

отстают в создании картинок высокого разрешения. Они не умеют творить – стилизуют, но не рисуют. По сути, ИНС используют плагиат, когда одна часть данных «впихивается» в другую программу.

В данной работе были рассмотрены и проанализированы результаты обработки нескольких наиболее широко известных и бесплатных нейросетей. Анализ осуществлялся на примерах цифровых растровых черно-белых изображений, имеющих отношение именно к технологическому университету. Хотя, конечно, выбор фотографий при обработке нейросетями практически не имеет никакого значения.

Первая рассмотренная ИНС была «Colourise». Главная положительная особенность данной сети Colourise – это то, что она умеет отличать фон от колоризируемого портрета, лучше справляется с тонированием, может раскрасить даже мелкие объекты на фото. Порой эта сеть даже справляется с достижением идеального цвета.

Еще одна рассмотренная ИНС – Colorize. Colorize умеет быстро раскрашивать фотографии (буквально за 10–30 секунд), но имеет ряд отрицательных свойств: низкое разрешение изображения на выходе, многочисленные цветные пятна, участки, избежавшие окрашивания.

Следующая рассмотренная ИНС – MyHeritage. MyHeritage: размер фото самый большой в сравнении с другими нейросетями, имеет водяной знак, тон выглядит ближе к колоризированным вручную снимкам. Описанные особенности являются существенными минусами в принятии решения об использовании данной ИНС.

В докладе, который был сделан на конференции для обработки мы использовали фотографии с сайта официального сайта Витебского государственного технологического университета:

<https://vstu.by/universitet/muzej/history-pages/>

При этом пользовались такими нейросетями, как:

1. <https://colorize.cc/>
2. <https://colourise.sg/>
3. <https://www.myheritage.com/photo-enhancer/>

Список использованных источников

1. ПВТ. Цифры и факты [Электронный ресурс]. – Режим доступа: <https://www.park.by/http/facts/>. – Дата доступа: 07.04.2022.
2. Нейросети [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Нейросети>. – Дата доступа: 25.04.2022.
3. Colorizing B&W Photos with Neural Networks [Электронный ресурс]. – Режим доступа: <https://blog.floydhub.com/colorizing-b-w-photos-with-neural-networks/>. – Дата доступа: 25.04.2022.
4. Раскрашиваем чёрно-белую фотографию с помощью нейросети из 100 строк кода [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/nix/blog/342388/>. – Дата доступа: 25.04.2022.
5. Роберт, Каллан. Основные концепции нейронных сетей. – Вильямс, 2001.

УДК 519.1

ПРИМЕНЕНИЕ АЛГОРИТМОВ ПОИСКА В ШИРИНУ И ДЕЙКСТРЫ ДЛЯ ОПРЕДЕЛЕНИЯ КРАТЧАЙШЕГО ПУТИ В ГРАФАХ

Коваленко А.В., ст. преп., Антонова Т.А., студ., Комиссарова Д.К., студ.

*Витебский государственный технологический университет,
г. Витебск, Республика Беларусь*

Реферат. В статье рассматриваются два типа алгоритмов и их применение для определения кратчайшего пути в графах. Построена математическая модель расчёта наименьшего пути связанного графа. С использованием математической модели разработан программный продукт, который позволяет определить кратчайший путь между вершинами графа.

Ключевые слова: граф, кратчайший путь, генерация графов, связанность графов, алгоритм поиска в ширину, алгоритм Дейкстры.

В логистике всегда возникает вопрос о развитии и оптимизации транспортных путей. Прокладывание путей в соответствии с критерием оптимальности всегда был трудной задачей. Необходимо учитывать длительность путей, их стоимость и другие характеристики. В этом случае удобно применять такую математическую модель, как определение кратчайшего пути в теории графов. В статье даётся описание таких схем как алгоритм поиска в ширину и алгоритм Дейкстры для определения минимального пути от одной вершины графа к другой. Также приведена наглядная иллюстрация отлаженных компьютерных программ для решения поставленной задачи.

Графом называется упорядоченная пара $G = \langle V, R \rangle$, где $V = \{v_1, v_2, \dots, v_n\}$ – множество элементов, которые называются вершинами, а $R = \{r_1, r_2, \dots, r_m\}$ – множество рёбер или дуг графа. Каждое ребро графа определяется парой вершин v_i и v_j , которое их соединяет. Если указывается направление движения на рёбрах, то граф называется ориентируемым, если направление не указано, то граф является неориентируемым. Графы используются как математические модели во многих прикладных задачах, а именно в статье будем использовать теорию графов для определения кратчайшего пути в транспортных задачах. Существуют различные способы задания графов: в теоретических исследованиях, при представлении графа в информационных технологиях, логистике. В частности граф может быть представлен матрицей смежности (булева матрица в которой строкам и столбцам поставлены в соответствие вершины графа) и матрицей инцидентности (булева матрица, в которой строкам поставлены в соответствие вершины графа, а столбцам – рёбра). Для генерации графов необходимо его построить, то есть сформировать матрицы графов и записать сформированный граф в файл в виде списка рёбер. Построенный граф будет являться связанным, если между любыми его вершинами существует некоторый путь.

Для определения связности графа будем использовать алгоритм поиска в ширину. Граф может быть как ориентированным, так и неориентированным, для использования алгоритма это не имеет значения. Если граф связанный, то в процессе применения алгоритма будет задействована каждая вершина графа, то есть строим некоторое дерево D .

На вход алгоритма подаём невзвешенный граф $G = \langle V, R \rangle$. Поиск в ширину будем начинать с любой первоначальной вершины, например v_0 , которая в данном случае становится основной вершиной. Затем для произвольной основной вершины v_i , находим такую смежную вершину v_j , которая не является отмеченной. Ребро $\langle v_i, v_j \rangle$ присоединяется к строящемуся дереву D , а вершина v_j в этом случае становится основной вершиной. Если для основной вершины не находится неотмеченная смежная вершина, то возвращаемся к вершине, которая была основной до неё. Окончание работы алгоритма определяется условием, которая требует выполнить возврат из первоначальной вершины v_0 . Если построенное в результате применения алгоритма поиска в ширину дерево D будет содержать все вершины графа, то граф $G = \langle V, R \rangle$ является связанным.

Алгоритм поиска в ширину для определения кратчайшего пути в графе реализован на языке программирования JavaScript.

Выбираем число вершин графа, например, зададим семь вершин. На рисунке 1 приведён интерфейс программы определения минимального пути между вершинами графа, с использованием алгоритма поиска в ширину.

The interface consists of the following elements:

- Input fields: "Начальная точка по x" and "у" (Start point by x and y), "Конечная точка по x" and "у" (End point by x and y), and "Результат =" (Result =).
- A 7x7 grid with columns labeled 0 through 6 and rows labeled 0 through 6.
- A button labeled "Минимальный путь" (Minimal path) located at the bottom right of the grid.

Рисунок 1 – Интерфейс программы минимальный путь

Заполняем матрицу смежности графа, используя принцип 0–1. Если имеется ребро между вершинами графа, то соответствующий элемент матрицы равен единице, если такого ребра нет, то элемент матрицы равен нулю. Указываем начальную и конечную вершину и нажимаем кнопку «минимальный путь». Например, для начальной вершины $A(0,0)$, конечной вершины $B(4,4)$ и элемента матрицы смежности $a_{55} = 1$, получаем значение наикратчайшего пути, который равен 8 (рис. 2).

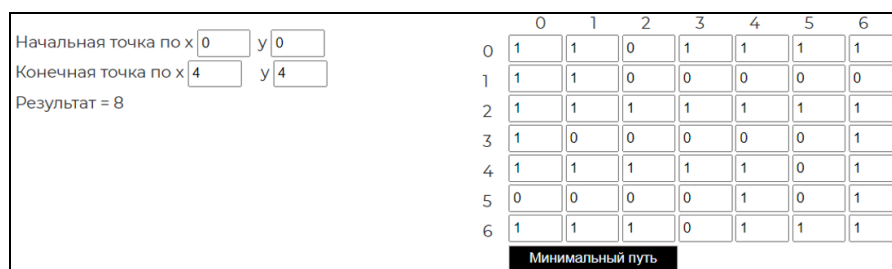


Рисунок 2 – Минимальный путь с элементом $a_{55} = 1$

Для начальной вершины $A(0,0)$, конечной вершины $B(4,4)$ и элемента матрицы смежности $a_{55} = 0$, получаем значение наикратчайшего пути, который равен 16 (рис. 3).

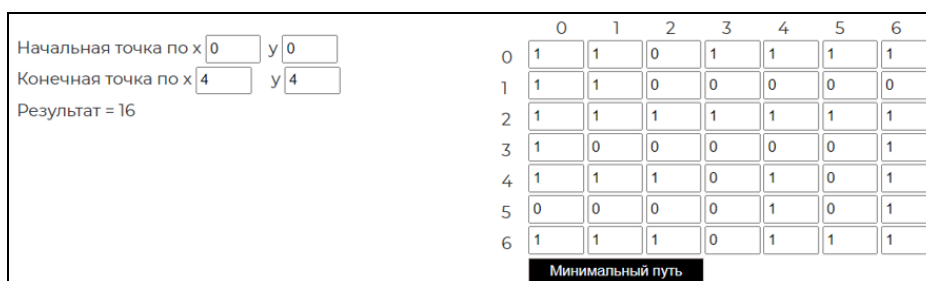


Рисунок 3 – Минимальный путь с элементом $a_{55} = 0$

Таким образом, разработанный программный продукт даёт возможность определить минимальный путь между вершинами графа, с использованием алгоритма поиска в ширину. При этом мы не учитываем вес рёбер графа.

Для определения кратчайшего пути между вершинами графа с учётом веса дуг воспользуемся алгоритмом Дейкстры, который является наиболее эффективным для поставленной задачи.

Пусть задан ориентированный граф $G = \langle V, R \rangle$, каждой дуге $\langle v_i, v_j \rangle$ которого поставлено в соответствие неотрицательное число $t(v_i, v_j)$ и представляющее собой вес дуги или его длину. Каждой вершине v графа поставим в соответствие число $q(v)$, равное числу инцидентных дуг для этой вершины. Рассмотрим две произвольные вершины $v_0, u_0 \in V$. Длина пути между этими вершинами представляет собой сумму длин отдельных дуг, составляющий путь между этими вершинами. В алгоритме Дейкстры вершина u_0 принимается за начальную вершину искомого пути, которая определяется как основная. На каждом шаге алгоритма исследуются все не рассмотренные вершины. Определяется ближайшая к вершине u_0 вершина x и соответствующий минимальный путь $q(u_0, x)$, после чего вершина становится основной.

Для применения алгоритма Дейкстры необходимо выполнить следующие действия:

- 1) положим $q(u_0) := 0$, $y := u_0$, $q(x) := \infty$ и отметим вершину u_0 как основную;
- 2) для каждой вершины x вычисляем новое значение $q(x) = \min\{q(x), q(y) + t(y, x)\}$;
- 3) выбираем неотмеченную вершину x , для которой величина $q(x)$ является

минимальной. Если для всех неотмеченных вершин $q(x) = \infty$, то вычисления прекращаются, так как в графе отсутствует путь от вершины u_0 до вершины v_0 ;

4) отмечаем дугу $\langle y, x \rangle$ как исследованную и положим $y = x$;

5) если $y = v_0$, то заканчиваем вычисления. Двигаемся из вершины v_0 против направления исследованных групп, восстанавливая путь из вершины u_0 в вершину v_0 , который будет являться единственным кратчайшим путём в графе. Если $y \neq v_0$, то переходим ко второму пункту.

Алгоритм Дейкстры для определения кратчайшего пути в графе реализован на языке программирования JavaScript.

Выбираем число вершин графа, например, зададим три вершины. Заполняем матрицу смежности. На рисунке 4 приведён результат работы программы определения минимального пути между начальной и конечной вершинами графа, с использованием алгоритма Дейкстры.



Рисунок 4 – Кратчайший путь по алгоритму Дейкстры

Таким образом, в статье построены математические модели алгоритма в ширину и Дейкстры для определения кратчайшего пути в графе. Для их реализации были написаны программы на языке JavaScript, которые рассчитывают минимальный путь между вершинами графа, заданного своей матрицей.

Список использованных источников

1. Волкова, В. Н. Теория систем и системный анализ / В. Н. Волкова, А. А. Денисов. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2019. – 462 с.

УДК 514

РОЛЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ПРИ ПРЕПОДАВАНИИ ПРЕДМЕТА НАЧЕРТАТЕЛЬНОЙ ГЕОМЕТРИИ

Абдурахимова Ф.А., к.т.н., доц., Ахмедбекова А.В., асс.

*Ташкентский институт текстильной и легкой промышленности,
г. Ташкент, Республика Узбекистан*

Реферат. *Статья посвящена роли информационных технологий при преподавании предмета начертательной геометрии. Современное инженерное образование невозможно представить без практического использования в учебном процессе новейших достижений научно-технического прогресса. Обучение с использованием современных информационных технологий, по общепринятому мнению, является наиболее интенсивной формой обучения. Компьютерные технологии позволяют наиболее гибко реагировать на растущие потребности современного студента.*

Ключевые слова: начертательная геометрия, инженерная дисциплина, компьютерная технология, пространственное мышление, методика, информационная технология, интернет-ресурс, интерактивная доска, объект, трёхмерное измерение.

Начертательная геометрия является одной из фундаментальных наук, составляющих основу инженерно-технического образования. Начертательная геометрия изучает методы