

УДК 007.52

ПОИСК МАРШРУТА ДЛЯ НАЗЕМНОГО РОБОТА: МОДИФИЦИРОВАННЫЙ АЛГОРИТМ ПЛАНИРОВАНИЯ НА ОСНОВЕ СПЛАЙНОВ

Р.О. Лавренов (*lavrenov@it.kfu.ru*)Е.А. Магид (*magid@it.kfu.ru*)Высшая Школа Информационных Технологий и Систем,
Казанский Федеральный Университет, Казань

Аннотация. Методы, основанные на принципе потенциальных полей, играют важную роль в задачах планирования пути. Но они страдают от проблемы застревания пути в локальных минимумах функции потенциального поля. В этой статье мы предлагаем модификацию оригинального сплайн-алгоритма для планирования маршрута наземного мобильного робота путем интеграции графа Вороного, благодаря чему успешно решаем проблему локальных минимумов. В новый алгоритм добавлены дополнительные критерии видимости стартовой и целевой точек для оптимизации выбора пути. Алгоритм был реализован в среде Matlab, и результаты моделирования продемонстрировали, что нам удалось преодолеть проблемы исходного алгоритма.¹

Ключевые слова: планирование маршрута; мобильный робот; граф Вороного; метод потенциального поля; безопасность маршрута.

Введение

В настоящее время разработки в сфере робототехники направлены на замену человеческого труда в случаях, когда это существенно ускорит или обезопасит какой-либо производственный процесс или проведение поисково-спасательных операций в опасных для людей условиях [Magid и др., 2011]. При этом, современные мобильные роботы должны обладать такими критическими навыками как взаимодействие с пользователями

¹ Работа поддержана Российским фондом фундаментальных исследований (РФФИ) и Министерством науки и технологий и Государством Израиль (совместный проект ID 15-57-06010). Часть работ выполнена в соответствии с Государственной программой конкурентоспособности Российской Федерации при Казанском федеральном университете.

[Pipe и др., 2014] и другими роботами [Rosenfeld и др., 2015] [Ronzhin и др., 2016] [Panov и др., 2017], автономное перемещение [Magid, 2006], выстраивание карты окружающего пространства и локализации на карте (SLAM) [Buyval и др., 2016]. Все вышеперечисленные навыки в обязательном порядке требуют интеграции с планировщиком маршрута.

Классический, но остающийся популярным, метод планирования маршрута – метод потенциального поля [Khatib и др., 2016]. В этом подходе со стороны целевой точки на робота действуют силы притяжения, а со стороны препятствий – силы отталкивания. Робот движется по линии градиента в направлении минимума потенциального поля. Потенциалы поля могут рассчитываться как для всей карты целиком, когда она известна заранее (глобальное планирование пути), так и только для некоторой локальной области карты, когда карта неизвестна роботу (локальное планирование пути). Недостатками этого метода является застревание робота в местах локальных минимумов потенциального поля или возникающие в узких проходах между препятствиями осцилляции.

В предыдущем исследовании нами был предложен метод планирования пути для мобильного робота, основанный на сплайнах [Magid и др., 2006]. Данный алгоритм использует принципы методов потенциальных полей и критерии оптимизации маршрута - такие как длина, кривизна и безопасность маршрута. Чтобы улучшить алгоритм, добавить гибкость для оптимизации пути и возможность быстрого динамического перепланирования, мы интегрируем подход диаграммы Вороного и новый критерии видимости точек старта и цели в наш исходный метод.

1. Оригинальный метод планирования пути

Алгоритм, описанный в работе [Magid и др., 2006], предлагает решение задачи планирования пути для мобильного робота, представленного точкой в пространстве конфигураций [Choset, 2005], в известной среде со статическими препятствиями в виде конечной группы пересекающихся кругов разного размера. Идея такого представления в том, что любой многоугольник можно аппроксимировать группой кругов.

1.1 Целевая функция

Чтобы гарантировать построение пути, не пересекающего препятствия, отталкивающая потенциальная функция должна обладать высоким значением внутри препятствия и на его границе, и небольшим значением вне препятствия. Во время локальной процедуры оптимизации пути, высокое значение потенциальной функции внутри препятствия выталкивает все точки пути из него наружу, чтобы минимизировать стоимость пути. Определим положение робота в момент времени t как

конфигурацию $q(t)=(x(t),y(t))$. Тогда вклад одного круга (где круг является частью препятствия) в общую функцию потенциала отталкивания определяется следующим уравнением:

$$U_{rep}(q) = 1 + \tanh(\alpha(\rho - \sqrt{(x(t) - x)^2 + (y(t) - y)^2})), \quad (1)$$

где ρ - это радиус круга (препятствия) с центром в точке с координатами (x,y) , а α - это эмпирически определяемый параметр, который отвечает за выталкивание пути за границы препятствия. Рисунок 1 (справа) демонстрирует пример выбора параметра $\alpha=0.5$ для среды с одним препятствием, которое образовано тремя пересекающимися кругами (слева). Потенциальная функция имеет четко выделяющиеся пики на пересечениях кругов.

Топология $T(q)$ – функция, учитывающая все N кругов окружающей среды и их влияние на работа на протяжении всего пути, которая определяется параметрическим уравнением на интервале $[0,1]$:

$$T(q) = \sum_{j=0}^{N-1} \int_{t=0}^1 U^j_{rep}(q) \cdot \delta l(t) \cdot dt, \quad (2)$$

где $\delta l(t)$ длина отрезка:

$$\delta l(t) = \sqrt{(x'(t))^2 + (y'(t))^2} \quad (3)$$

Функция **кривизны** пути также определяется параметрическим уравнением:

$$R(q) = \sqrt{\int_{t=0}^1 (x''(t))^2 + (y''(t))^2 dt} \quad (4)$$

Функция **длины** пути, суммирует длины всех сегментов пути:

$$L(q) = \int_{t=0}^1 \delta l(t) \cdot dt \quad (5)$$

Таким образом, итоговая стоимость пути состоит из суммы трех компонентов:

$$F(q) = \gamma_1 T(q) + \gamma_2 R(q) + \gamma_3 L(q), \quad (6)$$

где $\gamma_{i=1,3}$ - числовые значения влияния каждого компонента на общую стоимость маршрута. В частности, значение влияния потенциального поля $\gamma_1 = \beta/2$, где параметр β коррелирует с α из уравнения (1) [Magid, 2006]. В программной реализации интегралы считаются из разбиения кривой на отрезки с длиной, стремящейся к нулю.

1.2 Алгоритм

Оригинальный сплайн-алгоритм является итеративным. Путь на первой итерации представляет из себя сплайн, который определяется тремя точками: S , T и равноудаленной точкой, лежащей на прямом отрезке, соединяющем точки S и T . Дальнейшие итерации расчета пути используют

уравнение (6) в роли целевой функции стоимости и оптимизируют маршрут относительно его стоимости при помощи метода Нелдера-Мида [Lagarias и др., 1998] (симплекс-метод). Лучший результат (т.е. маршрут) каждой итерации служит начальным значением (маршрута) для следующей итерации. Маршрут перестраивается на каждой итерации, используя информацию с предыдущего этапа, увеличивая количество точек сплайна на единицу и корректируя параметры целевой функции стоимости. Алгоритм завершается, если количество итераций превышает заданный пользователем предел, или если новая итерация не улучшает предыдущую.

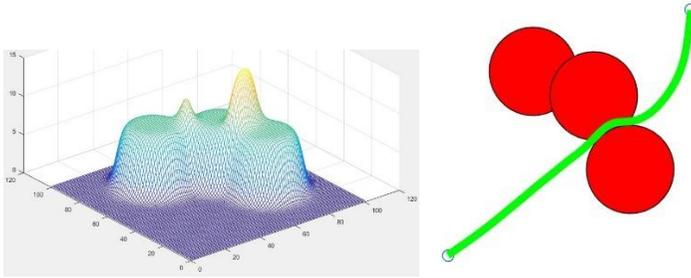


Рис. 1. Потенциальная функция одиночного препятствия, состоящего из трех кругов (слева) и путь, рассчитанный оригинальным алгоритмом (справа)

1.3 Слабые стороны оригинального алгоритма

Исходный сплайн-метод позволяет получить гладкий путь без столкновений с препятствиями, если каждое препятствие аппроксимируется одним кругом или линия начальной итерации не проходит через пересечение кругов. Однако, когда препятствия аппроксимированы группой пересекающихся кругов, пересечения образуют локальные максимумы потенциального поля (рис. 1, слева). Когда в такой среде начальный сплайн проходит через пересечение нескольких кругов, то целевая функция $F(q)$ выталкивает сплайн из области пересечения и далее сплайн «застревает» в локальных минимумах, а следующий итерационный сплайн не может «перепрыгнуть» некоторые компоненты препятствия из-за локального характера процесса оптимизации. На рис. 1 (слева) демонстрируется пример с одним сложным препятствием. Локальная оптимизация успешно устраняет локальные максимумы, но даже после 18 итераций, продолжавшихся в течение 23 минут, результирующий путь все еще пересекает препятствие и, следовательно, бесполезен для движения робота (рис. 1, справа).

2. Интеграция дополнительного критерия оптимизации

В этом разделе мы вводим два новых критерия оценки качества маршрута – время видимости стартовой и целевой точки – и исследуем их влияние на результирующий путь. Время видимости точки – это процентное отношение длины пути, где робот сохраняет точку старта (или целевую точку соответственно) в пределах своей прямой видимости.

Эти два критерия важны, если роботу необходимо максимизировать время прямого и непрерывного визуального или радиоконтакта с контрольным устройством или маршрутизатором, который поддерживает планирование маршрута, локализацию или любые другие функции робота. Критерий учитывает отношение видимого и невидимого от стартовой (или целевой) точки участков пути. Таким образом, мы минимизируем время, которое фактически измеряется как сумма длины участков пути (предполагается постоянная скорость робота, а в будущей работе мы также распространяем это на случаи различной скорости вдоль пути), на которых контрольные точки недоступны для прямой видимости робот. Значение критерия для начальной точки старта S вычисляется по формуле:

$$I_S = 1 - \lim_{\delta, \delta(t) \rightarrow 0} \frac{\sum_{t=0}^u \text{dist}(q(t), q(t+\delta t))}{\int_{t=0}^1 \delta(t) \cdot dt} \quad (7)$$

$$\forall t \in [0, u + \delta]: [q(t), S] \cap (\cup_{j=1}^N \text{Obs}_j) = 0$$

где числитель дроби в уравнении 7 отражает длину пути, видимого из начального положения S , а знаменатель дроби отражает полную длину пути из уравнения 5 и считается через аппроксимацию кривой отрезками с длиной, стремящейся к нулю. $q(t)$ - положение робота в момент времени t . Короткие отрезки пути, которые были пройдены между моментами времени t и $t+\delta t$, обозначаются $\text{dist}(q(t), q(t+\delta t))$. Нижняя строка уравнения 7 описывает свойство видимости: прямой отрезок $[S; q(t+\delta t)]$ не пересекается ни с каким препятствием Obs_j , где $j=1..N$, а N – число препятствий в окружающей среде. Таким образом, последняя видимая точка пути от стартовой точки описывается с помощью $q(u+\delta t)$. Аналогичным образом описывается критерий, который показывает отношение сегментов пути с прямой видимостью целевой точки T :

$$I_T = 1 - \lim_{\delta, \delta(t) \rightarrow 0} \frac{\sum_{t=w}^{1-\delta} \text{dist}(q(t), q(t+\delta t))}{\int_{t=0}^1 \delta(t) \cdot dt} \quad (8)$$

$$\forall t \in [w, 1]: [q(t), T] \cap (\cup_{j=1}^N \text{Obs}_j) = 0$$

Уравнение 8 описывает первую точку $q(w)$ сегмента пути $[q(w), q(I)=T]$, которая отмечает начало последнего сегмента пути, характеризующегося гарантированной прямой видимостью робота из целевой позиции T .

Целевая функция, которая объединяет все пять критериев, определяется следующим образом:

$$F(q) = \gamma_1 T(q) + \gamma_2 R(q) + \gamma_3 L(q) + \gamma_4 I_S + \gamma_5 I_T, \quad (9)$$

где γ_4 и γ_5 – весовые коэффициенты новых критериев в целевой функции.

На рис. 2 показано влияние двух новых критериев на найденный путь: в то время как при $\gamma_4 = 10$ и $\gamma_5 = 10$ эти критерии оказывают существенного влияния (рис. 2, слева), при $\gamma_4 = 20$ и $\gamma_5 = 20$ изменения траектории хорошо заметны (рис. 2, справа). Другие параметры установлены эмпирически как $\gamma_1 = 1$, $\gamma_2 = 1$ и $\gamma_3 = 0,5$ для обоих случаев.

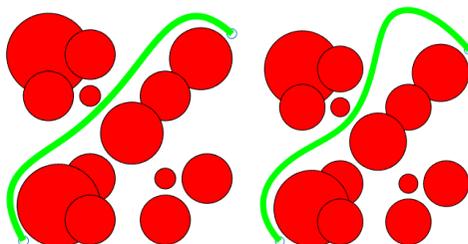


Рис. 2. Влияние критерия видимости начальной и конечной точек на маршрут

Так как оптимизация маршрута с помощью уравнения 9 выполняется только локально, влияние дополнительных параметров также локально. Путь остается в той же гомотопической группе даже в том случае, когда можно потенциально обойти какое-либо препятствие с другой его стороны. В рамках нашей будущей работы мы планируем применить оптимизацию в глобальном масштабе, которая позволит выбирать наиболее оптимальные пути среди нескольких гомотопий.

3. Решение на основе диаграммы Вороного

Для получения хорошего начального сплайна, мы строим граф Вороного. Для создания этого графа требуется подготовить информацию о препятствиях; процедура выполняется в два шага:

Сбор препятствий из пересекающихся кругов. Первоначально каждый круг отмечен как неиспользованный и имеет собственный индекс $i=1...M$, где M – число кругов в среде. Начнем с произвольного неиспользованного круга, закрепим его как принадлежащий препятствию O_I и пометим как использованный. Затем итеративно увеличиваем O_I , определяем все

неиспользованные круги, которые пересекаются с кругами препятствия O_1 , закрепляем их за O_1 и отмечаем как использованные до тех пор, пока в среде не останется больше неиспользованных кругов, которые пересекаются с O_1 . Если остаются еще неиспользованные круги, мы произвольно выбираем другой неиспользованный круг, закрепляем его за препятствием O_2 и повторяем процедуру роста. Сбор завершается, когда не остается неиспользованных кругов. Например, на рис. 2 есть 7 препятствий, которые образованы группами кругов, а на рис. 3 в среде находятся 12 препятствий. При сборке препятствий каждая пара пересекающихся кругов (i, j) образует одну точку пересечения θ_{ij} в случае граничного касания и пару точек θ_{ij}, θ_{ji} в общем случае пересечения кругов.

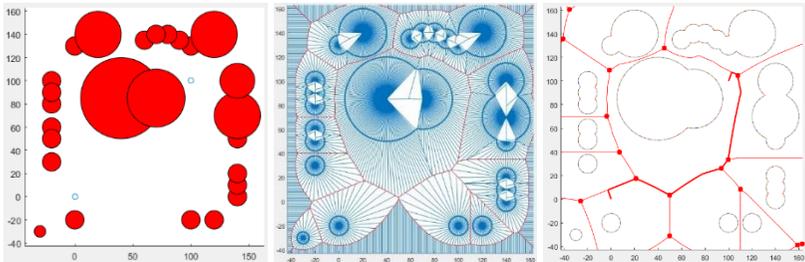


Рис. 3. Среда с препятствиями (слева), процедура построения на ней графа Вороного (центральное), построенный граф и путь в графе (справа)

Поиск внешних контуров препятствий множества $Obst = \{O_1, \dots, O_k\}$, где k - количество препятствий в окружающей среде. Исходя из радиуса самого маленького круга внутри препятствия O_1 , границы всех кругов, принадлежащих O_1 , разбиваются на короткие отрезки длиной σ , соединенные через точки пересечения θ_{ij}, θ_{ji} . Далее, каждые два сегмента соединяются, если между ними существует непрерывный путь, который построен из граничных сегментов. Эта процедура повторяется для каждого препятствия, и по ее завершению мы получаем группу контуров. Если размер последнего множества превышает k , это указывает на наличие внутренних контуров. Чтобы избавиться от таких контуров, для каждого конкретного препятствия O_i вычисляется выпуклая оболочка и удаляются те контуры, которые находятся внутри выпуклой оболочки. Таким образом, мы получаем несколько невыпуклых многоугольников, по одному для каждого препятствия из $Obst$, и инкапсулируем их в квадратную карту.

Далее, имея карту препятствий в виде многоугольников, строится граф Вороного [Choset, 2005]:

От границ препятствий и карты с равными промежутками строятся лучи. На рис. 3 (посередине) демонстрируется пример построения этих

лучей в виде тонких синих линий, а толстые синие линии отображают границы препятствий и карты.

Вычисляются точки пересечения лучей от разных препятствий. Точки пересечения соединены отрезками. Соединяя отрезки, мы образуем граф Вороного (тонкие красные линии на рис. 3, посередине и справа)

После построения графа Вороного (VG), для присоединения к графу начальной позиции S и целевой позиции T на графе выбираются точки S' и T' , ближайшие к точкам S и T , соответственно. Отрезки $[S, S']$ и $[T, T']$ должны не пересекать препятствия. Кратчайший путь между S' и T' находится в графе VG алгоритмом Дейкстры [Dijkstra, 1959]. Так, на рис. 3 (справа) построенный путь изображен толстыми красными линиями. Любой путь между точками (S, T) на графе Вороного VG гарантированно не пересекает препятствия и максимально безопасен в отношении расстояния от границ препятствий. Таким образом, он может обеспечить хороший начальный сплайн для исходного сплайн-метода [Magid и др., 2006].

Однако, нужно правильно выбрать точки для построения сплайна. На первом шаге выбирается точка S в качестве активной точки и находится точка VP_1 на пути по графу Вороного VG , таким образом, что следующая за VP_1 точка на графе VG уже не видна из S из-за окклюзии препятствиями. S добавляется к множеству L , которое хранит точки графа VG , характеризующие данный маршрут и достаточные для его описания. VP_1 получает статус следующей активной точки. Затем мы снова ищем на пути самую дальнюю точку, видимую из VP_1 – точку VP_2 . VP_1 добавляется в L , VP_2 становится следующей активной точкой, и процесс итеративно продолжается до тех пор, пока не станет видимой целевая точка T . Определим, что точка VP_{i+1} видна из точки VP_i , если они могут быть соединены прямым отрезком, который не пересекает ни одного препятствия окружающей среды. После завершения процесса точки множества L используются как точки для начального сплайна при запуске модифицированного сплайн-метода.

4. Результаты симуляций

Модифицированный алгоритм был реализован в среде Matlab и были проведен ряд виртуальных экспериментов (симуляций). Особое внимание было уделено случаям, когда оригинальный алгоритм [Magid и др., 2006] не мог найти путь для робота. Целевая функция стоимости (уравнение 9) использовалась со следующим набором эмпирически установленных параметров: $\gamma_1 = 1$, $\gamma_2 = 1$ и $\gamma_3 = 0,5$, $\gamma_4 = 5$ и $\gamma_5 = 5$. Алгоритму удалось найти пути во всех случаях, что явилось естественным следствием того, что начальный сплайн получался из графа Вороного. Граф Вороного считается за приемлемо малое время, которое гораздо меньше времени оптимизации

пути. Например, для среды, показанной на рис. 4 (а) начальный расчет пути на основе графа Вороного занял всего 2 секунды, а для рис. 4 (в) - 6 секунд. Общее время работы нового алгоритма уменьшилось в среднем в три раза по сравнению с исходным алгоритмом. Таким образом, конечный путь рис. 4 (б) был рассчитан всего за 2 итерации в течение 2,5 минут в Matlab, тогда как исходный сплайн-алгоритм провел 18 итераций и 38 минут, чтобы сделать вывод о своей неспособности найти приемлемый путь от точки старта до целевой точки. В другом случае (рис. 1) исходному алгоритму не удалось найти путь, в то время как новый алгоритм на основе Вороного успешно выполнил задачу за 3 итерации и 2 минуты.

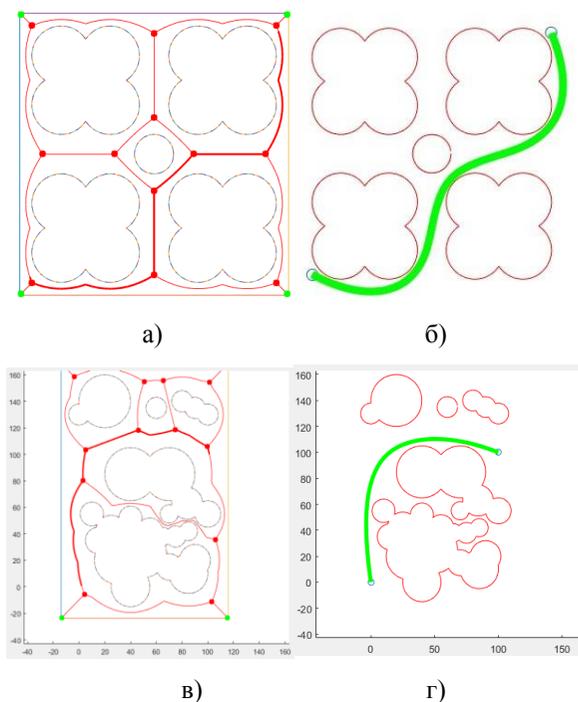


Рис. 4. Результаты планирования пути: решения на графе Вороного (а, в) и результирующие оптимизированные пути (б, г)

Заключение

В этой статье представлен комбинированный метод расчета плавного и безопасного пути для мобильного робота в статической плоской среде. Предложенная модификация оригинального сплайн-алгоритма помогла

избежать проблемы локальных минимумов и добавила больше гибкости для оптимизации пути. Мы усложнили целевую функцию, введя дополнительные критерии, которые максимизировали время пребывания робота в прямой видимости от начальной и целевой точек. Также в новом алгоритме был использован граф Вороного, благодаря чему мы получили успешную основу для дальнейшей оптимизации пути. В результате проверки алгоритма в Matlab было установлено, что он оказался более успешным и быстреедействующим, чем оригинальный метод. Подробная статистика работы алгоритма в настоящее время в процессе сбора и будет представлена в следующей работе.

В рамках нашей будущей работы мы планируем расширить алгоритм для 3D-среды и добавить новые параметры целевой функции, связанные с 3D-поверхностями. Алгоритм будет объединен в пакет ROS с реализацией на C++ и дополнительно проверен в реальной навигации гетерогенной группы роботов, выполняющей поисково-спасательные задачи.

Список литературы

- [Magid и др., 2011] E. Magid, T. Tsubouchi, E. Koyanagi, and T. Yoshida. Building a search tree for a pilot system of a rescue search robot in a discretized random step environment. *Journal of Robotics and Mechatronics*, T. 23(4), 2011, с. 567.
- [Pipe и др., 2014] A. Pipe, F. Dailami, and C. Melhuish. Crucial challenges and groundbreaking opportunities for advanced HRI. *IEEE/SICE International Symposium on System Integration (SII)*, 2014, с. 12–15.
- [Rosenfeld и др., 2015] A. Rosenfeld, N. Agmon, O. Maksimov, A. Azaria, and S. Kraus. Intelligent agent supporting human-multi-robot team collaboration. *Proceedings of the 24th International Conference on Artificial Intelligence*, 2015, с. 1902-1908.
- [Ronzhin и др., 2016] Ronzhin, I. Vatamaniuk, and N. Pavluk. Automatic control of robotic swarm during convex shape generation. *International Conference and Exposition on Electrical and Power Engineering*, 2016, с. 675–680.
- [Panov и др., 2017] A. I. Panov and K. Yakovlev. Behavior and Path Planning for the Coalition of Cognitive Robots in Smart Relocation Tasks, 2017, с. 3–20.
- [Magid, 2006] E. Magid. Sensor-based robot navigation. Master's thesis, Technion - Israel Institute of Technology, 2006.
- [Buyval и др., 2016] A. Buyval, I. Afanasyev, and E. Magid. Comparative analysis of ROS-based monocular SLAM methods for indoor navigation. *9th International Conference on Machine Vision (ICMV)*, Nice, France, 2016.
- [Khatib и др., 2016] O. Khatib and B. Siciliano, *Springer handbook of robotics*, 2016.
- [Magid и др., 2006] E. Magid, D. Keren, E. Rivlin, and I. Yavneh. Spline-based robot navigation. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006. с. 2296-2301.
- [Choset, 2005] H. M. Choset, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

- [Lagarias и др., 1998] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. SIAM Journal on optimization, T. 9(1), 1998, c. 112-147.
- [Dijkstra, 1959] E. W. Dijkstra. A note on two problems in connexion with graphs. Numerische mathematik, vol. 1(1), 1959, c. 269-271.