АКАДЕМИЯ НАУК СССР
СИБИРСКОЕ ОТДЕЛЕНИЕ
ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР

ИССЛЕДОВАНИЯ ПО ПРИКЛАДНОЙ ТЕОРИИ ГРАФОВ

Ответственный редактор
акад. А. С. Алексеев

НОВОСИБИРСК
ИЗДАТЕЛЬСТВО «НАУКА»
СИБИРСКОЕ ОТДЕЛЕНИЕ
1986

Часть 3

АЛГОРИТМЫ УПАКОВКИ И ПОКРЫТИЯ В КОМБИНАТОРНЫХ ОБЪЕКТАХ

АЛГОРИТМ МАКСИМАЛЬНОЙ УПАКОВКИ НЕЧЕТНОПОЛЮСНЫХ РАЗРЕЗОВ И ЕГО ПРИЛОЖЕНИЯ

А. В. КАРЗАНОВ

Излагается алгоритм нахождения максимальной упаковки нечетнополюсных разрезов неориентированной сети, а также нахождения оптимального двойственного объекта — T-соединения минимального веса. Оценка числа действий алгоритма — $O(pm\log n + p^3\log p)$, где n, m и p — число вершин, ребер и полюсов сети. Известными приложениями являются задачи: «о китайском почтальоне», о нахождении отрицательного цикла взвешенного неориентированного графа, о допустимом многопродуктовом потоке в плоской неориентированной сети.

1. ВВЕДЕНИЕ

Под *графом* будем понимать конечный неориентированный граф без петель и кратных ребер. Множества вершин и ребер графа H будут обозначаться соответственно VH и EH. *Циклом* графа будем считать непустой связный подграф, все вершины которого имеют степень 2. *Цепь*, или *st-цепь* графа,— это непустой связный подграф без циклов, в котором только вершины s и t (*концы* цепи) имеют степень меньше 2 (допускается $s=t$ — случай вырожденной цепи).

Исходным объектом рассмотрения будет сеть (G, T, l), состоящая из связного графа G, подмножества его вершин $T\subseteq VG$, называемых *полюсами*, и функции $l\in \mathbf{R}_+^{EG}$ *длин* ребер графа G ($\mathbf{R}_+$ — множество неотрицательных вещественных чисел). Далее всегда будет предполагаться, что $|T|$ четно. Подграф J в G называется *T-соединением*, если множество вершин нечетной степени графа J совпадает с T (это близко к понятию T-join, введенному в [1]). Ясно, что всякое T-соединение представимо в виде объединения непересекающихся по ребрам цепей и циклов, причем концы цепей различны и образуют множество T. T-соединения возникают в связи с известной задачей «о китайском почтальоне» [2, 3] (см. также [4]), состоящей в нахождении в G замкнутого маршрута минимальной длины, проходящего через каждое ребро по крайней мере один раз. Легко показывается, что длина такого маршрута равна $l(EG)+l(EJ)$, где J есть T'-соединение минимальной длины для множества T' вершин в G, имеющих нечетную степень (здесь и далее для произвольного множества S, числовой функции q на S и подмножества $S'\subseteq S$ через $q(S')$ обозначается величина $\Sigma(q(e): e\in S')$).

Как оказывается, T-соединения связаны минимаксным соотношением с упаковками специальных разрезов сети. Пусть $\partial X=\partial^G X$, где $X\subseteq VG$ обозначает множество ребер в G с одним концом в X и другим в $VG-X$ (*разрез* графа G). Подмножество X, а также разрез ∂X называются *нечетнополюсными*, если $|X\cap T|$ нечетно; при $T=VG$ такие подмножество и разрез называются *нечетными*. Пусть $D=D(G, T)$ обозначает совокупность всех нечетнополюсных подмножеств для G и T. Для произвольной совокупности $D'\subseteq 2^{VG'}$ подмножеств вершин графа G' и функции $l'\in\mathbf{R}_+^{EG'}$ функция $f': D'\to\mathbf{R}_+$ называется *l'-упаковкой*, или просто *упаковкой*, если для соответствующего семейства разрезов выполняется условие

$$\lambda^{f'}(e)\overset{\text{def}}{=}\sum(f'(X): X\in D',\ e\in\partial^{G'}X)\leqslant l'(e)\qquad \forall e\in EG'. \quad (1)$$

Упаковка f' называется *максимальной* (при данном D'), если величина $\mathbf{1}\cdot f'=\Sigma(f'(X): X\in D')$ максимально возможная.

Нетрудно проверить справедливость следующего свойства: подграф J в G является T-соединением тогда и только тогда, когда $|EJ\cap\partial X|$ нечетно для всех $X\in D(G, T)$. Отсюда для l-упаковки $f: D(G, T)\to\mathbf{R}_+$ и T-соединения J имеем

$$\mathbf{1}\cdot f\leqslant\sum_{X\in D} f(X)\,|\partial X\cap EJ| = \sum_{e\in EJ}\sum(f(X): X\in D,\ e\in\partial X) = \sum_{e\in EJ}\lambda^f(e)\leqslant l(EJ).$$

Эдмондс и Джонсон доказали основную теорему о T-соединениях и упаковках нечетнополюсных разрезов.

Теорема 1 [5]. $\max \mathbf{1}\cdot f=\min l(EJ)$, *где максимум берется по всем l-упаковкам $f: D\to\mathbf{R}_+$, а минимум — по всем T-соединениям J в G.*

Доказательство этой теоремы следует из приведенного в [5] алгоритма нахождения оптимальных f и J. Анализ этого алгоритма показывает, что он может быть реализован с оценкой числа действий $O(n^4)$ при памяти $O(n^2)$, где $n=|VG|$. Кроме того, в случае целочисленной функции l алгоритм строит полуцелочисленную оптимальную упаковку (независимое доказательство существования полуцелочисленной оптимальной упаковки дано в [6]). Этот результат был усилен Сеймуром следующим образом. Функцию $l\in\mathbf{Z}_+^{EG}$ ($\mathbf{Z}_+$ — множество неотрицательных целых чисел) назовем *циклически четной*, если любой цикл C в G имеет четную длину $l(EC)$.

Теорема 2 [7]. *Если функция l циклически четная, то равенство в теореме 1 достигается для целочисленной упаковки $f: D \to \mathbf{Z}_+$.*

Следует отметить, что доказательство в работе [7] неалгоритмическое; в то же время в случае циклически четной функции l алгоритм в [5] гарантирует построение лишь полуцелочисленного, но не целочисленного оптимального f.

Задачу нахождения оптимальных f и J будем обозначать $\mathscr{P}(G, T, l)$. В настоящей работе излагаются алгоритм решения этой задачи (при произвольных $l \in \mathbf{R}_+^{EG}$), имеющий трудоемкость $O(pnm + p^4)$ действий, и его модификация трудоемкости $O(pm \log n + p^3 \log p)$ при памяти $O(m + pn)$, где $m = |EG|$ и $p = |T|$. Если при этом функция l оказывается циклически четной, алгоритм, как и его модификация, строит целочисленное оптимальное f (таким образом, будет получено алгоритмическое доказательство теоремы 2). Конструкция предлагаемого алгоритма выглядит более простой, чем в работе [5]. Здесь используется идея редукции, высказанная в [8, п. 5] для более широкого класса мультиразрезных задач. В данном случае задача $\mathscr{P}(G, T, l)$ редуцируется к $\mathscr{P}(K_T, T, h)$, где K_T — полный граф с множеством вершин T и $h(st)$, $s, t \in T$, — расстояние между полюсами s и t в графе G с длинами ребер l. В свою очередь $\mathscr{P}(K_T, T, h)$ фактически представляет собой вариант задачи о паросочетаниях и решается при помощи техники чередующихся цепей. Алгоритм описан в разд. 2 и 3. В разд. 4 изложена экономная модификация алгоритма, основанная на использовании логарифмической справочной изменяющегося упорядоченного множества.

В заключение укажем два известных приложения задачи $\mathscr{P}(G, T, l)$.

А. Пусть $U \subseteq EG$ — выделенное подмножество ребер в G, а Q — совокупность всех подмножеств $X \subset VG$ таких, что $|\partial X \cap U| = 1$. Сеймур рассмотрел задачу о существовании l-упаковки $f': Q \to \mathbf{R}_+$, удовлетворяющей $\lambda^{f'}(e) = l(e)$ для всех $e \in U$ (*задача* $\mathscr{A}(G, U, l)$).

Теорема 3 [9]. *Задача $\mathscr{A}(G, U, l)$ имеет решение тогда и только тогда, когда для любого цикла C в G справедливо*

$$l(EC \cap U) \leqslant l(EC - U), \qquad (\text{А})$$

иными словами, когда в графе G с функцией длин ребер w, определенной как $w(e) = l(e)$ $(e \in EG - U)$ и $w(e) = -l(e)$ $(e \in U)$, отсутствуют циклы C отрицательной длины $w(EC)$.

В [7] показано, что она сводится к задаче $\mathscr{P}(G, T, l)$, где T определяется как множество вершин, инцидентных нечетному числу ребер в U. А именно: пусть f и J — оптимальные решения последней задачи. Поскольку U порождает T-соединение для данного T, то $l(EJ) \leqslant l(U)$. Если окажется, что $l(EJ) < l(U)$, то задача $\mathscr{A}(G, U, l)$ не имеет решения (нетрудно видеть, что в подграфе, порожденном множеством ребер $(EJ - U) \cup (U - EJ)$, содержится цикл C, для которого нарушено неравенство (А)). Если же $l(EJ) = l(U)$, то f определяет решение задачи $\mathscr{A}(G, U, l)$ (из $\mathbf{1} \cdot f = l(U)$ следует: если $X \in D(G, T)$, $f(X) > 0$, то $|\partial X \cap U| = 1$, и что $\lambda^f(e) = l(e)$ для всех $e \in U$). Отметим также, что из теоремы 2 вытекает: если функция l циклически четная и задача $\mathscr{A}(G, U, l)$ имеет решение, то она имеет и целочисленное решение (в [10, п. 8] показано, что этот результат можно непосредственно получить из теоремы 3).

Таким образом, алгоритм настоящей работы позволяет решить задачу $\mathscr{A}(G, U, l)$ и, в частности, может быть использован для обнаружения отрицательного цикла в неориентированном взвешенном графе. В этом случае его модификация имеет трудоемкость $O(\min\{pm \log n, pn^2\} + p^3 \log p)$, где p — число вершин, инцидентных нечетному числу ребер отрицательной длины (понижение оценки объясняется тем, что не требуется строить l-упаковку f).

Б. Рассмотрим задачу о допустимом многопродуктовом потоке для плоского графа G, в котором выделено подмножество ребер $U \subseteq EG$, и на ребрах $e \in EG - U$ заданы пропускные способности $c(e) \geqslant 0$, а на ребрах $u \in U$ — потоковые требования $d(u) \geqslant 0$. Требуется построить в графе $(VG, EG - U)$ многопродуктовый поток $\{F_u : u \in U\}$ (F_u — поток, полюсами которого являются концы ребра u) так, чтобы величина каждого F_u была равна $d(u)$ и сумма потоков по каждому ребру $e \in EG - U$ не превышала $c(e)$. Пусть G^* и U^* — объекты, двойственные G и U относительно планарности; положим $l(e^*) = c(e)$ $(e \in EG - U)$ и $l(e^*) = d(e)$ $(e \in U)$, где e^* обозначает ребро, двойственное e. Нетрудно понять, что наша задача равносильна $\mathscr{A}(G^*, U^*, l)$, поэтому из теорем 3 и 2 вытекает

Теорема 4 [7]. *Для указанных G, U, c, d допустимый многопродуктовый поток существует тогда и только тогда, когда для любого $X \subset VG$ выполняется неравенство Форда — Фалкерсона*

$$c(\partial X - U) - d(\partial X \cap U) \geqslant 0. \qquad (\text{Б})$$

Если функции c и d целочисленные и для любого $X \subset VG$ величина в левой части (Б) *неотрицательная и четная, то существует целочисленный допустимый многопродуктовый поток.*

2. РЕДУКЦИЯ И ВОССТАНОВЛЕНИЕ

Пусть xy обозначает неупорядоченную пару вершин x и y в G. Если $x, y \in T$, $x \neq y$, пару xy будем отождествлять с ребром полного графа K_T, соединяющим x и y. Множество ребер в K_T будем обозначать E_T.

Пусть $\mu_l(xy)$ — расстояние между вершинами x и y графа G, т. е. минимум величин $l(EL)$ по всем xy-цепям в G (напомним, что граф G предполагается связным, следовательно, $\mu_l(xy)$ конечно). Очевидно, функция μ_l удовлетворяет неравенствам треугольника, т. е. является метрикой. Предлагаемый алгоритм состоит из трех *стадий*.

Первая стадия — вычисление расстояний $\mu_l(st)$ для всех $s, t \in T$. Ее трудоемкость — $O(pn^2)$, если считать, что для нахождения расстояний от каждого фиксированного полюса s до остальных используется алгоритм трудоемкости $O(n^2)$, например алгоритм Дейкстры. Если же для их вычисления применяется $O(m \log n)$-алгоритм из [11], то трудоемкость первой стадии будет $O(pm \log n)$.

Пусть h обозначает ограничение функции μ_l на множество E_T. Легко доказать следующее утверждение.

(2.1) *Если функция l циклически четная, то функция h тоже циклически четная (в графе K_T).*

На второй стадии алгоритма, которая изложена в разделе 3, найдено решение редуцированной задачи $\mathcal{P}(K_T, T, h)$. А именно: построены h-упаковка нечетных разрезов $g : D(K_T, T) \to \mathbf{R}_+$ и *совершенное паросочетание* M графа K_T, для которых $1 \cdot g = h(M)$. (Напомним, что паросочетание — это подмножество ребер графа, все концы которых различны; паросочетание называется совершенным, если оно покрывает все вершины графа; ясно, что подграф, порожденный совершенным паросочетанием в K_T, является T-соединением.) Кроме того, если функция h циклически четная, то построенная упаковка g будет целочисленной. Положим $Q = \{A \in D(K_T, T) : g(A) > 0\}$. Функция g будет обладать также следующим важным свойством (P): для любых различных $A, B \in Q$ выполняется либо $A \subset B$, либо $B \subset A$, либо $A \cap B = \varnothing$.

(4)

(5)

(6)

Третья стадия алгоритма (*процедура восстановления*) заключается в преобразовании g и M в l-упаковку $f : D(G, T) \to \mathbf{R}_+$ и T-соединение J в G, для которых будет выполнено

$$1 \cdot f = 1 \cdot g, \quad l(EJ) = h(M),$$

(7)

откуда следует $1 \cdot f = l(EJ)$, т. е. $\{f, J\}$ — решение задачи $\mathcal{P}(G, T, l)$.

Требуемое T-соединение J строится очевидным образом. Для каждого $st \in M$ надо взять st-цепь L^{st} в G, для которой $l(EL^{st}) = h(st)$ (можно взять кратчайшие цепи, найденные на первой стадии алгоритма). Выделим множество ребер в G, принадлежащих нечетному числу цепей L^{st}, $st \in M$, и пусть J — подграф, порождаемый этими ребрами. Легко видеть, что J является T-соединением и $l(EJ) \leqslant h(M)$ (в действительности это неравенство выполняется как равенство).

Задача построения требуемой упаковки f не столь проста. Она решается следующим алгоритмом (для него не существенно, что множество Q состоит из нечетных подмножеств в T, важно лишь, что Q удовлетворяет свойству (P)).

Алгоритм. Выберем в Q минимальное по включению множество A. Положим

$$X = \{x \in VG : \min\{\mu_l(sx) : s \in A\} = 0\};$$
$$a = \min\{g(A), \min\{l(e) : e \in \partial X\}\}.$$

Положим $f(X) = a$ и пересчитаем g и l как $g(A) := g(A) - a$ и $l(e) := l(e) - a (e \in \partial X)$ (на остальных элементах функции g и l остаются прежними). Если стало $g(A) = 0$, то удаляем A из множества Q. Если все еще $g(A) > 0$, то на следующем шаге берется то же самое множество A. Повторяем такие шаги до тех пор, пока текущее Q не станет пустым.

Пусть f — получившаяся функция (считающаяся продолженной нулем на элементах $X \in D(G, T)$, не появлявшихся на шагах алгоритма). Докажем, что f является искомой упаковкой нечетнополюсных подмножеств. Доказательство разбивается на ряд утверждений.

(2.2) *Справедливо* $X \cap T = A$.

Доказательство. Если $x \in A$, то ввиду $\mu_l(xx) = 0$ имеем $x \in X$. Если $x \in T - A$, то для любого $s \in A$

$$\mu_l(sx) = h(sx) \geqslant \lambda^g(sx) = \Sigma(g(B) : B \in Q,\ sx \in \partial B) \geqslant g(A) > 0,$$

следовательно, $x \notin X$.

Из утверждения (2.2) следует, в частности, что каждое определяемое в алгоритме множество X принадлежит $D(G, T)$.

(2.3) *Если l' и g' — функции, получающиеся из l и g в результате одного шага, то $\lambda^{g'}(pq) \leqslant \mu_{l'}(pq)$ для всех $pq \in E_T$ (т. е. функция g' является h'-упаковкой, где h' — ограничение $\mu_{l'}$ на E_T).*

Доказательство. Положим $\lambda = \lambda^{g'}$. Надо доказать, что для произвольных ребра $pq \in E_T$ и pq-цепи L в G справедливо $l'(EL) \geqslant \lambda(pq)$. Проведем индукцию по параметру $k(L) = |EL \cap \partial X|$ (при всех pq и L). (8)

А. Если $k(L) = 0$, то $l'(EL) = l(EL) \geqslant \mu_l(pq) \geqslant \lambda^g(pq) = \lambda(pq)$.

Б. Если $k(L) = 1$, то, очевидно, строго один из полюсов p и q принадлежит A. Следовательно, $l'(EL) = l(EL) - a$ и $\lambda(pq) = \lambda^g(pq) - a$, откуда получаем требуемое неравенство.

В. Если $p, q \in A$, то ввиду минимальности A и свойства (P) имеем $\lambda(pq) = \lambda^g(pq) = 0$, и неравенство очевидно.

Г. Пусть имеет место случай, отличный от рассмотренных в А — В. Тогда найдется вершина $x \in VL$ такая, что $x \in X$ $k(L') \geqslant 1$ и $k(L'') \geqslant 1$, где L' и L'' — отрезки цепи L от p до x и от x до q соответственно. По определению множества X существует полюс $s \in A$ такой, что $\mu_l(sx) = 0$. Выберем sx-цепь P, для которой $l(EP) = 0$; очевидно, $VP \subseteq X$. Выделим ps-цепь L_1 в графе $L' \cup P$ и sq-цепь L_2 в графе $L'' \cup P$. Из сказанного следует $k(L_i) < k(L)$, $i = 1, 2$, откуда по предположению индукции имеем $l'(EL_1) \geqslant \lambda(ps)$ и $l'(El_2) \geqslant \lambda(sq)$. Наконец, $l'(EL) \geqslant l'(EL_1) +$

$+ l'(EL_2)$ и $\lambda(pq) \leqslant \lambda(ps) + \lambda(sq)$, поскольку если $B \subset T$ и $pq \in \partial B$, то $\{ps, sq\} \cap \partial B \neq \varnothing$.

(2.4) *Пусть в результате шага осталось $g(A) > 0$, и пусть X' — множество, найденное на следующем шаге. Тогда $X \subset X'$.*

Доказательство следует из того, что по определению числа a для некоторого ребра $e \in \partial X$ в результате шага будет $l(e) = 0$.

Утверждения (2.2)—(2.4) обосновывают корректность алгоритма. Причем (2.4) показывает, что алгоритм конечен и оканчивается через не более чем $n|Q|$ шагов, а все найденные множества X различны. Утверждение (2.3) обосновывает корректность шага, и из него следует: для каждого $A \in Q$ сумма значений $f(X)$ по всем $X \in D(G, T)$ таким, что $X \cap T = A$, равна $g(A)$. Таким образом, $1 \cdot f = 1 \cdot g$.

Оценим трудоемкость алгоритма. Следующее утверждение легко доказывается индукцией по $|T|$ (см., например, [13]).

(2.5) *Если $Q' \subseteq 2^T$ — множество, в котором для любых различных $A, B \in Q'$ выполняется одно из соотношений $A \subset B$, $B \subset A$, $A \cap B = \varnothing$, $A \cup B = T$, то $|Q'| \leqslant 4|T| - 6$.*

Следовательно, алгоритм состоит из $O(pn)$ шагов. Нетрудно видеть, что каждый шаг может быть выполнен за $O(m)$ действий. Таким образом, трудоемкость третьей стадии — $O(pnm)$. Требуемая память — $O(pn)$, если считать, что множества X, найденные при работе с одним и тем же $A \in Q$, выдаются в виде отрезков единого (для фиксированного A) массива вершин.

3. АЛГОРИТМ ДЛЯ РЕДУЦИРОВАННОЙ СЕТИ

В дальнейшем граф K_T и множество его ребер E_T будем для удобства обозначать K и E. Текущими объектами алгоритма решения задачи $\mathscr{P}(K, T, h)$ (т. е. второй стадии всего алгоритма решения исходной задачи) будут паросочетание $M \subseteq E$ (не обязательно совершенное), подсемейство нечетных подмножеств $D \subset D(K, T)$ и h-упаковка $g \in \mathbf{R}_+^D$. Ребро $e \in E$ назовем *насыщенным*, если $\lambda^g(e) = h(e)$. Пусть M_A обозначает множество ребер в M, имеющих оба конца в подмножестве $A \subseteq T$. Для M, D, g всегда будут выполняться следующие свойства:

9 (P1) каждое ребро паросочетания M насыщенное;

10 (P2) D — правильное семейство. Это означает:

(а) для любых различных $A, B \in D$ либо $A \subset B$, либо $B \subset A$, либо $A \cap B = \varnothing$;

(б) $|M_A| = (|A| - 1)/2$ для любого $A \in D$.

Вершину $s \in T$ назовем *покрытой* паросочетанием M', если s инцидентно некоторому ребру в M'. Пусть r_A обозначает единственную вершину в $A \in D$, не покрытую паросочетанием M_A. Следующее утверждение достаточно очевидно.

(11) (3.1): (а) $|M \cap \partial A| \leqslant 1$ *для любого $A \in D$ и равенство выполняется тогда и только тогда, когда r_A покрыто M;*

(12) (б) *если паросочетание M совершенное, то $1 \cdot g = h(M)$.*

Алгоритм будет состоять из $|T|/2$ *этапов*. В начале алгоритма полагается $M = \varnothing$ и $D = \varnothing$. В начале очередного этапа произвольно выбирается вершина $r \in T$, не покрытая паросочетанием M, и на этапе M, D и g преобразуются таким образом, что к множеству покрытых вершин добавится r и еще одна вершина. Как только паросочетание M станет совершенным, текущая упаковка g (считающаяся продолженной нулем на $D(K, T) - D$) и данное M (точнее, порожденное им T-соединение в K) будут решением задачи $\mathscr{P}(K, T, h)$ ввиду (3.1), (б). Отметим, что в силу (P2), (а) для g будет выполнено указанное в предыдущем разделе свойство (P).

Вначале опишем дополнительные структуры, с которыми оперирует алгоритм. Обозначим через $V = V^D$ множество, элементами которого служат вершины в T и множества в D. Зададим на V частичный порядок, полагая $v \prec v'$, если либо $v \in T$, $v' \in D$ и $v \in v'$, либо $v, v' \in D$ и $v \subset v'$ (в частности, $s \prec \{s\}$, если $s \in T$ и $\{s\} \in D$). Заметим, что в силу свойства (P2), (а) любые два несравнимых элемента в V не содержат общих вершин. Элемент v назовем *подчиненным* элементу v', если $v \prec v'$, и *непосредственно подчиненным*, если нет такого v'', что $v \prec v'' \prec v'$. Множество максимальных элементов в V обозначим W, а множество максимальных элементов, непосредственно подчиненных элементу $A \in D$, как W_A. Для $s \in T$ через $w(s)$ (соответственно через $w_A(s)$ при $s \in A \in D$) будем обозначать единственный элемент $v \in W$ (соответственно $v \in W_A$), для которого $s \prec v$ или $s = v$. Определим мультиграф $F = F^D$, множеством вершин которого является W, а ребра взаимно однозначно соответствуют ребрам $st \in E$, для которых $w(s) \neq w(t)$. Аналогично определим мультиграф F_A, $A \in D$ с множеством вершин W_A, ребра которого суть $st \in E$ при $w_A(s), w_A(t) \in W_A$, $w_A(s) \neq w_A(t)$ (нам удобно считать — и это хорошо согласуется с программной реализацией алгоритма,— что ребрами мультиграфов F и F_A являются соответствующие ребра исходного множества E, т. е. можно одновременно писать $st \in E$, $st \in EF$ и $st \in EF_A$). Вершину v в F (F_A) назовем *простой*, если $v \in T$, и *сложной*, если $v \in D$.

Пусть M^D (M_A^D) обозначает множество ребер в M, принадлежащих мультиграфу $F(F_A)$. Из нечетности всех $A \in D$ и из (3.1), (а) вытекает следующее утверждение.

(13) (3.2): (а) $|W|$ *четно, а* $|W_A|$ *нечетно для всех* $A \in D$;

(14) (б) M^D (M_A^D) *является паросочетанием в* F *(соответственно в* F_A).

Цепь в $F(F_A)$ назовем *чередующейся*, если все ее ребра насыщенные и из любых двух ее смежных ребер одно принадлежит M. Подграф H в F назовем *чередующимся деревом*, с корнем $w(r)$, если $w(r) \in VH$, граф H связен и не содержит циклов; для каждого $v \in VH$ цепь $L(v)$ в H с концами v и $w(r)$ чередующаяся; для каждой вершины $v \in VH$, имеющей в H степень единица, цепь $L(v)$ содержит четное число ребер. Обозначим VH^+ (VH^-) множество вершин $v \in VH$, для которых $L(v)$ имеет четное (нечетное) число ребер, и положим $W^0 = W - VH$. Очевидно, $w(r) \in VH^+$, и степень (в H) любой вершины $v \in VH^-$ равна двум. В процессе алгоритма будет выполняться следующее дополнительное свойство:

15 (Р3) для каждого $A \in D$ с $|A| > 1$ в F_A имеется (и явно выделен) цикл C_A, содержащий все вершины в F_A и такой, что все его ребра насыщенные и $(|EC_A| - 1)/2$ из них принадлежит M.

Очевидно, для каждого $v \in W_A$ в C_A содержится чередующаяся цепь с концами $w_A(r_A)$ и v, имеющая четное число ребер; обозначим ее через $L_A(v)$ (если $|A| = 1$, положим $L_A(v) = (\{r_A\}, \varnothing)$). В процессе алгоритма поддерживаются числа $\lambda^g(e)$, $e \in E$.

Этап алгоритма состоит из последовательности шагов. Шаг заключается в выполнении одной из процедур П0 — П5; на первом шаге выполняется процедура П0. Основная работа на этапе связана с «выращиванием» в F чередующегося дерева H с корнем $w(r)$. Вначале полагается $H = (\{w(r)\}, \varnothing)$.

Процедура П0. Находим некоторое ребро $st \in E$ такое, что $\lambda^g(st) = h(st)$ и либо (а): $w(s) \in VH^+$, $w(t) \in W^0$, либо (б): $w(s)$, $w(t) \in VH^+$, $w(s) \neq w(t)$. В случае (а) переходим к процедуре П1, если $w(t)$ не покрыто паросочетанием M^D, и к процедуре П2, если покрыто. В случае (б) переходим к П3. Если в E нет указанных ребер, то находим в VH^- некоторую вершину $v = A \in D$, для которой $g(A) = 0$, и переходим к П4. Если такой вершины нет,— к П5.

Процедура П1 (*увеличение паросочетания*). Положим $r' = t$, если $w(t) = t$, и положим $r' = r_A$, если $w(t) = A \in D$ (согласно (3.1), (а) вершина r' не покрыта M). Выделяем в H чередующуюся цепь $L(w(s))$ и, добавляя к ней ребро st и вершину $w(t)$, получаем чередующуюся цепь L в F, соединяющую не покрытые паросочетанием M^D вершины $w(r)$ и $w(t) = w(r')$. Если L содержит некоторую сложную вершину $v = B$, то перестраиваем L в соответствующую чередующуюся цепь мультиграфа $F^{D'}$, где $D' = D - \{B\}$. Для этого заменяем «вершину» v подходящей чередущейся цепью $L_B(v')$ цикла C_B. Делая такие перестройки и далее, в конце получаем чередующуюся цепь $\tilde{L}$ исходного графа K, соединяющую r и r'. Пусть $\tilde{M} = M \cap E\tilde{L}$. Преобразуем M вдоль $\tilde{L}$, полагая $M := (M - \tilde{M}) \cup (E\tilde{L} - \tilde{M})$.

Процедура П1 завершает рассматриваемый этап. В результате мощность паросочетания M увеличивается на единицу и вершины r и r' становятся покрытыми. Легко проверить, что для имеющихся D и g и нового M свойства (Р1) — (Р3) остаются справедливыми.

Процедура П2 (*наращивание H*). Выделяем в M^D ребро, инцидентное $w(t)$; пусть это ребро $t'z \in E$, где $w(t') = w(t)$ (из свойств дерева H следует $w(z) \in W^0$). Наращиваем H, добавляя к нему вершины $w(t)$, $w(z)$ и ребра st, $t'z$. Возвращаемся к П0.

Процедура П3 (*редукция по нечетному циклу*). Добавляем к H ребро st и в получившемся графе выделяем цикл C (он составлен отрезками цепей $L(w(s))$ и $L(w(t))$ до первой общей вершины и ребром st и содержит $(|EC| - 1)/2$ ребер из M^D). Образуем новое нечетное подмножество $A = \{s' \in T : w(s') \in VC\}$ и полагаем $D := D \cup \{A\}$, $g(A) = 0$ и $C_A = C$. Новое дерево H получается из старого отождествлением вершин множества VC. Возвращаемся к П0.

Процедура П4 (*разрушение сложной вершины*). Выделяем в H два смежных ребра st и $t'z$, где $t'z \in M^D$, $w(t) = w(t') = A$ (очевидно, $t' = r_A$). В C_A выделяем цепь $L_A(w_A(t))$. Полагаем $D := D - \{A\}$ и преобразуем H, заменяя в нем вершину A цепью $L_A(w_A(t))$. Возвращаемся к П0.

Легко проверить, что в результате каждой из процедур П2—П4 свойства (Р1) — (Р3) остаются справедливыми, и дерево H продолжает быть корректным. Пусть $E^{0+} = \{st \in E : w(s) \in VH^+, w(t) \in W^0\}$ и $E^{++} = \{st \in E : w(s),\ w(t) \in VH^+,\ w(s) \neq w(t)\}$. Из описания процедуры П0 следует, что при переходе к процедуре Р5 выполняются следующие свойства:

(16) (а) для любого ребра $st \in E^{0+} \cup E^{++}$ справедливо $\lambda^g(st) < h(st)$;

(17) (б) для любой сложной вершины $A \in VH^-$ справедливо $g(A) > 0$.

Чуть позднее докажем, что верно следующее утверждение.

(3.3) *При переходе к процедуре* П5 *каждая вершина в* VH^- *сложная.*

Процедура П5 (*изменение* g). Вычисляем величину $\varepsilon = \min\{\varepsilon^{0+}, \varepsilon^{++}, \varepsilon^-\}$, где

$$\varepsilon^{0+} = \min\{h(st) - \lambda^g(st) : st \in E^{0+}\};$$

$$\varepsilon^{++} = \frac{1}{2}\min\{h(st) - \lambda^g(st) : st \in E^{++}\};$$

$$\varepsilon^- = \min\{g(A) : A \in VH^-\}$$

(из сказанного ранее следует, что $\varepsilon > 0$). Пополним совокупность D всеми такими одноэлементными подмножествами $\{s\}$, что s — простая вершина в VH^+, и преобразуем g, полагая $g(A) := g(A) + \varepsilon$ для всех $A \in VH^+$ и $g(A) := g(A) - \varepsilon$ для всех $A \in VH^-$. Соответствующим образом исправляем значения $\lambda^g(st)$, $st \in E$, полагая: $\lambda^g(st) := \lambda^g(st) + 2\varepsilon$, если $st \in E^{++}$; $\lambda^g(st) := \lambda^g(st) + \varepsilon$, если $st \in E^{0+}$; $\lambda^g(st) := \lambda^g(st) - 2\varepsilon$, если $w(s)$, $w(t) \in VH^-$, $w(s) \neq w(t)$; $\lambda^g(st) := \lambda^g(st) - \varepsilon$, если $w(s) \in VH^-$, $w(t) \in W^0$ (на остальных ребрах λ^g не изменяется). Возвращаемся к П0.

Можно убедиться, что в результате процедуры П5 новая функция g будет h-упаковкой, свойства (Р1) — (Р3) останутся справедливыми, а дерево H — корректным.

Корректность и сходимость алгоритма. Докажем вначале утверждение (3.3). Предположим, что некоторая вершина $v \in VH^-$ простая, т. е. $v \in T$. Поскольку $v \in VH^-$, в H имеются смежные ребра sv, $vt \in E$, для которых $w(s)$, $w(t) \in \in VH^+$ и $w(s) \neq w(t)$. Согласно условию перехода от П0 к П5 имеем $\lambda^g(st) < h(st)$. Поскольку $\{v\} \notin D$, то для всякого $A \in D$ из $\{sv, vt\} \cap \partial A \neq \varnothing$ следует $|\{sv, vt\} \cap \partial A| = 1$ и $st \in \partial A$, откуда получаем $\lambda^g(st) = \lambda^g(sv) + \lambda^g(vt)$. Но $\lambda^g(sv) = h(sv)$ и $\lambda^g(vt) = h(vt)$, следовательно, $h(st) > h(sv) + h(vt)$, что невозможно, так как h — метрика.

В результате выполнения П5 возникает по крайней мере одна из следующих ситуаций:

для некоторого $st \in E^{0+} \cup E^{++}$ становится $\lambda^g(st) = h(st)$,

для некоторого $A \in D$ становится $g(A) = 0$ (случай, когда все множества E^{0+}, E^{++}, VH^- пусты, невозможен, поскольку тогда H состояло бы из единственной вершины и было $|W| = 1$, вопреки (3.2), (a)).

Таким образом, за процедурой П5 следует П0 и затем одна из процедур П1 — П4, и, следовательно, алгоритм может окончиться только выполнением процедуры П1. Пусть N_i — число применений на этапе процедуры Π_i и $N = N_0 + \ldots + N_5$. Для текущих D и H положим $T^+ = \{s \in T : w(s) \in VH^+\}$ и $D^0 = \{A \in D : B \cap A = \varnothing$ для всех $B \in D$, $B \in VH^+\}$. Имеем $N_1 \leqslant 1$, $N_0 = N_1 + N_2 + N_3 + N_4 + N_5$ и $N_5 \leqslant N_1 + N_2 + N_3 + N_4$, следовательно, N оценивается как $O(N_2 + N_3 + N_4)$. Нетрудно убедиться в справедливости следующих фактов:

каждая из процедур П0 — П5 не уменьшает множество T^+ и не увеличивает множество D^0;

каждая из процедур П2 и П3 расширяет множество T^+;

каждая процедура П4 уменьшает множество D^0. Таким образом, $N_2 + N_3 \leqslant |T| - 1$, и N_4 не превосходит числа элементов в начальной (на этапе) совокупности D. Ввиду (2.5) N_4 оценивается как $O(p)$. Следовательно, число шагов этапа — $O(p)$.

Чтобы оценить число действий на шагах этапа, надо уточнить используемую в алгоритме структуру данных. Множество $V = T \cup D$ задается в виде списка элементов, в котором выделены массив T и список максимальных элементов W (элементы $A \in D$ задаются в виде «идентификаторов», а не подмножеств T). Для каждого $v \in V - W$ имеется ссылка на элемент $v' \in V$, которому непосредственно подчинено v, а также ссылка на элемент $w(v)$. Для каждого $A \in D$ выделен список непосредственно подчиненных элементов W_A. Графы H и C_A, $A \in D$, задаются списками их вершин (идентификаторов) в W и ребер в E и соответствующими инциденциями; остальные требуемые структуры задаются естественным образом. Нетрудно убедиться, что при такой организации данных каждая из процедур П1 — П4 реализуется с трудоемкостью O(p) действий, а каждая из процедур П0 и П5 — с трудоемкостью $O(p^2)$ (такую трудоемкость имеет перебор элементов из E и исправление λ^g). Таким образом, трудоемкость этапа составляет $O(p^3)$, а всего алгоритма решения задачи $\mathscr{P}(K_T, T, h)$ — $O(p^4)$ при памяти $O(p^2)$.

Нам осталось показать, что в случае циклически четной функции h алгоритм строит целочисленную оптимальную h-упаковку. Достаточно доказать, что процедура П5 сохраняет целочисленность текущей функции g, т. е. что величина ε^{++} целая. Для этого покажем, что величина $h(st) - \lambda^g(st)$ четная для любых s, $t \in T$ таких, что $w(s)$, $w(t) \in VH^+$, $w(s) \neq w(t)$. Образуем цикл C в F, добавив к ребру st отрезки цепей $L(w(s))$ и $L(w(t))$ до первой общей вершины. Последовательно заменяя в C сложные вершины $A \in D$ соответствующими чередующимися цепями в C_A (подобно тому, как это делается в процедуре П1), получаем цикл $\tilde{C}$ графа K. Все его ребра, кроме st, насыщенные, следовательно,

$$h(st) - \lambda^g(st) = h(E\tilde{C}) - \lambda^g(E\tilde{C}) = h(E\tilde{C}) - \\ - \Sigma(g(A)\,|E\tilde{C} \cap \partial A| : A \in D).$$

Теперь четность величины $h(st) - \lambda^g(st)$ следует из четности $h(E\tilde{C})$, четности всех $|E\tilde{C} \cap \partial A|$ и целочисленности g.

4. ЭКОНОМНАЯ МОДИФИКАЦИЯ

В основе модификации лежит один способ организации данных, позволяющий сократить переборы ребер на этапах второй и третьей стадий алгоритма. В результате трудоемкость второй стадии станет $O(p^3 \log p)$ вместо $O(p^4)$, а третьей — $O(pm \log n)$ вместо $O(pnm)$.

Пусть имеется текущая линейно упорядоченная совокупность объектов (с возможным равенством различных объектов), с которой производится следующая пошаговая работа. На очередном шаге из нее удаляются некоторые объекты и добавляются новые, при этом требуется, чтобы на любом шаге был известен текущий минимальный объект (произвольный, если их несколько). Организация такой совокупности в виде логарифмической справочной [12] позволяет выполнить эти операции за весь период работы с суммарной трудоемкостью $O(\eta \log \omega)$ действий при памяти $O(\omega)$, где η — общее число объектов, побывавших в совокупности за период работы, ω — максимальное число элементов, одновременно присутствующих в совокупности. (Для наших целей подходит и более просто устроенная справочная работы [11], дающая оценки трудоемкости $O(\eta \log \eta)$ и памяти $O(\eta)$.)

Рассмотрим этап работы с одним и тем же множеством A на третьей стадии алгоритма, используя обозначения из разд. 2. Будем считать, что для текущего множества X множество ребер $\partial^G X$ организовано в виде справочной $\mathscr{R}$, каждому ребру $e \in \mathscr{R}$ приписано число $q(e)$, определяющее упорядочение в $\mathscr{R}$, и имеется текущее число d. Числа d и $q(e)$, $e \in E$, в каждый момент таковы, что $q(e) - d$ равно текущему значению $l(e)$. В начале этапа в пустую справочную $\mathscr{R}$ заносятся ребра разреза $\partial^G A$, для которых полагается $q(e)=l(e)$, $d=0$. Пусть перед началом очередного шага $\mathscr{R}$ состоит из ребер разреза $\partial^G\tilde{X}$, где $\tilde{X}$ — соответствующее множество предыдущего шага. Ясно: чтобы получить множество X данного шага, надо выделить в $\mathscr{R}$ множество ΔE элементов e, для которых $q(e)-d=0$ (ими являются минимальные элементы в $\mathscr{R}$ ввиду $l\geqslant 0$), взять множество ΔZ тех концов ребер из ΔE, которые лежат в $VG-\tilde{X}$, и расширить ΔZ до множества Z вершин в $VG-\tilde{X}$, соединимых цепями нулевой длины с какими-либо вершинами из ΔZ; тогда $X=\tilde{X}\cup Z$.

Таким образом, новая справочная $\mathscr{R}$ (соответствующая разрезу $\partial^G X$) получается из старой удалением всех ребер с одним концом в Z и другим в $\tilde{X}$ и добавлением всех ребер с одним концом в Z и другим в $VG-X$; для этих новых ребер полагается $q(e)=l(e)+d$. Число a данного шага равно $q(e_0)-d$, где $q(e_0)$ — минимальный элемент в новом $\mathscr{R}$. Наконец, при переходе к следующему шагу полагается $d:=d+a$ (это соответствует уменьшению на a длин ребер в $\partial^G X$). Очевидно, каждое ребро $e\in EG$ может быть включено в $\mathscr{R}$ не более одного раза, следовательно, трудоемкость работы со справочной за весь этап будет $O(m\log m)$ или $O(m\log n)$. При переходе к следующему этапу (с новым A) справочная $\mathscr{R}$ полностью очищается; при всяком удалении из $\mathscr{R}$ ребра e восстанавливается его истинная текущая длина $l(e)=q(e)-d$.

Опишем теперь модификацию этапа алгоритма второй стадии. Отличия от изложенного в разд. 3 состоят в следующем.

1. В модификации не требуется поддерживать функцию λ^g. Пусть $V(v)$ и $T(v)$, где $v\in V$, обозначают соответственно множества $\{v\}\cup\{x\in V: x\prec v\}$ и $T\cap V(v)$, а $g^*(v)$ — величину $\Sigma(g(A): A\in D,\ v\preceqq A)$. Нетрудно видеть, что $\lambda^g(st)=g^*(s)+g^*(t)$ для любых s, $t\in T$ таких, что $w(s)\neq w(t)$. Заметим, что из утверждения (2.5) следует, что $|V(v)|$ имеет порядок $O(|T(v)|)$. Для произвольного $v\in W$ значения $g^*(x)$ для всех $x\in V(v)$ можно вычислить рекуррентно за $O(|V(v)|)$ или $O(|T(v)|)$ действий. Таким образом, для произвольных v, $v'\in W$, $v\neq v'$, величины $\lambda^g(st)$ для всех ребер $st\in E$ с $s\in T(v)$ и $t\in T(v')$ можно определить за $O(|T(v)||T(v')|)$ действий.

2. Ребра текущего множества E^{++} организованы в виде справочной $\mathscr{R}^{++}$. Ребрам st в $\mathscr{R}^{++}$ приписаны числа $b(st)$ (определяющие упорядочение в справочной), а справочной в целом — число d; эти числа таковы, что $b(st)-d$ равно текущей величине $h(st)-\lambda^g(st)$. Для каждого $s\in T$ такого, что $w(s)\in VH^-\cup W^0$, ребра st, для которых $w(t)\in VH^+$, организованы в виде справочной $\mathscr{R}(s)$. Ребрам st в $\mathscr{R}(s)$ приписаны числа $c(st)$ (определяющие упорядочение в справочной), а справочной в целом — число $d(s)$; при этом $c(st)-d(s)$ равно текущей величине $h(st)-\lambda^g(st)$. В начале этапа справочные создаются заново, и все числа d и $d(s)$ полагаются равными нулю. Как только ребро st удаляется из справочной, для него за $O(1)$ действий восстанавливается текущее значение $\lambda^g(st)$.

3. Первая часть процедуры П0 сводится к просмотру минимальных элементов справочных $\mathscr{R}^{++}$ и $\mathscr{R}(s)$ для всех $s\in T$ таких, что $w(s)\in W^0$. Среди них ищется элемент st, для которого $b(st)-d=0$ при $st\in\mathscr{R}^{++}$ или $c(st)-d(s)=0$ при $st\in\mathscr{R}(s)$. Таким образом, трудоемкость однократного выполнения процедуры П0 — $O(p)$. При процедуре П5 число ε^{++} определяется как $(b(st)-d)/2$, а ε^{0+} — как $\min\{c(st)-d(s): s\in T,\ w(s)\in W^0\}$, где st — минимальный элемент соответствующей справочной. Изменению функции g на этой процедуре соответствует корректировка d и $d(s)$ для всех $s\in T$, $w(s)\in W^0$, а именно: $d:=d+2\varepsilon$, $d(s):=d(s)+\varepsilon$.

4. После каждой из процедур П2 — П4 следует производить корректировку справочных $\mathscr{R}^{++}$ и $\mathscr{R}(s)$. Опишем ее на примере процедуры П4, состоящей в разрушении сложной вершины $A\in VH^-$. Пусть $\tilde{D}=D-\{A\}$, $\tilde{W}=(W-\{A\})\cup W_A$ и $\tilde{H}$ — объекты, получающиеся из D, W и H в результате этой процедуры. Пусть, далее, $W_A^+=W_A\cap V\tilde{H}^+$, $W_A^-=W_A\cap V\tilde{H}^-$ и $W_A^0=W_A\cap\tilde{W}^0$; ясно, что $V\tilde{H}^+=VH^+\cup W_A^+$, $V\tilde{H}^-=(VH^- -\{A\})\cup W_A^-$, $\tilde{W}^0=W^0\cup W_A^0$. Вычислим значения $\lambda^g(st)$ для всех s, $t\in T$ таких, что $s\preceqq v'$ и $t\preceqq v''$, где v' пробегает множество $V\tilde{H}^-\cup\tilde{W}^0$, а v'' — множество W_A^+ (согласно сказанному выше, это требует числа действий, пропорционального числу таких ребер st). Теперь для каждого $s\in T$ такого, что $s\preceqq v'$, $v'\in V\tilde{H}^-\cup\tilde{W}^0$, надо внести в $\mathscr{R}(s)$ все ребра $st\in E$ для $t\preceqq v''$, $v''\in W_A^+$, положив $c(st)=h(st)-\lambda^g(st)+d(s)$. Кроме того, для каждого $s\in T$ такого, что $s\preceqq v''$, $v''\in W_A^+$, все элементы $st\in E$, для которых $t\preceqq x$, $x\in VH^+$, следует перенести из справочной $\mathscr{R}(s)$ в справочную $\mathscr{R}^{++}$, положив $b(st)=c(st)-d(s)+d$; после этого справочная $\mathscr{R}(s)$ перестает существовать.

Учитывая тот факт, что в процессе этапа множество $T^+=\{s\in T: w(s)\in VH^+\}$ не уменьшается (см. разд. 3), каждое ребро $st\in E$ может только один раз включиться в справочную $\mathscr{R}(s)$ или $\mathscr{R}(t)$ и может только один раз включиться в справочную $\mathscr{R}^{++}$. Следовательно, суммарная трудоемкость работы со всеми справочными на этапе — $O(p^2\log p)$. Отсюда получаем, что этап имеет трудоемкость $O(p^2\log p)$, а вся вторая стадия — $O(p^3\log p)$ при памяти $O(p^2)$. Это дает нам указанную оценку для модифицированного алгоритма решения исходной задачи $\mathscr{P}(G, T, l)$.

ЛИТЕРАТУРА

5. **Edmonds J., Johnson E. L.** Matchings, Euler tours and Chinesē postman.— Math. Programming, 1973. N 5, p, 88—124.
6. **Lovasz L.** 2-matchinfs and 2-covers of hypergraphs.— Acta Match. Acad. Aci. Hungaricae, 1975, N 26, p. 433—444.
7. **Seymour P. D.** On old cuts and plane multicommodity flows.— Proc. London Math Soc., 1981, N 42, v. 3, p. 178—192.

10. Karzanov A. V. A generalised MFMC-property and multicommodity cut problems.— In: Finite and Infinite Sets (Proc. of the 6th Hungarian Combinatirial Colloquium, Eger, Hungary, 1981), v. 2/Ed. A. Hajnal, L. Lovasz. V. T. Sos, Amsterdam: North-Holland, 1984, p. 443—486.
9. Seymour P. D. Sums of circuits.— In: Graph Theory and Related Topics/Ed. J. A. Bondy, U. S. R. Murty.— N. Y.: Acad. Press, 1978, p. 341—355.
8. Карзанов А. В. Задачи о мультиразрезах и методы их решения.— М.: ВНИИСИ, 1982.
11. Карзанов А. В. Справочная для выборки максимального элемента и ее приложения.— В кн.: Исследования по дискретной оптимизации/Под ред. А. А. Фридмана. М.: Наука, 1976, с. 348—359.
13. Карзанов А. В. Комбинаторные способы решения разрезных задач о мультипотоках.— В кн.: Комбинаторные методы в потоковых задачах.— М.: ВНИИСИ, 1979, вып. 3, с. 6—69.
2. Адельсон-Вельский Г. М., Ландис Е. М. Один алгоритм организации информации.— Докл. АН СССР, 1962, т. 146, № 2.
4. Кристофидес Н. Теория графов.— М.: Мир, 1978.
12. Edmonds J. The Chinese postman problem.— Operations Research. 13 Suppl., 1965, N 1, p. 373.
3. Mei-Ko K. Graphic Programming using odd or even points.— Chinese Mathematics, 1962, N 1. p. 273—277.
1. Seymour P. D. On multi-colourings of cubic graphs, and conjectures of Fulkerson and Tutte.— Proc. London Math. Soc., 1979, 9(3), N 38, p. 423—460.

ЭКСТРЕМАЛЬНЫЕ МНОЖЕСТВА И ЗАДАЧИ ПОКРЫТИЯ И УПАКОВКИ В МАТРОИДАХ

А. К. КЕЛЬМАНС, В. П. ПОЛЕССКИЙ

1. ВВЕДЕНИЕ

Работа посвящена исследованию различных аспектов задач покрытия и упаковки в матроидах [1], а именно: взаимосвязи этих задач между собой и с задачами построения базы суммы матроидов, алгоритмов их решения, многообразия возможных решений и т. д. Основные понятия и обозначения, а также некоторые сведения из теории матроидов изложены в разд. 2.

В 1971 г. А. К. Кельманс предложил конструктивное описание цикла суммы матроидов и понятие увеличивающего пути для увеличения независимого множества суммы матроидов [2]. Эти понятия и результаты изложены в разд. 3 и играют важную роль в дальнейших рассуждениях. В частности, видно, что из конструкции цикла суммы матроидов непосредственно следуют:

теоремы Нэш-Вильямса о ранге суммы матроидов, а также о том, что сумма матроидов есть матроид;

алгоритм отыскания базы суммы нескольких матроидов, а значит, и решения задач упаковки и покрытия, требующий $O(|E|)^3$ обращений к оракулам независимости.