ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Июль—декабрь 2005. Серия 2. Том 12, № 2, 44–71

УДК 519.85

О СЛОЖНОСТИ ЛОКАЛЬНОГО ПОИСКА В ЗАДАЧЕ О p-МЕДИАНЕ*)

Ю. А. Кочетов, М. Г. Пащенко, А. В. Плясунов

Исследуется сложность решения задачи поиска локального минимума для задачи о p-медиане с полиномиально проверяемыми окрестностями. Приводится достаточное условие, при выполнении которого задача поиска локального минимума с полиномиально проверяемой окрестностью является PLS-полной. Показана PSPACE-полнота задачи нахождения локального минимума с рядом полиномиально проверяемых окрестностей при фиксированной начальной точке. Установлено, что с каждой такой окрестностью алгоритм локального спуска в худшем случае требует экспоненциального числа шагов для нахождения локального минимума при любом выборе направления спуска.

Предложен пример исходных данных задачи о p-медиане, когда алгоритм наискорейшего спуска с некоторыми полиномиально проверяемыми окрестностями требует экспоненциального числа шагов. Доказано, что если P≠NP, то локальный минимум относительно полиномиально проверяемой окрестности может оказаться больше глобального минимума в произвольное число раз. Установлено, что существование точной полиномиально проверяемой окрестности эквивалентно совпадению классов P и NP.

Введение

Отсутствие точных эффективных алгоритмов решения NP-трудных задач привело к появлению большого числа приближённых методов решения [6]. Многие из них используют локальный поиск. В общем виде под локальным поиском понимается процесс последовательного движения от текущего решения к соседнему решению с лучшим значением целевой функции. Процесс завершается при получении локального оптимума относительно заданной окрестности. При уточнении способа выбора соседнего решения возникает конкретный алгоритм локального

*)Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект 03–01–00455).

спуска. Задачу нахождения локального оптимума для заданной окрестности будем называть задачей локального поиска. В дальнейшем будем интересоваться сложностью решения этой задачи не только в классе алгоритмов локального поиска, но и в классе всех возможных алгоритмов её решения.

Примером алгоритма локального поиска является симплекс–метод для решения задач линейного программирования. В этом случае решениями являются базисные допустимые решения, которые соответствуют вершинам многогранного множества. Окрестность текущего решения состоит из всех вершин, в которые можно попасть, двигаясь по ребру, инцидентному данной вершине. Вопрос о локальной оптимальности решения сводится к анализу знаков оценок замещения. При уточнении способа выбора соседнего решения возникает тот или иной вариант симплекс–метода [2, 4].

При изучении алгоритмов локального поиска будем интересоваться уклонением получаемых локальных оптимумов от глобального оптимума и сложностью их нахождения. Далее, говоря о сложности, будем иметь в виду вычислительную сложность в худшем случае. Современная теория сложности в значительной степени базируется на классах задач распознавания P и NP [1, 5, 6, 17]. С каждой оптимизационной задачей можно естественным образом связать соответствующую задачу распознавания, которая сводится к исходной задаче: для данного входа найти такое решение, что значение целевой функции не превосходит заданной величины, если рассматривается задача на минимум. Алгоритм решения оптимизационной задачи можно использовать для решения данной задачи распознавания. Поэтому классов P и NP обычно достаточно для изучения сложности оптимизационных задач, когда нас интересуют различные вопросы, связанные с нахождением оптимальных и приближённых решений. Для задач локального поиска таких естественных сводимостей, связывающих их с задачами распознавания, не известно. Рассмотрим, например, следующую задачу распознавания: существует ли локальный оптимум, на котором значение целевой функции не превосходит заданной величины? Неизвестно, как использовать алгоритм решения задачи локального поиска для того, чтобы найти ответ на эту задачу распознавания. По сути рассматриваемая задача распознавания представляется более сложной, чем задача нахождения какого-нибудь локального оптимума. Если эта или подобные задачи распознавания являются NP-полными, то это не означает, что задача локального поиска является трудноразрешимой. По этой причине для изучения сложности соответ-

ствующих задач локального поиска вводится класс PLS (сокращение от polynomial-time local search problems) [21]. С содержательной точки зрения этот класс содержит те задачи, в которых локальная оптимальность может быть проверена за полиномиальное время: для заданного решения требуется определить, является ли оно локальным оптимумом, и если нет, то найти соседнее решение с лучшим значением целевой функции. В этом случае соответствующую окрестность называют полиномиально проверяемой. Если предположить, что NP $\neq$ co-NP, то в классе PLS нет NP-трудных задач [18, 21]. Другими словами, не существует NP-полных задач, которые за полиномиальное время сводились бы к какой-нибудь задаче локального поиска из класса PLS. Следовательно, сложность задач из этого класса меньше сложности NP-полных задач [17]. Отметим, что гипотеза NP $\neq$ co-NP является более сильной, чем предположение о том, что P $\neq$ NP, так как совпадение последних двух классов влечёт совпадение первых двух [1].

В настоящей статье исследуется сложность нахождения локального оптимума в задаче о p-медиане: для заданной окрестности и произвольного входа требуется найти какой-нибудь локальный оптимум. Изучение этого вопроса не ограничивается алгоритмами локального поиска, а проводится в классе всех возможных алгоритмов. Это различие существенно. Например, для задачи линейного программирования для большинства известных вариантов симплекс-метода получены экспоненциальные нижние оценки для числа итераций, хотя сама задача полиномиально разрешима [2, 4]. Несмотря на это различие, интересно оценить временну́ю сложность алгоритма локального спуска. Нам удалось показать, что в задаче о p-медиане для целого ряда окрестностей при любом выборе направления спуска число шагов алгоритма не может быть ограничено сверху полиномом от длины записи исходных данных.

Для таких задач как клика, вершинное покрытие и других алгоритм локального спуска за полиномиальное время позволяет найти локальный минимум, начиная с произвольного допустимого решения. Таким образом, сложность нахождения локального оптимума для этих задач оказывается полиномиальной уже в классе алгоритмов локального спуска. Заметим, что начальное решение выбиралось произвольным образом, т. е. за полиномиальное время можно найти локальный оптимум, начиная с наперёд заданной начальной точки. Эта задача несколько отличается от предыдущей, когда требуется найти какой-нибудь локальный оптимум. Фиксация начальной точки может усложнить задачу. Теперь она формулируется следующим образом: для заданных окрестности, входа

и начального решения найти локальный оптимум, который может быть получен алгоритмом локального спуска из данного начального решения. Ряд таких задач относится к классу PSPACE-полных задач [21]. В статье приводятся окрестности, для которых задача о p-медиане, как задача локального поиска с фиксированной начальной точкой, также оказывается в этом классе.

Статья организована следующим образом. В § 1 вводятся основные определения и, в частности, определения класса PLS и его сводимости. В § 2 рассматриваются различные окрестности, вводится понятие силы окрестности и устанавливаются их элементарные свойства. В § 3 доказывается PLS полнота задачи о p-медиане с каждой из введенных окрестностей. Это означает, что существование полиномиального алгоритма для нахождения локального минимума хотя бы для одной из этих окрестностей приводит к полиномиальной разрешимости всех задач из класса PLS.

Заметим, что до сих пор неизвестно ни одного полиномиального алгоритма для решения PLS-полных задач, хотя скорее всего они не настолько сложны, чтобы среди них были NP-трудные задачи. В § 4 приводятся анализ сложности алгоритма локального спуска и задачи отыскания локального оптимума с заданной начальной точкой. Приводится пример исходных данных, когда алгоритм наискорейшего спуска с некоторыми окрестностями требует экспоненциального числа шагов. В § 5 рассматриваются вопросы о качестве получаемых локальных оптимумов и существовании точных полиномиальных окрестностей, т. е. окрестностей, в которых локальные оптимумы являются глобальными. Показано, что локальный минимум относительно полиномиально проверяемой окрестности может быть больше глобального минимума в произвольное число раз. Существование точной полиномиально проверяемой окрестности влечёт равенство P=NP.

§ 1. Основные определения

Существует несколько определений оптимизационной задачи [1, 6, 12], которые отличаются степенью детализации. В данной статье используется определение из [6].

Определение 1. *Оптимизационная задача* OP определяется следующим набором объектов $< I, Sol, f, goal >$, где

1) I – множество входов задачи OP;

2) Sol – функция, которая каждому входу $x \in I$ сопоставляет множество допустимых решений $Sol(x)$;

3) f – функция, которая задаёт вес $f(s, x)$ допустимого решения s на входе x;

4) величина $goal \in \{\min, \max\}$ уточняет, является ли задача OP задачей на максимум или минимум.

В оптимизационной задаче необходимо найти оптимальное решение для заданного входа x.

В дальнейшем будем рассматривать только задачи на минимум, хотя все последующие определения можно перенести и для задач на максимум.

Определение 2. *Задача локального поиска* — это пара $L = (OP, N)$, где OP — оптимизационная задача, а N — функция, которая каждому допустимому решению s на входе x ставит в соответствие множество $N(s,x) \subseteq Sol(x)$ соседних решений. Множество $N(s,x)$ называют *окрестностью* решения s. Задача локального поиска заключается в отыскании локального минимума для заданного входа x.

Если s — локальный минимум в задаче L, т. е. в его окрестности нет решений с меньшим значением целевой функции, то будем говорить, что решение s является *N-оптимальным.* Пусть (OP, N_1) и (OP, N_2) — две задачи локального поиска. Будем говорить, что окрестность N_2 *сильнее* окрестности N_1 (окрестность N_1 *слабее* окрестности N_2), если из N_2-оптимальности решения следует его N_1-оптимальность. Будем записывать это следующим образом $N_1 \preceq N_2$. Введённое отношение является рефлексивным и транзитивным.

В дальнейшем будем рассматривать только такие задачи локального поиска, для каждой из которых существует такой полином p, что для любого входа x длина каждого допустимого решения $s \in Sol(x)$ ограничена сверху значением полинома p от длины записи исходных данных, т. е. $|s| \leqslant p(|x|)$, где через $|y|$ обозначается длина слова y.

Определение 3 [21]. Задача L принадлежит классу PLS, если существует три полиномиальных алгоритма A, B и C такие, что

1) алгоритм A определяет, является ли любое слово x входом задачи; если $x \in I$, то алгоритм находит допустимое решение задачи OP;

2) для любого входа $x \in I$ и любого слова s алгоритм B определяет, является ли s допустимым решением; если $s \in Sol(x)$, то за полиномиальное время алгоритм находит значение целевой функции $f(s,x)$;

3) для любого входа $x \in I$ и любого решения $s \in Sol(x)$ алгоритм C определяет, является ли s локальным оптимумом; если нет, то алгоритм находит соседа $s' \in N(s,x)$ с меньшим значением целевой функции.

Определение 4 [21]. 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) по произвольному входу x задачи L_1 функция h вычисляет некоторый вход $h(x)$ задачи L_2;

2) по произвольному решению s для входа $h(x)$ функция g находит некоторое решение $g(s, x)$ для входа x;

3) если $x \in L_1$ и s — локальный оптимум для входа $h(x) \in L_2$, то $g(s, x)$ — локальный оптимум для входа x.

Далее будем говорить о сводимости, опуская обозначение PLS. Задачу L из класса PLS называют PLS-полной [18, 21], если все задачи из этого класса сводятся к L. Полные задачи — это самые трудные задачи в данном классе. Из полиномиальной разрешимости любой из них следует полиномиальная разрешимость всех задач из этого класса. Первая PLS-полная задача содержится в [13]. По аналогии с классом полиномиально разрешимых оптимизационных задач [6] введём класс P_{PLS} полиномиально разрешимых задач локального поиска.

Определение 5. Задача $L = (OP, N)$ из PLS принадлежит классу P_{PLS}, если существует полиномиальный алгоритм, который по любому входу x задачи L находит N-оптимальное решение этой задачи.

Этот класс не пуст, так как в нём содержатся многие «невзвешенные» задачи с любой полиномиально проверяемой окрестностью, например, задачу о максимальной клике, задачу о покрытии минимальной мощности, задачу линейного программирования. Примеры таких задач из теории расписаний можно найти в [7, 8].

В [21] между классом PLS и классами P_S и NP_S была установлена следующая цепочка вложений $\mathrm{P}_S \subseteq \mathrm{PLS} \subseteq \mathrm{NP}_S$, где P_S и NP_S — формальные аналоги задач поиска, решаемых за полиномиальное время на детерминированных и недетерминированных машинах Тьюринга соответственно. Первое включение означает, что любая полиномиально разрешимая задача поиска представима в виде задачи из класса PLS. Второе включение говорит о том, что любая задача из класса PLS представима в виде задачи поиска, решаемой за полиномиальное время на недетерминированной машине Тьюринга. В [21] показано, что данное включение является строгим, если NP≠co-NP. Напомним [17], что класс NP_S состоит из бинарных отношений $R \subseteq \{0,1\}^* \times \{0,1\}^*$, которые

— полиномиально ограничены, т. е. если $(x, y) \in R$, то длина слова y полиномиально ограничена длиной слова x;

— полиномиально распознаваемы, т. е. существует полиномиальный алгоритм, который для любой пары (x, y) проверяет, принадлежит ли она отношению R или нет.

Задача поиска, связанная с отношением R, заключается в том, чтобы для любого x найти y такой, что $(x, y) \in R$. Эта задача содержится в классе P_S, если существует полиномиальный алгоритм, который для любого x либо находит y такой, что $(x, y) \in R$, либо сообщает об отсутствии такого y.

Покажем, что класс P_{PLS} эквивалентен классу P_S. Пусть задача локального поиска L принадлежит классу P_{PLS}, а входы и решения задачи L являются двоичными словами. Тогда бинарное отношение

$$R_L = \{(x, y) \mid y - \text{ локальный оптимум для входа } x \text{ задачи } L\}$$

полиномиально ограничено, так как в классе PLS содержатся только такие задачи, для которых длина записи любого допустимого решения ограничена полиномом от длины записи исходных данных. Полиномиальная распознаваемость этого отношения следует из полиномиальности алгоритма C в определении класса PLS. Принадлежность отношения R_L классу P_S следует из существования полиномиального алгоритма для задачи L. Таким образом, $\mathrm{P}_{PLS} \subseteq \mathrm{P}_S$.

Покажем обратное включение. Рассмотрим произвольное отношение R из класса P_S. Соответствующую задачу поиска представим в виде задачи из класса P_{PLS}. Рассмотрим произвольное двоичное слово x. По определению за полиномиальное время можно узнать, существует ли такое y, что $(x, y) \in R$. Если не существует, то будем считать, что x не является входом задачи локального поиска. Если да, то положим: множество допустимых решений $Sol(x) = \{y\}$, целевая функция $f(y, x) = 0$, окрестность $N(y, x) = \{y\}$. Такая задача локального поиска удовлетворяет определению 3, т. е. $\mathrm{P}_S \subseteq \mathrm{P}_{PLS}$.

Завершая обсуждение классов PLS, P_{PLS}, P_S и NP_S, заметим, что в настоящее время нет прямых или косвенных аргументов в пользу совпадения классов P_{PLS} и PLS. Таким образом, наряду с центральной проблемой теории сложности P=NP? существует аналогичная проблема P_{PLS}=PLS? о локальных оптимумах. Если $\mathrm{P}_{PLS} \neq$ PLS, то P$\neq$ NP. Может быть доказать первое неравенство легче, чем второе. В любом случае, вопрос о совпадении классов P_{PLS} и PLS представляется авторам весьма интересным и актуальным.

Определение 6 [21]. *Графом переходов* $TG_L(x)$ для входа x задачи L называется ориентированный граф с множеством вершин $Sol(x)$ и множеством дуг вида (s, s'), где $s' \in N(s, x)$ и $f(s', x) < f(s, x)$. Высота вершины s есть длина кратчайшего пути в графе $TG_L(x)$ из вершины s в сток, т. е. в локальный минимум. Высота графа $TG_L(x)$ равна макси-

мальной высоте его вершин.

Из определения следует, что граф переходов является ациклическим и в нём может быть экспоненциальное число вершин относительно длины входа x.

Определение 7 [21]. *Алгоритм локального спуска* LD для задачи L задаётся следующей последовательностью шагов:

Шаг 1. Найти допустимое решение s.

Шаг 2. Повторять шаги 2.1 и 2.2 пока s не станет локальным минимумом.

Шаг 2.1. Найти решение $s' \in N(s,x)$ такое, что $f(s',x) < f(s,x)$.

Шаг 2.2. Положить $s := s'$.

Шаг 3. Выдать локальный минимум.

По аналогии с симплекс-методом правило выбора решения s' на шаге 2.1 называют правилом замещения. Вообще говоря, не требуется, чтобы правило замещения было полиномиально вычислимой функцией от длины входа. Если s не является локальным оптимумом, то переходу от s к s' в графе $TG_L(x)$ соответствует ребро (s, s'). Применяя алгоритм LD к разным начальным решениям, будем получать различные пути, ведущие из начальных вершин в стоки. Высота вершины по сути есть оценка снизу на число шагов алгоритма LD для получения локального оптимума независимо от применяемого правила замещения. Таким образом, алгоритм LD работает экспоненциальное число шагов при любом правиле замещения тогда и только тогда, когда найдутся исходные данные рассматриваемой задачи, для которых высота графа переходов является экспоненциальной функцией от длины записи исходных данных.

Определение 8 [21]. Пусть задача L_1 сводится к задаче L_2 и h, g — соответствующие функции. Говорят, что сводимость является *плотной*, если для любого входа x задачи L_1 можно указать такое подмножество R допустимых решений задачи L_2 на входе $y = h(x)$, что

1) в R содержатся все локальные минимумы для входа y;

2) существует полиномиальный алгоритм, который для каждого решения p на входе x позволяет находить такое решение $q \in R$ для входа y, что $g(q,x) = p$;

3) пусть в графе переходов $TG_{L_2}(y)$ содержится такой ориентированный путь из вершины $q \in R$ в вершину $q' \in R$, что в нём нет промежуточных вершин из R, и пусть $p = g(q,x)$, $p' = g(q',x)$ — соответствующие решения на входе x. Тогда $p = p'$ или граф переходов $TG_{L_1}(x)$ содержит дугу, исходящую из вершины p и входящую в вершину p'.

Рассмотрим произвольный путь Π в графе $TG_{L_2}(y)$ с начальной и

конечной вершинами из множества R. Возьмём образ вершин этого пути в графе $TG_{L_1}(x)$ относительно отображения g. Условие 3 гарантирует, что участку пути Π между любыми двумя вершинами q, q' из множества R (не содержащему промежуточных вершин из R) соответствует либо вершина, являющаяся образом вершин q и q', либо ребро, соединяющее образы этих вершин. Следовательно, пути Π в графе $TG_{L_1}(x)$ можно поставить в соответствие путь $\Pi(g)$, вершины которого являются образами вершин из Π. При этом длина пути $\Pi(g)$ не превосходит длины исходного пути.

Пусть p — произвольная вершина в графе $TG_{L_1}(x)$ и пусть $q = g^{-1}(p)$ — её прообраз из множества R. Рассмотрим один из путей минимальной длины, ведущий из вершины q в некоторый сток графа $TG_{L_2}(y)$. В графе $TG_{L_1}(x)$ этому пути соответствует путь меньшей длины, ведущий из вершины p в один из стоков этого графа. Таким образом, высота вершины q не меньше высоты вершины p в графе $TG_{L_1}(x)$. Следовательно, высота графа $TG_{L_2}(y)$ не меньше высоты графа $TG_{L_1}(x)$ [18, 21]. Поэтому плотная сводимость сохраняет нижние оценки на число шагов алгоритма локального поиска.

Заметим, что в условии 3 в качестве ориентированного пути может рассматриваться дуга из q в q'. В этом случае в графе переходов $TG_{L_1}(x)$ такой дуге соответствует либо вершина p, либо дуга (p, p'). Такая ситуация может возникать, если в качестве множества R выбирается, например, всё множество допустимых решений.

Задача L из класса PLS является плотно полной, если все задачи из этого класса плотно сводятся к ней. Известно, что задача $(Circuit, Flip)$ и ряд других задач являются плотно полными [18, 21].

§ 2. Окрестности и их свойства

Задача о p-медиане моделирует ситуацию, когда в возможных пунктах производства нужно разместить p предприятий так, чтобы суммарное расстояние от них до потребителей было минимальным. Приведём описание этой задачи в соответствии с определением 1. Обозначим через F множество возможных пунктов производства, а через J множество потребителей. Пусть g_{ij} — расстояние между j-м потребителем и i-м возможным пунктом производства.

Вход: матрица $(g_{ij}), i \in F, j \in J$, и целое положительное число p.

Решение: любое подмножество $S \subset F$ мощности p.

Целевая функция: суммарное расстояние от открываемых предприятий до потребителей, т. е. $\sum\limits_{j \in J} \min\limits_{i \in S} g_{ij}$.

Требуется найти решение с минимальным значением целевой функции.

Рассмотрим следующие окрестности.

1) Окрестность $Swap(S)$. Она состоит из подмножеств $S' \subset F$ мощности p, которые получаются из S перестановкой одного элемента из S с элементом из дополнения $\overline{S} = F \setminus S$. Окрестность содержит $p(|F| - p)$ допустимых решений.

2) Окрестность $LK(S)$ [14, 15]. Она строится с помощью следующей итерационной процедуры. На каждом шаге просматриваются все пары элементов $i_1 \in S, i_2 \in \overline{S}$, каждый из которых ранее не использовался в перестановках, и выбирается наилучшая пара, т. е. пара, которая максимально уменьшает сумму расстояний, а если таких пар нет, минимальным образом увеличивает её. Если имеется несколько таких пар, то выбирается любая из них. Процедура завершается через k шагов, $1 \leqslant k \leqslant \min\{p, |F| - p\}$, и даёт k соседних решений для S.

3) Окрестность $LK_1(S)$ состоит из одного подмножества, которое получается за один шаг предыдущей процедуры. По построению эта окрестность является частью окрестности $Swap$.

4) Окрестность $FM(S)$ [9]. Она строится аналогично окрестности $LK(S)$. Каждый шаг этой процедуры состоит из двух этапов. На первом этапе просматриваются элементы из S, каждый из которых ранее не использовался, и выбирается наилучший. Такой элемент переводится в дополнение множества S. На втором этапе среди элементов множества $\overline{S}$, каждый из которых ранее не использовался, выбирается наилучший и переводится в S. Наилучшим элементом считается элемент, перевод которого в другую долю максимально уменьшает сумму расстояний, а если таких элементов нет, минимальным образом увеличивает её. Если имеется несколько наилучших элементов, то выбирается любой из них. Окрестность содержит k допустимых решений, $1 \leqslant k \leqslant \min\{p, |F| - p\}$.

5) Окрестность $FM_1(S)$ состоит из одного подмножества, которое получается за один шаг предшествующей процедуры. Данная окрестность была введена для задачи о разбиении графа [18] и обозначалась через FM-$Swap$.

6) Окрестность k-$Swap(S)$ [20] состоит из подмножеств, которые получаются из S следующим образом. Выбирается не более k пар (i_1, i_2), $i_1 \in S, i_2 \in \overline{S}$, без общих элементов и для каждой пары производится замена. В окрестности имеется $O(p^k(|F| - p)^k)$ элементов и её мощность является полиномиальной при любом фиксированном k.

Легко проверить, что задача о p-медиане с любой из этих окрестно-

стей принадлежит классу PLS. Ниже будут установлены связи между этими задачами локального поиска. Предварительно выясним отношения предшествования между введёнными окрестностями.

Лемма 1. *Справедливы следующие соотношения*:

$$FM_1 \preceq Swap \preceq LK_1 \preceq LK,$$
$$LK_1 \preceq Swap \preceq k\text{-}Swap,$$
$$FM_1 \preceq FM.$$

Доказательство легко следует из определения окрестностей. Таким образом, окрестность FM_1 является самой слабой, а окрестности $Swap$ и LK_1 эквивалентны.

Лемма 2. *Пусть задачи* $L_1 = (OP, N_1)$, $L_2 = (OP, N_2)$ *принадлежат классу* PLS *и* $N_1 \preceq N_2$. *Тогда* L_1 *сводится к* L_2.

Доказательство. Так как из N_2-оптимальности решения следует его N_1-оптимальность, то в качестве функций h и g из определения 4 возьмём тождественные функции. Их полиномиальная вычислимость очевидна. Лемма 2 доказана.

Следствие 1.

1) *Задача* (p-*median*, *Swap*) *эквивалентна задаче* (p-*median*, LK_1) *и сводится к задачам* (p-*median*, LK) *и* (p-*median*, k-*Swap*).

2) *Задача* (p-*median*, FM_1) *сводится к задачам* (p-*median*, FM), (p-*median*, *Swap*), (p-*median*, LK_1), (p-*median*, LK), (p-*median*, k-*Swap*).

Сформулируем достаточное условие, которое гарантирует плотную сводимость двух задач из класса PLS.

Лемма 3. *Пусть задачи* $L_1 = (OP, N_1)$, $L_2 = (OP, N_2)$ *принадлежат классу* PLS *и* $N_1 \preceq N_2$. *Если для каждого входа* x *оптимизационной задачи* OP *граф* $TG_{L_2}(x)$ *является подграфом графа* $TG_{L_1}(x)$, *то задача* L_1 *плотно сводится к задаче* L_2.

Доказательство. По лемме 2 задача L_1 сводится к задаче L_2. Покажем, что это све́дение является плотным. В качестве функций h и g возьмём тождественные функции, а в качестве R — множество всех допустимых решений. Условия 1 и 2 определения 8 легко проверяются. По условию граф $TG_{L_2}(x)$ является подграфом графа $TG_{L_1}(x)$. Поэтому любой путь в графе $TG_{L_2}(x)$ является путём в графе $TG_{L_1}(x)$, а каждый сток (локальный минимум) в графе $TG_{L_2}(x)$ является стоком в графе $TG_{L_1}(x)$. Следовательно, выполняется условие 3 определения плотной сводимости. Лемма 3 доказана.

Следствие 2. *Задача* $(p\text{-}median, Swap)$ *плотно сводится к задаче* $(p\text{-}median, LK_1)$.

Так как граф переходов в задаче с окрестностью LK_1 является подграфом графа переходов в задаче с окрестностью $Swap$, то утверждение следует из леммы 3.

§ 3. PLS-полные задачи

При формулировке и доказательстве следующих результатов используется задача о разбиении множества вершин графа на два подмножества равной мощности (GP). Приведём её постановку.

Вход: граф $G = (V, E)$ с чётным числом вершин $|V| = 2n$ и весами рёбер $w_e \geqslant 0, e \in E$.

Допустимое решение: любое разбиение множества V на два подмножества V_1, V_2 равной мощности.

Целевая функция: вес разреза $W(V_1, V_2)$, т. е. сумма весов рёбер, каждое из которых соединяет некоторую вершину из V_1 с некоторой вершиной из V_2.

Требуется найти допустимое решение максимального веса.

В дальнейшем рассматривается задача нахождения локального оптимума для задачи GP с окрестностью FM_1, которая определяется также как для задачи о p-медиане. Плотная PLS-полнота задачи (GP, FM_1) доказана в [18, 21].

Для доказательства полноты задачи о p-медиане с любой из введённых выше окрестностей этот факт достаточно установить для задачи с окрестностью FM_1. В следующей теореме устанавливается более сильный результат, который впоследствии позволит оценить число шагов алгоритма LD для ряда задач локального поиска.

Теорема 1. *Задача* (GP, FM_1) *плотно сводится к задаче* $(p\text{-}median, FM_1)$.

Доказательство. По исходным данным задачи GP построим исходные данные задачи о p-медиане так, чтобы любому допустимому разрезу (V_1, V_2) соответствовал набор из p предприятий и сумма значений целевых функций на этих решениях была постоянной. Тогда в качестве множества R из определения 8 можно взять всё множество допустимых решений задачи о p-медиане, графы переходов будут изоморфны и сведе́ние будет плотным.

Обозначим через E_i множество рёбер в графе G, инцидентных вершине $i \in V$, и пусть $W_i = \sum\limits_{e \in E_i} w_e$, $W = \sum\limits_{e \in E} w_e$. Положим $F = \{1, \ldots, |V|\}$, $J = \{1, \ldots, |E| + |V|\}, p = n$. Каждому $j = 1, \ldots, |E|$ поставим в соответст-

вие ребро $e \in E$ и положим

$$g_{ij} = \begin{cases} 0, & \text{если} \;\; e = (i_1, i_2), (i = i_1) \vee (i = i_2), \\ 2w_e & \text{в противном случае,} \end{cases}$$

а для каждого $j = |E| + 1, \ldots, |E| + |V|$ положим

$$g_{ij} = \begin{cases} 0, & \text{если} \;\; i = j - |E|, \\ W - W_i & \text{в противном случае.} \end{cases}$$

Разрезу (V_1, V_2) поставим в соответствие решение $S = V_1$ и проверим, что

$$\sum_{j \in J} \min_{i \in S} g_{ij} + W(V_1, V_2) = nW.$$

Заметим, что

$$\sum_{j=1}^{|E|} \min_{i \in S} g_{ij} = 2 \sum (w_e \mid e = (i_1, i_2), i_1, i_2 \notin S)$$

и

$$\sum_{j=1+|E|}^{|J|} \min_{i \in S} g_{ij} = \sum_{i \notin S} (W - W_i) = nW - \sum_{i \notin S} W_i.$$

Так как в сумме $\sum\limits_{i \notin S} W_i$ вес каждого ребра из разреза (V_1, V_2) учитывается ровно один раз, а вес ребра $e = (i_1, i_2), i_1, i_2 \notin V_1$ — дважды, то

$$\sum_{i \notin S} W_i = W(V_1, V_2) + \sum_{j=1}^{|E|} \min_{i \in S} g_{ij}.$$

Отсюда следует утверждение теоремы 1.

Следствие 3. *Для полноты задачи (p-median, N) из класса* PLS *достаточно, чтобы окрестность N была не слабее окрестности FM_1.*

Доказательство. Если окрестность N совпадает с окрестностью FM_1, то утверждение непосредственно следует из теоремы 1. Если окрестность N сильнее окрестности FM_1, то утверждение следует из леммы 2 и теоремы 1.

Покажем, как следствие 3 может быть использовано для анализа сложности задач локального поиска. Из леммы 1 и следствия 3 вытекает, что задача о p-медиане с окрестностями $Swap$, LK, LK_1, FM, $k - Swap$

является полной. Рассмотрим одну из древесных окрестностей, которая может быть определена следующим образом. В ней содержится окрестность $Swap$ и к каждому элементу этой окрестности добавляется последовательность соседних решений по правилам окрестности LK (рис. 1). Новую окрестность обозначим через $TreeLK$.

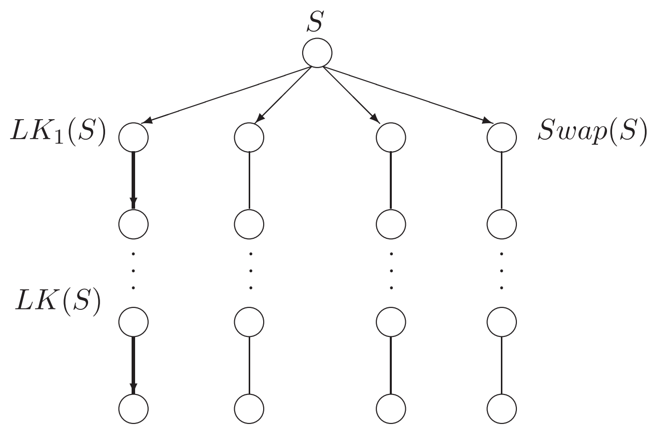

Рис. 1. Окрестность $TreeLK(S)$

Она является полиномиально проверяемой и $TreeLK$-оптимальное решение является $Swap$-оптимальным. Отсюда следует PLS-полнота задачи (p-$median, TreeLK$). Другие примеры древесных окрестностей можно найти в [16].

Следствие 4. *Если $P_{PLS} \neq$ PLS и задача о p-медиане с окрестностью N полиномиально разрешима, то $FM_1 \not\preceq N$.*

Доказательство. Если $FM_1 \preceq N$, то задача (p-$median$, N) является полной. Следовательно, $P_{PLS} =$ PLS. Таким образом, если N — полиномиально проверяемая окрестность, то для полиномиальной разрешимости задачи (p-$median$, N) необходимо, чтобы эта окрестность была либо слабее окрестности FM_1, либо эти окрестности были несравнимы. Первое означает, что множество FM_1-оптимальных решений должно быть собственным подмножеством множества N-оптимальных решений. Второе означает, что из FM_1-оптимальности решения не должна следовать N-оптимальность и наоборот. Следствие 4 доказано.

Аналогичные утверждения можно сформулировать для других известных оптимизационных задач: задача коммивояжера, задача о максимальном разрезе, задача о разбиении графа и другие [21].

Следующее утверждение показывает, что следствие 3 можно усилить и получить плотную полноту задачи о p-медиане с каждой из окрестностей $Swap, LK$, LK_1, FM. Вопрос о плотной полноте задачи (p-$median$, k-$Swap$) в настоящее время остаётся открытым.

Теорема 2. *Задача о p-медиане с каждой из окрестностей* $Swap, LK$, LK_1, FM, FM_1 *является плотно полной.*

Доказательство. Задача GP с каждой из указанных окрестностей, кроме окрестности LK_1, является плотно полной [18, 21]. Сводимость теоремы 1 такова, что существует взаимно однозначное соответствие между допустимыми разбиениями задачи GP и допустимыми решениями некоторого частного случая задачи о p-медиане. Значения целевых функций на соответствующих решениях в сумме дают постоянную величину. Таким образом, граф переходов задачи GP с любой из рассматриваемых окрестностей изоморфен графу переходов этого частного случая задачи о p-медиане с той же окрестностью. Отсюда следует утверждение теоремы.

Для окрестности LK_1 утверждение теоремы следует из плотной полноты задачи (p-$median, Swap$) и следствия 2.

§ 4. Сложность локального спуска

Оценим в худшем случае число шагов алгоритма LD для рассматриваемых окрестностей.

Теорема 3. *Алгоритм локального спуска для задачи о p-медиане в худшем случае требует экспоненциального числа шагов для нахождения локального минимума при любом выборе направления спуска с любой из следующих окрестностей*: $Swap, LK$, LK_1, FM, FM_1.

Справедливость утверждения следует из теоремы 2 и существования задачи экспоненциальной высоты в классе PLS [21].

Заметим, что в отличие от соответствующих оценок для симплекс-метода экспоненциальные оценки из теоремы 3 не зависят от правила замещения. Оценки остаются экспоненциальными при использовании любых детерминированных или недетерминированных процедур, полиномиально ограниченных или неограниченных алгоритмов для выбора соседнего решения с лучшим значением целевой функции.

Другое приложение теоремы 2 связано с задачей локального поиска при фиксированной начальной точке: для заданных входа, окрестности и начального решения найти локальный минимум, который может быть получен алгоритмом LD из данного начального решения. Это более сложная задача, чем предшествующая задача нахождения локального

минимума. Раньше требовалось найти любой локальный минимум. Теперь надо найти локальный минимум, достижимый из заданного начального решения. В силу определения 4 для любой задачи из класса PLS соответствующая задача нахождения локального минимума с фиксированной начальной точкой лежит в классе PSPACE [5]. Действительно, по предположению длина начального решения, как и всякого допустимого решения, ограничена полиномом от длины входа. Нахождение лучшего соседнего решения, если такое существует, осуществляется за полиномиальное время и, следовательно, требует полиномиально ограниченной памяти. Так как не нужно хранить все промежуточные решения, то требуемое пространство всегда полиномиально ограничено.

Теорема 4. *Нахождение локального минимума в задаче о p-медиане при фиксированной начальной точке является* PSPACE-*полной с любой из следующих окрестностей*: $Swap, LK$, LK_1, FM, FM_1.

Справедливость утверждения следует из теоремы 2, существования такой задачи в классе PLS, для которой отыскание локального оптимума, соответствующего заданной начальной точке, является PSPACE-полной [1, 2], а также из того факта, что плотная сводимость двух задач из класса PLS влечёт полиномиальную сводимость соответствующих задач нахождения локального оптимума при фиксированной начальной точке [18, 21].

В заключение параграфа приведём пример исходных данных задачи о p-медиане, когда алгоритм LD с окрестностью LK_1 выполняет экспоненциальное число шагов. Тогда из леммы 1 следует, что алгоритм наискорейшего спуска, на шаге 2 которого всегда выбирается только наилучший сосед, является экспоненциальным с окрестностями $Swap$ и LK_1.

В работе [20] для обобщённой задачи о раскраске вершин графа в два цвета (2-$GGCP$) предложено семейство исходных данных, для которых алгоритм наискорейшего спуска с окрестностью $Flip$ совершает экспоненциальное число шагов. Напомним постановку задачи.

Вход: взвешенный граф $G = (V, E)$, $w_e, e \in E$, — целочисленные веса рёбер.

Решение: раскраска вершин в два цвета, т. е. функция вида $c : V \mapsto \{1, 2\}$.

Целевая функция: суммарный вес рёбер, вершины которых окрашены в один цвет.

Окрестность $Flip$: все раскраски, отличающиеся от данной цветом одной вершины. Задача состоит в поиске раскраски с минимальным суммарным весом рёбер, вершины которых окрашены в один цвет. Задача

нахождения локального минимума для окрестности *Flip* является плотно PLS-полной [20].

Теорема 5. *Задача* (2-*GGCP*, *Flip*) *плотно сводится к задаче* (*p-median*, LK_1).

Доказательство. Положим $F = \{1, \dots, 2|V|\}$, $J = \{1, \dots, |V| + 2|E|\}$, $p = |V|$ и $W = \sum\limits_{e \in E} |w_e| + 1$. Каждой вершине $v \in V$ поставим в соответствие две строки i_v, i'_v матрицы g_{ij} и один столбец j_v. Каждому ребру $e = (j_1, j_2) \in E$ — два столбца $j_1(e)$, $j_2(e) \in J$.

При $j_v = 1, \dots, |V|$ положим

$$g_{ij_v} = \begin{cases} 0, & \text{если } (i = i_v) \vee (i = i'_v), \\ W & \text{в противном случае.} \end{cases}$$

При $e = (j_1, j_2) \in E$ и $w_e \geqslant 0$ положим

$$g_{ij_1(e)} = \begin{cases} 0, & \text{если } (i = j_1(e)) \vee (i = j_2(e)), \\ w_e, & \text{если } (i = j'_1(e)) \vee (i = j'_2(e)), \\ W & \text{в противном случае;} \end{cases}$$

$$g_{ij_2(e)} = \begin{cases} 0, & \text{если } (i = j'_1(e)) \vee (i = j'_2(e)), \\ w_e, & \text{если } (i = j_1(e)) \vee (i = j_2(e)), \\ W & \text{в противном случае.} \end{cases}$$

Структура матрицы g_{ij} схематично изображена на рис. 2.

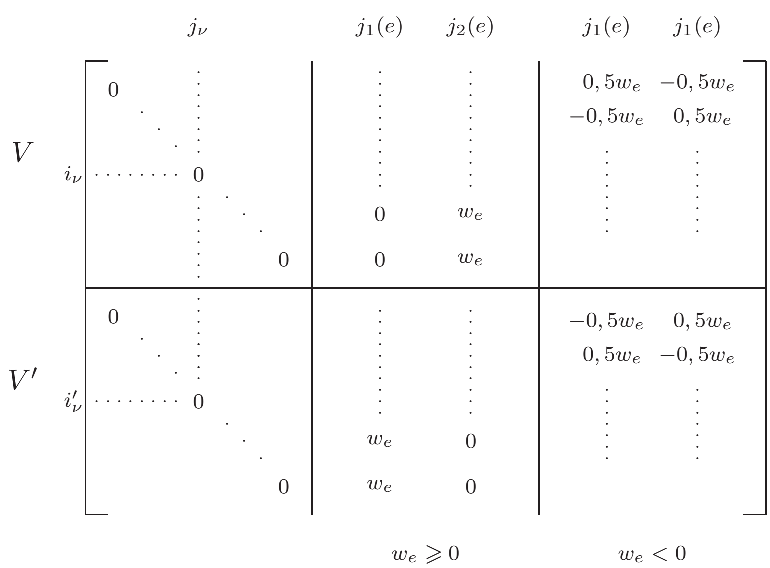

Рис. 2. Структура матрицы g_{ij}

Если же $w_e < 0$, то положим

$$g_{ij_1(e)} = \begin{cases} w_e/2, & \text{если } \ (i = j_1(e)) \vee (i = j_2'(e)), \\ -w_e/2, & \text{если } \ (i = j_1'(e)) \vee (i = j_2(e)), \\ W & \text{в противном случае;} \end{cases}$$

$$g_{ij_2(e)} = \begin{cases} w_e/2, & \text{если } \ (i = j_1'(e)) \vee (i = j_2(e)), \\ -w_e/2, & \text{если } \ (i = j_1(e)) \vee (i = j_2'(e)), \\ W & \text{в противном случае.} \end{cases}$$

Предложенная конструкция определяет полиномиально вычислимое отображение входов задачи 2-$GGCP$ во входы задачи о p-медиане. Пусть $S \subset F$ — локальный минимум относительно окрестности LK_1. Убедимся в том, что для любого $v \in V$ одна и только одна из строк i_v, i'_v входит в множество S. Предположим, что i_v, $i'_v \notin S$. Тогда найдётся такая вершина v_0, что строки i_{v_0}, i'_{v_0} входят в S, так как $|S| = |V|$. Замена одной из этих строк на любую из строк i_v, i'_v приводит к уменьшению минимума в столбце j_v на величину W и, возможно, к увеличению минимума в столбцах, соответствующих ребрам, инцидентным вершине v_0. Для каждого такого ребра суммарный рост не превосходит $|w_e|$. Таким образом, в окрестности решения S имеется решение с меньшим значением целевой функции, что противоречит локальной оптимальности решения S. Случай i_v, $i'_v \in S$ рассматривается аналогично.

По решению S определим раскраску вершин графа G:

$$c_S(v) = \begin{cases} 1, & \text{если } \ i_v \in S, \\ 2 & \text{в противном случае.} \end{cases}$$

Проверим, что раскраска $c_S(v)$ является локальным минимумом относительно окрестности $Flip$. Пусть некоторое ребро $e \in E$ инцидентно вершинам одного цвета. Рассмотрим два случая.

1) Пусть $w_e > 0$. Для ребра $e = (j_1(e), j_2(e))$ найдём строки i_1, i_2, соответствующие вершинам $j_1(e), j_2(e)$. Если $i_1, i_2 \in S$, то $i'_1, i'_2 \notin S$ и в матрице g_{ij} минимум в столбце $j_1(e)$ равен 0, а в столбце $j_2(e)$ равен w_e, т. е. величина w_e учитывается как в весе $c_S(v)$, так и в весе S. Если же $i_1, i_2 \notin S$, то $i'_1, i'_2 \in S$ и минимум в столбце $j_1(e)$ равен w_e, а в столбце $j_2(e)$ равен 0. Снова приходим к тому же выводу.

2) Пусть $w_e < 0$. Если $i_1, i_2 \in S$, то в обоих столбцах матрицы минимум равен $w_e/2$, что в сумме даёт w_e. Если же $i'_1, i'_2 \in S$, то снова в обоих столбцах получаем $w_e/2$, что в сумме даёт w_e.

Теперь рассмотрим ребро, соединяющее вершины разного цвета, и пусть $i_1 \in S$, $i_2 \notin S$. Если $w_e > 0$, то в столбцах $j_1(e), j_2(e)$ минимум

равен нулю и вес ребра не входит в суммарный вес решения S. Если же $w_e < 0$, то минимум в столбце $j_1(e)$ равен $w_e/2$, а в столбце $j_2(e)$ равен $-w_e/2$, что в сумме снова даёт ноль. Таким образом, значения целевых функций на решениях S и $c_S(v)$ совпадают.

Предположим, что решение $c_S(v)$ не является $Flip$ оптимальным. Тогда найдётся вершина $v \in V$, изменение цвета которой приводит к уменьшению целевой функции. В этом случае решение S не является LK$_1$-оптимальным, так как изменение цвета вершины v соответствует замене i_v на i'_v с тем же изменением целевой функции. Значит, $c_S(v)$ является локальным минимумом, и задача (2-$GGCP, Flip$) сводится к задаче (p-$median, LK_1$).

Проверим, что это сведе́ние является плотным. В качестве элементов множества R из определения 8 выберем все допустимые решения S, обладающие следующим свойством: для каждой вершины $v \in V$ либо $i_v \in S$, либо $i'_v \in S$. Ранее мы убедились, что это множество содержит все локальные минимумы, т. е. условие 1 выполнено.

Пусть $c_S(v)$ — произвольная раскраска вершин в два цвета. Включим в множество S строку i_v, если вершина v имеет цвет 1, и строку i'_v, если этот цвет 2. Такой алгоритм устанавливает взаимно однозначное соответствие между раскрасками и элементами множества R, т. е. условие 2 также выполнено.

Проверим последнее условие. Заметим, что для задачи (p-$median$, LK_1) в графе переходов нет дуг вида (q_1, q_2), $q_1 \in R$, $q_2 \notin R$, так как значение целевой функции для q_2 всегда больше, чем для q_1. Другими словами, любой путь, начинающийся и оканчивающийся в R, целиком лежит в R, и путь без промежуточных вершин из R есть дуга (q_1, q_2), $q_1, q_2 \in R$. Но такой дуге соответствует дуга в графе переходов задачи (2-$GGCP, Flip$). Отсюда следует утверждение теоремы 5.

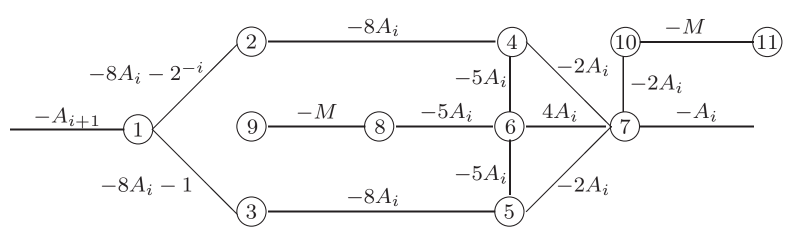

Рис. 3. Блок $i : A_i = 20^{i-1}$

Приведём описание примера для задачи (2-$GGCP, Flip$) (см. рис. 3 и 4). Граф состоит из K блоков и одной цепи из трёх вершин. Каждый блок

состоит из 11 вершин. Вершину с номером 1 назовём входной, а с номером 7 — выходной. Входная вершина блока i соединяется ребром с выходной вершиной блока $i+1, i = K-1, \ldots, 1$. Входная вершина блока K соединяется ребром с самой правой вершиной цепи. Рёбрам блоков приписаны положительные и отрицательные веса, величины которых приведены на рис. 3. Выходная вершина первого блока инцидентна только четырём вершинам с номерами 4, 5, 6 и 10. В графе имеются рёбра с весами M и $-M$. В качестве M выбрана достаточно большая положительная константа, чтобы в локальном минимуме гарантировать одинаковый цвет вершин, инцидентных ребру с весом $-M$, и разные цвета вершин, инцидентных ребру с весом M.

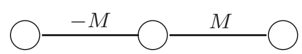

Рис. 4. Цепь с тяжелыми ребрами

Если алгоритм наискорейшего спуска начинает работу с раскраски всех вершин в первый цвет, то в соответствующем графе переходов получаем путь, содержащий 2^K дуг [20]. Так как в приведённом выше графе имеется $12K + 2$ дуг и $11K + 3$ вершин, то из доказательства теоремы 5 следует, что требуемый пример для задачи о p-медиане имеет $22K + 6$ строк и $35K + 7$ столбцов. Задача (2-$GGCP, Flip$) с таким графом плотно эквивалентна данному частному случаю задачи о p-медиане с окрестностью LK_1. Более того, их графы переходов изоморфны. Следовательно, алгоритм LD, начинающий работу с решения $S = V$, будет порождать путь длины 2^K.

§ 5. Погрешность локальных оптимумов

Окрестность называют *точной*, если любой локальный оптимум является глобальным. Существование точной полиномиально проверяемой окрестности делает локальный спуск точным методом, как например, симплекс–метод. Однако для задачи коммивояжёра существование такой окрестности влечёт равенство P=NP [2]. В данном параграфе будет показана бесперспективность поиска таких окрестностей и для задачи о p-медиане. Кроме того, будет установлено, что локальный минимум для любой полиномиально проверяемой окрестности может быть больше глобального в произвольное число раз, а число локальных минимумов может расти экспоненциально с ростом размерности задачи. В конце параграфа будет приведено семейство исходных данных, когда расстояние от глобального минимума до ближайшего локального минимума равно

$2p$, т. е. равно диаметру допустимой области. Более того, любой путь от локального минимума до глобального проходит через область бо́льших значений целевой функции, что является серьезным препятствием для таких методов как имитация отжига, поиск с запретами и др.

Далее, говоря о погрешности приближенного решения s для входа x оптимизационной задачи на минимум, будем иметь в виду отношение $f(x, s)/\mathrm{Opt}(x)$, где $\mathrm{Opt}(x)$ — оптимальное значение целевой функции. Известно [10], что для любого целого $\rho > 1$ задача вычисления приближённого решения с гарантированной оценкой точности ρ для задачи о p-медиане является NP-трудной. Этот результат основывается на полиномиальной сводимости задачи о вершинном покрытии к задаче о p-медиане. Задача о вершинном покрытии формулируется следующим образом [1].

Вход: граф $G = (V, E)$ и целое положительное число k.

Вопрос: существует ли подмножество вершин $V^{'} \subseteq V$ мощности не более k такое, что любое ребро из множества E инцидентно хотя бы одной вершине из $V^{'}$? Множество $V^{'}$ называют вершинным покрытием.

Рассмотрим пример задачи о p-медиане со следующими входными данными: $F = V$, $J = E$, $p = k$ и

$$g_{ij} = \begin{cases} 1, & \text{если вершина } i \text{ инцидентна ребру } e_j, \\ \rho(|E| + 1) & \text{в противном случае.} \end{cases}$$

В исходном графе G имеется вершинное покрытие размера k тогда и только тогда, когда оптимальное значение данной задачи о p-медиане равно величине $|E|$. Если в графе G нет вершинного покрытия размера k, то значение целевой функции на произвольном допустимом решении можно оценить снизу величиной

$$|E| - 1 + \rho(|E| + 1) = \rho|E| + |E| + \rho - 1 > \rho|E|.$$

Таким образом, если для задачи о p-медиане существует приближённый алгоритм с гарантированной оценкой ρ, то можно дать точный ответ о вершинном покрытии.

Предположим, что для некоторой полиномиально проверяемой окрестности существует такая константа ρ, что все локальные минимумы в задаче о p-медиане превышают глобальный минимум не более чем в ρ раз. Тогда алгоритм локального спуска будет давать ρ-приближённые решения. В частности, такие решения он получит и для указанного семейства исходных данных. Но для этого семейства алгоритм работает не более

$|E|$ шагов до достижения локального минимума. Действительно, на каждом шаге локального спуска происходит уменьшение целевой функции на величину, кратную $\rho(|E|+1)-1$. Другими словами, алгоритм LD является полиномиальным для этого семейства исходных данных. Таким образом, получаем точный полиномиальный алгоритм для решения задачи о вершинном покрытии. Значит, предположение о существовании окрестности с указанными свойствами влечёт равенство P=NP.

Теорема 6. *Если* P$\neq$NP, *то для любого* $\rho > 1$ *и любой полиномиально проверяемой окрестности найдутся исходные данные задачи о p-медиане, для которых существует локальный минимум, отклоняющийся более чем в* ρ *раз от оптимального решения задачи.*

Из теоремы 6 при $\rho = 1$ получаем следующее утверждение.

Следствие 5. *Если* P $\neq$NP, *то для задачи о p-медиане не существует точных полиномиально проверяемых окрестностей.*

В [21] для произвольной задачи (OP, N) из класса PLS анонсированы без доказательства два утверждения, которые позволяют оценивать качество локальных оптимумов как приближённых решений оптимизационной задачи OP. При формулировке первого из них предполагалась полиномиальная ограниченность значений целевой функции. Покажем, что этот результат справедлив и в более общей ситуации. Для этого введём понятие псевдополиномиальной ограниченности, которое означает, что диаметр области значений целевой функции растет не быстрее, чем значение некоторого полинома от длины входа x и величины $\mathrm{Max}(x)$, которая означает наибольшее по модулю число, содержащееся во входе x задачи [1, 6, 17]. Приведём точную формулировку данного определения.

Определение 9. Задача OP называется псевдополиномиально ограниченной, если существует такой полином p от длины входа x и $\mathrm{Max}(x)$, что для любого входа x и любого решения $s \in Sol(x)$ выполняется неравенство

$$f(s,x) - \mathrm{Opt}(x) \leqslant p(|x|, \mathrm{Max}(x)).$$

Множество оптимизационных задач, удовлетворяющих данному определению, обозначим через NPO_{DPPB}.

Теорема 7. *Пусть* $OP \in \mathrm{NPO}_{DPPB}$, $(OP, N) \in \mathrm{PLS}$ *и целевая функция принимает только целочисленные значения. Если* P$\neq$NP *и задача вычисления приближённого решения с гарантированной оценкой точности* ρ *является* NP-*трудной в сильном смысле, то найдутся исходные данные, для которых существует локальный оптимум, отклоняющийся более чем*

в ρ раз от оптимального решения.

Доказательство. Вместо задачи OP рассмотрим задачу нахождения приближённого решения с гарантированной оценкой точности ρ. Эту задачу обозначим через Π^ρ. По условию теоремы задача Π^ρ является NP-трудной в сильном смысле. Тогда по определению [1, 6, 17] существует полином q такой, что подзадача Π_q^ρ, образованная входами x, для которых выполняется неравенство $\mathrm{Max}(x) \leqslant q(|x|)$, является NP-трудной.

Предположим также, что существует такая полиномиально проверяемая окрестность N, что все её локальные оптимумы являются ρ-приближёнными решениями. Покажем, что в этом случае подзадача Π_q^ρ является полиномиально разрешимой (см. [1], теорема 7.2) и P $=$ NP.

Возьмём произвольный вход x этой подзадачи и некоторое начальное решение s^0, которое существует по определению класса PLS. Из псевдополиномиальной ограниченности задачи OP получаем, что

$$f(s^0, x) - \mathrm{Opt}(x) \leqslant p(|x|, \mathrm{Max}(x)) \leqslant p(|x|, q(|x|)).$$

Так как целевая функция по условиям теоремы принимает целые значения, то из последнего неравенства следует, что алгоритм локального спуска, начиная с решения s^0, за полиномиальное время найдёт некоторый локальный оптимум. Таким образом, если все локальные оптимумы являются ρ-приближёнными решениями, то NP-трудная задача Π_q^ρ является полиномиально разрешимой, что противоречит предположению о том, что P$\neq$NP. Следовательно, найдутся исходные данные задачи OP с таким локальным оптимумом, который отклоняется более чем в ρ раз от оптимального решения. Теорема 7 доказана.

Следствие 6. *Если* P $\neq$NP *и задача OP из* NPO_{DPPB} *является* NP-*трудной в сильном смысле, то для неё не существует точных полиномиально проверяемых окрестностей.*

Предыдущие результаты можно усилить, рассмотрев более широкий класс задач локального поиска, для которых диаметр графа переходов ограничен полиномом от $|x|$ и $\mathrm{Max}(x)$. Теорема 7 и следствие 6 будут выполняться для всех задач из этого класса. Легко проверить, что задачи из класса NPO_{DPPB} попадают в этот более широкий класс. Обратное, вообще говоря, не верно.

Теорема 8. *Если* NP $\neq$co-NP *и задача вычисления приближённого решения с гарантированной оценкой точности ρ для задачи OP является* NP-*трудной, то найдутся исходные данные, для которых существует локальный оптимум в задаче* (OP, N) из PLS, *отклоняющийся более чем*

в ρ раз от оптимального решения.

Доказательство. По предположению теоремы задача Π^ρ является NP-трудной. Следовательно, существует NP-полный язык π, который сводится по Тьюрингу за полиномиальное время к задаче Π^ρ. Это означает, что существует машина Тьюринга T с оракулом [1, 5, 17], которая за полиномиальное время распознает язык π. В процессе работы машина обращается к оракулу, который получив вход задачи Π^ρ, выдаёт ρ-приближённое решение. По определению оракульной машины [1] такое обращение принимается за один шаг её работы. Получив приближённое решение, машина продолжает работу в соответствии со своими инструкциями. Будем считать, что существует полином p такой, что T выполняет не более $p(|x|)$ шагов на каждом входе x [5, 17].

Предположим, что существует такая полиномиально проверяемая окрестность N, что все локальные оптимумы являются ρ-приближёнными решениями. Покажем, что в этом случае язык π^c, являющийся дополнением языка π, принадлежит классу NP. Тогда (см. [1], теорема 7.2) NP $=$ co-NP.

По машине T построим недетерминированную машину T', которая для произвольного входа x за полиномиальное время будет определять принадлежность этого входа языку π^c. Машина T' отличается от исходной тем, что обращение к оракулу она заменяет отгадыванием локального оптимума в задаче (OP, N). Это можно сделать за полиномиальное время на недетерминированной машине, так как по определению класса PLS длина записи локального оптимума и проверка его локальной оптимальности осуществляется за полиномиальное время. По предположению этот локальный оптимум является ρ-приближённым решением. Значит, машина T' будет давать тот же ответ что и исходная. Так как число шагов машины T не превышает $p(|x|)$, то количество предъявлений локального оптимума тоже ограничено полиномом $p(|x|)$. Таким образом, машина T' работает полиномиальное недетерминированное время. Если на последнем шаге работы машины T' получается ответ «да», то $x \notin \pi^c$, иначе $x \in \pi^c$. Итак, язык π является NP-полным, его дополнение π^c лежит в NP, значит NP=co-NP. Теорема 8 доказана.

Следствие 7. *Если* NP $\neq$co-NP *и задача* OP *является* NP*-трудной, то для неё не существует точных полиномиально проверяемых окрестностей.*

Приведём семейство исходных данных, когда для любой из окрестностей $Swap, LK$, LK_1, FM, FM_1, k-$Swap$ существует экспоненциальное число локальных минимумов, ни один из которых не является глобаль-

ным. Более того, все они в ρ раз отличаются от глобального оптимума. В этом примере матрица (g_{ij}) состоит из p блоков размера $p \times p$ и p дополнительных строк, которые соответствуют оптимальному решению задачи. Все элементы каждого блока равны ρ. Таким образом, матрица состоит из p^2 столбцов и $p^2 + p$ строк:

$$\left[\begin{array}{cccc}
\boxed{\begin{matrix}\rho & \cdots & \rho\\ \vdots & & \vdots\\ \rho & \cdots & \rho\end{matrix}} & & W & \\
 & \boxed{\begin{matrix}\rho & \cdots & \rho\\ \vdots & & \vdots\\ \rho & \cdots & \rho\end{matrix}} & & \\
 & & \ddots & \\
W & & & \boxed{\begin{matrix}\rho & \cdots & \rho\\ \vdots & & \vdots\\ \rho & \cdots & \rho\end{matrix}} \\
\boxed{\begin{matrix}1 & & W\\ & \ddots & \\ W & & 1\end{matrix}} & \boxed{\begin{matrix}1 & & W\\ & \ddots & \\ W & & 1\end{matrix}} & \cdots & \boxed{\begin{matrix}1 & & W\\ & \ddots & \\ W & & 1\end{matrix}}
\end{array}\right]$$

Пусть $W > \rho p^2$. Элементы, которые находятся в первых p^2 строках и не попали в блоки, положим равными W. В последних p строках разместим p квадратных матриц порядка p, в которых на диагонали находятся единицы, а остальные элементы равны W. Легко проверить, что последние p строк дают единственный глобальный минимум. Он равен p^2. Кроме него имеется p^p допустимых решений со значением целевой функции ρp^2. В каждом из них содержится ровно одна строка из каждого блока. Во всех оставшихся решениях значение целевой функции строго больше ρp^2. В окрестностях FM_1, FM, $Swap, LK_1, LK$, k-$Swap$ имеется p^p локальных минимумов, не являющихся глобальными. Эти локальные минимумы одинаковы. В окрестности каждого из них содержатся другие локальные минимумы. Они образуют своего рода «плато» экспоненциальной мощности. Попадая на него, такие методы как поиск

с запретами теряют способность целенаправленного поиска. Фактически уйти с такого плато метод уже не позволяет. В этом смысле блуждание по плато является аналогом зацикливания, которое наблюдается в симплекс-методе при попадании в вырожденную вершину. Для преодоления таких ловушек требуются специальные процедуры. Рассмотрим граф соседства [21], в котором вершинами являются допустимые решения, а дуга проводится в том случае, когда одно решение принадлежит окрестности другого. Для данного семейства исходных данных любой путь в таком графе от локального минимума до глобального будет проходить через решения с высокими значениями целевой функции. Алгоритму локального поиска с окрестностью $Swap$ потребуется не менее $\lfloor p/2 \rfloor$ шагов с ростом целевой функции вдоль любого такого пути. Образно говоря, глобальный минимум отделён от всех локальных минимумов своего рода «горным хребтом», преодоление которого может оказаться трудным для методов локального поиска. Увеличение константы W приводит к падению вероятности преодоления этого препятствия пороговыми методами, например, методом имитации отжига. Рост параметра p приводит к увеличению кратчайшего пути от локального оптимума до глобального, что отрицательно влияет на эффективность поиска с запретами. Генетические алгоритмы, имея начальную популяцию из первых p^2 строк, сталкиваются с теми же трудностями, так как оператор скрещивания не может дать последние p строк, и только случайные мутации могут это сделать. Если же генетический алгоритм использует процедуру локального спуска на стадии улучшения «потомства», то влияние мутаций будет сильно ослаблено и вероятность получить глобальный оптимум окажется еще ниже. Таким образом, данное семейство примеров, несмотря на свою простую структуру, является сложным для большинства метаэвристик.

§ 6. Заключение

В статье показано, что задача о p-медиане с рядом окрестностей оказывается в классе PLS-полных задач локального поиска. Учитывая ранее полученные результаты [13, 18, 21], можно констатировать, что ситуация с задачами локального поиска всё более напоминает классическую, связанную с исследованием переборных задач и возникновением центрального вопроса дискретной математики — можно ли исключить перебор при решении NP-трудных задач. В теории сложности математической формулировкой этой проблемы является вопрос о совпадении классов P и NP. Для задач локального поиска проблема уменьшения перебора математически формулируется как вопрос о совпадении классов P_{PLS} и

PLS.

Как уже отмечалось во введении, класс PLS скорее всего не настолько сложен, чтобы содержать NP-трудные задачи. Экспериментальные исследования [11] и многочисленные приложения показывают, что с практической точки зрения найти локальный минимум для данных окрестностей не представляет труда. Эта задача гораздо проще нахождения глобального минимума. Даже алгоритм LD в среднем требует линейного числа шагов от случайно выбранной точки до локального минимума [15, 19]. Эти данные косвенно свидетельствуют в пользу гипотезы о том, что в среднем трудоёмкость нахождения локального оптимума для PLS-полных задач полиномиальна. Однако без оценки роста числа шагов в асимптотике трудно сделать достаточно определённый вывод о совпадении или несовпадении классов P_{PLS} и PLS. Тем не менее теоретическая и экспериментальная проверка этой гипотезы является одним из важнейших направлений дальнейших исследований в области задач локального поиска.

ЛИТЕРАТУРА

1. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.
2. **Пападимитриу Х., Стайглиц К.** Комбинаторная оптимизация. Алгоритмы и сложность. М.: Мир, 1985.
3. **Роджерс Х.** Теория рекурсивных функций и эффективная вычислимость. М.: Мир, 1972.
4. **Схрейвер А.** Теория линейного и целочисленного программирования. М.: Мир, 1991.
5. **Atallah M.** Algorithms and theory of computation handbook. Boca Raton; FL: CRC Press, 1999.
6. **Ausiello G., Crescenzi P., Gambosi G., Kann V., Marchetti-Spaccamela A., Protasi M.** Complexity and approximation: combinatorial optimization problems and their approximability properties. Berlin: Springer, 1999.
7. **Brucker P., Hurink J., Werner F.** Improving local search heuristics for some scheduling problems. I // Discrete Appl. Math. 1996. V. 65, N 1–3. P. 97–122.
8. **Brucker P., Hurink J., Werner F.** Improving local search heuristics for some scheduling problems. II // Discrete Appl. Math. 1997. V. 72, N 1–2. P. 47–69.
9. **Fiduccia C. M., Mattheyses R. M.** A linear-time heuristic for improving network partitions // Proc. of the 19th Design Automation Conference. Los Alamitos, CA: IEEE Comput. Soc. Press, 1982. P. 175–181.

10. **Fisher M. L.** Worst-case analysis of heuristic algorithms // Manag. Sci. 1980. V. 26, N. 1. P. 1–18.

11. **Hansen P., Mladenović N.** Variable neighborhood search for the p-median // Location Science. 1997. V. 5, N 4. P. 207–226.

12. **Hromkovič J.** Algorithmics for hard problems. Berlin: Springer, 2004.

13. **Johnson D. C., Papadimitriou C. H., Yannakakis M.** How easy is local search? // J. Comput. and System Sciences. 1988. V. 37, N. 1. P. 56–87.

14. **Kernighan B. W., Lin S.** An effective heuristic procedure for partitioning graphs // Bell System Techn. J. 1970. V. 49. P. 291–307.

15. **Kochetov Yu., Alekseeva E., Levanova T., Loresh M.** Large neighborhood local search for the p-median problem // Yugoslav Journal of Oper. Res. 2005. V.15. P. 53–63.

16. **Mautor T.** Intensification neighborhoods for local search methods // Essays and surveys in metaheuristics. Boston: Kluwer Academic Publishers, 2002. P. 493–508.

17. **Papadimitriou C. H.** Computational comlexity. Reading, MA: Addison–Wesley Publ. Company, 1994.

18. **Schäffer A. A., Yannakakis M.** Simple local search problems that are hard to solve // SIAM J. Comput. 1991. V. 20, N. 1. P. 56–87.

19. **Tovey C.** Local improvement on discrete structures // Local search in combinatorial optimization. Chichester: Wiley, 1997. P. 57–89.

20. **Vredeveld T., Lenstra J. K.** On local search for the generalized graph coloring problem // Oper. Res. Lett. 2003. V. 31, N. 4. P. 28–34.

21. **Yannakakis M.** Computational complexity // Local search in combinatorial optimization. Chichester: Wiley, 1997. P. 19–55.

Адрес авторов:

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

Статья поступила
28 октября 2004 г.

Переработанный вариант —
20 октября 2005 г.