(19) Евразийское патентное ведомство

(21) 202192708 (13) A2

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОЙ ЗАЯВКЕ

(43) Дата публикации заявки
2022.01.31

(22) Дата подачи заявки
2019.02.14

(51) Int. Cl. ***G06F 16/29*** (2019.01)
G06Q 50/16 (2012.01)
G06Q 50/14 (2012.01)
G06Q 50/30 (2012.01)

(54) СПОСОБ И ИНСТРУМЕНТ ПОИСКА ИЛИ СРАВНЕНИЯ ТОЧЕК С ИСПОЛЬЗОВАНИЕМ МАРШРУТОВ ИЛИ ДЛИН МАРШРУТОВ МЕЖДУ ТОЧКАМИ И МЕСТАМИ В ТРАНСПОРТНОЙ СИСТЕМЕ

(31) **62/632,419; 62/758,710; 62/780,268; 62/800,428; 16/274,242**
(32) **2018.02.20; 2018.11.12; 2018.12.16; 2019.02.02; 2019.02.13**
(33) **US**
(62) **202091986; 2019.02.14**

(71)(72) Заявитель и изобретатель:
МАЛЕВИЧ ГЖЕГОЖ (PL)

(74) Представитель:
Шехтман Е.Л. (RU)

(57) Варианты реализации изобретения связаны с поиском или сравнением точек. Одним из вариантов реализации может быть способ поиска или сравнения объектов недвижимости по времени маршрутов до места работы и обратно. Указанный способ эффективно обрабатывает данные об общественном транспорте и объектах недвижимости, вычисляя время в пути между объектами недвижимости и остановками транспорта. Полученные значения сохраняются. Добавляется платформа для отправки запросов, которая позволяет быстро обрабатывать большое количество запросов на поиск или сравнение. При обработке запросов указанный способ определяет части маршрутов перемещения, которые привязаны к какому-либо объекту недвижимости. Так как значения времен для этих частей были рассчитаны предварительно и сохранены, указанный способ может определить время перемещения по маршрутам для каждого объекта недвижимости с возможностью масштабирования. В результате указанный способ быстро отвечает на запросы, отправляемые по рынку недвижимости в крупнейших городских агломерациях, существующих на сегодняшний день. К другим вариантам реализации относятся поиск или сравнение по финансовым затратам, передвижение на личном автомобиле, а также другие точки, отличающиеся от объектов недвижимости. Также указанный способ предусматривает использование компьютерной системы и компьютерного сервиса.

202192708 A2

202192708 A2

СПОСОБ И ИНСТРУМЕНТ ПОИСКА ИЛИ СРАВНЕНИЯ ТОЧЕК С ИСПОЛЬЗОВАНИЕМ МАРШРУТОВ ИЛИ ДЛИНАМ МАРШРУТОВ МЕЖДУ ТОЧКАМИ И МЕСТАМИ В ТРАНСПОРТНОЙ СИСТЕМЕ

ПЕРЕКРЕСТНЫЕ ССЫЛКИ НА РОДСТВЕННЫЕ ЗАЯВКИ

[**001**] Настоящая заявка основана на следующих заявках и объявляет следующие приоритеты изобретений:

[Страна]	[Номер заявки]	[Дата подачи]
США	62/632,419	20 февраля 2018 г.
США	62/758,710	12 ноября 2018 г.
США	62/780,268	16 декабря 2018 г.
США	62/800,428	2 февраля 2019 г.
США	16274242	13 февраля 2019 г.,

включенных в состав настоящей заявки в виде ссылки.

ПРЕДПОСЫЛКИ СОЗДАНИЯ ИЗОБРЕТЕНИЯ

[**002**] Данное изобретение связано с поиском или сравнением (searching or comparing) точек. Традиционно, цель поиска заключается в нахождении какой-либо точки (site) из ряда возможных вариантов, отвечающей задачам оптимизации, а именно минимизации длины маршрута (route length), исходя из определенной необходимости в перемещениях между точкой и местом, а также учитывая желаемые характеристики искомой точки. Например, при поиске объектов недвижимости, зная необходимые конечные точки пути до места работы и обратно (destinations of commutes) и характеристики объекта недвижимости (real estate property), цель может заключаться в составлении списка объектов недвижимости с нужными характеристиками и кратчайшим временем в пути до места работы и обратно. Другой целью может быть сравнение каких-либо объектов недвижимости по времени нахождения в пути до места работы и обратно.

КРАТКОЕ ИЗЛОЖЕНИЕ СУЩНОСТИ ИЗОБРЕТЕНИЯ

[**003**] Варианты реализации изобретения включают в себя способ (method) поиска или сравнения точек, компьютерную систему, которая использует и реализует этот способ, а также компьютерный сервис, который получает (receives) запросы по поиску и сравнению от пользователей и возвращает им ответы (responds) в виде информации (information) о точках и маршрутах.

[**004**] В зависимости от варианта реализации данного изобретения, предоставляется

способ поиска или сравнения точек по маршрутам или длинам маршрутов. В способе используется комплексная предварительная обработка для предварительного расчета (precompute) и сохранения в базе данных маршрутов или длин маршрутов между каждой точкой и представителями (representatives) в границах транспортной системы. Способ предлагает базовую платформу (framework) для поиска или сравнения точек. При получении запроса, содержащего спецификацию маршрута, по базе данных находятся предварительно рассчитанные данные, которые позволяют быстро вычислить маршрут или длину маршрута по каждой точке. Поиск или сравнение точек может производиться по маршрутам или длинам маршрутов.

[**005**] В зависимости от варианта реализации данного изобретения, предусматривается компьютерная система для поиска или сравнения точек по маршрутам или длинам маршрутов. Система представляет собой аппаратно-программный комплекс. Комплекс получает данные о транспортной системе и точках от одного или множества поставщиков данных (data providers). Система строит графы (graphs), моделирующие перемещение между точками и представителями в границах транспортной системы. Система рассчитывает пути графов (graph paths) и сохраняет плечо графов или длины плеча графов. Это позволяет быстро рассчитывать маршруты или длины маршрутов для каждой точки при получении запроса, а также производить поиск или сравнение точек по маршрутам или длинам маршрутов.

[**006**] В зависимости от варианта реализации данного изобретения, предусматривается компьютерный сервис по поиску или сравнению точек по маршрутам или длинам маршрутов. Сервис позволяет пользователю задать параметры поискового или сравнительного запроса через Интерфейс пользователя на каком-либо устройстве, например, с помощью смартфона. Запрос содержит спецификацию маршрута и условие для фильтра (filtering condition). В ответ сервис выдает точки, отвечающие условию фильтра, вместе с маршрутами или длинами маршрутов для соответствующих точек, либо сервис проводит сравнение точек по маршрутам или длинам маршрутов.

[**007**] Быстрота расчета длины маршрута для каждой точки имеет основополагающее значение. Мы составили математическое доказательство. Рассмотрим некоторый способ поиска или сравнения ***M***. Мы можем придумать такой пользовательский запрос (adversarial request), чтобы ***M*** выдал какой-либо список точек в заданном порядке. У пользователя (adversary) есть два механизма: (1) подать запрос, содержащий спецификацию маршрута, по которой будет создан упорядоченный список точек по длине маршрута, порядок задается пользователем, и (2) подать запрос с установкой условия для фильтра точек, которое будет соответствовать подмножеству точек, выбранному пользователем. Таким образом, ***M***

должен возвратить упорядоченный список точек, произвольно выбранных пользователем в момент подачи запроса. Подробности выведения доказательства выходят за рамки заявки на оформление патента.

[**008**] Способ, компьютерная система и компьютерный сервис, все вместе выполняют задачи, не являющиеся унифицированными и не в полной мере реализованные в известных технических решениях. Ранее известные технические решения включают: патент US 8,417,409 B2; продолжение патента US 8,738,286 B2; подраздел US 8,756,014 B2; KR 10-1692501 B1; продолжение патентов PCT/KR2016/01083; WO 2017/065431 A1; US 2018/0232824 A1; и KR 10-1905593 B1.

[**009**] Представленные здесь варианты реализации изобретения служат для иллюстрации; они не несут исчерпывающий характер. Люди с базовыми техническими знаниями хорошо представят себе многочисленные модификации и исполнения, не нарушающие общей структуры и сущности реализации изобретения.

[**010**] В данной презентации термины «первый», «второй», «данный», «указанный» и подобные им не используются в узко специализированном смысле, а служат только для различия, если иное однозначно не следует из контекста. Термин в единственном числе включает формы множественного числа, если иное однозначно не следует из контекста. Термины «обладающий», «включающий», «состоящий» и подобные им показывают наличие компонентов или характеристик, при этом не ограничивая возможность наличия или добавления других компонентов или характеристик.

КРАТКОЕ ОПИСАНИЕ СХЕМ

[**011**] Схемы, включенные в раскрытие сущности изобретения, служат в качестве иллюстрации характеристик и преимуществ тех или иных вариантов реализации изобретения:

- РИС. 1: показывает пример цветового оформления времени в пути по маршрутам до места работы и обратно, в зависимости от варианта реализации изобретения, подпись: «Пример цветового оформления времени в пути по маршрутам до места работы и обратно для объектов недвижимости с двумя маршрутами до места работы: объект недвижимости → geo.0.0 → geo.0.1 → объект недвижимости и объект недвижимости → geo.1.0 → объект недвижимости. Мы использовали фрагмент карты на ядре Google, вместо него может быть использован любой другой.»;
- РИС. 2: показывает пример принципиальной схемы предварительной обработки и сохранения данных в зависимости от варианта реализации изобретения, подпись: «Пример принципиальной схемы предварительной обработки и сохранения

данных.»;

- РИС. 3: показывает пример принципиальной схемы ответа на запрос с использованием сохраненных предварительно обработанных данных в зависимости от варианта реализации изобретения, подпись: «Пример принципиальной схемы ответа на запрос с использованием сохраненных предварительно обработанных данных.»;
- РИС. 4: показывает пример расчета времени в пути до места работы и обратно в зависимости от варианта реализации изобретения, подпись: «Пример расчета времени в пути до места работы и обратно.»;
- РИС. 5: показывает пример участков на графе системы общественного транспорта для предварительного расчета кратчайшего плеча графа, в зависимости от варианта реализации изобретения, подпись: «Пример участков на графе системы общественного транспорта для предварительного расчета кратчайшего плеча графа.»;
- РИС. 6: показывает пример табличного / векторного хранилища (vector storage) предварительно рассчитанных значений кратчайшего времени в пути в зависимости от варианта реализации изобретения, подпись: «Пример табличного / векторного хранилища предварительно рассчитанных значений кратчайшего времени в пути»;
- РИС. 7: показывает пример разложения маршрута до места работы и обратно в зависимости от варианта реализации изобретения, подпись: «Пример маршрута до места работы и обратно $H \rightarrow W_1 \rightarrow W_2 \rightarrow W_3 \rightarrow W_4 \rightarrow H$, разложенного (decomposed) на часть, привязанную к домам (зеленым цветом) и на часть, не привязанную к домам, (черным цветом), на примере трех домов в качестве мест проживания H_1, H_2 и H_3.»;
- РИС. 8: показывает пример расчета кратчайшего времени в пути из точки начала маршрута до места работы и обратно, в зависимости от варианта реализации изобретения, подпись: «Пример расчета кратчайшего времени в пути из точки начала маршрута до места работы и обратно.»;
- РИС. 9: показывает пример расчета кратчайшего времени в пути из конечной точки маршрута до места работы и обратно, в зависимости от варианта реализации изобретения, подпись: «Пример расчета кратчайшего времени в пути из конечной точки маршрута до места работы и обратно.»;
- РИС. 10: показывает пример языка описания алгоритмов (pseudocode) расчета

кратчайшего времени в пути из начальной точки маршрута до места работы и обратно в зависимости от варианта реализации изобретения, подпись: «Пример языка описания алгоритмов расчета кратчайшего времени в пути из начальной точки маршрута до места работы и обратно PathFromHomeDurations (→ W).»;

- РИС. 11: показывает пример принципиальной схемы компьютерной системы в зависимости от варианта реализации изобретения, подпись: «Пример принципиальной схемы компьютерной системы для предварительного расчета и хранения данных в базе данных, а также обработки запросов с использованием данных, извлеченных из базы данных.»;
- РИС. 12: показывает пример пользовательского запроса и ответа от компьютерного сервиса на смартфоне пользователя в зависимости от варианта реализации изобретения, подпись: «Пример пользовательского запроса и ответа от компьютерного сервиса на смартфоне пользователя.».

[**012**] Схемы приведены исключительно для иллюстрации. Пояснения к изобретению могут быть приведены на других схемах, не нарушая принципы изобретения, в чем легко сможет разобраться любой человек с базовыми техническими знаниями.

ПОДРОБНОЕ ОПИСАНИЕ СУЩНОСТИ ИЗОБРЕТЕНИЯ

4 Подробное описание

[**013**] Изобретение затрагивает общий случай поиска или сравнения произвольных точек для получения их оптимизированных вариантов по желанию пользователя на основе маршрутов или длин маршрутов (routes or route lengths) до произвольных мест. При этом, с целью упрощения презентации, мы демонстрируем изобретение в первую очередь на примере конкретных точек (sites), которыми являются объекты недвижимости, конкретных мест (places), которыми являются места работы, а также конкретной цели оптимизации, а именно минимизации времени в пути от объектов недвижимости до места работы и обратно. Иллюстративный пример не является исчерпывающим. В последних разделах нами был описан принцип работы этого способа в общем случае.

4.1 Объекты недвижимости и маршруты до места работы и обратно

[**014**] Поиск жилья это сложная задача. Люди тратят значительные временные и материальные ресурсы на его поиск. Наша технология призвана помочь людям в поиске жилья. Существует несколько интернет-сервисов, так называемых агрегаторов предложений объектов недвижимости, которые позволяют людям искать недвижимость прямо в браузере компьютера или приложении смартфона по определенным характеристикам, например, по цене, географическому расположению, количеству комнат

и так далее. Результатом поиска становится некоторый список вариантов, которые клиент, как правило, смотрит лично непосредственно на месте.

[**015**] Расположение (location), пожалуй, является самым важным критерием для выбора объекта недвижимости, не зря агентства недвижимости используют такую характеристику, как «хороший район», «удобное расположение», «развитая инфраструктура» и т.д. Наше изобретение строится вокруг этой характеристики.

[**016**] Клиентам нужно будет добираться до места работы, школы или в другие места. Время в пути по этим маршрутам определяет ценность того или иного объекта недвижимости для каждого конкретного клиента. Кроме того, для всех клиентов в крупных городах существует общая заинтересованность в экономии времени поездок, которое можно было бы освободить для более продуктивной деятельности, так и с точки зрения экономии сил на преодоление интенсивного городского трафика. Наше изобретение помогает удовлетворить как общие, так и персональные потребности.

[**017**] Для простого примера, представим семью из двух человек, проживающую в столице Южной Кореи. Один из членов семьи работает государственным служащим в городской администрации, а другой трудится библиотекарем в Главной библиотеке Национального Университета города Сеул. Сейчас они живут в двухкомнатной квартире площадью 69 квадратных метров с одним санузлом, необходимый депозит за которую составляет 350 000 000 южнокорейских вон. Квартира имеет координаты по широте и долготе (37.5333, 127.0746). Суммарное время их ежедневных поездок до места работы и обратно составляет 1 час 41 минуту (около 25 минут в один конец, в среднем). Но оказывается, что они могут уменьшить время на перемещения до 1 часа 16 минут (сэкономив 25 минут), если переедут в другую квартиру с теми же параметрами, но расположенную в координатах (37.5041, 126.8888). Кратчайшее время в пути для обеих квартир составляет 50 минут. Но новая квартира имеет другую цену и размер. (Иллюстрация от 19 февраля 2018 г.)

[**018**] В чем состоит сложность такой оптимизации для всех? Довольно примитивно рассматривать отдельно каждый вариант объекта недвижимости, представленного на рынке, отвечающего требованиям семьи по размеру и другим характеристикам, и, зная места работы, рассчитывать время в пути, запрашивая информацию через существующие интернет-сервисы построения маршрутов. Такой подход все же не пригоден для большого потока запросов. Одной из проблем является большое количество представленных на рынке объектов недвижимости современного крупного города. Другой проблемой является большое количество семей/пользователей, которые потенциально нуждаются в оптимизации. Даже если мы допустим возможность недорогого использования сервисов по

построению маршрутов, квадратичный характер проблемы по-прежнему будет означать удорожание по общему объему запросов в такие сервисы.

[**019**] Каким образом можно удовлетворить потребность в оптимизации для всех? Наше изобретение является решением. В изобретении имеются следующие компоненты:

1. Изобретение определяет модель маршрута до места работы и обратно (commute path). Данная модель является универсальной, за счет включения широкого набора реально существующих маршрутов до разных мест, например, сначала до школы, потом до репетитора по пианино, а затем обратно домой. Полезность нашей модели повышается благодаря тому, что мы можем быстро находить объекты недвижимости, для которых минимизируется время в пути.
2. Изобретение предлагает способ оптимизации для быстрого расчета времени в пути. Изобретение определяет части какого-либо маршрута до места работы и обратно, привязанные к какому-либо объекту недвижимости. Время в пути для данных частей рассчитывается заранее и сохраняется в базу. В результате, когда будет нужно найти маршрут до места работы и обратно, изобретение сможет быстро выстроить (assemble) временные отрезки, получая время в пути *для каждого* объекта недвижимости.
3. Вариантом реализации изобретения является компьютерный сервис с доступом через интернет. Сервис дает возможность 25 миллионам жителей столичной агломерации Сеула искать и сравнивать объекты недвижимости по времени в пути.

4.2 Описание способа

[**020**] Мы используем термин перемещение (travel) в широком смысле, который включает в себя движение объектов или данных. Описание перемещения (description of travel) это некое описание, которое может дать человек с базовыми техническими знаниями. Вот некоторые из примеров описания перемещения: (1) «эй, друг, тебе нужно пройти один квартал на север, затем взять чуть левее» или (2) «5 долларов». Мы можем использовать термин маршрут перемещения (travel path), когда имеется в виду название перемещения. Длина перемещения (length of travel) это числовое значение (numeric value), которое человек с базовыми техническими знаниями сможет привязать к перемещению, например, в качестве расстояния, стоимости в деньгах и т.д. В качестве другого примера, мы можем использовать термин время в пути (travel duration), когда имеется в виду длина перемещения, выраженная во времени. Длина перемещения сама по себе является описанием перемещения. Описание перемещения: не может включать любую длину перемещения, а может включать только определенную длину перемещения, либо также некоторые другие данные.

[**021**] Мы продемонстрировали некоторые возможности способа и ввели терминологию,

которая будет использоваться в дальнейшем. Способ может рассчитывать время в пути по маршрутам до места работы и обратно для каждого объекта недвижимости. Это показано на РИС. 1. Городская территория разделена на цветные квадраты. Цветом обозначено, как долго требуется добираться на общественном транспорте от определенного объекта недвижимости в каждом конкретном районе: зеленым цветом показан короткий маршрут, желтым более длинный, а красным – самый долгий маршрут. Маршрут до места работы и обратно начинается от *объекта недвижимости*, затем идет до определенных мест *geo,...*, и, наконец, снова возвращается в исходный *объект недвижимости*. В какой-то степени, *объект недвижимости* является свободной переменной (free variable), в то время как *geo, . . .* это фиксированные переменные. На РИС.1 представлено два маршрута до места работы и обратно: незамкнутый маршрут вида *объект недвижимости*→ *geo.*0.0 →*geo.*0.1 → *объект недвижимости* дважды в неделю, и маршрут в прямом и обратном направлении вида *объект недвижимости* → *geo.*1.0 →*объект недвижимости* три раза в неделю. Как видно, с учетом этих двух маршрутов до места работы и обратно, а также их регулярности, объекты недвижимости с небольшой взвешенной суммой значений времени в пути образуют своеобразную «кляксу» (темно-зеленого цвета), что неудивительно, учитывая густоту сети общественного транспорта.

[**022**] На высшем уровне способ состоит из двух частей. В первой части рассчитывается время в пути между объектами недвижимости и представителями (representatives), которые представляют собой остановки в транспортной системе (transportation system). Данное время в пути сохраняется (stored) в базе данных (database), и его можно брать в готовом виде при поступлении запроса (request). См. РИС. 2 для иллюстрации. Во второй части происходит обработка запросов. Запрос содержит маршрут до места работы и обратно, а также желаемые характеристики (features) объекта недвижимости. При получении (received) запроса, соответствующее время в пути берется из базы данных и используется вместе с другими данными для получения общего времени в пути по маршруту до места работы и обратно для каждого объекта недвижимости с выбранными характеристиками. См. РИС. 3 для иллюстрации. Подробности и варианты данного описания приведены в следующих пунктах.

4.3 Маршруты до места работы и обратно

[**023**] Наш способ обрабатывает обширную выборку маршрутов до места работы и обратно, используемых людьми в границах городской агломерации. Маршрут до места работы и обратно начинается от точки *H*, которую мы назовем дом (home). *H* имеет произвольное расположение. Она может быть представлена любым объектом недвижимости, например, квартирой, арендуемой комнатой, коттеджем с собственным

участком, крупным предприятием, отелем и т.д. Также она может являться и точкой, где работает клиент, например, рестораном, магазином и т.д. Но по договоренности о наименованиях, в раскрытии сущности изобретения мы, как правило, используем термин «дом»; такая договоренность не предполагает ограничений. В одном из вариантов реализации клиент перемещается в разные места, а затем возвращается в точку *H*.

[**024**] В одном из вариантов реализации, маршрут перемещения укладывается в рамки одного дня, например, клиент выходит из *H* утром и возвращается в *H* вечером того же дня. В другом варианте реализации маршрут укладывается на пару дней, например, если клиент работает в ночную смену или его рабочая смена составляет 25 часов. В одном из вариантов реализации какое-либо перемещение может начинаться в определенное время, либо заканчиваться в определенное время, например, в 8:12. В другом варианте реализации, какое-либо перемещение может начинаться или заканчиваться в пределах некоторого интервала, например, «утром».

[**025**] Проще говоря, клиент перемещается из *H* до места *W*, которое мы назовем работа (work). *W* имеет произвольное расположение. Сюда же относится школа, дом бабушки с дедушкой, площадка для игры в гольф по выходным, любимый ресторан, кабинет врача, культовое учреждение и т.д. Это также может быть и местом, где человек живет. Но по договоренности о наименованиях, в раскрытии сущности изобретения мы, как правило, используем термин «работа»; такая договоренность не предполагает ограничений. Затем клиент возвращается из *W* в *H*. Мы называем эти два перемещения *«маршрут в прямом и обратном направлении»*. См. РИС. 4A для иллюстрации.

[**026**] *Незамкнутый маршрут до места работы и обратно* представляет собой пример более сложного маршрута до места работы и обратно. См. РИС. 4B для иллюстрации. При таком маршруте клиент перемещается из *H* до места W_1. После чего клиент перемещается из места W_1 в другое место W_2. Наконец, клиент перемещается из места W_2 обратно в точку *H*. Можно привести один из примеров, когда вы идете в школу, а после школы идете к репетитору по пианино.

[**027**] В целом, наш способ допускает любые произвольные перемещения. На РИС. 4C показана иллюстрация маршрута до места работы и обратно с пропуском перемещения от W_4 до W_2 и повторением перемещения от W_2 до W_3. На РИС. 4D показана иллюстрация незамкнутых маршрутов до места работы и обратно: маршрут до места работы и обратно, выходящий из *H*, но без возврата в *H*; и маршрут до места работы и обратно, с возвратом в *H*, без предварительного выхода из *H*. На РИС. 4E показана иллюстрация «несвязанного» маршрута до места работы и обратно. Маршрут до места работы и обратно может начинаться из одного дома, а заканчиваться в другом доме. Наш способ определяет

маршрут до места работы и обратно следующим образом.

Определение 1 *Маршрут до места работы и обратно это совокупность перемещений* $W_2 \rightarrow W_3$, $W_4 \rightarrow W_5$, . . . , $W_{k-2} \rightarrow W_{k-1}$, *при* $k \geq 2$, *который считается целым числом, включая* $H_{первый} \rightarrow W_1$ *или* $W_k \rightarrow H_{последний}$ *(таким образом, маршрут до места работы и обратно всегда содержит, по крайней мере, один дом и, по крайней мере, одно место работы), осуществляемых в произвольные моменты времени.*

[**028**] Маршрут до места работы и обратно можно представить в виде спецификации (specification) маршрута в границах транспортной системы. Различные W и H в Определении 1 указывают направление для перемещения клиента.

[**029**] В одном из вариантов реализации мы рассматриваем более простой маршрут до места работы и обратно $H \rightarrow W_1 \rightarrow W_2 \rightarrow . . . \rightarrow W_k \rightarrow H$, при каком-либо $k \geq 1$, конечные точки перемещения по этому маршруту будут пересекаться, а дом в начале и в конце маршрута один и тот же. В раскрытии сущности изобретения мы используем, как правило, более простую форму, поскольку она является самой частой на практике и упрощает понимание нашей презентации. При этом любому человеку, обладающему базовыми техническими знаниями, очевидно, что наш способ применяется к нашему (основному) определению маршрута до места работы и обратно.

[**030**] Наш способ находит кратчайшее время в пути по любому из маршрутов до места работы и обратно. Само по себе это уже было отражено в различных известных технических решениях. Но наш способ находит время в пути для всех домов за чрезвычайно малое время.

4.4 Предварительная обработка транспортной системы

[**031**] Способ проводит предварительную обработку данных о системе общественного транспорта, чтобы предварительно рассчитать (precompute) и сохранить кратчайшее время в пути для всех домов.

4.4.1 Расчет кратчайшего времени перемещения

[**032**] Мы описываем способ эффективного расчета кратчайшего времени перемещения между всеми домами и всеми расположениями остановок общественного транспорта. Как правило, в раскрытии сущности изобретения мы используем термин остановки (stopstations) для остановок общественного транспорта, которые включают остановки автобусов, станции метро, либо оба этих вида транспорта сразу.

[**033**] Способ начинается с построения произвольного графа (graph) системы общественного транспорта GT, который моделирует систему общественного транспорта (public transportation system), и который можно взять из известных технических решений. Граф может содержать вершины (vertexes), которые представляют автобусные остановки, станции метро или и те, и другие сразу. На графе могут присутствовать и другие вершины,

означающие, например, место остановки или разворота транспорта, либо место остановки или поворота пешеходов. Вершины на графе, соответствующие местам остановки транспорта, имеют обозначение

$$STOPSTATION\_s,$$

с индексом s. На графе имеются ориентированные взвешенные ребра (directed weighted edges), каждое из которых обозначает время в пути вдоль отрезка (segment) от исходной вершины ребра до целевой вершины ребра. Граф также может иметь другие ребра. Ребро может обозначать отрезок пути на пересадку между разными маршрутами транспорта. Граф может содержать данные о времени отправления или времени прибытия разных маршрутов общественного транспорта. Как правило, используется Алгоритм Дейкстры по поиску кратчайшего плеча графа от одной вершины до всех остальных (shortest graph paths), либо поисковый алгоритм A* («A звезда») для расчета пути от одной из вершин графа с кратчайшим плечом графа (суммы весовых значений), обозначая кратчайшее время в пути от одной

$$STOPSTATION\_s'$$

до другой

$$STOPSTATION\_s'',$$

для любого s' и s'', также, если перемещение начинается в заданное время или заканчивается в заданное время.

[**034**] Наш способ расширяет граф системы общественного транспорта GT. См. РИС. 5 для иллюстрации.

[**035**] Сначала мы вводим кластеры остановок (clusters of stopstations). Способ группирует остановки по кластерам с помощью какого-либо алгоритма группирования в кластеры; в одном из вариантов реализации способ группирует две остановки в один кластер, если географическое расстояние между двумя остановками не превышает пороговое значение (threshold), например, 5 метров, либо, если время в пути между двумя остановками не превышает пороговое значение. Мы говорим о *расположении* кластера, имея в виду какое-либо географическое расположении в границах кластера, например, его центральную точку. Для каждого кластера остановок c наш способ добавляет две вершины

$$STOPSTATION\_CLUSTER\_SOURCE\_c$$

и

$$STOPSTATION\_CLUSTER\_TARGET\_c,$$

а также ребра, соединяющие кластер с остановками GT

$$STOPSTATION\_CLUSTER\_SOURCE\_c \rightarrow STOPSTATION\_s'$$

с пометкой FirstWaitGetOn, и

$$STOPSTATION\_s' \rightarrow STOPSTATION\_CLUSTER\_TARGET\_c$$

с пометкой Zero, для любого s', всегда, когда s' находится в кластере c. Ребра имеют нулевой вес. Получившийся граф обозначается GC, с вершинами VC и ребрами EC. Подмножество вершин

$$STOPSTATION\_CLUSTER\_SOURCE\_c,$$

для всех c обозначается VS.

[**036**] Другим расширением являются кластеры домов. Способ группирует дома в кластеры по какому-либо алгоритму группирования в кластеры; в одном из вариантов реализации способ использует такой же алгоритм, что и при группировании в кластеры остановок. Аналогичным образом определяется расположение кластера домов. Для каждого кластера s способ вводит вершину

$$HOME\_CLUSTER\_SOURCE\_s$$

и вершину

$$HOME\_CLUSTER\_TARGET\_s.$$

Наш способ соединяет каждый кластер домов с кластерами остановок в виде пеших участков. В частности, способ добавляет ребро

$$HOME\_CLUSTER\_SOURCE\_s \rightarrow STOPSTATION\_CLUSTER\_SOURCE\_c$$

с отметкой Walk, если имеется пеший участок от расположения кластера домов s до расположения кластера остановок c, для любых s и c, при этом вес ребра отражает время в пути по пешему участку; и перевернутое ребро

$$STOPSTATION\_CLUSTER\_TARGET\_c \rightarrow HOME\_CLUSTER\_TARGET\_t$$

с пометкой Walk, если имеется пеший участок в обратном направлении от расположения кластера остановок c до расположения кластера домов t, для любых c и t, при этом вес ребра отражает время в пути по пешему участку. В одном из вариантов реализации способ ограничивает пешие перемещения кратчайшими участками с заданной скоростью, например, 4 км/ч, продолжительностью не выше порогового значения, например, 1 час. Получившийся граф обозначается G. Обозначение VH мы присвоили вершинам

$$HOME\_CLUSTER\_SOURCE\_s,$$

для всех s; а обозначения EH множеству ребер

$$HOME\_CLUSTER\_SOURCE\_s \rightarrow STOPSTATION\_CLUSTER\_SOURCE\_c,$$

для всех s и c.

[**037**] Нас интересует расчет самого короткого плеча графа от каждого кластера остановок до каждого кластера домов, а также самое короткое плечо графа в обратном направлении от каждого кластера домов до каждого кластера остановок.

[**038**] Отметим, что группирование в кластеры позволяет нашему способу значительно

повысить качество расчета самого короткого плеча графа за счет унификации расположений, которые, в сущности, являются одними и теми же с точки зрения самого короткого плеча графа. Например, в высотном жилом комплексе могут находиться сотни квартир, а наш способ объединит их в один единый кластер домов. Таким образом, алгоритму расчета самого короткого плеча графа потребуется обработать всего один кластер домов, а не сотни входящих в него квартир.

[**039**] При этом типовое применение Алгоритма Дейкстры все-таки не обеспечивает высокое качество. В типовом применении алгоритм начинает обработку с каждого

$$STOPSTATION\_CLUSTER\_SOURCE\_c$$

в G без каких-либо вершин

$$HOME\_CLUSTER\_SOURCE\_s$$

после чего алгоритм переходит от каждой вершины

$$HOME\_CLUSTER\_SOURCE\_s$$

в G без каких-либо вершин

$$HOME\_CLUSTER\_TARGET\_t$$

Совокупная асимптотическая временная сложность (asymptotic time complexity) выглядит как

$$O(|VS| \cdot (|EC| + |EH| + (|VC| + |VH|)\log(|VC| + |VH|)))$$
$$+$$
$$O(|VH| \cdot (|EC| + |EH| + (|VC| + |VH|)\log(|VC| + |VH|))).$$

[**040**] В нашем способе улучшено типовое применение этого алгоритма. Мы заметили, что в крупных городских образованиях, количество кластеров домов значительно превосходит количество кластеров остановок $|VH| \gg |VS|$, при одинаковом пороговом значении для группирования в кластеры. Исходя из этого наблюдения, наш способ использует другой алгоритм расчета самого короткого плеча графа от кластеров домов: в нашем способе переворачиваются (reverses) ребра G, и для каждого

$$STOPSTATION\_CLUSTER\_TARGET\_c$$

на перевернутом графе рассчитывается самое короткое плечо до всех вершин

$$HOME\_CLUSTER\_SOURCE\_s$$

при помощи Алгоритма Дейкстры (любые вершины

$$HOME\_CLUSTER\_TARGET\_t$$

можно убрать). Это дает желаемый эффект, ведь, когда мы переворачиваем ребра какого-либо самого короткого плеча от

$$STOPSTATION\_CLUSTER\_TARGET\_c$$

до

$$HOME\_CLUSTER\_SOURCE\_s$$

на перевернутом графе мы получаем самое короткое плечо графа от

$$HOME\_CLUSTER\_SOURCE\_s$$

до

$$STOPSTATION\_CLUSTER\_TARGET\_c$$

на изначальном (не перевернутом) графе. Отсюда по нашему способу получаем асимптотическую временную сложность вида

$$O(|VS| \cdot ((|EC| + |EH|) + (|VC| + |VH|) \log(|VC| + |VH|))).$$

Фактически, наш способ предлагает существенное сокращение общего времени обработки на графе G для городской агломерации Сеула.

[**041**] В нашем способе используется симметричный алгоритм для противоположной ситуации, когда количество кластеров остановок превышает количество кластеров домов $|VH| < |VS|$. В этом случае наш способ переворачивает ребра при расчете самого короткого плеча графа от кластеров остановок до кластеров домов.

[**042**] В одном из вариантов реализации способ использует время отправления (departure times). Способ рассчитывает самое короткое плечо графа от каждого

$$STOPSTATION\_CLUSTER\_SOURCE\_c$$

до каждого

$$HOME\_CLUSTER\_TARGET\_t,$$

для заданного времени отправления из

$$STOPSTATION\_CLUSTER\_SOURCE\_c.$$

В этом случае способ использует соответствующий граф GT из известных технических решений, который позволяет задавать время отправления от остановок общественного транспорта; такой случай иногда обозначается вершиной, которая соответствует значению времени и географическому расположению, и ребром, которое обозначает время в пути из географического расположения в указанное время. Аналогичным образом способ работает на перевернутом графе с использованием заданного времени прибытия в

$$STOPSTATION\_CLUSTER\_TARGET\_c.$$

Длины самого короткого плеча графа переводят эти значения времени прибытия в значения времени отправления из каждого

$$HOME\_CLUSTER\_SOURCE\_s.$$

Аналогичный граф используется для расчета самого короткого плеча графа на основе сроков прибытия (arrival deadline). Соответствующим образом построенный граф может быть использован для расчета вероятности прибытия раньше срока.

[**043**] В одном из вариантов реализации мы не вводим в GT ни вершины

STOPSTATION_CLUSTER_SOURCE_c

ни вершины

STOPSTATION_CLUSTER_TARGET_c.

Вместо этого мы соединяем вершины

HOME_CLUSTER_SOURCE_s

и вершины

HOME_CLUSTER_TARGET_t

прямыми ребрами пеших участков с вершинами

STOPSTATION_s'.

[**044**] В одном из вариантов реализации мы не вводим в *GT* ни вершины

HOME_CLUSTER_SOURCE_s

ни вершины

HOME_CLUSTER_TARGET_t.

Вместо этого мы вводим вершины

HOME_s,

каждая из которых соответствует дому, и которые мы соединяем с вершинами

STOPSTATION_s'

ребрами пеших участков напрямую.

[**045**] В одном из вариантов реализации мы не группируем дома в кластеры.

[**046**] В одном из вариантов реализации мы не группируем остановки в кластеры.

[**047**] В одном из вариантов реализации мы рассчитываем самое короткое плечо графа по срокам прибытия: от кластеров остановок до кластеров домов с учетом срока прибытия в каждый кластер домов, или от кластеров домов до кластеров остановок с учетом срока прибытия в каждый кластер остановок.

[**048**] В одном из вариантов реализации мы используем некоторые алгоритмы расчета плеча графа, отличающиеся от Алгоритма Дейкстры, например, поисковый алгоритм A* («A звезда»). В одном из вариантов реализации мы используем аппроксимирующий алгоритм (approximation algorithm) для расчета самого короткого плеча графа. Мы можем использовать алгоритмы без каких-либо усовершенствований по качеству расчета.

[**049**] В одном из вариантов реализации весовые значения на некоторых ребрах графа обозначают денежную стоимость перемещения, а не время в пути. Тогда наш способ ищет и сравнивает дома, исходя из денежных затрат на дорогу до места работы и обратно. Можно использовать любые другие смысловые ряды для весовых значений ребер, например: количество пересадок между разными маршрутами общественного транспорта, время ожидания, финансовые издержки от ожидания или дальность перемещения.

[**050**] В одном из вариантов реализации мы применяем поиск с многоцелевой оптимизацией (multi-objective optimization search), на основе многомерной стоимости (multi-dimensional cost). Например, мы ищем самое короткое плечо графа, длина которого обозначает время в пути, так, чтобы денежная стоимость, которая обозначает плечо графа, не превышала пороговое значение, или когда денежная стоимость прибавляется в качестве издержки за протяженность плеча графа.

[**051**] В одном из вариантов реализации маршрут перемещения обозначен как плечо графа. Мы рассчитываем различные характеристики самого короткого маршрута перемещения с использованием самого короткого графа, например: первая остановка (т.е. посадка) или последняя остановка (т.е. высадка) на самом коротком плече графа, основную остановку пересадки на самом коротком плече графа, количество пересадок между разными маршрутами, транспорт с наибольшим временем в пути (например, автобус 1234), общее время ожидания на остановках, общее расстояние пеших участков, степень перегруженности конкретного плеча графа в часы-пик или последовательность географических расположений вдоль плеча графа. Такие характеристики могут использоваться при обработке запросов для фильтрования маршрутов перемещения, характеристики которых совпадают с условием, заданным в запросе.

[**052**] В одном из вариантов реализации мы рассчитали два или более плеча графа между некоторыми вершинами графа. Например, одно плечо графа от вершины *u* до вершины *v* можно преодолеть максимум на одном автобусе, а на другом плече графа от вершины *u* до вершины *v* может не быть метро, но время в пути на нем всего на 10 минут дольше, чем по самому короткому плечу графа.

[**053**] В одном из вариантов реализации мы рассчитали плечо графа, которое отвечает разным условиям фильтра (filtering conditions). Например, плечо графа, на котором предусмотрено не более одной пересадки или установлено максимальное время преодоления пеших участков.

[**054**] В одном из вариантов реализации, мы используем сервис построения маршрутов (например, на основе известных технических решений) для расчета кратчайшего времени в пути между домами и остановками общественного транспорта. Поэтому, в некоторых случаях мы не можем использовать графы, описанные в настоящем Разделе 4.4.1.

[**055**] Способ, описанный в Разделе 4.4.1, рассчитывает кратчайшее время в пути от каждого дома до каждой остановки, а также кратчайшее время в обратном направлении. Ниже описана структура данных для эффективного хранения и обработки данных о кратчайшем времени в пути, используемых нашим способом.

4.4.2 Хранение кратчайшего времени перемещения

[**056**] В одном из вариантов реализации наш способ сохраняет (stores) время в пути в виде векторов (vector form). Способ выстраивает последовательности кластеров остановок в виде $s_1, \ldots, s_n$, а также последовательности кластеров домов в виде $h_1, \ldots, h_m$ в некотором порядке, например, произвольно. В одном из вариантов реализации такие последовательности становятся фиксированными. Для каждого s_i наш способ хранит вектор кратчайшего времени в пути от s_i до кластеров домов, используя последовательность кластеров домов

$$v_i = (t_{i,1}, t_{i,2}, \ldots, t_{i,m}),$$

так, чтобы значение вектора (vector) v_i в координате (coordinate) j, $v_i[j]$, было $t_{i,j}$ равное кратчайшему времени в пути от кластера остановок s_i до кластера домов h_j. См. РИС. 6 для иллюстрации.

[**057**] Последовательность кластеров домов упрощает расчет кратчайшего времени в пути до *каждого* кластера домов. Например, если клиенту нужно переместиться от кластера остановок s_{i1} и далее от кластера остановок s_{i2} , мы просто складываем векторы v_{i1} и v_{i2} покоординатно, а получившаяся сумма содержит общее время в пути от обоих кластеров остановок до каждого кластера домов $h_1, \ldots, h_m$.

[**058**] В одном из вариантов реализации способ хранит время обратного пути, от домов до остановок, используя эту же последовательность кластеров домов. То есть, способ хранит вектор

$$v'_i = (t'_{i,1}, t'_{i,2}, \ldots, t'_{i,m}),$$

так, что $v'_i[j] = t'_{i,j}$, является кратчайшим временем в пути от кластера домов h_j до кластера остановок s_i (необходимо учитывать замену переменных; вектор v'_i используется для одного кластера остановок, даже если перемещение осуществляется в обратном направлении).

[**059**] Как правило, два вектора для одного и того же кластера остановок s_i не равны, $v_i \neq v'_i$, потому что минимальное время в пути до дома может отличаться от времени в пути из дома (как правило, перемещение это не симметричный (not symmetric) процесс). Однако в одном из вариантов реализации способ рассчитывает и сохраняет всего один из двух векторов, и использует его вместо второго, благодаря чему сокращается время расчета и экономится место в системе. В другом из вариантов реализации способ сохраняет средневзвешенное (weighted average) значение каждой координаты двух векторов, благодаря чему может снизиться наихудшая погрешность. Весовые значения могут быть заданы в виде 0,5, либо можно задать предпочтение маршруту до или из определенного кластера домов, исходя из, например, частоты запросов.

[**060**] В одном из вариантов реализации способ сохраняет время в пути, округленное до ближайшей целой минуты, используя один бит памяти компьютера, выраженное

беззнаковым целым числом от 0 до 254, при этом 255 обозначает неизвестное или слишком долгое время в пути. Хранение таких данных позволяет эффективно складывать векторы с помощью современного компьютерного оборудования, поддерживающего выполнение Векторных операций (Vector Operations) и Арифметику с насыщением (Saturation Arithmetic) (например, набор команд AVX-512/AVX-512 instruction set или внутренние механизмы графического ускорителя (GPU intrinsics)), при этом не давая погрешности выйти за практически приемлемый уровень не более тридцати секунд, и охватывая значения общего времени в пути вплоть до 4 часов и более. Можно применять любое другое округление (rounding), например, время в секундах можно поделить на 120 и округлить до целого числа, которым будет выражено время при разрешении в 2 минуты.

[**061**] В одном из вариантов реализации способ сохраняет вектор не для кластеров. Например, некоторые $s_1, \ldots, s_n$ обозначают остановки (а не кластеры остановок), или некоторые $h_1, \ldots, h_m$ обозначают дома (а не кластеры домов).

[**062**] В одном из вариантов реализации, наш способ использует другой вид векторов для сохранения значений времени в пути. Например, хеш-таблицу (hash map) или список (list) (координат, значений). Такие нечасто встречающиеся виды могут иметь преимущество в случае множества неизвестных или слишком больших значений времени в пути.

[**063**] В одном из вариантов реализации, способ сохраняет маршруты перемещения, используя векторы, которые идут по последовательности кластеров домов $h_1, \ldots, h_m$, либо другой вид векторов, например, хеш-таблицу или список (координат, значений). Такие маршруты перемещения могут использоваться при обработке запросов для фильтрования маршрутов перемещения, характеристики которых совпадают с условием, заданным в запросе.

4.5 Время в пути по одному маршруту до места работы и обратно

[**064**] Мы даем описание эффективного расчета нашим способом кратчайшего времени в пути по какому-либо маршруту до места работы и обратно. Рассмотрим маршрут до места работы и обратно, начинающийся от кластера домов H, проходящий через работу $k \geq 1$, и возвращающийся в кластер домов: $H \rightarrow W_1 \rightarrow W_2 \rightarrow \ldots \rightarrow W_k \rightarrow H$. Нам нужно найти кратчайшее время в пути для данного маршрута до места работы и обратно от каждого H. Такой расчет будет довольно затратным, ведь количество кластеров домов может быть каким угодно большим, например, 500,000, учитывая радиус в несколько метров, заданный для группирования в кластеры. Но наш способ предлагает решение, которое существенно ускоряет поиск: наш способ раскладывает (decomposes) маршрут до места работы и обратно на часть, не привязанную к домам, и часть, привязанную к домам, как показано на РИС. 7.

4.5.1 Средняя часть: перемещение $W_1 \rightarrow W_2 \rightarrow \ldots \rightarrow W_k$

[**065**] Наш способ находит кратчайшее время в пути для перемещения $W_1 \rightarrow W_2 \rightarrow \ldots \rightarrow W_k$, исключая кластер домов. Для каждого отрезка $W_i \rightarrow W_{i+1}$, указанный способ обращается к одному из сервисов построения маршрутов (например, из известных технических решений), который рассчитывает кратчайшее время в пути от W_i до W_{i+1} с использованием транспортной системы (например, включая пешие участки, поездки на метро и автобусе, пересадки, либо полностью пеший участок от W_i до W_{i+1}). После чего наш способ складывает значения кратчайшего времени в пути в виде разных i. Получившаяся сумма выражается следующим образом:

$$\text{PathNonHomeDuration}(W_1 \rightarrow W_2 \rightarrow \ldots \rightarrow W_k) = \sum_{1 \le i \le k-1} \text{RoutingEngineShortestDuration}(W_i \rightarrow W_{i+1}).$$

Отметим, что в сервис построения маршрутов потребуется отправить только $k-1$ обращений. Это число не привязано к количеству домов.

[**066**] В одном из вариантов реализации наш способ использует время отправления (departure time), срок прибытия (arrival deadline) или иные части запроса при обращении в сервис построения маршрутов.

[**067**] Затем наш способ рассчитывает кратчайшее время в пути для незамкнутой части (open-jaw part) маршрута до места работы и обратно: два отрезка $H \rightarrow W_1$ и $W_k \rightarrow H$, в которых присутствует кластер домов. Расчет должен производиться очень быстро, ведь его результат понадобится для каждого кластера домов.

4.5.2 Начальная часть/Start part: перемещение $H \rightarrow W_1$

[**068**] Наш способ рассматривает два пути перемещения от H до W_1. См. РИС. 8 для иллюстрации. Один из путей предусматривает полностью пеший участок. Если H и W_1 находятся рядом (nearby), зачастую реально самым быстрым будет полностью пешее перемещение. В таком случае наш способ обращается в сервис построения маршрутов (например, из известных технических решений) для расчета кратчайшего времени в пути $walk(H \rightarrow W_1)$. Второй путь предусматривает использование транспортной системы. Наш способ находит кластеры остановок (stopstation clusters), находящиеся в пределах пороговой досягаемости, например, 2 000 метров от W_1, обозначая их как множество A. Множество A представляет собой подмножество $\{s_1, \ldots, s_n\}$ всех кластеров остановок (на РИС. 8 $A = \{s_1, s_2, s_3\}$). Наш метод обращается к сервису построения пеших участков для получения кратчайшего времени в пути $walk(s_i \rightarrow W_1)$ для каждого s_i в A. Если H и W_1 не находятся рядом (not nearby), зачастую реально самым быстрым будет перемещение с использованием остановки из множества A, а затем продолжение перемещения по соответствующему пешему участку. Благодаря примененному нами подходу с двумя

вариантами (two-way approach), получившееся значение часто является кратчайшим временем в пути.

[**069**] Подход с двумя вариантами может быть применен к каждому кластеру домов. Напомним, что вектор v_i' содержит значения кратчайшего времени в пути до кластера остановок s_i от следующих друг за другом кластеров домов. Таким образом, наш способ рассчитывает кратчайшее время в пути от кластера домов h_j, извлекая векторы v_i' из базы данных и используя следующую формулу:

$$a_j = min\left\{walk(H_j \rightarrow W_1),\ \min_{s_i \in A}\{v_i'[j] + walk(s_i \rightarrow W_1)\}\right\}$$

(Уравнение 1)

[**070**] В одном из вариантов реализации, наш способ рассчитывает $walk(H_j \rightarrow W_1)$ только если расстояние от кластера домов H_j до W_1 меньше порогового значения, например, 2 000 метров, либо если время в пути от кластера домов H_j до W_1 меньше порогового значения. Давайте обозначим множество таких кластеров домов как J.

[**071**] В одном из вариантов реализации, благодаря обозначению векторов v', наш способ сразу рассчитывает время в пути от каждого кластера домов с помощью векторных операций (vector operations) с v' следующим образом:

при $j = 1$ до m

$a_j = \infty$

для всех $s_i \in A$

$w = walk(s_i \rightarrow W_1)$

$$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} min\{a_1, v_i'[1] + w\} \\ min\{a_2, v_i'[2] + w\} \\ \vdots \\ min\{a_m, v_i'[m] + w\} \end{pmatrix}$$

для всех $j \in J$

$w = walk(H_j \rightarrow W_1)$

$a_j = \min\{a_j, w\}$.

[**072**] В одном из вариантов реализации наш способ использует математическую формулу (mathematical formula):

$$\min_{s_i \in A}\{walk(s_i \rightarrow W_1) + v_i'\},$$

где операция «+» (operation) прибавляет число к значению в каждой координате вектора, а операция «min» вычисляет минимальное значение в каждой координате по нескольким векторам.

[**073**] Вектор $(a_1, \ldots, a_m)$ значений времени в пути обозначается следующим образом:

$$\text{PathFromHomeDurations}(\rightarrow W_1) = (a_1, \ldots, a_m).$$

[**074**] Пример языка описания алгоритмов для расчета PathFromHomeDurations приведен на РИС. 10.

[**075**] В одном из вариантов реализации используется формат числа uint8 для хранения a_j, $v'_i[\,j\,]$ или w. В одном из вариантов реализации сохраненное число выражено в минутах. В одном из вариантов реализации, сумма $v'_i[\,j\,] + w$ для некоторой j выполняется с использованием Арифметики с насыщением в диапазоне от 0 до порогового значения, например, 255. В одном из вариантов реализации, слагаемое $v'_i[\,j\,]$ сначала преобразуется в более широкий числовой формат, например, форматы fp16 или uint16, и только потом прибавляется к w, во избежание арифметического переполнения (arithmetic overflow) при сложении. В одном из вариантов реализации, некоторые расчеты, включенные в математическую формулу, выполняются с использованием, по крайней мере, одной инструкции для Векторных операций (в том числе, тензорных операций (tensor operations), например, инструкцию AVX-512 или внутренние механизмы графического ускорителя, поддерживаемые аппаратным обеспечением. В одном из вариантов реализации некоторые расчеты, включенные в математическую формулу, разделены (partitioned), и такие разделы выполняются параллельно.

[**076**] В одном из вариантов реализации, наш способ использует структуру данных с ближайшей соседней записью (nearest-neighbor), например, дерево KD для расположений кластеров остановок, чтобы быстро рассчитывать множество A при обработке запроса.

[**077**] В одном из вариантов реализации, наш способ ограничивает множество A значением, не превышающим количество кластеров остановок, ближайших к W_1, например, не более 100.

[**078**] В одном из вариантов реализации наш способ ограничивает множество A кластерами остановок в пределах пешего участка, не выходя за определенное расстояние до W_1, например, в радиусе 2 000 метров.

[**079**] В одном из вариантов реализации наш способ предварительно рассчитывает кратчайшее время в пути по пешим участкам от точек в пределах порогового расстояния от расположений каждого кластера домов до местоположения кластера домов, либо предварительно рассчитывает кратчайшее время в пути по пешим участкам от точек в пределах порогового расстояния от расположений кластеров остановок до расположения кластера остановок. Затем, во время обработки запросов, наш способ не обращается к каким-либо сервисам расчета пеших участков, а вместо этого использует предварительно рассчитанное время в пути по пешим участкам.

[**080**] В одном из вариантов реализации кратчайшее время в пути по пешему участку рассчитывается по геодезической линии, игнорируя любые препятствия. Это позволяет

ускорить расчет $walk(s_i \rightarrow W_1)$ в ущерб точности.

[**081**] В одном из вариантов реализации в Уравнении 1 используются маршруты перемещения. Например, если в пользовательском запросе задано условие, ограничивающее количество пересадок между транспортными маршрутами, мы задаем фильтр для s_i и j в уравнении: находим количество пересадок, сохраненное на маршруте перемещения, и если это число превысит заданное ограничение, мы игнорируем определенные s_i и j в уравнении.

[**082**] В одном из вариантов реализации группировка значений времени в пути по остановкам s_i, точно так же, как это делалось для v'_i, повышает качество работы с базой данных. Действительно, несмотря на то, что множество A зависит от пользовательского запроса, который мы не можем знать заранее, для каждой s_i из множества A необходимо получить ожидаемо большое количество значений времени в пути.

[**083**] В одном из вариантов реализации векторы v'_i для массива i случайным образом распределяются по разным блокам обработки. Это может уменьшить интервал ожидания при расчете вектора (a_1, . . . , a_m), поскольку остановки, включенные во множество A , зачастую равномерно распределены по блокам обработки.

[**084**] В одном из вариантов реализации векторы v'_i для массива i объединяются в блоки обработки по географическому принципу. Это может повысить скорость обработки при расчете вектора (a_1, . . . , a_m), за счет уменьшения потребности в передаче данных, в силу того факта, что множество A зачастую состоит из остановок, находящихся рядом друг с другом.

4.5.3 Конечная часть: перемещение $W_k \rightarrow H$

[**085**] Расчет аналогичен предыдущему пункту, но использует векторы v, а не v'. См. РИС. 9 для иллюстрации. Наш способ рассчитывает множество B кластеров остановок в пределах порогового расстояния от W_k (на РИС. 9 множество $B = \{s_4, s_5\}$). Таким образом, наш способ рассчитывает кратчайшее время в пути от кластера домов h_j в виде

$$b_j = min\left\{walk\left(W_k \rightarrow H_j\right),\ \min_{s_i \in B}\{walk(W_k \rightarrow s_i) + v_i[j]\}\right\}$$

(Уравнение 2)

[**086**] Как и выше, в одном из вариантов реализации при обозначении вектора как v, наш способ рассчитывает время в пути сразу до каждого кластера домов, используя векторные операции с v по Уравнению 2. В одном из вариантов реализации наш способ использует математическую формулу

$$\min_{s_i \in B}\{walk(W_k \rightarrow s_i) + v_i\}$$

[**087**] Вектор (b_1, . . . , b_m) значений времени в пути обозначается следующим образом:

$$\text{PathToHomeDurations}(W_k \rightarrow) = (b_1, \ldots, b_m).$$

[**082**] В нашем способе используются варианты, аналогичные Разделу 4.5.2. Например, в одном из вариантов реализации, наш способ использует векторную операцию, которая рассчитывает сразу часть вектора $(a_1, \ldots, a_m)$ и вектора $(b_1, \ldots, b_m)$.

4.5.4 Объединение начальной, средней и конечной частей

[**089**] Наконец, наш способ рассчитывает (computes) кратчайшее время в пути по маршруту до места работы и обратно для каждого кластера домов. Наш способ просто складывает два вектора и двигает значения по координатам. Мы обозначим получившийся вектор в виде:

$$\begin{aligned}\text{PathDurations}(\rightarrow W_1 \rightarrow W_2 \rightarrow \ldots \rightarrow W_k \rightarrow) = \\ \text{PathFromHomeDurations}(\rightarrow W_1) \\ + \text{PathToHomeDurations}(W_k \rightarrow) \\ + \text{PathNonHomeDuration}(W_1 \rightarrow W_2 \rightarrow \ldots \rightarrow W_k),\end{aligned} \quad \text{(Уравнение 3)}$$

где первый «+» означает покоординатное сложение векторов, а второй «+» означает прибавление числа к значению у каждой координаты вектора.

[**090**] В одном из вариантов реализации наш способ рассчитывает время в пути с учетом заданного времени отправления из какого-либо географического расположения или срока прибытия в какое-либо географическое расположение по какому-либо маршруту до места работы и обратно.

[**091**] В одном из вариантов реализации наш способ предварительно рассчитывает время в пути PathDurations для заданных маршрутов до места работы и обратно. Например, наш способ рассчитывает и сохраняет

$$\text{PathFromHomeDurations}(\rightarrow W)$$

и

$$\text{PathToHomeDurations}(W \rightarrow),$$

для каждого W, который представляет собой школу. В процессе обработки запроса, наш способ находит предварительно рассчитанные векторы в базе данных вместо того, чтобы рассчитывать их по Уравнению 1, Уравнению 2 и Уравнению 3.

[**092**] В одном из вариантов реализации значение PathDurations может быть изменено. Рассмотрим школу W, для приема в которую имеется требование прописки в определенном районе, что означает, что в нее могут ходить только дети из определенных домов. Это условие можно легко выполнить, установив значения координат вектора $\text{PathFromHomeDurations}(\rightarrow W)$ на бесконечность для тех кластеров домов, которые выходят за границы района (zone) прописки для приема в школу, и таким же образом установив значения координат вектора $\text{PathToHomeDurations}(W \rightarrow)$. Аналогичное изменение можно

сделать, если W это ресторан с ограничением по району доставки еды из него, или государственное учреждение, в зоне ответственности которого находится определенный район. Измененные таким образом значения могут быть сохранены и использованы при обработке запросов.

[**093**] В одном из вариантов реализации наш способ применяет произвольную функцию «манипулирования» (manipulation) к значениям координат векторов v_i, v'_i или PathDurations. Например, это может помочь в реализации государственной политики по снижению тарифа проезда при перемещениях между теми или иными частями городской агломерации в особой экономической зоне, пострадавшей от чрезвычайной ситуации. Такая функция может быть применена до обработки запроса (например, в качестве общей политики) или в процессе обработки (например, в качестве персонализированной политики).

[**094**] В одном из вариантов реализации наш способ рассчитывает PathDurations для подмножества кластеров домов. Например, подмножество определяется по условию характеристик дома, заданных в пользовательском запросе. В одном из вариантов реализации мы выбираем последовательность $h_1, \ldots, h_m$ из кластеров домов таким образом, что эти подмножества часто располагаются на коротком отрезке последовательности, что снижает объем передачи данных.

4.6 Поиск одного дома

[**095**] Наш способ демонстрирует эффективный подбор домов по маршрутам до места работы и обратно. Сначала мы покажем несколько примеров поисковых запросов, прежде чем представить общий поисковый запрос для одного дома.

4.6.1 Запрос со взвешенной суммой

[**096**] Рассмотрим семью, где один из родителей ходит на работу, которая находится в $geo1$, пять раз в неделю, а другой из родителей ходит на работу в $geo2$, три раза в неделю. Эта семья хочет подобрать дом с кратчайшим временем еженедельных перемещений. Мы можем найти время еженедельных перемещений семьи для каждого кластера домов в виде взвешенной суммы (weighted sum) двух векторов, а именно:

$$5 \cdot \text{PathDurations}(\rightarrow geo1 \rightarrow)$$
$$+ 3 \ \text{PathDurations}(\rightarrow geo2 \rightarrow).$$

4.6.2 Запрос минимального значения

[**097**] Рассмотрим случай матери-одиночки, которая работает из дома и провожает ребенка в школу. Ей нужно подобрать дом, который находился бы недалеко от какой-либо школы из множества E. Мы можем найти время ежедневных перемещений ребенка для каждого кластера домов в виде минимального покоординатного значения (coordinate-wise minimum) векторов, а именно:

$$\min_{e \in E} \text{PathDurations}(\rightarrow e \rightarrow)$$

4.6.3 Общий поисковый запрос для одного дома

[**098**] В одном из вариантов реализации наш способ определяет запрос как некую последовательность маршрутов до места работы и обратно $path_1, \ldots, path_q$ для любого q и функцию Deriver : $R^q \rightarrow R$, которая преобразует любой вектор чисел q в какое-либо число. Наш способ рассчитывает векторы времени в пути (от начальной позиции, путем извлечения предварительно рассчитанных векторов, либо и с тем, и с другим)

$$\text{PathDurations}(path_1) = (d_{1,1}, \ldots, d_{1,m}),$$

$$\ldots,$$

$$\text{PathDurations}(path_q) = (d_{q,1}, \ldots, d_{q,m}),$$

и покоординатно применяет функцию Deriver к векторам, чтобы получить вектор

$$\begin{aligned}\text{RequestDurations}(path_1, .., path_q, \text{Deriver}) = \\ (\ \text{Deriver}(d_{1,1}, \ldots, d_{q,1}),\\ \text{Deriver}(d_{1,2}, \ldots, d_{q,2}),\\ \ldots,\\ \text{Deriver}(d_{1,m}, \ldots, d_{q,m})\ )\end{aligned}$$

с «выведенным» временем в пути для каждого кластера домов.

[**099**] В одном из вариантов реализации функция Deriver представляет собой взвешенную сумму чисел

$$\text{Deriver}(x_1, \ldots, x_q) = w_1 \cdot x_1 + \ldots + w_q \cdot x_q,$$

для весовых значений (weights) $w_1, \ldots, w_q$. Весовые значения могут быть положительными или отрицательными. Если время в пути интерпретируется в виде денежной стоимости перемещения, тогда отрицательное весовое значение можно интерпретировать как финансовую экономию, например, если клиент хочет сэкономить на перемещениях по определенному маршруту до места работы и обратно (например, на доставках посылок). Весовые значения могут отражать относительную важность маршрутов перемещения до места работы и обратно, например, маршрут до места работы и обратно генерального директора может иметь более высокое весовое значение, чем маршрут до места работы и обратно руководителя первого звена.

[**100**] В одном из вариантов реализации функция Deriver представляет собой минимальное значение чисел

$$\text{Deriver}(x_1, \ldots, x_q) = \min_{1 \le i \le q} x_i$$

[**101**] В одном из вариантов реализации функция Deriver представляет собой взвешенную сумму и минимальное значение

$$\text{Deriver}(x_1, \ldots, x_r, x_{r+1}, \ldots, x_q) = w_1 x_1 + \ldots + w_r x_r + \min_{r+1 \le i \le q} w_i \cdot x_i$$

[**102**] В одном из вариантов реализации функция Deriver является условной, например

$$\text{Deriver}(x_1, x_2) = [$$

$$\text{if } x_1 < 30 \text{ then}$$

$$\text{return } x_1 + x_2$$

$$\text{else}$$

$$\text{return } \infty],$$

или какой-либо алгоритм.

4.7 Сравнение двух домов или более

[**103**] Наш способ демонстрирует эффективный сравнение множества домов по маршрутам до места работы и обратно. Мы показываем реализацию этого способа на нескольких примерах, прежде чем представить общий запрос на сравнение.

4.7.1 Дом, являющийся местом проживания в текущий момент

[**104**] Рассмотрим семью, которая живет в некотором доме. Члены семьи перемещаются по определенным маршрутам к местам работы, в школу и другие места, а также в обратном направлении. Семья рассматривает варианты переезда в другой дом и хочет сравнить общее время в пути для их текущего дома с общим временем в пути для других потенциальных новых домов. Наш способ делает такое сравнение гораздо проще.

[**105**] В одном из вариантов реализации способ рассчитывает время в пути для каждого кластера домов, включая кластер домов h_j текущего дома S семьи

$$\text{RequestDurations} = (q_1, \ldots, q_m).$$

Наш способ возвращает ответ, содержащий вектор «разности» (difference): указанный способ вычитает значение j-й координаты из значения каждой координаты

$$(q_1 - q_j \ , \ldots , \ q_{j-1} - q_j \ , \ 0 \ , \ q_{j+1} - q_j \ , \ldots , \ q_m - q_j).$$

Отрицательное значение координаты вектора «разности» означает, что дом, соответствующий координате, имеет более короткое время в пути по сравнению с текущим домом S.

[**106**] В одном из вариантов реализации наш способ использует одни маршруты до места работы и обратно для текущего дома, и другие маршруты до места работы и обратно для других домов. Таким образом, наш способ помогает семье оценить гипотетический сценарий: «Допустим, мы поменяем место работы и переедем в какой-то другой дом. Как новое время в пути до места работы и обратно будет соотноситься с нашим текущим временем в пути до места работы и обратно?»

4.7.2 Общее сравнение двух домов или более

[**107**] Рассмотрим семью, в которой и мама, и папа должны будут продолжать ходить в

свои прежние места работы (место работы не меняется), но дети могут поменять школы. Рассмотрим другой пример двух семей: родители и бабушка с дедушкой со стороны матери. Они хотят найти два дома в пределах не более 30 минут пути друг от друга, при этом один дом должен быть рядом с больницей, а другой дом рядом с определенными школами и местами работы. Наш способ делает сравнения таких домов с другими домами гораздо более простым.

[**108**] В одном из вариантов реализации наш способ рассчитывает время в пути PathDurations для кластеров домов по некоторому ряду маршрутов до места работы и обратно

$$\text{PathDurations}(path_1) = (d_{1,1}, \ldots, d_{1,m}),$$

$$\ldots,$$

$$\text{PathDurations}(path_q) = (d_{q,1}, \ldots, d_{q,m}).$$

После чего наш способ применяет «обобщенную» функцию Deriver, которая работает не покоординатно, как в Разделе 4.6.3, а одновременно по всем значениям $q \cdot m$ времени в пути и дает вектор одного или множества чисел. Другими словами, обобщенная функция выглядит следующим образом:

$$\text{Deriver} : \mathrm{R}^{q \cdot m} \rightarrow \mathrm{R}^{y},$$

для некоторого y.

4.8 Поиск или сравнение

[**109**] В одном из вариантов реализации, функция Deriver представляет собой какой-либо алгоритм, например, рандомизированный алгоритм, который берет какую-либо вводную (например, маршруты перемещения до места работы и обратно, значения кратчайшего времени в пути, дома или условие, заданное пользовательским запросом) и выдает какой-либо результат (например, «рейтинг» (top list) домов, отвечающих пользовательскому условию, с кратчайшим временем в пути, которое также отвечает пользовательскому условию, с сортировкой по кратчайшему времени в пути).

4.9 Варианты

[**110**] Люди с базовыми техническими знаниями хорошо представят себе множественные модификации и исполнения, не отклоняясь от общей структуры и сути реализации изобретения. Мы приведем несколько вариантов для иллюстрации.

4.9.1 Расширение перемещений

[**111**] В одном из вариантов реализации наш способ сначала рассчитывает кратчайшее время неполного перемещения, а затем расширяет (extends) его на выбранные кластеры домов. Здесь не выполняется дополнительная обработка и сохранение с целью снижения объема хранения и времени обработки за счет меньшего числа кластеров домов в векторах

v_i и v'_i.

[**112**] Метод определяет связующие точки (connectors) домов, которые представляют собой некоторые элементы транспортной системы. В одном из вариантов реализации, связующие точки домов это кластеры домов или кластеры остановок, которые могут отличаться от кластеров, описанных в Разделе 4.4.1. Затем наш способ предварительно рассчитывает и сохраняет в базе данных кратчайшее время в пути от кластеров остановок до кластеров домов, используя варианты реализации, аналогичные описанным в Разделе 4.4.

[**113**] При получении запроса наш способ рассчитывает кратчайшее время в пути от места работы W, содержащегося в запросе, до выбранных кластеров домов. Для этой цели наш способ определяет кластеры остановок рядом с работой W, и извлекает предварительно рассчитанные значения кратчайшего времени в пути от ближайших кластеров остановок до связующих точек домов, чтобы получить кратчайшее время в пути между местом работы W и связующими точками домов, как описано в Разделе 4.5. В одном из вариантов реализации такой расчет использует векторные операции, аналогичные описанным в Разделе 4.5.2 и 4.5.3. После чего данные значения кратчайшего времени в пути расширяются дальше связующих точек домов, получая значения кратчайшего времени в пути между местом работы W и выбранными кластерами домов. Это можно проделать, просто найдя для каждого кластера домов минимальное расширенное перемещение (которое может содержать не только пешие участки), маршрут которого проходит через какую-либо связующую точку домов рядом с кластером домов, например, в пределах 2 000 метров, а в некоторых случаях, найдя прямой кратчайший маршрут между местом работы W и кластером домов (аналогично Разделам 4.5.2 и 4.5.3). В одном из вариантов реализации, в зонах перекрытия, наш способ использует и связующие точки домов, и векторы v_i и v'_i.

[**114**] Качество экстраполяции перемещений пропорционально общему количеству связующих точек домов, а также пропорционально количеству связующих точек домов рядом с каким-либо выбранным кластером домов. В одном из вариантов реализации расширение используется лишь в нескольких частях городской агломерации, когда два количественных значения скорее всего будут небольшими. В одном из вариантов реализации, наш способ использует функцию издержек на выполнение для следующих целей: (1) определение связующих точек домов, (2) выбор кластеров домов, и (3) определение подмножества ближайших связующих точек домов для каждого из выбранных кластеров домов.

4.9.2 Перемещение по маршруту до места работы и обратно на автомобиле

[**115**] В одном из вариантов реализации мы рассчитали время в пути по маршруту до

места работы и обратно на автомобиле, а не на общественном транспорте. Это можно сделать путем внесения небольших модификаций в предыдущие пункты.

[**116**] При предварительной обработке транспортной системы для домов, по Разделу 4.4, вместо использования *GT* в качестве графа системы общественного транспорта, мы возьмем *GT* как граф схемы движения на автомобиле. Граф схемы движения на автомобиле можно получить, например, из известных технических решений. Такой граф может иметь вершины, обозначающие географические расположения на дорогах, и ребра, обозначающие движение автомобиля или поворот автомобиля; ребра могут содержать данные о времени движения в различное время суток, например, в часы-пик.

[**117**] Мы расширим *GT*. Для каждого кластера домов *s*, мы добавим вершины

HOME_CLUSTER_SOURCE_s

и

HOME_CLUSTER_TARGET_s.

Соединим каждый кластер домов с какими-либо дорогами, находящимися в пределах порогового расстояния, например, 100 метров, добавив вершину

CONNECTOR_r,

для, по крайней мере, одного *r*. В одном из вариантов реализации вершина обозначает кратчайшую проекцию (shortest-distance projection) расположения кластера домов на дорогу, которая находится в пределах порогового расстояния; дорога может находиться внутри автостоянки, назначенной кластеру домов. Эта вершина соединяется с двумя вершинами кластера домов ребрами с пометкой Zero и весовым значением 0. Вершина также соединяется с вершинами, обозначающими конечные точки отрезка дороги. Вместо добавления кластеров остановок, которые нам не нужны, поскольку теперь *GT* не моделирует ни станции метро, ни автобусные остановки, мы добавим вершины

REPRESENTATIVE_SOURCE_s'

и

REPRESENTATIVE_TARGET_t'

для массива *s'* и *t'*; данные вершины представляют (represent) расположения, которые часто встречаются (frequently occur) на кратчайших маршрутах; расположения могут быть получены, например, из известных технических решений. Данные вершины могут представлять кластеры. Данные вершины соответствующим образом соединяются с другими вершинами при помощи ребер.

[**118**] Мы используем расширенную модель *GT* для расчета кратчайшего времени движения маршрутам графа от каждого

REPRESENTATIVE_SOURCE_s'

до каждого

$$HOME\_CLUSTER\_TARGET\_s,$$

и от каждого

$$HOME\_CLUSTER\_SOURCE\_s$$

до каждого

$$REPRESENTATIVE\_TARGET\_t',$$

можно использовать варианты реализации, аналогичные вариантам, описанным в Разделе 4.4.1. Данные значения времени в пути будут сохранены в виде вектора v_i и v_i', как описано в Разделе 4.4.2.

[**119**] При расчете времени в пути по маршруту до места работы и обратно, мы использовали движение на автомобиле вместо прохождения пеших участков, как в Разделе 4.5.

[**120**] Для начальной части перемещения, мы делаем следующие модификации, по сравнению с Разделом 4.5.2. Вместо $walk(H_j \rightarrow W_1)$ мы используем время движения на автомобиле от кластера домов H_j до W_1, полученное, например, из известных технических решений; в одном из вариантов реализации мы вместо этого задаем такое время как бесконечность, если расстояние между H_j и W_1 превышает пороговое значение. Множество A представляет собой набор вершин

$$REPRESENTATIVE\_TARGET\_t'$$

расположенных в пределах порогового расстояния от W_1. Для каждой вершины $s_i \in A$, мы рассчитали время движения на автомобиле от s_i до W_1, используя, например, известные технические решения. Затем применим соответствующие модификации в Уравнении 1.

[**121**] Для конечной части перемещения выполним такие же модификации, но теперь в Раздел 4.5.3. Вместо $walk(W_k \rightarrow H_j)$ мы используем время движения на автомобиле от W_k до кластера домов H_j, полученное, например, из известных технических решений; в одном из вариантов реализации мы вместо этого задаем такое время как бесконечность, если расстояние между W_k и H_j превышает пороговое значение. Множество B представляет собой набор вершин

$$REPRESENTATIVE\_SOURCE\_s'$$

расположенных в пределах порогового расстояния от W_k. Для каждой вершины $s_i \in B$, рассчитаем время движения на автомобиле от W_k до s_i, используя, например, известные технические решения. Затем применим эти модификации в Уравнении 2.

[**122**] Используем другие варианты реализации, упомянутые в Разделе 4.5. Например, мы можем рассчитать время в пути на автомобиле по геодезической линии.

[**123**] Можно просто расширить структуру расчета из Раздела 4.6, чтобы иметь

возможность ограничивать виды транспорта, используемые на каждом из маршрутов до места работы и обратно. В результате мы можем определить, например, время в пути по маршруту до места работы и обратно для каждого дома семьи, в которой один из родителей ездит на работу, находящуюся в *geo*1 *на автомобиле* пять раз в неделю, а другой из родителей ездит на работу в *geo*2 *на общественном транспорте* три раза в неделю.

[**124**] Можно аналогичным образом расширить схему сравнения из Раздела 4.7. В результате мы можем, например, найти новый дом, из которого клиент будет ездить на автомобиле, и сравнить с текущим временем в пути на общественном транспорте. Результат, полученный нашим способом, можно использовать как мотив к покупке автомобиля в силу большего удобства при перемещении.

4.9.3 Перемещение с использованием других видов транспорта

[**125**] В одном из вариантов реализации наш способ использует маршруты до места работы и обратно с другими видами транспорта, например: только пешком; только на велосипеде; только на экспресс-автобусе и пешком; только на метро и пешком; только на экспресс-автобусе, метро и пешком; только в совместно используемых автомобилях и пешком; водным транспортом; на самолетах; и так далее. Мы просто используем модификации, аналогичные описанным в Разделе 4.9.2. В одном из вариантов реализации наш способ определяет другие дома и другие средства перемещения, и дает пользователю рекомендации и объясняет выгоду по сравнению с текущим домом и текущими средствами перемещения.

[**126**] Кратчайшее плечо графа на заданном графе можно рассчитать, не зная географических расположений разных вершин графа. Так, в одном из вариантов реализации, наш способ использует транспортную систему, различные элементы (elements) которой не имеют привязки к географическому расположению.

[**127**] Не нужно физически перемещать объекты по транспортной системе. Для нашего способа нужно только иметь возможность определить маршрут или длины маршрутов между элементами транспортной системы. Так, компьютерная сеть, по которой перемещаются данные, служит примером транспортной системы, в которую входят следующие транспортные элементы: кабели/линии (по аналогии с дорогами) и сетевые узлы/коммутаторы (по аналогии с остановками/поворотами). Любой человек, обладающий базовыми техническими знаниями, сможет привести множество примеров транспортных систем.

4.9.4 Условия по маршрутам перемещения

[**128**] Мы можем запросто реализовать разные условия для фильтра (filtering conditions) маршрутов перемещения. Например, мы можем построить граф G , который не имеет

пересадки между какими-либо линиями метро, но имеет пешие участки между кластерами домов и кластерами остановок. Вместе с модификациями, аналогичными описанным в Разделе 4.9.2, наш способ проведет поиск или сравнение домов, живя в которых, клиент будет с комфортом добираться от дома до работы (и дом, и работа должны будут находиться в шаговой доступности от станций метро на одной линии). Аналогичным образом мы можем построить G и изменить алгоритмы расчета кратчайшего плеча графа, чтобы ограничить плечо графа максимум одной пересадкой, либо из метро на автобус, либо из автобуса на метро, или пересадкой в определенный интервал времени, или видом маршрута, наиболее часто используемого клиентами. В нашем способе предусмотрены другие варианты установки фильтра маршрутов перемещения, которые может представить любой человек с базовыми техническими знаниями.

4.9.5 Условия по домам

[**129**] В одном из вариантов реализации наш способ получает условие фильтра по разным характеристикам домов, ищет или сравнивает дома, характеристики которых соответствуют условию. Характеристики могут быть следующие: тип (например, отдельный дом или квартира в высотке), форма проживания (например, с покупкой или арендой), цена, комиссия агентства недвижимости, налоги, максимальная сумма платежа по ипотеке, количество комнат или санузлов, площадь / размер дома, выход окон по сторонам света, номер этажа в доме, количество этажей в здании или стандартная сумма ежемесячных коммунальных платежей. Наш способ просто хранит перечень характеристик для каждого дома и, в зависимости от условия, определяет дома, характеристики которых соответствуют условию. Время перемещения для таких совпавших с условием домов можно получить по времени перемещения для кластеров домов. В нашем способе предусмотрены другие варианты установки фильтра домов, которые может представить себе любой человек с базовыми техническими знаниями.

4.9.6 Мета-поиск или сравнение

[**130**] В одном из вариантов реализации наш способ ищет или сравнивает дома на основе комбинации предыдущих запросов поиска и сравнения, и ответов на них. Это можно представить как мета-способ (meta method) (способ, который использует сам себя). Он полезен, например, для расчета значения, отражающего потенциал застройки новыми объектами недвижимости.

[**131**] Опишем один из вариантов реализации мета-способа. Мы получили некоторое количество u маршрутов до места работы и обратно $path_1, \ldots, path_u$. В одном из вариантов реализации маршруты до места работы и обратно берутся из истории использования компьютерного сервиса по поиску и сравнению домов, каждый маршрут до места работы и

обратно может быть указан разными пользователями сервиса. Для каждого маршрута до места работы и обратно $path_k$ рассчитаем время в пути для всех кластеров домов PathDurations($path_k$) согласно Разделу 4.5. Затем применим агрегатор (aggregator), который обработает значения времени в пути. В одном из вариантов реализации для каждого маршрута до места работы и обратно $path_k$ мы получаем вектор весовых значений w_k. В одном из вариантов реализации каждое весовое значение представляет собой вероятность выбора пользователем того или иного дома в соответствующем кластере домов, что может зависеть от условия по объектам недвижимости или маршрутам перемещения, заданного пользовательским запросом. Затем покоординатно рассчитаем агрегаты, как показано в следующей формуле:

$$\frac{1}{u}\sum_{k=1}^{u} w_k \cdot \text{PathDurations}(path_k)$$

j-й агрегат это средневзвешенное время в пути для кластера домов Hj. В одном из вариантов реализации j-й агрегат обозначает нормализованный вклад кластера домов в общее время в пути по всей городской агломерации; в некотором смысле он ставит кластер домов в центральное положение (centrality) относительно городской агломерации. Так как наш способ быстро рассчитывает PathDurations($path_k$), мы можем быстро рассчитать агрегаты. Это позволяет быстро рассчитывать востребованность каждого объекта недвижимости с точки зрения транспортного удобства.

[**132**] Люди с базовыми техническими знаниями легко представят себе многие другие варианты реализации мета-способа. В других вариантах реализации маршруты до места работы и обратно составляются от географических расположений домов и мест работы. В других вариантах реализации весовые значения задаются отличными от нуля для некоторого ряда кластеров домов с наименьшим временем пути, а для других кластеров с нулевым значением. В других вариантах реализации специалист по обработке данных оценивает гипотетические сценарии; ведь, благодаря преимуществам нашего способа, это можно сделать быстро. В других вариантах реализации агрегатор представляет собой произвольный алгоритм, например, алгоритм, рассчитывающий дисперсию, квантили, накопительную функцию распределения или вероятность превышения пороговых значений.

4.9.7 Маршрут до места работы и обратно с двумя домами или более

[**133**] В одном из вариантов реализации маршрут до места работы и обратно включает (involves) какие-либо два дома или более. Это полезно, например, при поиске и сравнении домов сразу для двух семей или более.

[**134**] В одном из вариантов реализации H_1 и H_2 являются двумя кластерами домов. Мы

можем рассчитать время в пути по какому-либо маршруту от места работы и обратно $H_1 \rightarrow path \rightarrow H_2$, который начинается в H_1, но заканчивается в H_2, который может отличаться от H_1. Мы просто используем j , соответствующий H_1 в Уравнении 1, но в Уравнении 2 используем j , соответствующий H_2.

[**135**] В одном из вариантов реализации предварительно рассчитаем оба вектора ($a_1, \ldots, a_m$) и ($b_1, \ldots, b_m$), а затем найдем время в пути для двух произвольных кластеров домов H_i и H_j, просто прибавив значение координаты ($a_1, \ldots, a_m$), соответствующее H_i, к значению координаты ($b_1, \ldots, b_m$), соответствующее H_j, плюс время в пути по промежуточной части *маршрута* (не привязанной ни к H_i , ни к H_j). Отметим, что время в пути для любой пары кластеров домов, одной из пар m^2, можно найти по константе времени $O(1)$, используя только векторное пространство $O(m)$. Это обусловлено аддитивной структурой (additive structure) времени в пути и тем фактом, что *маршрут* выступает разделителем между любыми двумя кластерами домов.

[**136**] В одном из вариантов реализации множество кластеров домов, разрешенное для H_1, может отличаться от множества кластеров домов, разрешенного для H_2. Например, пользовательское условие может задать ограничение по нахождению H_1 в восточной части города, а H_2 в западной части города. В этом случае мы можем использовать вектор ($b_1, \ldots, b_{m'}$) длиной m', отличной от m.

[**137**] В более общем виде, наш способ рассчитывает время в пути для какой-либо последовательности $k \geq 2$ кластеров домов $H_1 \rightarrow path_1 \rightarrow H_2 \rightarrow path_2 \rightarrow \ldots \rightarrow path_{k-1} \rightarrow H_k$. Используя предварительный расчет, это можно сделать по $O(k)$ времени и $O(m \cdot k)$ пространству.

[**138**] В одном из вариантов реализации маршрут до места работы и обратно включает прямое перемещение между кластерами домов, $H_x \rightarrow H_y$, как на этом примере маршрута до места работы и обратно: $H_1 \rightarrow path_1 \rightarrow H_x \rightarrow H_y \rightarrow path_2 \rightarrow H_2$. На этом примере наш способ делит расчет на три части: (1) расчет времени в пути $H_1 \rightarrow path_1 \rightarrow H_x$, (2) расчет времени в пути $H_x \rightarrow H_y$, и (3) расчет времени в пути $H_y \rightarrow path_2 \rightarrow H_2$. Части (1) и (3) могут выполняться способом, описанным выше в данном Разделе 4.9.7. Часть (2) может выполняться аналогично, если теоретически принять либо H_x, либо H_y в качестве места работы. Например, используя векторы v'_i в координате, соответствующей H_x, мы получим время в пути от кластера домов H_x до каждого кластера остановок; их можно извлечь в виде столбца из предварительно рассчитанных векторов v'_i, $1 \leq i \leq n$. Затем, расширим от каждого кластера остановок в пределах пороговой дистанции от H_y, для получения полного перемещения до H_y, аналогично Разделу 4.5.2, см. пример на РИС. 8 и Уравнение 1. Благодаря предварительному расчету по нашему способу, (3) можно выполнить по времени

пропорционально количеству кластеров остановок в пределах пороговой дистанции. В других вариантах реализации, так как кластеры домов известны заранее, мы предварительно рассчитываем время в пути между каждой парой кластеров домов, а затем просто берем время в пути как константу времени $O(1)$. Аналогичное деление естественным образом применяется к любому маршруту до места работы и обратно, который содержит одно или больше прямых перемещений между кластерами домов.

4.9.8 Исследования пространств

[139] Наш способ представляет высококачественные алгоритмы исследований пространств (exploration algorithms). Рассмотрим ситуацию, когда клиент перемещается по определенному маршруту до места работы и обратно $H_1 \rightarrow path \rightarrow H_2$, с некоторой стоимостью; функция стоимости (cost function) зависит от двух домов H_1 и H_2. Клиент ищет дома H_1 и H_2, чтобы минимизировать эту стоимость. Это полезно, например, когда семья желает сменить места работы (H_1 и H_2) обоих родителей *одновременно*, при этом минимизируя общее время в пути от расположения текущего дома (в пределах *маршрута*) семьи, это известная «задача о двух телах».

[140] Одним из вариантов реализации исследования пространства является алгоритм градиентного спуска (gradient descent algorithm). Функция стоимости может быть дифференцируемой функцией (differentiable function); например, при условии фиксированного *маршрута*, функция берет два дома H_1 и H_2 в качестве вводных данных и возвращает в качестве результата расстояние между двумя домами H_1 и H_2, умноженное на время в пути по маршруту до места работы и обратно $H_1 \rightarrow path \rightarrow H_2$ (данную функцию можно соответствующим образом расширить за пределы дискретного домена (discrete domain) пар домов, например, по географическому расположению домов и экстраполяции). На заданном шаге алгоритма спуска градиента, алгоритм рассчитывает градиент функции стоимости при заданной паре домов, а затем отбирает пару домов по направлению градиента (direction of the gradient) для выполнения следующего шага алгоритма. Градиент может быть рассчитан по значениям функции стоимости, например, с помощью двухточечной формулы (two-point formula), для которой нужно два значения. В одном из вариантов реализации результатом алгоритма спуска градиента является та пара домов, где градиент имеет достаточно низкий критерий (norm).

[141] В одном из вариантов реализации мы предварительно рассчитываем векторы (a_1, . . . , a_m) и (b_1, . . . , b_m), помещенные в векторное пространство $O(m)$, и на каждом этапе нашего способа рассчитываем градиент по константе $O(1)$ времени. Таким образом, алгоритм градиента спуска зачастую может ускорить работу.

[142] В одном из вариантов реализации мы не рассчитываем предварительно (a_1, . . . ,

a_m) и (b_1, . . . , b_m), а вместо этого получаем необходимые значения по Уравнению 1 и Уравнению 2, когда требуется. Такой подход может использоваться, если количество домов настолько большое, что предварительный расчет (a_1, . . . , a_m) или (b_1, . . . , b_m) практически нецелесообразен.

[**143**] В одном из вариантов реализации алгоритм исследования пространства может иметь ограничения по домам. Например, если требуется маршрут до одного дома в направлениях туда-обратно, ограничением будет являться $H_1 = H_2$. Мы можем ограничить географический регион, разрешенный для H_1 или разрешенный для H_2.

[**144**] В одном из вариантов реализации алгоритм исследования пространства использует маршрут до места работы и обратно, включая только один дом: $H_1 \rightarrow path$ или $path \rightarrow H_1$. В других вариантах реализации, алгоритм исследования пространства использует маршрут до места работы и обратно, включая два дома и более, как описано в Разделе 4.9.7.

[**145**] В одном из вариантов реализации функция стоимости зависит от домов и мест работы. Например, такая функция стоимости может использоваться, если клиент стремится минимизировать соотношение издержек по времени и деньгам. В одном из вариантов реализации функцией стоимости является: время перемещения между H и W , плюс денежная стоимость аренды H, минус заработная плата, выплачиваемая на W, с возможным присвоением весовых значений, которые будут означать относительную важность этих составляющих. В одном из вариантов реализации, благодаря преимуществам нашего способа, спуск градиента может эффективно использоваться для поиска домов и мест работы по такой функции стоимости.

4.10 Общий случай

[**146**] В одном из вариантов реализации слова «дом» и «работа» имеют произвольное смысловое значение. Например, рассмотрим ситуацию, когда клиент ищет работу, которая находилась бы недалеко от его текущего дома. Клиент не хочет переезжать в другой дом, а просто хочет найти работу поближе к текущему дому. В этом случае можно просто применить наш способ. При наличии ряда точек на всей городской агломерации, в которых люди работают (например, различные учреждения, предприятия и т.д.), наш способ рассчитывает время в пути между каждым кластером рабочих мест и каждым кластером остановок. Таким образом, наш способ можно считать инструментом «поиска или сравнения рабочих мест по маршруту до места работы и обратно». В одном из вариантов реализации наш способ ищет или сравнивает места работы по виду занятости пользователя или уровню зарплат, а также по времени в пути от текущего дома пользователя. В качестве другого примера, рассмотрим компанию, которая хочет перевезти свой головной офис в

новое место. Наш способ может быть использован для расчета совокупного «времени в пути компании» для каждого нового расположения головного офиса по всей городской агломерации. Таким образом компания может определить, как новое расположение повлияет на маршруты ее сотрудников до места работы и обратно. Новое расположение можно выбрать, например, так, чтобы: (1) уменьшить время в пути до работы и обратно при самых неблагоприятных условиях, и (2) среднее время в пути до работы и обратно было небольшим; тем самым решив задачи для каждого сотрудника и общества в целом.

[**147**] В нашем описании так много сказано про время в пути как основную цель поиска или сравнения. Однако, наш способ может брать за основу любые другие цели, например: денежная стоимость перемещения; расстояние в метрах; определенные характеристики или атрибуты маршрутов перемещения, например: количество пересадок или длина пеших участков; или характеристики домов, например: цена, размер или тип. В одном из вариантов реализации это просто решается построением соответствующих графов и заданием весовых значений ребер. Разные цели могут объединяться при поиске с многоцелевой оптимизацией (multi-objective optimization) на основе многомерной стоимости (multi-dimensional cost), например, при поиска дома, расположение которого минимизирует время в пути, что отражается в финансовых затратах на перемещение.

[**148**] В общем случае наш способ использует произвольные *точки* $S_1, \ldots, S_m$ (точками (site) мы называли дома в разделах выше) и произвольные *места* $P_1, \ldots, P_k$ (местами (place) мы называли места работы в разделах выше), и наш способ ищет или сравнивает точки $S_1, \ldots, S_m$ по маршрутам или длинам маршрутов (в разделах выше: маршрутами (route) называлось описание перемещения, а длиной маршрута (route length) называлось расстояние перемещения), которые начинаются или заканчиваются в некоторых точках или проходят через некоторые места, определяемые произвольными спецификациями маршрутов (спецификацией маршрутов (route specification) назывался маршрут до места работы и обратно в разделах выше), включающими точки и места. Для поиска или сравнения может использоваться любой из вариантов, описанных в Разделе 4.9. Таким образом, наш способ может возвращать в качестве ответа маршруты или длины маршрутов, либо их представление (representation). Информацию, рассчитываемую в одном из вариантов реализации способа, можно использовать рекурсивно (recursively), в качестве вводных для того или иного варианта реализации способа.

4.11 Компьютерная система

[**149**] Одним из вариантов реализации изобретения является компьютерная система (computer system), которая ищет или сравнивает объекты недвижимости по маршрутам до места работы и обратно. Мы проиллюстрировали вариант реализации компьютерной

системы на РИС. 11.

[**150**] В нашем описании мы используем термин «модуль» (module). Как известно из области техники, данный термин означает компьютерную (под)систему, выполняющую определенный функционал. Наша разбивка компьютерной системы на отдельные модули представлена в качестве одного из вариантов и не является обязательной. Люди с базовыми техническими знаниями отметят, что систему можно скомпоновать по модулям в ином виде, не нарушив сущность изобретения.

[**151**] В одном из вариантов реализации каждое перемещение по какому-либо маршруту до места работы и обратно содержит время отправления.

[**152**] Один модуль (1101) системы считывает данные о транспортной системе из источника данных (1102) и строит граф G. При построении графа модуль берет данные о домах из источника данных об объектах недвижимости (1103), а из другого источника данных (1104) берет кратчайшие пешие участки между кластерами домов и кластерами остановок, а также рядом с ними. Граф G содержит данные о расписании маршрутов транспорта. Модуль выдает граф без вершин

$$HOME\_CLUSTER\_SOURCE\_s$$

(1105), а также выдает граф без вершин

$$HOME\_CLUSTER\_TARGET\_s$$

но с перевернутыми ребрами (1106). Также модуль строит структуру данных с ближайшими соседними записями (1107), чтобы можно было находить остановки в пределах порогового расстояния от какого-либо географического расположения, и предварительно рассчитывает кратчайшие пешие участки рядом с кластерами домов и кластерами остановок (1108).

[**153**] Одновременно, другой модуль (1109) системы считывает два графа и рассчитывает кратчайшее плечо графа. В одном из вариантов реализации модуль просматривает временные интервалы каждые 5 минут в течение дня. Для каждого времени отправления модуль строит одну таблицу (1110), содержащую кратчайшее время в пути от кластеров остановок до кластеров домов, используя (1105), и еще одну таблицу (1111), содержащую кратчайшее время в пути от кластеров домов до кластеров остановок, используя (1106). В одном из вариантов реализации, каждое время в пути, округленное до ближайшей минуты, сохраняется как тип данных uint8 на языке программирования C++, резервируя максимальное значение 255 как обозначение неизвестного или очень долгого времени в пути. В одном из вариантов реализации таблицы размещаются (laid out) на дисках HDD по строкам (row-major order). В одном из вариантов реализации система использует иерархию кэширования (cache hierarchy), используя диски HDD, диски SSD и оперативную память. В одном из вариантов реализации таблицы или их части сжимаются (compressed) при помощи

алгоритма сжатия, наример, разностным сжатием (delta compression). По нашим наблюдениям, для любой пары из кластера домов и кластера остановок, время в пути часто является одинаковым в тех или иных периодах времени. Такое сходство также зачастую справедливо в соседнем окружении пары. В одном из вариантов реализации мы подбираем последовательность $h_1, \ldots, h_m$ кластеров домов так, чтобы любой h_i и h_{i+1}, являющиеся соседними (adjacent) в последовательности, с высокой вероятностью были ближайшими кластерами домов, и подбираем последовательность $s_1, \ldots, s_n$ кластеров остановок так, чтобы любая s_i и s_{i+1}, являющиеся соседними в последовательности, с высокой вероятностью были ближайшими кластерами остановок.

[**154**] Модули (1101) и (1109) работают в непрерывном режиме. В результате в системе обеспечивается актуальность данных о времени в пути с учетом определенном времени отправления.

[**155**] Параллельно, модуль времени в пути (1112) рассчитывает PathDurations. Имея маршрут до места работы и обратно, а также время отправления, модуль обращается (1113) за соответствующими значениями PathDurations, которые уже были предварительно рассчитаны. Недостающие значения рассчитываются из начальной позиции: модуль обращается к источнику навигационных данных (1114) для расчета значения PathNonHomeDuration части маршрута до места работы и обратно, не включающей какие-либо дома. Также модуль рассчитывает время в пути PathFromHomeDurations и PathToHomeDurations, включащие дома, путем обращения за векторами ближайших остановок (1107), пешеходных участков (1108) и времени в пути от дома (1110 и 1111) в соответствующее время отправления.

[**156**] Параллельно, модуль обработки запросов (1115) ищет или сравнивает объекты недвижимости. Любой запрос (1116) содержит маршруты до места работы и обратно, включая географические расположения на маршрутах до места работы и обратно, и время отправления, а также функцию Deriver. При получении запроса от пользователя, модуль извлекает PathDurations из модуля времени в пути (1112), применяет функцию Deriver, и возвращает пользователю информацию, выражающую результат функции Deriver (1117).

[**157**] Аспекты изобретения могут быть реализованы в форме устройства, программного обеспечения или аппаратно-программного комплекса. Шаги в изобретении, например, блоки принципиальной схемы, могут выполняться не по порядку, частично параллельно или браться из кеша, в зависимости от функционала или оптимизации. Аспекты изобретения могут быть реализовано в последовательной системе или параллельной/распределенной системе, в которой каждый компонент содержит в себе определенный аспект, можно предусматривать резервирование в других компонентах, а

связь между компонентами может быть реализована, например, через любого вида сети. Изобретение не описывается с отсылкой на какой-либо конкретный язык программирования. Компьютерная программа, выполняющая операции по аспектам изобретения, может быть написана на любом языке программирования, например, C++, Java или JavaScript. Какая-либо программа может исполняться на произвольной аппаратной платформе, например, в центральном процессоре (ЦП) или графическом процессоре (ГП), а также связанными запоминающими устройствами или носителями. Программа может исполнять аспекты изобретения на одной или нескольких программных платформах, включая, помимо прочих: смартфоны с операционными системами Android или iOS, или интернет бразуеры, например, Firefox, Chrome, Internet Explorer или Safari.

4.12 Компьютерный сервис

[**158**] Одним из вариантов реализации изобретения является компьютерный сервис (computer service) по поиску или сравнению объектов недвижимости по маршрутам до места работы и обратно. Сервис предоставляется пользователям через пользовательское устройство, например, в виде приложения на смартфоне или сайта в сети интернет. Любому человеку с базовыми техническими знаниями очевидно, что изобретение не ограничено в использовании только на перечисленных устройствах. Также очевидно, что представление сервиса на наших схемах можно изменить (например, изменив компоновку, размеры, цвета, форму, добавив или убрав компоненты), не нарушая сущности изобретения.

[**159**] В одном из вариантов реализации доступ к сервису осуществляется через приложение на смартфоне. См. РИС. 12 для иллюстрации. Пользователь вводит запрос. В одном из вариантов реализации запрос включает:

- желаемые характеристики объекта недвижимости (1201), например «3 комнаты, многоэтажное знание, верхние этажи»;
- маршруты до места работы и обратно (1202, 1203, «школы Towsend или Jericho High» 1204), время отправления (1205, 1206) и частоту перемещения по каждому маршруту (1207, 1208); и
- географическое расположение текущего дома (1209) пользователя.

[**160**] В ответ, сервис возвращает информацию о времени в пути. Например, сервис отображает (renders) географические расположения объектов недвижимости, соответствующие пользовательскому запросу (1210). Сервис показывает зависимость времени в пути от расположения объектов в сравнении с текущим домом. Сервис показывает краткую информацию (summary) о соответствующем объекте недвижимости, например, его цену. Объекты недвижимости можно нанести (stacked) на двухмерную карту

так, чтобы объекты недвижимости с более коротким временем в пути были на верхнем слое, а объекты недвижимости с более длинным временем в пути были на нижнем слое. Если количество объектов такое, что их нормальное отображение на карте затруднено, тогда сервис может сгруппировать объекты недвижимости и отобразить кластеры, размер которых коррелирует с количеством объектов недвижимости в кластере. Кластер может отображать краткую информацию, например, о количестве, типовых характеристиках объектов недвижимости в кластере.

[**161**] В одном из вариантов реализации сервис показывает объекты недвижимости с кратчайшим временем в пути. Краткая информация об объектах недвижимости также отображается (1211). Объекты недвижимости можно сортировать по времени в пути.

[**162**] В одном из вариантов реализации сервис отображает «карту интенсивности» (heatmap), на которой цветом выделено время в пути из каждого района городской агломерации, например, минимальное время для района, как можно увидеть на иллюстрации РИС. 1. Карта интенсивности может показывать разницу времени в пути для каких-либо домов и времени в пути для текущего дома пользователя. В одном из вариантов реализации карта интенсивности показывает только те объекты недвижимости, которые соответствуют желаемым пользовательским характеристикам объекта недвижимости.

[**163**] В одном из вариантов реализации сервис отображает гистограмму (histogram) времени в пути (1212). Гистограмма содержит время в пути по одной оси, а по другой оси долю объектов недвижимости, расположение которых обеспечивает соответствующее время в пути. В одном из вариантов реализации гистограмма показывает только те объекты недвижимости, которые соответствуют желаемым пользовательским характеристикам объекта недвижимости. В одном из вариантов реализации пользователь может наводить на какой-либо участок гистограммы (1213), а сервис будет отображать результаты по времени в пути для соответствующего участка. В одном из вариантов реализации наш способ использует другой вид графиков, например, круговую диаграмму (pie chart).

[**164**] В одном из вариантов реализации пользователь может ограничить маршруты перемещения, например, ограничив общее время прохождения пеших участков, количество пересадок и т.д., например, используя слайдеры диапазонов (1214).

[**165**] В одном из вариантов реализации сервис отображает краткую информацию о маршруте перемещения для объекта недвижимости.

[**166**] В одном из вариантов реализации сервис возвращает пользователю, по крайней мере, одно из следующего:

(a) географическое расположение: места или точки, отображенной на карте;

(b) время отправления или время прибытия для: места или точки;

(c) краткую информацию о точке; краткая информация может включать, по крайней мере, одно из следующего: имя или адрес точки, цену точки или размер точки;

(d) краткую информацию о точках, образующих кластер из ближайших точек;

(e) изображение кластера ближайших точек, размер изображения, коррелирующий с количеством точек в кластере ближайших точек;

(f) расположение точек по приоритету индекса z (z-index order) на основе длины маршрута, более короткие маршруты должны быть на верхнем слое карты;

(g) информация о маршруте и длине маршрута; информация может включать, по крайней мере, одно из следующего: (i) длину маршрута, время, денежные затраты, скорость или время ожидания; (ii) название: транспортного маршрута, автомобильной дороги, пешего участка, остановки, точки поворота или станции пересадки между маршрутами общественного транспорта; или (iii) географическое расположение с точки зрения: денежных затрат, скорости, времени ожидания, транспортного маршрута, автомобильной дороги, пешего участка, остановки, точки поворота или станции пересадки между маршрутами общественного транспорта;

(h) гистограмма длины маршрута через точки; или карта интенсивности длин маршрута через точки, изображенные на карте;

(i) минимальная длина маршрута; гистограмма минимальной длины маршрута через точки; или карта интенсивности минимальной длины маршрута через точки, изображенные на карте;

(j) взвешенная длина маршрута; гистограмма взвешенной длины маршрута через точки; или карта интенсивности взвешенных длин маршрута через точки, изображенные на карте;

(k) дифференцированная длина маршрута; гистограмма дифференцированной длины маршрута через точки; или карта интенсивности дифференцированных длин маршрута через точки, изображенные на карте;

(l) изображение результата применения функции Deriver;

(m) точка с наименьшей длиной маршрута через точки, при ограничении по денежным затратам на перемещение;

(n) какие-то из представленных выше подпунктов с (a) по (m) ограничиваются условием, заданным пользователем; или

(o) рейтинг точек из множества соответствующих точек, маршрутов и длин маршрутов, отвечающих заданному условию, с сортировкой по длине маршрута.

4.13 Заметка о формула изобретения

[**167**] Люди с базовыми техническими знаниями должны отметить возможность выполнения множества модификаций, а также замены на эквиваленты с большой степенью родства, без нарушения сущности данного изобретения. Кроме того, принципы изобретения можно адаптировать под ту или иную ситуацию без нарушения его сущности. Таким образом, несмотря на раскрытие упомянутых вариантов реализации изобретения, изобретение не должно быть ограничено только этими вариантами реализации. Напротив, изобретение будет предусматривать все варианты реализации, предусмотренные формулой изобретения, изложенной в приложении.

4.14 Глоссарий

[**168**] Мы включили глоссарий отдельных выражений, встречающихся в формуле изобретения, а также ссылки на спецификацию. Ссылки не являются исчерпывающими; существуют иные ссылки. Перечень выражений составлен в порядке по первому упоминанию этих выражений в тексте формулы изобретения.

[**169**] В формуле изобретения также используются следующие выражения, о значении которых мы даем пояснение:

1. выражение «at least one A» («по крайней мере один A») является эквивалентом «$\#A \geq 1$, где $\#A$ это номер A»;
2. выражение «one or more A» («один или более A») является эквивалентом «$\#A \geq 1$, где $\#A$ это номер A»;
3. выражение «plurality of As» («множество A») является эквивалентом «$\#A \geq 2$, где $\#A$ это номер A»;
4. выражение «at least two As» («по крайней мере два A») является эквивалентом «$\#A \geq 2$, где $\#A$ это номер A»;
5. выражение «one of: A, or B» («один из: A или B») является эквивалентом «$\#A + \#B = 1$, где $\#A$ это номер A, а $\#B$ is это номер B»;
6. выражение «at least one of: A, or B» («по крайней мере один из: A или B») является эквивалентом «$\#A + \#B \geq 1$, где $\#A$ это номер A, а $\#B$ это номер B»; и
7. выражение «at least one B or C» («по крайней мере один B или C») является эквивалентом «at least one A, wherein each A is (B or C)» («по крайней мере один A, где каждый A это (B или C)».

выражение	**ссылка на пример использования**
способ (method)	пункт [**003**]

поиск или сравнение (searching or comparing)	пункт [**002**]
точка (site)	пункт [**148**]
маршрут или длина маршрута (route or route length)	пункты [**013**] , [**127**]
место (place)	пункт [**148**]
транспортная система (transportation system)	пункт [**022**]
представитель (representative)	пункты [**022**] , [**117**]
хранение (storing)	пункт [**022**]
база данных (database)	пункт [**022**]
предварительно рассчитанный (precomputed)	Пункт 4.4
описание перемещения (description of travel)	пункт [**020**]
получение (receiving)	пункт [**003**]
запрос (request)	пункт [**022**]
начальное место (start place)	Пункт 4.5.2
конечное место (end place)	Пункт 4.5.3
расчет (computing)	пункт [**089**]
ближайший представитель (nearby representative)	пункты [**068**] , [**120**] , [**121**]
длина перемещения (length of travel)	пункт [**020**]

пороговое значение (threshold)	пункт [**035**]
извлечение (retrieving)	пункт [**022**]
удаленная точка (not nearby site)	пункт [**068**]
ответ (responding)	пункт [**003**]
информация (information)	пункт [**003**]
обозначение (representation)	пункт [**148**]
объект недвижимости / конечная точка маршрута до места работы и обратно (real estate property / commute destination)	пункт [**002**]
время отправления / срок прибытия (departure time /arrival deadline)	пункты [**066**] , [**090**]
место P_i , начальное место P_1, конечное место P_k (place P_i , start place P_1, end place P_k)	Определение 1
первая точка, последняя точка (first site, last site)	Раздел 4.9.7
включает (involves)	Раздел 4.9.7
маршрут или длина маршрута от точки до места (site-place route or route length)	пункт [**091**]
точка кластера / представитель кластера (cluster site /cluster representative)	пункты [**035**] , [**036**]
связующая точка кластера (site connector)	пункт [**112**]
сжатие (compression)	пункт [**153**]
функция манипулирования (manipulation function)	пункт [**093**]
район	пункт [**092**]

(zone)	
транспортные элементы (transportation elements)	пункт [**127**]
длина отрезка (segment length)	пункт [**033**]
граф / вершины графа (graph /graph vertexes)	пункт [**033**]
вершины транспортного элемента (transportation element vertexes)	Раздел 4.9.3
вершина точки / вершина представителя (site vertex / representative vertex)	пункт [**117**]
ребро графа (graph edge)	пункт [**033**]
исходная вершина ребра графа/ целевая вершина ребра графа / весовое значение ребра графа (graph edge source vertex / graph edge target vertex / graph edge weight)	пункт [**033**]
плечо графа или длина плеча графа (graph path or graph path length)	пункт [**033**]
время (time)	пункт [**042**]
алгоритм (algorithm)	пункт [**033**]
перевернутый (reversed)	пункт [**040**]
координата (coordinate)	пункт [**056**]
вектор *v*, значение *v[i]* (vector *v*, value *v[i]*)	пункт [**056**]
вектор *v'*, значение *v'[i]* (vector *v'*, value *v'[i]*)	пункт [**058**]
размещение (laying out)	пункт [**153**]
список / хеш-таблица	пункты [**062**] , [**063**]

(list / hash map)	
весовое значение w_i (weight w_i)	пункт [**059**]
нижняя граница lb_i , верхняя граница ub_i , коэффициент масштабирования sf_i (lower bound lb_i , upper bound ub_i , scaling factor sf_i)	пункт [**060**]
операция «+», операция «min» (+ operation, min operation)	пункт [**072**]
первая математическая формула / вторая математическая формула (first mathematical formula /second mathematical formula)	пункты [**072**] , [**086**]
делит (partitions)	пункт [**075**]
спецификация маршрута L_i (route specification L_i)	пункты [**028**] , [**148**]
функция deriver	пункт [**109**]
минимальная длина маршрута (minimum route length)	Раздел 4.6.2
весовые значения w_1 ,..., w_q (weights w_1 ,..., w_q)	пункт [**099**]
взвешенная длина маршрута (weighted route length)	Раздел 4.6.1
точка S (site S)	пункт [**105**]
дифференцированная длина маршрута (difference route length)	Раздел 4.7.1
условие (condition)	Разделы 4.9.4 , 4.9.5
установка фильтра (filtering)	пункт [**006**] , Разделы 4.9.4, 4.9.5
агрегатор (aggregator)	пункт [**131**]
центральное положение	пункт [**131**]

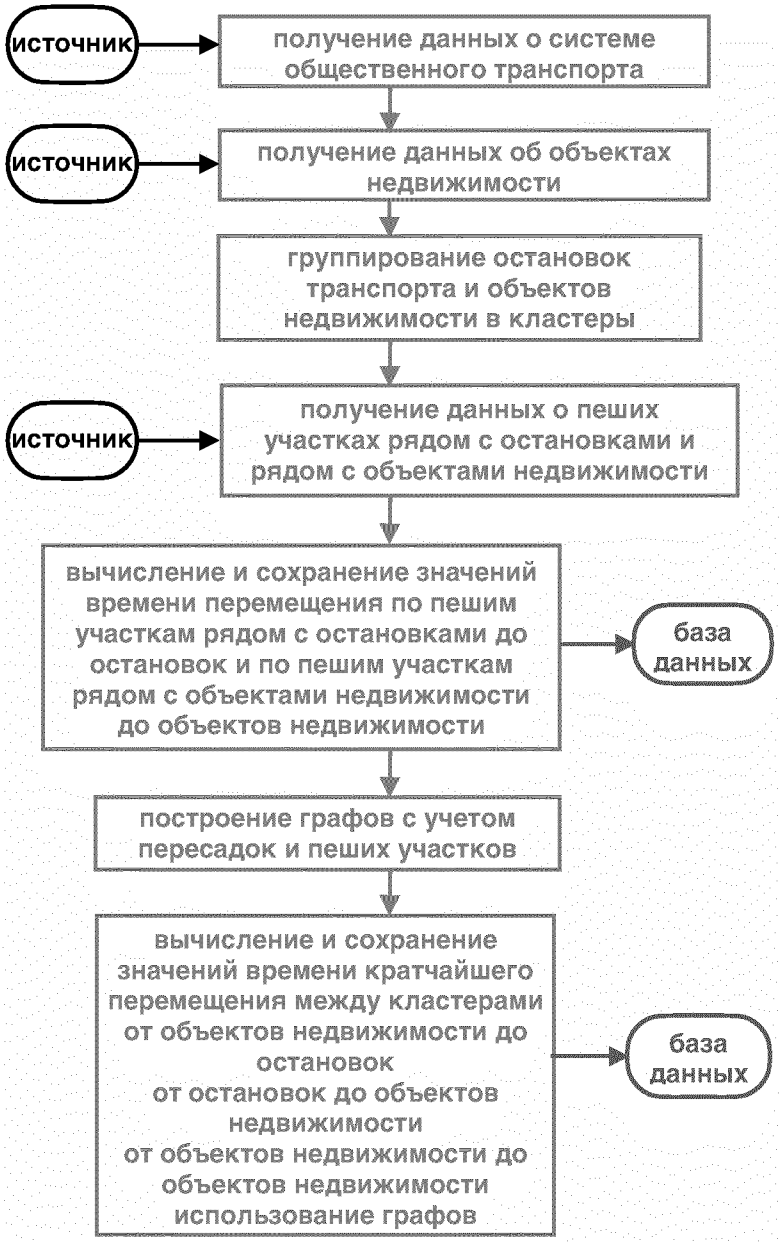
(centrality)	
функция стоимости (cost function)	пункт [**140**]
алгоритм исследования (exploration algorithm)	Раздел 4.9.8
дифференцируемая функция (differentiable function)	пункт [**140**]
спуск градиента (gradient descent)	пункт [**140**]
многоцелевая оптимизация/ многомерная стоимость (multi-objective optimization / multi-dimensional cost)	пункты [**050**] , [**147**]
компьютерная система (computer system)	Раздел 4.11
инструмент (apparatus)	Раздел 4.12
краткая информация / отображение / приоритет отображения (summary / rendering / stacking)	пункт [**160**]
карта интенсивности (heatmap)	пункт [**162**]
гистограмма (histogram)	пункт [**163**]
рейтинг (top list)	пункт [**109**]

1. Способ поиска или сравнения по крайне мере одной точки с использованием по крайней мере двух описаний перемещения в транспортной системе между указанной по крайней мере одной точкой и по крайней мере одним местом, при этом указанный способ включает:
   (a) получение запроса, включающего указанное по крайней мере одно место;
   (b) вычисление указанных по крайней мере двух описаний перемещения, которые включают:
      (i) описание перемещения в указанной транспортной системе между каждым из множества разных мест, включенных в указанное по крайней мере одно место, и точку, включенную в указанную по крайней мере одну точку, но не входящей в указанное множество разных мест, или
      (ii) описание перемещения в указанной транспортной системе между каждым из множества разных точек, включенных в указанную по крайней мере одну точку, и место, включенное в указанное по крайней мере одно место, но не входящее в указанное множество разных точек; и
   (c) реагирование на указанный запрос путем выведенного описания перемещения, полученного на основе указанных по крайней мере двух описаний перемещения.
2. Способ по п.1, отличающийся тем, что указанное выведенное описание перемещения вычисляется с использованием этапов, включающих:
   (a) расчет минимум множества длин перемещения, включенных в указанные по крайней два описания перемещения;
   (b) расчет взвешенной суммы множества длин перемещения, включенных в указанные по крайней два описания перемещения, с использованием числовых весов, дополнительно включенных в указанный запрос; или
   (c) расчет разницы между первой длиной перемещения, имеющей конечную точку, которая является первой точкой, и второй длиной перемещения, имеющей конечную точку, которая является второй точкой,
   при этом указанные две длины перемещения являются включенными в указанные по крайней мере два описания перемещения,
   при этом указанные две точки являются включенными в указанное множество разных точек, и
   при этом одна из двух точек дополнительно является включенной в указанный запрос.

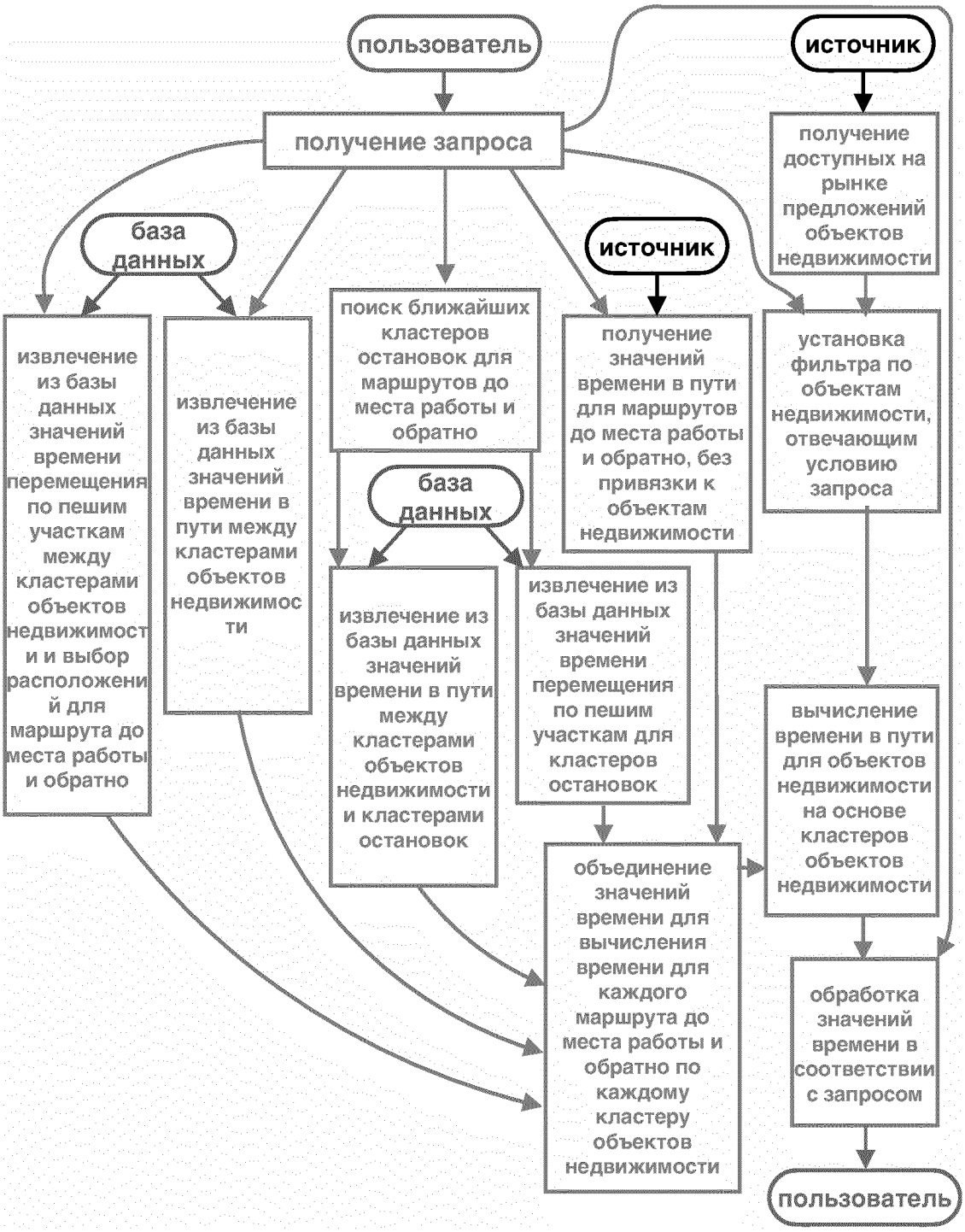
3. Способ по п.1, отличающийся что, что точка, включенная в указанную по крайней мере одну точку, или место, включенное в указанное по крайней мере одно место, представляют собой:
   (a) объект недвижимости,
   (b) место работы,
   (c) школу,
   (d) магазин, или
   (e) ресторан.


geo.0.0:(37.651516,127.071982)
geo.0.1:(37.558873,126.923962)
geo.1.0:(37.498000,127.025000)

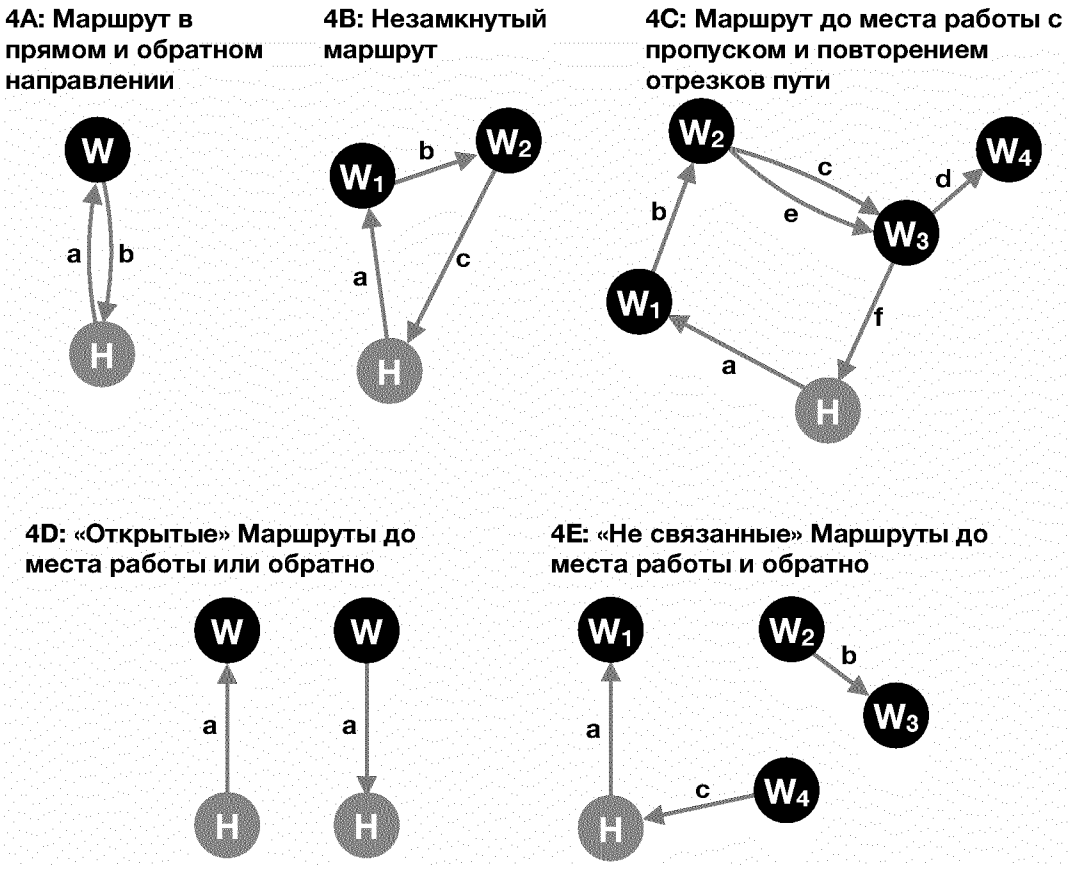
Фиг. 1


источник
получение данных о системе общественного транспорта
источник
получение данных об объектах недвижимости
группирование остановок транспорта и объектов недвижимости в кластеры
источник
получение данных о пеших участках рядом с остановками и рядом с объектами недвижимости
вычисление и сохранение значений времени перемещения по пешим участкам рядом с остановками до остановок и по пешим участкам рядом с объектами недвижимости до объектов недвижимости
база данных
построение графов с учетом пересадок и пеших участков
вычисление и сохранение значений времени кратчайшего перемещения между кластерами
от объектов недвижимости до остановок
от остановок до объектов недвижимости
от объектов недвижимости до объектов недвижимости
использование графов
база данных

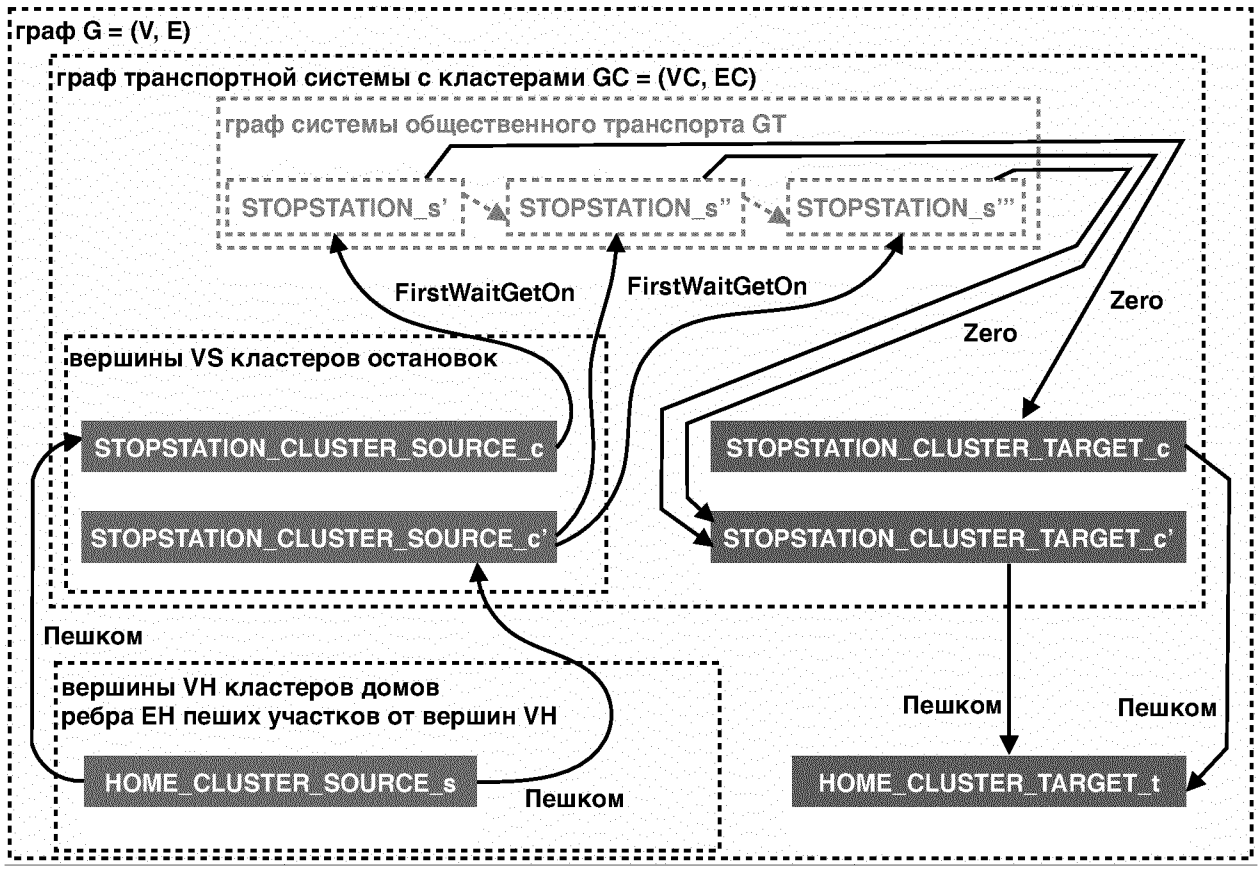
Фиг. 2


пользователь
источник
получение запроса
получение доступных на рынке предложений объектов недвижимости
база данных
источник
поиск ближайших кластеров остановок для маршрутов до места работы и обратно
получение значений времени в пути для маршрутов до места работы и обратно, без привязки к объектам недвижимости
установка фильтра по объектам недвижимости, отвечающим условию запроса
извлечение из базы данных значений времени перемещения по пешим участкам между кластерами объектов недвижимости и выбор расположений для маршрута до места работы и обратно
извлечение из базы данных значений времени в пути между кластерами объектов недвижимости
база данных
извлечение из базы данных значений времени в пути между кластерами объектов недвижимости и кластерами остановок
извлечение из базы данных значений времени перемещения по пешим участкам для кластеров остановок
вычисление времени в пути для объектов недвижимости на основе кластеров объектов недвижимости
объединение значений времени для вычисления времени для каждого маршрута до места работы и обратно по каждому кластеру объектов недвижимости
обработка значений времени в соответствии с запросом
пользователь

Фиг. 3


4A: Маршрут в прямом и обратном направлении
W
a
b
H
4B: Незамкнутый маршрут
W1
b
W2
a
c
H
4C: Маршрут до места работы с пропуском и повторением отрезков пути
W2
c
d
W4
b
e
W3
W1
f
a
H
4D: «Открытые» Маршруты до места работы или обратно
W
a
H
W
a
H
4E: «Не связанные» Маршруты до места работы и обратно
W1
a
H
W2
b
W3
c
W4

Фиг. 4


граф G = (V, E)
граф транспортной системы с кластерами GC = (VC, EC)
граф системы общественного транспорта GT
STOPSTATION_s'
STOPSTATION_s''
STOPSTATION_s'''
FirstWaitGetOn
FirstWaitGetOn
Zero
Zero
вершины VS кластеров остановок
STOPSTATION_CLUSTER_SOURCE_c
STOPSTATION_CLUSTER_SOURCE_c'
STOPSTATION_CLUSTER_TARGET_c
STOPSTATION_CLUSTER_TARGET_c'
Пешком
вершины VH кластеров домов
ребра EH пеших участков от вершин VH
HOME_CLUSTER_SOURCE_s
Пешком
Пешком
Пешком
HOME_CLUSTER_TARGET_t

Фиг. 5

от кластера остановок до кластера домов

noHomeClusterSource

StopstationClusterSource
➜
HomeClusterTarget

	h_1	h_2	h_3	h_4	h_5	...	h_m
s_1	12	255	52	0	134	...	49
s_2	2	23	67	9	243	...	194
.	.						
.	.						
.	.						
s_n	12	255	52	0	134	...	49

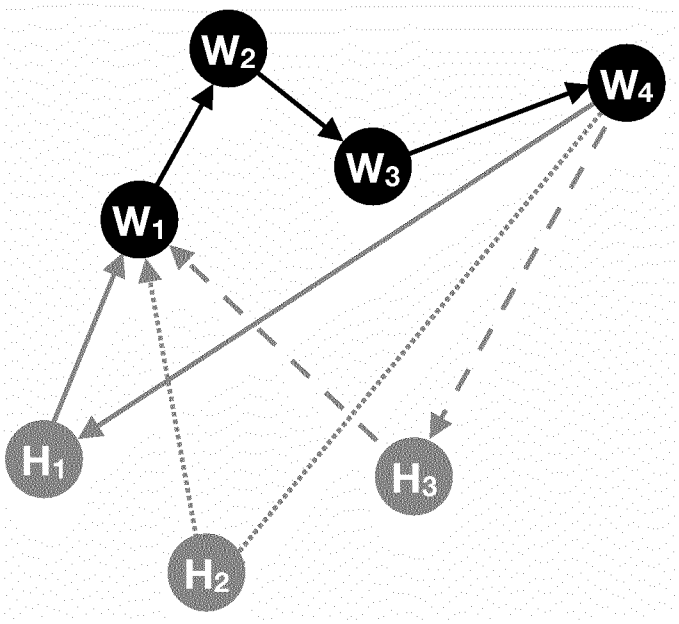
от кластера домов до кластера остановок

перевернутый, noHomeClusterTarget

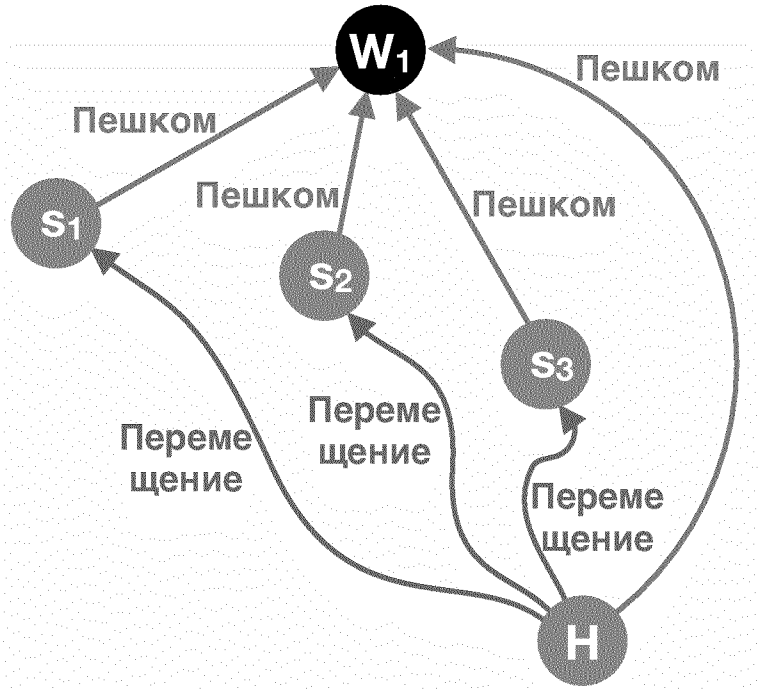
StopstationClusterTarget
➜
HomeClusterSource

	h_1	h_2	h_3	h_4	h_5	...	h_m
s_1	11	255	53	0	139	...	35
s_2	2	26	69	12	255	...	190
.	.						
.	.						
.	.						
s_n	13	250	50	0	130	...	50

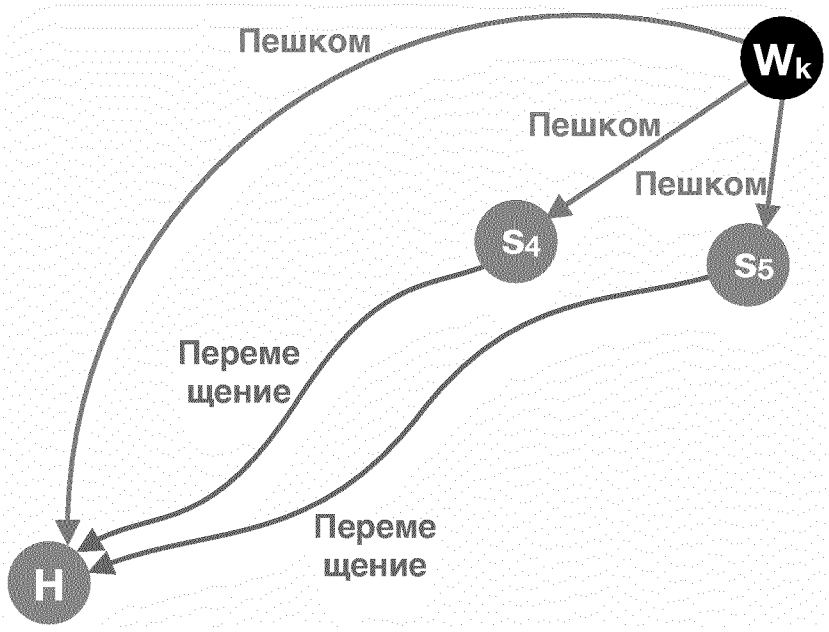
Фиг. 6



Фиг. 7


W1
Пешком
Пешком
Пешком
Пешком
S1
S2
S3
Переме
щение
Переме
щение
Переме
щение
H

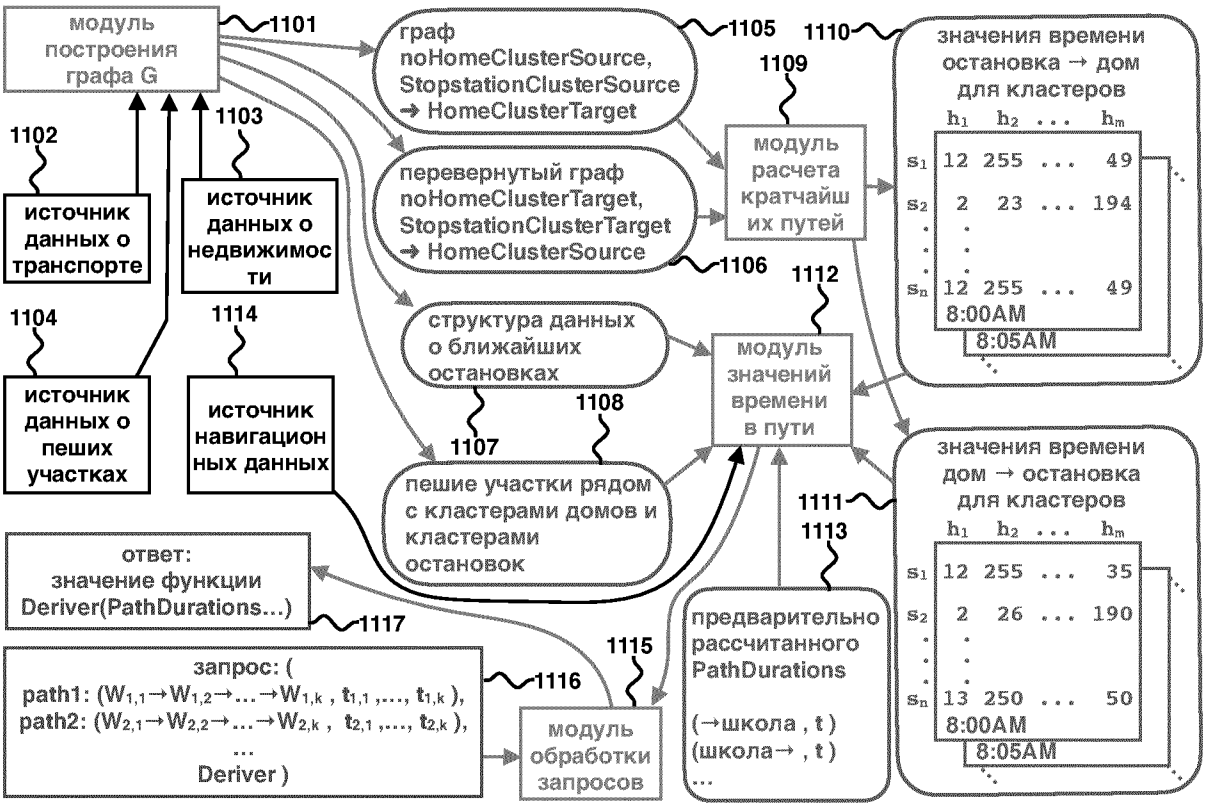
Фиг. 8


Пешком
Wk
Пешком
Пешком
S4
S5
Переме
щение
Переме
щение
H

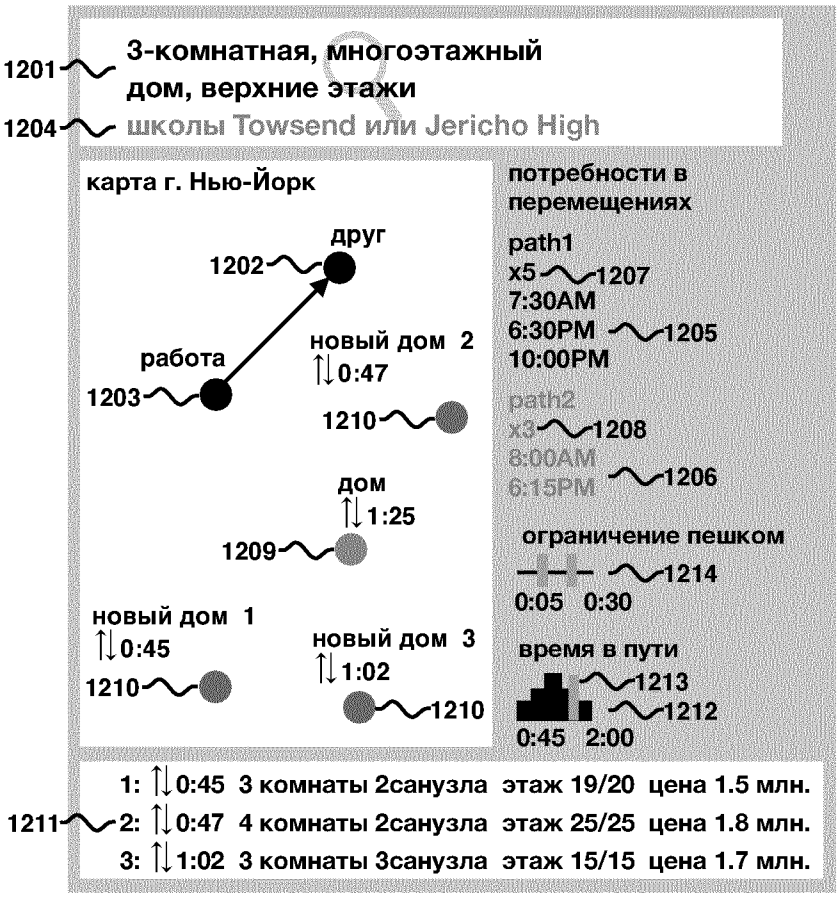
Фиг. 9

```
vector home_cluster_durations;
CoordinatewiseSetInfinity(home_cluster_durations);

limit_meters = 2000;

// Travel and walk.
vector stopstation_clusters;
vector walk_seconds;
(stopstation_clusters, walk_seconds) =
    storage.geo_walks.WalksFromStopstationClustersTo(
        W, limit_meters);
for i = 1 to stopstation_clusters.size() {
   stopstation_cluster = stopstation_clusters[i];
   walk_second = walk_seconds[i];

   vector home_cluster_seconds =
      storage.shortest_paths.HomeClusterSecondsToStopstationCluster(
         stopstation_cluster);
   // Dense vector operation.
   for j = 1 to home_cluster_seconds.size() {
      home_cluster_durations[j] =
         min(
            home_cluster_durations[j],
            home_cluster_seconds[j] + walk_second);
   }
}

// Direct walk.
vector home_clusters;
(home_clusters, walk_seconds) =
    storage.geo_walks.WalksFromHomeClustersTo(W, limit_meters);
// Sparse vector operation.
for i = 1 to home_clusters.size() {
   home_cluster = home_clusters[i];
   walk_second = walk_seconds[i];
   home_cluster_durations[home_cluster] =
      min(
         home_cluster_durations[home_cluster],
         walk_second);
}

return home_cluster_durations;
```

Фиг. 10


модуль построения графа G
1101
1102
1103
1104
1114
источник данных о транспорте
источник данных о недвижимости
источник данных о пеших участках
источник навигационных данных
граф noHomeClusterSource, StopstationClusterSource → HomeClusterTarget
1105
перевернутый граф noHomeClusterTarget, StopstationClusterTarget → HomeClusterSource
1106
структура данных о ближайших остановках
1107
1108
пешие участки рядом с кластерами домов и кластерами остановок
1109
модуль расчета кратчайших путей
1110
значения времени остановка → дом для кластеров
1112
модуль значений времени в пути
1111
1113
значения времени дом → остановка для кластеров
предварительно рассчитанного PathDurations
(→школа , t)
(школа→ , t)
...
ответ: значение функции Deriver(PathDurations...)
1117
запрос: (
path1: (W1,1→W1,2→...→W1,k , t1,1 ,..., t1,k),
path2: (W2,1→W2,2→...→W2,k , t2,1 ,..., t2,k),
...
Deriver)
1116
1115
модуль обработки запросов
8:00AM
8:05AM

Фиг. 11


1201
3-комнатная, многоэтажный дом, верхние этажи
1204
школы Towsend или Jericho High
карта г. Нью-Йорк
друг
1202
работа
1203
новый дом 2
0:47
1210
дом
1:25
1209
новый дом 1
0:45
1210
новый дом 3
1:02
1210
потребности в перемещениях
path1
x5
1207
7:30AM
6:30PM
1205
10:00PM
path2
x3
1208
8:00AM
6:15PM
1206
ограничение пешком
1214
0:05 0:30
время в пути
1213
1212
0:45 2:00
1: 0:45 3 комнаты 2санузла этаж 19/20 цена 1.5 млн.
1211
2: 0:47 4 комнаты 2санузла этаж 25/25 цена 1.8 млн.
3: 1:02 3 комнаты 3санузла этаж 15/15 цена 1.7 млн.

Фиг. 12