ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Январь—февраль 1111. Том 1, № 1. 3–??

УДК 519.87

ГИБРИДНЫЙ ЛОКАЛЬНЫЙ ПОИСК ДЛЯ ЗАДАЧИ МАРШРУТИЗАЦИИ ТРАНСПОРТНЫХ СРЕДСТВ С РАСПРЕДЕЛЕННЫМИ ПОСТАВКАМИ*)

Ю. А. Кочетов, А. В. Хмелев

Аннотация. В данной работе исследуется задача маршрутизации транспортных средств с разделенным обслуживание (SDVRP), вариант классической задачи маршрутизации транспортных средств, в которой запрос на поставку груза клиенту может быть разделен между несколькими транспортными средствами. Для этой NP-трудной задачи, мы показали, что существует оптимальное решение, которое может быть закодировано перестановкой клиентов. Это дало нам возможность разделить SDVRP на два подзадачи: найти наилучшую перестановку и найти наилучший набор маршрутов для заданной перестановки. Обе подзадачи являются NP-трудными. На основе этого подхода, мы разработали гибридную метаэвристику, которая сочетает в себе локальный спуск с чередующимися окрестностями и стохастический поиск с запретами для первой подзадачи, и два быстрых алгоритма декодирования для второй. Результаты расчетов показывают, что предложенный метод является очень конкурентоспособным. Алгоритм улучшает 23 известных наилучших найденных решения из 95 доступных тестовых примеров с числом клиентов до 288.

Ключевые слова: локальный поиск, метаэвристики, большие окрестности.

Введение

1. Введение

В классической задачи маршрутизации транспортных средств неограниченный автопарк одинаковых транспортных средств расположен на складе. Каждое транспортное средство имеет фиксированную вместимость. Маршрут транспортного средства должен начинаться и заканчиваться на складе. Каждый клиент имеет заданный запрос на доставку.

---

*)Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проекты 12–01–00077 и 14–01–313116)

Задача заключается в поиске набора маршрутов транспортных средств минимизируя суммарную длину маршрутов при условии, что все клиенты должны быть обслужены. В задаче маршрутизации транспортных средств с разделенным обслуживанием (SDVRP) каждый клиент может быть посещен несколько раз, и поставки по его запросу могут быть разделены между транспортными средствами. SDVRP является более реалистичной и может привести к экономии по отношению к классической модели [1]. Экономия может достигать 50%. Кроме того, это более общая модели, так как запросы клиентов могут превышать вместимость транспортных средств. Некоторые реальные приложения модели можно найти в [2].

SDVRP была представлена в [3, 4]. В последнее время ей уделяется большое внимание. Для этой, крайне сложной задачи комбинаторной оптимизации были предложены как точные так и приближенные алгоритмы. Обзор литературы доступен в [5]. В этой статье мы изучаем задачу и предлагаем гибридный эвристический метод, основанный на следующем свойстве задачи. Мы утверждаем, что существует оптимальное решение SDVRP которое может быть закодировано перестановкой клиентов. Следовательно, мы можем разбить задачу на две подзадачи: найти наилучшую перестановку и найти оптимальный набор маршрутов для данной перестаноски. Мы покажем, что каждая из этих подзадач являет NP-трудной. Они имеют различные структуры, и нужно использовать различные методы для них. Основываясь на этом наблюдении, мы разработали гибридную метаэвристику, которая сочетает в себе локальный спуск по чередующимся окрестностям (VND) и стохастический поиск с запретами (STS) для первой подзадачи и два быстрых алгоритма с линейной и квадратичной сложностью для второй подзадачи. Три типа окрестностей используются в методе VND. В первом типе окрестностей мы меняем перестановку посредством обмена, переноса и 2-опт ходов. Во втором типе, мы меняем решение путем создания или устранения разделенного обслуживания клиентов. В третьем типе, мы используем экспоненциальные по мощности цепи извлечений. Для второй подзадачи в основном используется квадратичная процедура декодирования. STS применяется для диверсификации, использует линейную процедуру декодирования и только небольшую часть от рандомизированной окрестности. Это помогает сменить район поиска перестановки. Для ускорения каждого шага STS, мы будем игнорировать шаги, которые приводят к длинным переездам между клиентами [6, 7].

В численных экспериментах использовались тестовые примеры из работ Archetti и др. [8], Chen и др. [2], и Belenguer [10]. Гибридный метод был сравнен с меметическим алгоритмом с управлением популяции (Boudia и др.)[11], поиском с запретами с построением словаря (Aleman и др.)[12], рандомизированным гранулированным поиском с запретами (Berbotto и др.)[7], поиском с разбросом (Mota и др.)[13] и методом генерации столбцов (Archetti и др.)[14]. Численные эксперименты показали что предложенный метод является конкурентоспособным для примеров большой размерности. Он улучшил 23 наилучших известных решения для данного множества тестовых примеров, размерностью до 288 клиентов.

Статья построена следующим образом. В разделе 2 введены некоторые обозначения и представлена математическая модель. В разделе 3 мы напоминаем известные свойства задачи и показываем, как разделить эту NP-трудную задачу на две других NP-трудных подзадачи. В разделе 4 мы описываем две полиномиальных процедуры декодирования для перестановок. В разделе 5 мы представляем четыре линейных окрестности, квадратичные окрестности, кубические окрестности, и окрестности цепей извлечения. В разделе 6 мы показываем известный метод VND где мы используем ранее описанные окрестности и квадратичную процедуру декодирования для расширенных перестановок. Этот метод позволяет находить локальные оптимумы высокого качества. Для перехода к другим, перспективным районам поиска мы применяем метод STS, который также представлен в разделе 6. В разделе 7 мы сообщаем результаты наших численных экспериментов. Выводы и направление дальнейших исследований завершают статью.

2. Постановка задачи и обозначения

Рассмотрим полный неориентированный граф $G = (V, E)$ на $n + 1$ вершине. Вершина 0 соответствует складу. Оставшиеся вершины соответствуют клиентам. Каждому клиенту i соответствует положительный запрос q_i. Каждому ребру (ij) из множества E соответствует неотрицательная длина c_{ij}. Предполагается что для всех вершин i, j, k из множества V выполняется неравенство треугольника $c_{ij} + c_{jk} \geqslant c_{ik}$. На складе находиться автопарк из m одинаковых транспортных средств. Каждое транспортное средство имеет положительную вместимость Q. Предполагается что автопарк достаточно большой (неограничен) и все клиенты могут быть обслужены. Маршрут каждого транспортного средства начинается и заканчивается на складе. Допускаются множественные посещения клиента. В задаче SDVRP необходимо найти такое множество маршрутов, чтобы все клиенты были обслужены и их суммарная длина была минимальной.

Представим задачу в терминах целочисленного линейного программирования со следующими переменными:

$$x_{ijk} = \begin{cases} 1 & \text{если транспортное средство } k \text{ едет напрямую от клиента } i \text{ к } j, \\ 0 & \text{иначе,} \end{cases}$$

$y_{ik} \geqslant 0$ количество единиц запроса, поставленного клиенту i транспортным средством k,

$u_{ik} \geqslant 0$ добавочная переменная для ограничения на подциклы.

Используя эти переменные, задача SDVRP может быть записана следующим образом:

$$\min \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=1}^{m} c_{ij} x_{ijk} \tag{1}$$

$$s.t. \quad \sum_{k=1}^{m} \sum_{i=0}^{n} x_{ijk} \geqslant 1, \quad j = 0, \ldots, n; \tag{2}$$

$$\sum_{i=0}^{n} x_{ijk} = \sum_{i=0}^{n} x_{jik}, \quad j = 0, \ldots, n; k = 1, \ldots, m; \tag{3}$$

$$u_{ik} - u_{jk} + (n+1) x_{ijk} \leqslant n, \quad i, j = 1, \ldots, n; k = 1, \ldots, m; \tag{4}$$

$$y_{ik} \leqslant q_i \sum_{j=0}^{n} x_{ijk}, \quad i = 1, \ldots, n; k = 1, \ldots, m; \tag{5}$$

$$\sum_{i=1}^{n} y_{ik} \leqslant Q, \quad k = 1, \ldots, m; \tag{6}$$

$$\sum_{k=1}^{m} y_{ik} = q_i, \quad i = 1, \ldots, n; \tag{7}$$

$$x_{ijk} \in \{0, 1\}, \quad i, j = 0, \ldots, n; k = 1, \ldots, m; \tag{8}$$

$$u_{ik}, y_{ik} \geqslant 0, \quad i = 1, \ldots, n; k = 1, \ldots, m. \tag{9}$$

Целевая функци (1) определяет суммарную длину всех маршрутов. Ограничение (2) гарантирует что каждого клиента посещает хотя бы одно транспортное средство. Ограничение (3) показывает что если транспортное средство k приезжает к клиенту j, то оно должно покинуть его. Ограничение (4) - известное ограничение на запрет подциклов. В заключении ограничения (5)–(7) определяют распределение единиц запросов клиентов среди транспортных средств. Согласно ограничению (5), клиент i

может быть обслужен только если транспортное средство посещает его. Ограничение (6) ограничивает максимальную загрузку транспортного средства величиной Q. Ограничение (7) гарантирует что запрос каждого клиента будет удовлетворен польностью.

Заметим, что мы можем рассматривать переменные u_{ik} как целочисленные и не более m. На самом деле они определяют перестановку клиентов для каждого транспортного средства. Ниже мы покажем что эти переменные могут быть заменены новыми переменными u_i, такими что $u_{ik} = u_i$ для всех $k = 1, \ldots, m$.

3. Свойства оптимальных решений

В этом разделе мы напомним некоторые известные свойства оптимальных решений задачи а также покажем новые. Пусть $(x^*_{ijk})(y^*_{ik})(u^*_{ik})$ оптимальное решение для задачи (1)–(9). Обозначим через $R^* = (r^*_1, \ldots, r^*_m)$ соответствующее ей множество циклов, $r^*_k = \{(i,j) \in E \mid x^*_{ijk} = 1\}, k = 1, \ldots, m$. Мы говорим что два цикла, например r^*_1 и r^*_2, содержат общее *разбиение* , если $y^*_{i1} > 0$ и $y^*_{i2} > 0$.

Теорема 1. *[4] Если (c_{ij}) удовлетворяет неравенству треугольника, тогда существует оптимальное решение задачи SDVRP, в котором никакие два цикла не имеют больше одного общего разбиения.*

Пусть $V_t = \{i_1, i_2, \ldots, i_t\} \subseteq V \setminus \{0\}$ и $t > 1$. Если существует t маршрутов транспортных средств таких что маршрут 1 содержит вершины i_1 и i_2, маршрут 2 содержит вершины i_2 и i_3, и т. д., и наконец, маршрут t содержит i_t и i_1, тогда V_t называется *t-split циклом.* Следующее утверждение обобщает предыдущее.

Теорема 2. *[4] Если (c_{ij}) удовлетворяет неравенству треугольника, тогда существует оптимальное решение задачи SDVRP, которое не содержит t-split циклов.*

Следующее свойство важно для дальнейшего рассмотрения. Положим снова что R^* - набор оптимальных маршрутов и обход по ним производиться один за другим в порядке r^*_1, r^*_2, и т. д. Другими слова, R^* - упорядоченное множество маршрутов. Мы говорим что вершина i является *недообслуженой* первыми k транспортными средствами, если $\sum_{k'=1}^{k} y^*_{ik'} < q_i$. Для маршрута r^*_k, пусть I^k - множество вершин в этом маршруте, которые недообслужены первыми k маршрутами из R^*, т.е.

$$I^k = \{i \in r_k | \sum_{k'=1}^{k} y^*_{ik'} < q_i\}, k = 1, ..., m.$$

Заметим что каждое оптимальное решение производит только одно множество I^k для каждого k.

Теорема 3. *[15] Если (c_{ij}) удовлетворяет неравенству треугольника, то всегда существует оптимальное решение задачи SDVRP и упорядоченное множество оптимальных маршрутов R^*, не содержащее t-split циклов, таких что $|I^k| \leqslant 1$ для всех $k = 1, \ldots, m$.*

Это свойство позволяет нам найти линейный порядок вершин, общий для всех маршрутов, который соответствует оптимальному решению. Пусть $\pi = \{i_1, \ldots, i_n\}$ - перестановка клиентов. Мы говорим что цикл r *соответствует* π если мы можем стартовать из депо перемещаться по маршруту r без нарушения линейного порядка в π. Говорим что множество маршрутов R соответствует π если каждый маршрут в R соответствует π.

Теорема 4. *Если (c_{ij}) удовлетворяет неравенству треугольника, тогда всегда существует перестановка клиентов π и упорядоченное множество R^* оптимальных маршрутов таких что R^* соответствует π.*

Доказательство. По теореме 3 у нас есть упорядоченное множество R^* оптимальных маршрутов, таких что $|I^k| \leqslant 1$ для всех $k = 1, \ldots, m$. Построим перестановку π используя это множество.

Рассмотрим первый цикл r_1^* в R^* и положим что вершина i_1 принадлежит I^1. Мы всегда можем представить этот маршрут как последовательность вершин $r_1^* = (\alpha_1, i_1, \alpha_2)$ для некоторых подпоследовательностей α_1 и α_2. Пусть теперь $i_1 \in I^2$ и $r_2^* = (\beta_1, i_1, \beta_2)$ для некоторых подпоследовательностей β_1, β_2. В этом случае мы можем заменить эти два маршрута новым множественным маршрутом $r_{12} = (\alpha_1, \beta_1, i_1, \alpha_2, \beta_2)$. Легко увидеть что r_1^* и r_2^* соответствуют этой последовательности вершин. Если $I^2 = \varnothing$, мы говорим что r_{12} является *терминальной* и удаляем её из R^*.

Теперь предположим что r_{12} не является терминальной и существует вершина i_2 из маршрута r_2^* которая принадлежит I^2. Если $i_2 \in r_3^*$, мы заменяем r_{12} и r_3^* на новый множественный маршрут r_{123} тем же самым образом. Если он терминальный, удаляем его из R^*. В противном случае, когда $i_2 \notin r_3^*$, находим первый маршрут в R^* с вершиной i_2, назовем его r_s, и заменяем r_{12} и r_s на новый множественный маршрут r_{12s} и вставляем его вместо r_s в R^*.

Алгоритм построения последовательности π состоит из $m - 1$ итерации. На каждой итерации мы рассматриваем первый маршрут r в оставшейся части R^*. Если r терминальный, мы удаляем его из R^*. В противном случае он содержит недообслуженую вершину i. Мы находим другой маршрут r' с минимальным порядковым номером в R^*, $i \in r'$ и заменяем r и r' на множественный маршрут так, как это было показано выше. Этот множественный маршрут вставляется в позицию r' в R^*. Т.к. r' имеет минимальный порядковый номер R^*, мы снова имеем $|I^k| \leqslant 1$ для всех $k \leqslant m$. Таким образом, мы можем удалить первый маршрут из R^* еще раз. Когда мы завершим все итерации, мы получим набор терминальных последовательностей. Необходимая нам перестановка получается путем конкатенации таких последовательностей в произвольном порядке. $\square$

Следствие 1. *Если (c_{ij}) удовлетворяет неравенству треугольника, тогда всегда существует оптимальное решение для задачи SDVRP, такое что $u^*_{ik_1} = u^*_{ik_2}$ для всех $i \geqslant 1$ и $k_1, k_2 = 1, \ldots, m$.*

Другими словами, мы можем уменьшить множество переменных и использовать новые переменные u_i, которые определены перестановкой π. Заметим, что подобная модель была рассмотрена в [9]. Здесь мы представили обоснование для такого подхода.

Это свойство показывает нам естественный способ решение задачи SDVRP: найти перестановка клиентов мощными методами для перестановочных задач и посчитать суммарную длину маршрутов для каждой перестановки. К сожалению последняя задача является сложной.

Теорема 5. *Если (c_{ij}) удовлетворяет неравенству треугольника, SDVRP является NP-трудной даже для заданной перестановки клиентов.*

Доказательство. Рассмотрим следующую задачу о разбиении. Дано n предметов с положительными весами w_i, $i = 1, \ldots, n$. Возможно ли разбить это множество на два подмножества A и $\overline{A}$ таких что $\sum_{i \in A} w_i = \sum_{i \in \overline{A}} w_i$? Это NP-трудная задача. Мы сведем эту задачу к задаче SDVRP с заданной перестановкой клиентов.

Пусть $Q = 0.5 \sum_{i=1}^{n} w_i$, $q_i = w_i$, $c_{ij} = 1$ for all $i, j = 0, \ldots, n$. Заметим что (c_{ij}) удовлетворяет неравенству треугольника. В этом случае два транспортных средства могут обслужить всех клиентов и все перестановки эквивалентны. Несложно увидеть что требуемое разбиение предметов существует тогда и только тогда, если суммарная длина оптимальных маршрутов равна $n + 2$. $\square$

Это свойство показывает что мы разделили SDVRP на две нетривиальные части. Первая часть является NP-трудной, т.к. задача коммивояжера - это частный случай задачи SDVRP. Вторая часть является NP-трудной по теореме 5. Эти две подзадачи имеют разные структуры и мы должны решать вторую задачу много раз для непредсказуемых последовательностей. Т.о. нам нужен быстрый декодирующий алгоритм для перестановки клиентов. В секции 3, мы представим для этого 2 процедуру: динамическое программирование и жадную эвристику.

4. Процедуры декодирования

Пусть π - произвольная перестановка клиентов.В задаче SDVRP каждый клиент может быть посещен несколько раз. Следуя [16], мы будем использовать расширенную перестановку μ с повтором клиентов вместо перестановки π. В этом случае нам нужен дополнительный вектор μ^q той же длины. Каждый элемент μ^q представляет количество единиц груза, доставленного клиенту. Заметим что сумма этих величин для каждого клиента должна быть равна его запросу. Для такой пары (μ, μ^q) мы можем посчитать суммарную длину маршрутов динамическим программированием [16]. Такая процедура разбиение будет иметь сложность $O(n^2)$. Очевидно, суммарная длина (μ, μ^q) может быть больше чем оптимальная длина для π т.к. мы разбили некоторые запросы в μ^q. Тем не менее, это полезная идея для локального поиска в методе VND, когда мы перемещаем клиентов в перестановке μ и изменяем некоторые разбиения.

Помимо этой процедуры, мы разработали быструю жадную эвристику с линейной временной сложностью. Мы будем использовать её для оценки соседних решений в методе STS. Основная идея проста и аналогична процедуре декодирования для классической задачи маршрутизации. Начиная с первого клиента в перестановке, мы разделяем её на маршруты, чтобы получить допустимое решение для SDVRP. Когда запрос клиента превышает остаточную вместимость транспортного средства, мы разделяем этот запрос и используем новое транспортное средство. В этой процедуре есть по крайней мере два слабых места. Во-первых, только последний автомобиль может быть загружен частично. Во-вторых, мы можем разделять требования только первого и последнего клиентов в маршрутах. Для улучшения этого подхода, введем некоторых фиктивных клиентов в расширенной перестановки. Посещение такого клиента бесплатно, но его запрос является положительным. Новая пара (η, η^q) с фиктивными клиентами будет использоваться для того же алгоритма. Чтобы создать пару (η, η^q) по паре (μ, μ^q) мы включаем фиктивных клиентов в конец каждого маршрута, если решение динамического программирования для (μ, μ^q) получает частично загруженные транспортные средства. Чтобы получить (μ, μ^q) из (η, η^q), мы просто удаляем всех фиктивных клиент.

5. Окрестности

Для расширенной перестановки μ, рассмотрим следующие окрестности, которые используются для задачи коммивояжера а также для задачи VRP.

Обмен: Обмен элементов i и j в μ. Для увеличения набора соседних решений мы также можем обмениваться парами $(i, i+1)$ или $(i-1, i)$ и j в μ.

2-опт: Выбираем элементы i и j в μ и инвертируем подпоследовательность от i до j.

Перенос: Выбираем элементы i и j в μ и переносим i или пару $(i, i+1)$, или $(i-1, i)$ перед j.

Эти окрестности содержат $O(n^2)$ элементов. Чтобы убрать из окрестности бесперспективные решения, мы создаем список ближайших соседей для каждого клиента и запрещаем перемещаться от него до какихлибо других. Длина такого списка постоянна.

Вторая группа окрестностей может разделять запросы клиентов или устранять некоторые разбиения. Эти окрестности вводятся в [11] для SDVRP. В первой окрестности убираем клиента i из всех маршрутов и вставляем его наилучшим образом без разбиения. Для этого мы формулируем вспомогательную задачу о ранце и решаем её эвристически. Пусть a_k - остаточная вместимость транспортного средства k и Δ_K минимальное подорожание маршрута k, если в него вставляется i. Новая булева переменная x_k равна 1, если маршрут k используется для вставки и 0 в противном случае. Оптимальное решение следующей задачи

$$\min\{\sum_{k=1}^{m} \Delta_k x_k | \sum_{k=1}^{m} a_k x_k \geqslant q_i, x_k \in \{0,1\}\}$$

показывает наилучшее посещение для этого клиента. Окрестность имеет $O(n)$ элементов.

Последние две окрестности в этой группе могу перемещать клиента в новый маршрут даже если все транспортные средства заполнены. В первой такой окрестности мы рассматриваем два элемента i и i' в μ из разных маршрутов k и k' соответственно. Если $\mu^q(i) < \mu^q(i')$, мы можем переместить i до или после i', разделить i' на два элемента, скажем i'_1 и i'_2, таких что $\mu^q(i') = \mu^q(i'_1) + \mu^q(i'_2)$ но $\mu^q(i'_1) = \mu^q(i)$, и вставляем i'_1 вместо i, и i'_2 вместо i'. Таким способом мы можем уменьшить суммарную длину маршрутов k и k'. Количество соседних решений здесь равно $O(n_\mu^2)$, где n_μ - длина μ. Во второй окрестности мы используем ту же самую идею для трех элементов из двух маршрутов.

В заключении, последняя окрестность это цепи извлечений [20]. Для задачи VRP она была применена в [22]. Это огромная окрестность которая может улучшить локальные оптимумы высокого качества в методе VND. Рассмотрим снова два элемента i и i' из разных маршрутов k и k'. *Извлечение* заключается в следующем: удаляем i из маршрута k; удаляем i' из маршрута k'; вставляем i в наилучшую позицию в k' без i'. Для такого перехода, пусть $w(i, i')$ - стоимость вставки i в k' учитывая что i' был извлечен из k'.

Во *вставке*, мы вставляем i в маршрут k' без требования какого-либо извлечения из этого маршрута. Для такого перехода мы определяем стоимость $w(i, k')$ как подорожание маршрута k' минус экономия, возникшая при удалении i из маршрута k. Заметим что такая стоимость может быть отрицательной.

Основываясь на этих переходах, мы строим *граф извлечений,* где вершины соответствуют клиентам и маршрута а дуги соответствуют переходам извлечения и вставки. Такой граф имеет слоистую структуру. Слой состоит из всех клиентов обслуживаемых в текущем маршруте. Также существуют фиктивные вершины на каждом слое, которые представляют непосредственно сам маршрут. Межслойные дуги могут быть двух типов в зависимости от перехода: дуги извлечений (i, i') стоимостью $w(i, i')$ и дуги вставки (i, k') стоимости $w(i, k')$. Наша цель найти кратчайший путь в графе, учитывая что мы можем извлекать не более одного клиента в каждом маршруте. Это NP-трудная задача. Следуя [22], мы применили адаптацию алгоритма Флойда-Уоршалла для нахождения всех кратчайших путей в графе.

6. Гибридный VND–STS метод

В этой секции мы представляем наш гибридный метод для задачи SDVRP, которые сочетает в себе метод спуска по чередующимся окрестностям[18] для нахождения локальных оптимумов высокого качества и метод поиска с запретами [17] для исследования новых районов в области допустимых решений.

Пусть N_l, $l = 1, \ldots, l_{max}$ - конечное число окрестностей. Для каждой пары (μ, μ^q), каждая окрестность определяет $N_l(\mu, \mu^q)$ соседних решений. Поскольку локальный оптимум для одной окрестности не всегда является локальным оптимумом для другой окрестной, регулярная смена окрестности при поиске может значительно повысить его эффективность. Это простая но работающая идея используется в методе VND который представлен в следующем псевдокоде на Рисунке 1.

1. Зададим окрестности $N_l, l = 1, \ldots, l_{max}$ и стартовое решение (μ, μ^q).
2. Повторяем следующие шаги пока возможно улучшение:
   2.1 Задаем $l \leftarrow 1$;
   2.2 Повторяем пока $l \neq l_{max}$
      (a) Находим наилучшее решение $(\overline{\mu}, \overline{\mu}^q)$ в $N_l(\mu, \mu^q)$;
      (b) Если $(\overline{\mu}, \overline{\mu}^q)$ лучше чем (μ, μ^q) тогда $(\mu, \mu^q) \leftarrow (\overline{\mu}, \overline{\mu}^q)$ and $l \leftarrow 1$
      иначе $l \leftarrow l + 1$;
3. Возвращаем (μ, μ^q) - локальный оптимум для всех N_l окрестностей.

Fig.1 Псевдокод метода VND

Стартовое решение строится алгоритмом Кларка–Райта [21] и представляется в виде пары (μ, μ^q). Окрестности используются в том же порядке, в котором были представлены в предыдущей секции. На Шаге 2.2 (a) мы используем правило *первого улучшение* для сокращения времени вычисления одной итерации.

В нашем предварительном тестировании метода, мы обнаружили, что, как правило, конечное решение содержит множество разделенных поставок. Очень трудно найти лучшее решение около такого локального оптимума. Его качество зависит от исходного решения. Таким образом, нужна мощная процедура диверсификации, чтобы улучшить метод. Простая процедура *встряски* из VNS здесь не работает. Таким образом, мы применяем метод стохастического поиска с запретами [23] для этих целей.

Пусть $N(\eta, \eta^q)$ - объединение окрестностей обмен, 2-опт, и перенос для пары (η, η^q). Мы определим рандомизированную окрестность $N_p(\eta, \eta^q)$ для заданного положительного параметра p, $0 < p < 1$, как подмножество $N(\eta, \eta^q)$. Каждый элемент окрестности $N(\eta, \eta^q)$ включен в множество $N_p(\eta, \eta^q)$ с вероятностью p независимо от других элементов. Мы используем эту рандомизированную окрестность для небольших значений p. Заметим что это множество может быть пустым или содержать только один элемент. Окрестности Обмен, 2-опт, и перенос оперируют двумя клиентами. таким образом мы можем использовать такие пары в списке запретов с дополнительной информацией о типе перехода. Длина списка запретов L_{tabu} является константой. Псевдокод процедуры STS представлен ниже на Рисунке 2.

1. Генерируем стартовое решение (η, η^q) и полагаем список запретов пустым.

2. Повторяем T_{max} итераций.

 2.1 Генерируем $N_p(\eta, \eta^q)$ исключая все запрещенные решения;

 2.2 Если множество пусто, переходим на Шаг 2.1;

 2.3 Находим наилучшее соседнее решение $(\overline{\eta}, \overline{\eta}^q)$;

 2.4 Заменяем $(\eta, \eta^q) \leftarrow (\overline{\eta}, \overline{\eta}^q)$ и обновляем список запретов;

3. Возвращаем наилучшее найденное решение.

Fig.2 Псевдокод процедуры STS

В качестве начального решения мы используем локальный оптимум полученный методом VND. Параметры T_{max}, p, и L_{tabu} мы определяем таким образом, чтобы значительно изменить начальное решение и получить новое стартовое решение высокого качества для процедуры VND.

7. Численные эксперименты

В этом разделе представлены численные эксперименты для гибридного метода VND-STS. Все эксперименты были проведены на ПК с процессором Core i7, 2.20 ГГц, 8Гб оперативной памяти. Метод был реализован на языке программирования Java. Эксперименты проводились на тестовых примерах Archetti и др. [8], Belenguer и др. [10] и Chen и др.[2]

- Множество1: 49 тестовых примеров [8]. Эти примеры получены из файлов 1 - 5 и 11 - 12 [19] для задачи VRP. Число клиентов варьируется от 50 до 199 а вместимость грузовика Q варьируется от 140 до 200. Запросы q_i были сгенерированы случайным образом используя шесть различных интервалов относительно Q (см. Таблицу **??**, второй столбец). Первая группа из этих семи примеров содержит оригинальные запросы.

- Множество2: 11 тестовых примеров [10]. Координаты клиентов взяты из библиотеки примеров TSPLIB: *eil51, eil76, eil22, eil23, eil30, eil33, eil101* без округления расстояний. Вместимость Q варьируется от 100 до 4000, q_i сильно зависит от Q.

- Множество3: 14 тестовых примеров [10]. Координаты клиентов взяты из библиотеки примеров TSPLIB *eil51, eil76, eil101*, но запросы клиентов сгенерированы случайно, $Q = 160$.

- Множество4: 21 тестовый пример [2]. Клиенты расположены на концентрических окружностях вокруг склада. Число клиентов лежит в диапазоне от 8 до 288, $Q = 100$, и запросы q_i либо 60 либо 90.

Исходя из наших предварительных тестов мы установили следующие параметры для гибридного метода. Для каждого клиента выбирается 25 ближайших соседей и запрещаются переезды к другим. Хорошее соотношение качества решения и времени выполнения получается, когда мы используем метод STS для диверсификации 15 раз, $p = 0.05$, $L_{tabu} = \lceil n/2 \rceil$, $T_{max} = 3000$. Таблицы **??**–**??** показывают сравнение гибридного метода и следующих методов:

- Tabu search with vocabulary building approach (TSVBA) [12], ПК 2.40 ГГц, 512 Мб RAM.
- Scatter Search (SS) [13], ПК 2.40 ГГц, 1Гб RAM.
- Memetic algorithm with population management(MA|PM) [11], ПК 3.0 ГГц.
- A Randomized Granular Tabu Search Heuristic (RGTS) [7], ПК 2.10 ГГц, 4Гб RAM.
- Heuristic with limited fleet (HLF) and unlimited fleet (HUL) [14], ПК 2.40 ГГц, 3Гб RAM.

Таблицы **??**–**??** показывают результаты гибридного метода на тестовых примерах из Множества 1. Все наилучшие известные решение здесь получены методом MA|PM. Наш гибридный метод смог получить новые лучшие решения для 13 примеров. В большинстве из них мы имеем большое число клиентов $n \geqslant 100$. Экономия $\varepsilon = 100\%(Hybrid - Best)/Best$ невелика, $\varepsilon \leqslant 1.5\%$, и мы понимаем что глобальный оптимум близок к этим значениям.

Второй эксперимент был проведен с 25 примерами из Множеств 2 и 3. Результаты вычислений представлены в Таблице **??** и **??**. Гибридный метод немного улучшает 6 наилучших известных решений: 2 примера из Множества 2 и 4 примера из Множества 3. Снова, мы имеем большое число клиентов: $n = 100$ в большинстве из них. Экономия мала, $\varepsilon \leqslant 1.5\%$, но отклонение от нижней оценки (LB для HUF) достаточно велико. Таким образом, у нас есть промежуток между нижней и верхней оценкой и мы можем улучшить результаты для этих примеров.

Последний эксперимент был проведен с 21 примером из Множества 4.Таблица **??** показывает что наш метод улучшает 4 наилучших известных решения для больших n.

8. Заключение и дальнейшие исследования

В этой работе мы изучили известную задачу маршрутизации транспортных средств с разделенным обслуживанием и представили некоторые теоретические и экспериментальные результаты для нее. Мы показали что некоторые оптимальные решения могут быть закодированы последовательностью клиентов. В результате появилась возможность сократить число переменных в математический формулировке. Это открыло новые направления для исследований связанных с мат-эвристиками и гибридами с классическими математическими инструментами. Для декодирования перестановки разработано два приближенных полиномиальных алгоритма, в связи с тем что эта проблема NP-трудная. Подзадача декодирования интересна сама по себе и также является предметом дальнейших исследований. В заключении мы представили гибридный метод, который сочетает классический VND и STS для диверсификации. Наши эмпирические результаты показали что стадия диверсификации важна для примеров большой размерности. Было улучшены наилучшие известные решения для 23 тестовых примеров.

ЛИТЕРАТУРА

1. **Archetti C, Savelsbergh M, Speranza MG.** Worst-case analysis for split delivery vehicle routing problems // Transp. Sci. — 2006. — Vol. 40, N 2. — P. 226–234.
2. **Chen S, Golden B, Wasil E.** The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results // Networks. — 2007. — Vol. 49, N 4. — P. 318–329.
3. **Dror M, Trudeau P.** Savings by split delivery routing // Transportation Science. — 1989. — Vol. 23, N 2. — P. 141–145.
4. **Dror M, Trudeau P.** Split delivery routing // Naval Research Logistics. — 1990. — Vol. 37, N 3. — P. 383–402.
5. **Archetti C, Speranza MG,** The split delivery vehicle routing problem: A survey. — New York: Springer-Verlag, 2008. — P. 103–122.
6. **Johnson DS, McGeoch LA.** The traveling salesman problem: a case study. //Local Search in Combinatorial Optimization. — Jonh Wiley & Sons Ltd., 1997. — P. 215-310.

7. **Berbotto L, Garcia S, Nogales FG.** A randomized granular tabu search heurstic for the split delivery vehicle routing problem // Annals of Oper. Res.— 2013. — DOI 10.1007/s10479-012-1282-3
8. **Archetti C, Hertz A, Speranza MG.** A tabu search algorithm for the split delivery vehicle routing problem // Transp. Sci. — 2006. — Vol. 40, N 1. — P. 64–73.
9. **Frizzell P, Giffin J.** The bounded split delivery vehicle routing problem with grid network distances // Asia-Pacific Journal of Operational Research. — 1992. — Vol. 9. — P. 101–116.
10. **Belenguer J, Martinez M, Mota E.** A lower bound for the split delivery vehicle routing problem // Oper. Res. — 2000. — Vol. 48, N 5. —P. 801–810.
11. **Boudia M, Prins C, Reghioui M.** An effective memetic algorithm with population management for the split delivery vehicle routing problem // Proc. HM–2007 — Berlin: Springer-Verl., 2007. — P. 16–30. (Lect. Notes Comp. Sci.; Vol. 4771)
12. **Aleman RE, Hill RR.** A tabu search with vocabulary building approach for the vehicle routing problem with split demands // Int. J. Metaheuristics. — 2010. — Vol. 1, N 1. — P. 55–80.
13. **Mota E, Campos V, Corberan A.** A new metaheuristic for the vehicle routing problem with split demands // Proc. EvoCOP–2007. — Berlin: Springer-Verl., 2007. — P. 121–129. (Lect. Notes Comp. Sci.; Vol. 4446)
14. **Archetti C, Bianchessi C, Speranza MG.** A column generation approach for the split delivery vehicle routing problem // Networks. — 2011. — Vol. 58. N 4. — P. 241–254.
15. **Lee CG, Epelman MA, White CC, Bozer Y.** A shortest path approach to the multiple-vehicle routing problem with split pick-ups // Transp. Res., Part B. — 2006. — Vol. 40, N 4. — P. 265–284.
16. **Prins C.** A simple and effective evolutionary algorithm for the vehicle routing problem // Comput. Oper. Res. — 2004. Vol. 31, N 12. — P. 1985–2002.
17. **Glover F, Laguna M.** Tabu Search. — Kluwer Academic Publishers, 1997.
18. **Mladenović N, Hansen P.** Variable neighborhood search // Comput. Oper. Res. — 1997. — Vol. 24, N 11. — P. 1097–1100.
19. **Gendreau M, Dejax P, Feillet D, Gueguen C.** Vehicle routing with time windows and split delivery. Technical Report N 851. —

Laboratoire Informatique d'Avignon, France, 2006

20. **Glover F.** New ejection chain and alternating path methods for traveling salesman problems // Computer Science and Operations Research: New Developments in their Interfaces, — Oxford: Pergamon Press, 1992. — P. 449–509.

21. **Clarke G, Wright J.** Scheduling of vehicles from a central depot to a number of delivery points // Oper. Res. — 1964. — Vol. 12, N 4. — P. 568–581.

22. **Potvin JY, Naud MA.** Tabu Search with ejection chains for the vehicle routing problem with private fleet and common carrier // J. Oper. Res. Soci. — 2011. — Vol. 62. — P. 326-336.

23. **Davydov I, Kochetov Y, Mladenović N, Urosevic D.** Fast metaheuristics for the discrete $(r|p)$-centroid problem // Automation and Remote Control. — 2014. N 4. P.