

Пожидаев Михаил Сергеевич

АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ МАРШРУТИЗАЦИИ ТРАНСПОРТА

Научный руководитель: Костюк Юрий Леонидович

Применение автоматизированных систем в области транспортной логистики — один из способов экономии ресурсов при транспортировке грузов.

Рассматривается задача построения маршрутов для посещения заданного множества адресов некоторым количеством транспортных средств с возвращением в некоторое общее начальное местоположение.

Современные алгоритмы позволяют получать качественные результаты, но требуют точной подстройки многочисленных управляющих параметров и обладают чрезмерно высокой трудоёмкостью при большой размерности задачи (до 1000 вершин и более).

Общая постановка задачи маршрутизации транспорта (ЗМТ)

1. Задаётся $V = \{v_0, v_1, \dots, v_n\}$ — множество всех вершин.
2. v_0 — это вершина, в которой построенные маршруты должны начинаться и заканчиваться. Назовём её “депо”.
3. $V' = V \setminus \{v_0\}$ — это множество из n целевых вершин для посещения.
4. Задаётся симметричная или несимметричная матрица стоимостей переездов между вершинами.
5. Требуется построить m маршрутов транспортных средств минимальной суммарной стоимости, которые начинаются и заканчиваются в депо v_0 , и каждая вершина из V' должна быть включена в маршрут одного и только одного транспортного средства. Число m задаётся заранее или вычисляется в ходе работы.

Важные разновидности задачи

- ▶ ЗМТ с учётом грузоподъёмности, когда отслеживается грузоподъёмность транспортного средства при указанных потребностях в товаре целевых вершин;
- ▶ ЗМТ при одновременном ограничении грузоподъёмности и количества вершин в маршруте;
- ▶ ЗМТ для нескольких депо.

1. **Конструктивные алгоритмы:** выполняют постепенное построение решения, отслеживая рост его стоимости, но не имеют фазы дальнейшего улучшения.
2. **Двухфазные (кластерные) алгоритмы:** задача разбивается на две части — группировку вершин для каждого будущего маршрута (кластеризацию) и решение задачи коммивояжёра (ЗК) для каждой полученной группы:
 - ▶ сначала кластеризация, затем поиск решения ЗК;
 - ▶ сначала решение ЗК, а затем разделение на несколько маршрутов. ЗК решается для всех вершин исходного множества.
3. **Улучшающие алгоритмы:** сначала ведётся поиск некоторого решения, а затем делаются попытки обмена вершин (рёбер) внутри каждого маршрута или между маршрутами.

Список метаэвристик для решения ЗМТ

1. Поиск с исключениями.
2. Моделируемый отжиг.
3. Детерминированный отжиг.
4. Генетический алгоритм.
5. Алгоритм на основе муравьиных колоний.
6. Нейронные сети.

Метаэвристики в некоторых случаях позволяют находить очень качественные решения, но имеют высокую трудоёмкость при больших размерностях задачи и содержат управляющие параметры в описании, для которых требуется подбирать значения для каждого нового набора входных данных.

Цель работы

Цель работы — получение приближённых алгоритмов решения ЗМТ, способных выполнять построение маршрутов для входных данных, содержащих до 1000 вершин и более при использовании матрицы стоимостей переездов и до 1000000 вершин при использовании геометрической информации.

В рамках указанной цели была поставлена задача разработать и исследовать эффективные приближённые алгоритмы, дающие сбалансированное по количеству вершин в отдельных маршрутах решение ЗМТ при заранее указанном количестве транспортных средств, решение ЗМТ с учётом грузоподъёмности транспортных средств и/или с ограничением количества вершин в одном маршруте, а также в случае нескольких депо и для случая несимметричной матрицы расстояний.

Новая процедура кластеризации

Предлагается новая стратегия выполнения этапа кластеризации с трудоёмкостью $O(n^2)$, основанная на принципе дихотомического разделения вершин на группы с применением условия сбалансирования.

1. Производится поиск двух вершин с максимальной стоимостью переезда между ними. Они являются начальным наполнением групп.
2. Все вершины разделяются между двумя группами при помощи специального критерия оценки близости.
3. Если общее количество групп меньше желаемого количества транспортных средств или нарушается критерий грузоподъёмности, то процедура повторяется рекурсивно для одной или обеих групп.

Варианты задачи для решения новой процедурой

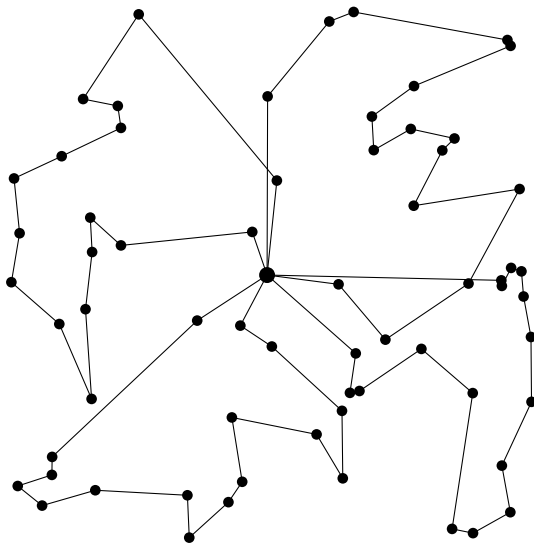
1. Новая постановка ЗМТ с условием сбалансирования количества вершин для каждой группы.
2. ЗМТ с учётом грузоподъёмности.
3. ЗМТ с учётом грузоподъёмности и ограничением количества вершин для каждого транспортного средства.
4. ЗМТ для нескольких депо.

Сбалансированная ЗМТ

Новая постановка задачи, называемая **сбалансированной ЗМТ**, имеет следующий вид:

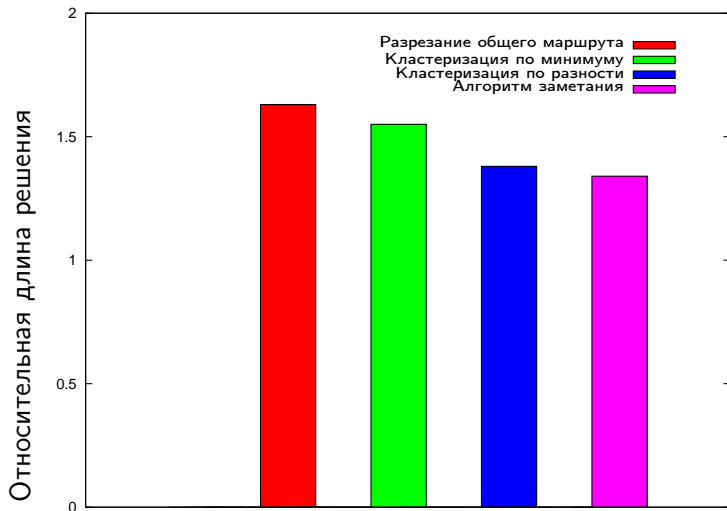
1. Пусть дано $n + 1$ вершин (n вершин-клиентов и одна вершина-депо), симметричная матрица стоимости переезда между всеми $n + 1$ вершинами и число k — желаемое количество маршрутов.
2. Вершина-депо должна входить во все результирующие маршруты.
3. Каждая вершина-клиент входит только в один результирующий маршрут.
4. Количество вершин для любой пары маршрутов не должно отличаться более, чем на единицу.
5. Суммарная стоимость объезда всех маршрутов должна быть минимальной.

Пример работы алгоритма кластеризации по разности



Пример посещения 64 вершин 4 транспортными средствами.

Качество разделения 128 вершин для 4 транспортных средств



Результаты тестирования алгоритма разрезания общего маршрута

Вершин / экипажей	2	4	8	16	32	64
32	1,59	1,98	-	-	-	-
64	1,51	1,80	2,35	-	-	-
128	1,43	1,63	2,01	2,78	-	-
256	1,38	1,55	1,84	2,37	3,49	-
512	1,33	1,44	1,64	2,02	2,80	4,42
1024	1,32	1,38	1,51	1,79	2,34	3,46

В таблице указана средняя длина решения относительно оценки снизу.

Результаты тестирования дихотомической кластеризации по минимуму

Вершин / экипажей	2	4	8	16	32	64
32	1,45	1,82	2,19	-	-	-
64	1,40	1,67	2,19	-	-	-
128	1,35	1,55	1,93	2,70	-	-
256	1,31	1,47	1,77	2,34	3,52	-
512	1,29	1,43	1,67	2,08	2,91	4,55
1024	1,30	1,39	1,58	1,91	2,52	3,68

В таблице указана средняя длина решения относительно оценки снизу.

Результаты тестирования дихотомической кластеризации по разности

Вершин / экипажей	2	4	8	16	32	64
32	1,31	1,60	-	-	-	-
64	1,29	1,47	1,94	-	-	-
128	1,28	1,38	1,70	2,44	-	-
256	1,26	1,33	1,55	2,06	3,14	-
512	1,25	1,29	1,44	1,79	2,54	4,11
1024	1,25	1,27	1,37	1,62	2,13	3,22

В таблице указана средняя длина решения относительно оценки снизу.

Результаты тестирования алгоритма заметания

Вершин / экипажей	2	4	8	16	32	64
32	1,31	1,56	-	-	-	-
64	1,26	1,41	1,92	-	-	-
128	1,28	1,34	1,65	2,50	-	-
256	1,27	1,29	1,48	2,00	3,38	-
512	1,25	1,28	1,37	1,66	2,59	4,68
1024	1,24	1,26	1,32	1,48	2,02	3,51

В таблице указана средняя длина решения относительно оценки снизу.

Выводы для сбалансированной ЗМТ

1. По качеству решения алгоритмы расположились в следующем порядке (от худшего к лучшему): “разрезание общего маршрута”, “дихотомическая кластеризация по минимуму”, “дихотомическая кластеризация по разности”. Алгоритм заметания превзошел все остальные при $k = 16$ и менее, но уступил другим при больших k .
2. При решении задачи в общем метрическом случае расстояний наиболее перспективным представляется использование алгоритма “кластеризация по разности”.

Вычислительный эксперимент для ЗМТ с учётом грузоподъёмности

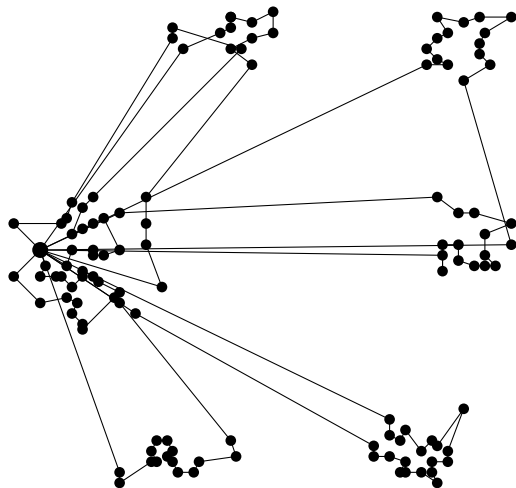
Производится сравнение следующих алгоритмов:

1. Алгоритм Османа поиска с исключениями.
2. Сбалансированный алгоритм.
3. Комбинированный алгоритм.

Наборы данных для тестирования:

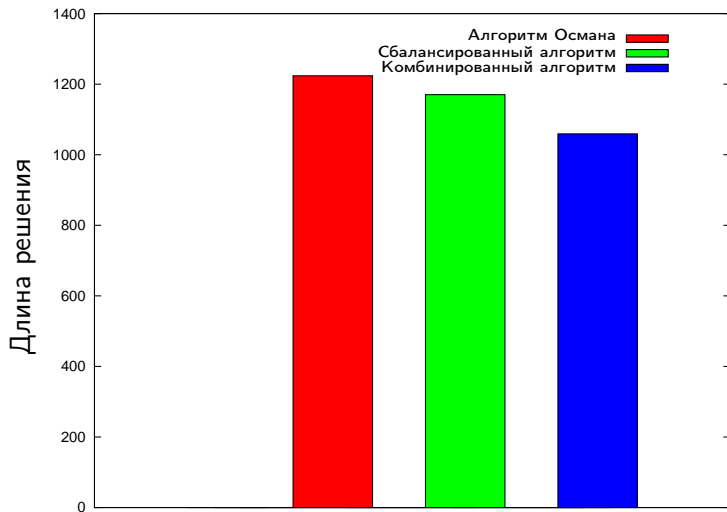
1. 7 из 14 эталонных задач Кристофидеса-Мингоззи-Тосса (КМТ).
2. Случайное равномерное расположение вершин внутри единичного квадрата. (количество выборок: для 50 вершин — 100, для 100 вершин — 50, для 150 вершин — 25, для 200 вершин — 10).

Пример решения задачи КМТ № 11 сбалансированным алгоритмом

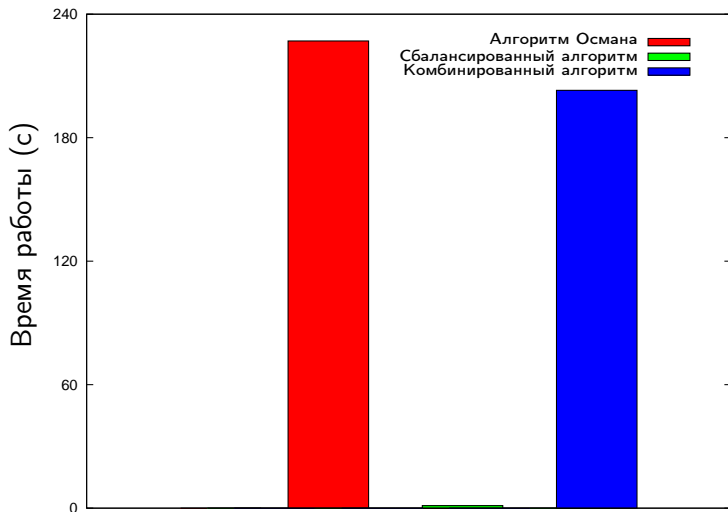


Маршрутов: 7; вершин: 120; грузоподъемность: 200; потребность в товаре: 2–28.

Качество решения задачи KMT № 11



Время решения задачи КМТ № 11



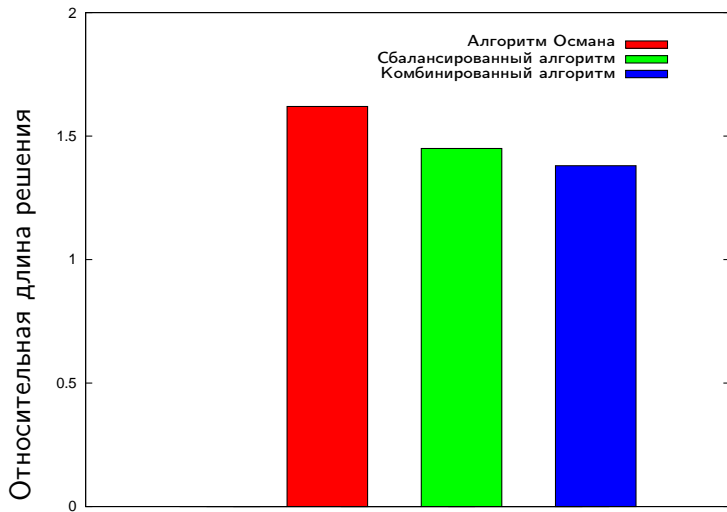
Вычислительный эксперимент проводился на системе
с процессором *Intel Core 2 Duo E8400* (3 ГГц).

Качество и время решения задач КМТ

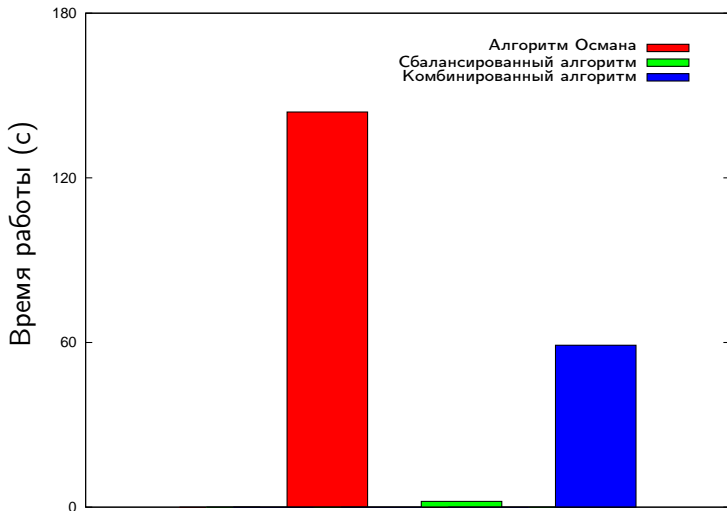
Задача, кол-во вершин	Алгоритм Османа	Сбал. алгоритм	Комб. алгоритм
1 (50)	537,6 (5 эк.), 9 с	559,7 (5 эк.), 0,07 с	558,2 (5 эк.), 6 с
2 (75)	885,1 (11 эк.), 35 с	937,9 (11 эк.), 0,11 с	881,6 (11 эк.), 26 с
3 (100)	867,2 (8 эк.), 161 с	1088,5 (8 эк.), 0,16 с	863,2 (8 эк.), 201 с
4 (150)	1122,5 (12 эк.), 993 с	1171,9 (12 эк.), 0,41 с	1078,9 (12 эк.), 682 с
5 (199)	1419,6 (17 эк.), 2749 с	1462,7 (17 эк.), 1,8 с	1379,4 (17 эк.), 3715 с
11 (120)	1223,7 (7 эк.), 227 с	1170,2 (7 эк.), 1,2 с	1059,2 (7 эк.), 203 с
12 (100)	905,2 (10 эк.), 220 с	1066,2 (10 эк.), 0,6 с	841,4 (10 эк.), 167 с

Вычислительный эксперимент проводился на системе
с процессором *Intel Core 2 Duo E8400* (3 ГГц).

Результаты моделирования для 200 вершин



Среднее время работы алгоритмов при моделировании



Вычислительный эксперимент проводился на системе
с процессором *Intel Core 2 Duo E8400* (3 ГГц).

Сравнение качества решения ЗМТ с учётом грузоподъёмности

Число вершин, грузоподъёмность	Алгоритм Османа	Сбаланс. алгоритм	Комбин. алгоритм
50 (50)	1,60/1,83/2,17	1,58/1,84/2,14	1,46/1,73/2,10
100 (50)	1,97/2,19/2,52	1,96/2,23/2,56	1,87/2,10/2,33
150 (50)	2,16/2,43/2,68	2,34/2,52/2,69	2,17/2,37/2,53
200 (50)	2,53/2,73/2,86	2,60/2,83/2,99	2,48/2,66/2,79
100 (100)	1,53/1,73/1,99	1,49/1,60/1,73	1,40/1,51/1,67
150 (150)	1,53/1,66/1,82	1,42/1,52/1,63	1,36/1,43/1,50
200 (200)	1,51/1,62/1,79	1,37/1,45/1,52	1,33/1,38/1,43

Приводятся длины решений относительно нижней оценки. Через косую черту перечислено минимальное, среднее и максимальное значение по всем выборкам.

Сравнение времени решения ЗМТ с учётом грузоподъёмности

Число вершин, грузоподъёмность	Алгоритм Османа	Сбалан. алгоритм	Комбин. алгоритм
50 (50)	0,5 с	0,04 с	0,5 с
100 (50)	3,6 с	0,2 с	2,7 с
150 (50)	23 с	0,5 с	17 с
200 (50)	87 с	0,9 с	58 с
100 (100)	5,5 с	0,3 с	4,1 с
150 (150)	33 с	0,8 с	19 с
200 (200)	144 с	2,1 с	59 с

Вычислительный эксперимент проводился на системе
с процессором *Intel Core 2 Duo E8400* (3 ГГц).

Выводы сравнения алгоритмов для ЗМТ с учётом грузоподъёмности

1. В случаях, когда количество вершин в каждом из полученных маршрутов в среднем менее, чем 10, сбалансированный алгоритм уступает алгоритму Османа поиска с исключениями по качеству решения в среднем от 0.5% до 3%.
2. Когда количество вершин в каждом из полученных маршрутов растёт с ростом общего числа вершин, картина противоположная, при этом преимущество сбалансированного алгоритма достигает более 10%.
3. Сбалансированный алгоритм затрачивает на вычисления в десятки раз меньше времени.
4. Применение комбинации этих двух алгоритмов всегда даёт выигрыш по сравнению с алгоритмом Османа.

Результаты тестирования для двух депо

Число маршрутов	16 вершин	32 вершины	64 вершин	128 вершин	256 вершин	512 вершин	1024 вершины
2	1,556/ 1,366	1,411/ 1,327	1,331/ 1,300	1,286/ 1,279	1,265/ 1,266	1,249/ 1,251	1,248/ 1,247
4	-	1,931/ 1,607	1,678/ 1,463	1,517/ 1,388	1,433/ 1,328	1,369/ 1,295	1,316/ 1,267
8	-	-	2,482/ 1,948	2,076/ 1,703	1,809/ 1,542	1,632/ 1,445	1,497/ 1,364
16	-	-	-	3,245/ 2,441	2,612/ 2,038	2,180/ 1,794	1,874/ 1,620
32	-	-	-	-	4,298/ 3,123	3,338/ 2,528	2,664/ 2,134
64	-	-	-	-	-	5,735/ 4,106	4,302/ 3,219
128	-	-	-	-	-	-	7,660/ 5,481

Депо располагались по углам диагонали квадрата. Через косую черту приведены результаты для одного депо (в центре квадрата) по тем же вершинам.

Результаты тестирования для четырёх депо

Число маршрутов	32 вершины	64 вершин	128 вершин	256 вершин	512 вершин	1024 вершины
4	1,617/ 1,607	1,447/ 1,463	1,358/ 1,388	1,314/ 1,328	1,276/ 1,295	1,264/ 1,267
8	-	1,992/ 1,948	1,711/ 1,703	1,539/ 1,542	1,437/ 1,445	1,368/ 1,364
16	-	-	2,497/ 2,441	2,057/ 2,038	1,789/ 1,794	1,611/ 1,620
32	-	-	-	3,177/ 3,123	2,550/ 2,528	2,135/ 2,134
64	-	-	-	-	4,162/ 4,106	3,242/ 3,219
128	-	-	-	-	-	5,548/ 5,481

Депо располагались в четырёх углах квадрата. Через косую черту приведены результаты для одного депо (в центре квадрата) по тем же вершинам.

Использование геометрической информации

При условии наличия геометрической информации и ограничения количества рёбер, выходящих из каждой вершины графа транспортной сети, время работы процедуры дихотомической кластеризации можно сократить до $O(n \log^2 n)$.

1. Поиск пары вершин с максимальной стоимостью переезда между ними выполняется при помощи нахождения максимально удалённых вершин в выпуклой оболочке.
2. Для вычисления минимальной стоимости переезда от двух выбранных вершин до всех остальных используется модифицированный алгоритм Дейкстры с применением структуры пирамиды.

Описанная процедура применяется только для предварительного деления вершин на группы, после чего используется обычный вариант на основе матрицы расстояний.

Основные этапы решения задачи построения маршрутов

1. Загрузка основных данных карты из так называемого *shape*-файла.
2. Обработка данных *shape*-файла и построение транспортного слоя на карте.
3. Редактирование данных карты для учёта текущей обстановки на улицах города.
4. Построения графа транспортной сети на основе данных транспортного слоя карты и с учётом расстояний или времени переездов
5. Выбор пунктов обслуживания (клиентов для объезда), депо, указание дополнительных атрибутов решения задачи.
6. Построение матрицы расстояний между всеми пунктами, включая депо.
7. Запуск алгоритмов решения ЗМТ для вычисления маршрутов.
8. Сохранение результатов.

Переход к симметричной матрице стоимости переездов

- ▶ При построении окончательного маршрута только метод ветвей и границ, пригодный для решения задач с не более, чем 25-30 вершинами, способен использовать несимметричную матрицу.
- ▶ Получаем симметричную матрицу путём выбора наименьшего значения среди двух направлений движения.
- ▶ Если отличие полученной матрицы от исходной не превышает $1 + \varepsilon$ раз и выполняется условие треугольника, то говорим о почти метрическом пространстве с точностью ε .
- ▶ При возможности пробного разворота для определения наилучшего направления на исходной матрице возврат к ней ухудшит результат не более, чем в $1 + \varepsilon/2$ раз.

1. Дерево слоёв.
2. Контейнер карты
3. Инспектор объектов.
4. Диалог настройки транспортной сети.
5. Диалог расчёта кратчайшего обхода.

Диалоговое окно запуска алгоритмов

Кратчайший обход пунктов

Количество экипажей
 Один Несколько Рассчитать

Задача маршрутизации транспорта [ЭМТ-Х]
Точность решения: Вместимость экипажа:

Скорость Качество

Посещаемые пункты
 Задать вручную по карте после закрытия данного окна
 Взять с точечного слоя:
Слой - пункты обхода

Названия пунктов: Потребности пунктов (для ЭМТ-Х):
НАЗВАНИЕ СПРОС

Способ передвижения:
 По улично-дорожной сети
 Запретить пересадки

Результат
 Нарисовать на карте временно
 Выдать на новый слой шейп-файла
 Выдать на существующий слой:
 Очистить перед выдачей

Отображать со снесением

Выдать текстовое описание найденного маршрута

OK Отмена

- ▶ Функция для решения сбалансированной ЗМТ:

```
int SolveBCVRP(size_t itemCount,  
               const double* rawMatrix,  
               size_t depoIndex,  
               size_t desiredVehicleCount,  
               size_t desiredQuality,  
               size_t* result);
```

- ▶ Функция для решения ЗМТ с учётом грузоподъёмности:

```
int SolveVRP(size_t itemCount,  
             const double* rawMatrix,  
             size_t depoIndex,  
             size_t desiredQuality,  
             size_t vehicleCapacity,  
             const size_t* quantities,  
             size_t maxResultLen,  
             size_t* result);
```

Научная новизна работы заключается в следующем:

1. Предложена модификация приближённого метода двухфазного решения ЗМТ на основе алгоритма сбалансированного дихотомического деления вершин на группы, позволяющая строить приближённые алгоритмы, обладающие трудоёмкостью $O(n^2)$, показывающие улучшение качества решений для задач, содержащих 200 вершин и более, по сравнению с распространёнными метаэвристиками.
2. Предложена модификация алгоритма сбалансированного деления вершин на группы с использованием геометрической информации, обладающая трудоёмкостью $O(n \log^2 n)$.
3. Получена оценка погрешности замены несимметричной матрицы стоимостей переездов симметричной.

Теоретическая значимость диссертационной работы заключается в следующем:

1. Предложенный в работе принцип сбалансированного дихотомического деления вершин на группы позволил разработать новый класс алгоритмов решения ЗМТ, обладающих низкой трудоёмкостью и высоким качеством решения в широком диапазоне объёмов входных данных.
2. Предложенный алгоритм способен послужить основой для разработки специализированных алгоритмов решения таких видов ЗМТ, как ЗМТ с ограничением на время доставки, ЗМТ с сопутствующим перевозчиком, ЗМТ с обратным грузом.

Практическая ценность диссертационной работы заключается в том, что созданное программное приложение пригодно для использования конечными пользователями для практического расчёта эффективных маршрутов в области транспортной логистики для задач, содержащих до 1000000 вершин и более.

Основные защищаемые положения

1. Модификация приближённого метода двухфазного решения ЗМТ на основе принципа сбалансированного дихотомического деления вершин на группы, а также алгоритмы, созданные на её основе, для решения сбалансированной ЗМТ, ЗМТ с учётом грузоподъёмности и/или с ограничением количества вершин в каждом маршруте, ЗМТ для нескольких депо.
2. Модификация алгоритма сбалансированного деления вершин на группы с использованием геометрической информации для решения различных видов ЗМТ.
3. Оценка погрешности замены несимметричной матрицы стоимости переездов симметричной.
4. Реализация предложенных алгоритмов решения различных видов ЗМТ в виде библиотеки классов.

Заключение

1. На основе изучения известных видов ЗМТ и алгоритмов её решения предложена сбалансированная ЗМТ.
2. Предложена процедура дихотомического разделения вершин, получены новые алгоритмы на её основе и исследована их эффективность. Предложенная процедура адаптирована для использования геометрической информации.
3. Оценена погрешность перехода от несимметричной матрицы к симметричной матрице стоимости переездов, необходимой для запуска алгоритмов.
4. Разработана библиотека с реализацией новых алгоритмов, предназначенная для решения ЗМТ в сбалансированном виде и с учётом грузоподъёмности, пригодная для включения в различные приложения.
5. Создана реализация в рамках геоинформационной системы *IndorGIS* с использованием разработанной библиотеки.