

ЛОКАЛЬНЫЙ ПОИСК С ЧЕРЕДУЮЩИМИСЯ ОКРЕСТНОСТЯМИ

Ю. Кочетов, Н. Младенович, П. Хансен

Приводится обзор современных методов локального поиска, основанных на идее чередующихся окрестностей. Показаны пути гибридизации этих методов с другими метаэвристиками. Приводятся примеры удачного применения данных методов при поиске дискретных структур, существование которых казалось проблематичным.

Введение

Для решения задач дискретной оптимизации методы локального поиска являются, по-видимому, наиболее естественными и наглядными. Впервые такие методы стали применяться в конце 50-х — начале 60-х годов прошлого века [12, 29, 90] и связаны в основном с задачей коммивояжера. Позднее эти идеи использовались и для решения задач размещения, построения сетей, расписаний и др. [83, 86, 65, 5]. Однако довольно быстро выяснилось, что простой локальный спуск не позволяет находить глобальный оптимум задачи, и отсутствие концептуального прогресса ослабило интерес к данному направлению. В последние 15–20 лет наблюдается возрождение этого подхода. Оно связано как с новыми алгоритмическими схемами, построенными на аналогиях с живой и неживой природой, так и с новыми теоретическими результатами в области локального поиска. Изменился общий взгляд на построение локальных алгоритмов. Требование монотонного улучшения по целевой функции больше не является доминирующим. При отказе от монотонности часто удается получать решения с меньшей погрешностью. Многие алгоритмы могут рассматриваться как способ порождения конечных неразложимых цепей Маркова на подходящем множестве состояний.

В настоящей статье приводится краткий обзор методов локального поиска, основанных на идее чередующихся окрестностей [78]. В разд. 1 вводятся основные определения и дается краткое введение в теорию сложности задач поиска локальных оптимумов. Схемы локального поиска с чередующимися окрестностями и гибридные схемы с другими

метаэвристиками приводятся в разд. 2. В разд. 3 на примере задачи коммивояжера демонстрируются разные способы построения окрестностей и их свойства. Наконец, в разд. 4 на задачах из теории графов иллюстрируются возможности данного подхода по обнаружению дискретных структур, существование которых представлялось проблематичным.

1. Основные определения

Пусть пара (\mathcal{S}, f) задает индивидуальную задачу дискретной оптимизации с конечным множеством допустимых решений \mathcal{S} и целевой функцией $f : \mathcal{S} \rightarrow R$. Для определенности будем считать, что требуется найти минимум функции f на множестве \mathcal{S} и само решение $x^* \in \mathcal{S}$, на котором этот минимум достигается. Решение x^* будем называть *глобальным* минимумом и множество глобальных минимумов будем обозначать через \mathcal{S}^* . Для каждого $x \in \mathcal{S}$ определим функцию окрестности $N : \mathcal{S} \rightarrow 2^{\mathcal{S}}$, которая для каждого допустимого решения задает множество *соседних* решений, в некотором смысле близких к данному. Множество $N(x)$ будем называть окрестностью решения x в множестве \mathcal{S} . Функции окрестности могут быть достаточно разнообразными, а отношение соседства не всегда симметрично (см., например, [69, 100]).

Решение $x \in \mathcal{S}$ называется *локальным* минимумом по отношению к функции N , если

$$f(x) \leq f(y) \text{ для всех } y \in N(x).$$

Множество локальных минимумов обозначим через $\widehat{\mathcal{S}}$. Это множество, очевидно, зависит от выбора функции N .

ОПРЕДЕЛЕНИЕ 1. Функцию окрестности будем называть *точной*, если $\widehat{\mathcal{S}} \subseteq \mathcal{S}^*$.

Стандартный алгоритм локального спуска начинает работу с некоторого начального решения x_0 , выбранного случайно или с помощью какого-либо вспомогательного алгоритма. На каждом шаге локального спуска происходит переход от текущего решения к соседнему решению с меньшим значением целевой функции до тех пор, пока не будет достигнут локальный оптимум.

Алгоритмы локального поиска широко применяются для решения NP-трудных задач дискретной оптимизации. Вместе с тем многие полиномиально разрешимые задачи могут быть решены точно методом локального спуска. При подходящем выборе полиномиальной окрестности справедливо утверждение: допустимое решение не является глобальным оптимумом, если и только если в его окрестности существует лучшее решение. Ниже приводятся несколько примеров таких задач, подробное описание которых можно найти в [4]. Данные примеры указывают на

важность локального поиска при построении оптимизационных алгоритмов и достаточно общий характер этого подхода.

1. **ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ.** Алгебраически, если предполагать невырожденность задачи, базисное допустимое решение не является оптимальным, если и только если оно может быть улучшено локальным изменением базиса, т. е. заменой одной базисной переменной на небазисную. Такой подход позволяет рассматривать задачу линейного программирования как комбинаторную. Геометрически алгоритм симплекс метода можно интерпретировать как движение по вершинам многогранника допустимой области. Вершина не является оптимальной, если и только если существует смежная с ней вершина с меньшим значением целевой функции. Получающаяся окрестность является точной и имеет полиномиальную мощность.

2. **МИНИМАЛЬНОЕ ОСТОВНОЕ ДЕРЕВО.** Остовное дерево не является оптимальным, если и только если локальной перестройкой, добавляя одно ребро и удаляя из образовавшегося цикла другое ребро, можно получить новое остовное дерево с меньшим суммарным весом. Операция локальной перестройки задает отношение соседства на множестве остовных деревьев. Окрестность любого дерева имеет полиномиальную мощность, а функция окрестности является точной.

3. **МАКСИМАЛЬНОЕ ПАРОСОЧЕТАНИЕ.** Паросочетание не является максимальным, если и только если существует увеличивающий путь. Два паросочетания называются соседними, если их симметрическая разность образует путь. Определенная таким образом окрестность является точной и имеет полиномиальную мощность.

Аналогичные утверждения справедливы для взвешенных паросочетаний, совершенных паросочетаний минимального веса, задач о максимальном потоке и потоке минимальной стоимости.

На каждом шаге локального спуска функция окрестности N задает множество возможных направлений движения. Очень часто это множество состоит из нескольких элементов и имеется определенная свобода в выборе следующего решения. Правило выбора может оказать существенное влияние на временную сложность алгоритма и результат его работы. Например, в задаче о максимальном потоке алгоритм Форда — Фалкерсона (который тоже можно рассматривать как вариант локального спуска) имеет полиномиальную временную сложность при выборе кратчайшего пути для увеличения потока и экспоненциальную временную сложность без гарантии получить глобальный оптимум при произвольном выборе пути [6]. Таким образом, при разработке алгоритмов важно не только правильно определить окрестность, но и верно задать правило выбора направления спуска. Интуитивно кажется, что

в окрестности надо брать элемент с наименьшим значением целевой функции. Однако, как увидим ниже, разумным оказывается не только такой выбор, но и движение в «абсурдном» направлении, когда несколько шагов с ухудшением впоследствии могут привести (и часто приводят) к лучшему локальному оптимуму.

При выборе окрестности хочется иметь множество $N(x)$ как можно меньшей мощности, чтобы сократить трудоемкость одного шага. С другой стороны, более широкая окрестность может привести к лучшему локальному оптимуму. Приходится искать оптимальный баланс между этими противоречивыми факторами. Один из путей разрешения этого противоречия состоит в разработке сложных окрестностей, размер которых можно варьировать в ходе локального поиска. Приведем один из удачных примеров такой окрестности для задачи о разбиении графа на две равные части.

Задан граф $G = (V, E)$ с четным числом вершин $|V| = 2n$ и неотрицательными весами на ребрах $w_e \geq 0$, $e \in E$. Требуется найти разбиение множества V на две равные части $V = V_1 \cup V_2$, $|V_1| = |V_2|$, с минимальной суммой весов ребер, соединяющих V_1 и V_2 . Для $a \in V_1$ и $b \in V_2$ обозначим через $g(a, b)$ изменение целевой функции при замене a на b :

$$g(a, b) = \sum_{(a,v) \in E, v \in V_2 \setminus \{b\}} w(a, v) - \sum_{(a,u) \in E, u \in V_1} w(a, u) \\ + \sum_{(b,u) \in E, u \in V_1 \setminus \{a\}} w(b, u) - \sum_{(b,v) \in E, v \in V_2} w(b, v).$$

Окрестность «1-замена» включает в себя n^2 разбиений по всем парным заменам. Более широкая окрестность « k -замена» определяется аналогично для всех k -парных замен. Понятно, что поиск в окрестности « k -замена» для больших k хотя и полиномиален, но вряд ли оправдан с вычислительной точки зрения. Окрестность Лина — Кернигхана содержит k разбиений, $k \leq n$. Она определяется следующей последовательностью шагов.

Шаг 1. Найти пару (a, b) , $a \in V_1$, $b \in V_2$ с максимальным значением $g(a, b)$.

Шаг 2. Поменять местами a и b .

Шаг 3. Повторить k раз шаги 1, 2, не рассматривая вершины, уже использованные на предыдущих шагах.

Полученная последовательность $\{(a^\tau, b^\tau)\}_{\tau \leq k}$ определяет k соседних решений для разбиения (V_1, V_2) . Изменение целевой функции для соседа τ определяется равенством

$$G_\tau = \sum_{i=1}^{\tau} g(a^i, b^i).$$

При $\tau = n$ множество V_1 перейдет в V_2 и $G_n = 0$.

Окрестность Лина — Кернигхана включает в себя окрестность «1-замена» и является частью окрестности « k -замена». Ее основная идея состоит в том, чтобы не просматривать всю окрестность « k -замена», а с помощью жадной процедуры постараться найти в ней наилучшую последовательность парных замен. Указанная идея может быть легко адаптирована для многих комбинаторных задач, включая задачи размещения, коммивояжера и др.

ОПРЕДЕЛЕНИЕ 2. *Графом окрестностей* $G_N = (\mathcal{S}, E)$ будем называть взвешенный ориентированный граф, вершины которого соответствуют допустимым решениям задачи, а дуги — упорядоченным парам (x, y) , если $y \in N(x)$. Веса в этом графе приписаны вершинам и равны соответствующим значениям целевой функции.

Граф окрестностей G_N иногда называют ландшафтом целевой функции, или просто ландшафтом (*landscape, fitness landscape* [75, 97]). При определении функции окрестности N важно следить за тем, чтобы получающийся граф G_N был *строго связан*, т. е. для каждой пары вершин x и y существовал путь из x в y . Это свойство является важным при анализе асимптотического поведения алгоритмов. Если же это свойство не выполняется, то стараются получить хотя бы свойство *вполне связности*, когда из любой вершины существует путь в некоторую вершину $x^* \in \mathcal{S}^*$ [19, 69]. Если же и этого свойства нет, то приходится либо ограничиться локальными оптимумами, либо переопределить функцию окрестности.

Анализ методов локального поиска в последние годы интенсивно ведется в двух направлениях: эмпирическом и теоретическом. Как ни странно, но эти направления дают существенно разные оценки возможностям локального поиска.

ЭМПИРИЧЕСКИЕ РЕЗУЛЬТАТЫ. Для многих NP-трудных задач локальный поиск позволяет находить приближенные решения, близкие по целевой функции к глобальному оптимуму. Трудоемкость алгоритмов часто оказывается полиномиальной, причем степень полинома достаточно мала. Так, для задачи о разбиении множества вершин графа на две равные части разработаны алгоритмы локального поиска со средней временной сложностью $O(n \log n)$, n — число вершин, которые дают всего несколько процентов относительной погрешности [62].

Для задачи коммивояжера алгоритмы локального поиска являются наилучшими с практической точки зрения. Один из таких алгоритмов с окрестностью Лина — Кернигхана в среднем имеет погрешность около 2%, и максимальная размерность решаемых задач достигает 10^6 городов [61, 36]. На случайно сгенерированных задачах такой большой

размерности итерационная процедура Джонсона позволяет находить решения с отклонением около 0,5% за несколько минут на современных компьютерах. Аналогичные результаты для реальных задач размерностью до 10^5 получены в [102].

Для задач теории расписаний, размещения, покрытия, раскраски и многих других алгоритмы локального поиска позволяют получать превосходные результаты. Более того, их гибкость при изменении математической модели, простота реализации и наглядность превращают локальный поиск в мощное средство для решения практических задач.

ТЕОРЕТИЧЕСКИЕ РЕЗУЛЬТАТЫ. Исследования локального поиска с точки зрения гарантированных оценок качества решения показывают границы его возможностей. Построены трудные для локального поиска примеры, из которых следует, что

- 1) минимальная точная окрестность может иметь экспоненциальную мощность [103];
- 2) число шагов для достижения локального оптимума может оказаться экспоненциальным [99];
- 3) значение локального оптимума может сколь угодно сильно отличаться от глобального оптимума [4].

Полученные теоретические результаты дают весьма пессимистическую картину, что объясняется, по-видимому, самим подходом — оценкой худшего случая. Эти результаты подталкивают к более детальному изучению задачи нахождения произвольного локального оптимума, ее трудоемкости в худшем и среднем случаях.

Абстрактная (массовая) задача локального поиска L задается множеством ее индивидуальных примеров, каждый из которых однозначно определяет целевую функцию, функцию окрестности и множество допустимых решений.

ОПРЕДЕЛЕНИЕ 3 [105]. Задача L принадлежит к классу PLS (polynomial-time local search), если за полиномиальное время от длины записи исходных данных можно выполнить следующие три операции:

- 1) для произвольного примера l проверить принадлежность l к L , и если $l \in L$, то найти допустимое решение x_0 задачи l ;
- 2) для любого решения x проверить его допустимость и вычислить на этом решении значение целевой функции;
- 3) узнать, является ли данное допустимое решение x локальным оптимумом для l , и если нет, то найти в его окрестности решение с лучшим значением целевой функции.

В классе PLS содержатся полиномиально разрешимые задачи, т. е. задачи, в которых за полиномиальное время можно найти локальный оптимум. Многие комбинаторные задачи без весовых функций

относятся к таковым: задачи о клике, о покрытии, о независимом множестве с любой полиномиальной по мощности окрестности и др. Однако в общем случае вопрос о вычислительной сложности нахождения локального оптимума остается открытым. Надежду на его положительное решение дает следующее утверждение.

Теорема 1 [105]. *Если задача L поиска локального оптимума принадлежит классу PLS и является NP-трудной, то $NP = co-NP$.*

Для задач локального поиска определяется понятие PLS-сведения.

ОПРЕДЕЛЕНИЕ 4. Пусть L_1 и L_2 — две задачи локального поиска. PLS-сведение задачи L_1 к задаче L_2 состоит в построении двух полиномиально вычислимых функций h и g таких, что

- 1) h переводит любую индивидуальную задачу $l \in L_1$ в индивидуальную задачу $h(l) \in L_2$;
- 2) g переводит пару (решение задачи $h(l), l$) в решение задачи l ;
- 3) если $l \in L_1$ и x — локальный оптимум для $h(l) \in L_2$, то $g(x, l)$ — локальный оптимум для задачи l .

Если такие функции h и g удается построить, то говорят, что задача L_1 PLS-сводится к задаче L_2 . Понятие PLS-сводимости обладает свойством транзитивности, и если L_2 полиномиально разрешима, а L_1 PLS-сводится к L_2 , то L_1 полиномиально разрешима. Говорят, что задача $L \in PLS$ является PLS-полной, если любая задача в PLS может быть PLS-сведена к ней. Проще говоря, PLS-полные задачи являются наиболее трудными в этом классе и если хотя бы одна из них может быть решена за полиномиальное время, то все остальные задачи могут быть решены за полиномиальное время. PLS-полнота установлена для следующих задач.

- Задача о максимальном разрезе [94]. Задан неориентированный граф $G = (V, E)$ с неотрицательными весами на ребрах $w_e \geq 0, e \in E$. Допустимым решением является любое разбиение множества вершин на две части. Целевая функция — суммарный вес ребер, соединяющих две части разбиения. Окрестность Flip определяется как множество всех решений, полученных из данного переносом любой вершины из одной части разбиения в другую.
- Задача о выполнимости на максимум с окрестностью Flip в двух постановках.
 1. Задана булева функция в конъюнктивной нормальной форме. Каждая конъюнкция имеет неотрицательный вес и содержит не более двух булевых переменных или их отрицаний. Требуется найти значения переменных, при которых суммарный вес

конъюнкций, принимающих значение «истина», является максимальным [68].

2. Заданы наборы, состоящие не более чем из трех литералов. Литерал есть булева переменная без отрицания или константа 0, 1. Набор принимает значение «истина», если не все значения его литералов совпадают. Каждому набору приписан неотрицательный вес. Требуется найти значение переменных, при которых суммарный вес наборов, принимающих значение «истина», является максимальным [94].

- Задача о разбиении графа на две равные части с окрестностями Лина — Кернигхана [64], «1-замена» и их модификациями [94].

Для указанных задач установлено, что в худшем случае стандартный алгоритм локального спуска требует экспоненциального числа шагов при любом выборе направления спуска. Однако для задачи линейного программирования симплекс метод в худшем случае тоже имеет экспоненциальную оценку числа шагов. Тем не менее локальный (глобальный) оптимум можно найти за полиномиальное время методом эллипсоидов. Таким образом, есть надежда, что и для перечисленных выше задач тоже будут найдены полиномиальные алгоритмы получения локального оптимума.

Приведем еще два результата о точной окрестности [105]. Пусть оптимизационная задача ОР и функция окрестности N таковы, что соответствующая задача поиска локального минимума лежит в классе PLS. Обозначим через DP задачу распознавания, в которой для произвольного числа v требуется определить, существует ли в задаче ОР решение со значением целевой функции не более v .

Теорема 2. *Если задача DP является NP-полной, то функция окрестности N не может быть точной, если $NP \neq co-NP$.*

Теорема 3. *Если задача ОР является NP-трудной и целевая функция полиномиально ограничена, то функция окрестности N не может быть точной, если $P \neq NP$.*

2. Поиск с чередующимися окрестностями

Методы локального поиска получили свое дальнейшее развитие в так называемых метаэвристиках [85]. Ниже пойдет речь об одной из них, получившей название поиск с чередующимися окрестностями (variable neighborhoods search) [78, 53, 55]. Её основная идея состоит в систематическом изменении функции окрестности и соответствующем изменении ландшафта в ходе локального поиска. Существует много вариантов ее

реализации, особенно для задач большой размерности. Легкость адаптации этой идеи к различным моделям, гибкость и высокая эффективность объясняют ее конкурентоспособность при решении NP-трудных задач, в частности задач о p -медиане [27, 93, 51, 73], задачи Вебера [17], задач кластеризации и размещения [18, 54, 56, 79], коммивояжера [78, 84, 52, 22, 37, 38, 93], построения маршрутов для транспортных средств [10, 14, 60, 96, 28, 39, 58, 77] и расписаний [11, 21, 23, 30, 31, 66, 50], построения сетей [15, 26], поиска линейных порядков [44], деревьев Штейнера [76, 92, 101], максимальных разрезов [35], клик максимальной мощности [57], деревьев с заданным числом ребер [81] и многих других.

2.1. Основные схемы

Обозначим через N_k , $k = 1, \dots, k_{\max}$, конечное множество функций окрестностей, предварительно выбранных для локального поиска. Предлагаемый метод с чередующимися окрестностями опирается на следующие три тезиса.

- *Локальный минимум относительно одной окрестности не обязательно является локальным минимумом относительно другой окрестности.*
- *Глобальный минимум является локальным относительно любой окрестности.*
- *Для NP-трудных задач в среднем локальные минимумы значительно ближе к глобальному, чем случайно выбранная точка. Распределение локальных минимумов не является равномерным. Они расположены достаточно близко друг к другу, занимая малую часть допустимой области.*

Последнее утверждение известно в литературе как гипотеза о большой долине [13]. Она позволяет сузить область поиска глобального оптимума, используя информацию об уже обнаруженных локальных оптимумах. Именно эта гипотеза лежит в основе алгоритмов связывающих путей (Path Relinking) [42], различных операторов скрещивания (crossover) для генетических алгоритмов [43] и многих других подходов.

Метод чередующихся окрестностей может быть реализован одним из трех способов: детерминированным, вероятностным или смешанным, сочетающим в себе два предыдущих.

(i) Детерминированный локальный спуск с чередующимися окрестностями (VND) предполагает фиксированным порядок смены окрестностей и поиск локального минимума относительно каждой из них. Ниже приведена общая схема этого алгоритма.

Алгоритм VND

Шаг 1. Выбрать окрестности N_k , $k = 1, \dots, k_{\max}$, и начальную точку x .

Шаг 2. Повторять, пока не выполнен критерий остановки.

2.1. $k \leftarrow 1$.

2.2. Повторять до тех пор, пока $k \leq k_{\max}$:

(a) найти $x' \in N_k(x)$ такое, что $f(x') = \min\{f(y) \mid y \in N_k(x)\}$;

(b) если $f(x') < f(x)$, то положить $x \leftarrow x'$, $k \leftarrow 1$, иначе $k \leftarrow k + 1$.

Наряду с указанным выше последовательным порядком смены окрестностей можно предложить и групповую стратегию. Например, для трех окрестностей данный алгоритм с $k_{\max} = 2$ будем применять для каждого элемента третьей окрестности [16, 81].

(ii) Вероятностный локальный спуск с чередующимися окрестностями (RVNS) получается из предыдущего метода при случайном выборе точек из окрестности $N_k(x)$. Этап поиска наилучшей точки в окрестности опускается.

Алгоритм RVNS

Шаг 1. Выбрать окрестности N_k , $k = 1, \dots, k_{\max}$, и начальную точку x .

Шаг 2. Повторять пока не выполнен критерий остановки.

2.1. $k \leftarrow 1$.

2.2. Повторять пока $k \leq k_{\max}$:

(a) случайно выбрать точку $x' \in N_k(x)$;

(b) если $f(x') < f(x)$, то положить $x \leftarrow x'$, $k \leftarrow 1$, иначе $k \leftarrow k + 1$.

Алгоритм RVNS наиболее продуктивен при решении задач большой размерности, когда применение детерминированного варианта требует слишком много машинного времени для выполнения одной итерации. Отметим, что в этом случае наилучшее значение параметра k_{\max} часто оказывается равным 2, а число итераций без смены лучшего найденного решения (шаг 2.2.(b)), как правило, используется в качестве критерия остановки. По сравнению с методами Монте-Карло данный алгоритм отличается большей систематичностью и эффективностью [80]. Для задачи о p -медиане алгоритм RVNS дает решение того же качества, что и алгоритм VND, но в 20–40 раз быстрее [56].

(iii) Основная схема локального поиска с чередующимися окрестностями (VNS) является комбинацией двух предыдущих вариантов [78].

Алгоритм VNS

Шаг 1. Выбрать окрестности N_k , $k = 1, \dots, k_{\max}$, и начальную точку x .

Шаг 2. Повторять, пока не выполнен критерий остановки.

2.1. $k \leftarrow 1$.

2.2. Повторять, пока $k \leq k_{\max}$:

(а) случайно выбрать точку $x' \in N_k(x)$;

(б) применить локальный спуск с начальной точки x' , не меняя координат, по которым x и x' совпадают. Полученный локальный оптимум обозначается x'' ;

(с) если $f(x'') < f(x)$, то полагается $x \leftarrow x'', k \leftarrow 1$, иначе $k \leftarrow k + 1$.

В качестве критерия остановки может использоваться максимальное время счета или максимальное число итераций без смены лучшего найденного решения. Как уже указывалось выше, наряду с последовательным порядком смены окрестностей могут использоваться различные групповые стратегии. Случайный выбор точки x' на шаге 2.2.(а) применяется для избежания заикливания, которое может иметь место при детерминированном варианте. На шаге 2.2.(б) локальный спуск может снова использовать различные схемы с чередующимися окрестностями.

2.2. Модификации основной схемы

Существует много вариантов предложенной выше основной схемы. Например, на шаге 2.2.(с) переход в следующую точку осуществляется только при уменьшении значения целевой функции. Это условие можно заменить по аналогии с алгоритмом имитации отжига [7, 8] и разрешать движение в точку с бóльшим значением целевой функции с некоторой вероятностью, зависящей от этого ухудшения. Идею наискорейшего спуска можно осуществить в рамках основной схемы внесением следующих изменений. Вместо шага 2.2, на котором выполняется переход согласно k -й окрестности, следует выбрать наилучший переход по всем рассматриваемым окрестностям. Другой способ состоит в модификации шага 2.2.(а) и выборе не случайной точки x' , а наилучшей в k -й окрестности. Порядок смены окрестностей тоже может варьироваться.

В случае задач большой размерности сложность выполнения одной итерации становится весьма большой и требуются новые подходы для разработки эффективных методов локального поиска. Ниже предлагаются три таких подхода.

(iv) Алгоритмы VNS с декомпозицией на стадии локального поиска (VNDS) подразумевают модификацию шага 2.2.(б) основной схемы. Пусть $V_k(x)$ обозначает часть окрестности $N_k(x)$, которая получается фиксацией части компонент вектора x . Тогда возникает два уровня оптимизации: первый уровень связан с выбором окрестности $V_k(x)$, а второй — с локальным поиском относительно выбранной окрестности.

Алгоритм VNDS можно рассматривать как один из способов встраивания в основную схему VNS классических приближенных методов, которые используются в комбинаторной оптимизации с 60-х годов прошлого столетия. Примеры успешного применения такого подхода можно найти в [96, 98].

(v) Алгоритм VNS с ориентацией на исследование удаленных частей допустимой области (SVNS). Эта модификация основной схемы особенно полезна в тех случаях, когда наилучшее решение уже найдено в некоторой достаточно большой окрестности текущего решения. В такой ситуации для успешного продолжения поиска необходимо найти новые перспективные части допустимой области. Обычно для этих целей используют простую процедуру старта с новой, например, случайной точки. В алгоритме SVNS предлагается более эффективная схема, когда на шаге 2.2.(с) кроме целевой функции во внимание принимается и расстояние $\rho(x, x'')$ от x до x'' . При заданном параметре $\alpha \geq 0$ переход в новую точку x'' осуществляется в том случае, если $f(x'') - \alpha\rho(x, x'') < f(x)$. Другими словами, предпочтение отдается тем точкам, которые имеют малые значения целевой функции на значительном удалении от текущего решения. Значение параметра α не должно быть слишком большим, чтобы переход в новую точку был действительно «разумным» и не слишком малым для реального влияния на поиск оптимального решения. По сути данный параметр является своего рода нормировкой, выравнивающей значения целевой функции и расстояния. Подбор значений α производится экспериментально или с помощью адаптации в ходе локального поиска.

(vi) Параллельный VNS (PVNS) — третий способ повышения эффективности основной схемы. Существует несколько способов распараллеливания [105, 32]:

- 1) параллельный локальный поиск;
- 2) наращивание числа решений, выбираемых в текущей окрестности, и параллельное выполнение локального поиска для каждого из них;
- 3) аналогично 2, но с обменом информацией о наилучшем найденном решении.

Вторая версия распараллеливания давала лучшие результаты. В [28] рассматривалась ее модификация, согласно которой разным процессорам назначались разные типы окрестностей. Работа процессоров прерывалась, как только одному из них удавалось улучшить текущее значение целевой функции. Такой подход хорошо показал себя на задаче о p -медиане. В частности, с его помощью удалось получить наилучшие значения для нескольких тестов большой размерности из электронной библиотеки TSP-LIB [91].

2.3. Гибридные схемы

Наряду с уже представленными схемами многие авторы используют гибридные конструкции VNS с другими метаэвристиками: вероятностными жадными алгоритмами, поиском с запретами, генетическими алгоритмами и др.

(i) VNS и поиск с запретами (Tabu Search или TS). Алгоритмы поиска с запретами [40, 41] имеют большое число приложений [42]. В ходе локального поиска они используют различные виды памяти: короткую память (short-term), где хранится информация о нескольких последних шагах алгоритма; длинную память (long-term) для управления короткой памятью и процедурами интенсификации и диверсификации поиска; память о частоте изменения компонент решения (frequency based memory) и др. Известно как минимум два способа гибридизации TS и VNS: применение TS в общей схеме VNS и последовательное применение разных окрестностей в схеме TS. К настоящему времени опробованы одна схема первого типа и две схемы второго типа. Для решения задачи о p -медиане с обходами клиентов [93] алгоритм TS используется в схеме VNS на шаге 2.2.(b) вместо локального поиска, а в [20] для задачи о составлении расписания дежурств TS используется наряду с локальным поиском. В [14] для задачи составления маршрутов с заданными временными окнами разработана гибридная схема VND и TS. Рассмотрены четыре варианта локального спуска. Каждый из них собирает информацию о ребрах, не входящих в решение, и эта информация используется на последующих итерациях для сокращения размера окрестностей.

Групповой вариант алгоритма VND может быть легко встроен в схему TS. В этом случае контролировать поиск с помощью запретов достаточно только по одной окрестности. В [16] для решения задачи Вебера предлагается несколько эвристик, основанных на групповом варианте VND. Относительно одной из окрестностей часть элементов может рассматриваться как запрещенная, в то время как другие окрестности используются последовательно без запретов. В [37] применяется один и тот же список запретов для нескольких различных окрестностей, используемых последовательно друг за другом, что дает возможность построения разных гибридных схем TS и VNS.

(ii) VNS и вероятностные жадные алгоритмы с адаптацией (Greedy Randomized Adaptive Search Procedure или GRASP). Основная идея алгоритмов типа GRASP [34] состоит в рандомизации жадных алгоритмов с последующим улучшением полученного решения алгоритмами локального поиска. Наиболее естественный путь для гибридизации VNS и GRASP состоит в замене последней стадии локального спуска на более гибкую схему VNS. Такой подход хорошо показал себя при нахождении

деревьев Штейнера [76, 92], в задаче о покупателе, которому нужно приобрести заданный перечень товаров на известных рынках [84], в задаче о максимальном разрезе [35] и др. Проведенные исследования показывают, что сочетание GRASP и VNS приводит к лучшим результатам, чем отдельно GRASP или VNS.

(iii) VNS и алгоритмы для задач со сложной логической структурой ограничений (Constraint Programming или CP). В последнее время идеология CP привлекает к себе все большее внимание исследователей в связи с ее высоким потенциалом при решении трудных практических задач. Под ограничением здесь понимается логическая связь между переменными задачи, которая используется при задании допустимой области. Такие ограничения часто задаются описательно, выражая трудноформализуемые требования к допустимой области. В таких условиях методы локального поиска оказываются эффективными [87, 88, 96]. Для задач составления маршрутов с временными окнами предложены два оператора перестройки решения, которые использовались для локального спуска (VND) и локального поиска по большим окрестностям (Large Neighborhood Search или LNS) в сочетании с методами CP. Алгоритм LNS состоит из двух стадий: сначала из текущего решения удаляется фиксированное число компонент, а затем полученное частичное решение достраивается до допустимого решением вспомогательной задачи небольшой размерности. В [10] эта идея воплощается более систематически: сначала удаляются две компоненты, затем три и т. д. до тех пор, пока процедура восстановления решения приводит к уменьшению целевой функции. Другой пример гибридизации VNS и CP можно найти в [74].

2.4. Асимптотическая сходимость

Свойство асимптотической сходимости исследовалось для различных метаэвристик. Алгоритм имитации отжига был, по-видимому, первым из них, для которого это свойство было установлено [8]. Центральным понятием при изучении асимптотического поведения алгоритмов является цепь Маркова [3], которая описывает поведение алгоритма на некотором множестве состояний. Для алгоритма имитации отжига этим множеством состояний является множество допустимых решений задачи \mathcal{S} . Для алгоритма поиска с запретами это множество имеет более сложную структуру и состоит из пар (x, φ) , где $x \in \mathcal{S}$, а φ — список запретов [1]. Для генетических алгоритмов под состоянием понимается популяция решений. Для VNS этим множеством состояний является множество локальных минимумов. Таким образом, разные по своей структуре метаэвристики несут в себе общую математическую конструкцию

конечных цепей Маркова, что и позволяет исследовать их асимптотические свойства.

ОПРЕДЕЛЕНИЕ 5 [16]. Систему окрестностей $N_k, k = 1, \dots, k_{\max}$, будем называть *остовой*, если $\bigcup_{k=1}^{k_{\max}} N_k(x) = \mathcal{S}$ для любого $x \in \mathcal{S}$.

Если система окрестностей является остовой, то на шаге 2.2.(а) алгоритма VNS при генерации случайной точки имеется положительная вероятность перейти в любую точку множества \mathcal{S} . Это очевидное, но важное свойство гарантирует асимптотическую сходимость наилучшего найденного решения к глобальному оптимуму.

Свойство VNS [16]. Если система окрестностей является остовой, то с ростом числа итераций алгоритма VNS вероятность нахождения точного решения стремится к единице.

Оценить скорость сходимости алгоритма или получить практически важные оценки на число итераций для получения точного решения или приближенного решения с заданной погрешностью пока не удается. Если исходная задача дискретной оптимизации сформулирована в общем виде и не делается никаких предположений о свойствах целевой функции, то возникают принципиальные трудности. Наиболее точно они представлены в теореме 4 (*No Free Lunch Theorem* [104]) (см. также [32, 95]). Предположим, что некоторый алгоритм A превосходит по точности другой алгоритм B , т. е. алгоритм A позволяет находить значение целевой функции с меньшей погрешностью, чем алгоритм B . Например, алгоритм локального подъема превосходит алгоритм локального спуска для задач на максимум. Теорема 4 утверждает, что мощность множества целевых функций, для которых это верно, равна мощности множества функций, для которых верно обратное. Другими словами, в среднем все алгоритмы одинаковы и равноценны, например, алгоритму случайного блуждания или выбору случайной точки в \mathcal{S} . Понятно, что для таких алгоритмов время поиска оптимального решения экспоненциально велико и оценки скорости сходимости не представляют практического интереса.

Приведем точную формулировку теоремы. Пусть множества \mathcal{S}, \mathcal{J} конечны и целевая функция f задает отображение \mathcal{S} в \mathcal{J} . Под алгоритмом будем понимать некоторый итерационный процесс, в ходе которого по уже известным парам $(x^\tau, f(x^\tau)), \tau = 1, \dots, t$, выбирается новая точка $x^{t+1} \in \mathcal{S}, x^{t+1} \neq x^\tau, \tau = 1, \dots, t$. Рассматриваемые алгоритмы не используют никакой дополнительной информации о задаче: линейности, выпуклости, существования производных и т. п., кроме значений целевой функции в уже пройденных точках (Black Box Optimization). Они отличаются друг от друга способом выбора новой точки и выполняют

ровно T итераций. Предположим, что конкретный алгоритм A для целевой функции f получит m значений $f(x^1), \dots, f(x^m)$. По ним можно построить некоторую диаграмму $\vec{c} = \{c_1, \dots, c_{|\mathcal{J}|}\}$. Величина c_i равна числу точек в последовательности $\{x^\tau\}_{\tau=1}^T$ с заданным значением целевой функции из \mathcal{J} . Если для алгоритма A известна диаграмма \vec{c} , то легко найти минимальное значение целевой функции, полученное алгоритмом, если решалась задача на минимум, или максимальное значение для задачи на максимум. Через $P(\vec{c} | f, T, A)$ обозначим условную вероятность получения диаграммы \vec{c} для функции f за T шагов алгоритма A .

Если алгоритм A является детерминированным, то $P(\vec{c} | f, T, A) = 1$ для некоторой диаграммы \vec{c} и $P(\vec{c} | f, T, A) = 0$ для остальных диаграмм. Для вероятностного алгоритма A распределение вероятностей будет более сложным.

Теорема 4 [104]. *Для любой пары алгоритмов A, B справедливо равенство*

$$\sum_f P(\vec{c} | f, T, A) = \sum_f P(\vec{c} | f, T, B).$$

Из теоремы 4 непосредственно следует, что для любой функции $\Phi(\vec{c})$ в среднем по всем f величина $P(\Phi(\vec{c}) | f, T, A)$ не зависит от A . Таким образом, для получения оценки сходимости итерационных алгоритмов необходимо делать какие-то дополнительные предположения о характере целевой функции либо выделять специальные классы исходных данных.

3. Окрестности

Выбор окрестности, как отмечалось выше, играет важную роль при построении алгоритмов локального поиска. От него существенно зависят сложность выполнения одного шага алгоритма, общее число шагов и в конечном счете погрешность получаемых алгоритмом решений. Для каждой задачи можно предложить несколько функций окрестности с разными по мощности множествами $N(x)$ и, как следствие, разными множествами локальных оптимумов. Ниже будут приведены три примера выбора окрестностей для задачи коммивояжера, которые иллюстрируют их свойства и возможные пути построения окрестностей.

3.1. Окрестности на перестановках

Напомним, что задача коммивояжера состоит в нахождении минимального по длине гамильтонова цикла в полном взвешенном ориентированном (или неориентированном) графе с n вершинами. Для удобства ниже будут рассматриваться только неориентированные графы с симметричной целочисленной неотрицательной матрицей расстояний

$w_{ij} = w_{ji}$, $1 \leq i, j \leq n$. Каждое допустимое решение задачи коммивояжера, или тур, будем представлять в виде перестановки $\pi = \{i_1, \dots, i_n\}$. Для $\pi \in \mathcal{S}$ определим значение функции окрестности $N(\pi)$ как множество всех перестановок, отличающихся от π только в двух позициях (*city-swap*). Множество $N(\pi)$ содержит ровно $n(n-1)/2$ элементов, и вычислительная сложность одного шага локального поиска с учетом вычисления целевой функции не превосходит $O(n^2)$ операций.

Обозначим через $W(\pi)$ длину гамильтонова цикла и определим разностный оператор ∇^2 для $W(\pi)$ следующим образом:

$$\nabla^2 W(\pi) = \frac{1}{|N(\pi)|} \sum_{\pi' \in N(\pi)} W(\pi') - W(\pi), \quad \pi \in \mathcal{S}.$$

Оператор $\nabla^2 W(\pi)$ задает среднее отклонение целевой функции $W(\pi)$ в окрестности данной перестановки π . Для локального оптимума $\pi \in \widehat{\mathcal{S}}$ справедливо неравенство $\nabla^2 W(\pi) \geq 0$. Пусть W_{av} — средняя длина тура на множестве всех допустимых решений. Тогда справедливы следующие утверждения.

Теорема 5 [45]. Функция $\widetilde{W} = W - W_{av}$ удовлетворяет уравнению

$$\nabla^2 \widetilde{W} = -\frac{4}{n} \widetilde{W}.$$

Следствие 1. Любой локальный минимум $\pi \in \widehat{\mathcal{S}}$ имеет длину $W(\pi) \leq W_{av}$.

Следствие 2. Начиная с произвольной перестановки, алгоритм локального поиска находит локальный оптимум за $O(nW)$ шагов, если длина максимального тура превосходит среднее значение W_{av} не более чем в 2^W раз.

Аналогичные утверждения справедливы и для следующих задач:

(i) о разбиении множества вершин $2n$ -вершинного графа на два n -элементных подмножества с минимальным суммарным весом ребер, соединяющих вершины этих подмножеств;

(ii) о правильной раскраске вершин графа в n цветов (т. е. такой раскраске, при которой смежные вершины имеют разные цвета);

(iii) о разбиении n -элементного множества на два подмножества одинакового веса;

(iv) о 3-выполнимости булевых формул в следующей постановке: для n булевых переменных задан набор троек; каждая переменная может входить в набор с отрицанием или без него; набор считается выполненным, если он содержит разные значения, т. е. хотя бы одно *истинное* и хотя бы одно *ложное*; требуется узнать, существуют ли значения переменных, при которых все наборы будут выполнены.

3.2. Окрестности Лина — Кернигхана

Рассмотрим гамильтонов цикл как последовательность ребер. Окрестностью гамильтонова цикла C будем называть множество всех гамильтоновых циклов, получающихся из цикла C заменой двух несмежных ребер на два других, инцидентных этим же вершинам. Такую окрестность будем называть *2-замена* (*2-opt*). Она содержит $n(n-3)/2$ элементов, что несколько меньше, чем в предыдущей окрестности на перестановках. Различие между ними изображено на рис. 1. Для окрестности *2-замена* удаление ребер (a, b) и (e, f) приводит к появлению двух новых ребер (a, e) и (b, f) и обратному обходу дуг от d до c . Если в перестановке поменять местами b и e , то появятся четыре новых ребра (a, e) , (e, c) , (d, b) и (b, f) , но обход дуг от c до d останется прежним.

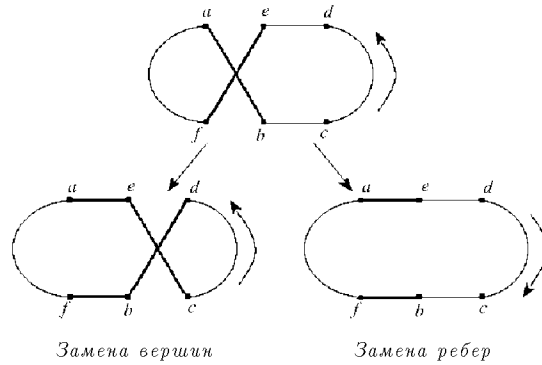


Рис. 1. Различия между окрестностями

В случае, когда вершины расположены на плоскости, а веса w_{ij} являются евклидовыми расстояниями, пересечение ребер (a, b) и (e, f) свидетельствует о возможности улучшения данного тура в силу неравенства треугольника. Однако это условие не является необходимым и улучшение возможно даже без пересечения ребер.

Окрестность *2-замена* может быть естественным образом расширена до *3-замена* и в общем случае до окрестности *k-замена*, где разрешается заменять не более k ребер данного тура на k новых ребер [71].

Теорема 6 [67]. Поиск локального оптимума в задаче коммивояжера с окрестностью *k-замена* является PLS-полной задачей для некоторого k .

Минимальное значение константы k , при котором утверждение остается верным, пока неизвестно. Из работы [67] следует, что утверждение справедливо при $k = 8$ и, возможно, это значение может быть уменьшено до $k = 6$. На практике широко применяются окрестности с $k = 2$,

но теоретических результатов о сложности таких алгоритмов пока не получено.

На основе окрестности 2-замена Лином и Кернигханом предложена эффективная эвристика [72]. Она позволяет заменять произвольное число ребер и переходить от одного тура к другому, используя принципы жадных алгоритмов. Основная идея эвристики заключается в следующем. Из гамильтонова цикла удаляется произвольное ребро, скажем, (a, b) . В полученном пути один конец (вершина a) считается фиксированным, а другой конец меняется за счет перестройки гамильтонова пути. При добавлении ребра из вершины b , например (b, c) , образовавшийся единственный цикл разрывается так, чтобы снова получить гамильтонов путь. Для этого придется удалить ребро, инцидентное вершине c . Обозначим его через (c, d) .

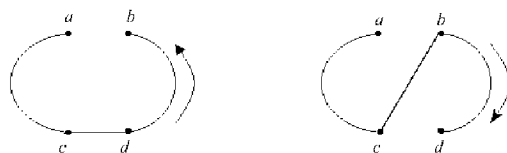


Рис. 2. Процедура ротации

Новый гамильтонов путь имеет концевые вершины a и d (рис. 2). Эту процедуру будем называть ротацией. Для получения нового гамильтонова цикла достаточно добавить ребро (a, d) . Согласно алгоритму Лина — Кернигхана переход от одного тура к другому заключается в удалении некоторого ребра, выполнении серии последовательных ротаций и, наконец, замыкании концевых вершин полученного гамильтонова пути. Существуют различные варианты этой основной схемы, которые отличаются правилами выбора ротаций и ограничениями на множества удаляемых и добавляемых ребер [63]. В алгоритме Лина — Кернигхана ротации выбираются так, чтобы минимизировать разность $w_{bc} - w_{cd}$. При этом множества удаляемых и добавляемых ребер в серии ротаций не должны пересекаться. Последнее ограничение гарантирует, что число ротаций не превысит n^2 и сложность одного шага (перехода от одного тура к другому) останется полиномиальной. Общее число шагов алгоритма, по-видимому, не может быть ограничено сверху полиномом, и известен вариант окрестности Лина — Кернигхана, для которого задача коммивояжера является PLS-полной [105].

3.3. Экспоненциальные окрестности

Одним из новых перспективных направлений в области локального поиска является исследование свойств окрестностей экспоненциальной

мощности. Если лучшее решение в такой окрестности можно найти за полиномиальное время, то алгоритм локального поиска с такой окрестностью получает определенное преимущество перед остальными. Для задачи коммивояжера первые примеры таких окрестностей были предложены в работе [20]. Не будем останавливаться на всех результатах в данной области. Их обзор можно найти в [9]. В качестве иллюстрации изложим идею одного из таких подходов. Этот пример наглядно показывает тесную связь между экспоненциальными окрестностями и полиномиально разрешимыми случаями задачи коммивояжера. Наличие такой связи, по-видимому, будет стимулировать поиск новых полиномиально разрешимых случаев для трудных комбинаторных задач, что, в свою очередь, может привести к дальнейшему прогрессу в области локального поиска.

Основная идея алгоритмов из [89] может быть представлена следующим образом. Предположим, что множество из n городов разбито на две части $\{v_1, \dots, v_m\}$ и $\{u_1, \dots, u_k\}$ так, что $k + m = n$ и $m \leq k$. Для удобства изложения в первую группу добавим $k - m$ фиктивных городов v_{m+1}, \dots, v_k и рассмотрим окрестность $N(u_1, \dots, u_k, v_1, \dots, v_m)$, состоящую из всех туров вида

$$u_1 v_{\tau_1} u_2 v_{\tau_2}, \dots, u_k v_{\tau_k} u_1,$$

где (τ_1, \dots, τ_k) — произвольная перестановка элементов $\{1, \dots, k\}$. Содержательно каждый такой тур получается вставкой городов v_1, \dots, v_k между городами u_1, \dots, u_k . Множество $N(u_1, \dots, u_k, v_1, \dots, v_m)$ состоит из $\lfloor \frac{n+1}{2} \rfloor!$ туров при $2m \leq n \leq 2m + 1$. Оптимальный тур в этом множестве может быть найден с помощью решения задачи о назначениях следующим образом. При $j = 1, \dots, k$ положим

$$d_{ij} = \begin{cases} w(u_j v_i) + w(v_i u_{j+1}) & \text{при } i = 1, \dots, m, \\ w(u_j u_{j+1}) & \text{при } i = m + 1, \dots, k, \end{cases}$$

введем переменные

$$z_{ij} = \begin{cases} 1, & \text{если } v_i \text{ располагается между } u_j, u_{j+1}, \\ 0 & \text{в противном случае} \end{cases}$$

и рассмотрим задачу: найти минимум линейной формы

$$\sum_{i=1}^k \sum_{j=1}^k d_{ij} z_{ij}$$

при условии, что

$$\begin{aligned} \sum_{i=1}^k z_{ij} &= 1, \quad j = 1, \dots, k, \\ \sum_{j=1}^k z_{ij} &= 1, \quad i = 1, \dots, k, \\ z_{ij} &\in \{0, 1\}, \quad i, j = 1, \dots, k. \end{aligned}$$

Оптимальное решение задачи дает наилучшее размещение городов v_i , $i = 1, \dots, m$, между городами u_j , $j = 1, \dots, k$, т. е. оптимальный тур в множестве $N(u_1, \dots, u_k, v_1, \dots, v_m)$. Решение этой задачи может быть найдено с временной сложностью $O(k^3)$, например, алгоритмом Форда — Фалкерсона для поиска максимального потока минимальной стоимости [4, 6] или построением совершенного паросочетания минимального веса в двудольном графе с долями $V = \{v_1, \dots, v_k\}$ и $U = \{u_1, \dots, u_k\}$.

Заметим, что максимальная мощность окрестности $N(u_1, \dots, u_k, v_1, \dots, v_m)$ достигается при $m < n/2$. Точнее, пусть $m = n/2 - p$, $p \geq 0$, целое, $n \geq 5$. Для действительного числа r и нечетного q обозначим через $\lfloor r \rfloor_1$ — максимальное число вида $q/2$, не превосходящее r . Тогда максимальная мощность $N_{\max}(n)$ окрестности превосходит $\lfloor (n+1)/2 \rfloor!$ и равна [46]

$$N_{\max}(n) = (n/2 + p_0)!/2p_0!,$$

где $p_0 = \lfloor \sqrt{\frac{1}{8}n + \frac{9}{8}} + \frac{3}{8} \rfloor$ при четных n и $p_0 = \lfloor \sqrt{\frac{1}{8}n + \frac{9}{8}} + \frac{3}{8} \rfloor_1$ при нечетных n . Применение этой формулы позволяет получить асимптотическое выражение для мощности окрестности.

Теорема 7 [46].

$$N_{\max}(n) = \begin{cases} \Theta(e^{\sqrt{n/2}} \lfloor (n+1)/2 \rfloor! n^{-1/4}), & \text{если } n \text{ четное,} \\ \Theta(e^{\sqrt{n/2}} \lfloor (n+1)/2 \rfloor! n^{-5/4}), & \text{если } n \text{ нечетное.} \end{cases}$$

Перейдем к наиболее интригующему свойству данной окрестности, связанному с расстоянием между турами в графе окрестностей. Оказывается, что с помощью множества $N(u_1, \dots, u_k, v_1, \dots, v_m)$ можно так определить новую окрестность, что расстояние между любыми двумя турами не будет превосходить 4. Сам факт, что максимальное расстояние между турами не зависит от размерности задачи, представляется удивительным. Более того, эта константа очень мала; достаточно сделать всего четыре шага, чтобы добраться из заданного решения до любого другого и, в частности, до оптимального. К сожалению, неизвестно, в какую именно сторону нужно сделать эти четыре шага, иначе задача

оказалась бы полиномиально разрешимой. Кроме того, мы не знаем, как много локальных оптимумов содержится в такой экспоненциальной окрестности. Тем не менее сам факт ограниченности диаметра свидетельствует о больших потенциальных возможностях такой окрестности.

Пусть T — произвольный гамильтонов цикл в полном взвешенном неориентированном n -вершинном графе G . Обозначим через \mathcal{F}_{nm} семейство всех подмножеств мощности m , состоящих из несмежных вершин тура T . Каждый элемент семейства $(v_1, \dots, v_m) \in \mathcal{F}_{nm}$ однозначно определяет множество туров $N(u_1, \dots, u_{n-m}, v_1, \dots, v_m)$. Объединение таких множеств обозначим через $N_m(T)$. Можно показать, что при фиксированном m

$$|\mathcal{F}_{nm}| = \binom{n-m}{m} + \binom{n-m-1}{m-1},$$

т. е. мощность семейства \mathcal{F}_{nm} выражается полиномом от n и, следовательно, оптимальный тур в множестве $N_m(T)$ можно найти за полиномиальное время. Будем говорить, что тур R является соседним к туру T , если в туре T найдутся m несмежных вершин v_1, \dots, v_m таких, что $R \in N(u_1, \dots, u_{n-m}, v_1, \dots, v_m)$.

Теорема 8 [47]. При $m = \lfloor (n-1)/2 \rfloor$ для любых туров T_1 и T_5 из G существуют такие туры T_2 , T_3 и T_4 в G , что T_i является соседним к T_{i-1} , $i = 2, 3, 4, 5$.

В общем случае при произвольном m длина пути между двумя произвольными турами в графе окрестностей G_N не превосходит $4\lfloor (n-1)/2 \rfloor / m$.

4. Поиск дискретных структур

Приведем два примера удачного применения методов локального поиска в теории графов. Оба примера используют один и тот же прием сведения задачи поиска объекта с заданными свойствами к задаче дискретной оптимизации. На содержательном уровне этот прием состоит в задании желаемых свойств в виде некоторых ограничений. Целевая функция задачи дискретной оптимизации состоит в поиске решения с минимальным значением суммарной невязки. Если для такой задачи удалось найти решение с нулевой невязкой, то нужный объект найден. Таким образом, для алгоритмов локального поиска имеется простой критерий останова. Если объект не существует, то алгоритм может работать бесконечно долго. Вероятностные алгоритмы, дающие точное решение, но время работы которых является случайной величиной, называются алгоритмами типа Лас-Вегас [82]. Приведем примеры применения алгоритмов этого типа.

4.1. Палиндромные деревья

Пусть G — связный граф и D — его диаметр. Обозначим через $d(G, k)$ число неупорядоченных пар вершин (v, u) в G , для которых расстояние между v и u равно k . По определению $d(G, 0)$ есть число вершин в G . Полином вида

$$H(G) = H(G, \lambda) = \sum_{k=0}^D d(G, k)\lambda^k$$

называют полиномом Хосойа [59].

ОПРЕДЕЛЕНИЕ 6 [24]. Граф G с диаметром D называется H -палиндромным, если равенство $d(G, k) = d(G, D - k)$ выполняется при всех $k = 0, 1, \dots, \lfloor D/2 \rfloor$.

ОПРЕДЕЛЕНИЕ 7 [24]. Граф G с диаметром D называется H^* -палиндромным, если равенство $d(G, k) = d(G, D - k + 1)$ выполняется при всех $k = 1, \dots, \lfloor D/2 \rfloor$.

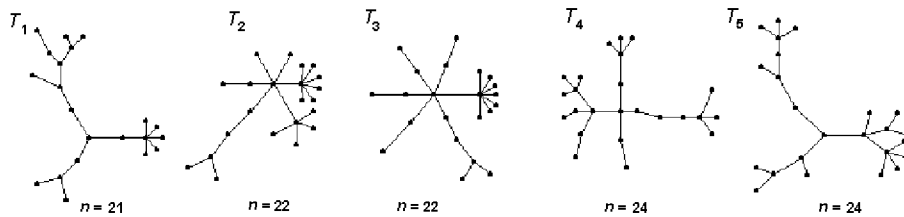


Рис. 3. H -палиндромные деревья

В 1993 г. И. Гутман сделал предположение, что H -палиндромные деревья не существуют [48]. Позднее было найдено три примера H^* -палиндромных деревьев с $n \leq 12$ вершинами [49] и семейство H -палиндромных графов, не являющихся деревьями [2]. Эти находки стимулировали дальнейший поиск H -палиндромных деревьев, которые и были обнаружены при $n \geq 21$. Для меньшей размерности таких объектов не существует. На рис. 3 изображены деревья, найденные методами локального поиска [52, 53]. Диаметры деревьев равны 6 или 8, а полиномы Хосойа имеют следующий вид:

$$\begin{aligned} H(T_1) &= 21 + 20\lambda + 34\lambda^2 + 25\lambda^3 + 31\lambda^4 + 25\lambda^5 + 34\lambda^6 + 20\lambda^7 + 21\lambda^8; \\ H(T_2) &= H(T_3) = 22 + 21\lambda + 52\lambda^2 + 63\lambda^3 + 52\lambda^4 + 21\lambda^5 + 22\lambda^6; \\ H(T_4) &= 24 + 23\lambda + 37\lambda^2 + 41\lambda^3 + 50\lambda^4 + 41\lambda^5 + 37\lambda^6 + 23\lambda^7 + 24\lambda^8; \\ H(T_5) &= 24 + 23\lambda + 39\lambda^2 + 41\lambda^3 + 46\lambda^4 + 41\lambda^5 + 39\lambda^6 + 23\lambda^7 + 24\lambda^8. \end{aligned}$$

Число деревьев, H - и H^* -палиндромных деревьев

n	Деревья	H -палиндромные деревья	H^* -палиндромные деревья
4	2	0	1
5	3	0	0
6	6	0	0
7	11	0	1
8	23	0	0
9	47	0	1
10	106	0	0
11	235	0	0
12	551	0	0
13	1301	0	3
14	3159	0	0
15	7741	0	0
16	19320	0	5
17	48629	0	6
18	123867	0	0
19	317955	0	0
20	823065	0	0
21	2144505	1	39
22	5623756	2	11
23	14828074	0	2
24	39299897	2	0
25	104636890	0	410
26	279793450	0	69

В данном случае целевая функция $f(G)$ оптимизационной задачи определяется выражением

$$f(G) = \sum_{k=0}^{\lfloor D/2 \rfloor} |d(G, k) - d(G, D - k)|$$

и может интерпретироваться как расстояние до H -палиндромного графа, если он существует. В качестве ограничений задачи выступает требование, чтобы граф был деревом. В таблице приведены результаты поиска H - и H^* -палиндромных деревьев с $n \leq 26$ вершинами. Полным

перебором было проверено, что других H - и H^* -палиндромных деревьев с $n \leq 26$ вершинами нет.

Все найденные H -палиндромные деревья имеют четный диаметр. Поэтому было бы интересно найти такое дерево с нечетным диаметром. Интенсивные поиски таких деревьев для $n = 32$ и $n = 36$ не дали положительных результатов, но было замечено, что при любом n , $5 \leq n \leq 30$, и нечетном диаметре

$$\min f(G) = \lceil n/2 \rceil.$$

Это наблюдение послужило поводом для нового предположения [24].

Гипотеза 1. Не существуют H -палиндромные деревья с нечетным диаметром.

4.2. Индекс Рандика и средние расстояния

На основе методов локального поиска и в первую очередь методов с чередующимися окрестностями разработана компьютерная система AutoGraphiX [25], ориентированная на экспериментальные исследования в теории графов. Эта система предназначена для достижения следующих целей:

- а) поиска графов, обладающих заданными свойствами;
- б) поиска оптимальных или близких к ним значений инвариантов;
- в) опровержения гипотез;
- г) выдвижения новых гипотез или уточнение старых;
- д) доказательства утверждений.

В частности, с помощью системы AutoGraphiX было опровергнуто три гипотезы, одна гипотеза доказана и несколько гипотез усилено [25].

Обозначим через $L = (l_{pq})$ матрицу, в которой l_{pq} есть расстояние между вершинами p и q некоторого графа $G = (V, E)$. Среднее расстояние $\bar{l}(G)$ определяется равенством

$$\bar{l}(G) = \frac{2}{n(n-1)} \sum_{p=1}^{n-1} \sum_{q=p+1}^n l_{pq}.$$

Зададим вес w_{pq} ребра (v_p, v_q) через степени вершин $d_G(v_p)$ и $d_G(v_q)$ [37]:

$$w_{pq} = \frac{1}{\sqrt{d_G(v_p) \cdot d_G(v_q)}}.$$

Тогда индекс Рандика $Ra(G)$ определяется как сумма весов ребер, т. е.

$$Ra(G) = \sum_{(v_p, v_q) \in E} w_{pq}.$$

Гипотеза 2 [37]. Для всех связных графов $\bar{l}(G) \leq Ra(G)$.

Система AutoGraphiX использовалась для минимизации разности $Ra(G) - \bar{l}(G)$ по всем связным графам при $n = 2, 3, \dots, 20$. При малых значениях n система выдавала *звезды* S_n . При бóльших значениях получались более сложные конструкции, так называемые двойные кометы, которые являются локальными оптимумами. Однако на звездах той же размерности значение разности всегда оказывалось меньше. Кроме того, указанная разность быстро росла с ростом n . Для звезд можно проверить, что

$$Ra(S_n) = \sqrt{n-1}, \quad \bar{l}(S_n) = 2 - \frac{2}{n}.$$

Таким образом, приходим к новому более сильному предположению.

Гипотеза 2'. Для всех связных графов

$$Ra(G) - \bar{l}(G) \geq \sqrt{n-1} + \frac{2}{n} - 2,$$

и эта граница достижима для всех $n \geq 1$.

Другие примеры применения системы AutoGraphiX можно найти в [25].

ЛИТЕРАТУРА

1. **Гончаров Е. Н., Кочетов Ю. А.** Вероятностный поиск с запретами для дискретных задач безусловной оптимизации // Дискрет. анализ и исслед. операций. Сер. 2. 2002. Т. 9, № 2. С. 13–30.
2. **Добрынин А. А.** Построение графов с палиндромным полиномом Винера // Вычислительные системы: Сб. науч. тр. Вып. 151. Новосибирск: Ин-т математики СО РАН, 1994. С. 37–54.
3. **Кемени Дж., Снелл Дж.** Конечные цепи Маркова. М.: Наука, 1970.
4. **Пападимитриу Х., Стайглиц К.** Комбинаторная оптимизация: Алгоритмы и сложность. М.: Мир, 1985.
5. **Растрингин Л. А.** Случайный поиск — специфика, этапы истории и предрассудки // Вопросы кибернетики. М.: Науч. совет по комплексной проблеме «Кибернетика» АН СССР, 1978. Вып. 33. С. 3–16.
6. **Форд Л., Фалкерсон Д.** Потoki в сетях. М.: Мир, 1966.
7. **Aarts E. H. L., Korst J. H. M.** Simulated annealing and Boltzmann machines. Chichester: Wiley, 1989.
8. **Aarts E. H. L., Korst J. H. M., Laarhoven van P. J. M.** Simulated annealing // Local search in combinatorial optimization. Chichester: Wiley, 1997. P. 91–120.

9. **Ahnja R. K., James O. E., Orlin B., Punnen A. P.** A survey of very large-scale neighborhood search techniques // *Discrete Appl. Math.* 2002. V. 123, N 1–3. P. 75–102.
10. **Bent R., Van Hentenryck P.** A two stage hybrid local search for the vehicle routing problem with time windows // Report CS-01-06. Dept. of Comput. Sci., Brown Univ., 2001.
11. **Besten den M., Stützle T.** Neighborhood revisited: An experimental investigation into the effectiveness of variable neighborhood descent for scheduling // MIC'2001 — 4th Metaheuristics International Conference. Porto, July 16–21, 2001. P. 545–549.
12. **Bock F.** An algorithm for solving traveling-salesman and related network optimization problems: Abstract // Bulletin Fourteenth National Meeting of the Operations Research Society of America, 1958. P. 897.
13. **Boese K. D., Kahng A. B., Muddu S.** A new adaptive multi-start technique for combinatorial global optimizations // *Oper. Res. Lett.* 1994. V. 16, N 2. P. 101–114.
14. **Braysy O.** Local search and variable neighborhood search algorithm for the vehicle routing with time windows. *Acta Wasaensia.* 2001. N 87. 168 p. (Preprint).
15. **Brimberg J., Hansen P., Lih K.-W., Mladenović N., Breton M.** An oil pipeline design problem // *Les Cahiers du GERAD G-2000-73.* Monreal, Canada, 2000.
16. **Brimberg J., Hansen P., Mladenović N.** Convergence of variable neighborhood search // *Les Cahiers du GERAD G-2000-73.* Monreal, Canada, 2000.
17. **Brimberg J., Hansen P., Mladenović N., Taillard E.** Improvements and comparison of heuristics for solving the uncapacitated multisource Weber problem // *Oper. Res.* 2000. V. 48, N 3. P. 444–460.
18. **Brimberg J., Mladenović N.** A variable neighborhood algorithm for solving the continuous location-allocation problem // *Stud. Locat. Anal.* 1996. V. 10. P. 1–12.
19. **Brucker P., Hurink J., Werner F.** Improving local search heuristics for some scheduling problems. Pt II // *Discrete Appl. Math.* 1997. V. 72, N 1–2. P. 47–69.
20. **Burkard R. E., Deineko V. G., Woeginger G. J.** The traveling salesman problem and the PQ-tree // *Integer programming and combinatorial optimization.* Proc. Berlin: Springer, 1996. P. 490–504. (Lecture Notes in Comput. Sci.; V. 1084).
21. **Burke E. K., Causmaecker de P., Petrovic S., Berghe G. V.** Variable neighbourhood search for nurse rostering problems // MIC'2001 — 4th Metaheuristics Intern. conf. Porto, July 16–21, 2001. P. 755–760.

22. **Burke E. K., Cowling P., Keuthen R.** Effective local and guided variable neighborhood search methods for the asymmetric traveling salesman problem // Proc. of the Evo Workshops. Berlin: Springer, 2001. P. 203–212. (Lecture Notes in Comput. Sci.; V. 2037).
23. **Campos de L. M., Puerta J. M.** Stochastic local algorithms for linearizing belief networks: searching in the space of the orderings // Symbolic and quantitative approaches to reasoning with uncertainty. 6th European conf. ECSQARU 2001 (Toulouse, September 19–21, 2001). Proceedings. Berlin: Springer, 2001. P. 228–239. (Lecture Notes in Comput. Sci.; V. 2143).
24. **Caporossi G., Dobrynin A. A., Gutman I., Hansen P.** Trees with palindromic Hosoya polynomials // Graph Theory Notes N. Y. 1999. V. 37. P. 10–16.
25. **Caporossi G., Hansen P.** Variable neighborhood search for extremal graphs // Discrete Math. 2000. V. 212, N 1–2. P. 29–44.
26. **Costa M–C., Monclar F–R., Zrikem M.** Variable neighborhood search for the optimization of cable layout problem // MIC'2001 — 4th Metaheuristics Intern. conf. Porto, July 16–21, 2001. P. 749–753.
27. **Crainic T. G., Gendreau M., Hansen P., Hoeb N, Mladenović N.** Parallel variable neighbourhood search for the p -median // MIC'2001 — 4th Metaheuristics Intern. conf. Porto, July 16–21, 2001. P. 595–599.
28. **Crispim J., Brandão J.** Reactive tabu search and variable neighbourhood descent applied to the vehicle routing problem with backhauls // MIC'2001 — 4th Metaheuristics Intern. conf. Porto, July 16–21, 2001. P. 631–636.
29. **Croes G. A.** A method for solving traveling salesman problems // Oper. Res. 1958. V. 6. P. 791–812.
30. **Davidović T., Hansen P., Mladenović N.** Variable neighborhood search for multiprocessor scheduling with communication delays // MIC'2001 — 4th Metaheuristics Intern. conf. Porto, July 16–21, 2001. P. 737–741.
31. **Desrosiers J., Mladenović N., Villeneuve D.** Design of balanced MBA student teams // MIC'2001 — 4th Metaheuristics Intern. conf. Porto, July 16–21, 2001. P. 281–285.
32. **Droste S., Jansen T., Wegener I.** Optimization with randomized search heuristics — the NFL theorem, realistic scenarios, and difficult functions // Theoretical Comput. Sci. 2002. V. 287, N 1. P. 131–144.
33. **Fajtlowicz S.** On conjectures of Graffiti // Discrete Math. 1988. V. 72, N 2. P. 113–118.
34. **Feo T., Resende M.** Greedy randomized adaptive search // J. Global Optim. 1995. V. 6. P. 109–133.
35. **Festa P., Pardalos P. M., Resende M. G. C. Ribeiro C. C.** GRASP and VNS for max-cut // MIC'2001 — 4th Metaheuristics Intern. conf. Porto, July 16–21, 2001. P. 371–376.

36. **Fridman M. L., Johnson D. S., McGeoch L. A., Ostheimer G.** Data structures for traveling salesman // *J. Algorithms*. 1995. V. 18, N 3. P. 432–479.
37. **Gendreau M., Hertz A., Laporte G.** New insertion and postoptimization procedures for the traveling salesman problem // *Oper. Res.* 1992. V. 40, N 6. P. 1086–1094.
38. **Gendreau M., Hertz A., Laporte G.** The traveling salesman problem with back hauls // *Comput. Oper. Res.* 1996. V. 23, N 5. P. 501–508.
39. **Ghiani G., Hertz A., Laporte G.** Recent algorithmic advances for arc routing problems // *Les Cahiers du GERAD G-2000-40*. Monreal, Canada, 2000.
40. **Glover F.** Tabu search: pt I // *ORSA J. Comp.* 1989. V. 1. P. 190–206.
41. **Glover F.** Tabu search: pt II // *ORSA J. Comp.* 1990. V. 2. P. 4–32.
42. **Glover F., Laguna M.** Tabu search. Boston: Kluwer Acad. Publ., 1997.
43. **Goldberg D. E.** Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley, 1989.
44. **González C. G., Pérez-Brito D.** A variable neighborhood search for solving linear ordering problem // *MIC'2001 — 4th Metaheuristics Intern. conf.* Porto, July 16–21, 2001. P. 181–185.
45. **Grover L. K.** Local search and the local structure of NP-complete problems // *Oper. Res. Lett.* 1992. V. 12, N 4. P. 235–244.
46. **Gutin G.** Exponential neighborhood local search for the traveling salesman problem // *Comput. Oper. Res.* 1999. V. 26, N 4. P. 313–320.
47. **Gutin G., Yeo A.** Small diameter neighborhood graphs for the traveling salesman problem: at most four moves from tour to tour // *Comput. Oper. Res.* 1999. V. 26, N 4. P. 321–327.
48. **Gutman I.** Some properties of the Wiener polynomial // *Graph Theory Notes N. Y.* 1993. V. 25. P. 13–18.
49. **Gutman I., Estrada E., Ivanciuc O.** Some properties of the Wiener polynomial of trees // *Graph Theory Notes N. Y.* 1999. V 36. P. 7–13.
50. **Hansen P., Jaumard B., Mladenović N., Parreira A.** Variable neighborhood search for weighted maximum satisfiability problem // *Les Cahiers du GERAD G-2000-62*. Monreal, Canada, 2000.
51. **Hansen P., Mladenović N.** Variable neighborhood search for the p -median // *Location Sci.* 1997. V. 5. P. 207–226.
52. **Hansen P., Mladenović N.** An introduction to variable neighborhood search // *Metaheuristics, advances and trends in local search paradigms for optimization*. Dordrecht: Kluwer Acad. Publ., 1999. P. 433–458.
53. **Hansen P., Mladenović N.** Variable neighborhood search: principles and applications (invited review) // *European J. Oper. Res.* 2001. V. 130, N 3. P. 449–467.

54. **Hansen P., Mladenović N.** J-Means: a new local search heuristic for minimum sum-of-squares clustering // *Pattern Recognition*. 2001. V. 34. P. 405–413.
55. **Hansen P., Mladenović N.** Development of variable neighborhood search // *Essays and surveys in metaheuristics*. Dordrecht: Kluwer Acad. Publ., 2002. P. 415–440.
56. **Hansen P., Mladenović N., Perez-Brito D.** Variable neighborhood decomposition search // *J. Heuristics*. 2001, V. 7, N 4. P. 335–350.
57. **Hansen P., Mladenović N.** Urosevic D. Variable neighborhood search for maximum clique // *Les Cahiers du GERAD G-2001-25*. Monreal, Canada, 2001.
58. **Hertz A., Laporte G., Mittaz M.** A tabu search heuristic for the capacitated arc routing problem // *Oper. Res.* 2000. V. 48, N 1. P. 129–135.
59. **Hosoya H.** On some counting polynomials in chemistry // *Discrete Appl. Math.* 1988. V. 19, N 2. P. 239–257.
60. **Hwang F. K., Richards D. S., Winter P.** The Steiner tree problem. Amsterdam: North-Holland, 1992.
61. **Johnson D. S.** Local optimization and the traveling salesman problem // *Automata, languages, and programming*. Proc. Berlin: Springer-Verl., 1990. P. 446–461. (Lecture Notes in Comput. Sci.; V. 443).
62. **Johnson D. S., Aragon C. R., McGeoch L. A., Schevon C.** Optimization by simulated annealing: an experimental evaluation, part I: Graph partitioning // *Oper. Res.* 1989. V. 37, N 6. P. 865–892.
63. **Johnson D. S., McGeoch L. A.** The traveling salesman problem: a case study // *Local search in combinatorial optimization*. Chichester: Wiley, 1997. P. 215–310.
64. **Johnson D. S., Papadimitriou S. H., Yannakakis M.** How easy is local search? // *J. Comput. System Sci.* 1988. V. 37, N 1. P. 79–100.
65. **Kernighan B. W., Lin S.** An efficient heuristic procedure for partitioning graphs // *Bell Syst. Tech. J.* 1970. V. 49. P. 291–307.
66. **Kolish R., Hartman S.** Heuristic algorithms for the resource-constrained project scheduling problem: classification and computational analysis // *Project scheduling: recent models, algorithms and applications*. Dordrecht: Kluwer Acad. Publ., 1999. P. 147–178.
67. **Krentel M. W.** Structure in locally optimal solutions // *Proc. of the 30th annual sympos. on foundation of comput. sci.* Los Alamihos, CA: IEEE Comput. Soc. Press, 1989. P. 216–222.
68. **Krentel M. W.** On finding and verifying locally optimal solutions // *SIAM J. Comput.* 1990. V. 19, N 4. P. 742–751.

-
69. **Laarhoven van P. J. M., Aarts E. H. L., Lenstra J. K.** Job shop scheduling by simulated annealing // *Oper. Res.* 1992. V. 40, N 1. P. 113–125.
 70. **Lepović M., Gutman I.** A collective property of trees and chemical trees // *J. Chem. Inform. Comput. Sci.* 1998. V. 38, N 5. P. 823–826.
 71. **Lin S.** Computer solutions of the traveling salesman problem // *Bell Syst. Tech. J.* 1965. V. 44. P. 2245–2269.
 72. **Lin S., Kernighan B. W.** An efficient heuristic algorithm for the traveling-salesman problem // *Oper. Res.* 1973. V. 21, N 2. P. 498–516.
 73. **Lopez F. G., Batista B. M., Moreno-Perez J., Moreno-Vega M.** The parallel variable neighborhood search for the p -median problem // *Res. Rep. Univ. of La Laguna, Spain*, 2000.
 74. **Loudni S., Boizumault P.** VNS/LDS+CP: A hybrid method for constraint optimization in anytime contexts // *MIC'2001 — 4th Metaheuristics Intern. Conf. Porto*, July 16–21, 2001. P. 761–765.
 75. **Macken C. A., Hagan P. S., Perelson A. S.** Evolutionary walks on rugged landscapes // *SIAM J. Appl. Math.* 1991. V. 51, N 3. P. 799–827.
 76. **Martins S. L., Resende M.G.C., Ribeiro C.C., Pardalos P.** A parallel GRASP for the Steiner tree problem in graphs using a hybrid local search strategy // *J. Global Optim.* 2000. V. 17. P. 267–283.
 77. **Mittaz M.** Problemes de cheminements optimaux dans des reseaux avec contraintes associées aux arcs // *Ph. D. Thesis, Ecole Polytechnique Federale de Lausanne, Switzerland*, 2001.
 78. **Mladenović N., Hansen P.** Variable neighborhood search // *Comput. Oper. Res.* 1997. V. 24. P. 1097–1100.
 79. **Mladenović N., Moreno J. P., Moreno-Vega A.** A chain-interchange heuristic method // *Yugos. J. Oper. Res.* 1996. V. 6. P. 41–54.
 80. **Mladenović N., Petrović J., Kovačević-Vujčić V., Čangalović M.** Solving spread spectrum radar polyphase code design problem by tabu search and variable neighborhood search // *SMG Rep. R-00-09. Univ. Libre Bruxelles. Belgium*, 2000 (to appear in *European J. Oper. Res.*).
 81. **Mladenović N., Urošević D.** Variable neighborhood search for the k -cardinality tree // *MIC'2001 — 4th Metaheuristics Intern. conf. Porto*, July 16–21, 2001. P. 743–748.
 82. **Motwani R., Raghavan P.** *Randomized algorithms.* Cambridge: Cambridge Univ. Press, 1995.
 83. **Nicholson T. A. J.** A sequential method for discrete optimization problems and its application to the assignment, traveling salesman and tree scheduling problems // *J. Inst. Math. Appl.* 1965. V. 13. P. 362–375.

84. **Ochi L. S., Silva M. B., Drummond L.** Metaheuristics based GRASP and VNS for solving the traveling purchaser problem // MIC'2001 — 4th Metaheuristics Intern. conf. Porto, July 16–21, 2001. P. 489–494.
85. **Osman I. H., Laporte G.** Metaheuristics: a bibliography // *Ann. Oper. Res.* 1996. V. 63. P. 513–628.
86. **Page E. S.** On Monte Carlo methods in congestion problems. I: Searching for an optimum in discrete situations // *Oper. Res.* 1965. V. 13, N 2. P. 291–299.
87. **Pesant G., Gendreau M.** A view of local search in constraint programming // *Principles and practice of constraint programming—CP 96.* Berlin: Springer, 1996. P. 353–366 (Lecture Notes in Comput. Sci.; V. 1118).
88. **Pesant G., Gendreau M.** A constraint programming framework for local search methods // *J. Heuristics.* 1999. V. 5. P. 255–279.
89. **Punnen A. P.** The traveling salesman problem: new polynomial approximation algorithms and domination analysis // *Res. Rep. St. John: Univ. New Brunswick*, 1996.
90. **Reiter S., Sherman G.** Discrete optimizing // *J. Soc. Indust. Appl. Math.* 1965. V. 13. P. 864–889.
91. **Resende M. G. C., Pitsoulis L. S., Pardalos P. M.** Approximate solution of weighted MAX-SAT problem using GRASP // *Satisfiability problem: theory and applications.* Providence, PI: Amer. Math. Soc., 1997. P. 393–405.
92. **Ribeiro C., Uchoa E., Werneck R.** A hybrid GRASP with perturbations for the Steiner problem in graphs // *Technical rep. Comput. Sci. Dep. Catholic Univ. of Rio de Janeiro*, 2001.
93. **Rodrigue I., Moreno-Vega M., Moreno-Perez J.** Heuristics for routing-median problems // *SMG Rep. Univ. Libre de Bruxelles. Belgium*, 1999.
94. **Schaffer A. A., Yannakakis M.** Simple local search problems that are hard to solve // *SIAM J. Comput.* 1991. V. 20, N 1. P. 56–87.
95. **Schumacher C.** Black box search — framework and methods. Ph. D. Thesis, 2000. The Univ. of Tennessee, Knoxville, USA.
96. **Shaw P.** Using constraint programming and local search methods to solve vehicle routing problems // *Principles and practice of constraint programming (CP'98)*, 1998. P. 417–431.
97. **Stadler P. F.** Correlation in landscapes of combinatorial optimization problems // *Europhys. Lett.* 1992. V. 20. P. 479–482.
98. **Taillard E., Voss S.** POPMUSIC — Partial optimization metaheuristic under special intensification conditions // *Essays and surveys in metaheuristics.* Dordrecht: Kluwer Acad. Publ., 2001. P. 613–630.
99. **Tovey C. A.** Local improvement on discrete structures // *Local search in combinatorial optimization.* Chichester: Wiley, 1997. P. 57–90.

100. **Verhoeven M. G. A.** Tabu search for resource-constrained scheduling // *European J. Oper. Res.* 1998. V. 106, N 2. P. 266–276.
101. **Verhoeven M. G. A., Severens M. E. M., Aarts E. H. L.** Local search for Steiner trees in graphs // *Modern search methods*. N. Y.: John Wiley and Sons, Inc., 1996. P. 117–129.
102. **Verhoeven M. G. A., Swinkels P. C. J., Aarts E. H. L.** Parallel local search and the traveling salesman problem // Working paper. Philips Res. Lab. Eindhoven, 1995.
103. **Weiner P., Savage S. L., Bagchi A.** Neighborhood search algorithms for quaranteeing optimal traveling salesman tours must be inefficient // *J. Comput. System Sci.* 1976. V. 12, N 1. P. 25–35.
104. **Wolpert D. H., Macready W. J.** No free lunch theorem for search // Rep. SFI-TR-95-02-010. The Santa Fe Inst., Santa Fe, 1995.
105. **Yannakakis M.** Computational complexity // *Local search in combinatorial optimization*. Chichester: Wiley, 1997. P. 19–55.

Адрес автора:

Институт математики
им. С. Л. Соболева СО РАН,
пр. Академика Коптюга, 4,
630090 Новосибирск, Россия.
E-mail: jkochet@math.nsc.ru

Статья поступила
11 ноября 2002 г.