

ГИБРИДНЫЙ АЛГОРИТМ ЛОКАЛЬНОГО ПОИСКА
ДЛЯ ЗАДАЧИ МАРШРУТИЗАЦИИ РАЗНОРОДНОГО
ОГРАНИЧЕННОГО АВТОПАРКА *)

Ю. А. Кочетов^{1,2}, А. В. Хмелёв²

¹Институт математики им. С. Л. Соболева,
пр. Коптюга, 4, 630090 Новосибирск, Россия

²Новосибирский гос. университет,
ул. Пирогова, 2, 630090 Новосибирск, Россия
e-mail: jkochet@math.nsc.ru, avhmel@gmail.com

Аннотация. Рассматривается задача оптимизации маршрутов разнородных транспортных средств для обслуживания заданного множества клиентов. Предполагается, что клиенты представлены точками на плоскости, а число транспортных средств каждого типа ограничено. Для решения задачи разработан гибридный алгоритм локального поиска с кодировкой решений в виде последовательности клиентов. Для декодирования последовательности соответствующая NP-трудная задача решается методом лагранжевых релаксаций. Предложены новые процедуры интенсификации и диверсификации поиска, а также новая окрестность экспоненциальной мощности. Приводятся результаты численных экспериментов на известных тестовых примерах с числом клиентов до 255. Для 15 примеров получены новые рекордные значения целевой функции. Табл. 7, ил. 5, библиогр. 26.

Ключевые слова: локальный поиск, экспоненциальная окрестность, лагранжева релаксация, субградиентная оптимизация.

Введение

Задачи маршрутизации транспортных средств — активно развивающаяся область исследования операций. Благодаря многочисленным приложениям рождаются всё новые и новые математические модели, разрабатываются оригинальные численные методы и проводятся специализированные международные конференции. В настоящей работе исследуется задача поиска оптимальных маршрутов разнородных транспортных

*) Исследование выполнено при финансовой поддержке Российского гуманитарного научного фонда (проект 13-22-10002).

средств ограниченной грузоподъёмности. Предполагается, что каждый тип транспортного средства имеет фиксированную стоимость и удельную стоимость проезда (стоимость бензина на километр). Число транспортных средств каждого типа ограничено. Все транспортные средства находятся в депо (на складе) и должны вернуться туда после обслуживания клиентов. Каждый клиент имеет определённый запрос на доставку грузов и обслуживается ровно одним транспортным средством. Склад и клиенты представлены точками на плоскости. Требуется найти маршруты транспортных средств для доставки грузов всем клиентам с минимальными суммарными затратами.

Наряду с данной задачей (Heterogeneous Fixed Fleet Vehicle Routing Problem, HFFVRP) часто рассматривают вариант с неограниченным автопарком, смешанный вариант и вариант с идентичными транспортными средствами. Все они NP-трудны и активно исследуются как теоретически, так и эмпирически для получения эффективных алгоритмов. Обзор последних результатов можно найти, например, в [5].

При разработке алгоритмов решения задач маршрутизации часто возникает следующая подзадача. Задана последовательность клиентов. Требуется разбить её на маршруты и назначить им транспортные средства так, чтобы суммарная стоимость обслуживания клиентов была минимальной. Умение решать эту подзадачу даёт возможность кодировать решения в виде одной последовательности клиентов и использовать хорошо зарекомендовавшие себя методы для таких задач, в частности, для задачи коммивояжёра. При неограниченном автопарке такая подзадача решается за полиномиальное время. При ограниченном автопарке она NP-трудна [20], но может быть решена точно методом динамического программирования. В настоящей работе используется приближённый метод лагранжевых релаксаций, который даёт верхнюю и нижнюю оценки оптимума. Этот метод применяется в схеме локального поиска с чередующимися окрестностями. Наряду с известными окрестностями вводится новая окрестность экспоненциальной мощности, позволяющая осуществлять глубокую перестройку решений с улучшением целевой функции. Новые процедуры интенсификации и диверсификации локального поиска применяются для повышения эффективности метода. Результаты экспериментальных исследований свидетельствуют о высокой конкурентоспособности нового гибридного метода. Для 15 тестовых примеров из широко используемых электронных коллекций удалось улучшить рекордные значения целевой функции.

В разд. 1 представлена точная постановка задачи и приводится крат-

кий обзор предшествующих исследований. В разд. 2 рассматривается задача разбиения последовательности клиентов на маршруты транспортных средств и метод лагранжевых релаксаций для её решения. В разд. 3 описан разработанный алгоритм локального поиска. В разд. 4 приведены результаты численных экспериментов. Заключение и направление дальнейших исследований завершают статью.

1. Постановка задачи

Рассмотрим полный неориентированный взвешенный граф $G = (V, A)$ с множествами вершин $V = \{0, 1, \dots, n\}$ и рёбер $A = \{(i, j) \mid i, j \in V, i < j\}$. Вершина 0 соответствует складу, где расположены транспортные средства. Остальные вершины представляют клиентов: $V' = V \setminus \{0\}$. Все вершины представлены точками на плоскости. Для каждой пары вершин (i, j) известно расстояние d_{ij} между ними. Для каждого $i \in V'$ задано требование на доставку q_i единиц груза. Автопарк состоит из разнородных транспортных средств (ТС). Множество типов ТС обозначим через K . Для каждого типа $k \in K$ известно число m_k доступных транспортных средств и их предельная вместимость Q_k . Использование ТС влечёт единовременные фиксированные затраты f_k . Проезд из i в j стоит $c_{ij}^k = d_{ij}c_k$, где c_k — удельные затраты на единицу расстояния для ТС типа k .

Пусть r — некоторый маршрут в графе G . Будем говорить, что маршрут допустим для ТС типа k , если он образует цикл, проходящий через вершину 0, и сумма запросов клиентов не превышает вместимости транспортного средства. Стоимость маршрута r складывается из фиксированной стоимости f_k для привлечения ТС и суммарной стоимости рёбер. Задача состоит в нахождении такого набора допустимых маршрутов и назначения им ТС, чтобы каждый клиент посещался ровно один раз, число привлекаемых ТС не превышало численности автопарка, а суммарная стоимость маршрутов была минимальной.

Приведём точную математическую постановку задачи. Введём следующие переменные:

$x_{ij}^k = 1$, если ТС типа k от клиента i едет к клиенту j , и $x_{ij}^k = 0$ в противном случае;

$y_{ij} \geq 0$ — количество груза в ТС при переезде от клиента i к j .

С использованием введённых переменных задача HFFVRP может быть сформулирована в терминах частично-целочисленного линейного

программирования:

$$\min \left(\sum_{k \in K} f_k \sum_{j \in V'} x_{0j}^k + \sum_{k \in K} \sum_{i, j \in V} c_{ij}^k x_{ij}^k \right), \quad (1)$$

$$\text{s. t.} \quad \sum_{k \in K} \sum_{i \in V} x_{ij}^k = 1, \quad j \in V', \quad (2)$$

$$\sum_{i \in V} x_{ij}^k = \sum_{i \in V} x_{ji}^k, \quad j \in V, \quad k \in K, \quad (3)$$

$$\sum_{j \in V'} x_{0j}^k \leq m_k, \quad k \in K, \quad (4)$$

$$\sum_{i \in V} y_{ij} - \sum_{i \in V} y_{ji} = q_j, \quad j \in V', \quad (5)$$

$$y_{0j} \leq \sum_{k \in K} Q_k x_{0j}^k, \quad j \in V', \quad (6)$$

$$y_{ij} \leq \sum_{k \in K} (Q_k - q_i) x_{ij}^k, \quad i \in V', \quad j \in V, \quad i \neq j, \quad (7)$$

$$x_{ij}^k \in \{0, 1\}, \quad y_{ij} \geq 0, \quad i, j \in V, \quad k \in K. \quad (8)$$

Целевая функция задачи определяет суммарные затраты на привлечение транспортных средств и доставку всех грузов клиентам. Ограничения (2) и (3) гарантируют, что каждый клиент посещается ровно один раз и транспортное средство должно покинуть клиента, если оно к нему прибыло. Ограничение (4) устанавливает верхнюю границу на число привлекаемых ТС каждого типа. Условие (5) является ограничением грузового потока: разница в количестве груза, находящегося в ТС, до и после посещения клиента должна равняться запросу клиента. Кроме того, это ограничение исключает циклы, не проходящие через вершину 0. Условие (6) гарантирует, что вместимость транспортных средств не будет превышена, а (7) устанавливает связь между переменными y_{ij} и x_{ij}^k : если ни одно ТС не использует ребро (i, j) , то грузовой поток по нему равен нулю. Ограничение (8) задаёт область изменения переменных.

Задача HFFVRP впервые рассматривалась в [22], где для её решения предлагался метод генерации столбцов в сочетании с локальным поиском и процедурой адаптивного управления памятью. Точный метод, основанный на сведении к задаче о разбиении множества (the set partitioning problem) разработан в [6]. Этот метод позволяет находить оптимальные решения для примеров с числом клиентов не более 100 и является на сегодняшний день наилучшим из точных методов. На этом же сведении

основан и гибридный метод итеративного локального поиска, представленный в [21]. В этом методе задача о разбиении множества решается многократно коммерческим пакетом для задач целочисленного линейного программирования. Пороговые алгоритмы (имитация отжига, Record to Record Travel Algorithms) исследовались в [14, 23, 24]. Варианты поиска с запретами с различными механизмами диверсификации поиска предложены в [7, 9, 19, 25]. Идея чередующихся окрестностей использовалась в [15, 16]. Новые окрестности, в частности, окрестности экспоненциальной мощности, а также случайный порядок их применения позволили улучшить рекордные значения целевой функции для многих тестовых примеров.

Задача о разбиении последовательности клиентов на маршруты использовалась в нескольких алгоритмах. В [20] для её решения использован точный метод динамического программирования. В [11–13] этот метод применяется для быстрого приближённого решения. Эта же идея успешно использована для задач маршрутизации с разделёнными поставками [7] и для задач с временными окнами [10]. В [26] предложена общая схема для решения задач маршрутизации, включающая в себя обобщённую процедуру разбиения. Эта схема была применена для 29 различных постановок задачи и показала свою эффективность. При ограниченном разнородном автопарке задача сильно усложняется. Возникают такие последовательности, для которых не существует допустимого решения (табл. 1). Как уже отмечалось во введении, задача становится NP-трудной. В связи с этим в настоящей работе используется другой подход, основанный на лагранжевых релаксациях с последующим применением методов субградиентной оптимизации.

Т а б л и ц а 1

Допустимая и недопустимая последовательности

$m_1 = 1, Q_1 = 10, m_2 = 1, Q_2 = 14$							
r	:	a	b	c	d	e	\rightarrow допустима
q	:	5	4	4	2	7	
r'	:	a	d	c	e	b	\rightarrow недопустима
q	:	5	2	4	7	4	

2. Метод лагранжевых релаксаций для разбиения последовательности

Представим задачу разбиения последовательности $\pi = (\pi_1, \dots, \pi_n)$, $\pi_i \in V'$, на маршруты в терминах поиска кратчайшего пути в ориен-

тированном взвешенном мультиграфе. Вершина π_1 будет единственным источником, вершина $n+1$ — единственным стоком. Для каждой пары (i, j) , $1 \leq i < j \leq n+1$, определим набор дуг $(i, j)^k$, $k \in K$, соответствующих проезду ТС типа k со склада к клиенту π_i , затем к клиенту π_{i+1} и т. д. до клиента π_{j-1} , а затем возвращению обратно на склад. Стоимость L_{ij}^k такой дуги определим следующим образом:

$$L_{ij}^k = \begin{cases} f_k + r_k \left(d_{0\pi_i} + \sum_{t=i}^{j-2} d_{\pi_t \pi_{t+1}} + d_{\pi_{j-1} 0} \right), & \text{если } \sum_{t=i}^{j-1} q_{\pi_t} \leq Q_k, \\ \infty & \text{в противном случае.} \end{cases}$$

Оптимальное разбиение последовательности получается нахождением пути минимальной стоимости из источника в сток при ограничении на число используемых дуг каждого типа (рис. 1).

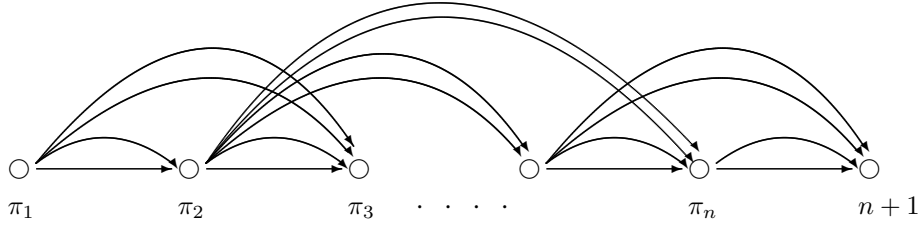


Рис. 1. Мультиграф для последовательности π

Запишем эту задачу в терминах целочисленного линейного программирования. Пусть булева переменная z_{ij}^k принимает значение 1, если и только если дуга $(i, j)^k$ входит в оптимальное решение. Тогда требуемый путь может быть получен из решения следующей задачи:

$$\min \sum_{k \in K} \sum_{i=1}^n \sum_{j=i+1}^{n+1} L_{ij}^k z_{ij}^k, \quad (9)$$

$$\text{s. t. } \sum_{k \in K} \sum_{j=i+1}^{n+1} z_{ij}^k - \sum_{k \in K} \sum_{j=1}^{i-1} z_{ji}^k = \begin{cases} 1, & \text{если } i = 1, \\ -1, & \text{если } i = n+1, \\ 0, & \text{если } 1 < i < n+1, \end{cases} \quad (10)$$

$$\sum_{i=1}^n \sum_{j=i+1}^{n+1} z_{ij}^k \leq m_k, \quad k \in K, \quad (11)$$

$$z_{ij}^k \in \{0, 1\}, \quad k \in K, \quad 1 \leq i < j \leq n+1. \quad (12)$$

Целевая функция (9) задаёт суммарную стоимость пути из источника в сток. Ограничение (10) гарантирует, что число входящих дуг совпадает с числом исходящих дуг для каждой вершины, не являющейся источником или стоком. Из источника выходит одна дуга, в сток заходит одна дуга. Ограничение (11) устанавливает верхнюю границу на количество используемых ТС каждого типа.

Задача (9)–(12) NP-трудна, но удаление ограничений (11) делает её полиномиально разрешимой. Введём неотрицательные множители Лагранжа λ_k для каждого из неравенств (11) и добавим в целевую функцию (9) слагаемое

$$\sum_{k \in K} \lambda_k \left(\sum_{i=1}^n \sum_{j=i+1}^{n+1} z_{ij}^k - m_k \right).$$

Приводя подобные члены, получаем релаксированную задачу:

$$LR(\lambda_k) = \min \sum_{k \in K} \sum_{i=1}^n \sum_{j=i}^n (L_{ij}^k + \lambda_k) z_{ij}^k - \sum_{k \in K} \lambda_k m_k$$

при ограничениях (10), (12).

Заметим, что эта задача является задачей о нахождении кратчайшего пути в мультиграфе для последовательности π (см. рис. 1) с рёбрами весом $\bar{L}_{ij}^k = L_{ij}^k + \lambda_k$. Она легко решается точно динамическим программированием и даёт нижнюю оценку целевой функции (9). Поиск наилучшей такой оценки по множителям Лагранжа приводит к двойственной задаче:

$$D = \max_{\lambda_k \geq 0} LR(\lambda_k).$$

Для её решения используем методы субградиентной оптимизации. Выберем некоторый начальный вектор λ^0 , например, равный нулю, и будем на очередной итерации t менять множители λ^t по правилу:

$$\lambda^{t+1} := \max \left\{ 0, \lambda^t + h^t \left(\sum_{i=1}^n \sum_{j=i+1}^{n+1} z_{ij}^k(\lambda^t) - m_k \right) \right\},$$

где $z_{ij}^k(\lambda^t)$ — оптимальное решение задачи $LR(\lambda^t)$, h^t — размер шага в направлении субградиента

$$\Delta_k^t = \sum_{i=1}^n \sum_{j=i+1}^{n+1} z_{ij}^k(\lambda^t) - m_k, \quad k \in K,$$

целевой функции двойственной задачи в точке $\lambda = \lambda^t$. Если $h^t \rightarrow 0$ и $\sum_{t=0}^{\infty} h^t \rightarrow \infty$, то $LR(\lambda^t) \rightarrow D$ [17]. Применение таких расходящихся рядов приводит к большим вычислительным затратам и медленной сходимости к оптимуму. Поэтому на практике применяют другие правила, гарантирующие быструю сходимость, но не гарантирующие оптимальности. Следуя [4], будем двигаться не вдоль субградиента, а вдоль разности двух последовательных субградиентов:

$$\lambda^{t+1} := \max\{0, \lambda^t + \alpha h^t \Delta^t - (1 - \alpha) h^{t-1} \Delta^{t-1}\},$$

где скаляр α выбирается в интервале от 0 до 1. Длина шага h^t вычисляется по субградиенту Δ^t , значению верхней оценки оптимума UB и параметру Θ^t :

$$h^t := \Theta^t (UB - LR(\lambda^t)) / \|\Delta^t\|^2.$$

Множитель Θ^t сначала полагается равным положительной константе, не превосходящей 2, а затем делится на величину, превосходящую единицу, через заданное число итераций, зависящее от размерности задачи.

Если на некоторой итерации субградиент оказался неположительным, $\Delta^t \leq 0$, то решение $z_{ij}^k(\lambda^t)$ допустимо и значение целевой функции (9) на этом решении даёт верхнюю оценку оптимума UB . Если при этом $\sum_{k \in K} \lambda_k^t \Delta_k^t = 0$, то значения верхней и нижней оценок совпадают и решение $z_{ij}^k(\lambda^t)$ оптимально. Разрыв двойственности в этом случае отсутствует, что наблюдается достаточно редко. Тем не менее такой подход быстро даёт приближённые решения с малой относительной погрешностью, а при нежёстких ограничениях на число привлекаемых транспортных средств часто приводит к точному решению задачи. При этом равенство $\sum_{k \in K} \lambda_k^t \Delta_k^t = 0$ может и не иметь места.

3. Локальный поиск

Для поиска наилучшей перестановки клиентов ниже будут использоваться метаэвристики, хорошо зарекомендовавшие себя при решении задач такого типа [13, 20, 26]. Одной из таких метаэвристик является локальный поиск с чередующимися окрестностями [2, 8]. Для успешной реализации этой схемы требуется набор разнотипных окрестностей. Они будут содержать только допустимые решения задачи.

Пусть π — некоторая перестановка и $R(\pi) = (r_1, \dots, r_{|R|})$ — набор маршрутов, каждому из которых приписан тип транспортного средства. По набору $R(\pi)$ однозначно восстанавливается решение задачи (1)–(8).

Рассмотрим пять окрестностей, в которых не более двух клиентов перемещаются между двумя маршрутами [15].

SHIFT(1, 0). Один клиент перемещается на наилучшую позицию в другом маршруте,

SWAP(1, 1). Два клиента из разных маршрутов меняются своими маршрутами и вставляются в них в наилучшие позиции.

SHIFT(2, 0). Два соседних клиента из одного маршрута переносятся в другой и вставляются в наилучшую позицию, возможно меня очередность.

SWAP(2, 1). Два соседних клиента из одного маршрута переносятся в другой и вставляются в наилучшую позицию после удаления одного из клиентов этого маршрута. Удалённый клиент переносится в исходный маршрут и вставляется там в наилучшую позицию. Как и в предыдущей окрестности, для пары перемещаемых клиентов рассматриваются два варианта вставки: исходный порядок и обратный.

SWAP(2, 2). Два соседних клиента из одного маршрута заменяются двумя соседними клиентами из другого. Четыре варианта очередности для этих пар проверяются, и для каждого варианта подбирается наилучшая позиция для вставки в маршрут.

Аналогичные окрестности для клиентов одного маршрута обозначим через $\text{Shift}_1(1, 0)$, $\text{Swap}_1(1, 1)$, $\text{Shift}_1(2, 0)$, $\text{Swap}_1(2, 1)$. Следующая окрестность меняет структуру двух маршрутов, разбивая их на четыре сегмента.

CROSS. Выбирается два маршрута. Каждый из них разбивается на два непустых упорядоченных набора клиентов, которые затем склеиваются в два новых маршрута. Рассматриваются все варианты такой склейки.

Наконец, последняя окрестность касается только одного маршрута и широко применяется для задачи коммивояжёра.

2-ОПТ. Маршрут представляется в виде цикла, из которого удаляются два несмежных ребра и добавляются два других ребра так, чтобы снова получился цикл.

При определении окрестностей предполагалось, что тип ТС не меняется при перестройке маршрутов, что может приводить к недопустимым решениям или решениям с завышенной стоимостью. Оптимизация автопарка может исправить этот недостаток, но требует решения задачи (9)–(12). Получается очень трудоёмкая процедура. Для сокращения трудоёмкости сначала просматриваются соседние решения без оптимизации

автопарка. Если среди них есть решения с меньшим значением целевой функции, то просмотр окрестности прекращается (правило первого улучшения). В противном случае снова проверяются соседние решения, но уже с оптимизацией автопарка.

Представленные окрестности используются в гибридном алгоритме локального поиска (HLS), основными элементами которого являются: процедура локального поиска с чередующимися окрестностями (RVND), процедуры интенсификации (Intensify), диверсификации поиска (Perturb) и постоптимизации (Postoptimize). По сути, это алгоритм получения локальных оптимумов по указанным выше окрестностям. Однако эти локальные оптимумы ищутся не в произвольных местах допустимой области, а рядом с наилучшим найденным решением [2]. В процедуре диверсификации поиска это решение незначительно модифицируется для получения хорошего стартового решения для нового локального поиска. В процедуре постоптимизации делается попытка улучшить полученный локальный оптимум за счёт окрестностей экспоненциальной мощности [1, 18]. Общая схема алгоритма показана ниже. Наилучший найденный локальный оптимум обозначен через π^* , а наименьшее значение целевой функции — через f^* . В алгоритме I раз выбирается стартовое решение π^0 и I_{LS} раз применяется процедура RVND.

ГИБРИДНЫЙ АЛГОРИТМ ЛОКАЛЬНОГО ПОИСКА (HLS)

1. $f^* \leftarrow \infty$
2. Цикл по $i := 1, \dots, I$
3. $\pi \leftarrow \pi^0; j \leftarrow 0;$
4. Пока $j \leq I_{LS}$
5. $\pi' \leftarrow \text{RVND}(\pi)$
6. Если $f(\pi) < f(\pi')$ и $j > I_{LS}/2$, то
7. $\pi' \leftarrow \text{Intensify}(\pi');$
8. Если $f^* > f(\pi')$, то $f^* \leftarrow f(\pi')$, $\pi^* \leftarrow \pi';$
9. $\pi \leftarrow \text{Perturb}(\pi'); j = j + 1;$
10. $\pi^* \leftarrow \text{Postoptimize}(\pi^*);$
11. Предъявить π^* и f^* .

Начальное решение π^0 (строка 3) строится с помощью следующей рандомизированной процедуры [15]. Для каждого ТС порождается маршрут, состоящий из одного клиента. Этот клиент выбирается случайным образом с равномерным распределением. Разные маршруты получают разных клиентов. Затем выбирается одно из двух правил для вставки оставшихся клиентов в маршруты и одна из двух стратегий: последовательная или параллельная. При последовательной стратегии случай-

ным образом выбирается один из маршрутов и заполняется клиентами до тех пор, пока это позволяет грузоподъемность ТС. При параллельной стратегии все маршруты рассматриваются одновременно и вставка клиентов производится в маршрут с наименьшими дополнительными издержками. Для выбора очередного клиента j рассматриваются следующие два правила оценки издержек: наименьшая стоимость проезда к нему от уже включённых в маршрут клиентов, т. е. $\min_i c_{ij}^k$, либо $\min_i [(c_{ij}^k + c_{ji'}^k - c_{ii'}^k) - \gamma(c_{0j}^k + c_{j0}^k)]$. В последнем выражении оцениваются издержки относительно стоимости нового маршрута с единственным клиентом j . Параметр γ для каждого клиента выбирается случайно из интервала $[0, 1,7]$.

Локальный поиск по указанным выше окрестностям осуществляется в процедуре RVND. На каждой итерации этой процедуры находится локальный оптимум по окрестностям $\text{Shift}_1(1, 0)$, $\text{Swap}_1(1, 1)$, $\text{Shift}_1(2, 0)$, $\text{Swap}_1(2, 1)$, 2-opt. Остальные окрестности применяются для перестройки этих оптимумов и получения стартовых точек для очередной итерации. Порядок использования окрестностей случайный. Общая схема RVND состоит в следующем. Пусть $N = \{N_1, \dots, N_l\}$ — набор окрестностей для перестройки маршрутов и π — некоторое допустимое решение задачи. На каждой итерации выбираем случайным образом одну из окрестностей и находим в ней наилучшее соседнее решение π' для решения π . Если π оказалось локальным оптимумом, т. е. π' не лучше исходного, то удалим данную окрестность из набора N . В противном случае применим к π' алгоритм локального улучшения по окрестностям $\text{Shift}_1(1, 0)$, $\text{Swap}_1(1, 1)$, $\text{Shift}_1(2, 0)$, $\text{Swap}_1(2, 1)$, 2-opt и получим новый локальный оптимум. Заметим, что на каждой итерации либо находится новый локальный оптимум, либо сокращается набор окрестностей.

В процессе интенсификации поиска по набору маршрутов R строится последовательность клиентов π и для неё решается задача (9)–(12). Последовательность строится следующим образом. Случайным образом выбирается первый маршрут и целиком добавляется в последовательность. Далее выбирается следующий маршрут, у которого крайний клиент расположен ближе всех к последнему клиенту в последовательности. Все клиенты этого маршрута добавляются в последовательность. Этот процесс продолжается до тех пор, пока все маршруты не будут использованы.

В процедуре диверсификации тем же способом по набору маршрутов строится последовательность клиентов, а затем L раз случайно выбираются два элемента последовательности и меняется порядок элементов

между ними на обратный (аналог переходов по окрестности 2-opt). Для полученной последовательности решается задача (9)–(12). Если решение задачи не является допустимым, то применяется альтернативная процедура диверсификации: L раз осуществляется переход в соседнее решение по окрестности Swap(1, 1), Shift(1, 0) или Shift(2, 0). Аналогичная процедура использовалась в [15, 16].

Финальный локальный оптимум подвергается процедуре постоптимизации (строка 10). Здесь применяются окрестности экспоненциальной мощности. Ниже приводится описание двух таких окрестностей, построенных на идее выталкивающих цепей (ejection chains). Первая окрестность ранее использовалась в [18], вторая окрестность новая.

Пусть, как и раньше, $R(\pi) = (r_1, \dots, r_{|R|})$ — набор маршрутов. Рассмотрим два элемента π_i и $\pi_{i'}$ из разных маршрутов r и r' . *Извлечением* будем называть удаление π_i из r , удаление $\pi_{i'}$ из r' и вставку π_i в наилучшую позицию в r' без $\pi_{i'}$. Пусть величина $w(\pi_i, \pi_{i'})$ задаёт стоимость такой перестройки маршрутов r и r' , т. е.

$$w(\pi_i, \pi_{i'}) = (c_{\pi_{i-1} \pi_{i+1}} - c_{\pi_{i-1} \pi_i} - c_{\pi_i \pi_{i+1}}) + (c_{\pi_{i'-1} \pi_{i'+1}} - c_{\pi_{i'-1} \pi_{i'}} - c_{\pi_{i'} \pi_{i'+1}}) + (c_{\pi_l \pi_i} + c_{\pi_i \pi_{l+1}} - c_{\pi_l \pi_{l+1}}),$$

если π_i встаёт в r' между элементами π_l и π_{l+1} (рис. 2).

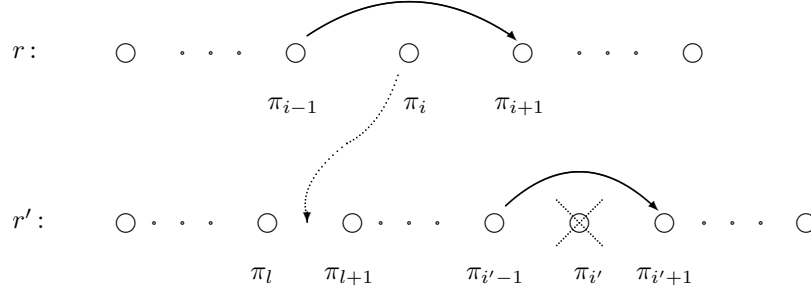


Рис. 2. Извлечение для $\pi_i \in r$, $\pi_{i'} \in r'$

Вставкой будем называть удаление π_i из r и добавление π_i в наилучшую позицию в r' . Для такой операции определим величину $w(\pi_i, r')$ как подорожание маршрута r' вместе с экономией от удаления π_i из r , т. е.

$$w(\pi_i, r') = (c_{\pi_{i-1} \pi_{i+1}} - c_{\pi_{i-1} \pi_i}) + (c_{\pi_{i'-1} \pi_{i'+1}} - c_{\pi_{i'-1} \pi_i} - c_{\pi_{i'} \pi_{i'+1}}) + (c_{\pi_l \pi_i} + c_{\pi_i \pi_{l+1}} - c_{\pi_l \pi_{l+1}}).$$

Заметим, что величины $w(\pi_i, \pi_{i'})$ и $w(\pi_i, r')$ могут быть отрицательными.

Определим *граф извлечений и вставок*, вершины которого соответствуют маршрутам и клиентам, а дуги — переходам извлечений и вставок. Такой граф будет иметь слоистую структуру (рис. 3). В слой входят фиктивная вершина (сток), соответствующая самому маршруту, и по одной вершине для каждого клиента в маршруте. Межслойные дуги могут быть двух типов в зависимости от переходов: дуги извлечений $(\pi_i, \pi_{i'})$ с весом $w(\pi_i, \pi_{i'})$ и дуги вставки (π_i, r') с весом $w(\pi_i, r')$. Конечные вершины дуг вставки являются фиктивными. В случае, если при таких переходах нарушаются ограничения на вместимость ТС, вес дуги полагается равным достаточно большому числу. Если в этом графе есть цикл или путь из некоторой вершины до фиктивной, проходящие через каждый слой не более одного раза и имеющие отрицательный вес, то текущее решение может быть перестроено с уменьшением стоимости. Заметим, что в случае нахождения пути фиктивная вершина в нём может быть только в конце.

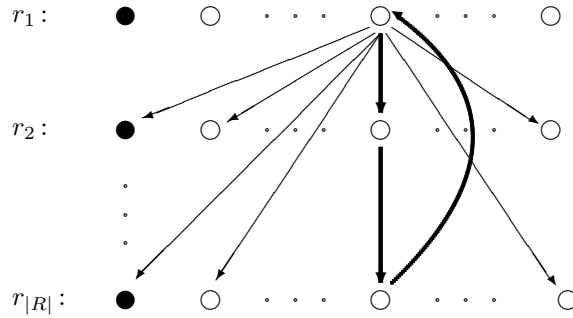


Рис. 3. Слоистый граф

Нахождение такого пути или цикла является NP-трудной задачей. Приближённое решение может быть получено модификацией алгоритма Флойда — Уоршалла [18]. Модификация заключается в добавлении следующей проверки. Пусть есть два пути: от вершины π_i до π_k и от π_k до π_j . Путь от π_i до π_j через вершину π_k возможен только в том случае, если пути от π_i до π_k и от π_k до π_j не содержат вершин из одного слоя.

Вторая окрестность, используемая в процессе постоптимизации, подразумевает разбиение маршрутов на две части и склейку частей из разных маршрутов. Отличие этих окрестностей показано на рис. 4.

Снова строится слоистый граф, каждый слой соответствует одному маршруту, но теперь он не будет содержать фиктивных вершин. Пусть первый слой соответствует маршруту $r_1 = (u_1, u_2, \dots, u_{r_1})$. Так как маршрут можно проходить в прямом и обратном направлении, для каждой вершины u_i введём её дубль u'_i и (для наглядности) представим эти вершины на слое сначала в прямом, а потом в обратном порядке.

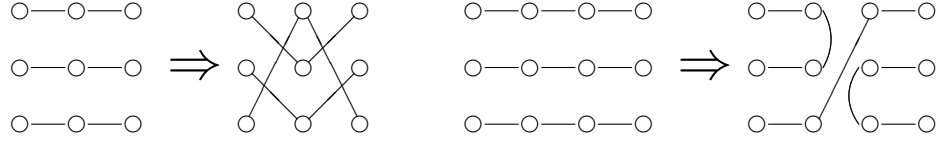


Рис. 4. Две окрестности большой мощности

Пусть $u_i \in r$ и $v_j \in \bar{r}$. Межслойные дуги теперь будут четырёх типов: $(u_i, v_j), (u_i, v'_j), (u'_i, v_j), (u'_i, v'_j)$.

Дуга (u_i, v_j) означает создание маршрута $(u_1, \dots, u_i, v_{j+1}, \dots, v_{\bar{r}})$, её вес $w(u_i, v_j)$ равен $c_{u_i v_{j+1}} - c_{u_i u_{i+1}}$.

Дуга (u_i, v'_j) означает создание маршрута $(u_1, \dots, u_i, v_j, \dots, v_1)$, её вес $w(u_i, v'_j)$ равен $c_{u_i v_j} - c_{u_i u_{i+1}}$.

Дуга (u'_i, v_j) означает создание маршрута $(u_r, \dots, u_{i+1}, v_{j+1}, \dots, v_{\bar{r}})$, её вес $w(u'_i, v_j)$ равен $c_{u_{i+1} v_{j+1}} - c_{u_i u_{i+1}}$.

Дуга (u'_i, v'_j) означает создание маршрута $(u_r, \dots, u_{i+1}, v_j, \dots, v_1)$, её вес $w(u'_i, v'_j)$ равен $c_{u_{i+1} v_j} - c_{u_i u_{i+1}}$.

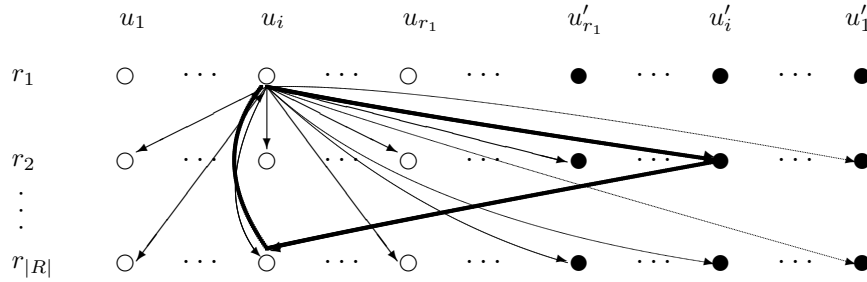


Рис. 5. Разбиение и склейка маршрутов

Если при создании нового маршрута нарушается ограничение по грузоподъёмности ТС, то вес ребра полагаем равным достаточно большому числу. Наличие в таком графе цикла, проходящего через каждый слой

не более одного раза и имеющего отрицательный вес, свидетельствует о возможности перестройки решения с уменьшением целевой функции. Поиск такого цикла осуществляется тем же способом, что и в предыдущем случае. На рис. 5 жирным показан цикл, соответствующий перестройке трёх маршрутов: первого, второго и последнего. Начало первого маршрута склеивается с началом второго, конец второго маршрута склеивается с концом последнего, начало последнего — с концом первого. Вариант такой склейки показан на рис. 4.

4. Численные эксперименты

Разработанные алгоритмы запрограммированы на языке Java и тестировались на персональном компьютере с процессором Intel®Core™ i7, 2.20 ГГц, 8Гб ОЗУ. В первом эксперименте проверялась эффективность метода субградиентной оптимизации для решения задачи (9)–(12) при заданной последовательности клиентов. Для этой цели использовались тестовые примеры из [12], в которых число клиентов менялось от 19 до 255. В этой же работе содержатся рекордные решения для этих примеров. Эти решения легко представить в виде последовательности клиентов так же, как это делается в процедуре интенсификации. Для метода субградиентной оптимизации использовались следующие параметры: $\alpha = 0,74$; $\Theta^0 = 1,8$; $UB = 1,3LR(\lambda^t)$. Величина Θ^i делилась на 1,1 через каждые четыре итерации. Каждый пример решался 10 раз. Метод останавливался в случае нахождения первого допустимого решения либо по достижению максимального числа итераций. Вычислялись среднее число итераций до получения первого допустимого решения $Iter_{avg}$, средняя погрешность относительно нижней оценки LbE_{avg} в процентах, среднее время работы t_{avg} и суммарное время работы алгоритма T в миллисекундах. Максимальное число итераций не превышало 2000.

Т а б л и ц а 2

Результаты при заданных последовательностях

n	<100	100–150	150–200	>200	19–255
число примеров	15	38	31	12	96
$Iter_{avg}$	4	8	18	8	11
Err_{avg}	0,06	0,43	0,33	0,33	0,33
LbE_{avg}	0,12	0,5	0,68	0,33	0,48
t_{avg}	1,24	4,45	13,74	20,83	8,99
T	186	1690	4259	2500	8635
не найдено решений	0	0	0	0	0

В табл. 2 представлены результаты расчётов, разбитые на пять колонок: примеры с $n < 100$, $100 \leq n < 150$, $150 \leq n < 200$, $n \geq 200$ и итоговая колонка для всех примеров $19 \leq n \leq 255$. Строка Err_{avg} показывает среднюю погрешность полученного решения относительно рекордного решения в процентах. В этом эксперименте для 4 примеров из 96 удалось превзойти рекордное решение. В 54 случаях алгоритм нашёл рекордное решение, а в остальных случаях проиграл, но средняя погрешность оказалась небольшой, всего 0,33%. Так как сравнение проводилось на известных тестовых примерах, где проверялись многие алгоритмы, применения методов субградиентной оптимизации можно считать целесообразным.

Т а б л и ц а 3

Результаты при перемешивании внутри маршрутов

n	<100	100–150	150–200	>200	19–255
число примеров	15	38	31	12	96
Iter_{avg}	31	68	110	37	72
Err_{avg}	–4,39	–3,89	–5,69	–5,92	–4,8
LbE_{avg}	1,73	2,89	2,47	1,87	2,45
t_{avg}	3,25	16,03	58,56	50,7	32,1
T	488	6093	18155	6084	30820
не найдено решений	1	3	6	0	10

Т а б л и ц а 4

Результаты при случайных последовательностях

n	<100	100–150	150–200	>200	19–255
число примеров	15	38	31	12	96
Iter_{avg}	134	218	249	138	205
LbE_{avg}	6,94	3,82	3,08	2,77	3,94
t_{avg}	8,37	40,15	91,42	110,79	60,57
T	1256	15258	28340	13295	58149
не найдено решений	6	15	16	2	39

Как уже отмечалось выше, при неудачной последовательности алгоритм может не найти допустимого решения задачи. Задача может просто не иметь решений. В связи с этим был проделан эксперимент на других последовательностях для тех же примеров. В первом случае внутри каждого маршрута выбиралась случайная последовательность клиентов. Во втором случае бралась произвольная последовательность всех клиентов. Заметим, что в первом случае допустимое решение всегда существует. Табл. 3 и 4 представляют результаты этого эксперимента. Величина

$E_{\text{rr}}_{\text{avg}}$ в табл. 3 показывает среднюю относительную погрешность финального решения по сравнению со стартовым. В первом случае только в 10 испытаниях из 960 не удалось найти допустимого решения. Для произвольной последовательности это число оказалось больше, но составило только 39, т. е. около 4% от общего числа испытаний. Средняя погрешность не превысила в среднем 4%, хотя при малом числе клиентов оказалась почти в два раза выше, точнее 6,94%. В итоге можно заключить, что для данных примеров редко не удаётся найти допустимое решение задачи, т. е. ограничения (4) не являются жёсткими.

Во втором эксперименте проверялась эффективность гибридного алгоритма HLS. Для этих целей использовались те же тестовые примеры, параметры алгоритма: $I = 100$, $I_{LS} = n$. В процедуре Perturb переход в следующее соседнее решение осуществлялся $\lceil 3I/I_{LS} \rceil$ раз. Разработанный алгоритм сравнивался двумя наиболее эффективными из существующих эвристик: GRASP-ELS — адаптивная жадная эвристика с эволюционным локальным поиском [12] и MS-ILS-SFR — гибридная схема итеративного локального поиска с мультистартом и перераспределением ТС [16]. Первая эвристика тестировалась на ПК с процессором Opteron, 2,1 ГГц. Вторая эвристика — на ПК с процессором Intel®Core™ i7 2,9 ГГц, 8 Гб ОЗУ. Все эксперименты проводились с использованием одного ядра процессора.

В табл. 5–7 представлены результаты численных экспериментов. В первой колонке приводится название примера, по которому можно легко восстановить характеристики привлекаемых ТС. Во второй колонке содержится размерность задачи. Для алгоритма HLS указаны минимальное и среднее значения, стандартное отклонение и среднее время счёта за 10 запусков алгоритма в секундах. Для других алгоритмов приводится значение целевой функции и время счёта в секундах. В предпоследней колонке содержится значение целевой функции наилучшего известного решения, в последней колонке — относительная погрешность среднего значения z_{avg} против известного рекорда в процентах. Жирным шрифтом выделены новые рекорды, полученные гибридным алгоритмом HLS.

В целом интеграция методов субградиентной оптимизации, локального поиска и новых процедур интенсификации и диверсификации показала свою эффективность. Для пятнадцати тестовых примеров удалось улучшить рекордные значения целевой функции. Средняя относительная погрешность на всех тестовых примерах составила 0,65%.

Результаты численных экспериментов при $n < 150$

Code	n	HLS				MS-ILS-SFR		GRASP _x ELS		z^*	g
		z_{\min}	z_{avg}	σ	t	z	t	z	t		
H-75	19	452,85	452,85	0,00	1	452,85	1	452,85	0	452,85	0,00
H-92	34	564,39	564,39	0,00	29	564,39	21	564,39	21	564,39	0,00
H-93	39	1036,99	1036,99	0,00	19	1036,99	17	1036,99	27	1036,99	0,00
H-94	46	1378,25	1378,25	0,00	45	1378,25	27	1378,66	16	1378,25	0,00
H-55	55	10256,16	10264,37	10,20	46	10244,34	28	10247,86	191	10244,34	0,20
H-52	58	4035,59	4035,59	0,00	61	4027,27	63	4029,42	40	4027,27	0,21
H-10	68	2108,10	2108,32	0,50	185	2107,55	119	2107,55	25	2107,55	0,04
H-39	76	2926,59	2928,99	2,88	202	2921,36	171	2934,55	182	2921,36	0,26
H-70	77	6689,61	6729,87	19,77	201	6684,56	125	6689,61	121	6684,56	0,68
H-82	78	4769,35	4772,58	1,87	180	4766,74	108	4774,26	145	4766,74	0,12
H-08	83	4596,52	4597,65	1,14	254	4591,75	93	4598,49	305	4591,75	0,13
H-36	84	5709,31	5738,34	24,22	358	5684,62	267	5759,34	104	5684,61	0,95
H-43	85	8737,13	8749,47	9,08	496	8707,94	178	8764,75	220	8707,94	0,48
H-01	91	9180,45	9187,28	5,93	387	9210,14	269	9210,14	52	9210,14	-0,25
H-11	94	3367,41	3372,57	2,86	430	3367,41	244	3370,47	265	3367,41	0,15
H-90	101	2350,68	2358,22	3,55	454	2347,50	332	2360,83	15	2346,13	0,52
H-17	104	5381,19	5403,10	15,82	439	5369,31	248	5370,05	173	5362,83	0,75
H-84	104	7228,38	7244,86	12,77	677	7236,49	344	7269,55	206	7227,88	0,23
H-81	105	10675,92	10690,71	9,57	546	10689,77	385	10715,28	84	10689,77	0,01
H-2B	106	8577,50	8601,68	17,30	859	8482,79	538	8537,31	303	8464,69	1,62
H-07	107	8074,64	8135,00	27,90	562	8089,21	245	8130,50	306	8089,21	0,57
H-87	107	3753,87	3757,12	4,60	511	3753,87	432	3753,87	104	3753,87	0,09
H-47	110	16291,49	16332,46	32,13	598	16175,22	391	16222,94	334	16175,22	0,97
H-48	110	21316,55	21444,07	58,62	663	21330,75	520	21413,92	371	21318,04	0,59
H-61	110	7308,84	7320,97	6,46	559	7292,03	519	7300,10	108	7292,03	0,40
H-12	111	3543,99	3545,82	1,87	632	3543,99	536	3543,99	71	3543,99	0,05
H-30	111	6329,09	6361,97	13,81	550	6322,39	425	6321,69	201	6313,39	0,77
H-2A	112	7820,37	7862,82	27,45	1200	7793,16	448	7885,93	299	7793,16	0,89
H-53	114	6483,51	6497,73	12,37	663	6434,83	511	6470,49	418	6434,83	0,98
H-05	115	10896,35	10937,73	31,76	898	10896,33	267	10963,62	489	10896,33	0,38
H-13	118	6709,28	6728,61	11,94	854	6696,43	368	6713,14	303	6696,43	0,48
H-06	120	11760,08	11783,45	15,76	1453	11711,35	389	11792,94	368	11692,85	0,77
H-03	123	10727,36	10747,53	14,35	1433	10730,10	673	11320,58	512	10730,10	0,16
H-83	123	10041,06	10050,45	7,75	1339	10029,60	635	10019,83	332	10019,15	0,31
H-74	124	11598,92	11634,25	27,60	1247	11592,72	536	11732,54	247	11586,87	0,41
H-21	125	5160,03	5175,71	13,23	835	5139,84	408	5154,38	330	5139,84	0,70
H-26	125	6393,47	6433,33	22,57	1071	6423,70	1045	6481,93	351	6423,70	0,15
H-88	126	12406,93	12452,74	17,95	1236	12448,38	410	12443,41	632	12402,85	0,40
H-16	128	4156,97	4160,43	2,26	1081	4156,97	799	4161,61	181	4156,97	0,08
H-31	130	4105,67	4130,97	15,92	2052	4091,81	1174	4103,88	308	4091,52	0,96
H-40	131	11156,86	11184,18	14,16	1401	11122,32	808	11172,98	615	11122,32	0,56
H-89	133	7099,68	7120,97	14,34	1331	7105,47	682	7135,36	246	7105,47	0,22
H-41	134	7606,16	7643,93	17,54	1915	7572,07	969	7679,32	326	7572,07	0,95
H-34	135	5784,25	5799,40	9,19	1565	5786,98	784	5800,12	406	5758,09	0,72
H-60	136	17073,80	17106,54	17,25	1262	17037,23	647	17067,85	444	17037,23	0,41
H-73	136	10195,13	10215,26	10,05	1636	10196,04	536	10243,66	598	10195,33	0,20
H-28	140	5538,45	5550,86	6,67	1560	5533,01	525	5542,76	343	5531,06	0,36
H-25	142	7209,61	7218,87	6,12	2438	7217,26	1722	7228,54	518	7206,64	0,17
H-85	145	8812,03	8842,44	16,86	2257	8797,92	787	8874,31	383	8779,76	0,71
H-79	146	7266,75	7310,23	16,36	1687	7274,18	1138	7314,89	474	7259,54	0,70
H-66	149	12790,56	12862,79	49,36	1978	12828,34	1316	13319,73	443	12828,34	0,27

Т а б л и ц а 6

Результаты численных экспериментов при $150 \leq n < 200$

Code	n	HLS				MS-ILS-SFR		GRASP _x ELS			
		z_{\min}	z_{avg}	σ	t	z	t	z	t	z^*	g
H-69	151	9222,25	9270,15	32,43	2130	9167,18	940	9276,93	509	9162,78	1,17
H-76	151	12007,57	12036,84	20,71	4265	12018,22	1241	12098,66	686	12018,22	0,15
H-56	152	31236,61	31286,03	42,21	2254	31090,53	1136	31292,81	394	31090,53	0,63
H-86	152	9056,31	9085,76	15,13	2512	9053,41	908	9076,63	383	9030,68	0,61
H-37	160	6894,98	6915,59	18,16	2468	6870,11	1553	6921,19	384	6858,23	0,84
H-64	160	17157,37	17180,65	21,47	1984	17135,16	792	17157,37	512	17135,16	0,27
H-24	162	9145,18	9178,51	23,99	3357	9118,01	1722	9183,78	610	9101,47	0,85
H-57	162	43378,37	43581,83	112,20	2883	44850,05	1361	45152,42	639	44818,18	-2,76
H-29	163	9151,41	9182,21	16,88	2207	9142,86	1363	9147,39	425	9142,86	0,43
H-09	166	7671,00	7700,14	20,40	3126	7651,33	877	7647,59	450	7619,19	1,06
H-35	167	9605,62	9640,89	26,18	3274	9592,43	1163	9640,80	459	9574,71	0,69
H-45	169	10664,81	10719,12	38,93	2518	10496,88	2118	10519,25	451	10484,23	2,24
H-80	170	6833,02	6844,95	9,06	2864	6825,46	1530	6839,96	230	6816,89	0,41
H-44	171	12549,34	12604,56	32,95	2863	12237,42	1852	12418,00	447	12197,46	3,34
H-54	171	10433,38	10477,30	37,21	3380	10370,09	2725	11511,62	364	10370,09	1,03
H-67	171	10971,29	11024,31	36,86	3547	10915,60	1495	11854,61	337	10915,60	1,00
H-63	173	20154,56	20262,46	55,19	3686	19994,01	984	20241,72	694	19951,76	1,56
H-14	175	5680,64	5705,25	14,04	3138	5667,82	2498	5679,80	449	5644,92	1,07
H-42	177	10940,03	11012,21	38,18	5513	10855,73	3696	11713,90	317	10855,73	1,44
H-02	180	11797,23	11839,34	16,63	4987	11718,86	1693	12102,01	326	11718,86	1,03
H-04	182	10861,29	10892,04	20,09	4237	10787,03	1709	11276,45	726	10787,03	0,97
H-95	183	6223,54	6255,38	22,29	1877	6175,62	907	6244,13	322	6175,62	1,29
H-71	185	9976,24	9988,03	11,32	3417	9891,50	1294	9960,84	640	9870,22	1,19
H-72	185	5950,67	5977,67	17,57	3647	5883,33	2238	5976,54	197	5883,33	1,60
H-50	186	12466,27	12482,40	15,01	5488	12385,32	4092	12508,77	647	12374,04	0,88
H-15	187	8255,95	8273,64	10,00	5706	8268,18	2224	8301,63	521	8236,40	0,45
H-33	188	9419,00	9452,46	25,23	6089	9437,30	2277	9543,17	603	9421,01	0,33
H-77	189	6943,61	7036,46	35,32	3852	6929,67	2804	6991,59	636	6929,67	1,54
H-78	189	7101,20	7162,22	22,33	4247	7039,90	1937	7069,82	471	7035,01	1,81
H-59	192	14299,28	14324,68	18,66	5405	14309,48	2974	14367,14	676	14282,59	0,29
H-91	195	6374,01	6398,88	16,00	4918	6381,13	3235	6437,14	544	6377,48	0,34

Т а б л и ц а 7

Результаты численных экспериментов при $n > 200$

Code	n	HLS				MS-ILS-SFR		GRASP _x ELS			
		z_{\min}	z_{avg}	σ	t	z	t	z	t	z^*	g
H-23	202	7769,33	7793,84	14,89	5154	7760,62	2657	7809,20	802	7750,27	0,56
H-38	204	11224,72	11271,90	24,97	6177	11217,53	2613	11439,58	422	11217,53	0,48
H-27	219	8417,62	8444,78	21,38	7536	8436,55	3424	8520,74	996	8436,55	0,10
H-58	219	23480,52	23543,71	40,30	8724	23504,15	2640	23530,10	1028	23397,76	0,62
H-65	222	13036,33	13102,95	28,69	8043	13013,89	4348	13077,63	636	13013,89	0,68
H-19	223	11760,02	11819,28	28,73	6831	11702,98	2588	11805,34	1010	11702,77	1,00
H-62	224	23035,91	23094,21	49,26	8710	22952,06	3220	23434,56	829	22952,06	0,62
H-22	238	13240,93	13270,10	15,43	8341	13103,51	2424	13162,90	836	13068,03	1,55
H-32	243	9378,30	9418,40	22,65	16911	9412,56	5771	9537,48	1131	9412,56	0,06
H-49	245	16282,44	16368,02	54,37	18026	16219,41	8694	16417,30	990	16219,41	0,92
H-46	249	24697,34	24832,07	84,31	11514	24428,54	7371	24805,27	1475	24428,54	1,65
H-18	255	9676,56	9703,47	17,31	13305	9668,17	6127	9797,61	1216	9668,17	0,37

5. Заключение

Рассмотрена задача маршрутизации транспортных средств с разнородным ограниченным автопарком. Для её решения применён подход, разбивающий задачу на две подзадачи: поиск наилучшей последовательности клиентов и оптимальное распределение автопарка при заданной последовательности. Для первой подзадачи применяется метод локального поиска с чередующимися окрестностями, для второй подзадачи — метод субградиентной оптимизации. В схему локального поиска добавлены новые процедуры интенсификации и диверсификации, а также процедура постоптимизации, основанная на окрестностях экспоненциальной мощности. Проведённые численные эксперименты показали эффективность предложенного подхода. Для 15 тестовых примеров удалось улучшить рекордные значения целевой функции.

ЛИТЕРАТУРА

1. Давыдов И. А., Кононова П. А., Кочетов Ю. А. Локальный поиск с окрестностью экспоненциальной мощности для задачи балансировки нагрузки на серверы // Дискрет. анализ и исслед. операций. 2014. Т. 21, № 6. С. 21–34.
2. Давыдов И. А., Кочетов Ю. А., Младенович Н., Уросевич Д. Быстрые метаэвристики для дискретной задачи о $(r|p)$ -центроиде // Автоматика и телемеханика. 2014. № 4. Р. 106–119.
3. Кононова П. А., Кочетов Ю. А. Локальный поиск с чередующимися окрестностями для задачи Джонсона с пассивным буфером // Дискрет. анализ и исслед. операций. 2012. Т. 19, № 5. С. 63–82.
4. Стецюк П. И. Методы эллипсоидов и r -алгоритмы. Кишинэу: Эврика, 2014. 488 с.
5. Baldacci R., Battarra M., Vigo D. Routing a heterogeneous fleet of vehicles // The vehicle routing problem: latest advances and new challenges. New York: Springer-Verl., 2008. P. 3–27.
6. Baldacci R., Mingozzi A. A unified exact method for solving different classes of vehicle routing problems // Math. Program., Ser. A. 2009. Vol. 120, No. 2. P. 347–380.
7. Boudia M., Prins C., Reghioui M. An effective memetic algorithm with population management for the split delivery vehicle routing problem // Hybrid Metaheuristics. 2007. Vol. 4771. P. 16–30. (Lect. Notes Comput. Sci; Vol. 4771).
8. Brandão J. A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem // Eur. J. Oper. Res. 2009. Vol. 195, No. 3. P. 716–728.

9. **Brandão J.** A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem // *Comput. Oper. Res.* 2011. Vol. 38, No. 1. P. 140–151.
10. **Desrosiers J., Soumis F., Desrochers M., Sauvé M.** Methods for routing with time windows // *Eur. J. Oper. Res.* 1986. Vol. 23, No. 2. P. 236–245.
11. **Duhamel C., Gouinaud C., Lacomme P., Prodhon C.** A multi-thread GRASP_xELS for the heterogeneous capacitated vehicle routing problem // *Hybrid Metaheuristics*. Heidelberg: Springer-Verl., 2013. P. 237–269. (Stud. Comput. Intell.; Vol. 434).
12. **Duhamel C., Lacomme P., Prodhon C.** A GRASP_xELS with depth first search split procedure for the HVRP // *Res. Rep. LIMOS/RR-10-08*, LIMOS, 2010.
13. **Duhamel C., Lacomme P., Prodhon C.** Efficient frameworks for greedy split and new depth first search split procedures for routing problems // *Comput. Oper. Res.* 2011. Vol. 38, No. 4. P. 723–739.
14. **Li F., Golden B., Wasil E.** A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem // *Comput. Oper. Res.* 2007. Vol. 34, No. 9. P. 2734–2742.
15. **Penna P. H. V., Subramanian A., Ochi L. S.** An iterated local search heuristic for the heterogeneous fleet vehicle routing problem // *J. Heuristics*. 2013. Vol. 19, No. 2. P. 201–232.
16. **Penna P. H. V., Vidal T., Ochi L. S., Prins C.** New compound neighborhoods structures for the heterogeneous fixed fleet vehicle routing problem // *Proc. Conf. Simp. Bras. Pesqui. Operac.* (Natal, Brasil, Sept. 16–19, 2013). P. 3623–3633.
17. **Poljak B. T.** Subgradient methods: a survey of Soviet research // *Nonsmooth Optimization*. IIASA Proc. Ser., 1977, Vol. 3.
18. **Potvin J.-Y., Naud M.-A.** Tabu search with ejection chains for the vehicle routing problem with private fleet and common carrier // *J. Oper. Res. Soc.* 2011. Vol. 62, No. 2. P. 326–336.
19. **Prins C.** Efficient heuristics for the heterogeneous fleet multitrip VRP with application to a large-scale real case // *J. Math. Model. Algorithms*. 2002. Vol. 1, No. 2. P. 135–150.
20. **Prins C.** Two memetic algorithms for heterogeneous fleet vehicle routing problems // *Eng. Appl. Artif. Intell.* 2009. Vol. 22, No. 6. P. 916–928.
21. **Subramanian A., Penna P. H. V., Uchoa E., Ochi L. S.** A hybrid algorithm for the heterogeneous fleet vehicle routing problem // *Eur. J. Oper. Res.* 2012. Vol. 221, No. 2. P. 285–295.
22. **Taillard E. D.** A heuristic column generation method for heterogeneous fleet VRP // *RAIRO, Oper. Res.* 1999. Vol. 33, No. 1. P. 1–14.
23. **Tarantilis C. D., Kiranoudis C. T., Vassiliadis V. S.** A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem // *J. Oper. Res. Soc.* 2003. Vol. 54, No. 1. P. 65–71.
24. **Tarantilis C. D., Kiranoudis C. T., Vassiliadis V. S.** A threshold ac-

cepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem // Eur. J. Oper. Res. 2004. Vol. 152, No. 1. P. 148–158.

- 25. Tarantilis C. D., Zachariadis E. E., Kiranoudis C. T.** A guided tabu search for the heterogeneous vehicle routing problem // J. Oper. Res. Soc. 2008. Vol. 59, No. 2. P. 1659–1673.
- 26. Vidal T., Crainic T. G., Gendreau M., Prins C.** A unified solution framework for multi-attribute vehicle routing problems // Eur. J. Oper. Res. 2014. Vol. 234, No. 3. P. 658–673.

*Кочетов Юрий Андреевич,
Хмелёв Алексей Владимирович*

Статья поступила
13 марта 2015 г.
Исправленный вариант —
15 июня 2015 г.

DISKRETNYYI ANALIZ I ISSLEDOVANIE OPERATSII
September–October 2015. Volume 22, No. 5. P. 5–29

UDC 519.85

DOI: 10.17377/daio.2015.22.479

HYBRID LOCAL SEARCH FOR THE HETEROGENOUS FIXED FLEET VEHICLE ROUTING PROBLEM

Yu. A. Kochetov^{1,2}, A. V. Khmelev²

¹Sobolev Institute of Mathematics,
4 Koptug Ave., 630090 Novosibirsk, Russia

²Novosibirsk State University,
2 Pirogov St., 630090 Novosibirsk, Russia
e-mail: jkochet@math.nsc.ru, avhmel@gmail.com

Abstract. We consider the heterogeneous fixed fleet vehicle routing problem and assume that customers are presented by points in Euclidean plane and a limited fleet of heterogeneous vehicles is available. The proposed hybrid local search algorithm uses permutations of customers for coding feasible solutions. For given permutation, the Lagrangian relaxation approach is applied as decoding method for this NP-hard problem. New intensification and diversification procedures are proposed and a new exponential neighborhood is introduced. Computational results for test instances with number of customers up to 255 are reported. New best found solutions are discovered for 15 test instances. Tab. 7, ill. 5, bibliogr. 26.

Keywords: local search, exponential neighborhood, Lagrangian relaxation, subgradient optimization.

REFERENCES

1. I. A. Davydov and Yu. A. Kochetov, VNS-based heuristic with an exponential neighborhood for the server load balancing problem, *Electron. Notes Discrete Math.*, **47**, 53–60, 2015.
2. I. A. Davydov, Yu. A. Kochetov, N. Mladenović, and D. Urošević, Fast metaheuristics for the discrete $(r|p)$ -centroid problem, *Avtom. Telemekh.*, No. 4, 106–119, 2014. Translated in *Autom. Remote Control*, **75**, No. 4, 677–687, 2014.
3. P. A. Kononova and Yu. A. Kochetov, The variable neighborhood search for the two machine flow shop problem with a passive prefetch, *Diskretn. Anal. Issled. Oper.*, **19**, No. 5, 63–82, 2012. Translated in *J. Appl. Ind. Math.*, **7**, No. 1, 54–67, 2013.

4. **P. I. Stetsyuk**, *Metody ellipsoidov i r-algoritmy* (Ellipsoid methods and r -algorithms), Evrika, Chisinau, 2014.
5. **R. Baldacci, M. Battarra, and D. Vigo**, Routing a heterogeneous fleet of vehicles, in B. Golden, S. Raghavan, and E. Wasil, eds., *The Vehicle Routing Problem: Latest Advances and New Challenges*, pp. 3–27, Springer, New York, 2008 (Oper. Res./Comput. Sci. Interfaces, Vol. 43).
6. **R. Baldacci and A. Mingozzi**, A unified exact method for solving different classes of vehicle routing problems, *Math. Program., Ser. A*, **120**, No. 2, 347–380, 2009.
7. **M. Boudia, C. Prins, and M. Reghioui**, An effective memetic algorithm with population management for the split delivery vehicle routing problem, in *Hybrid Metaheuristics* (Proc. 4th Int. Workshop Hybrid Metaheuristics, Dortmund, Germany, Oct. 8–9, 2007), pp. 16–30, Springer, Berlin, 2007 (Lect. Notes Comput. Sci., Vol. 4771).
8. **J. Brandão**, A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem, *Eur. J. Oper. Res.*, **195**, No. 3, 716–728, 2009.
9. **J. Brandão**, A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem, *Comput. Oper. Res.*, **38**, No. 1, 140–151, 2011.
10. **J. Desrosiers, F. Soumis, M. Desrochers, and M. Sauvé**, Methods for routing with time windows, *Eur. J. Oper. Res.*, **23**, No. 2, 236–245, 1986.
11. **C. Duhamel, C. Guinaud, P. Lacomme, and C. Prodhon**, A multi-thread GRASPxELS for the heterogeneous capacitated vehicle routing problem, in El-G. Talbi, ed., *Hybrid Metaheuristics*, pp. 237–269, Springer, Heidelberg, 2013 (Stud. Comput. Intell., Vol. 434).
12. **C. Duhamel, P. Lacomme, and C. Prodhon**, A GRASPxELS with depth first search split procedure for the HVRP, *Res. Rep. LIMOS/RR-10-08*, Inst. Supér. Inform., Modél. Appl., Aubière, France, 2010. Available at http://www.isima.fr/~lacomme/doc/RR_HVRP1-4_V1.pdf. Accessed Aug. 24, 2015.
13. **C. Duhamel, P. Lacomme, and C. Prodhon**, Efficient frameworks for greedy split and new depth first search split procedures for routing problems, *Comput. Oper. Res.*, **38**, No. 4, 723–739, 2011.
14. **F. Li, B. Golden, and E. Wasil**, A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem, *Comput. Oper. Res.*, **34**, No. 9, 2734–2742, 2007.
15. **P. H. V. Penna, A. Subramanian, and L. S. Ochi**, An iterated local search heuristic for the heterogeneous fleet vehicle routing problem, *J. Heuristics*, **19**, No. 2, 201–232, 2013.
16. **P. H. V. Penna, T. Vidal, L. S. Ochi, and C. Prins**, New compound neighborhoods structures for the heterogeneous fixed fleet vehicle routing problem, *Proc. XLV Simp. Bras. Pesqui. Operac., Natal, Brazil, Sept. 16–19, 2013*, pp. 3623–3633, Soc. Bras. Pesqui. Oper., Rio de Janeiro, 2013. Available at <http://ws2.din.uem.br/~ademir/sbpo/sbpo2013/pdf/arq0110.pdf>.

Accessed Aug. 24, 2015.

17. **B. T. Poljak**, Subgradient methods: A survey of Soviet research, in C. Lemarechal and R. Mifflin, eds., *Nonsmooth Optimization*, (Proc. IASA Workshop, Laxenburg, Austria, Mar. 28 – Apr. 8, 1977), pp. 5–29, Pergamon Press, Oxford, GB, 1977 (IASA Proc. Ser., Vol. 3).
18. **J.-Y. Potvin** and **M.-A. Naud**, Tabu search with ejection chains for the vehicle routing problem with private fleet and common carrier, *J. Oper. Res. Soc.*, **62**, No. 2, 326–336, 2011.
19. **C. Prins**, Efficient heuristics for the heterogeneous fleet multitrip VRP with application to a large-scale real case, *J. Math. Model. Algorithms*, **1**, No. 2, 135–150, 2002.
20. **C. Prins**, Two memetic algorithms for heterogeneous fleet vehicle routing problems, *Eng. Appl. Artif. Intell.*, **22**, No. 6, 916–928, 2009.
21. **A. Subramanian**, **P. H. V. Penna**, **E. Uchoa**, and **L. S. Ochi**, A hybrid algorithm for the heterogeneous fleet vehicle routing problem, *Eur. J. Oper. Res.*, **221**, No. 2, 285–295, 2012.
22. **E. D. Taillard**, A heuristic column generation method for the heterogeneous fleet VRP, *RAIRO, Oper. Res.*, **33**, No. 1, 1–14, 1999.
23. **C. D. Tarantilis**, **C. T. Kiranoudis**, and **V. S. Vassiliadis**, A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem, *J. Oper. Res. Soc.*, **54**, No. 1, 65–71, 2003.
24. **C. D. Tarantilis**, **C. T. Kiranoudis**, and **V. S. Vassiliadis**, A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem, *Eur. J. Oper. Res.*, **152**, No. 1, 148–158, 2004.
25. **C. D. Tarantilis**, **E. E. Zachariadis**, and **C. T. Kiranoudis**, A guided tabu search for the heterogeneous vehicle routing problem, *J. Oper. Res. Soc.*, **59**, No. 12, 1659–1673, 2008.
26. **T. Vidal**, **T. G. Crainic**, **M. Gendreau**, and **C. Prins**, A unified solution framework for multi-attribute vehicle routing problems, *Eur. J. Oper. Res.*, **234**, No. 3, 658–673, 2014.

Yury A. Kochetov,
Aleksey V. Khmelev

Received
13 March 2015
Revised
15 June 2015