УДК 519.854

В.А. МИХАЙЛЮК

СЛОЖНОСТЬ РЕОПТИМИЗАЦИИ ЗАДАЧИ ВЫЧИСЛЕНИЯ ХРОМАТИЧЕСКОГО ЧИСЛА ГРАФА С ЗАДАННЫМ МНОЖЕСТВОМ ОПТИМАЛЬНЫХ РЕШЕНИЙ

Аннотация. Используются сведения, вводящие и сохраняющие разрыв. Показано, что для множественной реоптимизации задачи о вычислении хроматического числа графа с заданным экспоненциальным множеством оптимальных решений при вставке произвольной вершины с не более чем двумя ребрами, ей инцидентными, а также при удалении произвольной вершины со всеми инцидентными ей ребрами не существует полиномиально приближенной схемы (PTAS). Такой же результат имеет место для обычной реоптимизации.

Ключевые слова: множественная реоптимизация, сведения задач, вводящих и сохраняющих разрыв, *APX*-трудность, полиномиально приближенные схемы (PTAS).

ПОСТАНОВКА ЗАДАЧИ

Как известно, нахождение оптимальных решений NP-трудных задач комбинаторной оптимизации является вычислительно сложной задачей, и предлагаются различные подходы (эвристики и другие приближенные алгоритмы) для вычисления «хороших» (не обязательно оптимальных) решений. Однако даже нахождение приближенных решений может оказаться труднорешаемой (NP-трудной) задачей.

Один из подходов, известных для решения NP-трудных задач, заключается в рассмотрении экземпляров задач (задач с заданными значениями входных параметров) не только как изолированных задач, но и в использовании добавочных знаний о подобных экземплярах (полученных при попытках решений подобных экземпляров). В данном случае возникает такая идея: исходя из оптимального (или близкого к нему) решения экземпляра задачи использовать эту информацию для нахождения решений (экземпляров задач), полученных в результате незначительных локальных модификаций исходного экземпляра.

Концепция реоптимизации формально описывает этот подход.

Рассмотрим некоторую задачу дискретной оптимизации Q с заданными значениями входных параметров (пусть задан экземпляр I задачи Q). Если значение входных параметров определенным образом изменено, получаем экземпляр I' задачи Q. Возникает вопрос: каким образом, зная оптимальное решение x^* экземпляра I задачи Q, найти оптимальное решение экземпляра I'? Понятие реоптимизации заключается в следующем: как эффективно использовать знания об оптимальном решении x^* экземпляра I для нахождения точного или приближенного решения экземпляра I'? Цель реоптимизации при использовании приближенных методов — применение знаний о решении начального экземпляра I для выполнении одного из условий: повышения качества приближения (аппроксимационного отношения) I', создания более эффективного (по быстродействию) алгоритма определения оптимального либо близкого к нему решения I' или выполнения предыдущих двух условий. Если в результате реоптимизации удается улучшить качество приближения (аппроксимационного отношения), то будет ли оно улучшаемым (пороговым) в классе параметрических приближенных алгоритмов некоторого класса (например, полиномиальных)?

В этом контексте качество приближения измеряется как отношение значения вычисляемого решения к значению оптимального. Иными словами, для минимизационных задач U и алгоритма A для U аппроксимационное отношение $\rho_A(n)$ для введенных данных размера n определяется как $\max_I \{A(I)/Opt(I) : I$ — экземпляр U размера $n\}$. Если $\rho_A(n)$ можно оценить сверху как константу, то считают, что A — константный приближенный алгоритм для U. Класс APX содержит все задачи, для которых существует константный приближенный алгоритм. Полиномиально приближенной схемой (Polinomial Time Approximation Scheme — PTAS) является алгоритм, который для заданного $\varepsilon > 0$ вычисляет $(1+\varepsilon)$-приближенное решение за полиномиальное время относительно n (но, возможно, экспоненциальное по $1/\varepsilon$). Задачи, которые APX-трудны, являются наиболее сложными задачами в классе APX. Для таких задач не существует PTAS, если $P \neq NP$.

Впервые понятие реоптимизации появилось в середине 1980-х годов при решении таких полиномиальных задач оптимизации, как минимальное остовное дерево [1] и кратчайший путь в графе [2]. Суть состояла в поддержке и использовании оптимального решения задачи при модификации входных данных (вставка или удаление ребра, изменение веса ребра) для достижения результата (оптимального решения) с меньшими вычислительными затратами. Затем исследовались многие классические NP-полные (NP-трудные) задачи дискретной оптимизации (задача коммивояжера, задача о минимальном дереве Штейнера, задача о покрытии, задача о ранце и т.д.). (Обзор известных результатов по реоптимизации можно найти в [3–5].)

В работах [6, 7] предлагается обобщение реоптимизационного подхода. Считается, что задано не единственное оптимальное решение исходной задачи, а даже некоторое их множество (или множество решений, близких к оптимальным). Возникает вопрос: может ли это способствовать улучшению качества реоптимизации? В работах [6, 7] дается отрицательный ответ относительно задачи о минимальном дереве Штейнера и задачи о коммивояжере.

Данная статья посвящена исследованию этого вопроса для задачи о минимальной раскраске графа.

ПРЕДВАРИТЕЛЬНЫЕ ОПРЕДЕЛЕНИЯ И ОБОЗНАЧЕНИЯ

Приведем некоторые понятия, необходимые для дальнейшего изложения [3, 8].

Определение 1. NP-оптимизационная (класс NPO) задача П определяется как четверка (E, Sol, m, opt) такая, что:

1) E — множество экземпляров задачи П, распознаваемое за полиномиальное время;

2) для данного $I \in E$ имеем $Sol(I)$ — множество допустимых решений I; для каждого $S \in Sol(I)$ $|S|$ (размер S) полиномиален по $|I|$ (размер I); для любого данного I и любого S (полиномиального по $|I|$) за полиномиальное время можно определить $S \in Sol(I)$;

3) для данных $I \in E$ и $S \in Sol(I)$ имеем $m(I, S)$ — значение (числовое) S; m — полиномиально вычислимая целевая функция;

4) $opt \in \{\min, \max\}$ — тип оптимизационной задачи.

Для данной NPO-задачи $\Pi = \{E, Sol, m, opt\}$ оптимальное решение экземпляра I задачи П будем обозначать $S^*(I)$, а его величину $m(I, S^*(I))$ — оптимальным решением $opt(I)$.

Определение 2. Для данной NPO-задачи $\Pi = (E, Sol, m, opt)$ приближенный (аппроксимационный) алгоритм A для данного экземпляра I задачи П выдает допустимое решение $S \in Sol(I)$.

Если A выполняется за полиномиальное время относительно $|I|$, то этот алгоритм называется полиномиальным приближенным алгоритмом для П.

Качество приближенного алгоритма обычно оценивается как отношение $\rho_A(I)$ (аппроксимационное отношение) между значением приближенного решения $m(I, A(I))$ и значением оптимального решения $opt(I)$. Таким образом, для минимизационных задач аппроксимационное отношение находится в пределе $[1, \infty)$, для максимизационных — в пределе $[0, 1]$.

Относительно качества приближенных алгоритмов NPO-задачи классифицируют следующим образом.

Определение 3. NPO-задача П принадлежит классу *APX*, если существуют полиномиально приближенный алгоритм A и рациональное число r такие, что для данного $I \in \Pi$ имеем $\rho_A(I) \le r$ (соответственно $\rho_A(I) \ge r$), при этом П — минимизационная (соответственно максимизационная) задача. В этом случае A называется r-приближенным (аппроксимационным) алгоритмом (а задача П — r-аппроксимируема алгоритмом A).

Для отдельных задач из класса *APX* можно ввести более сильную форму аппроксимационности. Для таких задач и любого рационального $r > 1$ (или $r \in (0, 1)$ для максимизационных задач) существуют алгоритм A_r и подходящий полином p такой, что A_r — r-аппроксимационный (приближенный) алгоритм со временем, измеряемым как p от $|I|$. Семейство алгоритмов A_r (параметризованное с помощью r) называется полиномиально приближенной схемой (PTAS).

Определение 4. NPO-задача П принадлежит классу PTAS, если для любого рационального $\varepsilon > 0$ и любого экземпляра I задачи П существует полиномиально приближенная схема A_ε такая, что $\rho_{A_\varepsilon}(I) \le 1 + \varepsilon$ (соответственно $\rho_{A_\varepsilon}(I) \ge 1 - \varepsilon$) для П-минимизационной (соответственно максимизационной) задачи.

Если $P \ne NP$, то имеет место включение $\mathrm{PTAS} \subset APX \subset \mathrm{NPO}$.

Для экземпляра $I \in \Pi$ обозначим $c(I)$ значение целевой функции, которое находит алгоритм (или ожидаемое значение, если алгоритм случайный), $OPT(I)$ — значение оптимального решения.

Определение 5 (отношение аппроксимации). Полагают, что алгоритм A достигает отношения аппроксимации C-приближенного алгоритма, если для каждого экземпляра $I \in \Pi$ имеем $c(I) \le C \cdot OPT(I)$, где П — минимизационная задача.

Под эффективным алгоритмом для задачи П понимают алгоритм со временем работы не более чем полином от размерности входа. Для задачи П установлена верхняя оценка отношения аппроксимации C, если существует полиномиальный C-приближенный алгоритм для решения П. Для задачи П установлена нижняя оценка отношения аппроксимации d, если для произвольного $\varepsilon > 0$ не существует полиномиального приближенного алгоритма для П, на котором достигается отношение аппроксимации $d - \varepsilon$. Если задача П является NP-трудной или NP-полной, то, естественно, добавляется условие $P \ne NP$. Если $C = d$, то для задачи П установлен порог отношения аппроксимации. Соответствующий алгоритм называется оптимальным или пороговым (и отношение аппроксимации — оптимальное или пороговое).

Чтобы охарактеризовать нижние оценки отношения аппроксимации приближенных алгоритмов, вводится понятие сложности отношения аппроксимации.

Определение 6 (сложность, трудность аппроксимации). Назовем задачу α-сложной (α-трудной) для аппроксимации (приближения), если не существует приближенного полиномиального алгоритма с отношением аппроксимации меньше α при $P \ne NP$.

Таким образом, если задача П является NP-трудной или NP-полной, то установление для нее нижней оценки отношения аппроксимации c означает, что эта задача есть c-сложной (c-трудной) для приближения (аппроксимации). Поэтому установление нижних оценок отношения аппроксимации называют также сложностью (трудностью) аппроксимации (hardness of approximation) или неаппроксимированностью (innapproximability).

СВЕДЕНИЯ ЗАДАЧ, ВВОДЯЩИХ И СОХРАНЯЮЩИХ РАЗРЫВ. ВЛИЯНИЕ НА СЛОЖНОСТЬ АППРОКСИМАЦИИ

Введем понятия сведений задач, вводящих и сохраняющих разрыв [9]. Для определенности будем рассматривать задачи минимизации.

Определение 7. Предположим, что L является NP-трудной задачей распознавания свойств, а Π — минимизационная задача. Пусть g — функция, вычислимая за полиномиальное время, которая отображает да-экземпляры L в множество S_1 экземпляров задачи Π и нет-экземпляры L в множество S_2 экземпляров задачи Π. Предположим, что существует полиномиально вычислимая функция h такая, что

- для произвольного да-экземпляра L: $OPT(g(x)) \le h(g(x))$;
- для произвольного нет-экземпляра L: $OPT(g(x)) > \alpha(|x|) \cdot h(g(x))$,

где $OPT(y)$ — значение оптимального решения y задачи Π, а $|x|$ — длина (размер) x. Тогда g назовем сведением, вводящим разрыв от L к Π, а α — размером разрыва.

Лемма 1. Если $P \ne NP$ и для задачи Π существует сведение g, вводящее разрыв от L к Π (α — размер разрыва), то для Π не существует полиномиального алгоритма с отношением аппроксимации, не большим α.

Доказательство. Предположим, что для задачи Π существует полиномиальный β-приближенный алгоритм Alg с отношением аппроксимации $\beta \le \alpha$, $Alg(y)$ — значение алгоритма Alg на экземпляре $y \in \Pi$. Предположим, что x есть да-экземпляр L, тогда $Alg(y) = Alg(g(x)) \le \beta \cdot OPT(y) \le \alpha \cdot OPT(y) \le \alpha \cdot h(y)$. Для нет-экземпляра L получим $Alg(y) = Alg(g(x)) > OPT(g(x)) > \alpha \cdot h(g(x)) = \alpha \cdot h(y)$. Таким образом, существует возможность за полиномиальное время (с помощью алгоритма Alg) отличить да-экземпляры от нет-экземпляров NP-трудной задачи L; следовательно, $P = NP$. Лемма доказана.

Рассмотрим задачу раскраски графа (CG).

Условие. Даны граф $G = (V, E)$ и целое положительное число K $(K \le |V|)$.

Возникает вопрос: действительно ли граф G можно раскрасить K цветами? (Граф называется раскрашиваемым K цветами, если существует некоторая функция $f: V \to \{1, 2, \dots, K\}$, когда при $(u, v) \in E$ имеем $f(u) \ne f(v)$.)

Известно, что задача CG является NP-полной (а значит, и NP-трудной) для $K \ge 3$ целого [14]. С задачей CG сопоставим NPO-задачу о минимальной раскраске графа (Min_CG): нахождение минимального K, когда граф $G = (V, E)$ есть K-раскрашиваемым. Задачу Min_CG запишем в виде $G = (V, E, w)$, где w — целевая функция (число цветов), $w \to \min$.

Хроматическое число графа $G = (V, E)$, обозначаемое $\chi(G)$, есть решение оптимизационной задачи Min_CG. Используя лемму 1, а также то, что задача CG для $K = 3$ является NP-полной (следовательно, NP-трудной), получаем следующий результат по сложности (трудности) аппроксимации оптимизационной задачи вычисления хроматического числа заданного графа G.

Лемма 2. Если $P \ne NP$, то для любого $\varepsilon \ge 0$ не существует $(4/3 - \varepsilon)$-приближенного полиномиального алгоритма для вычисления хроматического числа графа.

Доказательство. Построим сведение, вводящее разрыв от задачи CG при $K = 3$ к задаче вычисления хроматического числа (Min_CG). Пусть S_1 — множество графов с хроматическим числом, не превосходящим трех, а S_2 — с хроматическим числом, не меньшим, чем 4. Тогда да-экземпляры отображаются на S_1, а нет-экземпляры — на S_2. Согласно лемме 1 для *Min_ CG* не существует полиномиального алгоритма с отношением аппроксимации, не большим 4/3.

Если существует сведение, вводящее разрыв от L к задаче Π_1, то можно определить сложность (трудность) аппроксимации для другой задачи Π_2 введением сведения, сохраняющего разрыв от Π_1 к Π_2.

Определение 8. Сведение, сохраняющее разрыв (функция F) от Π_1 к Π_2, задается четырьмя параметрами (функциями): f_1, α, f_2, β. Для заданного экземпляра $x \in \Pi_1$ сведение вычисляет за полиномиальное время экземпляр $y \in \Pi_2$ ($y = F(x)$) такой, что

$$OPT(x) \le f_1(x) \Rightarrow OPT(y) \le f_2(y),$$

$$OPT(x) > \alpha(|x|) \cdot f_1(x) \Rightarrow OPT(y) > \beta(|y|) \cdot f_2(y).$$

Лемма 3. Пусть g — сведение, вводящее разрыв от L к Π_1 ($\alpha(|x|)$ — размер разрыва), а F — сведение, сохраняющее разрыв от Π_1 к Π_2 с параметрами f_1, α, f_2, β. Тогда если $P \ne NP$, то для задачи Π_2 не существует полиномиального алгоритма с отношением аппроксимации, не большим β.

Доказательство. Рассмотрим композицию сведения, вводящего разрыв для $\Pi_1(g)$, и сведения, сохраняющего разрыв для $\Pi_2(F)$: $g \circ F = F(g(x))$. Композиция $g \circ F$ задает сведение, вводящее разрыв от L к Π_2 (β-разрыв). Осталось применить лемму 1 к задаче Π_2.

ОБ ОПТИМАЛЬНОЙ РАСКРАСКЕ ОДНОГО ГРАФА

Построим граф H_i, представляющий цепь-(K_3, i), где K_3 — полный 3-вершинный граф; i — натуральное число. Граф H_i состоит из i звеньев и строится индуктивно. Граф H_1, представляющий цепь-$(K_3, 1)$, — это граф $K_3 = (V_{K_3}, E_{K_3})$, где вершины $V_{K_3} = \{a_i, b_i, c_i\}$ назовем соответственно вершинами типа a, типа b, типа c (в каждом последующем звене); ребра $E_{K_3} = \{(a_i, b_i); (b_i, c_i); (c_i, a_i)\}$. Пусть задан граф H_i, представляющий цепь-(K_3, i) и имеющий i звеньев (K_3). Граф H_{i+1} = цепь-$(K_3, i+1)$ образуется из графа H_i = цепь-(K_3, i), к которому добавляется граф K_3 (($i+1$)-е звено) вместе с тремя ребрами (они соединяют вершины типа a, типа b, типа c ($i+1$)-го звена K_3 с соответствующими вершинами типа a, типа b, типа c последнего звена графа H_i = цепь-(K_3, i)). Будем строить оптимальные раскраски графа H_i = цепь-(K_3, i).

Лемма 4. Граф H_i = цепь-(K_3, i) можно раскрасить тремя цветами, и существует 2^i оптимальных раскрасок тремя цветами (оптимальных решений *Min_ CG*) этого графа.

Доказательство. Поскольку граф H_i = цепь-(K_3, i) содержит в качестве подграфа хотя бы один граф K_3, раскрасить этот граф, меньше, чем тремя цветами, невозможно. Поэтому оптимальная раскраска использует не менее трех цветов. Граф H_i будем раскрашивать тремя цветами, перенумеровав их как 1, 2, 3. Раскраску графа K_3 будем задавать цветами в порядке следования вершин типа b, типа c, типа a. Например, раскраска графа K_3 как 123 означает, что вершина типа b имеет цвет 1, типа c — цвет 2, типа a — цвет 3. Будем считать, что для раскраски графа H_1 цвет вершины типа b — фиксированный (и заранее заданный). Раскраску графа H_i = цепь-(K_3, i) проведем индуктивно. Граф H_1 можно раскрасить двумя способами при фиксированном цвете вершины типа b. Таким образом, если цвет b есть 1, то правильная раскраска имеет вид $\{123, 132\}$; если цвет b есть 2, то правильная раскраска — $\{213, 231\}$; если цвет b есть 3, то правильная раскраска — $\{312, 321\}$. Пусть граф H_i можно раскрасить 2^i способами ($i = 1, 2, \ldots$).

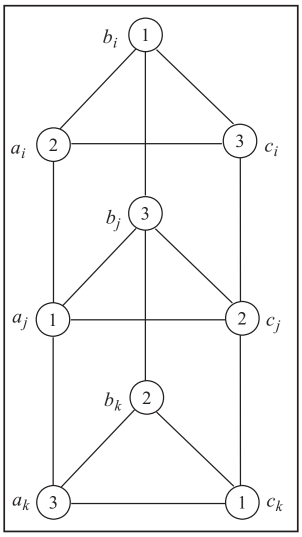

Рис. 1. Оптимальная раскраска графа $H_3 =$ цепь-$(K_3,3)$ (из восьми возможных)

Покажем, что граф H_{i+1} можно правильно раскрасить $2 \cdot 2^i = 2^{i+1}$ способами. Простым перебором можно показать, что если i-е звено графа H_i правильно раскрашено как bca (где $a, b, c \in \{1, 2, 3\}$), то при добавлении $(i+1)$-го звена K_3 его можно правильно раскрасить двумя способами: cab, abc и записать как

$$bca \to cab, abc. \qquad (1)$$

Раскраски (1) составляют такое множество раскрасок:

$$\{123 \to 231, 312;\ 132 \to 321, 213;\ 213 \to 132, 321;$$

$$231 \to 312, 123;\ 312 \to 123, 231;\ 321 \to 213, 132\}. \quad (2)$$

Таким образом, при добавлении $(i+1)$-го звена число правильных раскрасок увеличится ровно вдвое: $2 \cdot 2^i = 2^{i+1}$. Лемма доказана.

На рис. 1 показана оптимальная раскраска графа $H_3 =$ цепь-$(K_3, 3)$ из возможных восьми оптимальных раскрасок. Данная раскраска соответствует подмножеству раскрасок $\{132 \to 321, 213;\ 321 \to 213, 132\}$ из (2).

ПОНЯТИЕ МНОЖЕСТВЕННОЙ РЕОПТИМИЗАЦИИ

Известно, что задача о коммивояжере TSP (Travelling Salesman Problem) является задачей нахождения кратчайшего гамильтонова пути в реберно-взвешенном полном графе. Если $P \neq NP$, то для этой задачи не существует полиномиального приближенного алгоритма с аппроксимационным отношением 2^n в n-вершинном графе с произвольными весами ребер [10]. Пусть $Inc\text{-}Edge\text{-}Reopt\text{-}TSP$ — реоптимизационная версия задачи TSP, в которой увеличился вес одного ребра. Как и в случае TSP, имеет место результат.

Теорема 1 [11, 12]. Является NP-трудным аппроксимировать задачу $Inc\text{-}Edge\text{-}Reopt\text{-}TSP$ с аппроксимационным отношением, которое представляет полином от размерности задачи.

Обобщение реоптимизации (которую будем называть множественной реоптимизацией) состоит в том, что известным является не единственное оптимальное решение «старого» экземпляра, а множество оптимальных решений или близких к ним (возможно, с экспоненциальным числом), доступ к которым не ограничен [6, 7].

Определение 9. Представим задачу $Inc\text{-}Edge\text{-}Reopt_{ALL}\text{-}TSP$ как задачу, заданную полным графом $G = (V, E)$, с двумя заданными весовыми функциями: c_{old} и c_{new} такими, что (G, c_{old}) и (G, c_{new}) являются допустимыми экземплярами для TSP, и такими, что c_{old} и c_{new} совпадают всеми ребрами, кроме ребра $e_{\text{change}} \in E$, где $c_{\text{old}}(e_{\text{change}}) < c_{\text{new}}(e_{\text{change}})$. Более того, известны все оптимальные TSP-решения (G, c_{old}). Необходимо вычислить оптимальное TSP-решение для (G, c_{new}).

Теорема 2 [7]. В предположении $P \neq NP$ не существует полиномиального приближенного алгоритма с отношением аппроксимации 2^n для задачи $Inc\text{-}Edge\text{-}Reopt_{ALL}\text{-}TSP$, где n — число вершин экземпляра.

Таким образом, при сравнении теорем 1 и 2 приходим к выводу, что в данном случае множественная реоптимизация не способствует улучшению качества приближения.

Задача о минимальном дереве Штейнера (STP) представляется следующей оптимизационной задачей: для заданного неориентированного полного графа $G=(V, E)$, метрической весовой функции ребер $c: E \to Q^{+}$ (Q^{+} — множество положительных рациональных чисел) и множества $S \subseteq V$ терминалов необходимо найти минимальное по весу поддерево G, которое содержит все терминалы (и, возможно, некоторые не терминалы). Такое поддерево, содержащее все терминалы, называется минимальным деревом Штейнера для G.

Определение 10. Реоптимизационная задача для минимального дерева Штейнера с локальной модификацией lm, например добавлением терминальной вершины с ребрами к графу (сокращенно $Reopt$-STP-lm), является такой оптимизационной задачей. Ввод состоит из STP-экземпляра (G, S, c), оптимального дерева Штейнера T_{old} для него и локально модифицированного STP-экземпляра (G', S', c') по отношению к lm. Необходимо найти минимальное дерево Штейнера для экземпляра (G', S', c').

Рассмотрим модификацию $lm = AddTerm$ (добавление терминальной вершины). Показано [13], что $Reopt$-STP-$AddTerm$ является APX-трудной задачей. Это означает, что не существует PTAS для задачи при $P \neq NP$.

Определение 11. Минимальная реоптимизационная задача дерева Штейнера с k решениями (сокращенно k-Sol-$Reopt$-STP-lm) представляет следующую задачу. Ввод состоит из STP-экземпляра (G, S, c), последовательности k наилучших деревьев Штейнера $T_{\text{old}}^{(1)}, \ldots, T_{\text{old}}^{(k)}$ для него и локально модифицированного экземпляра (G', S', c') по отношению к lm. Задача состоит в нахождении минимального дерева Штейнера для (G', S', c').

С помощью AP-сведений, сохраняющих аппроксимационное отношение, установлен следующий факт.

Теорема 3 [6]. Задача n^{n-2}-Sol-$Reopt$-$AddTerm$ является APX-трудной.

Таким образом, и в этом случае множественная реоптимизация не способствует улучшению качества приближения.

МНОЖЕСТВЕННАЯ РЕОПТИМИЗАЦИЯ ЗАДАЧИ О МИНИМАЛЬНОЙ РАСКРАСКЕ ГРАФА

Определение 12. k-множественная реоптимизация Min_CG с локальной модификацией lm (сокращенно k-Set-$Reopt$-Min_CG-lm) представляет следующую оптимизационную задачу. Ввод состоит из экземпляра $G=(V_{\text{old}}, E_{\text{old}}, w_{\text{old}})$ задачи Min_CG с k оптимальными решениями и локально модифицированного экземпляра $G=(V_{\text{new}}, E_{\text{new}}, w)$ по отношению к lm. Необходимо найти минимальное число красок w_{new} для $G=(V_{\text{new}}, E_{\text{new}}, w)$.

Возможен вариант $k = k(n)$, где n — число вершин графа. В качестве модификации lm будем рассматривать варианты: а) $lm = AddVer$ (вставка произвольной вершины с не более чем двумя ребрами, инцидентными ей); б) $lm = DelVer$ (удаление любой вершины со всеми инцидентными ей ребрами).

Будем считать, что V_{old} ($|V_{\text{old}}| = n$) содержит хотя бы один граф K_3 в качестве подграфа (для корректной реализации предлагаемой конструкции).

Рассмотрим следующую реоптимизационную задачу. Пусть граф $G_1 = (V_1, E_1, w)$ получен из графа $G = (V, E, w)$ с использованием вставки произвольной вершины с не более чем двумя ребрами, инцидентными ей. Задача реоптимизации G_1 заключается в нахождении ее решения (приближенного) в предположении, что оптимальное решение G со значением $w_{\min}$ известно.

Теорема 4 [8]. Если $P \neq NP$, то для реоптимизации задачи $G_1 = (V_1, E_1, w)$ не существует PTAS.

Пусть $G_2 = (V_2, E_2, w)$ получается из графа $G = (V, E, w)$ удалением произвольной вершины со всеми инцидентными ей ребрами.

Теорема 5 [8]. Если $P \neq NP$, то для реоптимизации задачи $G_2 = (V_2, E_2, w)$ не существует PTAS.

Теорема 6. Если $P \neq NP$, то для задачи 2^n-*Set*-*Reopt*-*Min*_*CG*-*AddVer* не существует PTAS.

Доказательство. Пусть L — задача CG при $K = 3$, а Π_1 — реоптимизация $G_1 = (V_1, E_1, w)$. Построим сведение g, вводящее разрыв от L к Π_1 (фактически это сведение рассматривалось в [8, доказательство теоремы 4]). Пусть $x \in L$ — экземпляр задачи L, $g(x)$ — преобразование (полиномиальное), состоящее в отображении V на V_1 и E на E_1, которое задается добавлением вершины υ к V (получим V_1) с не более чем двумя ребрами, инцидентными ей (получим E_1). Если $x \in L$ есть да-экземпляр, то это является 3-раскраской графа G. Покажем, что она будет 3-раскраской графа G_1. Действительно, исходя из 3-раскраски G, можно раскрасить вершину υ графа G_1 (добавлены не более чем два инцидентных ей ребра) цветом, отличным от двух цветов (если добавлены ровно два ребра), которыми раскрашены в G соответствующие вершины (соединенные ребром). При добавлении одного ребра вершина υ раскрашивается цветом, отличным от цвета соседней к υ вершины. Таким образом, получим 3-раскраску графа G_1, т.е. экземпляр $y = g(x) \in G_1$ со значением $w \leq h(y) = 3$. Нет-экземпляр задачи L — это раскраска графа G не менее четырьмя цветами, что согласно сказанному выше дает возможность раскрасить G_1 также не менее чем четырьмя цветами, т.е. получим экземпляр $y = g(x) \in \Pi_1$ со значением $w \geq \alpha(|x|) \cdot h(y) = 4$, откуда $\alpha(|x|) = 4/3$. Таким образом, получим сведение, вводящее разрыв $g(x)$ от L к Π_1; $\alpha(|x|) = 4/3$ — размер разрыва.

Построим сведение F, которе сохраняет разрыв от Π_1 к задаче $\Pi_2 = 2^n$-*Set*-*Reopt*-*Min*_*CG*-*AddVer*.

Суть конструкции состоит в замене произвольной вершины u (только не новой добавленной) экземпляра $x \in \Pi_1$ графом $H_n =$ цепь-(K_3, n), который имеет $3n$ вершин. Вершина типа b первого звена H_n получит такой же цвет, который она имела в оптимальном решении со значением $w_{\min}$. Остальные два цвета раскраски для H_n выбираются отличными от цвета u, при этом они присутствуют в раскраске G_1. Таким образом, для $x \in \Pi_1$ $F(x)$ — полиномиальное преобразование, состоящее в отображении V_1 на V_3 (добавлена $3n-1$ вершина к V_1 соответствующим образом) и E_1 на E_3 (добавлением соответствующего числа ребер к E_1). Тогда множество оптимальных решений для $G_3 = (V_3, E_3, w)$ может быть вычислено при замене раскраски u множеством оптимальных раскрасок графа H_n (число которых составляет 2^n). Таким образом, общая структура графа не изменяется описанным преобразованием и знание всего экспоненциального числа раскрасок (2^n) не способствует решению нового экземпляра. Согласно определению 8 для $x \in \Pi_1$ имеем $y = F(x) \in \Pi_2$, т.е. $f_1(x) \equiv f_2(y) \equiv 3$, $\alpha(|x|) \equiv \beta(|y|) \equiv 4/3$, на основании свойств построенного преобразования $F(x)$, которые аналогичны свойствам $g(x)$.

Применяя к сведениям g и F лемму 3, получаем следующее: если $P \neq NP$, то для задачи Π_2 не существует полиномиального алгоритма с отношением аппроксимации, не большим $\beta(|y|) = 4/3$, или не существует PTAS.

Теорема 7. Если $P \neq NP$, то для задачи 2^n-*Set*-*Reopt*-*Min*_*CG*-*DelVer* не существует PTAS.

Доказательство аналогично предыдущему и следует из того факта, что из 3-раскраски графа $G=(V, E, w)$ вытекает 3-раскраска графа $G_3=(V_3, E_3, w)$ и из не менее чем 4-раскраски G следует не менее чем 4-раскраска графа G_3.

ЗАКЛЮЧЕНИЕ

В настоящей статье показано, что для множественной реоптимизации задачи о вычислении хроматического числа графа не существует PTAS при вставке произвольной вершины с не более чем двумя ребрами, ей инцидентными, а также при удалении произвольной вершины со всеми ребрами, ей инцидентными, если задано множество из 2^n оптимальных раскрасок исходной задачи, где n — число вершин в графе.

Подобные результаты имеют место (не улучшается качество приближения — аппроксимационное отношение в сравнении с обычной реоптимизацией) для множественной реоптимизации задачи о минимальном дереве Штейнера с добавлением терминальной вершины [6] и для задачи о коммивояжере при увеличении веса одного ребра [7]. Представляют интерес подобные и отличающиеся от них результаты, полученные с использованием множественной реоптимизации при локальных изменениях данных для других классов дискретных оптимизационных задач. В частности, автору неизвестны результаты множественной реоптимизации, которая имела бы преимущество в качестве приближения по сравнению с обычной реоптимизацией.

СПИСОК ЛИТЕРАТУРЫ

1. Frederickson G.N. Data structures for on-line updating of minimum spanning trees with application // SIAM Journal on Computing. — 1985. — **14**, N 4. — P. 781–798.
2. Rohnert H. A dynamization of the all-pairs least cost problems // Proc. 2nd Symp. on Theoretical Aspects of Computer Science, 1985. — P. 279–286.
3. Ausiello G., Bonifaci V., Escoffier B. Complexity and approximation in reoptimization // Computability in Context: Computation and Logic in the Real World (S.B. Cooper, A. Sorbi (Eds.). — London: Imperial College Press, 2011. — P. 101–130.
4. Bockenhauer H.-J., Hromkovic J., Momke T., Widmayer P. On the hardness of reoptimization / V. Geffert, J. Karhumaki, A. Bertoni, etc. (Eds.) // SOFSEM, Lecture Notes in Computer Science. — 2008. — **4910.** — P. 50–65.
5. Boria N., Paschos V.Th. A survey on combinatorial optimization in dynamic environments // RAIRO — Operations Research. — 2011. — **45.** — P. 241–.294.
6. Bockenhauer H.-J., Hromkovic J., Sprock A. On the hardness of reoptimization with multiple given solutions // Fundamenta Informaticae. — 2011. — **110.** — P. 59–76.
7. Bockenhauer H.-J., Hromkovic J., Sprock A. Knowing all optimal solutions does not help for TSP reoptimization / J. Kelemen and A. Kelemenova (Eds.) // Lecture Notes in Computer Science. — 2011. — **6610.** — P. 7–15.
8. Михайлюк В.А. К вопросу о существовании полиномиально приближенных схем для реоптимизации дискретных задач оптимизации // Кибернетика и системный анализ. — 2011. — № 3. — С. 42–50.
9. Vazirani V.V. Approximation algorithms. — Berlin: Springer, 2001. — 380 p.
10. Sahni S., Gonzales T.F. P-complete approximation problems // Journal of the ACM. — 1976. — **23**, N 3. — P. 555–565.

11. B o c k e n h a u e r H.-J., F o r l i z z i L., H r o m k o v i c J., K n e i s J., K u p k e J., P r o i e t t i G., W i d m a y e r P. Reusing optimal TSP solutions for locally modified input instances (extended abstract) // G. Navarro, L.E. Bertissi, Y. Kohayakava (Eds.) // Proc. of the 4th IFIP International Conference on Theoretical Computer Science (TCS). — 2006. — IFIP, Springer. — **209.** — P. 251–270.

12. B o c k e n h a u e r H.-J., F o r l i z z i L., H r o m k o v i c J., K n e i s J., K u p k e J., P r o i e t t i G., W i d m a y e r P. On the approximability of TSP on local modifications of optimality solved instances // Algorithmic Operations Research. — 2007. — **2**, N 2. — P. 83–93.

13. B o c k e n h a u e r H.-J., F r e i e r m u t h K., H r o m k o v i c J., M o m k e T., S p r o c k A., S t e f f e n B. Steiner tree reoptimization in graphs with sharpened triangle inequality // Journal of Discrete Algorithms. — 2012. — **11.** — P. 73–86.

14. Г э р и М., Д ж о н с о н Д. Вычислительные машины и труднорешаемые задачи. — М.: Мир, 1982. — 416 с.

Надійшла до редакції 05.08.2015

В.О. Михайлюк

СКЛАДНІСТЬ РЕОПТИМІЗАЦІЇ ЗАДАЧІ ОБЧИСЛЕННЯ ХРОМАТИЧНОГО ЧИСЛА ГРАФА ІЗ ЗАДАНОЮ МНОЖИНОЮ ОПТИМАЛЬНИХ РОЗВ'ЯЗКІВ

Анотація. Використано зведення, що вводять і зберігають розрив. Показано, що для множинної реоптимізації задачі про обчислення хроматичного числа графа із заданою експоненціальною множиною оптимальних розв'язків при уставленні довільної вершини з не більш ніж двома ребрами, їй інцидентними, а також при видаленні довільної вершини з усіма інцидентними їй ребрами не існує поліноміально наближеної схеми (PTAS). Такий же результат має місце для звичайної реоптимізації.

Ключові слова: множинна реоптимізація; зведення задач, які вводять і зберігають розрив; *APX*-складність, поліноміально наближені схеми (PTAS).

V.A. Mikhailyuk

HARDNESS OF REOPTIMIZATION OF THE PROBLEM OF CALCULATING THE CHROMATIC NUMBER OF A GRAPH WITH A GIVEN SET OF OPTIMAL SOLUTIONS

Abstract. The author uses gap-introducing and gap-preserving reductions and shows that for multiple reoptimization of the problem of calculating the chromatic number of a graph with a given exponential set of optimal solutions, when an arbitrary vertex with no more than two edges incident to it is inserted as well as when any vertex with all incident edges is deleted, polynomial time approximation scheme (PTAS) does not exist. The same result holds for ordinary reoptimization..

Keywords: multiple reoptimization, gap-introducing (gap-preserving) reductions, *APX*-difficulty, polynomial time approximation schemes (PTAS).

Михайлюк Виктор Алексеевич,
доктор физ.-мат. наук, доцент, заведующий кафедрой Восточноевропейского национального университета имени Леси Украинки, Луцк, e-mail: mikhailyukvictor2@gmail.com.