

Методы и алгоритмы построения оптимального пути

Белков Д.Н., Пономарев И.В.

Алтайский государственный университет, г. Барнаул

dan120699@mail.ru, igorpon@mail.ru

Аннотация

В статье рассматриваются методы для построения оптимального пути с учетом специфики окружающей обстановки. Начиная от создания графа и заканчивая самим поиском пути. Все действия будут производиться на подробной карте местности, т.е. на карте, на которой изображены все различимые объекты местности.

Ключевые слова: граф, коэффициент преодолемости, алгоритм A-star.

Для того, чтобы построить оптимальный путь, понадобится точки, через который этот маршрут будет строиться. Логичным решением, будет взять за отдельную точку какой-либо объект местности либо его ярко-выраженные изгибы. Так как у нас карта местности подробная, то и точек будет достаточно большое количество. Как раз это нам и будет давать большую гибкость при построении оптимального маршрута.

Подробную карту местности используют не везде. Но она всегда используется в спортивном ориентировании, отсюда мы и возьмем пример карты. Для начала, стоит ознакомиться с некоторыми условными знаками карт.

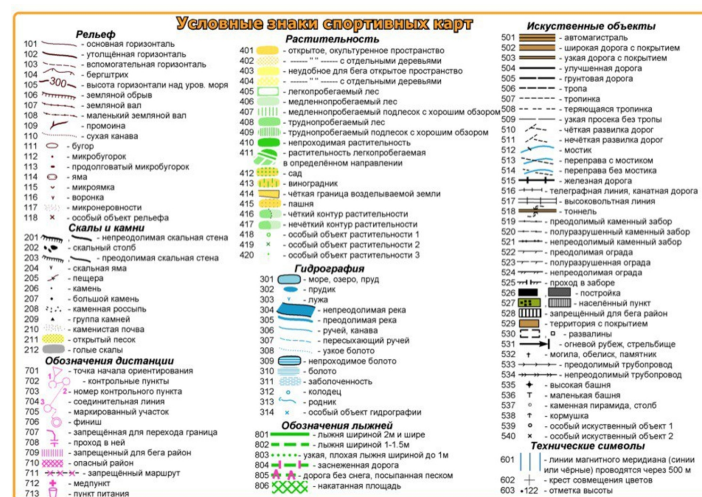


Рисунок 1. Условные знаки

Карта состоит полностью из условных знаков, приведенных на рисунке 1. Для примера возьмем небольшую часть карты и расставим на ней точки.

Для того, чтобы была возможность перемещаться по данным точкам, необходимо расставить связи между точками. Строить полный граф мы не можем, так как из-за этого потеряется смысл данной задачи. Из одной точки в любую другую, можно будет пройти за один шаг и через непреодолимые объекты. Нужно как-то ограничить количество отходов от точки. Для этого, существует отличный метод построения графа – “Метод единичных кругов” [1–3]. Суть данного метода является в том, что у каждой точки имеется окружность единичного радиуса, и те точки, окружности которых пересекаются, будут иметь

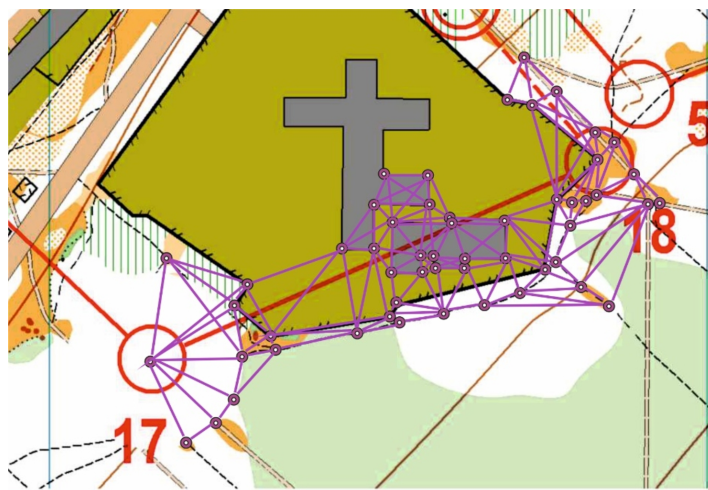


Рисунок 2. Пример построения графа

связь друг с другом. Данная окружность имитирует $\frac{1}{2}$ радиуса видимости объектов на местности. Это значит, что все точки, которые находятся в радиусе видимости той точки, в которой мы находимся, будут иметь с ней связь.

Теперь нужно избавиться от тех связей, которые проходят через непреодолимые объекты и другие области растительности. Для этого, для каждой точки, в ее окружности проверяется наличие непреодолимых объектов или других областей растительности. Если такие есть, то находятся крайние точки объекта или области, через которые, проведенный луч от рассматриваемой точки не будет пересекать непреодолимый объект либо область. Основываясь на этих лучах, строится новая область, так называемая “Слепая зона” для рассматриваемой точки. Почему здесь же рассматриваем и область растительности? Потому как, передвигаясь из одной области в другую, мы в любом случае будем попадать на границу этой области. Так же, с помощью данного подхода, будет проще использовать коэффициент проходимости в дальнейших методах.

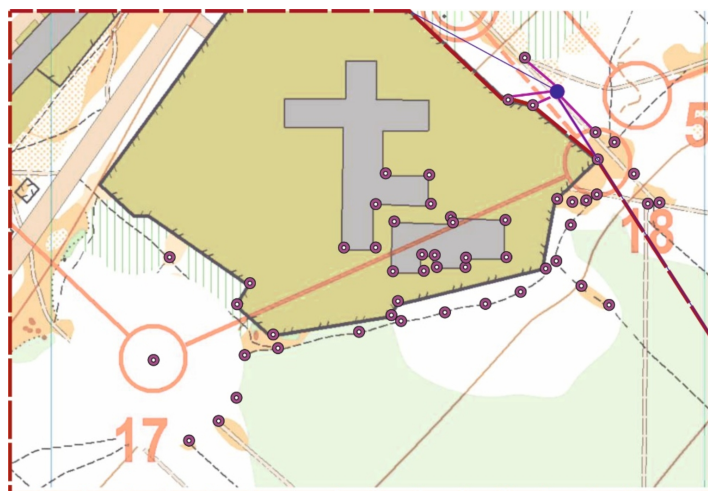


Рисунок 3. Построение “Слепой зоны”

Все точки, находящиеся в “Слепой зоне”, будут недоступны для рассматриваемой точки и не будут иметь с ней связей. Тем самым, мы избавляемся от невозможных связей.

Имея верно построенный граф, появляется возможность найти оптимальный путь из точки в любую другую точку. Для этого отлично подходит алгоритм A^* (A-star) [4]. Данный алгоритм находит один единственный путь, который является самым оптимальным. Для его расчёта, в алгоритме используется вспомогательная функция (эвристика). Она

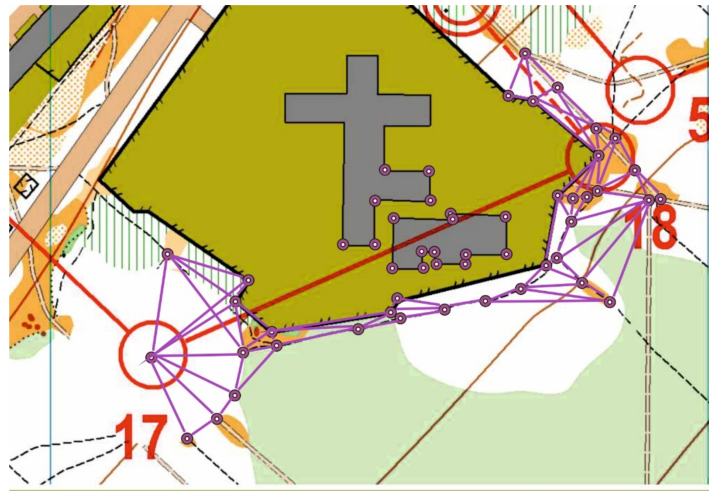


Рисунок 4. Правильно построенный граф

нужна, чтобы устанавливать направление алгоритма поиска и считать приближенные затраты на преодоление маршрута до конечной точки. Выглядит она следующим образом:

$$F(x_i) = g(x_i) + h(x_i),$$

$$g(x_i) = g(x_{i-1}) + w(x_{i-1}, x_i) \cdot k(x_i),$$

где $g(x_i)$ – оптимальное расстояние от начальной точки до точки x_i ; $w(x_{i-1}, x_i)$ – расстояние от точки x_{i-1} до x_i ; $k(x_i)$ – коэффициент преодолимости (он будет варьироваться от 1 до 100); $h(x_i)$ – некая функция, которая задает направление алгоритму.

В данном случае за данную функцию $h(x_i)$, берется прямое расстояние (расстояние по птичьему полету) от вершины x_i до конечной.

Алгоритм делит вершины на три класса:

- **Неизвестные вершины:** вершины к которым не известен путь. В начале все вершины, кроме начальной будут относиться к данному классу.
- **Известные вершины (открытый список):** здесь содержатся все вершины, которые являются смежными с вершинами в закрытом списке. Исключая сами вершины закрытого списка. К этим вершинам мы знаем (возможно не оптимальный) путь и знаем, чему равна эвристика у каждой из вершин. В начале работы алгоритма, к известным вершинам относится только начальная вершина.
- **Исследованные вершины (закрытый список):** до этих вершин мы знаем кратчайший путь. Вершины заносятся в данный список, для того, чтобы не рассматривать их в дальнейшей работе алгоритма.

Каждая вершина из открытого или закрытого списка будет иметь указатель на предыдущую вершину. Благодаря этому, мы сможем восстановить свой маршрут от конечной точки [5, 6].

Чтобы было проще совершать расчеты с коэффициентами преодолимости мы создадим для них отдельный список. В него будут заноситься коэффициенты тех граничных точек областей, через которые мы проходим. Этот список будет действовать по принципу “скобок”. Если мы входим в какую-нибудь область, то (открываем скобку) заносим в список коэффициент проходимости данной области, он берется из ее граничной точки. Когда мы будем выходить из этой области, то удаляем последний коэффициент в списке, и будем считать эвристику со следующим последним коэффициентом (закрываем скобку

и удаляем эту пару скобок). Таким образом будет происходить сохранение прошедшего коэффициента преодолимости и когда мы снова вернемся в данную область, то будем считать эвристику с коэффициентом преодолимости для данной области.

Начиная со стартовой точки, в открытый список заносится все смежные с ней точки. Сама же стартовая точка заносится в закрытый список, и мы к ней больше не будем возвращаться. Считается эвристика для всех точек открытого списка, и выбирается точка с наименьшим значением эвристики. Далее рассматриваем уже эту точку и заносим ее в закрытый список. Снова в открытый список заносится все смежные с данной точки. Считается эвристика для точек открытого списка. И выбирается точка с наименьшим ее значением. И так продолжается до тех пор, пока не достигнем конечной точки.

В конце, восстанавливаем оптимальный путь по указателям на предыдущие точки из закрытого списка.

Приведенные в данной статье методы, отлично подходят для решения подобных типов задач. Простота алгоритмов позволяет производить вычисления за короткое время. Методы являются надежными и эффективными, что говорит о доверии к выбранному оптимальному пути.

Список литературы

1. Нильсон Н. Искусственный интеллект. Методы поиска решений. — М. : Мир, 1973.
2. Карпов Д.Н. Теория графов. — 2017.
3. Муратшин Б. М* – алгоритм поиска кратчайшего пути, через весь мир, на смартфоне [Электронный ресурс]. — URL: <https://habr.com/ru/company/2gis/blog/326638>.
4. Amit Patel. Введение в алгоритм A* [Электронный ресурс]. — URL: <https://habr.com/ru/post/331192>.
5. Madhav V. Marathe, H. Breu, Harry B. Hunt III et al. Geometry based heuristics for unit disk graphs // arXiv:math/9409226 [math.CO]. — 1994.
6. Козинев Е.А., Сиднев А.А. Образовательный комплекс. «Параллельные численные методы». Лабораторная работа Поиск путей на графе. — Нижний Новгород : Нижегородский государственный университет им. Н.И. Лобачевского, 2011.