Локальный поиск с чередующимися окрестностями

Ю. Кочетов, Н. Младенович, П. Хансен

Аннотация

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

При решении задач дискретной оптимизации методы локального поиска являются, по-видимому, наиболее естественными и наглядными. Первые шаги их реализации относятся к концу 50-х, началу 60-х годов двадцатого столетия [11, 28, 89] и связаны в основном с задачей коммивояжера. Позднее эти идеи использовались и для задач размещения, построения сетей, расписаний и др. [82, 85, 63, 4]. 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 новыми теоретическими результатами в области локального поиска. Изменился общий взгляд на построение локальных алгоритмов. Требование монотонного улучшения по целевой функции больше не является доминирующим. Наиболее мощные алгоритмы допускают произвольное ухудшение и многие из них могут рассматриваться как способ порождения конечных неразложимых цепей Маркова на подходящем множестве состояний.

В данной работе приводится краткий обзор методов локального поиска, основанных на идее чередующихся окрестностей [77]. В § 1 вводятся основные определения и дается краткое введение в теорию сложности задач поиска локальных оптимумов. В § 2 на примере задачи коммивояжера демонстрируются разные способы построения окрестностей и их свойства. Наиболее продуктивные схемы локального поиска с чередующимися окрестностями и гибридные схемы с другими метаэвристиками приводятся в § 3. Наконец, в § 4 на примерах из теории графов иллюстрируются возможности данного подхода по обнаружению дискретных структур, существование которых представляется проблематичным.

1 Основные определения

Пусть пара $(\mathcal{I}, f)$ задает индивидуальную задачу дискретной оптимизации с конечным множеством допустимых решений $\mathcal{I}$ и целевой функцией $f : \mathcal{I} \to R$. Для определенности будем считать, что требуется найти минимум фунции f на множестве $\mathcal{I}$ и само решение $x^* \in \mathcal{I}$, на котором этот минимум достигается. Решение x^* будем называть глобальным минимумом и множество глобальных минимумов будем обозначать через $\mathcal{I}^*$. Для каждого $x \in \mathcal{I}$ определим функцию окрестности $N : \mathcal{I} \to 2^{\mathcal{I}}$, 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 $\mathcal{I}$. Функция окрестности может быть достаточно сложной и отношение соседства не всегда симметрично, то есть x может быть соседом y, но y может не быть соседом x (см., например, [68, 100]).

Определение 1. Решение $x \in \mathcal{I}$ 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 $\widehat{\mathcal{I}}$. Это множество, очевидно, зависит от выбора функции N.

Определение 2. Функцию окрестности будем называть точной, если $\widehat{\mathcal{I}} \subseteq \mathcal{I}^*$.

Стандартный алгоритм локального спуска начинает свою 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 меньшем 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 рассматриваться как задачи, легко решаемые локальным спуском. При подходящем выборе полиномиальной окрестности соответствующая теорема может быть сформулированна в следующем виде: допустимое решение не является глобальным оптимумом, если и только если оно может быть улучшено некоторым локальным образом. Ниже приводится несколько примеров таких задач, подробное описание которых можно найти в [3]. Данные примеры указывают на важность локального поиска при построении оптимизационных алгоритмов и достаточно общий характер этого подхода.

1. <u>Линейное программирование</u>. 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. Алгебраически, предполагая 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. Получающаяся таким образом окрестность является точной и имеет полиномиальную мощность.

2. <u>Минимальное остовное дерево</u>. 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 называют 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 трудоемкость 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 [5]. 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 мы увидим ниже, разумным оказывается не только такой выбор, но и движение в "абсурдном"направлении, когда несколько шагов с ухудшением 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, вообще говоря, приводит 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 включает в себя 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 максиальным значением $g(a, b)$.
Шаг 2. Временно поменять местами a и b.
Шаг 3. Повторять k раз шаги 1, 2, не трогая пары вершин, уже использованные на предыдущих шагах.
Полученная последовательность $\{(a^\tau, b^\tau)\}_{\tau \leq k}$ определяет k соседних решений для раз-

биения (V_1, V_2). Изменение целевой функции для соседа τ определяется равенством

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

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 3. Графом окрестностей $G_N = (\mathcal{I}, E)$ будем называть взвешенный ориентированный граф, вершинами которого являются допустимые решения задачи, а дугами — упорядоченные пары (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 еще ландшафтом целевой функции или просто ландшафтом (*landscape, fitness landscape* [74, 97]). При определении функции окрестности N важно следить за тем, чтобы получающийся граф G_N был *строго связен*, 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 вершину $x^* \in \mathcal{I}^*$ [18, 68]. Если же и этого свойства нет, то мы теряемувереность в достижении глобального оптимума локальными методами и должны либо ограничиться локальными оптимумами, либо переопределить функцию окрестности.

Анализ вычислительной сложности 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 трудоемкостью $O(n \log n)$, n — число вершин, которые дают всего несколько процентов погрешности [59].

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, 34]. На случайно сгенерированных задачах такой колоссальной 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 показывают 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. Исследование локального поиска с точки зрения гарантированных оценок качества показывают границы его возможностей. Построены трудные для локального поиска примеры, из которых следует, что [99]

1) минимальная точная окрестность может иметь экспоненциальную мощность;

2) число шагов для достижения локального оптимума может оказаться экспоненциальным;

3) значение локального оптимума может сколь угодно сильно отличаться от глобального оптимума.

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 4 [104]. 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 для него значение целевой функции.
3. Узнать, является ли данное допустимое решение x локальным оптимумом для l, и если нет, то найти в окрестности решение с лучшим значением целевой функции.

В классе PLS содержатся все задачи локального поиска, для которых проверка локальной оптимальности может быть осуществлена за полиномиальное время. В этом классе есть полиномиально разрешимые задачи, т.е. задачи, в которых за полиномиальное время можно найти локальный оптимум. Многие "невзвешенные"задачи относятся к таковым: задачи о клике, о покрытии, о независимом множестве с любой полиномиальной по мощности окрестности и др. Однако в общем случае вопрос остается открытым. Надежду на его положительное решение дает следующее утверждение.

Теорема 1 [104] *Если задача L из класса PLS является NP-трудной, то NP=co-NP.*

Для задач локального поиска определяется понятие PLS-сведения.

Определение 5. 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 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. для всех $l \in L_1$, если 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. Так определенное 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 вершин на две части. Целевая функция — суммарный вес ребер, соединяющих две части разбиения. Окрестность FLIP содержит в себе все решения, полученные из данного переносом 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 переменных или их отрицаний. Требуется найти значения булевых переменных, при которых суммарный вес выполненных конъюнкций был бы максимальным [67].
2. Заданы наборы, состоящие не более чем из трех булевых переменных без отрицаний или констант 0, 1. Набор считается выполненным, если не все его значения совпадают. Каждому набору приписан неотрицательный вес. Требуется найти значение булевых переменных, при которых суммарный вес выполненных наборов был бы максимальным [94].

Задача о разбиении графа на две равные части с окрестностями Лина-Кернигхана [62], "1-замена"и их модификациями [94].

Приведенный список, конечно, не является исчерпывающим. Для указанных задач установлено, что в худшем случае стандартный алгоритм локального спуска требует экспоненциального числа шагов при любом выборе направления спуска. Таким образом, если локальный оптимум в этих задачах и может быть найден с полиномиальной трудоемкостью, то какими-то другими методами.

В завершение данного раздела приведем еще два результата, касающихся точной окрестности и возможности получить локальный оптимум с заданной гарантированной относительной погрешностью. Пусть оптимизационная задача P и функция окрестности N таковы, что соответствующая задача локального поиска лежит в классе PLS.

Теорема 2.[104] *Если задача P является NP-трудной (NP-трудной в сильном смысле), то окрестность N не может быть точной, если NP≠ co-NP (P≠NP).*

Теорема 3.[104] *Если для задачи P поиск приближенного решения с относительной погрешностью не более r является NP-трудной задачей (NP-трудной в сильном смысле), то существование полиномиального алгоритма нахождения локального оптимума с относительной погрешностью не более r влечет NP=co-NP (P=NP).*

2 Окрестности

Выбор окрестности, как уже отмечалось выше, играет важную роль при построении алгоритмов локального поиска. От него существенно зависит трудоемкость одного шага алгоритма, общее число шагов и, в конечном счете, качество получаемого локального оптимума. На сегодняшний день нет и, возможно никогда не будет, единого правила выбора окрестности. Для каждой задачи функцию N приходится определять заново, учитивая специфику данной задачи. Более того, по-видимому 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 возможные пути построения окрестностей и их свойства.

2.1 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 будет рассматриваться только неориентированные графы с симметричной целочисленной неотрицательной матрицей расстояний $w_{ij} = w_{ji}, 1 \le i, j \le n$. Каждое допустимое решение задачи коммивояжера или тур будем представлять в виде перестановки $\pi = \{i_1, \dots, i_n\}$. Для $\pi \in \mathcal{I}$ 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 [41]:

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

Оператор $\nabla^2 W(\pi)$ задает среднее отклонение целевой функции $W(\pi)$ в окрестности данной перестановки π. Для локального оптимума $\pi \in \widehat{\mathcal{I}}$ справедливо неравенство $\nabla^2 W(\pi) \ge 0$. Пусть W_{av} — средняя длина тура на множестве всех допустимых решений $\mathcal{I}$. Тогда справедливы следующие утверждения [41].

Теорема 4. *Функция $\widetilde{W} = W - W_{av}$ удовлетворяет уравнению*

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

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

Следствие 2. *Алгоритм локального поиска, начиная с произвольной перестановки, достигнет локального оптимума 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 $2n$-вершиного графа на две части по n вершин с минимальным суммарным весом ребер, соединяющих эти части;

(ii) о раскраске вершин графа в n цветов так, чтобы смежные вершины имели разные цвета;

(iii) о разбиении n-элементного множества на два подмножества так, чтобы суммарный вес одного подмножества совпал с суммарным весом другого подмножества;

(iv) о 3-выполнимости 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, существует ли назначение переменных, при котором все наборы будут выполнены.

2.2 Окрестности Лина-Кернигхана

Рассмотрим гамильтонов цикл как последовательность ребер. Окрестностью 2-*opt* будем называть множество всех гамильтоновых циклов, получающихся из заданного заменой двух несмежных ребер. Такая окрестность содержит $n(n-3)/2$ элементов, что несколько меньше, чем в окрестности *city-swap*. На рисунке 1 показано различие между окрестностями 2-*opt* и *city-swap*. Для окрестности 2-*opt* 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 окрестности *city-swap* 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.

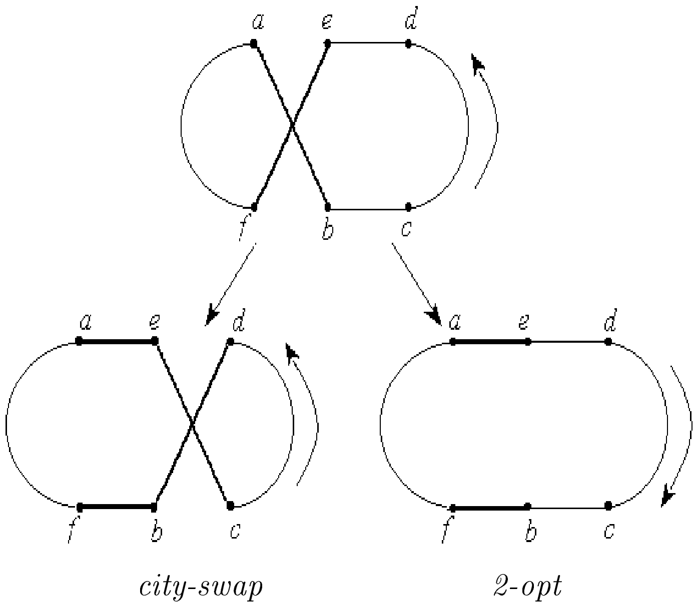

Рис. 1. Различия между окрестностями

В случае, когда точки 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-*opt* может быть естественным образом расширена до 3-*opt* и в общем случае до k-*opt* окрестности, где разрешается заменять не более k ребер данного тура на k новых ребер [70].

Теорема 5 [66]. *Поиск локального оптимума в задаче коммивояжера с окрестностью k-opt является PLS-полной для некоторого k.*

Минимальное значение константы k, при котором утверждение остается верным, пока неизвестно. Из работы [66] 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-*opt* Лином и Кернигханом предложена очень эффективная эвристика [71]. Она позволяет заменять произвольное число ребер и переходит от одного тура к другому, используя принципы жадных алгоритмов. Основная идея эвристики заключается в следующем. Удалим из гамильтонового цикла произвольное ребро, скажем (a, b). В полученном пути один конец (вершину a) будем считать фиксированной, а другой конец будем менять, перестраивая гамильтонов путь. Добавим ребро из вершины b, например (b, c), и разорвем образовавшийся единственный цикл так, чтобы снова получить гамильтонов путь. Для этого придется удалить ребро, инцидентное вершине c. Обозначим его (c, d). Новый гамильтонов путь имеет концевые вершины a и d (см. 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 состоит из удаления некоторого ребра, выполнении серии последовательных ротаций и, наконец, замыкания концевых вершин полученного гамильтонова пути. Существуют различные варианты этой основной схемы, которые отличаются правилами выбора ротаций и ограничениями на множества удаляемых и добавляемых ребер [60]. 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 трудоемкость одного шага (перехода от одного тура к другому) останется полиномиальной. Общее число шагов алгоритма, по-видимому, не может быть ограничено полиномом и известен вариант окрестности Лина-Кернигхана, для которого задача коммивояжера становится PLS-полной [104].

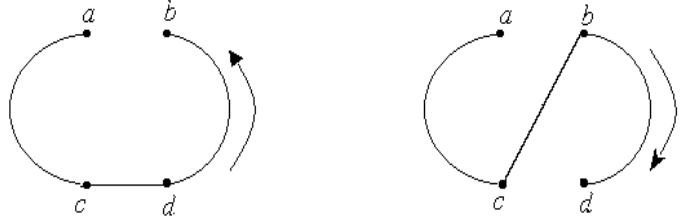

Рис. 2. Процедура ротации

2.3 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 работах [19, 8]. Мы не будем останавливаться на всех результатах в данной области. Изложим, в качестве иллюстрации, 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 [88] 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 $m \le k$. Для удобства изложения добавим в первую группу $k - m$ 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 $(\tau_1, \dots, \tau_k)$ — произвольная перестановка элементов $\{1, \dots, k\}$. Содержательно, каждый такой тур получается вставкой городов v_i между парами городов u_j, u_{j+1}. Множество $N(u_1, \dots, u_k, v_1, \dots, v_m)$ имеет экспоненциальную мощность $[\frac{n+1}{2}]!$ при $2m \le n \le 2m + 1$. Оптимальный тур в этом множестве может быть найден с помощью решения задачи о назначениях следующим образом. При $j = 1, \dots, k$ положим

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

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

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

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

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

$$z_{ij} \in \{0, 1\}, \ i, j = 1, \dots, k.$$

Оптимальное решение задачи дает наилучшее размещение городов $v_i, i = 1, \dots, m$, между городами $u_j, j = 1, \dots, k$, то есть 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 максимизации потока в сети [5] или построением совершенного паросочетания минимального веса для двудольного графа с долями $V = \{v_1, \ldots, v_k\}$ и $U = \{u_1, \ldots, u_k\}$.

Заметим, что максимальная мощность окрестности $N(u_1, \ldots, u_k, v_1, \ldots, v_m)$ достигается при $m < n/2$. Точнее, пусть $m = n/2 - p, p \geq 0$, целое, $n \geq 5$ и $\sigma(n) \equiv n \pmod 2$. Для действительного числа r обозначим через $[r]_0$ (соответственно $[r]_1$) максимальное целое (полуцелое, то есть число вида $q/2$ для нечетных q), не превосходящее r. Тогда максимальная мощность $N_{max}(n)$ окрестности превосходит $[\frac{n+1}{2}]!$ и равна [44]

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

где $p_0 = [\sqrt{\frac{1}{8}n + \frac{9}{8}} + \frac{3}{8}]_{\sigma(n)}$. Применение этой формулы позволяет получить асимптотическое выражение для мощности окрестности.

Теорема 6 [44]. $N_{max}(n) = \Theta(\frac{e^{\sqrt{n/2}}[\frac{n+1}{2}]!}{n^{1/4+[1/2]_{\sigma(n)}}})$.

Перейдем теперь к наиболее интригующему свойству данной окрестности, связанному к расстоянием между турами в графе окрестностей. Оказывается, что с помощью множества $N(u_1, \ldots, u_k, v_1, \ldots, v_m)$ 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, это константа очень мала; всего 4 шага достаточно сделать, чтобы добраться из заданного решения до любого другого и, в частности, до оптимального. К сожалению, мы не знаем в какую именно сторону нужно сделать эти четыре шага, иначе задача стала бы полиномиально разрешимой. Кроме того, мы не знаем, как много локальных оптимумов типа 2-*opt* или *city-swap* 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-вершином графе. Обозначим через $\mathcal{F}_{nm}$ семейство всех подмножеств мощности m, состоящих из несмежных вершин тура T. Каждый элемент семейства $(v_1, \ldots, v_m) \in \mathcal{F}_{nm}$ однозначно определяет множество туров $N(u_1, \ldots, u_{n-m}, v_1, \ldots, v_m)$. Объединение таких множеств обозначим через $N_m(T)$. Можно показать, что при фиксированном m мощность семейства $\mathcal{F}_{nm}$ равна

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

то есть 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 $(v_1, \ldots, v_m)$ таких, что $R \in N(u_1, \ldots, u_{n-m}, v_1, \ldots, v_m)$.

Теорема 7 [45]. *При $m = \lfloor (n-1)/2 \rfloor$ для любых туров T_1 и T_5 существуют такие туры T_2, T_3 и T_4, что T_i является соседним к $T_{i-1}, i = 2, 3, 4, 5$.*

В общем случае при произвольных m длина пути в графе окрестностей G_N между двумя произвольными турами ограничена сверху величиной $4\lceil \lfloor (n-1)/2 \rfloor / m \rceil$.

3 Поиск с чередующимися окрестностями

Идеи локального поиска получили свое дальнейщее развитие в так называемых метаэвристиках [84]. Ниже пойдет речь об одной из них, получившей название поиск с чередующимися окрестностями (variable neighborhoods search) [77, 49, 52]. Её основная идея состоит в систематическом изменении функции окрестности N и соответствующем изменении ландшафта в ходе локального поиска. Существует много вариантов реализации этой идеи особенно для задач большой размерности. Легкость реализации для различных моделей, гибкость и высокая эффективность объясняют ее конкурентноспособность при решении NP-трудных задач, в частности, задач о p-медиане [26, 92, 50, 72], задачи Вебера [16], задач кластеризации и размещения [17, 53, 54, 78], коммивояжера [77, 83, 96, 51, 20, 35, 36, 92], построения маршрутов для транспортных средств [10, 13, 58, 93, 27, 37, 56, 76] и расписаний [9, 21, 22, 29, 30, 65, 48], построения сетей [14, 25], поиска линейных порядков [43], деревьев Штейнера [75, 91, 101], максимальных разрезов [33], клик максимальной мощности [55], деревьев с заданным числом ребер [80] и многих других.

3.1 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 положения:

- *Локальный минимум относительно одной окрестности не обязательно будет 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 [12]. Это экспериментально проверенное, но формально пока не доказанное утверждение позволяет сузить область поиска глобального оптимума, используя информацию об уже обнаруженных локальных оптимумах. Именно эта гипотеза лежит в основе алгоритмов связывающих путей (Path Relinking) [40], различных операторов скрещивания (crossover) для генетических алгоритмов [42] и многих других подходов.

Идея чередующихся окрестностей может быть реализована 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 представлена 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 $f(x') = \min\{f(y)|y \in N_k(x)\}$;
(b) Если $f(x') < f(x)$, то полагаем $x \leftarrow x', k \leftarrow 1$ иначе $k \leftarrow k+1$.

Наряду с указанным выше последовательным порядком смены окрестностей можно предложить и групповую стратегию. Например, для трех окрестностей будем применять данный алгоритм с $k_{max} = 2$ в каждой точке 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 $N_k, k = 1, \ldots, k_{max}$ 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 полагаем $x \leftarrow x', k \leftarrow 1$ иначе $k \leftarrow k+1$.

Алгоритм RVNS эффективен 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 улучшения рекорда (шаг 2.2.(b)) как правило используется в качестве критерия остановки. По сравнению с методами Монте–Карло данный алгоритм отличается большей систематичностью и эффективностью. Например, в [79] алгоритм RVNS дает на 90% лучше решение задачи, чем метод Монте–Карло. Для задачи о p–медиане алгоритм RVNS дает решение того же качества, что и быстрые эвристики замены, но в 20–40 раз быстрее [54].

(iii) Основная схема локального поиска с чередующимися окрестностями (VNS) является комбинацией двух предыдущих вариантов [77].

Алгоритм VNS
Шаг 1. Выбрать окрестности $N_k, k = 1, \ldots, k_{max}$ 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) Применить локальный спуск к точке x', не меняя координат, по которым x и x' совпадают. Полученный локальный оптимум обозначим x'';
(c) Если $f(x'') < f(x)$, то полагаем $x \leftarrow x'', k \leftarrow 1$ иначе $k \leftarrow k+1$.

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

3.2 Модификации основной схемы

Существует много вариантов предложенной выше основной схемы. Например, на шаге 2.2.(c) переход в следующую точку осуществляется только при уменьшении значения целевой функции. Это условие можно заменить по аналогии с алгоритмом имитации отжига [6, 7] и разрешать движение в точку с большим 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, где выполняется переход согласно k-й окрестности, выбрать наилучший переход по всем рассматриваемым окрестностям. Другой способ состоит в модификации шага 2.2.(a) и выборе не случайной точки x', а наилучшей в k-й окрестности. Порядок смены окрестностей тоже может варьироваться.

Одной из серьезных проблем для алгоритмов локального поиска, и для VNS в частности, являются задачи большой размерности. В этом случае трудоемкость выполнения одной итерации становится неоправданно высокой и требуются новые идеи для повышения эффективности данного класса методов. Ниже предлагаются три возможных способа решения данной проблемы.

(iv) Алгоритмы VNS с декомпозицией на стадии локального поиска (VNDS) подразумевают модификацию шага 2.2.(b) 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 может рассматриваться как один из способов встраивания классических приближенных методов, которые используются в комбинаторной оптимизации с 60-х годов прошлого столетия, в основную схему VNS. Примеры успешного применения такого подхода можно найти в [93, 98].

(v) Алгоритм VNS с ориентацией на исследование удаленных частей допустимой области (SVNS). Эта модификация основной схемы особенна 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, при которой на шаге 2.2.(c) 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. Значения параметра α не должны быть слишком большими, чтобы переход в новую точку был действительно "разумным"и не слишком малым, чтобы реально влиять на поиск оптимального решения. По сути, данный параметр является своего рода нормировкой, выравнивающей значения целевой функции и расстояние. 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 [106, 32, 116]: 1) параллельный локальный поиск; 2) наращивание числа решений, выбираемых в текущей окрестности и выполнение локального поиска для каждого из них; 3) аналогично 2, но с обменом информации о наилучшем найденном решении. Вторая версия распараллеливания давала наилучшие результаты. В [27] рассматривалась другая схема, 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 рекордные значения для нескольких тестов большой размерности из электронной библиотеки TSP–LIB [90].

3.3 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 [38, 39] имеют громадное число приложений [40]. 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 наоборот. К настоящему времени опробованы одна схема первого типа и две схемы второго типа. В [92] для решения задачи о p–медиане с обходами клиентов 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. Рассмотрено четыре варианта локального спуска, в ходе которых собирается информация 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 так как для избежания зацикливания достаточно контролировать поиск только по одной из окрестностей. В [16] для решения задачи Вебера предлагается несколько эвристик, основанных на групповом варианте VND. Относительно одной из окрестностей (relocation) часть элементов могут рассматриваться как запрещенные, в то время как две другие окрестности (realocation, altrnate) 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 [32] состоит в рандомизации жадных алгоритмов с последующим улучшением полученного решения алгоритмами локального поиска. Наиболее естественный путь для гибридизации VNS и GRASP состоит замене последней стадии локального спуска на более гибкую схему VNS. Такой подход хорошо показал себя при поиске деревьев Штейнера [75, 91], в задаче о покупателе, которому нужно приобрести заданный перечень товаров на известных рынках [83], в задаче о максимальном разрезе [33] и др. Проведенные исследования показывают, что сочетание GRASP и VNS приводит к лучшим результатам, чем просто GRASP или VNS в отдельности.

(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. Методы локального поиска оказываются эффективными в таких условиях [86, 87, 93]. Для задач составления маршрутов с временными окнами предложены два оператора перестройки решения, которые использовались для локального спуска (VND) и локального поиска по окрестностям большой мощности (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 оптимальным образом при решении вспомогательной задачи небольшой размерности. В [10] эта идея воплощается более систематическим образом: сначала удаляются две компоненты, затем три и т.д. до тех пор пока процедура восстановления решения приводит к падению целевой функции. Другой пример гибридизации VNS и CP можно найти в [73].

3.4 Асимптотическая сходимость

Свойство асимптотической сходимости исследовалось для различных метаэвристик. Алгоритм имитации отжига был, по-видимому, первым из них, для которого это свойство было установлено [7]. Центральным понятием при изучении асимптотического поведения алгоритмов является цепь Маркова [2], которая описывает поведение алгоритма на некотором множестве состояний. Для алгоритма имитации отжига этим множеством состояний является множество допустимых решений задачи $\mathcal{I}$. Для алгоритма поиска с запретами это множество имеет более сложную структуру и состоит из пар (x, φ), где $x \in \mathcal{I}$, а φ — список запретов [64]. 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 гарантирует им асимптотическую сходимость.

Определение 7 [15] Систему окрестностей $N_k, k = 1, \dots k_{max}$ будем называть остовной, если $\bigcup_{k=1}^{k_{max}} N_k(x) = \mathcal{I}$ для любого $x \in \mathcal{I}$.

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

Свойство VNS [15] *Если система окрестностей является остовной, то с ростом числа итераций алгоритма VNS вероятность найти точное решение 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 не представляется возможным. 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 о небесплатных завтраках [95, 103]. Предположим, что некоторый алгоритм A превосходит по точности другой алгоритм B. Например, алгоритм локального подъема превосходит алгоритм локального спуска для задач на максимум. Теорема утверждает, что множество целевых функций для которых это верно, по мощности равно множеству функций для которых верно обратное. Другими словами, в среднем все алгоритмы одинаковы и равноценны, например, алгоритму случайного блуждания или выбору случайной точки в $\mathcal{I}$. Понятно, что для таких алгоритмов время поиска оптимального решения экспоненциально велико и оценки скорости сходимости не представляют практического интереса.

Приведем точную формулировку теоремы. Пусть множество $\mathcal{I}$ конечно и целевая функция f задает отображение $\mathcal{I}$ в конечное множество $\mathcal{J}$. 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 $x^{t+1} \in \mathcal{I}$, $x^{t+1} \neq x^\tau$, $\tau = 1, \dots, t$. Алгоритмы отличаются друг от друга способом выбора новой точки, но все они выполняют ровно T итераций. Конкретный алгоритм A для целевой функции f получит набор значений $\{f(x_1), \dots, f(x_m)\}$, по которому 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 равным $f_i \in \mathcal{J}$. 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 получить диаграмму $\vec{c}$ для функции f за T шагов алгоритма A.

Теорема 8 [103] *Для любой пары алгоритмов A, B справедливо*

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

Из теоремы непосредственно следует, что какую бы функцию $\Phi(\vec{c})$ мы ни брали (минимум или максимум), в среднем по всем f величина $P(\Phi(\vec{c})\,|\,f, T, A)$ не зависит от A. Таким образом, чтобы получить оценки сходимости итерационных алгоритмов необходимо делать какие-то дополнительные предположения о характере целевой функции, либо выделять специальные классы исходных данных.

4 Поиск дискретных структур

В этом разделе будут приведены примеры удачного применения методов локального поиска в теории графов. Оба примера используют один и тот же прием сведéния задачи поиска объекта с заданными свойствами к задаче дискретной оптимизации. На содержательном уровне этот прием состоит в задании желаемых свойств в виде ограничений на равенства или неравенства, а 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 остановки, и если нужный объект существует, то рано или поздно он будет обнаружен. Если же объект не существует, то алгоритм может работать бесконечно долго. Вероятностные алгоритмы, которые дают точное решение, но время работы которых является случайной величиной, называются алгоритмами типа Лас-Вегас [81]. В отличие от них алгоритмы типа Монте-Карло могут никогда не дать точное решение, но их погрешность стремится к нулю с ростом числа итераций. Ниже будут приведены примеры применения алгоритмов типа Лас-Вегас.

4.1 Палиндромные деревья

Пусть G — связный граф и D — его диаметр. Обозначим через $d(G,k)$ число пар вершин в G, находящихся на расстоянии k друг от друга. 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 [23, 57, 69].

Определение 7 [23]. Граф G с диаметром D называется H-палиндромным, если равенство $d(G,k) = d(G,D-k)$ выполняется при всех $k = 0,1,\ldots,\lfloor D/2 \rfloor$.

Определение 8 [23]. Граф G с диаметром D называется H^*-палиндромным, если равенство $d(G,k) = d(G,D-k+1)$ выполняется при всех $k = 1,\ldots,\lceil D/2 \rceil$.

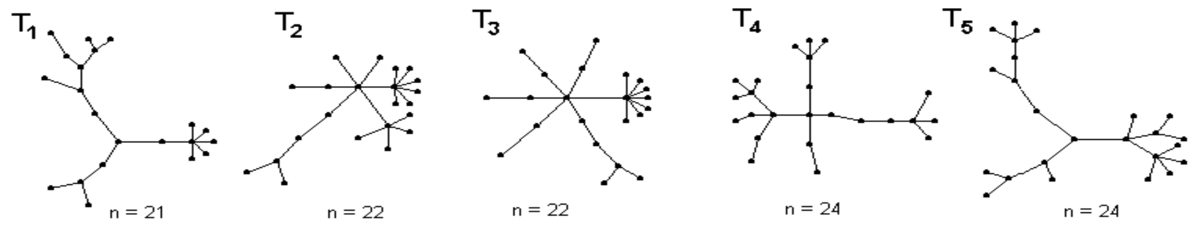

Рис. 3. H-палиндромные деревья

В 1993 году И. Гутман сделал предположение, что H-палиндромные деревья не существуют [46]. Позднее было найдено три примера H^*-палиндромных деревьев с $n \leq 12$ вершинами [47] и семейство H-палиндромных графов, не являющихся деревьями [1]. 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 [49, 51]. Диаметры деревьев равны 6 или 8, а полиномы Хосойа имеют вид:

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

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 1 приведены результаты поиска H- и H^*-палиндромных деревьев с $n \leq 26$ вершинами. Так как методами локального поиска нельзя доказать несуществование объектов, то полным перебором было проверено, что других H- и H^*-палиндромных деревьев с $n \leq 26$ вершинами нет.

Таблица 1. 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

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 [23].

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 [24], 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[24].

Обозначим через $L = (l_{pq})$ матрицу расстояний для пар вершин некоторого графа $G = (V, L)$. 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

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

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

Система AutoGraphiX использовалась для минимизации разности $Ra(G) - \bar{l}(G)$ по всем связным графам при $n = 2, 3, \ldots, 20$. 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 значения разности всегда оказывалось меньше. Кроме того, указанная разность быстро росла с ростом n. Для звезд можно проверить, что

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

Таким образом, приходим к новому более сильному предположению.

Гипотеза $2'$ *Для всех связных графов*

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

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

Другие примеры применения системы AutoGraphiX можно найти в [24].

Список литературы

[1] Добрынин А. А. Построение графов с палиндромным полиномом Винера // Вычислительные системы. Сб. науч. тр. Новосибирск: Ин-т математики СО РАН, 1994. Вып. 151. С. 37–54.

[2] Кемени Дж., Снелл Дж. Конечные цепи Маркова. М.: Наука, 1970.

[3] Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. М.: Мир, 1985.

[4] Растригин Л. А. Случайный поиск — специфика, этапы истории и предрассудки // Вопросы кибернетики. 1978. Вып. 33, С. 3–16.

[5] Форд Л., Фалкерсон Д. Потоки в сетях. М.: Мир, 1966.

[6] Aarts E. H. L., Korst J. H. M. Simulated annealing and Boltzmann machines. Chichester: Wiley, 1989.

[7] Aarts E. H. L., Korst J. H. M., Laarhoven van P. J. M. Simulated annealing // Aarts E. H. L. et al. (Eds) Local Search in Combinatorial Optimization. Chichester: Wiley, 1997. P. 91–120.

[8] Balas E., Simonetti N. Linear time dynamic programming algorithms for some new classes of restricted TSP's // Proc. IPCO V (Lecture Notes in Computer Science, V. 1084). Berlin: Springer, 1996. P. 316–329.

[9] Basten den M., Stutzle T. Neighborhoods revisited: Investigation into the effictiveness of variable neighborhood descent for scheduling // MIC-2001, Porto, July 16–21, P. 545–549.

[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 Computer Science, Brown University, 2001.

[11] 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.

[12] 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.

[13] Braysy O. Local search and variable neighborhood search algorithm for the vehicle routing with time windows // Acta Wasaensia, 2001, V.87, Universitas Wasaenis, Vassa.

[14] 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 (to appear in Oper.Res).

[15] Brimberg J., Hansen P., Mladenović N. Convergence of veriable neighborhood search

[16] Brimberg J., Hansen P., Mladenović N., Taillard E. Improvements and comparison of heuristics for solving the multisource Weber problem // Oper. Res., 2000, V.48(3), P. 444–460.

[17] Brimberg J., Mladenović N. A variable neighborhood algorithm for solving the continuous location-allocation problem // Stud. in Locat. Analysis, 1996, V. 10, P. 1–12.

[18] Brucher P., Hurink J., Werner F. Improving local search heuristics for some scheduling problems. Part II // Discrete Appl. Math. 1997. V. 72, P. 47–69.

[19] Burkard R. E., Deineko V.G., Woeginger G. J. The traveling salesman problem and the PQ-tree // Proc. IPCO V (Lecture Notes in Computer Science, V. 1084). Berlin: Springer, 1996. P. 490–504.

[20] Burke E.K., Cowling P, Keuthen R. Effective local and guided variable neighborhood search methods for the asymmetric traveling salesman problem // Proceedings of the Evo Workshops, Springer, Lecture Notes in Computer Science, 1999, P. 203–212.

[21] Burke E., Causmaecker de P., Petrovic S., Berghe G. V. Variable neighborhood search for nurse rostering problem // MIC-2001, Porto, July 16–21, P. 755–760.

[22] Campos de L. M., Puerta J. M. Stochastic local search algorithm for linear belief networks: searching in the space of orderings // ESCQARU 2001, Lecture Notes in AI 2143, P. 228–239.

[23] Caporossi G., Dobrynin A. A., Gutman I., Hansen P. Trees with palindromic Hosoya polynomials // Graph Theory Notes N.Y. V. 37, 1999. P. 10–16.

[24] Caporossi G., Hansen P. Variable neighborhood search for extremal graphs: 1 The AutoGraphiX system // Discrete Math. 2000. V. 212, N 1–2. P. 29–44.

[25] Costa M-C., Monclar F-R., Zrikem M. Variable neighborhood search of cable layout problem // MIC-2001, Porto, July 16–21, P. 749–753.

[26] Crainic T., Gendreau M., Hansen P., Hoeb N, Mladenović N. Parallel variable neighborhood search for the p-median // MIC-2001, Porto, July 16–21, P.595–599.

[27] Crispim J., Brandao J. Reactive tabu search and variable neighborhood descent to the vehicle routing problem with backhauls // MIC-2001, Porto, July 16–21, P. 631–636.

[28] Croes G. A. A method for solving traveling salesman problems // Oper. Res. 1958. V. 6, P. 791–812.

[29] Davidovic T., Hansen P., Mladenović N. Variable neighborhood search for multiprocessor scheduling with communication delays // MIC-2001, Porto, July 16–21, P. 737–741.

[30] Desrosiers J., Mladenović N., Villeneuve D. Disign of balansed MBA student teams // MIC-2001, Porto, July 16–21, P. 281–285.

[31] Fajtlowicz S. On conjectures of Graffiti // Discrete Mathematics, 1988, V.72, P.113–118.

[32] Feo T., Resende M. Greedy randomized adaptive search // J. Global Optim., 1995, V. 6, P. 109–133.

[33] Festa P., Pardalos P., Resende M. Riberio C. GRASP and VNS for Max-cut // MIC-2001, Porto, July 16–21, P. 371–376.

[34] Fridman M. L., Johnson D. S., McGeoch L. A., Ostheimer G. Data structures for traveling salesman // J. Algorithms. 1995. V. 18, P. 432–479.

[35] Gendreau M., Hertz A., Laporte G. New insertion and postoptimization procedures for the traveling selesman problem // Oper. Res., 1992, V. 40, P. 1086–1094.

[36] Gendreau M., Hertz A., Laporte G. The traveling selesman problem with back-hauls // Computers Oper. Res., 1996, V. 23, P. 501–508.

[37] Ghiani G., Hertz A., Laporte G. Recent algorithmic advances for arc routing problems // Les Cahiers du GERAD G-2000-40, Monreal, Canada, 2000.

[38] Glover F. Tabu search: part I // ORSA J. Comp. 1989. V. 1. P. 190–206.

[39] Glover F. Tabu search: part II // ORSA J. Comp. 1990. V. 2. P. 4–32.

[40] Glover F., Laguna. M Tabu search. Boston: Kluwer Acad. Publ. 1997.

[41] Grover L. K. Local search and the local structure of NP-complete problems // Oper. Res. Lett. 1992. V. 12, N. 4. P. 235–244.

[42] Goldberg D. E. Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley. 1989.

[43] Gonzales C. G., Brito D. P. A variable neighborhood search for solving linear ordering problem // MIC-2001, Porto, July 16–21, P. 181–185.

[44] Gutin G. Exponential neighborhood local search for the traveling salesman problem // Comput. Oper. Res. 1999. V. 26, P. 313–320.

[45] Gutin G., Yeo A. Small diameter neighborhood graphs for the traveling salesman problem: four moves from tour to tour // Comput. Oper. Res. 1999. V. 26, P. 321–327.

[46] Gutman I. Some properties of the Wiener polynomial // Graph Theory Notes N.Y. 1993. V. 25, P. 13–18.

[47] Gutman I., Estrada E., Ivanciuc O. Some properties of the Wiener polynomial of trees // Graph Theory Notes of New York, V.36, New York Akademy of Sciences, 1999. P. 7–13.

[48] 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.

[49] Hansen P., Mladenović N. Variable neighborhood search: principles and applications //(invited review) European J. Oprnl. Res. V. 130, N 3, P. 449–467.

[50] Hansen P., Mladenović N. Variable neighborhood search for the p-median // Location Sci. 1997. V. 5, P. 207–226.

[51] Hansen P., Mladenović N. An introduction to variable neighborhood search // S. Voss and al.(Eds.) Metaheuristics, Advances and Trends in Local Search Paradigms for Optimization, 1999, Kluwer, Dordrecht, P. 433–458.

[52] Hansen P., Mladenović N. Development of variable neighborhood search // Ribeiro C. J. et al. (Eds.) Essays and Surveys in Metaheuristics, Kluwer Akademic Publishes, 2002, P. 415–440.

[53] Hansen P., Mladenović N. J-Means: A new local search heuristic for minimum sum-of-squares clustering // Patern Recognition, 2001, V. 34, P. 405–413.

[54] Hansen P., Mladenović N., Perez-Brito D. Variable neighborhood decomposition search // J. of Heuristics, 2001, V. 7(4), P. 335–350.

[55] Hansen P., Mladenović N. Urosevic D. Variable neighborhood search for maximum clique // Les Cahiers du GERAD G-2001-25, Monreal, Canada, 2001.

[56] Hertz A., Laporte G., Mittaz M. A taboo search heuristic for the capacitated arc routing problem // Oper. Res., 2000, V. 48, P. 129–135.

[57] Hosoya H. On some counting polynomials in chemistry // Discrete Appl. Math. 1988. V. 19, N 2. P. 239–257.

[58] Hwang F. K., Richards D. S., Winter P. The Steiner tree problem. North-Holland, Amsterdam,1992.

[59] 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.

[60] Johnson D. S., McGeoch L. A. The traveling salesman problem: a case study // Aarts E. H. L. et al. (Eds.) Local Search in Combinatorial Optimization. Chichester: Wiley, P. 215–310.

[61] Johnson D. S. Local optimization and the traveling salesman problem // Automata, Languages, and Programming. Berlin: Springer Verlag, 1990. (Lecture Notes in Computer Science, V. 443) P. 446–461.

[62] Johnson D. S., Papadimitriou S. H., Yannakakis M. How easy is local search? // Journal of Computer and System Sciences, 1988. V. 37, P.79–100.

[63] Kernighan B. W., Lin S. An efficient heuristic procedure for partitioning graphs // Bell System Technical Journal. 1970. V. 49, P. 291–307.

[64] Kochetov Yu. A., Goncharov E. N., Behavior of a probabilistic tabu search algorithm for the multi stage uncapacitated facility location problem // Proc. INFORMS-KORMS, Seul 2000.

[65] Kolish R., Hartman S. Heuristic algorithms for the resource-costrained project scheduling problem: classification and computational analysis // Weglarz J. (ed.) Project Scheduling: Recent Models, Algorithms and Applications, Kluwer, 1999, P. 147–178.

[66] Krentel M. W. Structure in locally optimal solutions // 30th Annual Symposiun on Foundation of Computer Science. 1989, P. 216–222.

[67] Krentel M. W. On finding and verifying locally optimal solutions // SIAM Journal on Computing, 1990. V. 19, P. 742–751.

[68] Laarhoven van P. J. M., Aarts E. H. L., Lenstra J. K. Job shop scheduling by simulated annealing // Oper. Res. 1992. V. 40, P. 113–125.

[69] Lepović M., Gutman I. A collective property of trees and chemical trees // J. Chem. Inform. Comput. Sci. 1998. V. 38, P. 823–826.

[70] Lin S. Computer solutions of the traveling salesman problem // Bell Sys. Tech. J. 1965. V. 44, P. 2245–2269.

[71] Lin S., Kernighan B. W. An efficient heuristic algorithm for the traveling-salesman problem // Oper. Res. 1973. V. 21, P. 498–516.

[72] Lopez F. G., Batista B. M., Moreno-Perez J., Moreno-Vega M. The parallel variable neighborhood search for the p-median problem // Research Repot, University of La Laguna, Spain, 2000.

[73] Loudin S., Boizumault P. VNS/LDS+CP: A hybrid method for constraint optimization in anytime contexts // MIC-2001, Porto, July 16–21, P. 761–765.

[74] Macken C. A., Hagan P. S., Perelson A. S. Evolutionary walks on rugged landscapes // SIAM J. Appl. Math. 1991. V. 51, N 3. P. 799–828.

[75] 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. of Global Optimization, 2000, V. 17, P. 267–283.

[76] Mittaz M. Problemes de cheminements optimaux dans des reseaux avec contraintes associess aux arrcs // Ph.D. Thethis, Department of Mathematics, Ecole Polytechnique Federale de Lausanne, Switzerland.

[77] Mladenović N., Hansen P. Variable neighborhood search // Computers Oper. Res., 1997, V. 24, P. 1097–1100.

[78] Mladenović N., Moreno J. P., Moreno-Vega A. A chain-interchange heuristic method // Yugoslav J. Oper.Res., 1996, V. 6, P. 41–54.

[79] Mladenović N., Petrović J., Kovačević-Vujčić V. and Čangalović M. Solving spread spectrum radar polyphase cade design problem by tabu search and variable neighborhood search // SMG Report R-00-09, Universite Libre Bruxelles, Belgium, 2000 (to appear in European J. Oper. Res).

[80] Mladenović N., Urosevic D. Variable neighborhood search for the k-cardinality tree // MIC-2001, Porto, July 16–21, P. 749–753.

[81] Motwani R., Raghavan P. Randomized algorithms. Cambridge: Cambridge Univ. Press, 1995.

[82] 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.

[83] Ochi L. S., Silva M. V., Drummond L. Metaheuristics based GRASP and VNS for solvimg traveling salesman problem // MIC-2001, Porto, July 16–21, P. 489–494.

[84] Osman I. H., Laporte G. Metaheuristics: a bibliography // Ann. Oper. Res. 1996. V. 63, P. 513–628.

[85] Page E. S. On Monte Carlo methods in congestion problems: I, searching for an optimum in discrete situations // Oper. Res. 1965. V. 13, P. 291–299.

[86] Pesant G., Gendreau M. A view of local search in constraint programming, principles and practice of constraint programming // Lecture Notes in Computer Science, 1996, V. 1118, P. 353–366,

[87] Pesant G., Gendreau M. A constraint programming framework for local search methods // J. of Heuristics, 1999, V. 5, P. 255–279.

[88] Punnen A. P. The traveling salesman problem: new polynomial approximation algorithms and domination analysis // Research Report. St.John: University New Brunswick. 1996.

[89] Reiter S., Sherman G. Discrete optimizing // J. Soc. Indust. Appl. Math. 1965. V. 13, P. 864–889.

[90] Resende M. G. S., Pitsoulis L. S., Pardalos P. M. Approximate solution of weighted max-sat problem using GRASP // Dingzhu D. et al. (Eds.) Satisfiability Problem: Theory and Applications, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, AMS, Providence, Phode Island, 1997.

[91] Ribeiro C., Uchoa E., Werneck R. A hybrid GRASP with perturbations for the Steiner problem in graphs. Technical report, Computer Science Department, Catolic University of Rio de Janeiro, 2001.

[92] Rodriggue I., Moreno-Vega M., Moreno-Perez J. Heuristics for routing-median problems // SMG Report, Universite Libre de Bruxelles, Belgium,1999.

[93] 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.

[94] Schaffer A. A., Yannakakis M. Simple local search problems that are hard to solve. SIAM Journal on Computing, 1991. V. 20, P. 56–87.

[95] Schumacher C. Black box search — framework and methods. Ph.D. Thesis, 2000, The University of Tennessee, Knoxville, USA.

[96] Silva M.B., Drummond L., Ochi L. S. Variable neighborhood search for the traveling purchaser problem // 27th Int. Conf. on Comp. Indust. Engineering, 2000.

[97] Stadler P. F. Corelation 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. // Ribeiro C., Hansen P. (Eds.) Essays and Surveys in Metaheuristics, Kluwer Academic Publishers, 2001, P. 613–630.

[99] Tovey C. A. Local improvement on discrete structures // Aarts E. H. L. et al. (Eds.) Local Search in Combinatorial Optimization. 1997. Chichester: Wiley, 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 // Rayward-Smith V. J. et al. (Eds.) Modern Search Methods, John Wiley and Sons, 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 research laboratories, Eindhoven, 1995.

[103] Wolpert D. H., Macready W. J. No free lunch theorem for search // Report SFI-TR-95-02-010, The Santa Fe Institute, Santa Fe, 1995.

[104] Yannakakis M. Computational complexity // Aarts E. H. L. et al. (Eds.) Local Search in Combinatorial Optimization. 1997. Chichester: Wiley, P. 19–55.