УДК 519.8

П.И. СТЕЦЮК, А.П. ЛИХОВИД

О ЛП-ОРИЕНТИРОВАННЫХ ВЕРХНИХ ОЦЕНКАХ ДЛЯ ВЗВЕШЕННОГО ЧИСЛА УСТОЙЧИВОСТИ ГРАФА

Ключевые слова: *взвешенное число устойчивости графа, многогранник устойчивых множеств, ЛП-ориентированная верхняя оценка, t-совершенные графы, p-колесо, W_p-совершенные графы.*

ВВЕДЕНИЕ

Задача о максимальном независимом (устойчивом) множестве вершин графа является одной из центральных в теории графов. Она имеет много важных приложений, связанных с выбором в графе экстремальных множеств вершин с заданными свойствами. Так, например, непосредственное приложение этой задачи связано с нахождением максимального объема помехоустойчивых кодов (кодов, корректирующих ошибки при передаче информации). К задаче о максимальном независимом множестве вершин графа сводятся задача о максимальной клике графа и задача о максимальной k-клике графа. Последние могут быть использованы при нахождении взаимосвязанных подмножеств при информационном анализе социологических, биологических, телекоммуникационных и других массивов данных. Такие подмножества в социологических данных могут характеризовать родственные, криминальные или профессиональные связи; в телекоммуникациях — группы абонентов, которые часто общаются; в биологических данных нахождение таких подмножеств может означать наличие причинно-следственных связей при функционировании отдельных частей живого организма. С помощью задачи о максимальном независимом множестве вершин графа можно представить и классическую задачу о раскраске вершин неориентированного графа k-красками.

Задача о максимальном независимом множестве вершин графа для произвольного графа является *NP*-трудной. Более того, в общем случае не найдено полиномиального алгоритма, который бы гарантировал для нее сколь угодно большую фиксированную относительную погрешность по оптимальному значению целевой функции (его называют числом устойчивости графа). Поэтому актуален поиск верхних оценок, достаточно хорошо аппроксимирующих сверху число устойчивости графа.

В работе обсуждаются верхние оценки для взвешенного числа устойчивости графа, частным случаем которого является обычное число устойчивости графа. Они базируются на аппроксимации многогранника устойчивых множеств с помощью линейных неравенств для нечетных циклов и p-колес в графе. Алгоритмы нахождения обсуждаемых верхних оценок используют решение задачи линейного программирования с конечным числом неравенств, при построении которых используется алгоритм Дейкстры для нахождения кратчайших путей. Практическая эффективность верхних оценок для числа устойчивости графа подтверждается результатами тестовых экспериментов для графов с несколькими сотнями и тысячами вершин.

ПОСТАНОВКА ЗАДАЧИ И ЕЕ СЛОЖНОСТЬ

Пусть $G=(V,E)$ — неориентированный граф (не содержащий петель) с множеством вершин $V(G)$ и множеством ребер $E(G)$. Для каждой вершины $i \in V(G)$ задан положительный вес w_i. Подмножество вершин $S \subseteq V(G)$ называется устойчивым (или независимым) множеством графа G, если для любых $i, j \in S$ ребро

[1]Работа выполнена при частичной финансовой поддержке гранта UKM2-2812-KV-06 (CRDF Cooperative Grants Programm).

(i, j) не принадлежит $E(G)$. Взвешенное число устойчивости графа G определяется как $\alpha(G, w) = \max \sum_{i \in S} w_i$, где $S \subseteq V(G)$ — устойчивое множество. Подмножество S^*, на котором достигается $\alpha(G, w)$, называется максимальным взвешенным устойчивым (или независимым) множеством графа G. В частном случае, когда все веса вершин в графе равны единице, имеем обычное число устойчивости графа G, которое принято обозначать $\alpha(G)$. Число устойчивости $\alpha(G)$ характеризует мощность максимального по числу входящих в него вершин устойчивого множества в графе G.

В общем случае задачи нахождения $\alpha(G, w)$ и $\alpha(G)$ принадлежат к *NP*-трудным задачам [1]. Пусть $STAB(G)$ — многогранник устойчивых множеств (stable set polytope). Он является выпуклой оболочкой инцидентных векторов устойчивых множеств S в G и может быть представлен в следующем виде:

$$STAB(G) = \text{conv}\{x \in \{0, 1\}^{|V|} : x_i + x_j \le 1 \ \forall \ (i, j) \in E(G)\}. \tag{1}$$

Нахождение $\alpha(G, w)$ связано с задачей максимизации линейной функции на выпуклом многограннике $STAB(G)$, которая имеет вид

$$\alpha(G, w) = \max \sum_{i \in V(G)} w_i x_i, \quad x \in STAB(G). \tag{2}$$

Максимум в задаче (2) достигается в одной или нескольких вершинах многогранника $STAB(G)$. В общем случае многогранник $STAB(G)$ может иметь сложную структуру, из-за чего задача (2) принадлежит к *NP*-трудным. Поэтому теоретический и практический интерес представляет нахождение верхних оценок, достаточно хорошо аппроксимирующих сверху $\alpha(G, w)$.

Предметом обсуждения в статье являются верхние оценки, связанные с решением задач линейного программирования (ЛП-задач). Такие оценки назовем ЛП-ориентированными верхними оценками. В самом общем виде их можно описать следующим образом. Пусть $\mathcal{L}STAB(G)$ — некоторый многогранник, заданный с помощью системы линейных ограничений-неравенств, и пусть он аппроксимирует (сверху) многогранник $STAB(G)$. Тогда ЛП-ориентированная верхняя оценка $\alpha^*_{\mathcal{L}}(G, w)$ связана с решением ЛП-задачи

$$\alpha^*_{\mathcal{L}}(G, w) = \max \sum_{i \in V(G)} w_i x_i, \quad x \in \mathcal{L}STAB(G).$$

В силу того что многогранник $\mathcal{L}STAB(G)$ аппроксимирует (сверху) многогранник $STAB(G)$, имеем $\alpha^*_{\mathcal{L}}(G, w) \ge \alpha(G, w)$, и, следовательно, оценка $\alpha^*_{\mathcal{L}}(G, w)$ всегда является оценкой сверху для $\alpha(G, w)$. Свойства конкретной оценки $\alpha^*_{\mathcal{L}}(G, w)$ зависят от того, с помощью каких подмножеств линейных неравенств описывается многогранник $\mathcal{L}STAB(G)$.

ОЦЕНКА $\alpha^*_C(G, w)$ И АЛГОРИТМЫ LPCSTAB

Пусть C_{2k+1}, $k = 1, 2, \ldots$, — нечетный цикл в графе G (содержит нечетное количество вершин). Для многогранника $STAB(G)$ справедливы следующие классы линейных неравенств:

$$0 \le x_i \le 1 \quad \forall i \in V(G), \tag{3}$$

$$x_i + x_j \le 1 \quad \forall (i, j) \in E(G), \tag{4}$$

$$\sum_{i \in V(C_{2k+1})} x_i \le k \quad \forall C_{2k+1} \in G. \tag{5}$$

Семейство неравенств (3) принято называть вершинными ограничениями (vertix constraints), семейство неравенств (4) — реберными ограничениями (edge constraints), а семейство неравенств (5) — неравенствами нечетных циклов (odd-cycle constraints). Все вместе они определяют многогранник нечетных циклов (odd-cycle polytope)

$$CSTAB(G) = \{x \in R^{|V|}:\ x \text{ удовлетворяет (3)–(5)}\},$$

который аппроксимирует (сверху) многогранник $STAB(G)$.

Оценка $\alpha_C^*(G, w)$ связана с решением следующей ЛП-задачи:

$$\alpha_C^*(G, w) = \max \sum_{i \in V(G)} w_i x_i, \quad x \in CSTAB(G). \tag{6}$$

Для произвольного графа G всегда имеет место неравенство $\alpha_C^*(G, w) \geq \alpha(G, w)$. Если граф G принадлежит семейству t-совершенных графов, для которых $STAB(G) = CSTAB(G)$, то оценка $\alpha_C^*(G, w)$ является точной для $\alpha(G, w)$, т.е. $\alpha_C^*(G, w) = \alpha(G, w)$.

Задача (6) в общем случае может содержать неполиномиальное количество ограничений, что обусловлено наличием ограничений вида (5) для всех возможных нечетных циклов. Несмотря на это, задача (6) разрешима за полиномиальное время. В [2] описан полиномиальный алгоритм ее решения, базирующийся на использовании метода эллипсоидов и том факте, что для точки $\bar{x}$, удовлетворяющей ограничениям (3), (4), за полиномиальное время можно либо убедиться, что она удовлетворяет ограничениям (5) для каждого нечетного цикла, либо найти такой нечетный цикл, для которого ограничение вида (5) является максимально нарушенным. Метод из работы [2] имеет теоретическую ценность, однако он малопригоден для практического нахождения оценок $\alpha_C^*(G, w)$.

Если граф содержит несколько сотен вершин, для вычисления оценки $\alpha_C^*(G, w)$ эффективные алгоритмы можно реализовать на основе современных ЛП-программ, для которых решение ЛП-задач с сотнями переменных и десятками или сотнями тысяч ограничений не представляет особых проблем. Эти алгоритмы назовем ЛП-ориентированными алгоритмами нахождения $\alpha_C^*(G, w)$ и будем их ориентировать на решение ЛП-задач с конечным числом линейных ограничений. В основу ЛП-ориентированного алгоритма для нахождения $\alpha_C^*(G, w)$ положим полиномиальный алгоритм из [2] для нахождения максимального нарушенного ограничения, связанного с нечетным циклом, и ЛП-задачу

$$f_C^* = \max \sum_{i \in V} w_i x_i \tag{7}$$

при ограничениях:

$$0 \leq x_i \leq 1 \quad \forall i \in V(G), \tag{8}$$

$$x_i + x_j \leq 1 \quad \forall (i, j) \in E(G), \tag{9}$$

$$\sum_{i \in V(C_{2k+1})} x_i \leq k \quad \forall C_{2k+1} \in C_{\text{odd}} \in G. \tag{10}$$

Здесь C_{odd} — конечное множество нечетных циклов (возможно, пустое).

Тогда ЛП-ориентированный алгоритм для нахождения $\alpha_C^*(G, w)$ (назовем его LPCSTABa) состоит в следующем.

Начальная установка. Положим itn = 0 и $C_0 = \varnothing$ (множество нечетных циклов пусто). Перейдем к шагу 1.

Шаг 1. Имеем множество нечетных циклов C_{itn}. Положим $C_{\text{odd}} = C_{\text{itn}}$ и, решив ЛП-задачу (7)–(10), найдем f_C^* и $x^* = (x_1^*, \dots, x_{|V|}^*)$.

Шаг 2. Построим взвешенный неориентированный граф $G' = (V', E')$, где множество вершин $V'(G')$ включает $2|V|$ вершин (вершина i и ее копия i' для всех $i \in V(G)$). Множество ребер $E'(G')$ состоит из ребер, соединяющих все те пары вершин (i, j') и (i', j), для которых пара вершин i и j соединена ребром (i, j) в графе G. Присвоим ребрам (i, j') и (i', j) один и тот же положительный вес (длина ребра), равный $1 - x_i^* - x_j^*$.

Шаг 3. Рассмотрим граф G' как ориентированный (ребру (i, j') соответствуют дуги (i, j') и (j', i) с длинами, равными длине ребра (i, j'), т.е. $l(i, j')$) и найдем для каждой вершины $i \in V$ кратчайший путь, который начинается в вершине i и заканчивается в вершине i' (копии вершины i). Число таких кратчайших путей равно $|V|$. Пусть длины найденных кратчайших путей равны l_i, $i = 1, \dots, |V|$.

Шаг 4. Среди найденных $|V|$ кратчайших путей выберем кратчайший путь с минимальной (наименьшей) длиной $l^* = \min\limits_{i=1,2,\dots,|V|} (l_i)$ (не обязательно единственный). Если длина l^* больше либо равна единице, то $\alpha_C^*(G, w) = f_C^*$ и останов (достаточное условие того, что точка x^* удовлетворяет ограничениям в форме (5) для всех нечетных циклов). Иначе переходим к шагу 5.

Шаг 5. Полагаем $C' = C_{2k+1}$, где C_{2k+1} — нечетный цикл, соответствующий вершинам, через которые проходит кратчайший путь минимальной длины, с учетом замены вершин «i'» вершинами «i». Нечетный цикл C' определяет ограничение в форме (5), которое в точке x^* является максимально нарушенным (оно не обязательно единственное). Перейдем к шагу 6.

Шаг 6. Положим $C_{\text{itn}+1} = C_{\text{itn}} \cup C'$ и itn = itn+1. Перейдем к шагу 1.

Одна итерация алгоритма LPCSTABa требует решения ЛП-задачи (шаг 1) и нахождения $|V|$ раз кратчайшего пути (шаг 3). Если граф G содержит порядка нескольких сотен вершин, то добавление к ЛП-задаче ста нечетных циклов равносильно ста итерациям алгоритма и может потребовать значительных затрат по времени. Поэтому алгоритм LPCSTABa легко усовершенствовать, если использовать информацию трудоемкого по вычислениям шага 3 и добавить на каждой итерации все те нечетные циклы, для которых ограничение в форме (5) нарушено. Для модифицированного ЛП-ориентированного алгоритма нахождения $\alpha_C^*(G, w)$ (назовем его LPCSTABb) шаги 4 и 5 заменяются следующими.

Шаг 4a. Среди найденных V кратчайших путей оставляем только те, которые определяют нечетный цикл (такая проверка требуется, поскольку не каждый кратчайший путь определяет нечетный цикл в графе G). Пусть число кратчайших путей, определяющих нечетные циклы, равно n ($n \le |V|$), и длины этих кратчайших путей равны l_i, $i = 1, \dots, n$. Если все l_i, $i = 1, \dots, n$, не меньше единицы, то $\alpha_C^*(G, w) = f^*$ и останов. Иначе переходим к шагу 5a.

Шаг 5a. Сформируем список нечетных циклов C', включающий все нечетные циклы, соответствующие кратчайшим путям из $1, \dots, n$, для которых $l_i < 1$. Каждый из нечетных циклов из списка C' определяет ограничение в форме (5), которое в точке x^* является нарушенным (не обязательно максимально нарушенным, хотя по крайней мере один из таких нечетных циклов дает максимально нарушенное ограничение). Перейдем к шагу 6.

Оба алгоритма, LPCSTABa и LPCSTABb, позволяют найти оценку $\alpha_C^*(G, w)$, и если граф G принадлежит семейству t-совершенных графов, то оценка $\alpha_C^*(G, w)$ будет точной для $\alpha(G, w)$, т.е. $\alpha_C^*(G, w) = \alpha(G, w)$. Однако время работы каждого алгоритма разное. Для оценки времени нахождения $\alpha_C^*(G, w)$, когда граф содержит порядка нескольких сотен вершин, оба алгоритма программно реализованы (про-

граммы LPCSTABa и LPCSTABb). Для решения ЛП-задачи использовалась программа SOPLEX [3], а для нахождения кратчайшего пути в ориентированном графе — программа DIKH (из библиотеки SPLIB) [4].

Результаты тестовых экспериментов для сравнения времени работы обоих алгоритмов приведены в табл. 1. Вычисления проводились на процессоре AMD Athlon 1,81GHz. Верхняя оценка $\alpha_C^*(G)$ для взвешенного числа устойчивости графа ($w_i = 1$ $\forall i \in V(G)$) вычислена для ряда тестовых наборов из DIMACS-библиотеки [5]. Здесь *NCa* — максимальное количество включенных в ЛП-задачу нечетных циклов программой LPCSTABa. Оно равно и количеству итераций, затраченных на нахождение оценки $\alpha_C^*(G)$. Для программы LPCSTABb приведено количество итераций (столбец itnb). Из табл. 1 легко видеть, что по времени программа LPCSTABb всегда выигрывает, иногда существенно, у программы LPCSTABa (столбец t_a / t_b), незначительно проигрывая в количестве накопленных нечетных циклов (столбец *NCb* / *NCa*). Например, для графа hamming8-4 достигнут выигрыш по времени в пятьдесят раз, нечетных циклов накапливается не так много (в два раза больше). При этом пришлось решать ЛП-задачу четыре раза, а находить кратчайшие пути в ориентированном графе с 256 вершинами и 47104 дугами 4×256 раз.

Таблица 1

Тестовые наборы из DIMACS-графов	Результаты тестовых экспериментов для оценки $\alpha_C^*(G)$ и затраты на ее нахождение							
	$\|V\|$	$\|E\|$	$\alpha(G)$	$\alpha_C^*(G)$	*NCa*	itnb	t_a / t_b	*NCb* / *NCa*
c-fat200-1	200	18366	12	66,6667	241	4	43,5	2,2
c-fat200-2	200	16665	24	66,6667	232	4	40,1	2,4
c-fat200-5	200	11427	58	66,6667	237	4	36,1	2,2
johnson16-2-4	120	1680	8	40,0000	143	4	29,7	2,1
johnson8-2-4	28	210	4	9,3333	32	4	5,8	2,1
johnson8-4-4	70	560	14	23,3333	81	4	14,5	2,1
keller4	171	5100	11	57,0000	211	4	35,1	2,2
hamming6-2	64	192	32	32,0000	0	1	1,0	—
hamming6-4	64	1312	4	21,3333	73	4	14,7	2,2
hamming8-2	256	1024	128	128,0000	0	1	1,0	—
hamming8-4	256	11776	16	85,3333	322	4	50,6	2,1
san200-0.7-2	200	5970	18	66,6667	232	5	30,7	3,3
san200.0.9-1	200	1990	70	70,0000	175	6	14,8	4,7
san200-0.9-2	200	1990	60	66,6667	274	12	12,9	4,7
san200.0.9-3	200	1990	44	66,6667	247	8	17,4	3,7
brock200-1	200	5066	21	66,6667	237	4	31,4	2,9
mann-a27	378	702	126	135,0000	146	7	15,4	1,6
mann-a9	45	72	16	18,0000	12	2	4,2	1,7

Относительно точности оценки $\alpha_C^*(G)$ для рассмотренных примеров отметим следующее. В трех случаях, графы san200.0.9-1, hamming6-2 и hamming8-2, оценка $\alpha_C^*(G)$ оказалась точной верхней оценкой для $\alpha(G)$, для графов hamming6-2 и hamming8-2 она была точной при наличии только реберных ограничений в ЛП-задаче и не было найдено ни одного нечетного цикла (в связи с этим t_a / t_b равно единице). Однако для остальных примеров оценка $\alpha_C^*(G)$ неточна, и для графов c-fat200-1 и hamming8-4 она сильно завышена. В следующих разделах рассматривается улучшенный

вариант ЛП-ориентированного алгоритма, который позволит в ряде случаев найти более точные верхние оценки для числа устойчивости этих графов.

ОЦЕНКА $\alpha^*_{W_p}(G, w)$ И W_p-СОВЕРШЕННЫЕ ГРАФЫ

Рассмотрим более точную верхнюю оценку для $\alpha(G, w)$, чем оценка $\alpha^*_C(G, w)$, и обсудим ее связь с одной из верхних оценок, предложенной Н.З. Шором [6].

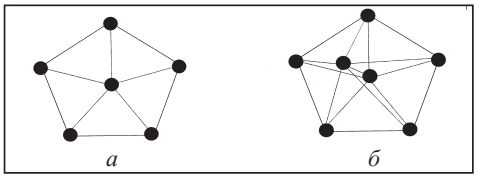

Рис. 1

Пусть имеется граф W_{2k+1+p}, вершинами которого являются вершины непересекающихся нечетного цикла C_{2k+1} и клики Q_p (полный подграф, содержащий p вершин). Если $p=1$, то клика состоит из одной вершины. Множество ребер в графе W_{2k+1+p} включает все ребра нечетного цикла C_{2k+1}, все ребра клики Q_p, а также ребра, связывающие каждую вершину C_{2k+1} со всеми вершинами клики Q_p. Граф W_{2k+1+p} принято называть p-колесом (p-wheel) [7]. Примеры 1-колеса и 2-колеса на базе нечетного цикла C_5 приведены на рис. 1, *а,б* соответственно.

С p-колесом связано семейство линейных неравенств

$$\sum_{i\in V(C_{2k+1})} x_i + k \sum_{j\in V(Q_p)} x_j \le k \quad \forall\, W_{2k+1+p} \in G, \tag{11}$$

которые справедливы для многогранника $STAB(G)$. Семейство неравенств (11) принято называть p-колесными ограничениями (p-wheel constraints). Они означают, что для каждого p-колеса из графа G в устойчивое (независимое) множество может быть включена либо одна из вершин клики Q_p, либо k вершин из нечетного цикла C_{2k+1}. Если семейство неравенств (11) добавить к линейным неравенствам, которые определяют многогранник $CSTAB(G)$, то получим многогранник

$$W_p STAB(G) = \{x \in R^{|V|}:\ x \text{ удовлетворяет (3)–(5), (11)}\},$$

который аппроксимирует (сверху) многогранник устойчивых множеств $STAB(G)$. Семейство графов, для которых $STAB(G) = W_p STAB(G)$, назовем W_p-совершенными (W_p-perfect).

Оценка $\alpha^*_{W_p}(G, w)$ связана с решением ЛП-задачи

$$\alpha^*_{W_p}(G, w) = \max \sum_{i\in V(G)} w_i x_i, \quad x \in W_p STAB(G). \tag{12}$$

Если граф G принадлежит семейству W_p-совершенных графов, то $\alpha^*_{W_p}(G, w) = \alpha(G, w)$. Для произвольного графа G всегда имеет место неравенство

$$\alpha^*_C(G, w) \ge \alpha^*_{W_p}(G, w) \ge \alpha(G, w), \tag{13}$$

т.е. оценка $\alpha^*_{W_p}(G, w)$ является верхней оценкой (оценкой сверху) для $\alpha(G, w)$ и она всегда не хуже, чем оценка $\alpha^*_C(G, w)$.

ЛП-задача (12) в общем случае может содержать неполиномиальное количество ограничений. В отличие от ЛП-задачи (6) она не может быть разрешена за полиномиальное время. Так, например, при $k=1$ (т.е. рассматриваются только нечетные циклы, совпадающие с 3-кликой) и произвольном p, $1 \le p \le |V|-3$, из многогранника $W_p STAB(G)$ следует известный кликовый многогранник (clique polytope)

$$QSTAB(G) = \begin{cases} \sum\limits_{i\in V(Q)} x_i \le 1 & \text{для каждой клики } Q \in G, \\ 0 \le x_i \le 1 & \text{для каждой вершины } i \in G. \end{cases}$$

Здесь неравенства для 2-клик следуют из реберных неравенств (4), неравенства для 3-клик — из неравенств для нечетных циклов (5) при $k=1$, а неравенства для клик с большим количеством вершин — из неравенств для p-колес (11) при $k=1$ и произвольном p. С многогранником $QSTAB(G)$ связана ЛП-ориентированная верхняя оценка

$$\alpha_Q^*(G,w)=\max\sum_{i\in V(G)}w_ix_i,\quad x\in QSTAB(G),$$

которая называется дробным взвешенным числом устойчивости графа G. В общем случае нахождение $\alpha_Q^*(G,w)$ — NP-трудная задача, поскольку это связано с нахождением максимальной клики в графе G, а такая задача для произвольного графа G является NP-трудной [1]. Однако когда граф G принадлежит семейству совершенных графов (для них $STAB(G)=QSTAB(G)$), то задача нахождения $\alpha_Q^*(G,w)$ разрешима за полиномиальное время. Это объясняется свойствами не самой оценки $\alpha_Q^*(G,w)$, а более точной верхней оценки для $\alpha(G,w)$, а именно известного числа Ловаса $\vartheta(G,w)$ [2]. Число Ловаса можно найти за полиномиальное время, и для произвольного графа G оно удовлетворяет неравенству

$$\alpha_Q^*(G,w)\ge\vartheta(G,w)\ge\alpha(G,w),$$

которое, если граф G принадлежит семейству совершенных графов, превращается в точное равенство

$$\alpha_Q^*(G,w)=\vartheta(G,w)=\alpha(G,w).$$

Отметим еще один интересный многогранник $WSTAB(G,w)$, с которым связано семейство W-совершенных (W-perfect) графов. Он получается из $W_pSTAB(G)$ при $p=1$, т.е. неравенства для произвольного p-колеса в форме (11) заменяются линейными неравенствами

$$\sum_{i\in V(C_{2k+1})}x_i+k\,x_{i_{2k+2}}\le k\quad\forall\,W_{2k+2}\in G$$

для обычного колеса W_{2k+2}, когда клика состоит всего из одной вершины. Семейство этих неравенств принято называть колесными ограничениями (wheel constraints). С многогранником $WSTAB(G)$ связана верхняя оценка для $\alpha(G,w)$, являющаяся решением следующей ЛП-задачи:

$$\alpha_W^*(G,w)=\max\sum_{i\in V(G)}w_ix_i,\quad x\in WSTAB(G).$$

Нахождение оценки $\alpha_W^*(G,w)$ может быть осуществлено за полиномиальное время для произвольного графа G [2]. Если граф G принадлежит семейству W-совершенных (W-perfect) графов (для них $STAB(G)=WSTAB(G)$), то оценка $\alpha_W^*(G,w)$ является точной для $\alpha(G,w)$.

Итак, для совершенных, t-совершенных и W-совершенных графов задача нахождения $\alpha(G,w)$ разрешима за полиномиальное время. Поэтому интерес представляет следующий вопрос: будет ли полиномиально разрешимой задача нахождения $\alpha(G,w)$ для семейства W_p-совершенных графов? Оказывается — будет. Это объясняется свойствами одной из оценок Шора, которая для произвольного графа G является более точной верхней оценкой для $\alpha(G,w)$, чем оценка $\alpha_{W_p}^*(G,w)$.

Оценка Шора (обозначим ее $\psi_1^*(G,w)$) является оптимальной (наилучшей) лагранжевой (двойственной) оценкой для квадратичной булевой задачи

$$\alpha(G,w)=\max\sum_{i\in V}w_ix_i \tag{14}$$

при ограничениях:

$$x_ix_j=0\quad\forall\,(i,j)\in E(G), \tag{15}$$

$$x_i^2 - x_i = 0 \quad \forall i \in V(G), \tag{16}$$

$$x_i x_k + x_j x_k \le x_k \quad \forall (i, j) \in E(G),\ k \ne i, j, \tag{17}$$

в виде которой формулируется задача о максимальном взвешенном устойчивом множестве графа. Оценку $\psi_1^*(G, w)$ можно найти с любой заданной точностью с помощью методов минимизации негладких выпуклых функций [6]. Здесь булева переменная $x_i \in \{0, 1\}$ равна единице, если вершина $i \in V$ включается в устойчивое множество, и равна нулю в противном случае. Булевы переменные для всех вершин описаны квадратичными ограничениями-равенствами (16). Квадратичные ограничения (15) означают, что если две вершины связаны ребром в графе G, то они обе не могут одновременно принадлежать устойчивому множеству. Квадратичные ограничения в форме неравенств (17) функционально избыточны, т.е. не изменяют множество оптимальных решений квадратичной задачи (14)–(16). Они получаются как результат умножения реберных ограничений в виде (4) на переменные x_k, $k \ne i, j$. Знак неравенств не изменится в силу того, что $x_k = x_k^2 \ge 0$, и в результате получаем квадратичные неравенства (17).

Именно наличие ограничений (17) придает оценке $\psi_1^*(G, w)$ ряд замечательных свойств, которые связаны со специальными семействами графов. Так, например, в [6] показано, что верхняя оценка $\psi_1^*(G, w)$ является точной для $\alpha(G, w)$, когда граф G t-перфектный. Схема доказательства основана на том, что ослаблением (релаксацией) квадратичных ограничений (15)–(17) легко получить ЛП-задачу для оценки $\alpha_C^*(G, w)$. Однако из квадратичных ограничений (15)–(17) следуют неравенства в форме (11) [8]. Это означает, что для произвольного графа G справедливо неравенство

$$\alpha_{W_p}^*(G, w) \ge \psi_1^*(G, w) \ge \alpha(G, w). \tag{18}$$

Для W_p-совершенных графов неравенство (18) превращается в строгое равенство

$$\alpha_{W_p}^*(G, w) = \psi_1^*(G, w) = \alpha(G, w), \tag{19}$$

которое означает, что оценка $\psi_1^*(G, w)$ для них является точной верхней оценкой для $\alpha(G, w)$. Учитывая, что оценку $\psi_1^*(G, w)$ можно найти за полиномиальное время, получаем, что если граф G принадлежит семейству W_p-совершенных графов, то задача нахождения $\alpha(G, w)$ полиномиально разрешима. Однако ответ, который дает равенство (19), носит скорее теоретический характер, чем практический. Вычисление оценки $\psi_1^*(G, w)$ связано с решением задачи минимизации выпуклой негладкой функции, определенной на параметрически заданном семействе неотрицательно определенных симметричных матриц размера $|V| \times |V|$, линейно зависящих от неизвестных множителей Лагранжа (соответствуют ограничениям (15), (16) и (17)). Количество неизвестных множителей Лагранжа имеет порядок $O(|V|^3)$, и когда граф содержит несколько сотен вершин, достаточно точное нахождение оценки $\psi_1^*(G, w)$ является практически неразрешимой задачей для современных методов недифференцируемой оптимизации.

ОЦЕНКА $\alpha_W(G, w)$ И АЛГОРИТМ LPWSTAB

Нахождение оценки $\alpha_{W_p}^*(G, w)$ — *NP*-трудная задача. Более того, *NP*-трудной является и задача нахождения на основе уже имеющегося нечетного цикла такого p-колеса, для которого в точке $x^* = \{x_1^*, \ldots, x_{|V|}^*\}$ максимально нарушается линейное неравенство в виде (11). Поэтому вместо оценки $\alpha_{W_p}^*(G, w)$ рассмотрим

ослабленную ЛП-ориентированную верхнюю оценку для $\alpha(G, w)$ на основе линейных неравенств для таких p-колес в графе G, которые легко построить. Эту оценку назовем оценкой $\alpha_W(G, w)$ и в основу алгоритма для ее нахождения положим следующее.

На каждой итерации алгоритма будем решать ЛП-задачу в виде

$$f_W^* = \max \sum_{i \in V} w_i x_i \tag{20}$$

при ограничениях:

$$0 \le x_i \le 1 \quad \forall i \in V(G), \tag{21}$$

$$x_i + x_j \le 1 \quad \forall (i, j) \in E(G), \tag{22}$$

$$\sum_{i \in V(C_{2k+1})} x_i \le k \quad \forall C_{2k+1} \in C_{\text{odd}} \in G, \tag{23}$$

$$\sum_{i \in V(C_{2k+1})} x_i + k \sum_{j \in V(Q_p)} x_j \le k \quad \forall W_{2k+1+p} \in W_{\text{odd}} \in G. \tag{24}$$

Здесь C_{odd} и W_{odd} — конечные множества нечетных циклов и p-колес (возможно, пустые). Для формирования множества нечетных циклов применим алгоритм нахождения нечетных циклов из [2], используя $x^* = \{x_1^*, \dots, x_{|V|}^*\}$ — оптимальное решение ЛП-задачи (20)–(24). Для формирования множества p-колес используем уже найденные нечетные циклы и дополним их до p-колес с помощью простейшего по количеству вычислений алгоритма (по типу «жадного» алгоритма). Такой алгоритм назовем алгоритмом «последовательного включения наилучших вершин».

Пусть $i_1, \dots, i_{2k+1}$ — вершины, через которые проходит нечетный цикл C_{2k+1}, и вектор x^* — оптимальное решение ЛП-задачи (20)–(24). Чтобы на основе нечетного цикла C_{2k+1} построить p-колесо в алгоритме «последовательного включения наилучших вершин», требуется выполнение следующей последовательности действий. Найдем подмножество вершин из $V(G)$, для которого каждая из входящих в него вершин связана со всеми вершинами нечетного цикла C_{2k+1}. Если такое подмножество вершин непусто, то выберем из него ту вершину, для которой значение компоненты вектора x^* максимально, и добавим его к множеству вершин с номером $(2k+2)$. В результате получим 1-колесо с множеством вершин $i_1, \dots, i_{2k+1}, i_{2k+2}$. Для него точно так же найдем подмножество вершин из $V(G)$, для которого каждая из входящих в него вершин связана со всеми вершинами 1-колеса. Если подмножество вершин непусто, то выберем вершину с максимальным значением компоненты вектора x^* и добавим к множеству вершин с номером $(2k+3)$. В результате получим 2-колесо с множеством вершин $i_1, \dots, i_{2k+1}$, i_{2k+2}, i_{2k+3}. Для него применим такую же процедуру, и либо получим 3-колесо, либо прервем процесс, если 3-колеса не существует (подмножество вершин из $V(G)$, связанных с каждой из вершин 2-колеса, окажется пустым). Пусть 3-колесо получено. Будем продолжать процесс до тех пор, пока подмножество вершин из $V(G)$, связанных с каждой из вершин уже найденного p-колеса, не окажется пустым.

Алгоритм для нахождения оценки $\alpha_W(G, w)$ (назовем его LPWSTAB) состоит в следующем.

Начальная установка. Положим itn = 0 и $C_0 = \varnothing$, $W_0 = \varnothing$ (множества нечетных циклов и p-колес пустые). Перейдем к шагу 1.

Шаг 1. Имеем множество нечетных циклов C_{itn} и множество p-колес W_{itn}. Положим $C_{\text{odd}} = C_{\text{itn}}$, $W_{\text{odd}} = W_{\text{itn}}$ и, решив ЛП-задачу (20)–(24), найдем f_W^* и $x^* = (x_1^*, \dots, x_{|V|}^*)$.

Шаг 2. Построим взвешенный неориентированный граф $G' = (V', E')$. Множество вершин $V'(G')$ включает вершину i и ее копию i' для всех $i \in V(G)$. Множес-

тво ребер $E'(G')$ состоит из ребер, которые соединяют все те пары вершин (i, j') и (i', j), для которых пара вершин i и j соединена ребром (i, j) в графе G. Присвоим ребрам (i, j') и (i', j) один и тот же положительный вес (длина ребра) $l(i', j) = l(i, j') = 1 - x_i^* - x_j^*$.

Шаг 3. Рассмотрим граф G' как ориентированный (ребру (i, j') соответствуют дуги (i, j') и (j', i) с длинами, равными длине ребра (i, j'), т.е. $l(i, j')$), и для каждой вершины $i \in V$ найдем кратчайший путь, который начинается в вершине i и заканчивается в вершине i' (копии вершины i). Число таких кратчайших путей равно $|V|$. Из них оставляем только те, которые определяют нечетный цикл. Пусть число таких кратчайших путей равно n $(n \le |V|)$. Найдем нечетные циклы $C_{2k_1+1}, \ldots, C_{2k_n+1}$ (они соответствуют n найденным кратчайшим путям).

Шаг 4. Установим множества C' и W' пустыми и заполним их следующим образом. Каждый из нечетных циклов C_{2k_i+1} для $i = 1, \ldots, n$ с помощью алгоритма «последовательного включения наилучших вершин» дополняем до p-колеса в графе G. Пусть такое p-колесо получено. Если линейное неравенство в форме (11) для него нарушено, то найденное p-колесо включаем в множество W'. Если построить p-колесо на базе нечетного цикла не удалось, но линейное неравенство в виде (5) для этого нечетного цикла является нарушенным, то такой нечетный цикл включаем в множество C'. Если после завершения этой процедуры для всех $i = 1, \ldots, n$ оба множества C' и W' пустые, то $\alpha_W(G, w) = f_W^*$ и останов. Иначе перейдем к шагу 5.

Шаг 5. Положим $C_{\text{itn}+1} = C_{\text{itn}} \cup C'$, $W_{\text{itn}+1} = W_{\text{itn}} \cup W'$, itn = itn + 1. Перейдем к шагу 1.

ЛП-задача (20)–(24) построена как расширение ЛП-задачи (7)–(10) за счет конечного набора справедливых для p-колес линейных неравенств в виде (24). Поэтому оценку $\alpha_W(G, w)$ можно рассматривать как улучшение оценки $\alpha_C^*(G, w)$ с помощью использования линейных неравенств для p-колес в графе G. В результате она будет сохранять свойства оценки $\alpha_C^*(G, w)$, а в случае обнаружения p-колес, для которых линейные неравенства в виде (11) являются нарушенными, оценка $\alpha_W(G, w)$ может оказаться более точной верхней оценкой для $\alpha(G, w)$, чем оценка $\alpha_C^*(G, w)$.

В целях проверки свойств оценки $\alpha_W(G, w)$ алгоритм реализован программой LPWSTAB на языке C++. Для решения ЛП-задачи использована программа SOPLEX [3], а для нахождения кратчайшего пути в ориентированном графе — программа DIKH [4]. В программе LPWSTAB используется параметр ε, который означает, что линейные неравенства, связанные с нечетным циклом C_{2k+1} и p-колесом W_{2k+1+p}, считаются нарушенными (в точке x^*) и включаются в ЛП-задачу, если для них выполняются условия:

$$\sum_{i \in V(C_{2k+1})} x_i^* \ge k + \varepsilon, \qquad \sum_{i \in V(C_{2k+1})} x_i^* + k \sum_{j \in V(Q_p)} x_j^* \ge k + \varepsilon.$$

Первая серия экспериментов с программой LPWSTAB заключалась в проверке точности оценки $\alpha_W(G, w)$, где $w_i = 1 \; \forall i \in V(G)$, для того же набора DIMACS-графов, для которого вычислялась оценка $\alpha_C^*(G)$. Рассматривались 16 графов. Исключены только два графа — hamming6-2 и hamming8-2, для которых не требовалось включения ни одного нечетного цикла. Для программы LPWSTAB использовалось значение $\varepsilon = 0{,}01$. Результаты эксперимента отражены в табл. 2. Для каждого из 16 графов приведены значения оценки $\alpha_W(G)$ и затраты на ее нахождение — количество итераций (столбец itn) и время в секундах на процессоре AMD Athlon 1,81GHz (столбец t_W). Остальные столбцы в таблице характеризуют структуру накопленных программой LPWSTAB (включенных в ЛП-задачу на последней итерации) линейных неравенств, исключая «реберные» неравенства. Столбец N задает полное количество включенных в ЛП-задачу линейных неравенств, а в столбцах NC, NW и NQ указано, сколько среди них было нечетных циклов (сюда же включены 3-клики), p-колес с $k \ge 2$ и клик, которые получены как p-колеса при $k = 1$ и произвольных p.

Из табл. 2 видно, что оценка $\alpha_W(G)$ является точной для $\alpha(G)$ для девяти графов из 16. Структура накопленных неравенств для графа san200.0.9-1 позволяет утверждать, что он не может быть t-совершенным, хотя для него оценка $\alpha_C^*(G)$ является точной верхней оценкой для $\alpha(G)$ (см. табл. 1). Такой вывод дают возможность сделать 16 накопленных 4-клик (см. столбец NQ в табл. 2), и, следовательно, для графа san200.0.9-1 можно подобрать такие веса вершин, при которых оценка $\alpha_C^*(G, w)$ не будет точной для $\alpha(G, w)$.

Таблица 2

Тестовые наборы из DIMACS-графов	Результаты тестовых экспериментов для оценки $\alpha_W(G)$ и затраты на ее нахождение								
	$\|V\|$	$\alpha(G)$	$\alpha_W(G)$	itn	t_W	N	NC	NW	NQ
c-fat200-1	200	12	12,00	31	485,8	5787	0	0	5787
c-fat200-2	200	24	24,00	17	219,5	3342	0	12	3330
c-fat200-5	200	58	66,66	3	29,0	533	533	0	0
johnson16	120	8	8,00	3	2,3	128	36	7	85
johnson8-2	28	4	4,00	3	0,7	41	8	7	26
johnson8-4	70	14	14,00	3	1,0	139	12	4	123
keller4	171	11	14,82	14	32,8	1151	41	36	1074
hamming6-4	64	4	5,33	5	2,1	301	0	16	285
hamming8-4	256	16	16,00	5	54,8	1144	48	48	1048
san200-0.7	200	18	19,35	18	58	1569	134	1	1434
san200.0.9-1	200	70	70,00	4	6,4	663	646	1	16
san200-0.9-2	200	60	60,00	11	16	1369	1299	0	70
san200.0.9-3	200	44	44,00	21	23,4	1526	1378	0	148
brock200-1	200	21	38,58	32	88,5	1543	227	28	1288
mann-a27	378	126	135,00	7	5,1	245	245	0	0
mann-a9	45	16	18,00	1	0,5	21	21	0	0

Затраты по времени нахождения оценки $\alpha_W(G)$ превышают (иногда существенно) затраты по времени на нахождение оценки $\alpha_C^*(G)$ программой LPCSTABb. Одну из причин этого объясняют накопленные неравенства для графа san200.0.9-1, что подсказывает путь для усовершенствования алгоритма LPWSTAB. Так, для оценки $\alpha_W(G)$ накоплено 663 неравенства (из них нечетных циклов — 646 и 4-клик — 16). Их меньше, чем количество накопленных нечетных циклов программой LPCSTABb (равно 828) при той же точности $\varepsilon = 0{,}01$, и существенно больше, чем число нечетных циклов, накопленных программой LPCSTABa (равно 175, см. табл. 1). Это означает, что процедура включения неравенств в ЛП-задачу в программе LPWSTAB требует доработки и дополнительного исследования с целью уменьшить количество включаемых в ЛП-задачу нарушенных неравенств. Например, можно устроить их ранжирование по точности нарушения неравенств, ограничить количество включаемых неравенств на одной итерации. Создание более конструктивного критерия для включения нарушенных неравенств может способствовать и уменьшению общего количества итераций LPWSTAB-алгоритма, что может сделать затраты по времени нахождения оценки $\alpha_W(G, w)$ более близкими к затратам на нахождение оценки $\alpha_C^*(G, w)$ с помощью алгоритма LPCSTABb.

Кроме тестовых примеров из DIMACS-библиотеки программа LPWSTAB проверена на тестовых задачах для помехоустойчивых кодов из сайта http://www.reseach.att.com/~njas/doc/graphs.html, для которых получены помехозащищенные коды максимального объема [9]. Например, для всех тестовых задач,

связанных с корректированием единичной ошибки в Z-канале, программа LPWSTAB нашла верхние оценки для максимального объема кода, которые совпали с числом Ловаса. Для тестовой задачи, где граф включает 1024 вершины и 33280 ребер, программа затратила время $t = 1139$ с на процессоре AMD Athlon. При этом пришлось решать ЛП-задачу 12 раз и находить кратчайшие пути в ориентированном графе (содержит 2048 вершин и 133120 дуг) 12×1024 раз. Количество накопленных неравенств при этом составило 2768 — из них 249 нечетных циклов, 2468 клик и 51 p-колесо.

Вторая серия экспериментов для программы LPWSTAB связана с ее выходом, а именно с ЛП-задачей, которая соответствует оценке $\alpha_W(G)$. Целью экспериментов было выяснить: насколько быстрее с помощью программ для решения задачи целочисленного линейного программирования можно решить булеву ЛП-задачу, подготовленную программой LPWSTAB, чем булеву ЛП-задачу с реберными ограничениями (включает только линейные неравенства для всех ребер графа). Для этой цели была выбрана свободно распространяемая программа SCIP (Solving Constraint Integer Programming) [10]. Для тех же наборов DIMACS-графов результаты экспериментов отражены в табл. 3. Здесь N_{SCIP} — количество переменных в обеих булевых ЛП-задачах (равно количеству вершин графа), M_{ESCIP} — количество ограничений в булевой ЛП-задаче с реберными ограничениями (равно количеству вершин и ребер в графе), M_{WSCIP} — количество ограничений в булевой ЛП-задаче для оценки $\alpha_W(G, w)$ (равно сумме количества вершин и ребер в графе и накопленных программой LPWSTAB линейных неравенств), M_{SCIP} — количество ограничений в булевой ЛП-задаче, которое осталось после препроцессинга программы SCIP (задача с этим количеством ограничений является стартовой для метода ветвей и границ, реализованной в программе SCIP), t_{ESCIP} и t_{WSCIP} — время решения программой SCIP булевой ЛП-задачи с реберными ограничениями и булевой ЛП-задачи для оценки $\alpha_W(G)$ соответственно. Последний столбец характеризует отношение затрат по времени решения обеих булевых ЛП-задач: указано, во сколько раз быстрее программа SCIP решила булеву ЛП-задачу для оценки $\alpha_W(G)$, чем булеву ЛП-задачу для реберных ограничений.

Из табл. 3 видно, что булева ЛП-задача для оценки $\alpha_W(G)$ проиграла только в четырех случаях: незначительно для графов c-fat200-5, san200-0.9-2 и mann-a9 и существенно (почти в три раза) для графа san200.0.9-3. В остальных случаях булева ЛП-задача для оценки $\alpha_W(G)$ выиграла, и очень существенно (более чем в 20 раз для графа hamming8-4, более чем в восемь раз для графа c-fat200-1, более чем в четыре раза для графа c-fat200-2, почти в четыре раза для графа keller4, более чем в два раза для графа san200-0.7-2, более чем в шесть раз для графа mann-a27). Наиболее значительный выигрыш по реальному времени получился для графа brock200-1 — 2 ч 13 мин против 6 ч 19 мин. Для графов johnson16-2-4, johnson8-2-4 и johnson8-4-4 как незначительный, так и существенный выигрыш по времени ничего не значат. Для них общее время решения обеих булевых ЛП-задач очень мало. В этом случае представляет интерес количество линейных неравенств, которые оставлены в булевых ЛП-задачах после препроцессинга. Оно невелико по сравнению с количеством вершин в этих графах, и это объясняется нахождением алгоритмом LPWSTAB достаточно «хороших» кликовых ограничений, которые сравнительно небольшим количеством покрывают вершины графа.

С помощью программы SCIP и подготовленной программой LPWSTAB ЛП-задачи для оценки $\alpha_W(G)$ найдена точная верхняя оценка для максимального объема кода, корректирующего одно удаление бита, для графа 1dc1024. Затраты по времени составили 55 ч на процессоре AMD Athlon 1,81GHz. Это значительно меньше, чем время, указанное на сайте http://www.reseach.att.com/~njas/doc/graphs.html. Оно равно 298 ч и было достигнуто с помощью метода ветвей и границ и оценок Ловаса на мощном компьютере IBM P690. Для графа 1dc1024 оценка $\alpha_W(G)$ равна 96,412 (точное решение $\alpha(G) = 94$). Программа LPWSTAB накопила 3483 неравенства (из них 276 — нечетных циклов, 3154 — клик, 53 — p-колес) и затратила 1128,7 с. Булева ЛП-задача содержала 28570 линейных неравенств до препроцессинга и 2162 линейных неравенства после препроцессинга программы SCIP.

Таблица 3

Тестовые наборы из DIMACS-графов	Результаты сравнительных экспериментов с помощью программы SCIP для булевых ЛП-задач						
	N_{SCIP}	M_{ESCIP}	t_{ESCIP}	M_{WSCIP}	M_{SCIP}	t_{WSCIP}	$\frac{t_{ESCIP}}{t_{WSCIP}}$
c-fat200-1	200	18366	1041	24353	5420	125	8,33
c-fat200-2	200	16665	410	20207	5284	91	4,51
c-fat200-5	200	11427	1194	12160	10831	1472	0,81
johnson16-2-4	120	1680	1,41	1928	48	0,14	10,07
johnson8-2-4	28	210	0,04	237	16	0,03	1,33
johnson8-4-4	70	560	1,64	769	135	0,09	18,22
keller4	171	5100	829	6422	1155	259	3,20
hamming6-4	64	1312	10,1	1677	209	2,31	4,39
hamming8-4	256	11776	746	13176	1324	26,64	28,00
san200-0.7-2	200	5970	410	7739	2213	112	3,66
san200.0.9-1	200	1990	5,5	2853	2312	2,39	2,30
san200-0.9-2	200	1990	13,3	3559	2638	4,09	3,25
san200.0.9-3	200	1990	111,9	3716	2562	346	0,32
brock200-1	200	5066	22766	6809	2515	8065	2,82
mann-a27	378	702	359	1325	597	55	6,50
mann-a9	45	72	0,21	138	58	0,26	0,91

При работе с большими графами (порядка тысяч вершин) проявился основной недостаток алгоритма LPWSTAB. Он связан с тем, что лежащая в его основе ЛП-задача только накапливает линейные ограничения, не отсеивая заведомо лишние. Из-за этого в программе SCIP значительное время занимает препроцессинг ЛП-задачи. В результате в булевой ЛП-задаче для оценки $\alpha(G, w)$ остается намного меньше ограничений. Главным образом по этой причине приостановлены попытки решать задачи, связанные с графами больших размеров. Очевидно, что отсев лишних линейных ограничений в алгоритме LPWSTAB только ускорит время нахождения оценки $\alpha(G, w)$, и это является резервом для разработки более быстрых реализаций LPWSTAB-программы. Так, например, простейшие усовершенствования здесь очевидны: те реберные неравенства, для которых пара вершин входит в какую-либо из клик для ограничений, входящих в ЛП-задачу, следует отбросить. Более тонкие отсевы связаны с анализом двойственных переменных для ЛП-задачи на каждой из итераций LPWSTAB-алгоритма.

ЗАКЛЮЧЕНИЕ

Отметим ряд важных моментов для задачи нахождения взвешенного числа устойчивости графа $\alpha(G, w)$, которые следуют из вычислительных экспериментов для оценок $\alpha_C^*(G, w)$ и $\alpha_W(G, w)$.

1. Современные общедоступные ЛП-программы позволяют создавать для современных ПЭВМ практически пригодные (по времени) алгоритмы для нахождения ЛП-ориентированных верхних оценок для $\alpha(G, w)$ для графов, содержащих несколько сотен вершин. Это подтверждают тестовые эксперименты с алгоритмами LPCSTABb и LPWSTAB.

2. Точность ЛП-ориентированных верхних оценок существенно зависит от использования того набора неравенств, с помощью которых аппроксимируется мно-

гогранник $STAB(G)$. Например, даже простейший учет p-колес в графе G позволяет очень много сделать для улучшения точности ЛП-ориентированной оценки по сравнению с использованием только нечетных циклов. Усовершенствованные схемы построения p-колес могут усилить точность ЛП-ориентированных верхних оценок.

3. На основе современных ЦЛП-программ (целочисленного линейного программирования) можно получить и обосновать точную верхнюю границу для $\alpha(G, w)$ и, более того, найти одно из булевых решений, на котором достигается эта граница. При этом качество ЛП-ориентированных оценок играет важную роль для ускорения работы ЦЛП-программ. Это доказывают численные эксперименты с программой SCIP.

Можно ли расширить границы применимости ЛП-ориентированных оценок на случай графов с несколькими тысячами вершин, используя коммерческие ЛП-программы и ЦЛП-программы? Ответ требует более детального исследования. Пример с графом 1dc1024 указывает на то, что с помощью программы CPLEX доказательство максимального объема помехоустойчивого кода для этого графа потенциально можно осуществить за время $t * 0{,}08$, где t — время доказательства с помощью программы SCIP. Это следует из того, что работа программы SCIP оценивается по времени в соотношении 1 к 0,08 по отношению к промышленному варианту программы CPLEX (http://scip.zib.de). При этом можно ожидать, что на процессоре AMD Athlon 1,81GHz обоснование точности верхней оценки максимального объема помехоустойчивого кода для графа 1dc1024 можно осуществить с помощью CPLEX за время, меньшее пяти часов (ср.: 55 часов с помощью программы SCIP). Подтвердится ли это предположение? Ответ на этот вопрос может дать только численный эксперимент.

СПИСОК ЛИТЕРАТУРЫ

1. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. — М.: Мир, 1982. — 416 с.
2. Grötschel M., Lóvasz L., Schrijver A. Geometric algorithms and combinatorial optimization. — Berlin: Springer-Verlag, 1988. — 362 p.
3. Wunderling R. Paralleler und objektorientierter simplex-algorithmus. — Berlin: Techn. Univ., 1996 (http://www.zib.de/Publications/abstracts/TR-96-09/).
4. Cherkassky B.V., Goldberg A.V., Radzik T. Shortest paths algorithms: Theory and experimental evaluation // Math. Program. — 1996. — **73**. — P. 129–174 (http://www.avglab.com/andrew/soft.html).
5. DIMACS (1995). Cliques, coloring, and satisfiability: second DIMACS implementation challange (http://dimacs.rutgers.edu/Challenges/).
6. Shor N.Z. Nondifferentiable optimization and polynomial problems. — Boston; Dordrecht; London: Kluwer Acad. Publ., 1998. — 412 p.
7. Cheng E., Cunningham W.H. Wheel inequalities for stable set polytopes // Math. Program. — 1997. — **77**, N 3. — P. 389–421.
8. Стецюк П.И., Чумаков Б.М. О свойствах одной верхней оценки Н.З. Шора для взвешенного числа устойчивости графа // Праці міжнар. симп. «Питання оптимізації обчислень (ПОО-XXXIII)». — К.: Ін-т кібернетики ім. В.М. Глушкова НАН України, 2007. — С. 271–272.
9. Сергиенко И.В., Шило В.П. Проблемы дискретной оптимизации: сложные задачи, основные подходы к их решению // Кибернетика и системный анализ. — 2006. — № 4. — С. 3–25.
10. Achterberg T. Constraint integer programming. — Berlin: Techn. Univ., 2007 (http://opus.kobv.de/tuberlin/volltexte/2007/1611/).

Поступила 30.03.2008