

ИСПОЛЬЗОВАНИЕ ПРИКЛАДНОГО ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ПОИСКА К КРАТЧАЙШИХ ПУТЕЙ В ГОРОДЕ

А. Л. Абрамов¹, П. А. Пугач², В. С. Тынченко³

¹Дальневосточный федеральный университет, Владивосток, Россия

²Российская таможенная академия, Владивостокский филиал, Россия

³Московский государственный технический университет им. Н.Э. Баумана, Россия

E-mail: abramov.al@dvfu.ru, 679097@mail.ru, vadimond@mail.ru

В работе рассмотрена программная реализация модуля в программно-техническом комплексе прикладного искусственного интеллекта BAUM AI PLATFORM, позволяющего построить первичную графовую модель города с помощью библиотеки языка python с открытым исходным кодом OSMnx, и решить задачу поиска K кратчайших путей между заданными начальной и конечной вершинами (адресами в городе) с помощью алгоритма двойного поиска.

USING APPLIED ARTIFICIAL INTELLIGENCE TO SEARCH K SHORTEST PATHS IN A CITY

A. L. Abramov, P. A. Pugach, V. S. Tynchenko

The paper discusses the software implementation of the module in the software and hardware complex of applied artificial intelligence BAUM AI PLATFORM, which allows you to build a primary graph model of the city using the python language library with open source code OSMnx, and solve the problem of finding K - shortest paths between given initial and final vertices (addresses in the city) using a double search algorithm.

Любой город можно представить в виде сети, в которой решаются различные вопросы: размещение социальной, экономической, транспортной, производственной и энергетической инфраструктуры; расположение служб первой и другой необходимости; оценки стоимости земли и недвижимости [1].

Следовательно, пространство регионов и страны также можно смоделировать как «сеть сетей», в которой каждый отдельный город моделируется собственной сетью. Математические модели подобных пространственных структур рассматриваются в теории сложных сетей. В отношении городов модель сложной сети образует определенную иерархию, на нижнем уровне которой располагается первичный граф, отображающий уличные сети [2].

Одна из основных целей программы «Умный город» – комплексное повышение эффективности городской инфраструктуры [3]. Для её реализации, нужен доступный инструмент позволяющий получать данные в режиме реального времени о ситуации в городе, сетях, строить прогнозы, содержащие все этапы конвейера обработки данных. Одним из вариантов решения данной проблемы является использование no-code технологий, позволяющей аналитикам работать без прямого кодирования, создавая пайплайны анализа данных в реальном времени используя блок-схемы в нотации BPMN 2.0.

Для реализации данного подхода используется программно-технический комплекс BAUM AI PLATFORM, предназначенный для обработки данных любого типа, как структурированных, так и неструктурированных, который позволяет строить модели для решения задач предиктивной аналитики в различных сферах деятельности, платформа позволяет выполнять полный цикл data science. Модуль BAUM AI – это конструктор, на основе которого можно создавать модели искусственного интеллекта или использовать уже предобученные модели без прямого кодирования [4].

Постановка задачи. К настоящему моменту реализовано построение первичной графовой модели города (ПГМГ) и алгоритм поиска k кратчайших путей (КП) между любой парой адресов в городе в соответствии с внутренними требованиями платформы BAUM AI.

Первичная графовая модель города строилась с помощью библиотеки языка python с открытым исходным кодом OSMnx [1].

Задача построения k кратчайших путей заключается в следующем: дан связный ориентированный взвешенный граф G . Задача о поиске наикратчайшего пути состоит в том, чтобы найти подграф G_{st} , включающий наикратчайшие пути от заданной вершины V_s до вершины V_t , при этом сумма весов ребер должна быть минимальна. Данная постановка задачи формально описывается следующим образом:

$$\varphi : G = (V, A) \rightarrow \left[G_{st} = (V_{st}, A_{st}) \mid V_s, V_t \in V, \sum_{\substack{l \in V_{st} \\ l \in \{1, 2, \dots, m\}}} c_l^{st} \rightarrow \min \right]$$

Во многих случаях требуется знать не только кратчайшие пути между двумя заданными вершинами, но и вторые, третьи и т.д. по длине пути между этими же вершинами или всеми парами вершин. Такая ситуация возникает, например, когда необходимо объехать пробку или перевезти груз по тем транспортным путям, для которых провоз груза разрешен в связи с весовыми ограничениями по проезду мостов или высоте тунелей и т.д.

В этом случае задача поиска кратчайшего пути становится задачей о поиске K – КП между заданной вершиной и каждой из вершин графа, т.е. состоит в том, чтобы найти упорядоченное множество подграфов $\{G_s\}$, каждый из которых является подграфом, содержащим КП $1, 2 \dots K$ длины из вершины V_s в каждую из вершин графа G .

В данной статье описывается решение данной задачи с помощью алгоритм двойного поиска, который находит k кратчайших путей из некоторой фиксированной вершины ко всем остальным вершинам исходного графа.

Решение задачи с помощью программно-технического комплекса BAUM AI. Для решения задачи проведена разработка следующих функции (блоков пайплайна): «Загрузка графа», «Вычисление k кратчайших путей (КП) в графе», «Визуализация k кратчайших путей». Кроме того, для каждой функции написан соответствующий визуализирующий компонент – GUI, реализованный

в двух частях: GUI_dict – присутствует в заключительной части тела только тех функций, которые предоставляют пользователю в результате работы функции визуальный материал в виде таблиц, рисунков или графиков; GUI.py – отдельный блок кода, который обеспечивает корректное отображение функции в виде блока на платформе. Этот блок оформляется для каждой функции и в дальнейшем при имплементации функции на платформу объединяется с общим GUI.py платформы путем добавления строк кода в общий код, который уже функционирует на платформе.

Функция «Загрузка графа» (блок «Источник данных»). Визуальное представление пользователю на платформе VAUM AI функции «Загрузка графа» (блок «Источник данных») проиллюстрировано на рис. 1.

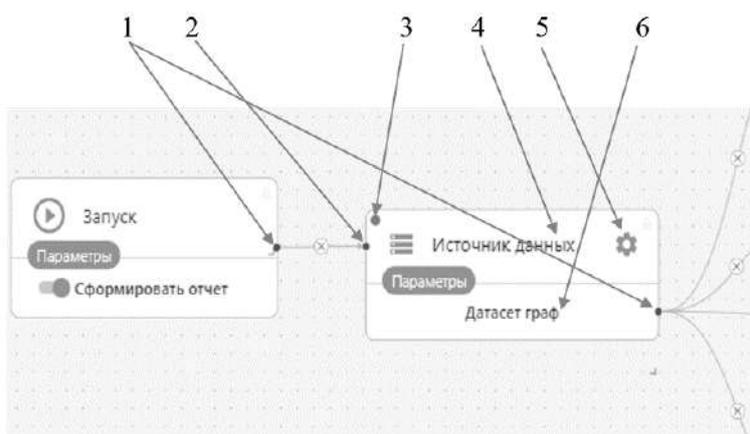


Рис. 1. Отображение пользователю функции «Загрузка графа» (блок «Источник данных») в пайплайне на платформе VAUM AI.

Обозначения: 1 – коннекторы для дальнейшего использования данных, полученных в результате работы функции; 2 – коннекторы для передачи входных параметров; 3 – индикатор корректного запуска блока (зелёный цвет – код корректен, красный цвет – код содержит ошибку); 4 – наименование блока; 5 – шестерёнка ввода пользовательских параметров для работы функции; 6 – Название данных, полученных в результате работы функции.

Блок «Запуск» обозначает начало блок-схемы и всегда является её первым и обязательным элементом, блок «Источник данных» – это второй и обязательный элемент любой блок схемы, именно он определяет какие данные будут использоваться при построении сценария.

Данная функция «Загрузка графа» предназначена для выбора файла для загрузки графа города в виде файла формата *.graphml и дальнейшего его преобразования в переменную graph_out с типом данных networkx.MultiDiGraph. В формате.graphml данные по каждой вершине графа города должны содержаться в следующем виде: {"type": "node", "id": 261690280, "lat": 43.760792, "lon": 131.9901532}, где "type": "node" означает тип объекта – вершина, "id": 261690280 – идентификационный номер вершины в формате данных int, "lat": 43.760792 – координата широты вершины в формате float, "lon": 131.9901532 – координата

долготы вершины в формате float.

На данном этапе пользователь платформы выбирает файл графа города, анализ которого в дальнейшем хочет провести с помощью пайплайна на платформе VAUM AI.

Функция «Вычисление k кратчайших путей в графе» реализована в виде блока процесса «Кратчайшие пути» представленный на рис. 2. Функция предлагает пользователю возможность определить k кратчайших путей между парой заданных пользователем вершин в графе города.

Блок «Процесс» предназначен для выполнения операций над данными. Блок-схема пайплайна может содержать несколько элементов «Процесс», настроенных пользователем на выполнение различного функционала. Для запуска блока «Кратчайшие пути» кроме внутренних данных (переменная `graph_out` с типом данных `networkx.MultiDiGraph`), передаваемых из предыдущего блока «Загрузка графа», пользователь должен ввести следующий набор параметров, нажав на визуальный компонент блока «Шестеренка»: `graph_k_short_path` - количество маршрутов; `origin_node` – точка отправления; `destination_node` - точка прибытия.

В результате работы функции в систему возвращается переменная `routes_df` с типом данных `pd.DataFrame`, которая включает в себя вычисленные маршруты (каждый столбец датафрейма – это k -ый путь).

Используя функции на базе платформы VAUMAI, была создана рабочая область и создан дашборд для решения задачи поиска k кратчайших путей между двумя объектами города с использованием ПГМГ, что может являться стартовой точкой для решения более сложных графовых задач. Пайплайн для создания и валидации модели полностью представлен на рис. 2.

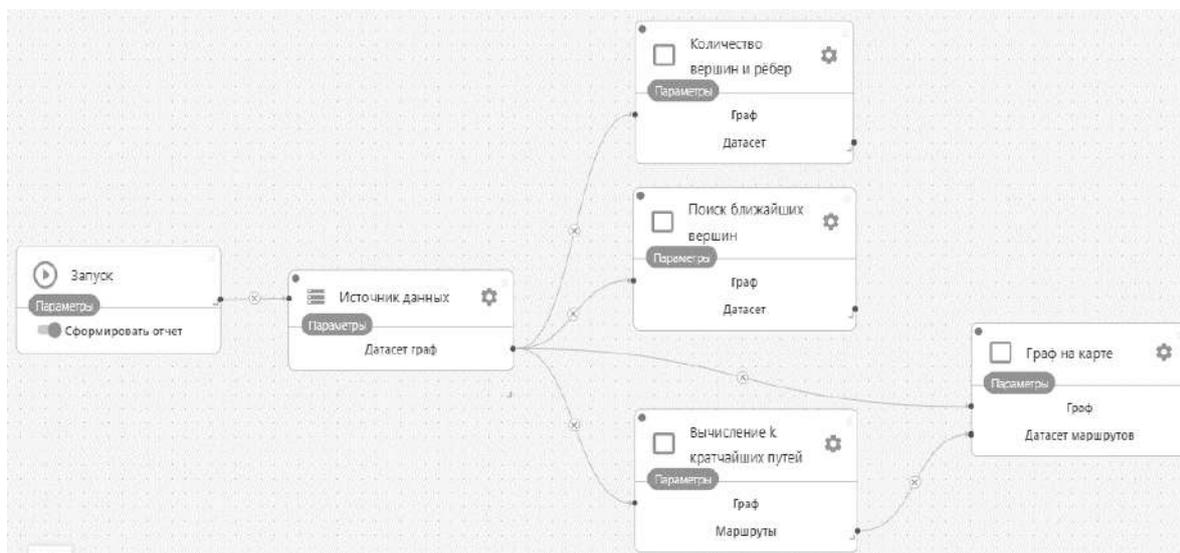


Рис. 2. Пайплайн для вычисления k кратчайших путей с помощью ПГМГ и их визуализации на платформе VAUM AI

Функция «Визуализация k кратчайших путей» также реализованная в виде

блока «Процесс», предоставляет пользователю возможность визуализировать результат работы предыдущего блока, выводя найденные k кратчайших путей, т.е. возвращается изображение «Кратчайшие пути в графе», включающее в себя граф города и k кратчайших путей между парой заданных вершин в графе города, представленных на рис. 3.



Рис. 3. Блок процесса «Кратчайшие пути», визуализация k кратчайших путей» в пайплоне на платформе VAUM AI на примере города Уссурийск (Приморский край)

Заключение. Будущее современной аналитики, основанной на технологиях data science и data mining, это системы позволяющие пользователям строить свои процессы обработки данных с помощью технологий no-code (совсем без кода) и low-code (с малой долей кода), потому что они «демократизируют» аналитику, эксперты и учёные не владеющие навыками программирования, получают доступ к инструментам и методам, необходимым для принятия решений на основе данных.

Технология обучения программного аппаратного комплекса VAUM AI позволила для разработанной первичной графовой модели города Уссурийска (Приморский край) создать алгоритм нахождения k кратчайших путей, основанный на идеи двойного поиска, и решить задачу поиска произвольного количества кратчайших путей между заданной начальной и конечной точкой на территории города.

Работа выполнена при финансовой поддержке Министерства науки и образования Российской Федерации и достижения результатов федерального проекта «Передовые инженерные школы» (№ 075-15-2022-1143 от 07.07.2022) в рамках соглашения, заключенного ПИШИББиПС ДВФУ и МГТУ им. Н.Э. Баумана.

СПИСОК ЛИТЕРАТУРЫ

1. *Абрамов А. Л., Пугач П. А., Графовые модели городов // Математическое и компьютерное моделирование в экономике, страховании и управлении рисками. 2021. № 6. С. 3-7.*
2. *Liu X., Derudder B., Witlox F., Hoyler M. Cities As Networks within Networks of Cities: The Evolution of the City. Firm-Duality in the World City Network, 2000-2010 // Journal of Economic and Human Geography. 2014. Vol.105. Is. 4. P. 465-482.*
3. *Проект Цифровизации городского хозяйства «Умный город». [Электронный ресурс].*

URL: <https://minstroyrf.gov.ru/trades/gorodskaya-sreda/proekt-tsifrovizatsii-gorodskogo-khozyaystva-umnoy-gorod/> (дата обращения: 10.09.2023).

4. БАУМ АІ. [Электронный ресурс]. URL: <https://bauminform.ru/ai/platform> (дата обращения: 10.09.2023).