

Валерия Глотка

студентка 2 курса специальности

121 «Инженерия программного обеспечения»

Научный руководитель: **Н. Е. Хацько**,

канд. техн. наук, доцент кафедры программной инженерии
Харьковский национальный университет радиоэлектроники

г. Харьков

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ ДЛЯ ПЛАНИРОВАНИЯ ПУТИ БЕСПИЛОТНЫХ УСТРОЙСТВ

Планирование пути – важнейшая задача в области навигации мобильных роботов. Во-первых, спланированный путь должен пролегать от заданной начальной точки к заданной конечной точке. Во-вторых, этот путь должен обеспечивать движение робота с обходом возможных препятствий. В-третьих, путь должен среди всех путей, удовлетворяющих первым двум требованиям, быть в определенном смысле оптимальным.

Методы на основе графов [1]. Основные методы этого класса отражают состояния, в которых может находиться робот. Переходы между состояниями характеризуются функцией затрат. Это позволяет выделить путь, который имеет минимальную общую стоимость достижения цели.

Алгоритмы поиска по графу получают в качестве входных данных граф. Граф – это набор точек («узлов») и соединений («рёбер») между ними. Определяемый алгоритмом путь состоит из узлов и рёбер. Алгоритм сообщает нам, что нужно перемещаться из одной точки в другую, но не сообщает, как это нужно делать. Мы сами должны решить, чем будет являться ребро графа, возвращённое алгоритмом – движением по прямой линии, открытием двери или бегом по кривому пути.

Существует множество алгоритмов, работающих с графами. Поиск в ширину выполняет исследование равномерно во всех направлениях. Алгоритм Дейкстры позволяет нам задавать приоритеты исследования путей. Вместо равномерного исследования всех возможных путей он отдаёт предпочтение путям с низкой стоимостью. В жадном поиске по первому наилучшему совпадению для

порядка очереди с приоритетами используется оцененное расстояние до цели. Точка, ближайшая к цели, будет исследована первой. A^* – это модификация алгоритма Дейкстры, оптимизированная для единственной конечной точки. Алгоритм Дейкстры может находить пути ко всем точкам, A^* находит путь к одной точке. Он отдаёт приоритет путям, которые ведут ближе к цели.

Сформулируем гипотезу: алгоритм A^* эффективнее, чем алгоритм Дейкстры и жадный алгоритм.

Алгоритм Дейкстры хорош в поиске кратчайшего пути, но он тратит время на исследование всех направлений, даже бесперспективных. Жадный поиск исследует перспективные направления, но может не найти кратчайший путь. Алгоритм A^* использует и подлинное расстояние от начала, и оцененное расстояние до цели.

A^* берёт лучшее от двух алгоритмов. Поскольку эвристика не оценивает расстояния повторно, A^* не использует эвристику для поиска подходящего ответа. Он находит оптимальный путь, как и алгоритм Дейкстры. A^* использует эвристику для изменения порядка узлов, чтобы повысить вероятность более раннего нахождения узла цели.

Проверим работу этих 3-х алгоритмов на реальном примере. Предположим, что у нас есть беспилотный автомобиль, который должен припарковаться в указанной точке. Смоделируем ситуацию достижения конечной точки на участке местности. На рис. 1.1 и рис. 1.2 изображен участок улицы (1.1) и выделенные на карте цветом препятствия (1.2), которые непреодолимы для автомобиля.



Рис. 1.1



Рис. 1.2

С помощью программы на Python смоделируем маршрут [2] авто по алгоритму Дейкстры (рис. 1.3), жадному алгоритму (рис. 1.4) и A^* (рис. 1.5).

