УДК 519.714.71

ИНТЕРАКТИВНАЯ КАРТА: ПОИСК ОПТИМАЛЬНОГО ПУТИ МЕЖДУ ВЕРШИНАМИ ГРАФА

Устинович А.А., Пригодич Р.В.
Научный руководитель – Юринок В.И., канд. техн. наук, доцент.

В докладе представлена интерактивная карта, полученная в результате нахождения кратчайшего пути с помощью алгоритма Дейкстры. Основная идея заключается в том, чтобы создать массив $\boldsymbol{d}$, в котором для каждой вершины v будем хранить текущую длину кратчайшего пути из $\boldsymbol{s}$ в $\boldsymbol{v}$. Изначально $d = 0$, а для всех остальных вершин расстояние равно бесконечности (или любому числу, которое заведомо больше максимально возможного расстояния).

Во время работы алгоритма будет постепенно обновляться этот массив, находя более эффективные пути к вершинам и уменьшая расстояние до них. Когда узнаем, что найденный путь до какой-то вершины v оптимальный, пометим эту вершину, поставив $(a_v = 1)$ в специальном массиве a, изначально заполненным нулями.

Сам алгоритм состоит из n итераций, на каждой из которых выбирается вершина v с наименьшей величиной d среди еще не помеченных $v = \arg\min d_u$.

Выбранная вершина отмечается в массиве a, после чего из вершины v производятся *релаксации:* просматриваем все исходящие ребра (v, u) и для каждой такой вершины u пытаемся улучшить значение d_u, выполнив присвоение $d_u = \min(d_u, d_v + w)$, где w – длина ребра (v, u).

На этом текущая итерация заканчивается, и алгоритм переходит к следующей: выбирается вершина с наименьшей величиной d, из нее производятся релаксации, и так далее. После n итераций, все вершины графа станут помеченными, и алгоритм завершает свою работу.

Рассмотрим на примере ориентированный граф, представленный на рис. 1. Требуется найти путь из вершины s в вершину t.

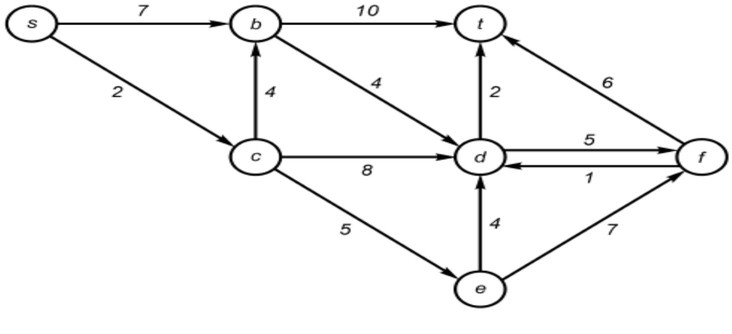

Рис. 1. Ориентированный граф с начальной вершиной s и конечной t.

Результат работы алгоритма показана в таблице 1. Первая строка соответствует шагу 1 алгоритма. Каждая из последующих строк соответствует одному основному циклу, состоящему из шагов 2 и 3. Значения $d(x)$ помеченных вершин (их окончательные значения) выделены полужирным шрифтом.

Номер цикла	s	y	b	c	d	e	f	t
	0	s	∞	∞	∞	∞	∞	∞
1	**0**	c	7	**2**	∞	∞	∞	∞
2	**0**	b	**6**	**2**	∞	7	∞	∞
3	**0**	e	**6**	**2**	10	**7**	∞	16
4	**0**	d	**6**	**2**	**10**	**7**	14	16
5	**0**	t	**6**	**2**	**10**	**7**	14	**12**

Табл. 1. Результат работы алгоритма Дейкстры.

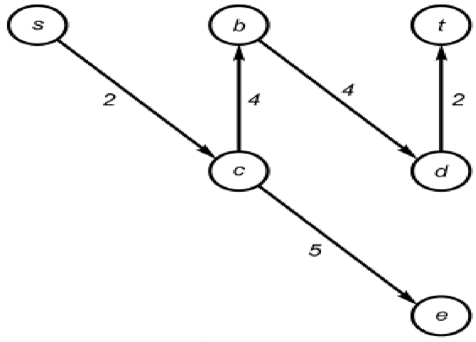

Рис. 2. Минимальный остов исходного графа для пути из вершины s в вершину t.

На рис. 2 представлено дерево T_5 для примера, минимальный путь которого из вершины s в вершину t равен 12. К этому алгоритму была разработана интерактивная карта (приложение) на языке программирования C#, скриншот которой представлен на рис. 3, где изображен ориентированный граф и кратчайший путь из вершины s в вершину t. Кратчайший путь представляет собой выделенные ребра и вершины.

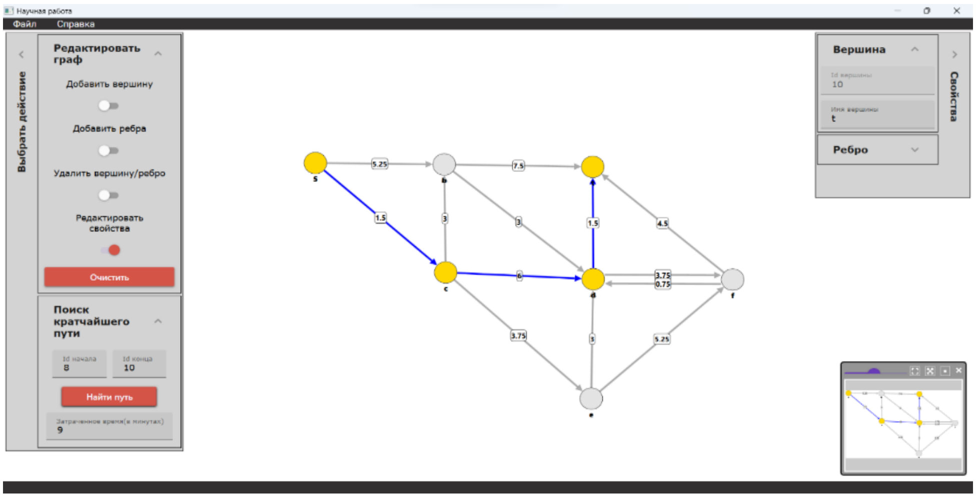

Рис. 3. Результат нахождения кратчайшего пути в программе.

В программе можно указать помимо расстояния дополнительные свойства, которые влияют на вычисления кратчайшего пути. Дополнительными условиями являются: тип дороги, погодные условия, видимость и тип передвижения. Основным же свойством является расстояние (км). Вычисление кратчайшего пути в приложении происходит по времени, которое вычисляется на основе свойств каждого ребра.

Данную научную работу можно использовать для нахождения оптимального пути при проведении экскурсии по городам(местам) стран, наилучшего расположения для пунктов экстремальной помощи (МЧС или скорая помощь), оптимального расположения телевышек, электростанций и т.д.

УДК 681.3.06

НАХОЖДЕНИЕ КРАТЧАЙШЕГО ПУТИ АЛГОРТИМОМ ДЕЙКСТРЫ В МАТЕМАТИЧЕСКОМ ПАКЕТЕ MATHCAD

Лемяшевич В.А., Якименко Д.Д.
Научный руководитель – Юринок В.И., к.т.н., доцент

Mathcad – широко используемый математический программный пакет с обширным набором функций. Однако ему не хватает возможностей работы с графами, которые имеют множество других приложений.

В рамках данной научной работы был реализован алгоритм для решения одной из самых популярных задач в теории графов — нахождение кратчайшего пути между вершинами во взвешенном графе. Значимость данной задачи определяется её различными практическими применениями.

Для реализации нахождения кратчайшего пути можно перебрать все возможные пути в графе, или использовать какой-либо алгоритм. Есть большое количество алгоритмов и преимущество многих из них в том, что они работают гораздо быстрее полного перебора всех возможных путей графа, благодаря чему расчет кратчайшего пути происходит быстрее (особенно актуально для больших графов).

Предположим, что у нас есть улицы, перекрёстки и мы знаем время пути между ними. Нарисуем это в виде взвешенного графа (рис. 1). Чтобы найти самый быстрый путь из A в H, мы смотрим, какие ребра инцидентны вершине A. Видно, что ребро (A, E) имеет меньший вес, чем все остальные, поэтому выберем путь через это ребро.