

УДК 004.4 : 656 (1-21) (470.22-25)

## ВЕБ-СЕРВИС ДЛЯ ВИЗУАЛИЗАЦИИ ДОРОЖНОЙ СЕТИ ГОРОДА ПЕТРОЗАВОДСКА

В. А. Хитрая, А. В. Хитрый\*

*Институт прикладных математических исследований КарНЦ РАН,  
ФИЦ «Карельский научный центр РАН» (ул. Пушкинская, 11, Петрозаводск,  
Республика Карелия, Россия, 185910), \*andrey.khitryy@gmail.com*

В работе приведено описание программного комплекса, позволяющего визуализировать информацию о состоянии транспортной сети города Петрозаводска. Описана архитектура системы, приведен список источников открытых данных, использованных для построения модели.

Ключевые слова: веб-сервис; программный комплекс; транспортная сеть

Для цитирования: Хитрая В. А., Хитрый А. В. Веб-сервис для визуализации дорожной сети города Петрозаводска // Труды Карельского научного центра РАН. 2023. № 4. С. 54–63. doi: 10.17076/mat1780

Финансирование. Исследование проводилось совместно с органами власти Республики Карелия, при финансовой поддержке Российского научного фонда (грант № 22–11–20015) и Фонда венчурных инвестиций Республики Карелия.

### V. A. Khitraya, A. V. Khitryy\*. WEB SERVICE FOR THE PETROZAVODSK ROAD NETWORK VISUALIZATION

*Institute of Applied Mathematical Research, Karelian Research Centre, Russian  
Academy of Sciences (11 Pushkinskaya St., 185910 Petrozavodsk, Karelia, Russia),  
\*andrey.khitryy@gmail.com*

The paper describes a software that allows to visualize information about the state of the transport network of the city of Petrozavodsk. The architecture of the system is described and the list of open data sources used to build the model is provided.

Keywords: web service; software; transport network

For citation: Khitraya V. A., Khitryy A. V. Web service for the Petrozavodsk road network visualization. *Trudy Karel'skogo nauchnogo tsentra RAN = Transactions of the Karelian Research Centre RAS.* 2023;4:54–63. doi: 10.17076/mat1780

Funding. The study was carried out in collaboration with the Republic of Karelia authorities with financial support from the Russian Science Foundation (grant # 22–11–20015) and the Venture Investment Fund of the Republic of Karelia.

### ВВЕДЕНИЕ

Состояние транспортной системы города крайне важно для его социально-экономического развития, поскольку транс-

портная инфраструктура напрямую влияет на качество жизни каждого жителя. Анализ состояния транспортной системы является решающим этапом для выявления существующих

щих проблем, а также прогнозирования возможных проблем в будущем. Такой анализ может быть существенно упрощен при обеспечении достаточного уровня наглядности, что позволит легко и быстро оценивать текущее состояние системы.

Однако информация о состоянии транспортной системы (например, города Петрозаводска) представлена в не связанных между собой источниках, а иногда практически недоступна без дополнительной обработки. В связи с этим возникает необходимость создания системы, способной объединить в себе данные о структуре транспортной сети, распределении жителей и организаций, характеристиках дорог, а также оценки загруженности транспортной сети, основанные на построении матрицы корреспонденций и мерах центральности. Результаты построения матрицы корреспонденций для г. Петрозаводска представлены в работе [7].

В данной работе представлено описание программного продукта для автоматизированной генерации графа транспортной сети и моделирования транспортных процессов на примере города Петрозаводска, в котором необходимые вычисления могут быть встроены в итерационное построение модели и не требуют ручного вмешательства. Разработанный программный комплекс позволяет повысить качество и эффективность анализа транспортной сети города.

## АРХИТЕКТУРА ПРОГРАММНОГО КОМПЛЕКСА

Наибольший интерес для разработки представляет система, которая могла бы в автоматическом режиме обновлять дорожный граф при изменении любого набора входных данных, а также осуществлять анализ модели после каждого ее изменения.

Чтобы предоставить возможность вносить изменения в процесс построения графа и визуализировать дорожную сеть на любом этапе, система разделена на несколько независимых модулей:

1. Модуль дорожного графа обновляет информацию о дорожной сети.
2. Модули обновления данных насыщают граф необходимой информацией (параметры ребер и вершин).
3. Вычислительные модули выполняют расчет характеристик графа.
4. Модуль визуализации отображает граф со всеми его характеристиками и позво-

ляет просматривать параметры отдельных элементов графа.

На рис. 1 представлена высокоуровневая архитектура решения.

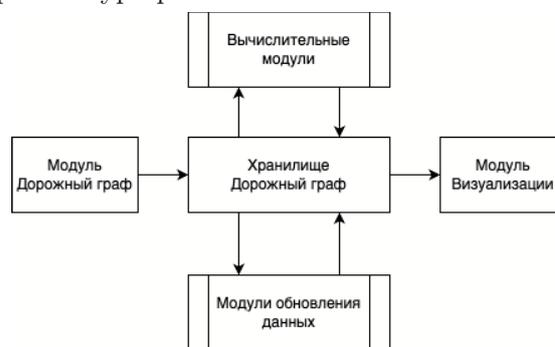


Рис. 1. Высокоуровневая архитектура системы  
Fig. 1. High-level system architecture

Стоит отметить, что визуализация возможна на любом из этапов работы модели, начиная от получения исходного графа и далее на каждом этапе работы конкретного модуля. Это позволяет итерационно оценивать результаты работы.

Модули 1–3 являются вычислительным ядром системы и написаны на языке программирования Python. Модуль 4 является веб-сервисом и написан на языке программирования JavaScript.

Для удобства работы в различных пакетах программ информация о графе хранится в формате GEXF. Существует множество библиотек для различных языков программирования, позволяющих обрабатывать данный формат. Кроме того, он позволяет хранить не только структуру графа, но и дополнительные данные о вершинах и ребрах.

## МОДУЛЬ ДОРОЖНОГО ГРАФА

Множество сервисов предоставляют актуальные и точные карты (Яндекс.Карты, 2GIS и др.), но их невозможно получить в виде набора данных для построения графа, т. к. это не предусмотрено лицензиями на использование сервиса, а применение автоматизированных сборщиков данных является нарушением пользовательского соглашения.

В то же время существуют проекты открытых данных, которые собирают и актуализируют информацию, в том числе о дорожных сетях. Одним из наиболее известных проектов является Open Street Map, некоммерческий проект, цель которого – создание географической карты с помощью интернет-пользователей.

В работе [6] представлен автоматизированный способ построения графа для дальнейшей работы с ним при помощи пакета NetworkX [9]. Разработанная авторами библиотека позволяет получить ориентированный граф для любого географического объекта с необходимым типом дорог.

В данной работе при построении модели транспортной сети города Петрозаводска построен граф, ребра которого соответствуют дорогам с возможностью движения автомобильного транспорта, вершины графа соответствуют перекресткам автомобильных дорог (листинг 1).

*Листинг 1.* Получение дорожного графа из OpenStreetMap

```
og = graph_from_place(
    query = 'Petrozavodsk, Russia',
    network_type='drive'
)
ogn = og.nodes(data=True)
oge = og.edges(data=True)
```

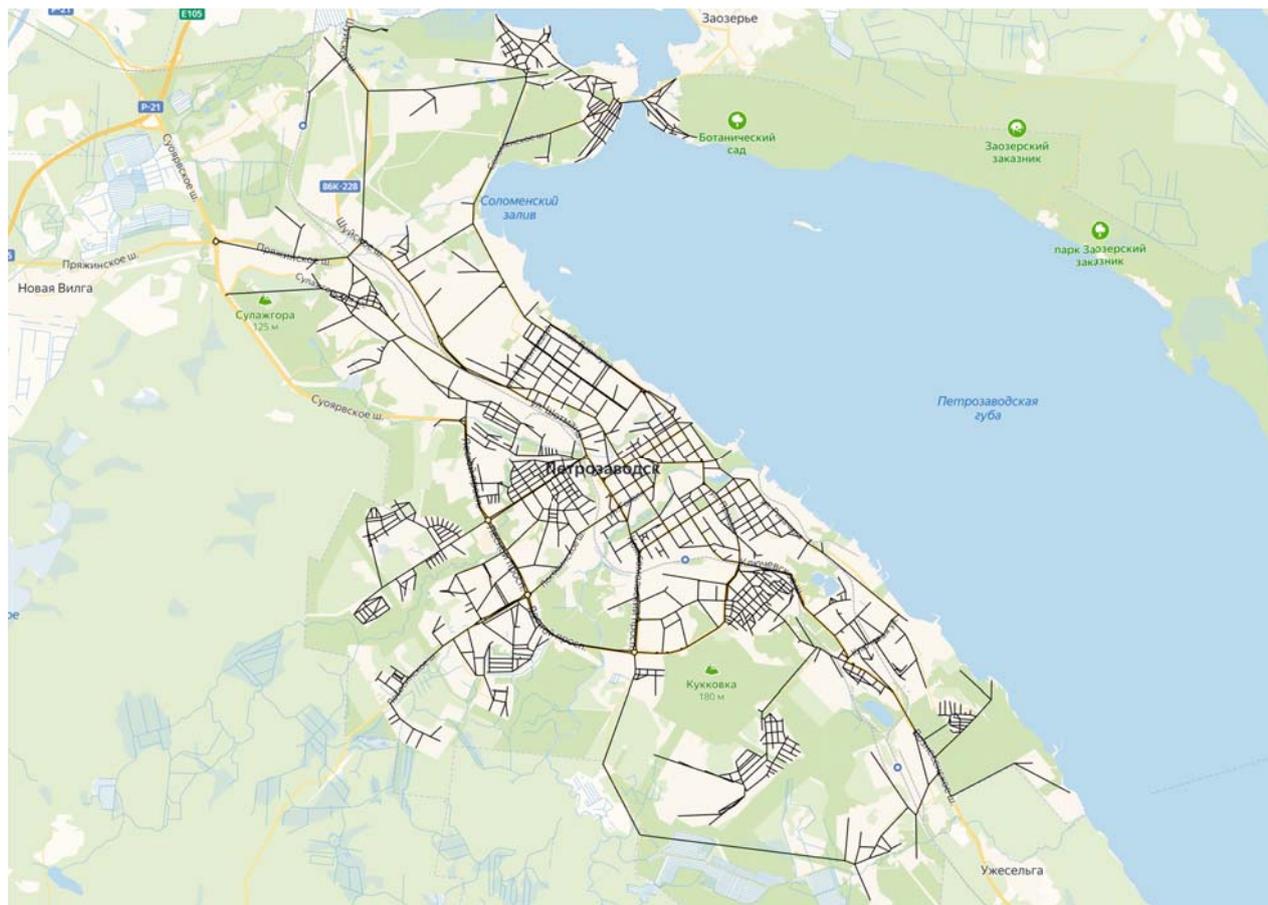
Поскольку дальнейшие вычисления предполагают работу с неориентированным гра-

фом, полученный ориентированный граф был преобразован с помощью объединения параллельных ребер. Данные о вершинах содержат географическую широту и долготу, что позволяет соотносить дополнительные данные, имеющие географическую привязку, с вершинами и ребрами графа (листинг 2).

*Листинг 2.* Преобразование графа к неориентированному виду

```
g = nx.Graph()
nodes = [(n[0], {
    'lon': n[1]['x'],
    'lat': n[1]['y'],
}) for n in ogn]
edges = [(e[0], e[1], {
    'id': f'{e[0]}_{e[1]}',
    'name': str(e[2].get('name', None)),
}) for e in oge]
```

Результат работы модуля дорожного графа представлен на рис. 2. Граф дорожной сети города Петрозаводска содержит 1531 вершину и 2081 ребро.



*Рис. 2.* Дорожный граф города Петрозаводска  
*Fig. 2.* Road graph of the city of Petrozavodsk

## Модули обновления данных

### Данные о жителях города

При моделировании транспортной системы критически важным является понимание распределения транспортных потоков между узлами транспортной сети. Для построения модели перемещения жителей по транспортной сети необходимо вычислить вектор исходящих коммуникаций, описывающий число жителей, маршруты которых начинаются в вершинах графа. Согласно переписи населения РФ за 2021 год [2], население Петрозаводского городского округа составляет 234 897 человек. Однако достоверной информации о распределении жителей по адресам проживания нет в открытом доступе.

На портале Реформа ЖКХ [1] представлена информация о домах Республики Карелия, которая раскрывается в рамках Постановления Правительства РФ от 23 сентября 2010 года № 731 «Об утверждении стандарта раскрытия информации организациями, осуществляющими деятельность в сфере управления многоквартирными домами». По каждому дому выгрузка содержит адрес в соответствии с ФИАС, а также технические характеристики здания. Если принять за допустимое приближение, что в среднем в жилом помещении проживает равное число человек, то такие данные можно использовать для приближенной оценки плотности населения в конкретной части города.

При обработке данных, представленных на портале Реформа ЖКХ (листинг 3), используются следующие поля: `id` (идентификатор здания), `city` (город), `street` (улица), `house` (номер дома), `building` (номер строения), `block` (корпус), `letter` (литера), `address` (строка адреса), `quarters_count` (число помещений), `living_quarters_count` (число жилых помещений).

*Листинг 3.* Обработка данных о количестве жилых помещений

```
CITY = 'Петрозаводск'

f = open(HOUSES_DATA, 'r', newline='')
r = csv.reader(f, delimiter=";")
header = next(r)

houses = filter(
    is_valid_house,
    map(lambda row: {
        'id': str(row[0]),
        'city': row[10],
        'street': row[12],
        'house': row[13],
```

```
        'building': row[14],
        'block': row[15],
        'letter': row[16],
        'address': row[17],
        'quarters_count': row[31],
        'living_quarters_count': row[32]
    },
    [row for row in r if row[10] == CITY])
```

Пусть  $q_i$  – число помещений в здании, общее число жилых помещений  $Q = \sum q_i$ , а общее число жителей в Петрозаводске равно  $C$ , тогда число жителей в помещении  $c_i$ :

$$c_i = \frac{C}{Q} \cdot q_i.$$

Для привязки данных о жителях к вершинам моделируемого графа координаты зданий, находящихся на территории Петрозаводского городского округа, были определены с помощью сервисов обратного геокодирования, где по текстовому адресу возвращаются координаты. Далее на основе кратчайшего расстояния каждое здание было привязано к вершине графа. Число жителей, проживающих в непосредственной близости от вершины графа, интерпретируется как вес вершины.

На рис. 3 представлена тепловая карта распределения жителей. Большой размер вершины соответствует большему весу.

### Данные об организациях

Для построения вектора входящих коммуникаций необходима информация о точках интереса – объектах, в которые направляются участники движения. В простейшем случае в качестве таких объектов могут выступать места работы и учебы жителей.

Сведения об организациях города могут быть получены из открытых данных, которые предоставляют налоговые службы. Но эти сведения не могут быть использованы в модели по следующим причинам:

1. Юридический адрес организации может не совпадать с физическим адресом офиса, предприятия, торговой точки.
2. Отсутствует информация об учреждениях образования и здравоохранения и других подобных организациях.
3. Большое количество данных об ИП, которые могут иметь адрес в жилых домах.
4. Неочевидно влияние организаций на дорожный трафик жителей города.

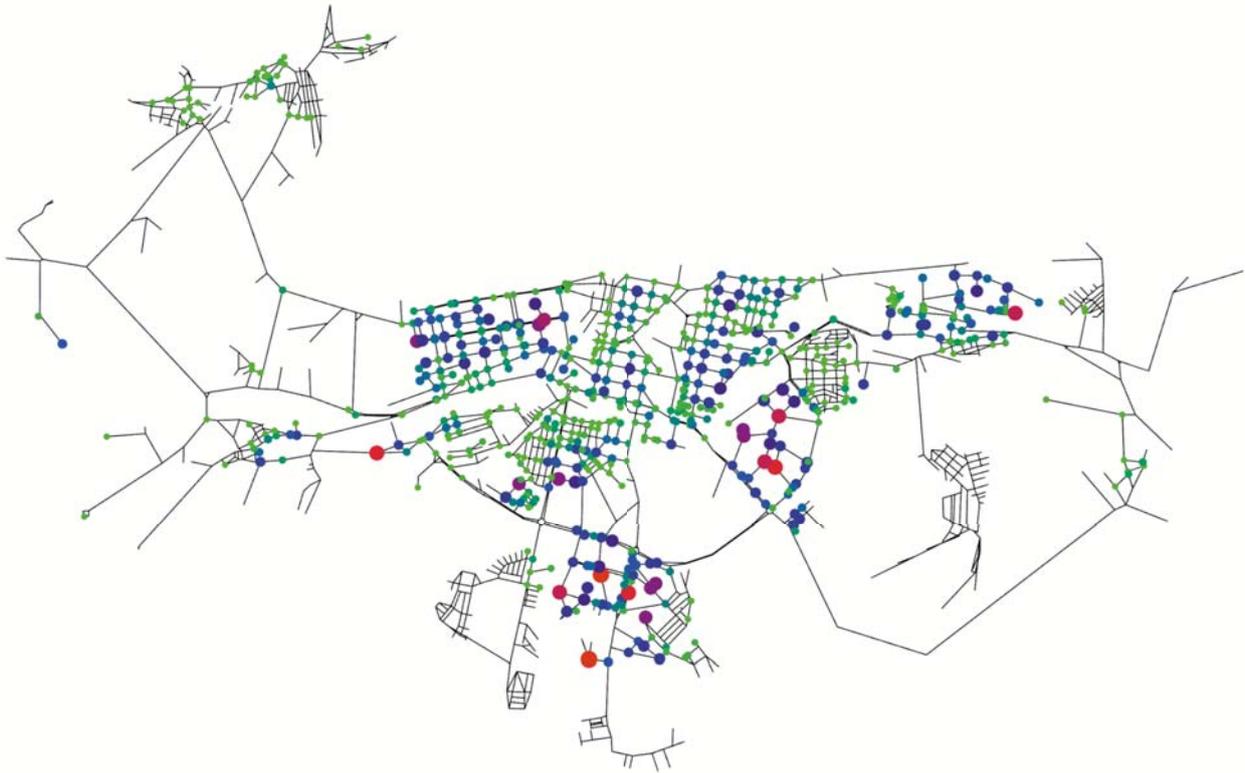


Рис. 3. Тепловая карта распределения жителей  
 Fig. 3. Heatmap of population distribution

В сервисе OpenStreetMap, который был использован ранее для построения дорожного графа, также присутствует информация об организациях. Она может быть неполной, т. к. заполняется исключительно силами сообщества, но имеет несколько важных параметров, которые позволяют включить ее в модель:

1. Для каждой организации на карте можно определить тип и подтип (например, тип медицина, подтип ветеринарная клиника). Это позволяет задать вес каждой организации в городе в соответствии с предполагаемым количеством людей, которые посещают данную организацию.
2. Имеются данные о географическом положении.
3. Возможно получение данных о более широком классе объектов, таких как заправки, почтовые отделения и прочие места, посещаемые жителями в течение дня.

Для получения этих данных использован сервис Overpass API [5], который позволяет выполнять запросы к OpenStreetMap при помощи языка запросов Overpass. Для модели использованы поля `name` (имя объекта), `tags`

(параметры объекта, включая его тип и подтип), а также географическая широта и долгота. Как и в случае с жилыми домами, на основании географических координат объектов была произведена привязка организаций к узлам графа.

Были выбраны все объекты, описание которых содержало такие значения, как, например, офис, образовательное учреждение, торговый центр и т. п. (листинг 4). Для каждого типа объектов был выбран вес, основанный как на приблизительном количестве сотрудников, обычно работающих в организациях такого типа, так и на привлекательности подобных организаций для посетителей. Так, количество сотрудников и поток посетителей в городскую поликлинику или школу будет превышать аналогичные показатели для торговой организации. Значение веса выбиралось из полуинтервала  $(0, 1]$ . Каждому узлу была поставлена в соответствие дополнительная характеристика «вес организаций», которая представляет собой взвешенную по типам сумму числа организаций, привязанных к данному узлу.

Для построения запроса к Overpass API в качестве области поиска выбран город Петрозаводск, выбраны все вершины, ребра и отно-

шения в этой области, которые имеют необходимые типы.

Листинг 4. Запрос к Overpass API для получения объектов необходимого типа

```
[out:json];
area[place=city]
[name = "Петрозаводск"]->.a;
nwr["{feature}"="{filter}"]
(area.a);
out center meta;
```

Пример структуры данных объекта «Федеральное государственное бюджетное учреждение науки Федеральный исследовательский центр «Карельский научный центр Российской академии наук», полученной в резуль-

тате выполнения запроса, приведен ниже (листинг 5).

Листинг 5. Пример ответа Overpass API

```
{
  "lat": 61.7911192,
  "lon": 34.3793148,
  "name": "КарНЦ_РАН",
  "feature_name": "amenity",
  "feature_value": "research_institute",
  "weight": "1"
},
```

На рис. 4 представлена тепловая карта распределения организаций по вершинам графа. Большой размер узла соответствует большому значению взвешенной суммы организаций.



Рис. 4. Тепловая карта распределения организаций  
Fig. 4. Organization distribution heatmap

### Дополнительные данные о дорогах

Разрабатываемая модель должна включать в себя некоторые характеристики дорог, такие как число полос, протяженность отрезка дороги, соответствующего ребру графа, максимально разрешенная скорость, время проезда по пустой дороге, а также пропускная способность отрезка дороги. На рис. 5 представлен пример вывода информации о ребре графа.

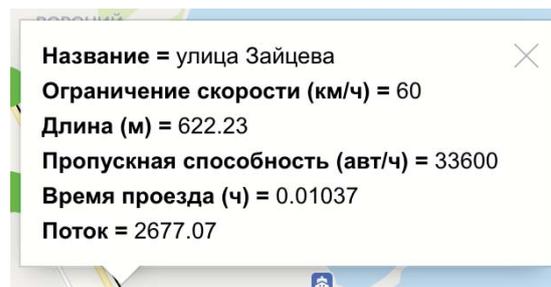


Рис. 5. Параметры ребра  
Fig. 5. Edge parameters

Физические характеристики дорог, а также максимально разрешенная скорость движения по участкам дорог были получены на первом этапе при построении графа с помощью OpenStreetMap. Информация о числе полос содержится в поле `lanes`, протяженность участков дороги – в поле `length`, для определения, является ли дорога односторонней, использовался флаг `oneway`.

Значение максимальной скорости может быть представлено как в числовом виде, так и в виде строк `RU:urban`, `RU:rural`, `RU:default`, которые для дорог Российской Федерации определяют скорость соответственно 60, 90 и 60. Ассоциативный массив `ms` представлен в листинге 6.

*Листинг 6.* Сопоставление строчного и числового представлений скорости

```
ms = {
    'RU:urban': 60,
    'RU:rural': 90,
    'RU:default': 60
}
```

Поле со значением `maxspeed` может содержать несколько значений, поэтому для определения числового значения скорости необходимо выполнить простое преобразование (листинг 7).

*Листинг 7.* Определение числового значения максимальной скорости

```
raw = e.get('maxspeed', 'RU:default')
speed_limit = 0
if(isinstance(raw, list)):
    for limit in raw:
        if(limit.isdigit()):
            speed_limit = int(limit)
            break
    else:
        speed_limit = ms.get(raw)
elif raw.isdigit():
    speed_limit = int(raw)
else:
    speed_limit = ms.get(raw)
```

Приоритетным считается значение скорости, заданное в числовом виде; если же оно не может быть получено, используется предопределенное значение, соответствующее полученной строке.

Время проезда по пустой дороге вычисляется как  $t_{min} = l/s_{max}$ , где  $l$  – это длина ребра, а  $s_{max}$  – максимально допустимая скорость на этом участке.

Для оценки пропускной способности дорог использованы методические рекомендации Федерального дорожного агентства (Росавтодор). Однако эти рекомендации учитывают

большое число параметров, таких как продольный уклон или радиус кривой в плане. Поскольку подобные данные недоступны, было принято решение упростить формулу для расчета, основываясь на доступных характеристиках дорог г. Петрозаводска. Пропускная способность на ребре графа вычислялась по формуле

$$P = nP_{max}f_v,$$

где  $n$  – число полос в одном направлении,  $P_{max}$  – величина максимальной практической пропускной способности, связанная с числом полос и наличием разделительной полосы,  $f_v$  – коэффициент, учитывающий ограничение скорости.

## Вычислительные модули

В архитектуру программного комплекса могут быть включены вычислительные модули, принимающие в качестве входных данных граф со всеми его характеристиками. В результате работы модуля граф обогащается дополнительными параметрами. В качестве примера можно привести модуль расчета меры центральности вершин графа, основанного на методе PageRank [10] (листинг 8). Для вычисления значений центральности вершин значения весов ребер выбираются равными величинам, обратным длинам соответствующих участков дорог.

*Листинг 8.* Пример реализации вычислительного модуля

```
for e in G.edges(data=True):
    e[2]['li'] = 1.0 / e[2]['length']
pr_w = nx.pagerank(g, weight='li')

nodes = g.nodes(data=True)
for (i, n) in enumerate(nodes):
    n[1]['pr'] = pr_w[i]
```

На рис. 6 представлено распределение значений центральности вершин графа дорожной сети города Петрозаводска. Большой размер вершины соответствует большему значению PageRank.

Поскольку программный комплекс имеет модульную структуру, возможно подключить любой вычислительный модуль, который будет выполнять необходимое преобразование графа и сохранять его в хранилище. Для визуализации полученных результатов достаточно добавить идентификатор нужного значения в список доступных отображений.



Рис. 6. Распределение значений центральности по методу PageRank  
 Fig. 6. Distribution of centrality values according to the PageRank method

### Модуль визуализации

Модуль визуализации позволяет отображать результаты работы программного комплекса и представляет собой веб-сервис, который выполняет функции отображения всех параметров графа с использованием наглядных цветовых схем. Кроме того, модуль предоставляет возможность настройки параметров отображения и выбора необходимого слоя (рис. 7).

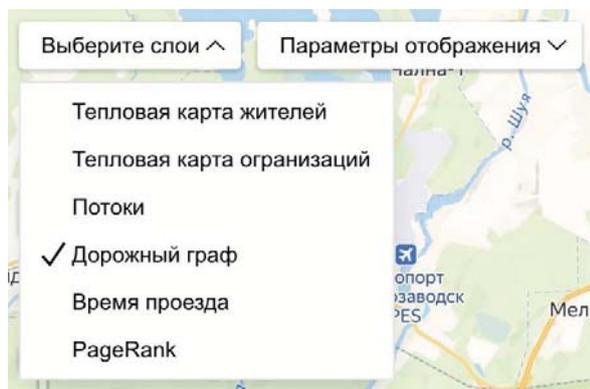


Рис. 7. Параметры отображения  
 Fig. 7. Mapping parameters

Также он позволяет просматривать значения параметров для отдельных ребер и вер-

шин дорожного графа. На рис. 8 приведен пример отображения параметров вершины, соответствующей перекрестку улицы Антикайнена и проспекта Ленина.

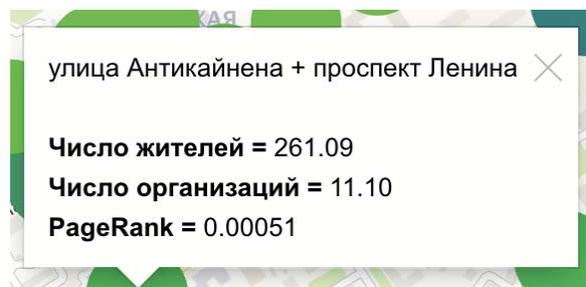


Рис. 8. Параметры вершины  
 Fig. 8. Vertex parameters

Модуль написан на языке программирования JavaScript с использованием библиотеки React [4]. В качестве основы использована библиотека API Яндекс Карт [3] сервиса Яндекс.Карты, с помощью которой реализован функционал по отображению ребер и вершин графа, их стилизации по цвету и размеру. Работа со структурой графа реализована при помощи библиотеки Graphology [8], которая в том числе позволяет загружать графы в формате GEXF.

Для визуализации вершин графа используется тепловая карта. Для каждого слоя задается параметр вершины, например, число организаций или жителей, и на основе его положения относительно максимального и минимального значения этого параметра формируется размер и цвет вершины. В листинге 9: *c* – подпись для всплывающего окна, *v* – значение параметра, *mv* – максимальное значение параметра.

*Листинг 9.* Построение элемента тепловой карты

```
createCircle(lat, lon, c, v, mv) {
  const color = gradientArray[
    Math.ceil((maxGrad / mv) * v)
  ];
  return new window.ymaps.Circle(
    [
      [lat, lon],
      minSize + (maxSize / mv) * v
    ],
    {
      balloonContentBody: c
    },
    {
      draggable: false,
      fillColor: color,
      strokeColor: "#000000",
      strokeWidth: 0,
    }
  );
}
```

Программно задаются градиент и разброс радиуса (листинг 10).

*Листинг 10.* Градиент и радиус элемента тепловой карты

```
const maxSize = 80;
const minSize = 50;
const maxGrad = 19.0;
const gradientArray = new Gradient()
  .setColorGradient(
    "#00FF00",
    "#0000FF",
    "#FF0000"
  )
  .setMidpoint(20)
  .getColors();
```

Аналогичный процесс определения цвета производится и для ребер, но ширина линии ребра всегда одинакова и задается программно.

Библиотека React в совокупности с использованием Яндекс.Карт позволяет заранее создавать объекты отображения (коллекции геообъектов) и отображать их при необходимости. Это позволяет избежать повторного создания объектов, даже при пересоздании самой карты. Компонент для отображения (листинг 11) является универсальным и может быть использован для любого параметра.

*Листинг 11.* Компонент для отображения тепловых карт

```
export const DistributionLayout = ({
  map,
  graph,
  parameter,
  enabled
}) => {
  const [dst, setDst] = useState(null);
  useEffect(() => {
    setDst(buildDst(graph, parameter));
  }, []);

  useEffect(() => {
    if (dst && enabled) {
      map.geoObjects.add(dst);
    } else {
      map.geoObjects.remove(dst);
    }
  }, [dst, map, enabled]);
  return null;
};
```

При использовании компонента (листинг 12) достаточно задать параметр вершины/ребра и передать флаг выбора слоя.

*Листинг 12.* Использование компонента для отображения тепловых карт

```
<DistributionLayout
  map={ymaps}
  graph={graph}
  parameter={"weight"}
  enabled={true}
/>
```

## ЗАКЛЮЧЕНИЕ

В работе приведено описание программного комплекса, позволяющего визуализировать данные о состоянии транспортной сети города Петрозаводска. Были обработаны сведения, представленные в различных открытых источниках, что позволило создать адекватную модель городской транспортной сети. Входные данные обрабатываются атомарно и итерационно, что обеспечивает консистентность модели.

На каждом этапе обновления данных возможна визуализация структуры графа с привязкой к карте местности и отображение количественных и качественных характеристик вершин и ребер графа.

Благодаря использованию формата хранения GEXF подключение новых вычислительных модулей и модулей обновления данных легко реализуемо.

Вэб-сервис визуализации графа транспортной сети города Петрозаводска доступен по адресу: <https://transport-graph-web.vercel.app/>.

## ЛИТЕРАТУРА

1. Национальный проект «Жилье и городская среда» [Электронный ресурс]. URL: <https://фрт.рф/new/reformagkh.ru/opendata?gid=2348078> (дата обращения: 02.02.2023).
2. Федеральная служба государственной статистики [Электронный ресурс]. URL: [https://rosstat.gov.ru/vpn\\_popul](https://rosstat.gov.ru/vpn_popul) (дата обращения: 02.02.2023).
3. API Яндекс Карты [Электронный ресурс]. URL: <https://yandex.ru/dev/maps/> (дата обращения: 02.02.2023).
4. React JavaScript-библиотека для создания пользовательских интерфейсов [Электронный ресурс]. URL: <https://ru.legacy.reactjs.org> (дата обращения: 02.02.2023).
5. A database engine to query the OpenStreetMap data [Электронный ресурс]. URL: [https://wiki.openstreetmap.org/wiki/Overpass\\_API](https://wiki.openstreetmap.org/wiki/Overpass_API) (дата обращения: 02.02.2023).
6. *Boeing G.* OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks // *Computers, Environment and Urban Systems*. 2017. Vol. 65. P. 126–139. doi: 10.1016/j.compenvurbsys.2017.05.004
7. *Ermolin N. A., Khitraya V. A., Khitryi A. V., Mazalov V. V., Nikitina N. N.* Modeling of the city's transport network using game-theoretic methods on the example of Petrozavodsk // *Contributions to Game Theory and Management*. 2022. Vol. 15. P. 18–31. doi: 10.21638/11701/spbu31.2022.02
8. Graphology [Электронный ресурс]. URL: <https://graphology.github.io/> (дата обращения: 02.02.2023).
9. *Hagberg A., Schult D., Swart P.* Exploring network structure, dynamics, and function using NetworkX // *Proceedings of the 7th Python in Science Conference (SciPy2008), Pasadena, CA USA*. 2008. P. 11–15.
10. *Page L., Brin S., Motwani R., Winograd T.* The pagerank citation ranking: Bringing order to the Web // *Proceedings of the 7th International World*

Wide Web Conference, Brisbane, Australia. 1998. P. 161–172.

## REFERENCES

1. National project ‘Housing and urban environment’. URL: <https://фрт.рф/new/reformagkh.ru/opendata?gid=2348078> (accessed: 02.02.2023). (In Russ.)
2. Federal State Statistics Service. URL: [https://rosstat.gov.ru/vpn\\_popul](https://rosstat.gov.ru/vpn_popul) (accessed: 02.02.2023). (In Russ.)
3. API Yandex Maps. URL: <https://yandex.ru/dev/maps/> (accessed: 02.02.2023). (In Russ.)
4. React JavaScript-library for creating user interfaces. URL: <https://ru.legacy.reactjs.org> (accessed: 02.02.2023). (In Russ.)
5. A database engine to query the OpenStreetMap data. URL: [https://wiki.openstreetmap.org/wiki/Overpass\\_API](https://wiki.openstreetmap.org/wiki/Overpass_API) (accessed: 02.02.2023).
6. *Boeing G.* OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*. 2017;65:126–139. doi: 10.1016/j.compenvurbsys.2017.05.004
7. *Ermolin N. A., Khitraya V. A., Khitryi A. V., Mazalov V. V., Nikitina N. N.* Modeling of the city's transport network using game-theoretic methods on the example of Petrozavodsk. *Contributions to Game Theory and Management*. 2022;15:18–31. doi: 10.21638/11701/spbu31.2022.02
8. Graphology. URL: <https://graphology.github.io/> (accessed: 02.02.2023).
9. *Hagberg A., Schult D., Swart P.* Exploring network structure, dynamics, and function using NetworkX. *Proceedings of the 7th Python in Science Conference (SciPy2008), Pasadena, CA USA*. 2008. P. 11–15.
10. *Page L., Brin S., Motwani R., Winograd T.* The pagerank citation ranking: Bringing order to the Web. *Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia*. 1998. P. 161–172.

Поступила в редакцию / received: 05.05.2023; принята к публикации / accepted: 25.05.2023.

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interest.

## СВЕДЕНИЯ ОБ АВТОРАХ:

**Хитрая Виталия Андреевна**  
аспирант

*e-mail: dobvitalia@yandex.ru*

**Хитрый Андрей Владимирович**  
аспирант

*e-mail: andrey.khitryy@gmail.com*

## CONTRIBUTORS:

**Khitraya, Vitalia**  
Graduate Student

**Khitryy, Andrey**  
Graduate Student