informed RRT\*, EP-RRT\*, каждый алгоритм запускался 100 раз. Для каждого запуска вычислялись величины: – число итераций, необходимое для получения субоптимального пути стоимостью ; кроме того, вычислялась величина итогового пути, полученного алгоритмом, .

На слайде приведены результаты для обычной среды. Можно видеть, что как и ожидалось, RRT\* беспорядочно сэмплирует точки во всем пространстве, Informed RRT\* сэмплирует в пределах уменьшающегося эллипса, EP-RRT\* сэмплирует в достаточно узкой области пространства, он и имеет лучшую скорость сходимости, затем идет Informed RRT\*, и затем RRT\*. Алгоритмы ранжируются в таком же порядке и по возрастанию стоимости оптимального пути. Справа приведены боксплоты для величин и , на которых также виден тот же порядок качества работы алгоритмов.

В хаотичной среде предложенный метод оказывается немного лучше Informed RRT\*, который тоже под нее приспособлен. Алгоритмы достаточно быстро сходятся, а RRT\* тоже сходится, но значительно позже. По боксплотам также видно, что Informed RRT\* действительно подходит для этой задачи и дает результат ненамного хуже, хотя и сходится медленнее.

В случае с узким коридором EP-RRT\* работает отлично, а другие алгоритмы сходятся медленнее и дают результат намного хуже. Можно видеть, что Informed RRT\* в такой среде работает очень плохо и, судя по всему, хуже RRT\*.

В случае лабиринта EP-RRT\* опять работает лучше остальных алгоритмов, и если верить боксплотам, немного предпочтительнее здесь RRT\*, чем Informed RRT\*, хотя на графике он сходится к лучшему решению.

В заключение стоит сказать, что предложенный метод увеличивает вероятность сэмплирования важных точек, что повышает скорость сходимости. Метод может быть совмещен с любым алгоритмом, основанном на сэмплировании.

Результаты экспериментов показывают, что предложенный метод по сравнению с прочими быстрее сходится и дает более оптимальное решение, в особенности, в случае узких коридоров и лабиринтов. Алгоритм вдохновлен идеей Informed RRT\*, но работает отлично в тех средах, где Informed RRT\* работает очень плохо.

Однако в средах, где есть множество альтернативных путей различной стоимости, алгоритм требует области расширения больших размеров, чтобы избежать попадания в локальный оптимум.

Дальнейшие исследования могут быть направлены на изучение возможности адаптивной настройки параметров предложенного алгоритма.