**Техническое задание**

на выпускную квалификационную работу

студенту группы Б08–181–1

1. **Тема**: Разработка веб–приложения для решения логистических задач.
2. **Цель работы**: Разработка интерактивного информационного инструментария для решения задач логистики топливообеспечения.

**Объект:** распределенная система топливообеспечения.

**Предмет:** средства разработки веб–приложений.

1. **Программно–аппаратные средства:** Программное обеспечение: MS Windows 10, MS Office 2016, PyCharm.
2. **Состав задач:**

4.1 Обзор средств разработки веб–приложения.

4.2 Создание базы данных элементов распределенной системы топливообеспечения.

4.3 Создание интерактивной карты региона.

4.4 Реализация математических моделей логистики системы топливообеспечения на основе древесного сырья.

4.5 Решение задачи оптимизации производства и поставок топлива из древесного сырья на территории Удмуртской Республики при различных критериях оптимизации.

1. **Критерии окончания работы:** Веб–приложение для решения логистических задач. Отчет о результатах исследования.

Утверждаю:

зав. каф. МОИС,

д.т.н., профессор И.Г. Русяк

Руководитель:

к.т.н., доцент Д.Г. Нефедов

Исполнитель:

студент группы Б08–181–1 А.Г. Асабина

УДК 004.94

РЕФЕРАТ

Объем записки: 53 стр., 21 рис., 3 табл., 30 библ. наим.

Ключевые слова: кластерный анализ, решение задач логистики, задача кластеризации, задача маршрутизации, логистика, топливоснабжение, маршрутизация, кластеризация, оптимизация, веб–приложение.

Работа посвящена разработке веб–приложения для решения логистических задач на примере оптимизации доставки топлива из древесного сырья в системе топливообеспечения Удмуртской республики.

В работе изложена теоретическая основа построения модели распределенной системы топливообеспечения. Рассмотрены модели и алгоритмы решения задач логистики, проведен обзор средств и технологий разработки веб–приложений для реализации алгоритмов решения поставленных задач и визуализации полученных результатов на карте.

Разработано веб–приложение, позволяющее решать логистические задачи маршрутизации транспортных средств и кластеризации на примере территории Удмуртской республики.

**ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ**

Объектом исследования является распределенная система топливообеспечения.

Актуальность темы обусловлена отсутствием свободно распространяемого решения по оптимизации маршрутов доставки грузов с доступом через сеть Интернет.

Цель работы. Разработка веб–приложения для решения задач логистики (на примере топливообеспечения Удмуртской Республики).

Методика работы основана на использовании методов математического моделирования, исследования операций, прикладной логистики и теории оптимального управления и оптимизации, использованы средства разработки программного обеспечения и анализа данных.

На защиту выносятся

1. Методы решения задачи оптимального размещения элементов распределенной производственной структуры;
2. Программно–вычислительный комплекс, реализующий решение задачи маршрутизации транспортных средств;

Научная новизна. Предложенный метод решения задач логистики позволяет показать и проанализировать оптимальные маршруты доставки груза в условиях ограниченной технической оснащенности.

Практическая ценность. Разработанный программный комплекс может быть использован для решения задач оптимального управления экономической системой на региональном уровне. Сформулированная экономико–математическая модель может быть применена для прогнозирования динамики факторов экономического развития.

Публикации. По результатам работы написан отчет.

Структура и объем работы. Работа содержит введение, четыре главы и заключение, изложенные на 53 страницах. В работу включены 21 рисунок, 3 таблицы и список литературы из 30 наименований.

Оглавление

[ВВЕДЕНИЕ 6](#_Toc43822638)

[1 АНАЛИЗ ОБЪЕКТА ИССЛЕДОВАНИЯ 8](#_Toc43822639)

[2 МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ЛОГИСТИКИ ТОПЛИВОСНАБЖЕНИЯ РЕГИОНА 15](#_Toc43822640)

[2.1 Содержательная и концептуальная постановка задачи логистики топливоснабжения 15](#_Toc43822641)

[3. АЛГОРИТМЫ И МЕТОДИКА РЕШЕНИЯ ЗАДАЧИ ЛОГИСТИКИ 23](#_Toc43822642)

[3.1 Алгоритм и методика решения задачи маршрутизации 23](#_Toc43822643)

[3.2. Алгоритм и методика решения задачи кластеризации 24](#_Toc43822644)

[4. РАЗРАБОТКА ВЕБ–ПРИЛОЖЕНИЯ ДЛЯ РЕШЕНИЯ ЛОГИСТИЧЕСКИХ ЗАДАЧ 29](#_Toc43822645)

[4.1 Обзор технологий и средств разработки веб–приложений 29](#_Toc43822646)

[4.2 Разработка интерактивной карты региона 33](#_Toc43822647)

[ЗАКЛЮЧЕНИЕ 44](#_Toc43822648)

[Литература. 45](#_Toc43822649)

[Приложение 1. Листинг программы для визуализации оптимальных маршрутов 48](#_Toc43822650)

ОСНОВНЫЕ ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

УР – Удмуртская Республика;

тут/год – тонн условного топлива в год;

– среднеквадратичное отклонение;

– число кластеров;

– число элементов в *i*–м кластере, ;

– элементы выборки, входящие в *i*–й кластер;

– элемент, являющийся центром *i*–го кластера;

ППТ – пункт подготовки топлива;

HTML – (англ. Hyper Text Markup Language) язык гипертекстовой разметки;

CSS – (англ. Cascading Style Sheets) каскадные таблицы стилей;

HTTP – (англ. HyperText Transfer Protocol) протокол передачи гипертекста;

SQL – (англ. structured query language) язык структурированных запросов;

API – (англ. application programming interface) программный интерфейс приложения, интерфейс прикладного программирования;

xlsx – (англ. Microsoft Excel Open XML Spreadsheet) формат электронных таблиц, создаваемый с помощью Microsoft Office Excel;

csv – (англ Comma–Separated Values) значения, разделённые запятыми – текстовый формат, предназначенный для представления табличных данных;

# ВВЕДЕНИЕ

Логистические системы – это системы больших размеров, которые географически рассредоточены в пространстве. Их сложность заключается во множестве факторов: взаимодействие между пунктами доставки и отправки, процессы транспортировки и складирования,

Логистика – это управление цепочками поставок, планирование и контроль потока материальных, трудовых, энергетических и информационных потоков между пунктами производства и потребителями.

Управление и контроль современных логистических систем основаны на методах теории управления, исследования операций и искусственного интеллекта. Осуществление определенных действий по контролю возможно благодаря разнообразию классических и современных электронных, коммуникационных и информационных технологий, которые являются жизненно важными частями логистической инфраструктуры. Эти технологии вносят существенный вклад в эффективное распределение, сокращают время в пути и загруженность транспорта, снижают производственные и транспортные расходы и повышают уровень обслуживания.

Так как во всех сферах деятельности человека в обществе на данном этапе развития основным свойством является внедрение информационных технологий и, в общем, информатизация, было решено разработать веб–приложение для решения задач логистики.

В работе рассматриваются некоторые методы и алгоритмы решения задач логистики, производится краткий обзор средств и технологий разработки веб–приложений.

В ходе работы ставилась задача разработать средствами языка программирования Python приложение для визуализации пользовательских данных об оптимальных маршрутах в кластерах.

Для поиска лучшего маршрута на каждом пути, необходимо знать расстояния от каждой точки до каждой. Следовательно, в качестве исходных данных необходима матрица расстояний между ними, которая определяется по географическому расположению теплоисточников. Для этого была использована библиотека openrouteservice–py версии 0.4 [6]; для решения промежуточных задач использованы библиотеки: Pandas версии 1.0.4 – для работы с данными [27]; SciPy версии 1.4.1 – для решения задачи кластеризации [25]; для визуализации данных на подложке карты была использована библиотека Folium 0.11.0 [5].

# 1 АНАЛИЗ ОБЪЕКТА ИССЛЕДОВАНИЯ

Математическое описание проблемы реального мира называются математическими моделями. Математические модели используются для прогнозирования поведения системы в будущем, планирования будущего развития системы и определения различных стратегий и действий управления.

Определение цели моделирования и постановка задачи представляет собой первый шаг в схеме решения проблемы.

На следующем этапе словесное описание проблемы должно быть заменено соответствующей математической формулировкой. Чтобы построить математическую модель, устанавливаются различные логические и математические отношения между конкретными переменными; определяется целевая функция, а также набор ограничений, которые должны быть выполнены. Очень трудно, а в большинстве случаев и невозможно, создать математическую модель, которая охватывала бы все различные аспекты рассматриваемой проблемы. Таким образом, математические модели представляют упрощенное описание реальной проблемы. Практически все математические модели представляют собой компромисс между желанием точно описать реальную проблему и возможностью решить математическую модель.

Методы математического программирования используются для решения проблем, связанных с наиболее эффективным распределением ограниченных ресурсов (товаров, капитала, рабочей силы и т.д.) для достижения поставленных целей. Типичные проблемы включают увеличение доли рынка, планирование производства, планирование и составление списков персонала, маршрутизацию и планирование транспортных средств, размещение объектов в сети, планирование развития парка и т.д. Их решения можно найти с помощью одного из методов математического программирования [31].

Задача маршрутизации является очень важной и сложной задачей комбинаторной оптимизации, которая находит применение во многих логистических системах. Часто виды задач маршрутизации делят на задачи, связанные с количеством и вместимостью транспортных средств, числом пунктов отправки и доставки, но фактически все задачи маршрутизации сводятся к хорошо известной задаче коммивояжера, которую можно рассматривать как задачу маршрутизации с одним транспортным средством достаточной грузоподъёмности для обслуживания всех покупателей.

В случае, когда необходимо обслуживать большое количество узлов, используется алгоритм заметающей прямой (Sweep algorithm) в рамках подхода «кластеризация–маршрутизация». В этом случае, учитывая направление, например, по часовой стрелке, следует проверить соотношение совокупных потребностей и вместимости транспортного средства (включая все другие ограничения). Узел, который не может быть включен из-за нарушения вместимости транспортного средства или других ограничений, становится первым узлом в другом кластере. Таким образом, весь регион делится на кластеры (зоны). На следующем этапе транспортная задача решается внутри каждого кластера отдельно. Кластеризация завершается, тогда, когда все узлы принадлежат кластерам. Таким образом, транспортная задача превращается в несколько задач коммивояжера.

В соответствии с целями концепции развития лесного фонда Удмуртской Республики 2019–2030 гг. важным значением является повышение вклада лесной отрасли в социально–экономическое развитие республики, это можно достичь посредством совершенствования и развития технологий переработки, вследствие чего возможно увеличение инвестиционной привлекательности отрасли.

Каменноугольные месторождения в регионе почти не используются, газ поступает в основном из Западной Сибири, а топливо из местных источников – дрова и торф – занимает всего 2,5% в топливном балансе Удмуртской Республики.

Важность переоценки роли местных возобновляемых источников энергии заключена в государственной программе РФ «Развитие промышленности и повышение ее конкурентоспособности» и утверждена постановлением Правительства Российской Федерации от 15 апреля 2014 года №328 [12].

Так как, в Республике развиты лесозаготовительная и деревообрабатывающая промышленность, имеет место разработка концепции по созданию распределенной системы топливообеспечения из возобновляемых источников энергии. [19]

Лесистость территории Удмуртской Республики на 01.01.2019 года составляет 46,1 %, причем её распределение по районам очень неравномерно. Самая низкая степень облесенности в Каракулинском (6,9 %), Алнашском (16,3 %), Киясовском (23,1 %), Сарапульском (25,4 %), Юкаменском (26,4 %) районах. Высокая степень облесённости в Сюмсинском (72,3 %), Якшур–Бодьинском (65 %), Игринском (63,5 %), Красногорском (63,8 %), Селтинском (63,2 %) районах. (см. рисунок 1 и таблица 1) [20].

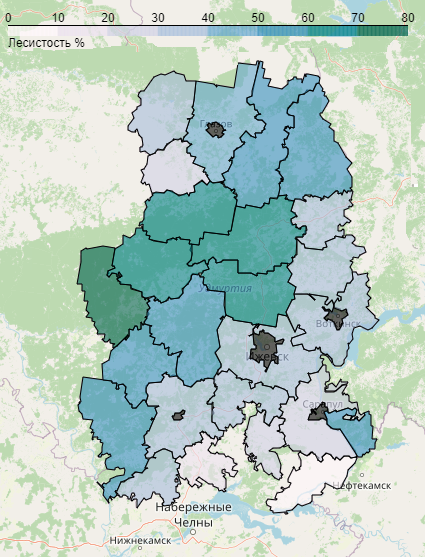


Рисунок – Лесистость территории УР

Городские леса располагаются на землях населенных пунктов городов Ижевска (8,2 тыс. га), Воткинска (2,3 тыс. га), Глазова (1,8 тыс. га), Сарапула (2,0 тыс. га), Можги (0,3 тыс. га) и не включаются в общую площадь района

*Таблица 1 –* Характеристика лесных ресурсов районов УР на 01.01.2019

| **№ п/п** | **Район** | **Площадь района,***км2* | **Площадь, покрытая лесами,** *км2* | **Покрытие лесами,** % | **Общий запас древесины,** *млн. м3* |
| --- | --- | --- | --- | --- | --- |
| 1 | Алнашский | 896 | 146 | 16,3 | 3,2 |
| 2 | Балезинский | 2 435 | 1 351 | 55,7 | 23,2 |
| 3 | Вавожский | 1 679 | 905 | 51,4 | 13,3 |
| 4 | Воткинский | 1 864 | 682 | 38,0 | 11,2 |
| 5 | Глазовский | 2 160 | 969 | 43,0 | 12,8 |
| 6 | Граховский | 968 | 346 | 35,2 | 6,1 |
| 7 | Дебесский | 1 033 | 388 | 36,3 | 6,4 |
| 8 | Завьяловский | 2 203 | 861 | 38,8 | 17,8 |
| 9 | Игринский | 2 267 | 1 484 | 63,5 | 22,1 |
| 10 | Камбарский | 763 | 363 | 55,0 | 5,7 |
| 11 | Каракулинский | 1 193 | 91 | 6,9 | 0,9 |
| 12 | Кезский | 2 321 | 1 354 | 56,9 | 18,3 |
| 13 | Кизнерский | 2 131 | 1 237 | 56,9 | 21,4 |
| 14 | Киясовский | 821 | 195 | 23,1 | 2,8 |
| 15 | Красногорский | 1 860 | 1 202 | 63,8 | 16,6 |
| 16 | Малопургинский | 1 223 | 411 | 32,8 | 7,7 |
| 17 | Можгинский | 1 997 | 825 | 38,6 | 14,2 |
| 18 | Сарапульский | 1 878 | 425 | 25,4 | 6,8 |
| 19 | Селтинский | 1 884 | 1 229 | 63,2 | 20,7 |
| 20 | Сюмсинский | 1 790 | 1 328 | 72,3 | 21,1 |
| 21 | Увинский | 2 445 | 144 | 55,3 | 24,9 |
| 22 | Шарканский | 1 405 | 486 | 33,5 | 8,9 |
| 23 | Юкаменский | 1 020 | 274 | 26,4 | 4,2 |
| 24 | Якшур–Бодьинский | 1 780 | 1 191 | 65,0 | 20,3 |
| 25 | Ярский | 1 524 | 534 | 33,5 | 7,7 |
| **Итого:** | | **41 540** | **18 475** | **46,1** | **318,3** |

В целом лесные ресурсы Удмуртской Республики достаточны не только для обеспечения внутренних потребностей рынка Удмуртии в различных видах лесоматериалов, но и для поставки в значительных объемах за пределы республики. Потенциал лесного комплекса позволяет вносить существенный вклад в его развитие, т.к. УР обеспечена достаточными запасами возобновляемых сырьевых ресурсов, которые без экологического ущерба позволяют заготавливать около 2,6 млн. куб. м. древесины

*Таблица 2 –* Потребности тут/год

| **№ п/п** | **Район** | **Потребности тут/год** |
| --- | --- | --- |
| 1 | Алнашский | 394.800 |
| 2 | Балезинский | 1785.100 |
| 3 | Вавожский | 1976.390 |
| 4 | Воткинский | 152.00 |
| 5 | Глазовский | 4019.380 |
| 6 | Граховский | 484.600 |
| 7 | Дебесский | 2302.500 |
| 8 | Завьяловский | 589.500 |
| 9 | Игринский | 9295.880 |
| 10 | Камбарский | 3229.800 |
| 11 | Каракулинский | 902.300 |
| 12 | Кезский | 1851.380 |
| 13 | Кизнерский | 2645.00 |
| 14 | Киясовский | 1474.700 |
| 15 | Красногорский | 826.210 |
| 16 | Малопургинский | 508.400 |
| 17 | Можгинский | 2333.280 |
| 18 | Сарапульский | 59.100 |
| 19 | Селтинский | 921.600 |
| 20 | Сюмсинский | 2204.808 |
| 21 | Увинский | 605.770 |
| 22 | Шарканский | 3458.162 |
| 23 | Юкаменский | 2432.450 |
| 24 | Якшур–Бодьинский | 765.600 |
| 25 | Ярский | 3353.308 |
|  | Общий итог | 48572.093 |

# 2 МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ЛОГИСТИКИ ТОПЛИВОСНАБЖЕНИЯ РЕГИОНА

# 2.1 Содержательная и концептуальная постановка задачи логистики топливоснабжения

При переводе распределенной системы теплоснабжения региона на древесные виды топлива особую актуальность приобретает задача логистики топливоснабжения.

Информация о местах расположения предприятий лесозаготовки и деревообработки, населенных пунктах с теплоисточниками, которые планируется перевести с традиционных видов топлива (уголь, мазут, электроэнергия) на альтернативный вид топлива (щепа, пеллеты) используется в качестве исходной информации для решения задач логистики.

Логистическая схема снабжения теплоисточников топливом состоит из четырех уровней (рисунок 2). На первом уровне образуется сырье, которое затем доставляется на пункты накопления, откуда затем поступает на пункты подготовки сырья в топливо и уже оттуда поступает к потребителям. [19]

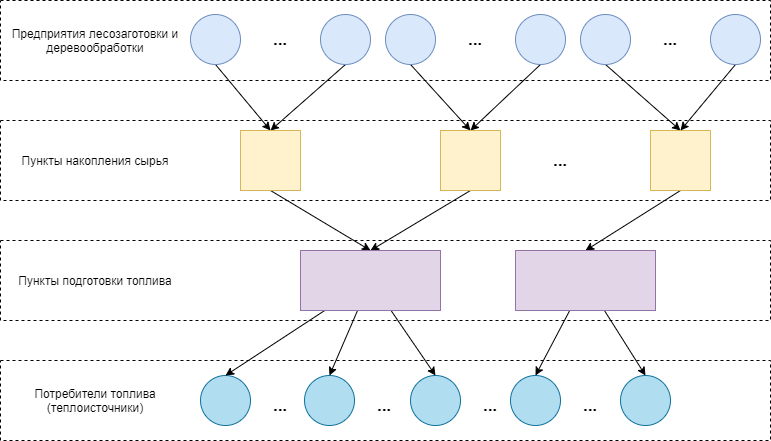


Рисунок – Логистическая схема снабжения теплоисточников топливом

Решения задач маршрутизации транспорта, кластеризации пунктов доставки и оптимизации распределения ресурсов являются основными в решении задач логистики.

*Задача маршрутизации.*

Решение задачи маршрутизации заключается в определении оптимального набора маршрутов на графе для доставки сырья и топлива, полученных в результате минимизации транспортных расходов.

(1)

Задача маршрутизации сводится к задаче поиска оптимального маршрута на связном неориентированном графе. На граф могут быть наложены дополнительные условия, такие как, например, ограничение скорости при передвижении по ребру, ограниченный объем склада в узле, максимальная пропускная способность ребра и т.д. [21]

Для решения задачи маршрутизации используют следующие категории методов [19]:

1. Простые (алгоритм Дейкстры [14], поиск в ширину (Breadth–First Search) [13]) –поиск по всем направлениям и для нахождения оптимального маршрута необходимо как минимум раз пройти по всем вершинам графа. На рисунке 3 приведен пример алгоритма Дейкстры на псевдокоде.

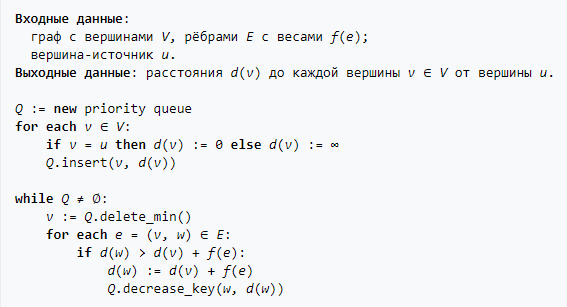


Рисунок – Псевдокод алгоритма Дейкстры

1. Информированные (алгоритм А\*(A star, A звезда) [18], волновой алгоритм (алгоритм Ли) [16]) – поиск в наиболее вероятном направлении, для нахождения оптимального маршрута так же, как и в простейших, необходимо пройти все вершины, но решение находится быстрее. На рисунке 4 приведен псевдокод алгоритма А\*.
2. Эвристические (генетический алгоритм [3]) – поиск в наиболее вероятном направлении и вокруг наиболее вероятного узла. Решение эвристическими методами быстрее, находится близкий к оптимальному маршрут. На рисунке 5 представлен псевдокод генетического алгоритма.

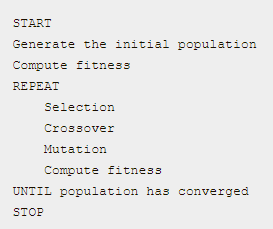


Рисунок – Псевдокод процесса обучения генетического алгоритма

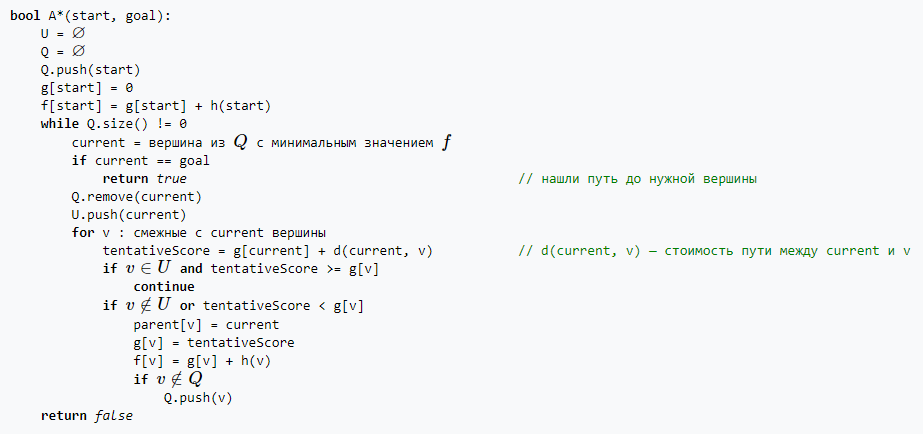


Рисунок – Псевдокод алгортима А\*

*Задача кластеризации*

Для определения месторасположения пунктов накопления древесного сырья и пунктов подготовки топлива решается задача кластеризации

Кластерный анализ – группировка наблюдений таким образом, чтобы при создании группы (или кластера, или класса) соблюдалось следующее свойство: внутри группы наблюдения должны быть максимально похожими, в то время как наблюдения, принадлежащие разным группам, должны быть различающимися насколько возможно. Осуществляется путем вычисления меры близости между объектами [15].

Существует ряд различных методов, которые можно использовать для проведения кластерного анализа, эти методы можно классифицировать следующим образом:

1. Иерархические: агломеративные и дивизимные. Агломеративные – это такие методы, при которых субъекты начинают в своем отдельном кластере. Затем два «ближайших» (наиболее похожих) кластера объединяются, и это делается многократно, пока все субъекты не окажутся в одном кластере. В конце концов, оптимальное количество кластеров затем выбирается из всех кластерных решений. Дивизимные это те алгоритмы, при которых все предметы начинаются в одном кластере, и вышеуказанная стратегия применяется в обратном порядке, пока каждый предмет не окажется в отдельном кластере. Таким образом кластерное решение представляет из себя иерархическую структуру вложенных друг в друга кластеров. Результатом иерархического кластерного анализа являются дендрограммы, которые описывают близость отдельных точек и кластеров друг к другу [9].

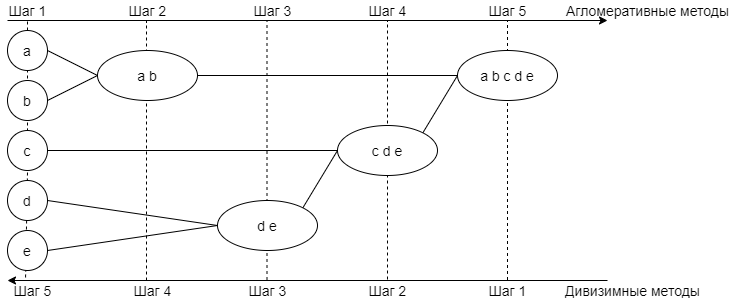


Рисунок – Дендограмма агломеративных и дивизимных методов

1. Неирархические методы используются при большом количестве наблюдений. Данные методы направлены на группировку объектов, которая максимизирует или минимизирует некоторый критерий оценки. Многие из этих алгоритмов итеративно назначают объекты в различные группы (кластеры, классы) определяя некоторое оптимальное значение критерия. В процессе дробления новые классы формируются пока не будет достигнут критерий останова. Таким образом, неирахические методы требуют предварительных предположений о конечном числе кластеров. Наиболее распространённым неирархическим методом является алгоритм *k*-средних Кластеризация этим алгоритмом предназначена для назначения объектов определенному пользователем количеству кластеров (*k*) таким образом, чтобы максимизировать разделение этих кластеров при минимизации внутрикластерных расстояний относительно среднего значения или центроида кластера. Суть алгоритма в минимизации среднеквадратичного отклонения на точках каждого кластера:

(2)

где – число кластеров; – число элементов в *i*–м кластере, ; – элементы выборки, входящие в *i*–й кластер; – элемент, являющийся центром *i*–го кластера.

Основная идея метода *k*–средних заключается в том, что на каждой итерации пересчитываются центр каждого кластера, полученного на предыдущем шаге, затем элементы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике. Алгоритм завершает свою работу, когда дальнейшее изменение кластеров не происходит [11].

1. Методы кластеризации “с обучением” предполагают, что количество кластеров известно заранее и имеется обучающая выборка – набор объектов, для которых известно к каким классам они принадлежат. Остальные объекты классифицируются по степени близости их признаков к объектам из обучающей выборки.

Задача проведения кластерного анализа ставит вопрос адекватности полученного решения, но проверить это возможно с использованием других методов [11].

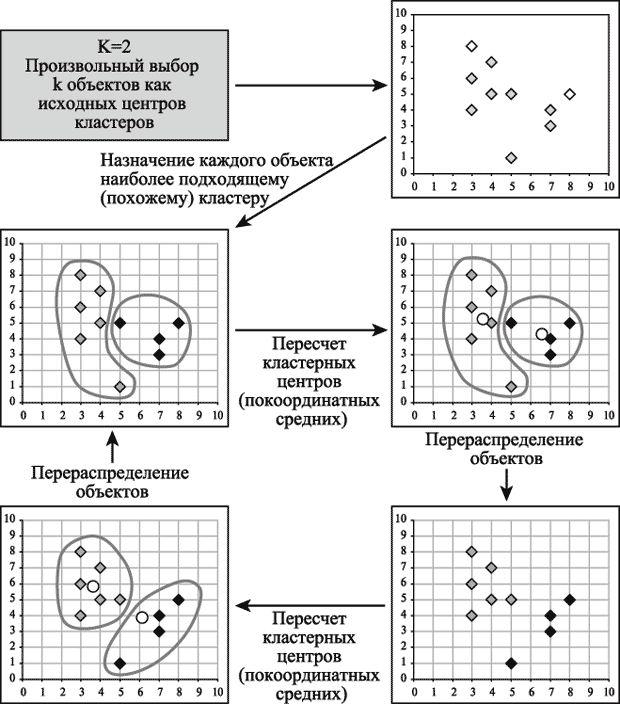


Рисунок – Пример работы алгоритма k–средних при k=2

*Задача оптимального распределения ресурсов*

Задача оптимального распределения ресурсов решается на двух уровнях: районном и региональном. На районном уровне это закрытая задача распределения внутри одного района, на региональном это уже отрытая задача о распределении между районами.

Цель решения задачи в том, чтобы удовлетворить спрос потребителей и эффективно использовать ресурсы. В качестве критерия оптимизации выступает критерий минимизации транспортных расходов, сперва в пределах одного района, затем, в следствии разницы объемов производства и потребления на районном уровне, необходимо перераспределить излишки между ППТ на региональном уровне.

Внутри района с ППТ на теплоисточники доставляется щепа, а между районами из-за больших расстояний транспортировки целесообразно перевозить пеллеты.

Рассмотрим задачу оптимального распределения ресурсов на районном уровне. Критерий минимизации имеет вид:

, (3)

где – удельные транспортные затраты на доставку древесного сырья с *i*-го ПНС на *j*–й ППТ, *руб./т у.т.*, годовой объем древесного сырья, поставляемый с *i-*го ПНС на –й ППТ, *т у.т./год*; *M –* количество ПНС в районе; *N –* количество ППТ в районе.

Введем ограничения на объемы поставок:

*,* (4)

*,* (5)

Где – годовой объем переработки древесного сырья на *i*–м ПНС, *т у.т./год*; – годовой объем переработки древесного сырья на *j*–м ППТ, *т у.т./год.*

На региональном уровне критерий минимизации имеет вид:

, (6)

, (7)

, (8)

где – удельные транспортные затраты на доставку излишков топлива с *i*–го ППТ на *j*–й ППТ с дефицитом топлива, *руб./т у.т.*; – годовой объем перевозки топлива между *i-*м и *j*–м ППТ, *т у.т./год*; годовой дефицит топлива на *j*-м ППТ, *т у.т./год*; – годовой излишек топлива на *i*–м ППТ, *т у.т./год*; – количество ППТ, на которых образуется излишек топлива;количество ППТ с дефицитом топлива.

# 3. АЛГОРИТМЫ И МЕТОДИКА РЕШЕНИЯ ЗАДАЧ ЛОГИСТИКИ

# 3.1 Алгоритм и методика решения задачи маршрутизации

Первым шагом в решении задач логистики является решение задачи маршрутизации. В качестве критерия оптимизации рассматривается минимизация транспортных расходов, то есть нахождение кратчайших маршрутов от пунктов подготовки топлива до потребителей.

Общая постановка задачи маршрутизации транспорта выглядит следующим образом:

1. Задается множество вершин .
2. – вершина, в которой построенные маршруты начинаются и заканчиваются, то есть в случае, рассматриваемом в данной работе – пункт накопления топлива.
3. *–* множество из *N* целевых вершин для посещения.
4. Задается матрица расстояний, в качестве элементов содержащая значения расстояний или стоимости доставки между заданными потребителями топлива.
5. Построение заранее заданного или определенного в ходе работы числа маршрутов транспортных средств минимальной суммарной длины или стоимости, которые начинаются и заканчиваются в вершине , а каждая вершина из включена в маршрут.

Таким образом исходными данными для решения задачи маршрутизации транспортных средств является матрица расстояний, содержащая в качестве элементов значения расстояния между заданными потребителями топлива – вершинами графа и , , *N* – количество рассматриваемых населенных пунктов. Для ее определения входными данными являются географические координаты населенных пунктов с теплоисточниками.

Для решения задачи маршрутизации в системе топливоснабжения региона использованы модули distance–matrix и directions библиотеки openrouteservice–py, которые позволяют построить матрицу расстояний и проложить маршрут по существующей транспортной сети между пунктами, соответственно. Критерием оптимизации модулю directions был определен параметр минимального расстояния между пунктами отправления и назначения [6].

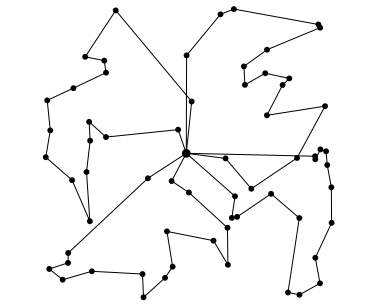


Рисунок – Пример работы алгоритма заметающей прямой

В результате решения находятся маршруты транспортировки топлива между объектами логистической системы (пункты потребления топлива и пункты подготовки топлива).

Для нахождения оптимальных маршрутов необходимо решить задачу кластеризации как несколько задач коммивояжера.

# 3.2. Алгоритм и методика решения задачи кластеризации

Кластерный анализ в решении задач логистики применительно к распределенной системе топливообеспечения проводится для определения местоположения пунктов подготовки топлива и связанных с ними потребителей топлива. После определения центров кластеров снова решается задача маршрутизации уже отдельно для каждого кластера.

Исходными данными являются сведения о местонахождении населенных пунктов с теплоисточниками, которые планируется перевести с угля на щепу и пеллеты. Необходимо оптимальным образом объединить населенные пункты с теплоисточниками в группы расположенных близко друг к другу объектов и в каждом кластере выделить один населенный пункт, в котором планируется разместить пункт подготовки топлива при условии минимизации расходов на доставку топлива до теплоисточников.

Пусть – множество населенных пунктов с теплоисточниками, , *N* – количество рассматриваемых населенных пунктов. Обозначим множество близкорасположенных друг к другу населенных пунктов, образующих *p*-й кластер , , *K* – количество кластеров, *j* – номер населенного пункта в кластере, , – количество населенных пунктов в кластере.

Необходимо провести разбиение множества населенных пунктов на кластеры таким образом, чтобы общие затраты *Z* (руб./год) на транспортировку топлива с ППТ до теплоисточников были минимальны:

(9)

где – населенный пункт, в котором планируется расположить ППТ (центр кластера); – транспортные затраты на транспортировку топлива (руб./т у.т.) от центра кластера до *j*–го населенного пункта; суммарная годовая потребность в топливе теплоисточников *p*–го кластера, расположенных в *j*–м населенном пункте, т у.т./год.

Общий алгоритм решения задачи кластеризации объектов региональной распределенной системы теплоснабжения состоит из двух этапов: на первом этапе применяется иерархический кластерный анализ, на втором – метод *k*–средних. Иерархический кластерный анализ используется для определения оптимального количества кластеров. Метод *k*–средних применяется для распределения объектов по кластерам и определения оптимальных мест расположения ППТ, которые выступают центрами данных кластеров [28].

*Алгоритм иерархического кластерного анализа*

1. Задается множество населенных пунктов–теплоисточников .
2. Рассчитывается матрица, элементами которой являются расстояния между теплоисточниками.
3. В начальный момент каждый населенный пункт содержится в своем собственном кластере.
4. Итеративно происходит слияние двух ближайших кластеров до тех пор, пока все кластеры не объединятся в один или не будет найдено необходимое количество кластеров.

Для вычисления расстояния между кластерами используются различные целевые функции в зависимости от специфики задачи. Для решения задачи кластеризации объектов региональной распределенной системы теплоснабжения была использована функция метода одиночной связи (single linkage):

(10)

где U, V – кластеры, – расстояние между одноэлементными кластерами.

В данной работе для решения задачи кластеризации использовались модули scipy.cluster.hierarchy и scipy.cluster.vq библиотеки SciPy [26].

По результатам проведения иерархического кластерного анализа строится дендрограмма (рисунок 10), которая отображает близость отдельных населенных пунктов и кластеров друг к другу.

В результате выполнения иерархического кластерного анализа определяется оптимальное число кластеров. Для распределения объектов по кластерам и определения их центров используется метод *k*–средних.

Алгоритм метода *k*–средних заключается в том, что метод произвольно разбивает множество наблюдений на заранее известное количество кластеров *K,* в которых каждое наблюдение относится к кластеру с ближайшим средним (центры или центроиды кластеров), который по сути является прототипом кластера. Затем на каждой итерации пересчитываются центр каждого кластера, полученного на предыдущем шаге, затем элементы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике. Алгоритм завершает свою работу, когда дальнейшее изменение кластеров не происходит.

Рисунок – Построение дендрограммы на примере Вавожского района УР



*Алгоритм метода k–средних*

1. Задается количество кластеров *K*, полученное в результате проведения иерархического кластерного анализа и определяются центры кластеров:

, (11)

Где *N* - количество населенных пунктов.

1. Каждый населенный пункт закрепляется за определенным кластером, , исходя из минимизации транспортных расходов до центров кластеров:

(12)

1. Переопределяются центры кластеров:

(13)

1. Если происходит изменение центров кластеров или перераспределение объектов, то осуществляется переход к шагу 3.

Если перераспределения не происходит, то полученные центры кластеров и распределение объектов по кластерам считаются оптимальными.

В результате кластерного анализа определяются местоположения ППТ и происходит распределение объектов по кластерам.

# 3.3. Алгоритм и методика решения задачи оптимального распределения ресурсов

Задача оптимального распределения ресурсов решается на двух уровнях: районном и региональном.

Исходными данными для решения задачи оптимального распределения на районном уровне являются данные, полученные после проведения кластерного анализа в отношении местоположения ППТ, а также информация об объемах накопления сырья в ПНС и об объемах подготовки топлива ППТ.

Если существуют различия в объемах производства и расходе топлива внутри кластеров, необходимо решить проблему оптимального распределения избыточного топлива между ППТ в разных районах.

Алгоритм решения задачи оптимального распределения ресурсов приведен в пункте 2.1.

Результатами решения задач оптимального распределения ресурсов на районном и на региональном уровнях являются оптимальные планы поставок древесного сырья и топлива.

# РАЗРАБОТКА ВЕБ–ПРИЛОЖЕНИЯ ДЛЯ РЕШЕНИЯ ЛОГИСТИЧЕСКИХ ЗАДАЧ

# Обзор технологий и средств разработки веб–приложений



Рисунок – Структурные уровни сайта

Основной характеристикой современного общества является широкое внедрение информационных технологий и информатизация общества во всех сферах деятельности.

Привлекательность веб–технологий как средства доставки информации во многом определяет универсальный и интуитивно понятный интерфейс. Следствием этого является широкое употребление Интернета как канала коммуникации. Тем привлекательнее использование веб–приложений, что даже на самом устаревшем компьютере или мобильном устройстве присутствует программа для просмотра веб–страниц (браузер).

Веб–приложение – это клиент–серверное приложение, основная часть которой содержится на удаленном сервере, а пользовательский интерфейс отображается в браузере в виде веб–страниц.

Для веб–разработки в данное время используется множество технологий, но основой всего Интернета является язык гипертекстовой разметки (HTML). Клиентская часть реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него. Для реализации клиентской части веб–приложения используется HTML. Он необходим для отображения содержимого сайта: текст, изображения, таблицы и т.д.

Для настройки стиля отображения веб–страниц используют каскадные таблицы стилей (CSS), которые с помощи специального языка задают форматирование элементов в HTML–документе.

Для придания динамики и интерактивности используют JavaScript – скриптовый язык программирования, используемый в составе HTML–документа. JavaScript используют, например, для создания документа с помощью сценария, проверки введенных пользователем данных.

Использование HTML, CSS и JavaScript не требуют установки и настройки на сервере, так как выполняются непосредственно на устройстве пользователя.

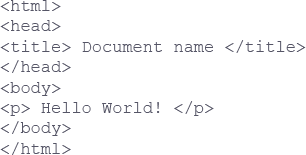


Рисунок – Пример содержимого HTML-документа

Серверная часть веб–приложения — это программа или скрипт на сервере, обрабатывающая запросы браузера. То есть, сервер получает запрос от клиента, выполняет вычисления, после этого формирует веб–страницу и отправляет её клиенту по сети с использованием протокола HTTP. Серверная часть реализуется на таких языках программирования как Python, PHP, Ruby и других, способных осуществлять вывод в стандартную консоль.

Для хранения содержимого сайта используются базы данных. База данных представляет собой набор таблиц, где каждая таблица – это сущность, в которой хранятся однотипные данные. Преимущество хранения контента в базе данных заключается в том, что благодаря поддержке SQL и наличию логических связей, управление базой становится простым и быстрым.

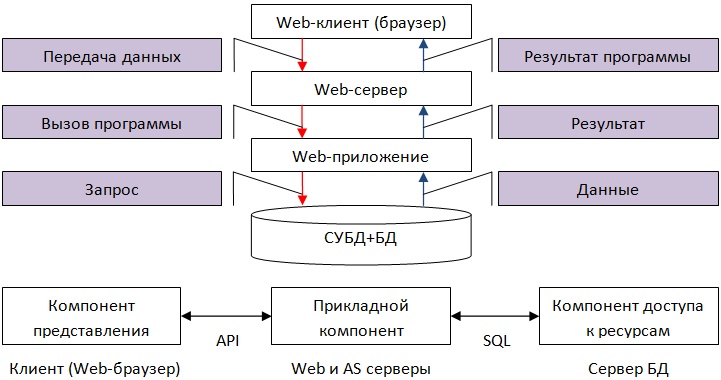


Рисунок 12 – Обобщенная схема работы веб–приложений

Для разработки веб–приложения для решения задач логистики был выбран, кроме основополагающих HTML, язык программирования Python, так как он позволяет разрабатывать веб–приложения с помощью фреймворка Django, который, в свою очередь, имеет встроенную по умолчанию систему управления базами данных SQLite, в следствии чего отпадает проблема сложности выбора и изучения множества средств для решения поставленных выше задач. Фреймворк необходим для того, чтобы пользователи могли взаимодействовать с приложением через браузер.

Язык программирования Python был выбран как основной, так же потому, что он имеет большое количество хорошо документированных библиотек, с помощью которых процесс разработки упрощается. С его помощью приложение генерирует HTML–код со стороны сервера.

Фреймворк Django так же хорошо документирован по всем версиям и поэтому его изучение и использование не вызывает больших трудностей. Так же несомненный положительный аспект выбора Django, это то, что при создании веб–приложения с его помощью, автоматически создается строго структурированная система директорий и интерфейс администратора базы данных.

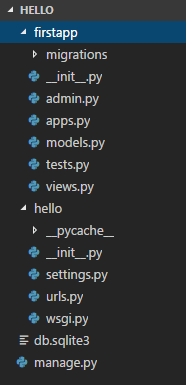


Рисунок – Пример структуры проекта Django

Структура проекта на Django состоит из таких основных файлов:

* settings.py содержит в себе все настройки проекта: здесь регистрируются приложения, задается размещение статичных файлов, настройки базы данных и т.д;
* urls.py задает ассоциации url адресов с представлениями. Обычно в одном проекте таких файлов несколько – по одному на каждое приложение;
* wsgi.py используется для связи приложения и сервера;
* **manage.py** используется для создания приложений, работы с базами данных и для запуска отладочного сервера;
* views.py – файл для контроллеров;
* models.py – файл для моделей;
* admin.py содержит настройки административной части;
* apps.py – файл для регистрации приложений.

При создании проекта эти файлы создаются автоматически и в некоторых из них уже содержится некоторый шаблонный код.

Так же были создан файл \_\_init\_\_.py – пустой файл для распознования папки как Python–модуль и использования его объектов внутри других частей проекта; и папка migrations – используется для хранения "миграций" — файлов, которые позволяют автоматически обновлять базу данных по мере изменения моделей.

# Разработка интерактивной карты региона

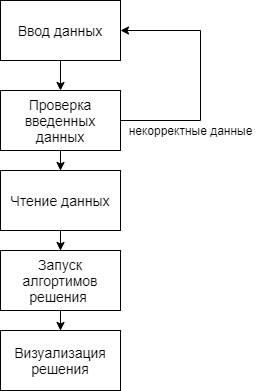
Прежде чем приступить к реализации веб–приложения для решения задач логистики, в частности – маршрутизации транспортных средств, было решено провести анализ существующих на рынке программного обеспечения решений, так как логистика сейчас это одна из развивающихся прикладных наук.

В таблице 3 представлен сравнительный анализ различного программного обеспечения от российских и иностранных компаний, представленных на рынке. По представленным в ней данным видно, что малое количество программного обеспечения предоставляют онлайн–доступ для оптимизации маршрутов транспортных средств. Это и явилось одним из факторов разработки веб–приложения для решения этой проблемы.

*Таблица 3* – Сравнительный анализ программного обеспечения

| Характеристики | Наименование программного обеспечения | | | | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Автобаза (БОРНИКА) | ДТКсофт (ИАС "Грузоперевозки") | РАРУС (Управление автотранспортом) | Управление перевозками (Axelot) | ФОРЕС (Автотранспорт) | RoutePlanner (ArcGIS, ESRI) | IBM ILOG Transportation Analyst (IBM) | logvrp (NetAkıl) | Roadnet Transportation Suite (Roadnet Technologies) | ITOB (Центр логистики) | CDC (ОПТИМУМ ГИС) | АНТОР (LogisticsMaster) | Деловая карта (ИНГИТ) | ТопПлан (TopLogistic) | Эрмасофт (СИТИ – Доставка) |
| геокодирование | – | – | – | – | – | + | + | + | + | + | + | + | + | + | + |
| online–сервис | – | – | – | – | – | + | – | + | – | – | – | – | – | – | – |
| итоговые отчеты | – | + | + | + | + | + | + | + | + | + | – | + | + | + | + |
| учет временных окон | – | – | – | – | – | + | + | + | – | – | – | – | – | – | – |
| возможность построения сразу нескольких маршрутов | – | – | – | – | – | + | + | + | + | + | + | + | + | + | + |
| возможность разбиения заказа по ТС | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + |
| учет грузоподъемности ТС | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + |
| учет типа и качества дорог и т.д. | – | – | – | – | – | + | + | + | + | + | + | – | + | – | + |
| учет множества депо | – | – | – | – | – | + | + | + | – | – | – | – | – | – | – |
| учет раздельной доставки | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – |
| учет периода планирования | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + |
| алгоритм | нет информации | нет информации | нет информации | нет информации | нет информации | эвристический | эвристический | эвристический | эвристический | нет информации | эвристический | эвристический | эвристический | эвристический | нет информации |
| стоимость, руб (2015 г) | 22150 | 51840 | 45900 | 62000 | 35000 | 28421 | – | 6000 | 26315 | 131000 | – | 1754 | 600 | 2105 | – |

Рисунок – Упрощенная схема работы с данными



На первом этапе разработки было решено использовать данные картографического сервиса OpenStreetMap [7].

Большинство данных было представлено в форматах o5m и osm – форматы, предоставляемые проектом OpenStreetMap. Внутри они по своей сути являются разновидностью XML файлов и для их обработки необходимо воспользоваться специальными программами:

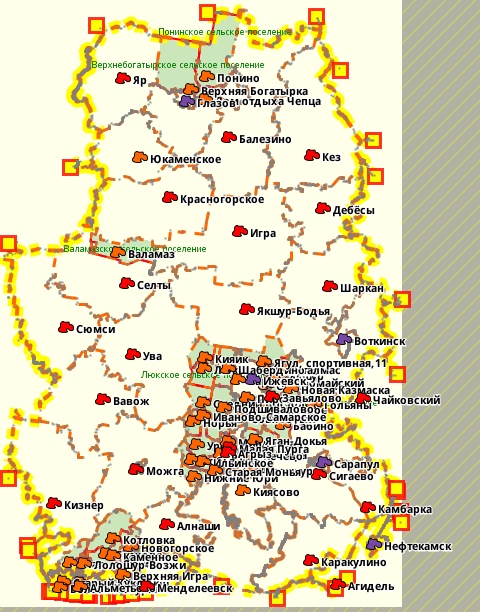


Рисунок 15 – Отображение данных о районах (после использования osmfilter) в заданном прямоугольнике в программе JOSM

* JOSM – для обработки данных так, как они отображаются на карте – это достаточно ресурсозатратно для отображения всех данных по всему региону (например, файл с данными в полигоне, ограниченном прямыми проходящими по координатам 51.059, 55.785,54.488, 58.588 (мин. долгота, мин. широта, макс. долгота, макс. широта) по УР размером около 500МБ делал невозможным работу в этом приложении на рядовом ПК). Присутствует возможность сохранить файл в json–формате.
* osmfilter – инструмент для работы с файлами в osm–формате, посредством которого из исходного файла можно удалить не интересующую информацию. Работа с этой программой проходит в консоли, что означает, что разработчику необходимо иметь базовые навыки работы в консоли. Так же необходимо быть внимательным, чтобы не удалить случайно нужную информацию

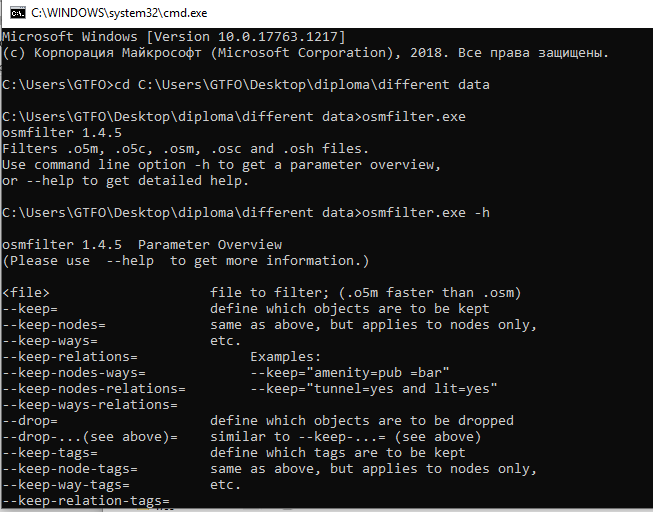


Рисунок 16 – Вызов справки в приложении osmfilter

* osmconvert – так же, как и osmfilter является консольным приложением, но предназначена для конвертирования файлов различные форматы проекта OpenStreetMap и преобразует данные в выбранный формат выходного файла.

Итогом работы с данными стали несколько файлов в форматах json и xlsx, с которыми и должна была заключаться основная часть работы по формированию слоев для отображения на подложке карты и решения задач логистики. Но проанализировав данные и то, как они выводятся, стало ясно, что данные, предоставляемые OpenSteetMap являются избыточными и работа с ними трудоемка и никак не помогает приблизиться к адекватному решению поставленных задач. На рисунке 17 приведен фрагмент визуализированных данных о границах районов и дорогах, демонстрирующий избыточность данных.

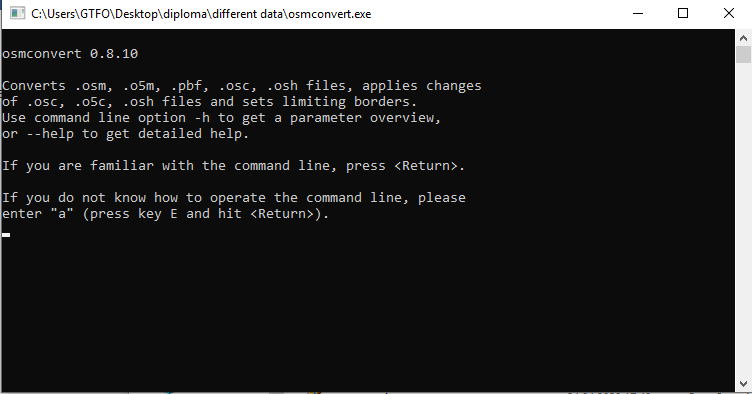


Рисунок 18 – Окно приложения osmconvert

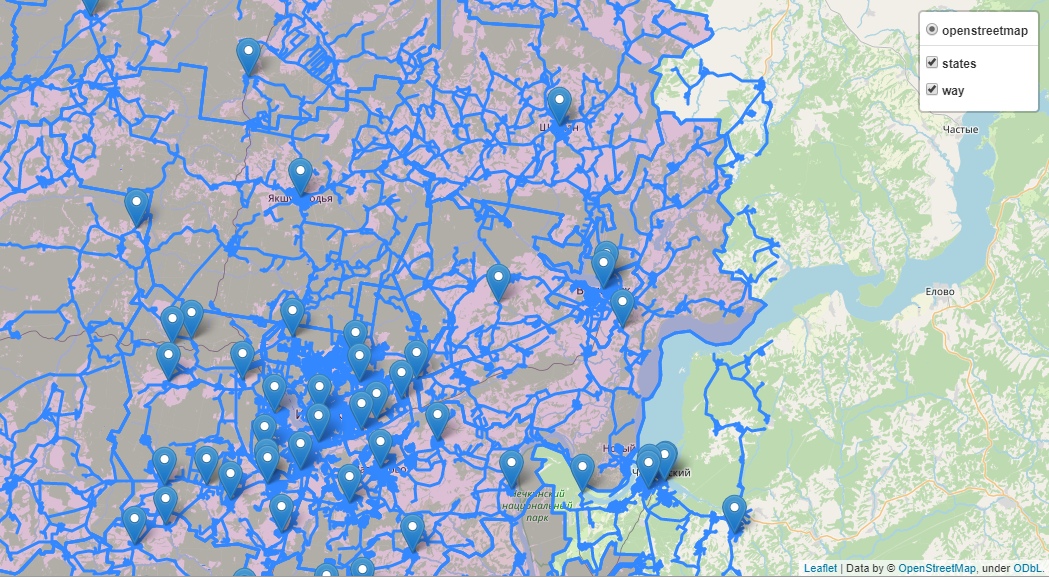
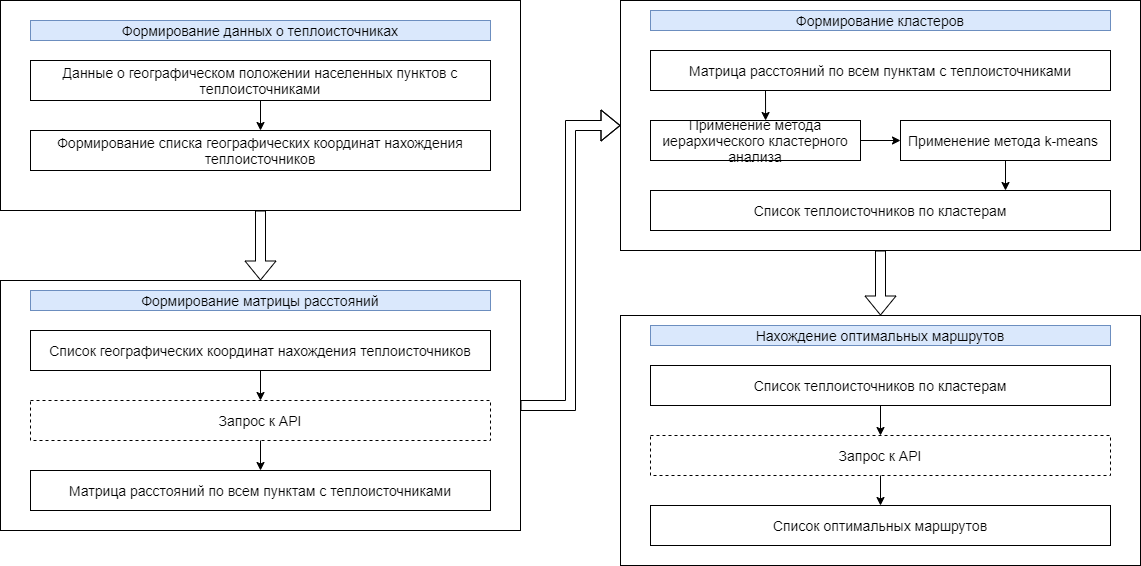


Рисунок – Фрагмент визуализированных данных о границах районов и дорогах, реализованных по данным OpenStreetMap

Поэтому было решено использовать API OpenRouteService, который также использует данные OpenSteetMap, но работа с ними упрощается, благодаря реализации в соответствующей библиотеке функций для построения маршрутов, их оптимизации и т.д.

Для разработки интерактивной карты использовался язык программирования Python и библиотеки:

* Pandas – библиотека для обработки и анализа данных;
* Folium – библиотека для визуализации данных на карте;
* openrouteservice–py – библиотека для решения задачи маршрутизации транспорта;
* SciPy – для решения задачи кластеризации;

На рисунке 19 представлен упрощенный алгоритм решения задач логистики. Итогом работы данного алгоритма является сгенерированный HTML–документ, который отображается пользователю.

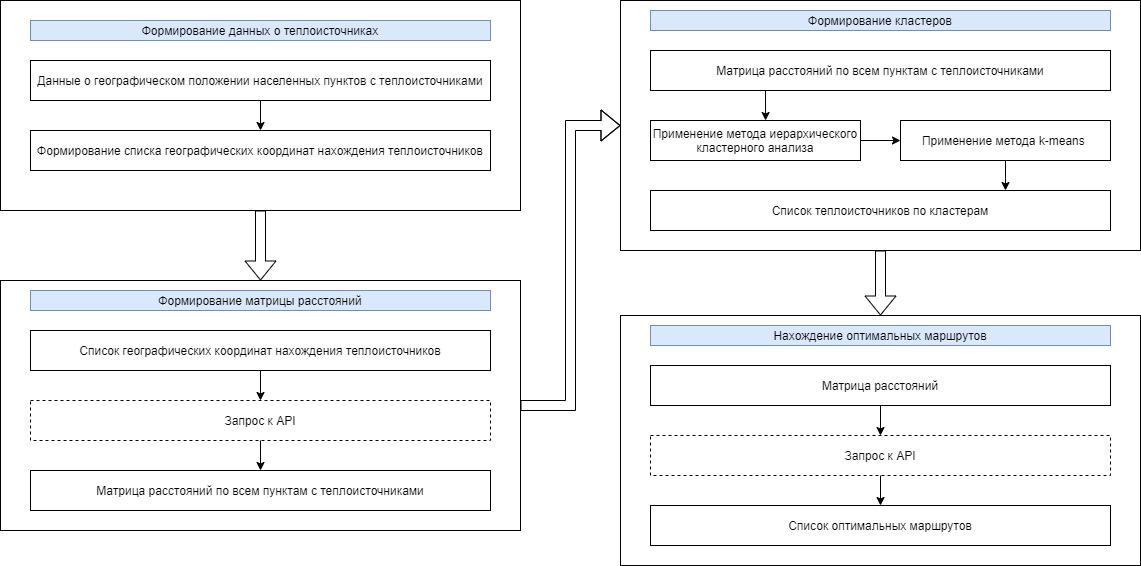


Рисунок – Алгоритм решения задач

Опишем проделанную работу по данному алгоритму.

Формирование данных о теплоисточниках происходит по данным, вносимым пользователем – пользователь загружает через форму на сайте документ в формате xlsx или csv содержащий в себе название населенного пункта, долготу, широту и потребности теплоисточника, этот файл сохраняется на сервер и в базу данных для дальнейшего взаимодействия. На сервере с помощью библиотеки Pandas считывается список населенных пунктов и их координаты, указанные в документе, и происходит формирование матрицы расстояний с помощью модуля distance\_matrix библиотеки Openrouteservice. Для нахождения расстояний модуль использует алгоритм Дейкстры.

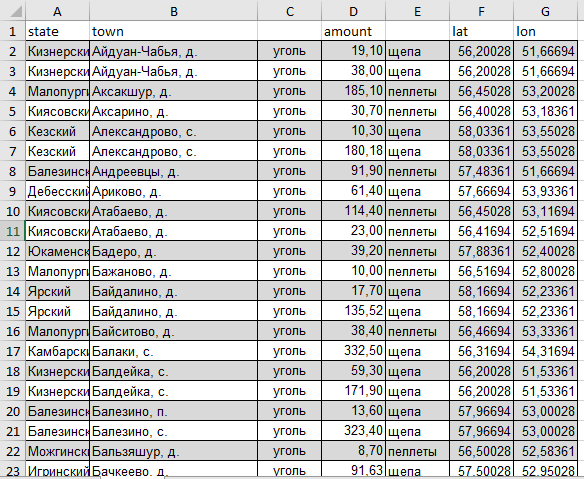


Рисунок 20 – Пример исходных данных о населенных пунктах с котельными

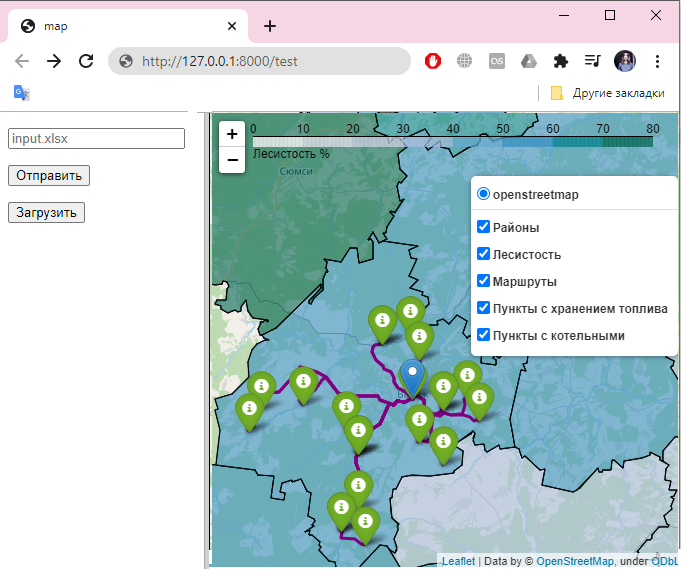
После формирования матрицы расстояний решается задача кластеризации. Для нахождения кластеров использовалась библиотека SciPy, модули scipy.cluster.hierarchy – для иерархического кластерного анализа и scipy.cluster.vq – для использования алгоритма k–means по определению кластеров.

После определения состава кластеров и их центров, используя модуль directions библиотеки openrouteservice–py, строятся оптимальные маршруты внутри каждого кластера. Оптимальный маршрут – маршруты с наименьшим общим расстоянием. Однако, если нет других ограничений, оптимальным решением будет назначить только одно транспортное средство для посещения всех мест и найти кратчайший маршрут для этого транспортного средства. Маршрут начинается и заканчивается в одной и той же точке – пункт накопления топлива заданного кластера.

Задача размещения производства состоит в выборе варианта расположения объектов, при котором оптимизируется некоторый показатель, характеризующий результативность размещения, в данной работе этим показателем является расстояние от пунктов переработки топлива до потребителей. Задача размещения предприятий и распределения потребителей составляет основу проектирования распределенной системы.

Таким образом решаются поставленные задачи логистики и происходит визуализация решения по данным пользователя.

Рисунок - Интерфейс веб-приложения с примером решения задачи



# ЗАКЛЮЧЕНИЕ

В настоящей бакалаврской работе рассмотрена проблема топливообеспечения населенных пунктов УР топливом на основе древесного сырья. Проведен анализ концепции развития лесохозяйственного комплекса региона и существующего программного обеспечения для решения транспортных задач

Анализ концепций показал, что регион заинтересован в реализации перехода на топливо из возобновляемых источников (древесное сырье и биогаз) уже более 10 лет; а анализ существующего программного обеспечения показал, что на данный момент на рынке малочисленно программную решения с онлайн доступом за приемлемую стоимость.

Так же были рассмотрены модели и алгоритмы решения задач логистики, проведен обзор средств и технологий разработки веб–приложений для реализации алгоритмов решения задач логистики и визуализации полученных результатов на карте.

В процессе разработки изучено создание современных веб–приложений средствами языка программирования Python

В работе были использованы библиотеки

* openrouteservice–py – для нахождения матриц расстояний, маршрутов и их оптимизации;
* Pandas – для работы с данными представленными в виде электронных таблиц;
* SciPy – для решения задачи кластеризации;
* Folium – для визуализации решения на подложке карты.

Итогом работы стало веб–приложение, позволяющее решать логистические задачи маршрутизации транспортных средств и кластеризации на примере территории Удмуртии.

# Литература

* 1. Аникин Б. А. и др. Логистика. – 2015.
  2. Буйначев С. К., Боклаг Н. Ю. Основы программирования на языке Python: учебное пособие. – 2014.
  3. Гладков Л. А., Курейчик В. М., Курейчик В. В. Генетические алгоритмы. – 2006.
  4. Документация Django [Электронный ресурс] https://docs.djangoproject.com/en/3.0/ (дата обращения 23.05.2020)
  5. Документация Folium–Library [Электронный ресурс] https://python–visualization.github.io/folium/ (дата обращения 23.05.2020)
  6. Документация openrouteservice–py [Электронный ресурс] https://openrouteservice–py.readthedocs.io/en/stable/ (дата обращения 23.05.2020)
  7. Документация Python 3.7 [Электронный ресурс] https://docs.python.org/3.7 (дата обращения 23.05.2020)
  8. Дронов В. А. Django 2.1: практика создания Веб–сайтов на Python. – БХВ–Петербург, 2019.
  9. Дюран Б., Оделл П. Кластерный анализ. – М.: Статистика, 1977. – 128 с.
  10. Жиглявский А. А., Жилинскас А. Г. Методы поиска глобального экстремума. – Наука. Гл. ред. физ.–мат. лит., 1991. – С. 248.
  11. Ким Д. О., Мьюллер Ч. У., Клекка У. Р. Факторный, дискриминантный и кластерный анализ. – М.: Финансы и статистика, 1989. – 215 с.
  12. Концепция развития лесного хозяйства и лесной промышленности Удмуртской Республики на период 2019–2030 гг. [Электронный ресурс] http://www.minpriroda–udm.ru/images/docs/deyatelnost/proekti\_npa/Koncepciya\_LPK\_UR\_2019–2030\_ot\_12.2018.pdf (дата обращения 23.05.2020)
  13. Кормен Т., Лейзерсон Ч. Алгоритмы: построение и анализ – Introduction to Algorithms. Метод уменьшения размера задачи: Поиск в ширину. – М.: Издательский дом “Вильямс”, 2006. – 1296 c.
  14. Левитин А. В. Алгоритмы: введение в разработку и анализ. Жадные методы: Алгоритм Дейкстры. – М.: Издательский дом “Вильямс”, 2006. – 576 с.
  15. Нефедов Д. Г. Математические модели и методы решения задач оптимального размещения элементов распределенной производственной структуры //Челябинск. – 2015. – С. 005569163.
  16. Нечепуpенко М. И. Алгоpитмы и пpогpаммы pешения задач на гpафах и сетях. – Hовосибиpск: Hаука, 1990. – 260 с.
  17. Нильсон Н. Искусственный интеллект: методы поиска решений Problem–solving Methods in Artificial Intelligence. – М.: Мир, 1973. – 273 с.
  18. Русяк И. Г. и др. Логистика топливоснабжения региона возобновляемыми видами топлива, получаемыми из древесного сырья. – 2011.
  19. Состав земель лесного фонда и земель иных категорий, на которых расположены леса [Электронный ресурс] https://data.gov.ru/opendata/resource/c2c1a1a4–34a3–4270–bcbd–3e0565f2675e (дата обращения 23.05.2020)
  20. Bodin, L. et al., Routing and Scheduling of Vehicles and Crews: The State of the Art, Comput. Oper. Res.,10, 63–211, 1983.
  21. Boschetti A., Massaron L. Python data science essentials. – Packt Publishing Ltd, 2015.
  22. Gillett, B. and Miller, L., A Heuristic Algorithm for the Vehicle Dispatch Problem, Oper. Res., 22, 340–352, 1974.
  23. Hillier, F.S. and Lieberman, G.J., Introduction to Operations Research, McGraw–Hill Science, Columbus, OH, 2002.
  24. Holland, J., Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.
  25. Larson, R. and Odoni, A., Urban Operations Research, Prentice–Hall, Englefood Cliff s, NJ, 1981.
  26. Milovanović I. Python Data Visualization Cookbook. – Packt Publishing Ltd, 2013.
  27. Taha, H., Operations Research: An Introduction (8th Edition), Pearson Prentice Hall, Upper Saddle River, NJ, 2006.
  28. Taylor G. D. (ed.). Logistics engineering handbook. – CRC press, 2007.
  29. VanderPlas J. Python data science handbook: Essential tools for working with data. – "O'Reilly Media, Inc.", 2016.
  30. Westra E. Python geospatial development. – Packt Publishing Ltd, 2013.

# Приложение 1. Листинг программы для визуализации оптимальных маршрутов

from scipy.cluster.hierarchy import \*

from scipy.cluster.vq import kmeans

import folium

import openrouteservice

import pandas as pd

# # defining amount of clusters with hierarchy algo but wrong

# dist = pd.read\_excel('dist.xlsx')

# sam = linkage(dist, method='single', optimal\_ordering=True) # single == min

# de = dendrogram(sam,

# leaf\_rotation=90,

# leaf\_font\_size=6,

# )

# plt.show()

# # print(sam)

# # ths is some shit i dont understand

# centroid, label = kmeans2(dist, 2, minit='points')

# d = single(dist)

# # z = fcluster(d, 100, criterion='distance')

# pprint(centroid)

# # all fine uncomment for new file and etc

client = openrouteservice.Client(key='5b3ce3597851110001cf62488f209e533a5b439d9d9808213c37ec64')

data = pd.read\_excel('output.xlsx')

# # max locals == 50 coz api

lat = data['lat'][:40]

lon = data['lon'][:40]

town = data['town'][:40]

# finding and saving data about distance

coords = []

for i in range(len(lat)):

coords.append((lon[i], lat[i]))

# computing distances with api

matrix = distance\_matrix(

client,

locations=coords,

metrics=['distance'],

units='km',

)

# # writing result in json

with open('matrix.json', 'w') as file:

json.dump(matrix, file)

# saving only distances from distance–matrix

with open('matrix.json', 'r') as file:

dictData = json.load(file)

for key in dictData:

if key == 'distances':

with open('save.json', 'w') as f:

json.dump(dictData[key], f)

# нужно попробовать этот json сохранить сразу в xlsx

# # map

m = folium.Map(location=[56.8502777777778, 53.2169444444444], zoom\_start=7)

fg\_kot = folium.FeatureGroup(name='Пункты с котельными')

fg\_punk = folium.FeatureGroup(name='Пункты с хранением топлива')

# # coord of cluster's centers – non automatic =.=

folium.Marker(location=[58.30327778, 52.26684444], popup='Казаково, д.').add\_to(fg\_punk)

folium.Marker(location=[56.21194444, 52.60027778], popup='Шайтаново, д.').add\_to(fg\_punk)

for lat\_v, lon\_v, town\_v in zip(lat, lon, town):

# folium.Marker(location=[lat, lon], popup=town).add\_to(marker\_cluster)

folium.Marker(location=(lat\_v, lon\_v), popup=town\_v, icon=folium.Icon(color='green')).add\_to(fg\_kot)

style = {'fillColor': 'None', 'color': 'black', 'weight': '1', 'fillOpacity': 0.5} # setting for style\_function

with open('negh.json', 'r', encoding="utf–8") as read\_file:

state\_geo = json.load(read\_file)

# полигоны и маркеры из json (райoны)

folium.GeoJson(state\_geo, name='Районы', style\_function=lambda x: style).add\_to(m)

# # # setting lon lat for routing

cluster\_1 = [4, 5, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25]

coords\_1 = []

coords\_2 = []

for i in range(len(lat)):

if i in cluster\_1:

coords\_1.append((lon[i], lat[i]))

# dirc1 = directions(client, [[52.26684444, 58.30327778], coords\_1], format\_out='geojson')

# folium.features.GeoJson(data=dirc1, name='routes', style\_function=lambda x: style).add\_to(m)

else:

coords\_2.append((lon[i], lat[i]))

# dirc2 = directions(client, [[52.26684444, 58.30327778], coords\_2[i]], format\_out='geojson')

# folium.features.GeoJson(data=dirc2, name='routes', style\_function=lambda x: style).add\_to(m)

# # # routes 40 requests to api per minute so comment/uncomment loops

fg\_routes = folium.FeatureGroup(name='Маршруты') # featuregroup of routes

style\_1 = {'color': 'purple'} # setting for style\_function

go = []

for j in range(len(cluster\_1)):

go.append(–1)

dirc1 = directions(client, [[52.26684444, 58.30327778], coords\_1[j]], format\_out='geojson', radiuses=[go[j], go[j]])

folium.features.GeoJson(data=dirc1, name='Маршруты', style\_function=lambda x: style\_1).add\_to(fg\_routes)

# got = []

# for k in range(38):

# got.append(–1)

# dirc2 = directions(client, [[52.60027778, 56.21194444], coords\_2[k]], format\_out='geojson', radiuses=[got[k], got[k]])

# folium.features.GeoJson(data=dirc2, name='routes', style\_function=lambda x: style\_1).add\_to(m)

# # folium.features.GeoJson(data=dirc1, name='routes', style\_function=lambda x: style).add\_to(m)

# # folium.features.GeoJson(data=dirc2, name='routes', style\_function=lambda x: style).add\_to(m)

# # Choropleth map of foresting

chor = folium.Choropleth(geo\_data=state, data=d,

columns=['name', 'val'],

key\_on='feature.properties.name',

fill\_color='BuPu',

highlight=True,

legend\_name='Лесистость %',

name='Лесистость').add\_to(m)

chor.geojson.add\_child(

folium.features.GeoJsonTooltip(['name', 'val'], labels=False))

fg\_routes.add\_to(m)

fg\_punk.add\_to(m)

fg\_kot.add\_to(m)

folium.LayerControl(collapsed=False).add\_to(m)

# # saving map data in html

m.save('map.html')