



Ссылка на статью:

// Ученые записки УлГУ. Сер. Математика и информационные технологии. УлГУ. Электрон. журн. 2018, № 1, с. 109-122.

Поступила: 03.06.2018

Окончательный вариант: 09.06.2018

© УлГУ

УДК 519.85

Онлайн-сервис для решения задачи коммивояжера с использованием генетического алгоритма и Google API

Юрин Д.А., Чекал Е.Г. *

checal@mail.ru

УлГУ, Ульяновск, Россия

В данной статье описывается разработанное кроссплатформенное программное средство решения задачи коммивояжера, использующее генетический алгоритм и средства Google API. Предложен способ кодирования хромосомы, использующий индексный подход, который сокращает время работы алгоритма. Модифицированный алгоритм скрещивания особей позволяет исключить дублирование точек маршрута. Онлайн-сервис располагается по адресу <http://u0160992.cp.regruhosting.ru/getroute/>

Ключевые слова: задача коммивояжера, комбинаторная оптимизация, генетический алгоритм, службы Google API, онлайн-сервис, программное средство.

Введение

Задача коммивояжера — одна из самых известных задач комбинаторной оптимизации [1, 2, 3, 4], заключающаяся в отыскании кратчайшего маршрута, проходящего через указанные города хотя бы один раз с последующим возвратом в исходный город.

Ранним вариантом задачи может считаться игра Икосиан, созданная У. Р. Гамильтоном в 1857 году, в правилах которой сказано, что участникам надо соединить 20 точек додекаэдра так, чтобы каждая точка использовалась не более одного раза, а конечная точка пути совпадала с исходной. Математические вопросы, связанные с задачей коммивояжера, были рассмотрены в 1800-ых годах У. Р. Гамильтоном и Т. П. Киркменом.

Затем появились новые формулировки и названия задачи К. Менгера и Г. Уитни.

В середине прошлого столетия интерес к задаче коммивояжера вновь возник в связи с попытками применения задачи коммивояжера в сельскохозяйственном секторе (К. Менджер, П. Ч. Мэхаланобис, Г. Джессен и др.).

Задача коммивояжера оказалась простой для понимания, но крайне сложной для вычисления. В 1972 Р. М. Карп опубликовал доказательство, что задача гамильтонова цикла принадлежит к классу NP-полных задач. Это открытие дало научное объяснение очевидной вычислительной трудности нахождения оптимальных маршрутов.

Методы решения задачи коммивояжера становились более сложными, количество городов возрастало. В 1954 Д. Б. Данциг, Д. Р. Фалкерсон и К. Джонсон издали описание метода для решения задачи и практически проиллюстрировали его, решив задачу с 49 узлами, соединив таким образом города каждого штата Америки. В 1971 году М. Хельд и Р. М. Карп решили задачу поиска оптимального пути между 64 узлами, а в 1973 М. Лин и Б. У. Керниган нашли применение задачи коммивояжера в инженерии – они соединили 318 пунктов, полученных в результате резки лазером.

В 1987 году М. Грегчел, М. Падберг и Д. Ринальди нашли оптимальный тур, связывающий уже 2393 города. В 1994 году усилиями Д. Аплгейта, Р. Биксби, В. Шватала, У. Кука, Г. Райнелта количество городов увеличилось до 7397, а спустя 10 лет - до 24978, и, наконец, в 2006 году было получено решение задачи коммивояжера с 85900 городами.

Несмотря на достаточно большое количество алгоритмов решения задачи коммивояжера (например, «Алгоритм полного перебора», «Алгоритм имитации отжига», «Алгоритм Прима-Эйлера», «Алгоритм Литтла», «Алгоритм ближайшего соседа», «Генетический алгоритм», отсечений, ветвей и границ и др.) у современного пользователя отсутствует достаточно удовлетворительный сервис, предоставляющий решение поставленной задачи в короткий промежуток времени. Существующие картографические сервисы, такие как «Яндекс.Карты», «Google Maps», «2gis», «Meganavigator», «PoncyRu», «SpeedyRoute» либо не предоставляют пользователям возможность поиска оптимального маршрута с учетом возвращения в исходную позицию, или не имеют удобного способа выбора точек маршрута, либо не выдают информации о построенном маршруте, и не предоставляют пользователю возможность задания критериев оптимального маршрута, либо не обладают адаптивным к различным устройствам отображения интерфейсом.

1. Этапы генетического алгоритма

Основываясь на сравнительном анализе алгоритмов решения задачи коммивояжера, проведенном в статье [2] на примере выбора маршрута прокладки кабеля сети кольцевой архитектуры, за основу алгоритма построения маршрута для разрабатываемого онлайн-сервиса был взят генетический алгоритм.

Целевая функция генетического алгоритма имеет вид:

$$F = \sum_{i=1}^n R_i \rightarrow \min$$

где

R_i – расстояние i -го ребра гамильтонова графа,
 n – количество ребер гамильтонова графа.

Таким образом, критерием наилучшего маршрута является наименьшее суммарное расстояние между точками маршрута.

Однако пользователь имеет возможность влиять на построение маршрута, управляя параметрами:

- наилучшее время;
- использовать/не использовать платные дороги.

В этих случаях учитывается дорожно-транспортная обстановка: например, при наличии «пробок» выбирается иной маршрут, возможно длиннее, но время прохождения которого будет меньше. Аналогично учитывается платность дорог.

Для кодирования особи предлагается индексный подход. Хромосома особи представляет собой одномерный массив, в качестве генов хромосомы выступают индексы точек маршрута. Каждый ген хромосомы связан с индексом в дополнительном массиве, содержащем десятичные координаты точек (рис. 1). Такой способ кодирования хромосомы позволяет сократить вычислительную нагрузку и упростить процесс работы генетического алгоритма.

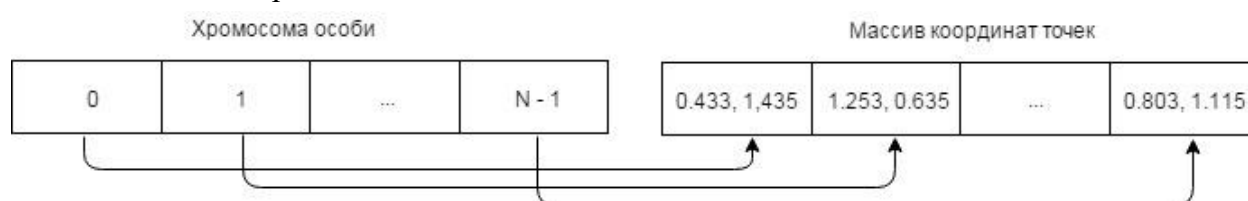


Рис. 1. Принцип кодирования хромосомы.

Для выбора родителей применяется турнирный отбор, заключающийся в выборе двух особей с наилучшим значением целевой функции (длины маршрута) из временной популяции случайно отобранных особей. Такой отбор уменьшает преждевременное исчезновение особей, имеющих низкий коэффициент приспособленности.

Скрещивание особей в генетическом алгоритме для задачи коммивояжера является нестандартным процессом, так как простая операция скрещивания путем обмена особей частями хромосом приводит к дублированию точек маршрута. Поэтому предложен иной метод скрещивания хромосом (рис. 2), где:

1. каждая из двух хромосом разбивается на три части;
2. происходит обмен средней частью обеих хромосом;
3. происходит перекрестный обмен крайними частями хромосом.

Мутация особи в популяции реализована путем взаимной перестановки двух генов хромосомы (рис. 3), которая естественным образом порождает новые маршруты.

Вычислительные затраты генетического алгоритма для решения задачи коммивояжера складываются из затрат на оператор скрещивания $T_{ск}$, оператор селекции T_c , оператор мутации T_m и расчета целевой функции $T_{ц}$. Затраты $T_{ск}$, T_m и $T_{ц}$ при обработке одной хромосомы имеют линейную зависимость от числа ребер n гамильтонова графа. T_c имеет

линейную зависимость от объема популяции M . Поэтому временные затраты в пределах поколения имеют оценку $O(n \cdot M)$. Для N поколений временная сложность алгоритма имеет оценку $O(n \cdot M \cdot N)$. Учитывая что параметры M и N сравнимы и значительно меньше n , можно считать, что оценка временной сложности всего генетического алгоритма в целом лежит в пределах от $O(n^2)$ до $O(n^3)$.

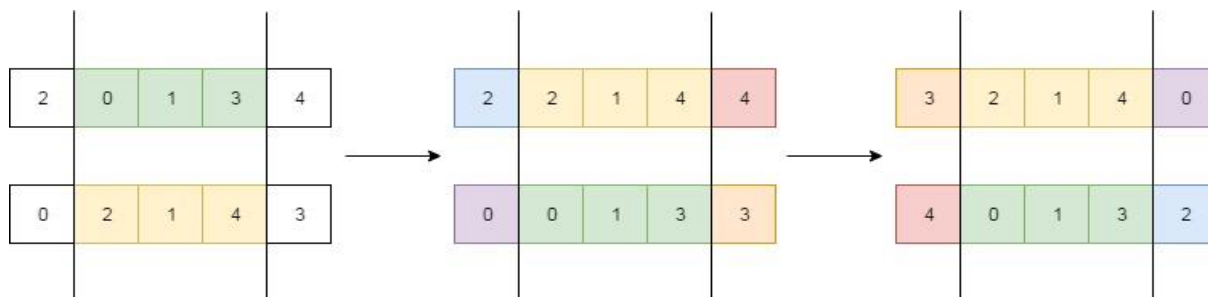


Рис. 2. Процесс скрещивания особей.



Рис. 3. Процесс мутации особи.

Ограничения реализованного генетического алгоритма в свободно распространяемом онлайн-сервисе вызваны ограничениями сервиса Google Maps API при бесплатном использовании своих функций.

Каждый запрос, отправляемый в Google Maps Distance Matrix API, ограничен количеством разрешенных элементов, которое определяется произведением количеств начальных и конечных точек. Пользователи стандартного API имеют следующие ограничения:

- 2 500 бесплатных элементов в день, рассчитывается как сумма запросов на стороне клиента и на стороне сервера;
- Не более 25 исходных точек или 25 точек назначения на запрос;
- 100 элементов на запрос, рассчитывается как сумма запросов на стороне клиента и на стороне сервера.

Завершение работы генетического алгоритма устанавливается количеством поколений. Как только число поколений достигает пятидесяти, алгоритм заканчивает порождение поколений и выводит наилучший результат среди особей последнего поколения. Исходя из ограничений Google Maps API исходных точек может быть не более 10, а указанного числа поколений достаточно, чтобы получить корректный ответ в приемлемое время.

В случае создания коммерческого онлайн -сервиса эти ограничения могут быть сняты.

2. Модели онлайн-сервиса

Функциональная модель приложения в виде диаграммы прецедентов представлена на рис. 4.

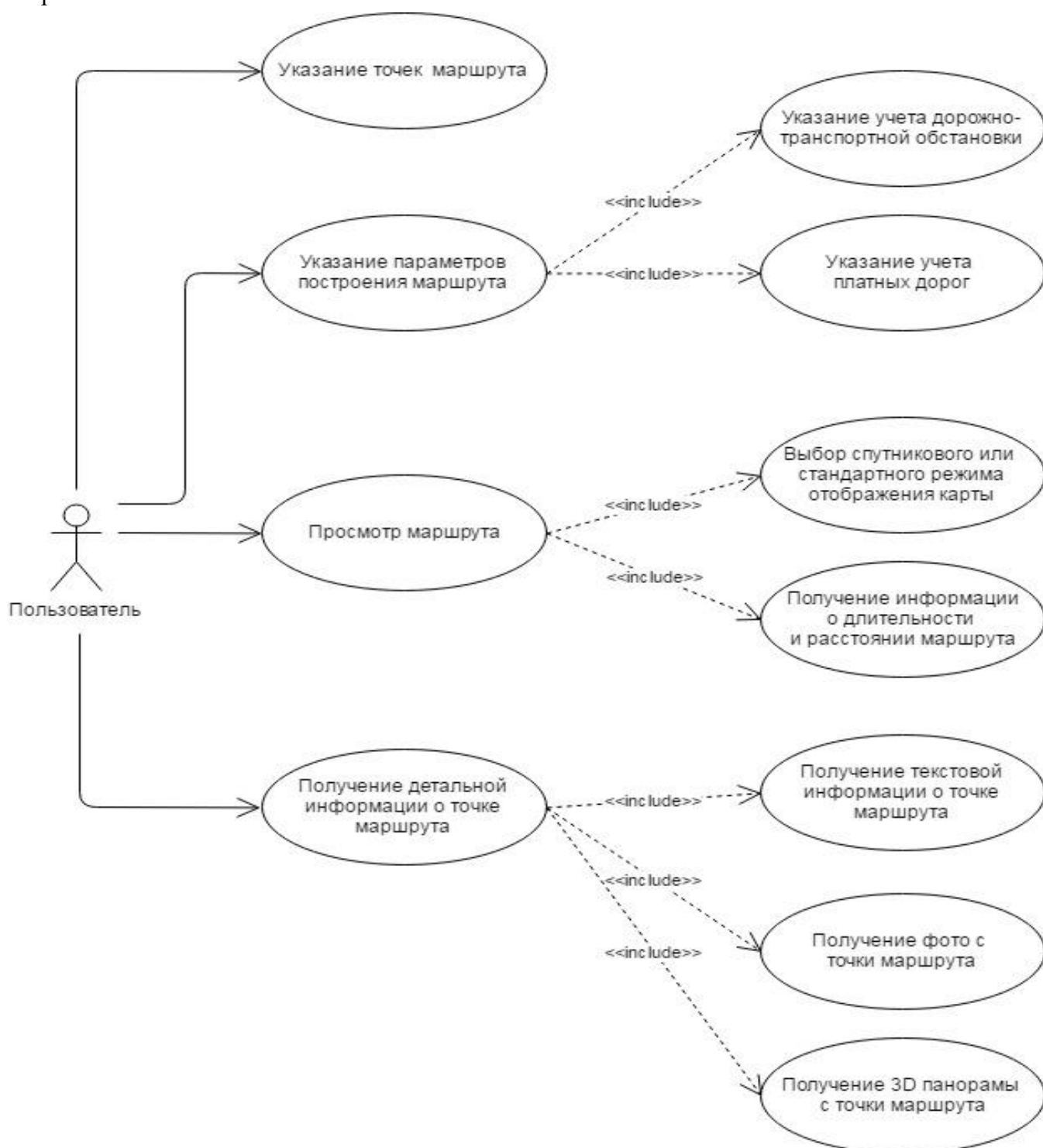


Рис. 4. Диаграмма прецедентов.

Динамическая модель в виде диаграммы состояний показана на рис.5.

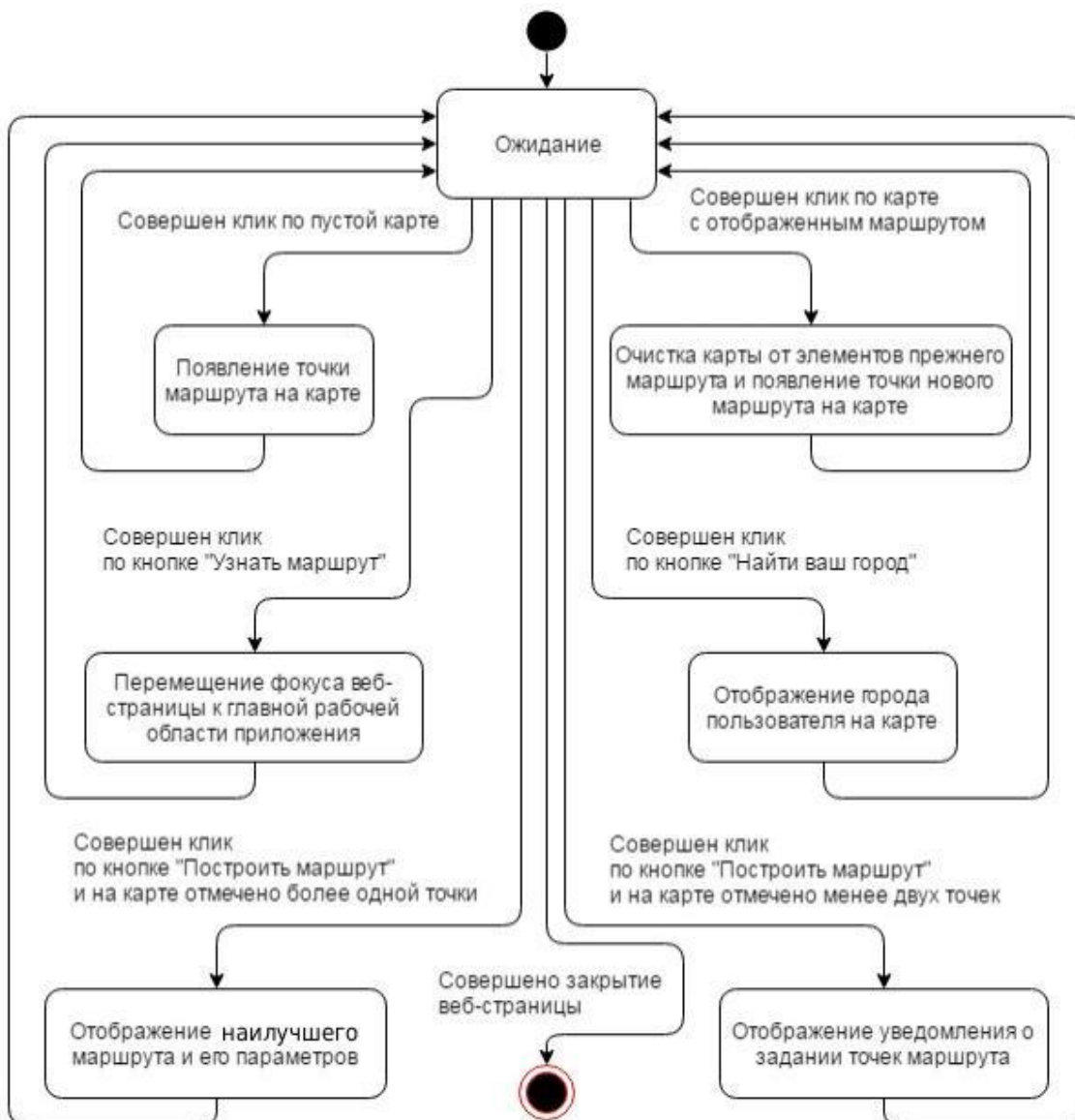


Рис. 5. Диаграмма состояний.

Обобщенный алгоритм функционирования приложения в виде диаграммы деятельности показан на рис. 6.

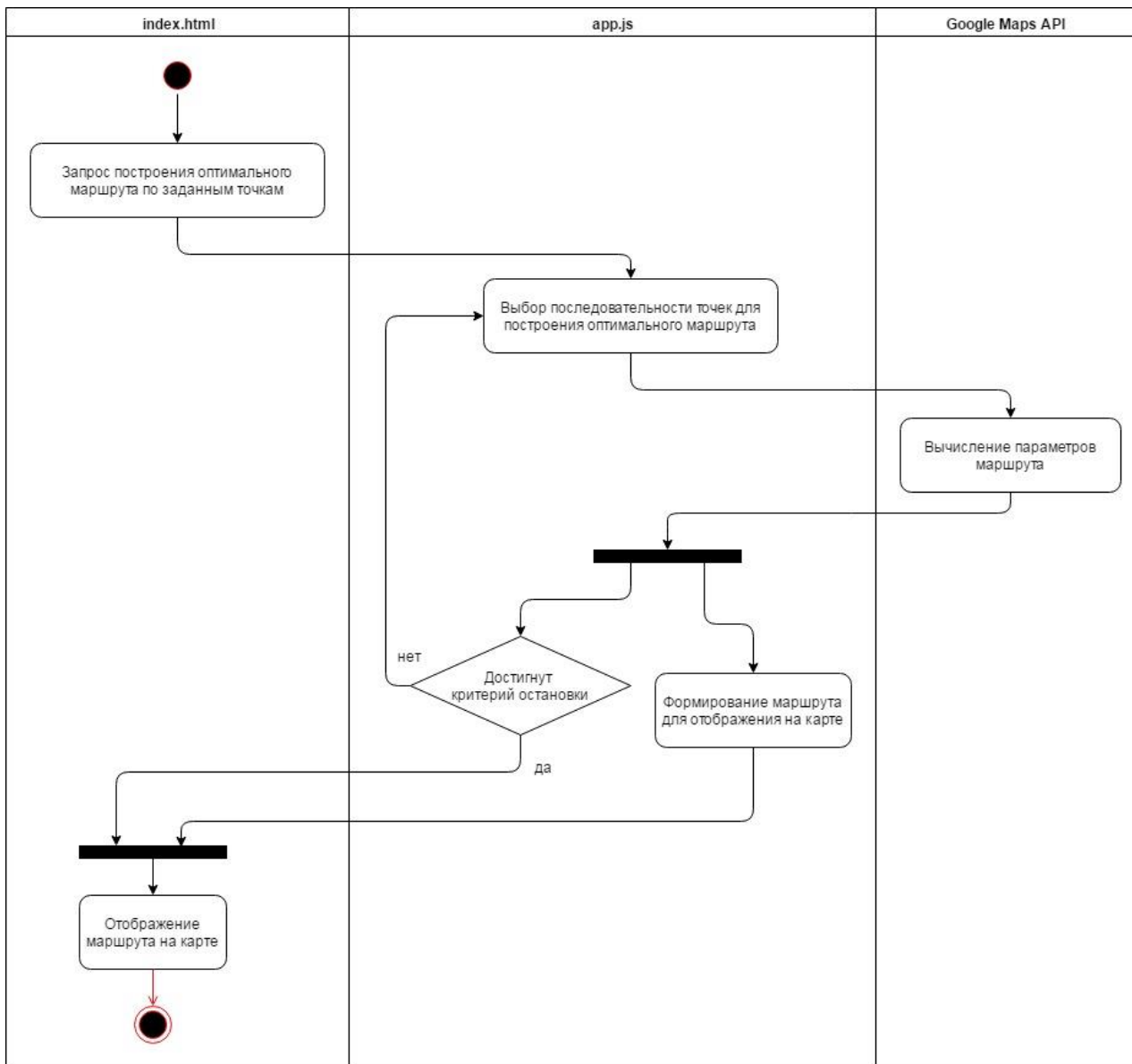


Рис. 6. Диаграмма деятельности.

Алгоритм построения маршрута представлен на рис.7.

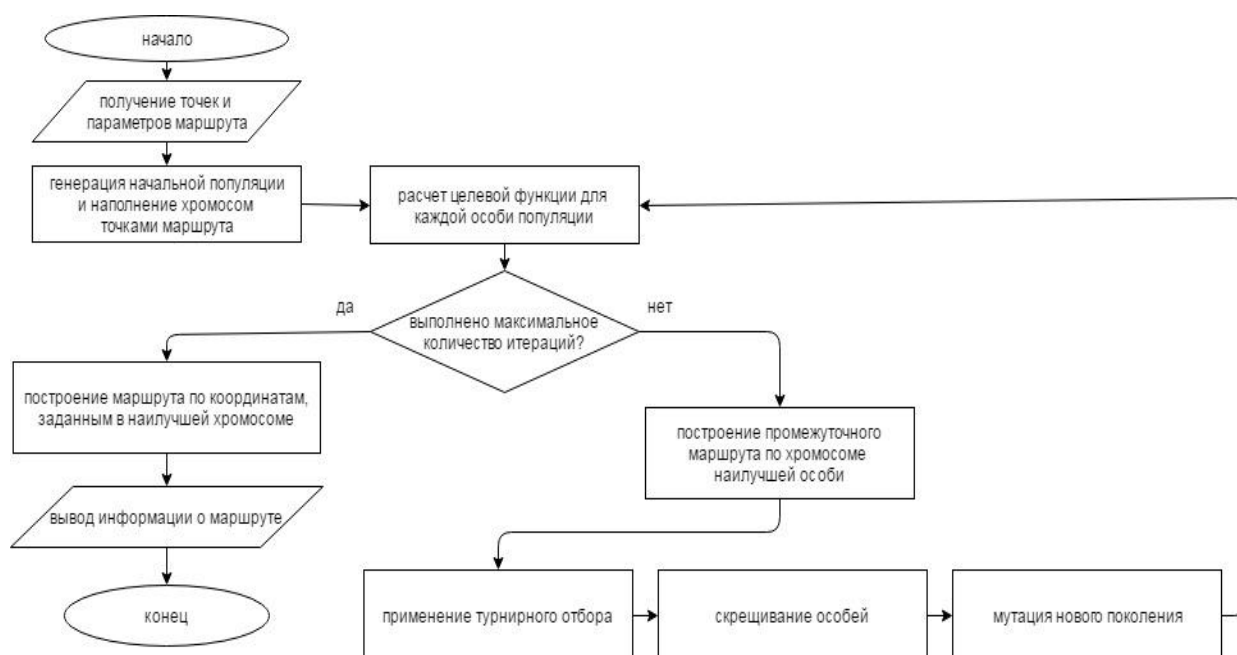


Рис. 7. Блок-схема алгоритма построения маршрута.

3. Службы «Google Maps API»

Для отображения карты в разметке страницы приложения создается элемент-контейнер, содержимое которого заполняется разметкой карты Google после инициализации карты.

Отображение начального маршрута на карте реализовано с помощью функции Google Maps API под названием «Polyline», прорисовывающей прямую ломанную линию между заданными точками.

Для отображения конечного маршрута на карте используется функция Google Maps API «setDirections».

Расчет целевой функции генетического алгоритма производится с помощью службы Google «Google Maps Distance Matrix API», на вход которой передаются параметры и точки маршрута.

Расчет длительности маршрута реализован с помощью функции «DistanceMatrixService» службы Distance Matrix. Функция выстраивает кратчайший (как заявлено разработчиками) путь между двумя точками с учетом заданных параметров построения маршрута. Параметры функции позволяют исключить возможность использования платных дорог при построении маршрута, а также учитывать «пробки» на дорогах.

4. Пользовательский интерфейс

В приложении использован односторонний интерфейс, взаимодействующий с пользователем с помощью вспомогательных модальных окон.

Первоначально приложение открывает карту Поволжья, и чтобы сократить процесс поиска текущего местоположения пользователя, необходимо совершить клик по надписи «Найти ваш город», в результате чего на карте будет отображен город, в котором находится пользователь (см. рис. 8).

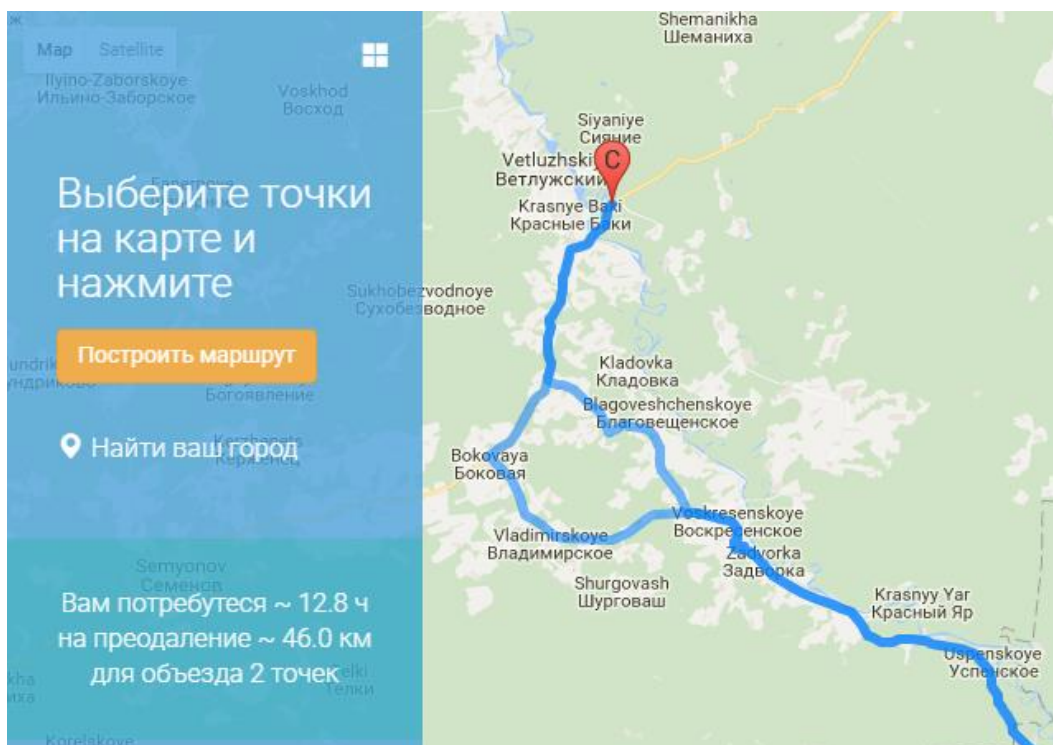
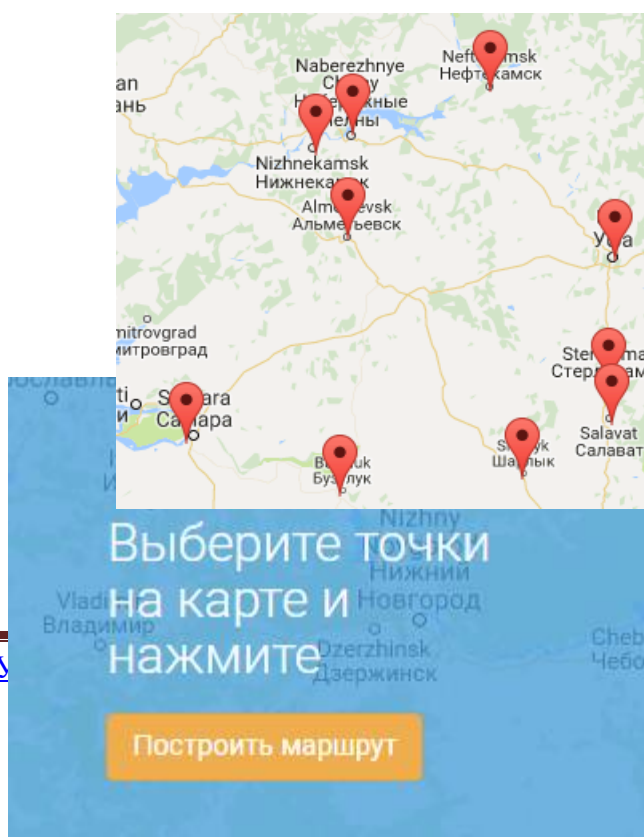


Рис. 8. Карта Поволжья.

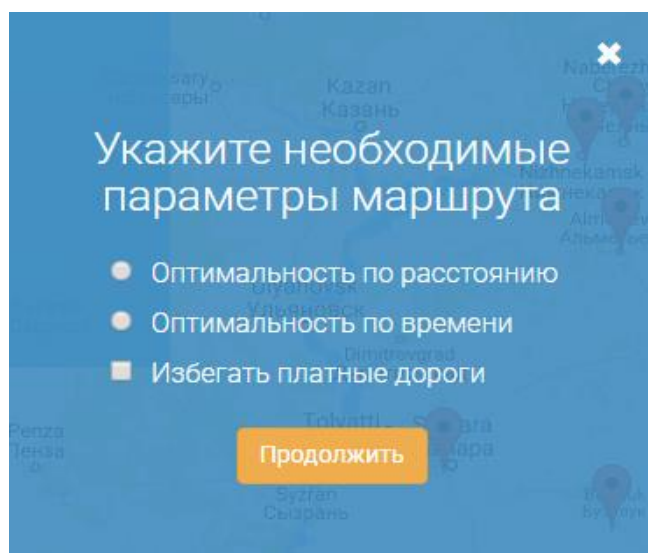


Для построения маршрута необходимо произвести клик в области карты, отметив необходимый пункт посещения. Данную операцию следует повторить необходимое количество раз, пока все пункты маршрута не будут заданы на карте красными маркерами (рис. 9).

Рис. 9. Указание точек маршрута.

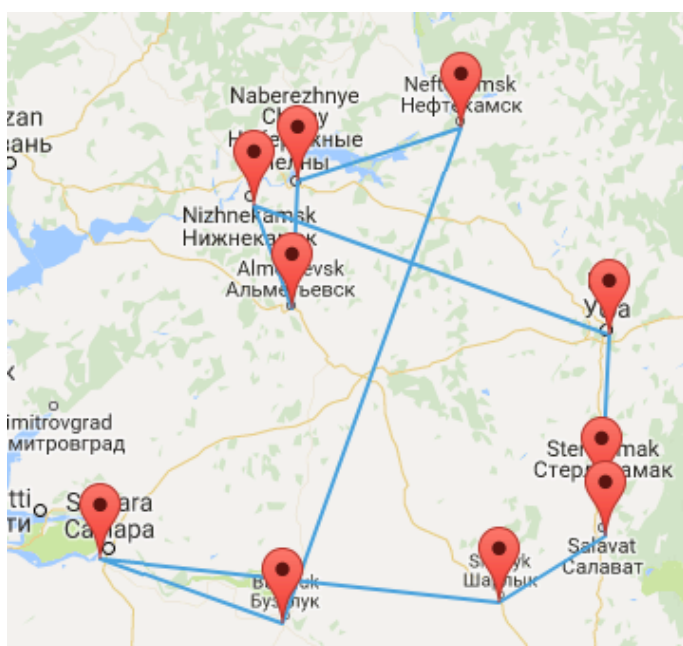
Далее необходимо произвести клик по кнопке «Построить маршрут» (рис. 10).

Рис. 10. Запуск построения маршрута.



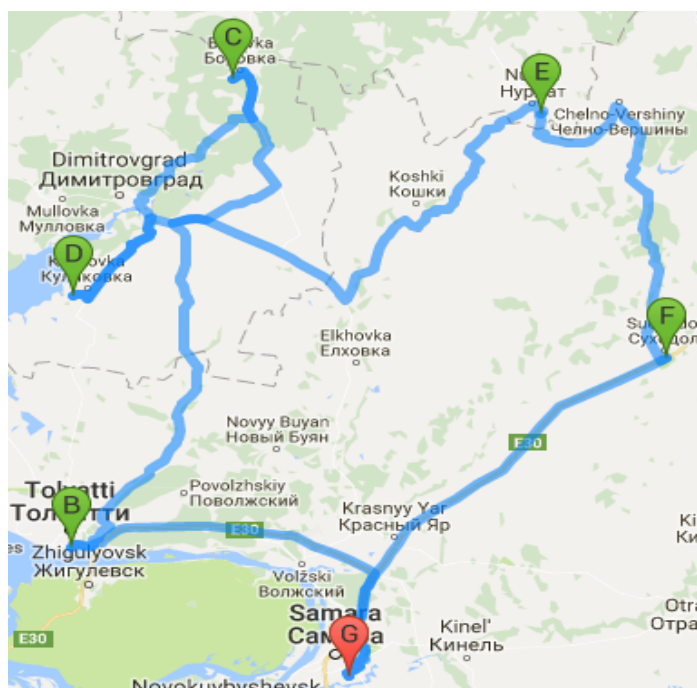
Если на карте задано не менее двух точек маршрута, приложение откроет модальное окно, предоставляющее выбор параметров построения наилучшего маршрута (рис. 11).

Рис. 11. Выбор параметров построения маршрута.



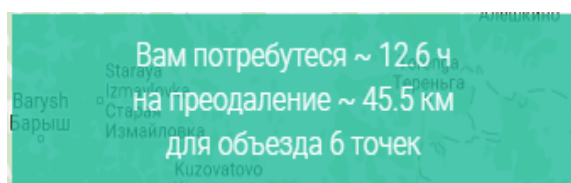
После задания параметров и совершения клика пользователем по кнопке «Продолжить», приложение начнет отображать процесс поиска наилучшего маршрута, соединяя точки маршрута ломанной линией (рис. 12).

Рис. 12. Процесс поиска оптимального маршрута.



Как только операция поиска будет завершена, приложение проложит оптимальный маршрут по дороге, соединяющей все точки маршрута (рис. 13). Красным маркером будет обозначена исходная позиция в маршруте.

Рис. 13. Окончательный маршрут.



Информация о предстоящем маршруте, расстояние маршрута и время, необходимое для безостановочного преодоления маршрута (рис. 14).

Рис. 14. Информация о маршруте.

Чтобы расширить область просмотра проложенного маршрута имеется возможность изменить вид панели управления. Для этого следует совершить клик по иконке в правом верхнем углу окна панели управления. Панель управления примет более компактный вид, который может быть изменен на прежний путем выполнения аналогичной операции.

Интерфейс карты представляет два режима отображения местности — в виде растровой карты и снимков со спутника. Переключая положение «Карта» в положение «Спутник» в верхнем левом углу карты, имеется возможность отобразить местность более подробно, данная функция особенно полезна при проезде грунтовых дорог в лесистой местности (рис. 15).

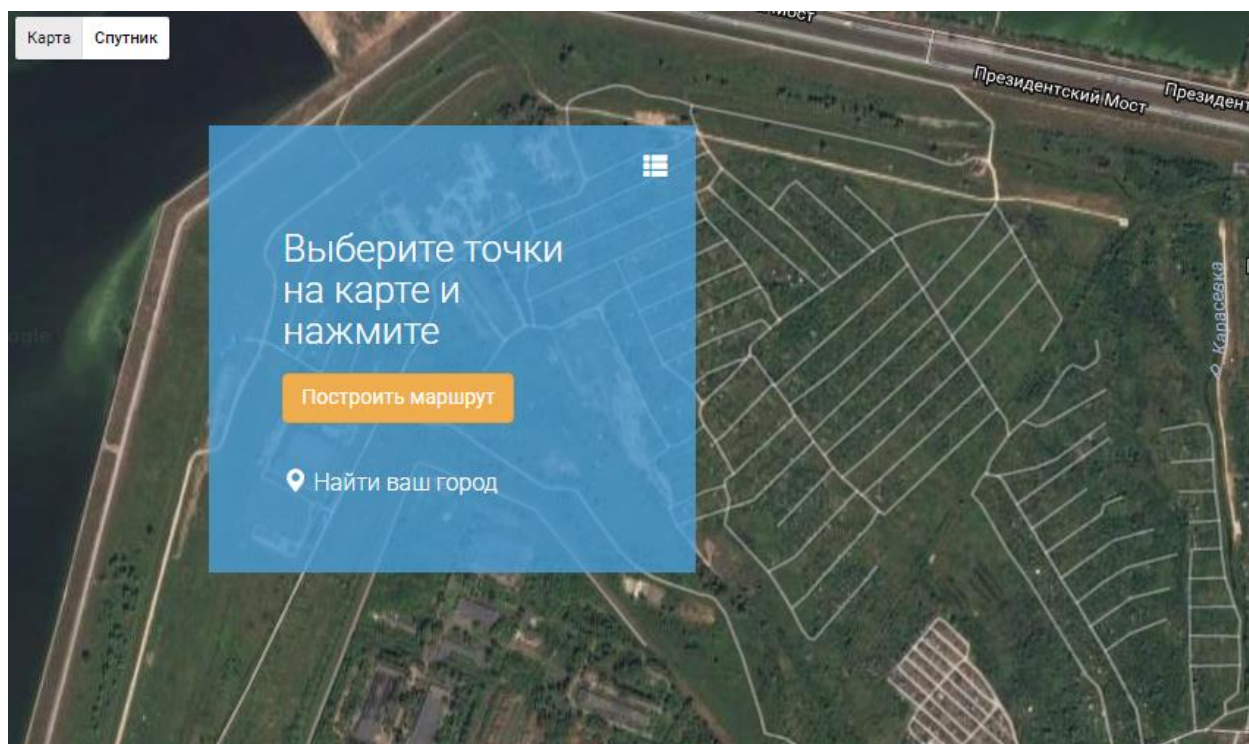


Рис. 15. Карта в режиме отображения «Спутник».

Для детального обзора местности следует переместить курсор мыши на иконку пользователя в нижнем правом углу карты и перетащить иконку на нужный участок карты. До того, как пользователь отпустит кнопку мыши, интерфейс карты отобразит название и фотографию местности (рис. 16).

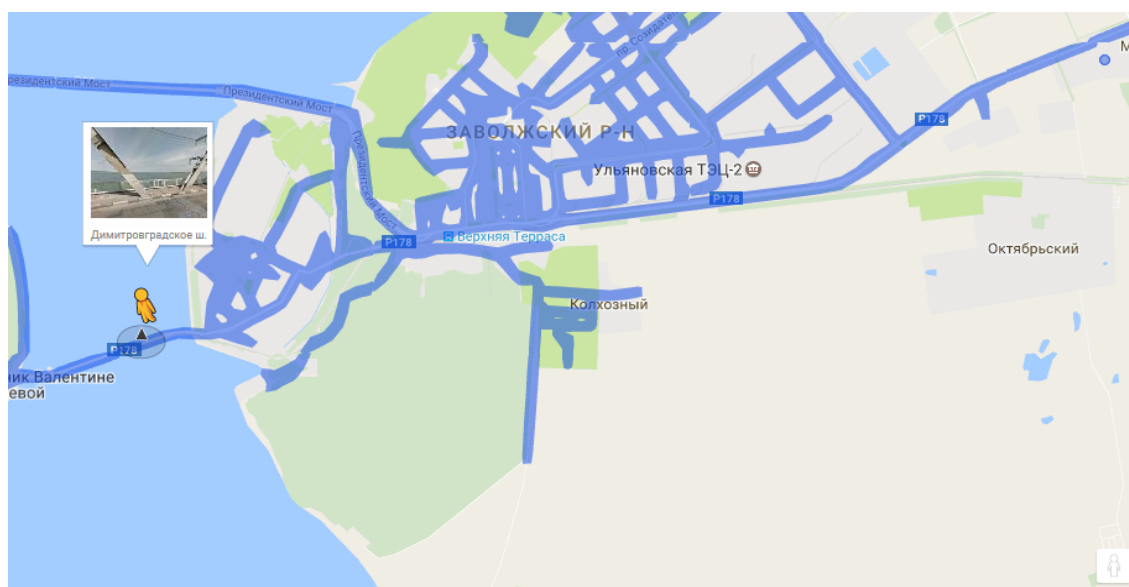


Рис. 16. Обзор местности.

Как только кнопка мыши будет отпущена, откроется 3D панорама данной местности, представленная сервисом Google Maps (рис. 17).



Рис. 17. 3D-панорама местности.

Заключение

Отличительной особенностью разработанного онлайн-сервиса является адаптивность к различным средствам отображения интерфейса приложения, применение генетического алгоритма к поиску наилучшего маршрута с предлагаемыми индексным подходом к кодированию особи и модифицированным оператором скрещивания, использование в приложении API Google Maps для отображения маршрута коммивояжера на карте, возможность выбора человеком параметров построения маршрута, что повышает мобильность и юзабилити сервиса.

Приложение может быть полезно для людей, занимающихся транспортной логистикой и туристической деятельностью, и размещено по адресу <http://u0160992.cp.regruhosting.ru/getroute/>.

Список литературы

1. Бронштейн Е. М., Заико Т. А. Детерминированные оптимизационные задачи транспортной логистики // *Автоматика и телемеханика*, 2010, №10, с. 123-124.
2. Гараба И.В, Сравнительный анализ методов решения задачи коммивояжера для выбора маршрута прокладки кабеля сети кольцевой архитектуры // *Молодежный*

научно-технический вестник ФГБОУ ВПО им. Н. Э. Баумана, 2013. №11. Режим доступа: <http://sntbul.bmstu.ru/doc/636966.html> (дата обращения 03.06.2018).

3. Гончарова А.Б., Поборчий И.В. Исследование методов решения задачи коммивояжера при управлении транспортными потоками предприятия // *Процессы глобальной экономики: Сборник научных трудов XX Всероссийской научно-практической конференции с международным участием. СПб. : Изд-во Политехн. ун-та, 2015, с. 418-419.*
4. Козлов М. В., Костюк Ф. В., Сорокин С. В., Тюленев А. В. Решение задачи коммивояжера методом целочисленного линейного программирования с последовательным исключением подциклов: описание и алгоритмическая реализация // *Научный журнал «Advanced science» ФГБОУ ВПО «Вятский государственный университет», 2013, № 1920, с. 124-141.*