

УДК 004.421:004.67
DOI

CC BY 4.0

К ВОПРОСУ ВЫЧИСЛИТЕЛЬНОГО МОДЕЛИРОВАНИЯ МЕСТОРАСПОЛОЖЕНИЯ ТОРГОВЫХ ОБЪЕКТОВ

^{1,2}Калач А. В. ORCID ID 0000-0002-8926-3151,¹Чикунов С. В. ORCID ID 0000-0001-5365-8947,¹Никольников П. А. ORCID ID 0009-0007-5140-9622,²Лебедев Д. Н. ORCID ID 0009-0003-6490-042X¹Федеральное государственное бюджетное образовательное учреждение высшего образования
«Воронежский государственный университет инженерных технологий»,
Воронеж, Российская Федерация;²Федеральное казенное образовательное учреждение высшего образования
«Воронежский институт Федеральной службы исполнения наказаний»,
Воронеж, Российская Федерация, e-mail: a_kalach@mail.ru

В работе обсуждаются вычислительные аспекты моделирования месторасположения торговых объектов на онлайн-картах. Геолокационные данные важны при моделировании, так как они позволяют точно определить местоположение торговых точек объектов, их доступность для клиентов и другие важные параметры. Цель исследования – вычислительное моделирование месторасположения торговых объектов, позволяющее предоставить клиенту необходимые данные для анализа торговых точек конкурентов на основе их геолокационных данных в виде встраиваемого в приложения виджета. Исследования основаны на использовании методов математического моделирования и объектно-ориентированного программирования. В ходе исследования выполнено моделирование месторасположения объектов на карте и определение расстояний между ними с использованием геолокационных данных. Предложено описание модели поиска точек в заданном радиусе от исходной с использованием разработанного авторами алгоритма с применением ограничений для сужения области поиска решения, позволяющего уменьшить его вычислительную сложность. Описаны структура хранения данных и создание функций Oracle для эффективного нахождения объектов на карте. На тестовых примерах проведен анализ вычислительной сложности полученной алгоритмической модели. Выполнена программная реализация в виде виджета отображения торговых точек конкурентов на карте согласно заданным требованиям, отсутствующим у большинства программ-конкурентов, реализованы мероприятия для уменьшения времени загрузки найденных точек. Результаты вычислительных экспериментов демонстрируют эффективность предложенной модели по повышению производительности геопространственного анализа. Сделан вывод о том, что применение предложенных моделей позволит уменьшить время обработки данных и понизить нагрузку на систему. Клиенты смогут пользоваться результатами исследования в личном кабинете на сайте или интегрировать его в свою систему с помощью API.

Ключевые слова: торговые точки, онлайн-касса, геолокационные данные, географические координаты, алгоритм, карты, вычислительное моделирование

COMPUTATIONAL MODELING OF RETAIL FACILITY LOCATIONS

^{1,2}Kalach A. V. ORCID ID 0000-0002-8926-3151,¹Chikunov S. V. ORCID ID 0000-0001-5365-8947,¹Nikolnikov P. A. ORCID ID 0009-0007-5140-9622,²Lebedev D. N. ORCID ID 0009-0003-6490-042X¹Federal State Budget Educational Institution of Higher Education
“Voronezh State University of Engineering Technologies”, Voronezh, Russian Federation;²Federal State Educational Institution of Higher Education
“Voronezh Institute of the Federal Penitentiary Service”, Voronezh, Russian Federation,
e-mail: a_kalach@mail.ru

This paper discusses computational aspects of modeling the locations of retail facilities on online maps. Geolocation data plays a crucial role in modeling, as it allows for the precise determination of retail locations, their accessibility to customers, and other important parameters. The objective of the study is computational modeling of retail facility locations, enabling the provision of clients with the necessary data for analyzing competitors' retail outlets based on their geolocation data in the form of an embedded application widget. The research is based on mathematical modeling and object-oriented programming methods. The study involved modeling the locations of objects on the map and determining the distances between them using geolocation data. A description of a model for searching for points within a given radius from the initial point is proposed using an algorithm developed by the authors. This algorithm employs constraints to narrow the search space, reducing its computational complexity. The data storage structure and the creation of Oracle functions for efficiently finding objects on the map are described. Test examples are used to analyze the computational complexity of the resulting algorithmic model. A software implementation was developed in the form of a widget displaying competitors' retail outlets on a map according to specified requirements, which are not met by most competing programs. Measures were implemented to reduce the loading time of the identified points. The results of computational experiments demonstrate the effectiveness of the proposed model in improving the performance of geospatial analysis. The conclusion is that the use of the proposed models will reduce data processing time and decrease the system load. Clients will be able to access the research results in their personal account on the website or integrate it into their system using the API.

Keywords: point of sale, online cash register, geolocation data, geographic coordinates, algorithm, maps, computational modeling

Введение

В современном мире конкуренция на рынке розничной торговли становится все более интенсивной. Для успешного ведения бизнеса компаниям необходимо не только привлекать новых клиентов, но и эффективно анализировать деятельность своих конкурентов. Одним из ключевых инструментов в этом процессе является анализ торговых точек конкурентов, который позволяет выявить их сильные и слабые стороны, а также определить потенциальные возможности для собственного развития [1].

Геолокационные данные играют важную роль в этом анализе, так как позволяют точно определить местоположение торговых точек, их доступность для клиентов и другие важные параметры. Однако сбор и обработка таких данных требуют использования специализированных информационных систем, которые могут эффективно обрабатывать большие объемы информации и предоставлять их для дальнейшего анализа [2–4].

В 2016 г. обязанность применять при расчетах в торговле онлайн-кассы охватила практически всех предпринимателей [5]. Для фискализации платежей бизнесу необходимо подключение к оператору фискальных данных (ОФД) – компании, которая принимает информацию с онлайн-касс, передает сведения о продажах в федеральную налоговую службу (ФНС) [6]. При регистрации/перерегистрации кассы в ФНС ей присваивается уникальный идентификатор ФИАС ID, необходимый для идентификации конкретного адреса в базе данных Федеральной информационной адресной системы (ФИАС), и заполняются координаты здания, в котором она находится. В итоге получают множество торговых точек, каждая из которых имеет свои геокоординаты, хранящиеся в базе данных.

Личный кабинет клиента (ЛКК) и партнера на сайте ОФД оснащены уникальным функционалом, позволяющим создавать нужные шаблоны всевозможных данных по статистике продаж в автоматическом режиме. Все чеки кассы проходят через ОФД, из этого складывается возможность на основе данных предоставлять различную аналитику продаж клиенту. Также в ЛКК реализована логика сравнения торговых точек клиента на основе их выручки. Адрес установки подтягивается автоматически из последнего отчета с данными о регистрации/перерегистрации кассы.

В настоящее время для анализа торговых точек конкурентов можно использовать известные приложения-аналоги, на-

пример 2GIS, Wikimapia, OpenStreetMap, Яндекс Карты и др. [7, 8]. Однако следует отметить, что они не обладают достаточной функциональностью, необходимой для анализа торговых точек конкурентов по следующим критериям: возможность просмотра своих торговых точек, разделение на свои и чужие, информация об удаленности точки, просмотр конкурентов из другой области бизнеса, частота актуализации информации, поиск конкурентов в определенном радиусе.

Цель исследования – вычислительное моделирование месторасположения торговых объектов, позволяющее предоставить клиенту необходимые данные для анализа торговых точек конкурентов на основе их геолокационных данных в виде встраиваемого в приложения виджета.

Для достижения цели требуется решить следующие задачи:

- 1) сформулировать требования к разрабатываемой модели и алгоритмам;
- 2) разработать математическую и алгоритмическую модели поиска точек на карте в заданном радиусе от исходной, провести их тестирование;
- 3) разработать программную реализацию модели в виде клиент-ориентированного виджета.

Материалы и методы исследования

Реализация функционала виджета «Mars» – отображение и сравнение торговых точек конкурентов по коду общероссийского классификатора видов экономической деятельности (ОКВЭД) и удаленности от торговых точек клиента в ЛКК на сайте осуществлялась по результатам вычислительного моделирования месторасположения торговых объектов на основе их географических координат. Разработка такого геоаналитического виджета требует реализации клиент-серверной архитектуры с агрегацией данных на бэкенде [9]. Для обеспечения скорости загрузки не более 15 с, все вычисления, привязку к кодам ОКВЭД и фильтрацию следует проводить на стороне сервера.

При реализации данной архитектурно-технической структуры исходили из следующих требований к виджету: 1) возможность выбора кода ОКВЭД для поиска торговых точек конкурентов; 2) возможность скрытия отображения своих торговых точек; 3) возможность выбора радиуса отображения точек конкурентов, совпадающих коду ОКВЭД, заданному в поиске; 4) время отображения данных в разделе должно занимать не более 15 с.

Использование в виджете алгоритма поиска точек в заданном радиусе от исходной с применением предлагаемых в работе ограничений позволит уменьшить время обработки данных и понизить нагрузку на систему.

Исследования основаны на использовании методов математического моделирования и объектно-ориентированного программирования.

Моделирование месторасположения объектов на карте и определение расстояний между ними. Задача определения расстояний между объектами на земной поверхности по географическим координатам (широта и долгота) имеет огромное значение в различных областях человеческой деятельности. Однако особенности формы Земли, используемые на практике методы расчета расстояний, каждый из которых обладает своими преимуществами и недостатками, уменьшение длины градуса долготы по направлению к полюсам не позволяют

точно определять расстояния между объектами [10–12].

Исходя из физической формы Земли при выполнении расчетов полагали, что расстояние между двумя параллелями, отличающимися на 1° по широте, равно 111,11 км, а длина 1° долготы уменьшается к полюсам в зависимости от широты и составляет

$$L_n = L_\varphi \cdot \cos(\varphi),$$

где L_n – длина дуги 1° параллели, L_φ – длина дуги 1° экватора, равная 111,3 км, φ – географическая широта точки [13, с. 25].

Моделирование процедуры поиска точек объектов от исходной в заданном радиусе. Задача моделирования заключается в том, чтобы из имеющегося множества точек объектов, имеющих географические координаты, отобрать все те, которые попадают в заданный радиус R от исходной. Тогда расстояние между двумя точками (φ_i, λ_i) и (φ_j, λ_j) можно рассчитать по следующему выражению:

$$L = R_{\text{ЗЕМ}} \cdot \cos^{-1}(\cos \varphi_i \cdot \cos \varphi_j \cdot \cos(\lambda_j - \lambda_i) + \sin \varphi_i \cdot \sin \varphi_j), \quad (*)$$

где (φ_i, λ_i) – географические координаты (широта, долгота) точки объекта A , (φ_j, λ_j) – географические координаты точки объекта B , $R_{\text{ЗЕМ}}$ – средний радиус Земли, равный 6371 км. Вычислительная сложность модели определяется используемым алгоритмом отбора точек из имеющегося множества точек, находящихся на земной поверхности.

Классический алгоритм полного перебора обеспечивает поиск всех вариантов решений за приемлемое время при относительно небольших размерах множества искомым точек. Однако при увеличении количества торговых объектов, вычислительная сложность данного алгоритма возрастает, что делает его малоприменимым для больших наборов данных [14]. В связи с этим для сужения множества проверяемых точек и уменьшения вычислительной сложности принято решение использовать алгоритм, в основе которого лежит перебор элементов множества и их проверка на соответствие определенным ограничениям.

Алгоритм с применением ограничений

Полагали, что долготу (λ_2) точки, расположенной на земной поверхности на расстоянии S от заданной, можно определить как

$$\lambda_2 = \lambda_1 + \Delta\lambda, \quad \Delta\lambda = \frac{S \cdot 180^\circ}{R_{\text{ЗЕМ}} \cdot \cos(\varphi_1) \cdot \pi},$$

где (φ_1, λ_1) – широта и долгота заданной точки, $\Delta\lambda$ – изменение долготы, S – расстояние от заданной точки.

Все точки, расположенные в пределах круга радиуса $R_{\text{кр}}$ от заданной точки (φ_n, λ_n) , попадают в прямоугольную область, ограниченную по широте $\varphi_n \pm \Delta\varphi$ и долготы $\lambda_n \pm \Delta\lambda$. При этом $R_{\text{кр}} = S_n$, а $\Delta\varphi$ зависит только от значения радиуса $R_{\text{кр}}$. Такой подход позволяет сократить объем проверок, поскольку достаточно рассмотреть только точки внутри области, ограниченной внешним квадратом со стороной $2 \cdot R_{\text{кр}}$, в который вписан данный круг.

Внутри этого круга можно выделить вписанный в него квадрат, длина стороны которого равна $\sqrt{2} \cdot R_{\text{кр}}$. Следовательно, все точки, принадлежащие внутреннему квадрату, попадают в область, ограниченную

$$\text{по широте } \varphi_n \pm \Delta\varphi / \sqrt{2}$$

$$\text{и долготы } \lambda_n \pm \Delta\lambda / \sqrt{2}.$$

Рассмотрим особенности функционирования алгоритма.

1. Ввод исходных данных. Вводятся координаты заданной точки (φ_i, λ_i) , определяется множество точек, из которых будет осуществлен отбор, задается радиус поиска $R_{\text{кр}}$.

2. Задание внешних ограничений в виде квадрата. Точки, находящиеся за его границами, исключаются из рассмотрения.

3. Задание внутренних ограничений в виде квадрата. Точки, находящиеся внутри его границ, будут отобраны без проверки.

4. Из множества оставшихся точек, расположенных между внешним и внутренним

квадратами, выбираются координаты очередной непроверенной точки (φ_j, λ_j) .

5. С помощью выражения (*) вычисляется расстояние между этими точками.

6. Проверяется условие попадания очередной точки в область с радиусом $R_{кр}$ ($L \leq R_{кр}$).

7. В случае выполнения условия очередная точка (φ_j, λ_j) выбирается для отображения.

8. Повторение этапов 4–7 до тех пор, пока множество точек не станет пустым.

Таким образом, для каждой исходной точки можно сразу определить подмножество точек, удовлетворяющих условию $L \leq R_{кр}$, и не тратить время и вычислительные ресурсы на проверку неподходящих, что приводит к уменьшению количества итераций этапов 4–7 алгоритма. В итоге необходимо лишь проверить небольшую часть площади внешнего квадрата на присутствие требуемых точек, применяя формулу (*).

Разработанная модель предоставляет механизм для поиска всех точек объектов из множества возможных, находящихся в пределах заданного радиуса от начальной точки.

Результаты исследования и их обсуждение

Реализация алгоритма с применением ограничений. Программные эксперименты по исследованию предлагаемой модели проведены в СУБД Oracle с использованием средств интегрированной среды PL/SQL Developer. Так как в Oracle отсутствует специализированный тип данных для хранения градусов, информация о них представлена в числовом формате (geo_lat, geo_lon), совпадающем с форматом хранения данных в БД ФИАС.

Для корректного выполнения тригонометрических вычислений числовые данные переводятся в радианы с использованием функции RADIANS:

```
CREATE OR REPLACE FUNCTION RADIANS (p_degree IN NUMBER)
RETURN NUMBER AS
BEGIN
RETURN p_degree /(180/ACOS(-1));
END;
```

Поиск расстояния L между двумя точками с использованием выражения (*) реализован функцией GET_DISTANCE_KM:

```
CREATE OR REPLACE FUNCTION GET_DISTANCE_KM(
in_lat1 IN NUMBER,
in_lon1 IN NUMBER,
in_lat2 IN NUMBER,
in_lon2 IN NUMBER)
RETURN NUMBER
AS
v_distance NUMBER;
BEGIN
v_distance:= 6371 * --радиус земли
ACOS(
COS(RADIANS(in_lat1)) * COS(RADIANS(in_lat2)) *
COS(RADIANS(in_lon2) - RADIANS(in_lon1)) +
SIN(RADIANS(in_lat1)) * SIN(RADIANS(in_lat2)));
RETURN v_distance;
END;
```

Пример реализации алгоритма для поиска точек в радиусе 3 км от заданной:

```
With cte as (
Select geo_lat, geo_lon,
      geo_lat - 3/√2/111.3 niz_do_3 ,
      geo_lat + 3/√2/111.3 verx_do_3 ,
      geo_lon - 3/√2/6371*(180/ACOS(-1))/cos(RADIANS(ts.geo_lat)) levo_do_3 ,
      geo_lon + 3/√2/6371*(180/ACOS(-1))/cos(RADIANS(ts.geo_lat)) pravo_do_3 ,
      geo_lat - 3/111.3 niz_ot_3 ,
      geo_lat + 3/111.3 verx_ot_3 ,
      geo_lon - 3/6371*(180/ACOS(-1))/cos(RADIANS(ts.geo_lat)) levo_ot_3 ,
      geo_lon + 3/6371*(180/ACOS(-1))/cos(RADIANS(ts.geo_lat)) pravo_ot_3
from shops)
```

Эти условия задают ограничения, используемые для выделения области на карте, расположенной между внутренними и внешними границами, обозначенными квадратами.

Полученные ограничения служат основой для определения критериев формирования ограниченной выборки точек из множества оставшихся вариантов (функция GET_DISTANCE_KM):

```
Select tc.geo_lat, tc.geo_lon
From shops tc, Cte c
where tc.geo_lat between c.niz_do_3 and c.verx_do_3
and tc.geo_lon between c.levo_do_3 and c.pravo_do_3
and not (tc.geo_lat between c.niz_ot_3
and c.verx_ot_3
and tc.geo_lon between c.levo_ot_3
and c.pravo_ot_3)
and GET_DISTANCE_KM(tc.geo_lat, tc.geo_lon,
c.geo_lat, c.geo_lon) < 3;
```

A-Time	Buffers	OMem	lMem
00:01:54.29	191K		
00:01:54.29	191K	3954K	1407K
00:01:54.29	191K		
00:00:00.01	0		
00:00:00.01	0	2048	2048
00:00:00.01	0		
00:00:00.01	0		
00:00:00.01	0		
00:00:00.01	0		
00:00:00.01	0		
00:01:54.29	191K		

а)

A-Time	Buffers	OMem	lMem
00:00:02.85	95540		
00:00:02.85	95540	906K	906K
00:00:02.26	95540		
00:00:00.01	0	2048	2048
00:00:00.01	0		
00:00:00.01	0		
00:00:00.01	0		
00:00:00.01	0		
00:00:00.01	0		
00:00:00.01	0		
00:00:02.26	95540		
00:00:00.87	95540	22M	1723K
00:00:00.52	95540		

б)

Рис. 1. Нагрузка при выполнении: а) алгоритма полного перебора, б) алгоритма с ограничениями
Примечание: составлен авторами по результатам данного исследования

Таким образом, изменяя значение радиуса поиска $R_{кр}$ в рассмотренном примере, можно найти географические координаты всех точек в любом радиусе от исходной точки с минимальными затратами вычислительных ресурсов [15].

Определение вычислительной сложности алгоритма с применением ограничений.

На одном и том же наборе данных из БД одновременно были запущены два запроса: с использованием алгоритма полного перебора и алгоритма с применением ограничений. Оба алгоритма выдали одинаковые результаты в ответе. После их завершения были получены планы работы запросов с соответствующей нагрузкой (рис. 1).

Карта торговых точек

Ваши торговые точки

Список торговых точек вашей компании.

Выбрано: Разработка ком... (62)

Радиус до 3 км.

Конкуренты, расположенные в пределах трёх километров от ваших торговых точек.

Радиус от 3 до 10 км.

Конкуренты, находящиеся на расстоянии от трёх до десяти километров от ваших торговых точек.

Радиус свыше 10 км.

Конкуренты, расположенные дальше десяти километров от ваших торговых точек.

Рис. 2. Параметры отображения торговых точек
Примечание: составлен авторами по результатам данного исследования

Выбрано: Торговля ро... (47)

Поиск

☒	Торговля розничная, кроме торговли автотранспортными средствами и мотоциклами	(47)
☐	Растениеводство и животноводство, охота и предоставление соответствующих услуг в этих областях	(01)
☐	Лесоводство и лесозаготовки	(02)
☐	Рыболовство и рыбоводство	(03)
☐	Добыча угля	(05)
☐	Добыча сырой нефти и природного газа	(06)
☐	Добыча металлических руд	(07)

Рис. 3. Выбор кода ОКВЭД
Примечание: составлен авторами по результатам данного исследования

Время выполнения запроса (A-Time) алгоритмом полного перебора составило почти 2 мин. Время выполнения алгоритмом с ограничениями составило 3 с. Так же на основе

полученной экспериментальным путем статистики используемых вычислительных ресурсов (Buffers, OMem, IMem) видно, что во втором случае сокращается нагрузка на БД.

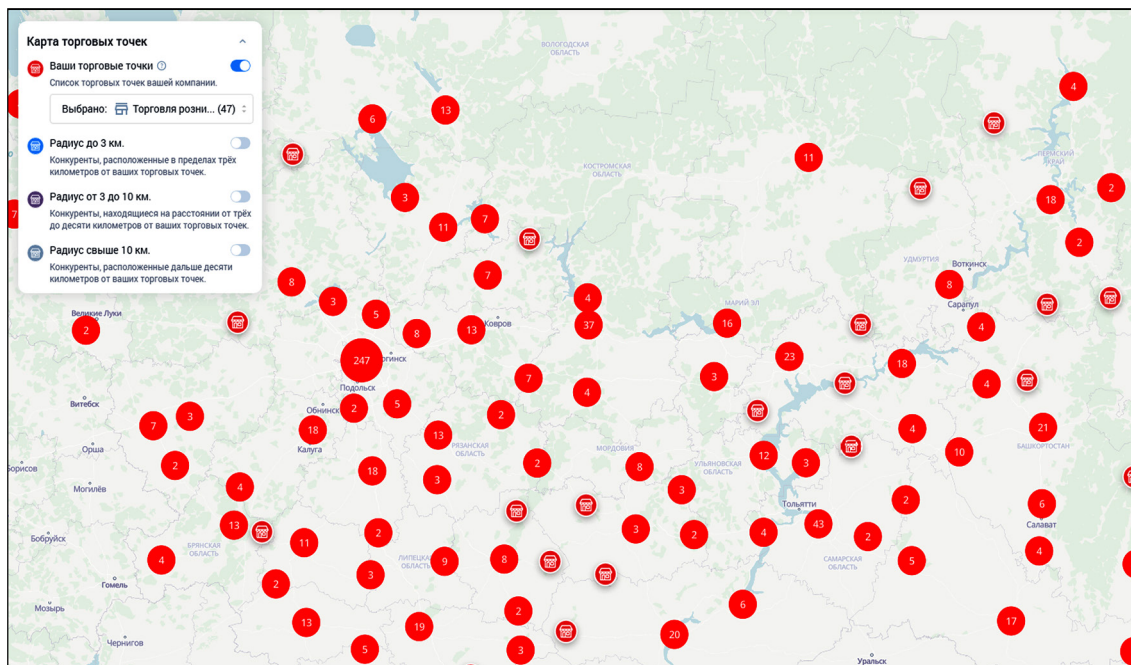


Рис. 4. Торговые точки клиента на карте

Примечание: составлен авторами по результатам данного исследования

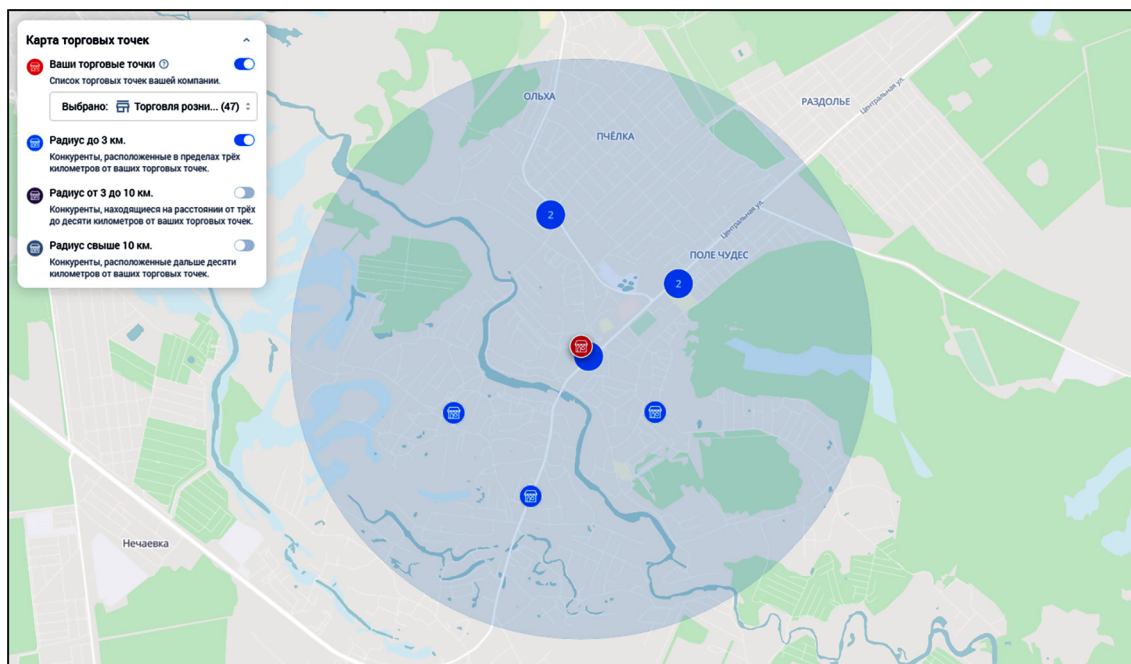


Рис. 5. Отрисовка торговых точек конкурентов в радиусе 3 км

Примечание: составлен авторами по результатам данного исследования

SQL_ID bthvb30cnc08u, Plan hash value: 269327563							

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.00	0.00	0	0	1	0
Execute	1	0.09	0.10	18	2458	3	0
Fetch	1	8.96	39.39	90088	100756	0	1

total	3	9.06	39.50	90106	103214	4	1

Рис. 6. Фрагмент трассировочного файла

Примечание: составлен авторами по результатам данного исследования

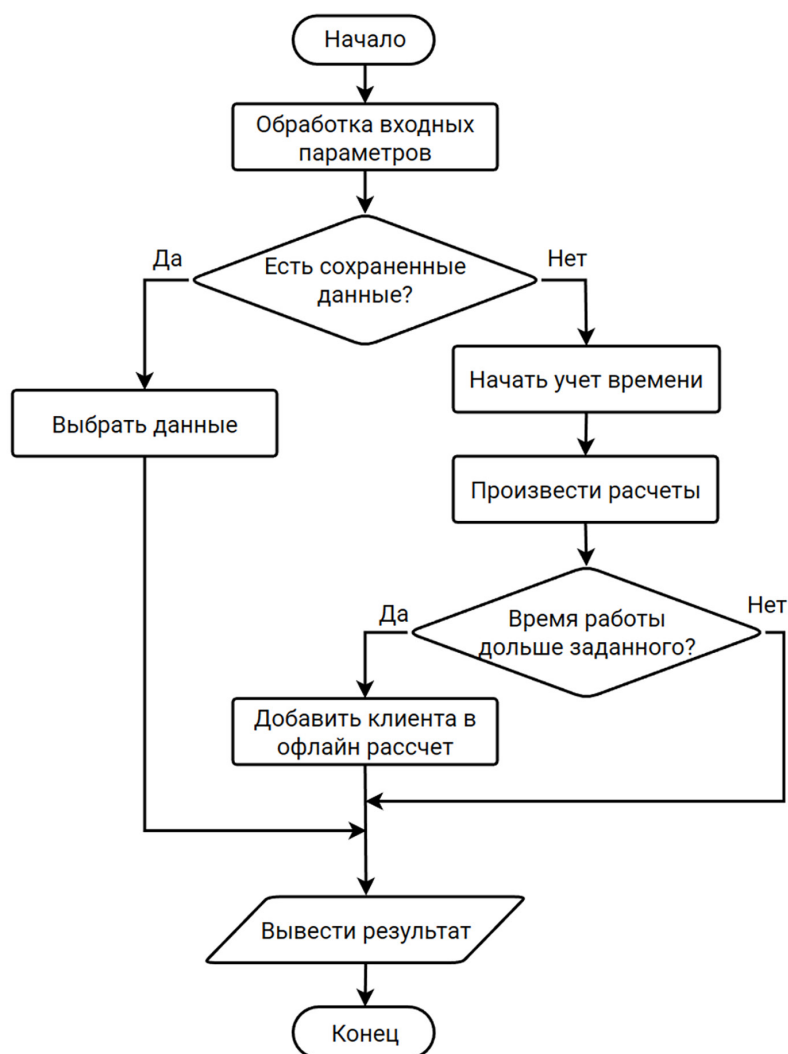


Рис. 7. Схема алгоритма процедуры поиска точек

Примечание: составлен авторами по результатам данного исследования

Программная реализация модели в виде виджета. Для доступа к карте с главной страницы ЛКК сайта был реализован виджет «Maps», содержащий информацию о количестве зарегистрированных касс

клиента в каждом федеральном округе. Режим отображения точек имеет 4 различных параметра (рис. 2): 1) торговые точки клиента; 2) конкуренты, расположенные в пределах 3 км от торговых точек клиента;

3) конкуренты, расположенные в пределах от 3 до 10 км от торговых точек клиента; 4) конкуренты, расположенные в пределах от 10 до 20 км от торговых точек клиента. Необходимый код ОКВЭД выбирается из выпадающего списка (рис. 3) и определяется через связь в системе «онлайн-касса → ИНН → основной код ОКВЭД». Следует отметить, что поиск по коду ОКВЭД происходит по верхнему уровню.

После загрузки на карте отображаются торговые точки клиента (рис. 4). Отрисовка конкурентов в радиусе 3 км от торговой точки клиента изображена на рис. 5. При уменьшении масштаба точки группируются для удобства отображения их на карте. При поиске в радиусе 10 и 20 км круги визуально не выделяются. При выборе своей точки отобразится адрес, юрлицо, бренд, описание кода ОКВЭД и количество касс в данной точке. В торговых центрах, сдающих свои площади в аренду, адреса установки торговых точек разных предпринимателей совпадают. Для таких случаев предусмотрена логика объединения торговых точек в список, точки группируются по ФИАС ID.

Тестирование и оптимизация. Уменьшение времени загрузки карты. При открытии виджета спустя продолжительное время отображаются точки на карте. У клиентов с малым количеством касс загрузка происходит за пару секунд. На клиентах с числом касс близким к тысяче время ожидания занимает более 30 с. Если ожидание ответа превышает 60 с, сервер автоматически прервет выполнение метода и данные не отображаются.

На рис. 6 приведен фрагмент трассировочного файла, в котором виден SQL_ID запроса с самым долгим временем выполнения (elapsed total = 39.5).

Анализ плана запроса показал, что процедура Oracle, отвечающая за поиск точек для отрисовки на карте виджета, выполняется долго из-за большого числа исходных точек клиента, рядом с которыми происходит поиск других точек. Чем больше исходных точек, тем больше время выполнения процедуры.

Поскольку известно, что кассы редко меняют свое местоположение и проходят регистрацию, данные для ответа виджета меняются нечасто. Вследствие этого предлагается рассчитывать и обновлять их в фоновом режиме и после при необходимости подгружать их.

Таким образом, сокращается время ответа процедуры и, следовательно, время загрузки карты. На рис. 7 представлена схема алгоритма, реализующего данную процедуру после проведения оптимизации.

Необходимо отметить, что после доработки алгоритма время работы процедуры сократилось почти в 13 раз и не превысило 3 с, что отвечает изначально заданным требованиям к скорости отображения данных в виджете.

Заключение

Таким образом, разработаны математическая и алгоритмическая модели, позволяющие предоставить клиенту данные для анализа торговых точек конкурентов на основе их геоданных. Для поиска торговых точек конкурентов предложены и реализованы алгоритм поиска с применением предлагаемых в работе ограничений и позволяющий найти торговые точки конкурентов в любом радиусе от исходной точки за наименьшее время и с минимальными затратами вычислительных ресурсов; алгоритм отображения торговых точек конкурентов на карте, позволяющий сократить время их загрузки; структура хранения данных и функции СУБД Oracle для эффективного нахождения торговых объектов на карте.

Созданный на основе полученных моделей программный продукт в виде клиент-ориентированного виджета позволяет предоставить информацию для проведения анализа торговых точек конкурентов на основе геолокационных данных, информации из базы ОКВЭД и удаленности от торговых точек клиента. Пользователи могут использовать виджет в личном кабинете на сайте или интегрировать его в свою систему с помощью API.

Основные преимущества разработанного программного продукта заключаются в возможности выбора кода ОКВЭД для поиска торговых точек конкурентов; адаптивном выборе радиуса отображения точек конкурентов, совпадающих по коду ОКВЭД, заданному в поиске; возможности получения результатов поиска за меньшее время (время отображения данных на карте не превышает 15 с).

Список литературы

1. Артемова Д. А. Современные подходы к повышению конкурентоспособности организации // Вестник Московского Международного Университета. 2025. № 2 (6). С. 9–12. EDN: GUDOSH.
2. Лезина Т. А., Рябчук К. Г. Обзор подходов к определению местоположения объектов продуктового ритейла // Экономика и бизнес: теория и практика. 2019. № 6–2. С. 23–26. URL: <https://www.elibrary.ru/item.asp?id=38577836> (дата обращения: 26.05.2026). DOI: 10.24411/2411-0450-2019-10851.
3. Головин О. К., Игонина А. А. Оптимизация схемы размещения торговых объектов на основе математико-картографической модели // Модели, системы, сети в экономике, технике, природе и обществе. 2021. № 2 (38). С. 115–129. DOI: 10.21685/2227-8486-2021-2-8. EDN: WTULAZ.

4. Алчинов А. И., Гороховский И. Н., Акифьева Е. В. Состояние и тенденции развития географических информационных систем // Проблемы управления. 2024. № 2. С. 3–22. URL: <https://www.elibrary.ru/item.asp?id=67239026> (дата обращения: 26.05.2026). DOI: 10.25728/pu.2024.2.1.
5. Адаменко А. А., Хорольская Т. Е., Маркелов В. В. Актуальные вопросы перехода на онлайн-кассы российскими предпринимателями // Естественно-гуманитарные исследования. 2018. № 21 (3). С. 70–77. URL: <https://elibrary.ru/item.asp?id=36497958> (дата обращения: 26.05.2026). EDN: YOYQMX.
6. Щасная К. Ю., Сергеева И. И. Основные принципы и технологии функционирования онлайн-касс // Экономическая среда. 2017. № 2 (20). С. 23–26. EDN: ZAGWNX.
7. Кузнецов М. А., Кравченя П. Д., Скринкович В. С., Володченков Л. В. Использование геоинформационных систем для анализа привлекательности торговых точек города // Прикаспийский журнал: управление и высокие технологии. 2021. № 3 (55). С. 62–71. EDN: ESDRIG.
8. Пестун М. В. Методы построения навигационных описаний маршрутов для картографических компьютерных систем: дис. ... канд. физ.-мат. наук. Москва, 2015. 115 с. [Электронный ресурс]. URL: <http://library.keldysh.ru/diss.asp?id=2015-pestun> (дата обращения: 26.05.2026).
9. Поленов М. Ю., Иванов Д. А. Модифицированная распределенная архитектура обработки данных для геоинформационных систем // Известия ЮФУ. Технические науки. 2020. № 6 (216). С. 99–108. URL: <https://www.elibrary.ru/item.asp?id=45602902> (дата обращения: 26.05.2026). DOI: 10.18522/2311-3103-2020-6-99-108.
10. Буряковский П. К. Подход к определению кратчайшего расстояния между объектами разнородной геопространственной геометрии // Системная инженерия и информационные технологии. 2023. Т. 5. № 5 (14). С. 88–94. URL: <https://elibrary.ru/item.asp?id=58807244> (дата обращения: 26.05.2026). DOI: 10.54708/2658-5014-SIIT-2023-no5-p88.
11. Смолев А. М., Михеева Т. И., Клепиков Н. М. Адресное позиционирование и его реализация в геоинформационной системе // ИТ & Транспорт. Сборник статей. Самара: ООО «Научно-производственный центр «Интеллектуальные транспортные системы», 2024. С. 4–10. EDN: VAPXAS.
12. Ботнев В. А., Устинов С. М. Методика определения расстояния от точки до отрезка в задачах навигации // Научно-технические ведомости СПбГПУ. Информатика, телекоммуникации и управление. 2019. Т. 12. № 2. С. 68–79. URL: <https://infocom.spbstu.ru/userfiles/files/articles/2019/2/68-79.pdf> (дата обращения: 26.05.2026). DOI: 10.18721/JCSTCS.12206.
13. Сяо Н. Алгоритмы ГИС / Пер. с англ. А. А. Слинкина. М.: ДМК Пресс, 2021. 328 с. URL: <https://ibooks.ru/bookshelf/388451/reading> (дата обращения: 26.05.2026). ISBN 978-5-97060-908-8.
14. Никольников П. А. Алгоритм поиска точек на карте в определенном радиусе от заданной точки на основе геолокации // Парадигма. 2025. № 11–4. С. 73–76. EDN: LSGNGI.
15. Никольников П. А. Реализация алгоритма поиска точек в определенном радиусе от заданной точки в Oracle // Парадигма. 2025. № 11–4. С. 69–72. EDN: NWSPXR.

Конфликт интересов: Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest: The authors declare that there is no conflict of interest.

Финансирование: Авторы заявляют об отсутствии внешнего финансирования.

Financing: The research was performed without external funding.