

ДИСКРЕТНЫЕ ЗАДАЧИ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

А. А. Корбут, Ю. Ю. Финкельштейн

§ 1. ВВЕДЕНИЕ

Как известно, общая задача математического программирования заключается в максимизации

$$f(X) \tag{1.1}$$

при условиях

$$g(X) \leq 0, \tag{1.2}$$

$$X \in D. \tag{1.3}$$

Здесь $X = (x_1, \dots, x_n)$, f — скалярная функция, $g(X) = (g_1(X), \dots, g_m(X))$, D — некоторая область пространства R^n (чаще всего в качестве D берется неотрицательный ортант).

Среди задач (1.1) — (1.3) можно выделить класс регулярных задач, т. е. задач, в которых выполняются следующие условия.

1) Для каждой точки $X \in G$ (область G определяется условиями (1.2), (1.3)) можно определить некоторым образом понятие непустой окрестности G_X , состоящей из точек, принадлежащих G .

2) Можно указать достаточно легко проверяемые необходимые и достаточные условия локальной оптимальности. На основе этих условий локальный оптимум целевой функции (1.1) на множестве G может быть найден при помощи некоторого конечного (или бесконечного сходящегося) процесса.

3) Локальный оптимум целевой функции совпадает с глобальным.

К регулярным задачам относятся, например, задачи выпуклого программирования (функция f — вогнута, функции g_i — выпуклы) и, в частности, задачи линейного программирования (функция f и функции g_i линейны, D есть неотрицательный ортант). В решении регулярных (прежде всего линейных, а также выпуклых) задач математического программирования достигнуты значительные успехи. При решении же нерегулярных задач возникают большие трудности.

К задачам, не являющимся регулярными, относятся, в частности, так называемые многоэкстремальные задачи, т. е. задачи, в которых глобальный экстремум может не совпадать с локальным (многоэкстремальным задачам уделено много места в обзоре Д. Б. Юдина [69, 70]).

Другим обширным классом нерегулярных задач, привлекающим к себе в последние годы все большее внимание, является класс дискретных задач, в которых множество D не является связным (например, множество D может быть конечным или счетным, либо декартовым произведением конечного или счетного множества на континуальное).

Разработки в этом направлении, начатые меньше 15 лет тому назад, к настоящему времени продвинуты настолько далеко, что сейчас мы уже вправе говорить о самостоятельном разделе математического (оптимального) программирования — дискретном программировании.

В литературе употребляются также термины «целочисленное программирование» и реже «комбинаторное программирование». Однако термин «дискретное программирование» представляется наиболее полно отражающим существо вопроса, хотя здесь и возникает некоторая опасность смещения этой дисциплины, скажем, с дискретным анализом. Впрочем, эта опасность является скорее кажущейся (сейчас уже никто не путает программирование с линейным программированием).

В терминах дискретного программирования формулируются многие важные задачи экономики, управления, планирования, военного дела, биологии и т. п. Кроме того, к задачам дискретного программирования удается свести ряд экстремальных комбинаторных задач. Однако при решении задач дискретного программирования возникает ряд существенных и специфических затруднений. Эти затруднения носят не только технический, но и принципиальный характер. Дело в том, что в отличие от ставших уже почти классическими регулярных задач математического программирования, область допустимых решений в дискретных задачах является невыпуклой и несвязной. Эта область определяется условиями двух типов —

«регулярными» условиями (1.2) и «дискретными» (или частично «дискретными») условиями (1.3). Условия дискретности (или частичной дискретности), как правило, отделены от других условий и, кроме того, отсепарированы по отдельным переменным, так что их можно записать следующим образом

$$x_j \in D_j, \quad j = 1, \dots, n_1. \quad (1.4)$$

Здесь $n_1 \leq n$, а каждое D_j — либо конечное множество, содержащее не менее двух элементов, либо счетное множество.

Если $n_1 = n$, то естественно говорить о полностью дискретной задаче, если же $n_1 < n$, то задачу можно назвать частично дискретной.

В простейшем случае, когда $n_1 = n$ и все D_j — конечные множества, задача дискретного программирования представляет собой задачу нахождения условного экстремума на конечном множестве.

Невыпуклость и несвязность области допустимых решений делают невозможным применение к дискретным задачам обычных приемов «регулярного» математического программирования (продвижение из одной вершины многогранника в другую, перемещение по градиенту в окрестности данной точки и т. п.). Идея перебора для задач с конечным множеством допустимых решений также оказывается практически неприменимой, так как количество точек (x_1, \dots, x_n) , удовлетворяющих условию (1.4) (при $n_1 = n$), равно

$$\prod_{j=1}^n |D_j| \geq 2^n,$$

$|D_j|$ — количество элементов, содержащихся в множестве D_j) и с увеличением количества переменных объем вычислительной работы растет весьма быстро. Поэтому для решения задач дискретного программирования требуется создание совершенно новых, специализированных методов.

В настоящей статье дается обзор моделей дискретного математического программирования и основных численных методов. Приводятся также сведения о машинном решении дискретных задач. Обзор написан по материалам литературы за 1958—1966 гг. Мы не упоминаем здесь о ряде, по-видимому, интересных работ, опубликованных в ведомственных зарубежных сборниках, фирменных отчетах и т. п., поскольку подобные издания практически недоступны для советского читателя.

Нами принят следующий порядок изложения: § 2 посвящен моделям задач дискретного программирования. В § 3 дан обзор основных идей дискретного программирования и класси-

ранце. Имеется n объектов весом a_1, \dots, a_n . Ценность каждого объекта составляет c_1, \dots, c_n . Требуется загрузить ранец вместимости A набором грузов с максимальной суммарной ценностью. Если ввести переменные x_1, \dots, x_n , имеющие следующий смысл

$$x_j = \begin{cases} 1, & \text{если } j\text{-й объект подлежит загрузке,} \\ 0 & \text{в противном случае,} \end{cases} \quad (2.5)$$

то задача о ранце сведется к максимизации

$$c_1 x_1 + \dots + c_n x_n \quad (2.6)$$

при условиях

$$x_j = \begin{cases} 1 \\ 0 \end{cases}, \quad j = 1, \dots, n, \quad (2.7)$$

$$a_1 x_1 + \dots + a_n x_n \leq A. \quad (2.8)$$

Задачи с фиксированными доплатами. В этих задачах требуется минимизировать функцию

$$c_1(x_1) + \dots + c_n(x_n), \quad (2.9)$$

где

$$c_j(x_j) = \begin{cases} 0 & , \text{ если } x_j = 0, \\ c_j x_j + d_j, & \text{ если } x_j > 0 \end{cases} \quad (2.10)$$

при линейных ограничениях на ее переменные. Такого рода задачи могут быть сформулированы как частично целочисленные задачи линейного программирования.

Из задач этого типа более подробно рассматривалась в литературе транспортная задача с фиксированными доплатами (см. работы И. В. Гирсанова и Б. Т. Поляка [7], А. А. Корбута [27], А. А. Корбута и В. В. Малинникова [30, 31], Л. Е. Каменецкого и А. А. Корбута [23], А. А. Корбута, Л. Е. Каменецкого и Б. М. Вайсбурда [29], Р. А. Поляка [46], Балинского [77], Куна и Баумоля [174], Шпильберга [207]).

Задачи линейного программирования на неклассических областях. Сведение к целочисленным или частично целочисленным задачам линейного программирования позволяет рассматривать задачи нахождения экстремумов линейных функций на невыпуклых или несвязных областях, задаваемых, помимо обычных линейных ограничений, дополнительными альтернативными условиями типа «либо—либо» или логическими условиями типа «если $F(X) > 0$, то $G(X) < 0$ » (здесь F и G — линейные формы). С постановками такого рода задач можно ознакомиться в монографии Данцига [10], в обзорной статье А. А. Корбута [28] и в монографии Е. Г. Гольштейна и Д. Б. Юдина [8]; общая задача линейного программирования с дополнительными логическими условиями рассматривалась Радо [198, 199].

Задачи линейного программирования с ограничением на общее число ненулевых переменных. В обычных задачах линейного программирования (без требования целочисленности) иногда может накладываться дополнительное требование, заключающееся в том, что в окончательном решении отличными от нуля должны быть не более k переменных. Такие задачи и различные их варианты могут быть сведены к частично целочисленным задачам линейного программирования (см. статью Динкельбаха и Стеффенса [110] и обзор А. А. Корбута [28]).

Из сказанного можно усмотреть, что целочисленными моделями линейного программирования охватывается широкий круг задач, могущих в той или иной мере выходить за рамки линейного программирования и примыкающих скорее к задачам комбинаторного типа. И действительно, анализ некоторых экстремальных комбинаторных задач с использованием аппарата линейного программирования послужил одним из первых толчков к развитию целочисленного программирования.

К этому классу относятся задачи, в которых требуется найти экстремум целочисленной линейной функции, заданной на конечном множестве элементов, либо же сами элементы этого конечного множества, обладающие указанным экстремальным свойством. Для того, чтобы «погрузить» подобную задачу в задачу линейного программирования, обычно интерпретируют элементы конечного множества как целочисленные точки евклидова пространства и ищут экстремум линейной функции на выпуклой оболочке этих точек или на более широком выпуклом многограннике. Решение получившейся задачи линейного программирования будет иметь «обратную» комбинаторную интерпретацию только в случае своей целочисленности. Однако в рассмотренных первоначально задачах требование целочисленности не вносило никакой дополнительной специфики, ибо большинство этих задач сводится к задачам типа транспортной. Последние же, как отметил еще Данциг, при целочисленных исходных данных всегда обладают целочисленным решением. Из такого рода задач отметим, задачу о выборе системы различных представителей или теорему Дилворта о частично упорядоченных множествах, рассмотренные с точки зрения линейного программирования в сборнике [33].

Близкими к этим задачам являются разнообразные задачи о потоках в транспортных сетях. Изящное изложение теории транспортных потоков дано в книге Гейла [6]; см. также книгу Форда и Фулкерсона [58] и обзорную статью Фулкерсона [120].

Отмеченное выше свойство целочисленности решений транспортной задачи связано со специфической структурой ее матрицы ограничений. В связи с этим возникает проблема ха-

рактеризации класса задач линейного программирования, имеющих целочисленные решения при целых исходных данных. Эта проблема была исследована Гофманом и Краскалом [9]. Их результат сводится к следующему: для того чтобы задача линейного программирования с ограничениями

$$AX \leq b, 0 \leq X \leq d$$

(матрица A , векторы b и d целочисленны) имела только целочисленные базисные решения при любых векторах b и d (и фиксированной матрице A), необходимо и достаточно, чтобы любой из миноров матрицы A был равен нулю или ± 1 .

Тем самым класс задач, обладающих указанным свойством (их естественно назвать «унимодулярными» задачами), оказывается весьма узким. Поэтому для решения задач, не являющихся унимодулярными, необходимо создание совершенно новых методов. О них будет идти речь в следующем параграфе.

Следует упомянуть еще о двух важных в прикладном отношении комбинаторных задачах, также сводящихся к моделям целочисленного линейного программирования. Это задача о бродячем торговце (задача коммивояжера) и общая задача теории расписаний. При всем самостоятельном интересе, который представляют эти задачи, они все же лежат в стороне от предмета настоящего обзора (соответствующие целочисленные линейные модели приведены в монографии Е. Г. Голыштейна и Д. Б. Юдина [8] и в обзоре А. А. Корбута [28]). Ряд обзоров по теории расписаний опубликован В. В. Шкурбой [65, 66, 67, 68].

Значительное внимание уделяется последнее время так называемым задачам о покрытии (см. обзор Балинского [78]). Простейшая задача о покрытии — это задача о построении минимального подмножества множества ребер графа, содержащего все вершины графа. Более общей является следующая задача.

Минимизировать

$$\sum_{j=1}^n c_j x_j \quad (2.11)$$

при условиях

$$\sum_{j=1}^n a_{ij} x_j \geq 1, \quad i = 1, \dots, m, \quad (2.12)$$

$$x_j = 0 \text{ или } 1, \quad j = 1, \dots, n. \quad (2.13)$$

Здесь $c_j > 0$, $j = 1, \dots, n$, $a_{ij} = 0$ или 1.

Задачи подобного типа возникают в дискретном анализе и при синтезе логических сетей. Упомянем работу В. К. Коробкова [32], устанавливающую связь между монотонными функциями алгебры логики и задачей целочисленного линейного программирования, работу Т. Л. Майстровой [35], в которой дана целочисленная линейная модель для задачи минимизации нормальных форм булевых функций, а также работы, посвященные оптимальному синтезу логических сетей: Брейера [93], Карпа и др. авторов [167]. Задачу типа (2.11)—(2.13) рассматривал также Нельсон [193]. Задачам оптимального кодирования посвящены работы Карпа [166] и Мак-Класки [189].

Из других комбинаторных задач, поддающихся трактовке с помощью целочисленного линейного программирования, назовем задачу об ортогональных латинских квадратах и задачу о четырех красках. Эти вопросы освещены в монографии Данцига [10].

Задачу линейного программирования с дополнительным требованием дискретности всех или некоторых переменных рассматривали Дальтон [100] и Дальтон и Ллевелин [101].

Наконец, следует отметить, что в терминах дискретного (в основном — целочисленного линейного) программирования сформулировано много прикладных задач: задача размещения производства (см., например, работу Л. Е. Минца и Ю. Ю. Финкельштейна [37]), задача оптимизации комплекса технических средств доставки грузов речным транспортом (Б. А. Атлас, С. С. Гдалевич [1]), задача оптимизации системы телемеханики для комплексов с сосредоточенными объектами (Г. А. Каплан [24]), задача оптимальной расстановки рыболовецких судов по районам промысла (М. П. Москаленко, Н. С. Кошечева [41]), некоторые задачи оптимального проектирования (И. Б. Моцкус [42]), задача оптимизации избыточности (А. Л. Райкин [48]), задача распределения сверхурочных часов при проектировании сезонных предприятий (В. И. Шембель [64]), задача оптимального извлечения информации из параллельных систем памяти (Дэй [108]), сельскохозяйственные задачи (Эдвардс [112]), задача минимизации количества судов для обеспечения заданного плана перевозок (д'Эзопо, Лефковиц [114]), задача оптимального определения границ избирательных округов (Гесс и др. [155]), задача оптимального распределения машинной памяти (Келли [169]), задача минимизации порожнего пробега автомашин, выполняющих заданный план перевозок (Креко [172]), задача синхронизации сигналов уличного движения (Литтл [184]). Этот список можно было бы продолжить.

§ 3. ПОДХОДЫ К ЗАДАЧАМ ДИСКРЕТНОГО ПРОГРАММИРОВАНИЯ

Теория дискретного математического программирования на сегодняшний день состоит в основном в теории численных методов решения дискретных задач. Давая их общую характеристику, можно прежде всего выделить три группы методов, принципиально отличающихся по подходу к проблеме.

Для методов первой группы характерна прежде всего «регуляризация» задачи, достигаемая погружением ее исходной области допустимых решений (1.2), (1.3) в объемлющую ее выпуклую область (иными словами, временным отбрасыванием условий дискретности (1.3)). К получившейся регулярной задаче затем применяются стандартные методы оптимизации. Если получившееся в результате решение уже удовлетворяет требуемым условиям дискретности — задача решена. Если же это не так, то требуется дальнейший переход к целочисленному решению. Следует иметь в виду, что этот переход не может, вообще говоря, быть достигнут простым округлением компонент полученного нецелочисленного решения; действительно, можно построить простые примеры задач целочисленного линейного программирования, в которых любые варианты округления нецелочисленного оптимального решения не только не дают целочисленного оптимума, но, более того, выводят за пределы исходной допустимой области.

Таким образом, указанный переход и составляет самую суть методов этой группы. Впервые идея такого перехода была указана в работе Данцига, Фулкерсона и Джонсона [107] (см. также работу тех же авторов [106]) применительно к задаче о бродячем торговце; позднее аналогичная идея использовалась для другой конкретной задачи Марковицем и Манном [187]. Для общей целочисленной задачи линейного программирования соответствующая идея была высказана Данцигом [103]. Она заключается в следующем. Если в результате первого шага (решение задачи без учета требования целочисленности) получено нецелочисленное решение, то к ограничениям исходной задачи добавляется новое линейное ограничение, обладающее двумя свойствами:

- 1) полученное нецелочисленное решение ему не удовлетворяет;
- 2) искомое целочисленное решение ему заведомо удовлетворяет.

Затем решается полученная расширенная задача, в случае необходимости добавляется еще одно ограничение и т. д. Процесс повторяется до получения решения, обладающего нужными свойствами целочисленности. Таким образом, решение полностью или частично целочисленной задачи линейного про-

граммирования сводится к решению некоторой последовательности обычных задач линейного программирования.

Геометрически добавление каждого такого линейного ограничения отвечает проведению гиперплоскости, отсекающей от многогранника решений «регуляризированной» задачи оптимальную точку с дробными координатами, но не затрагивающей ни одной из целочисленных точек этого многогранника. Поэтому методы, опирающиеся на эту идею, получили в литературе название «методов отсекающих плоскостей», или «методов отсечения».

Идея отсечения естественным образом приводит к двум проблемам:

1. нахождение универсального правила формирования дополнительных ограничений;
2. доказательство конечности соответствующего процесса отсечения.

Только решение этих двух проблем может привести к универсальному и реализуемому в вычислительном отношении алгоритму. Впервые это было сделано Гомори в 1958 г. (первая публикация алгоритма [139]; см. также [140, 145, 146]). Именно с этого момента начинается развитие общих методов отсечения.

В дальнейшем Гомори обобщил идею формирования дополнительных ограничений и получил другую форму алгоритма для полностью целочисленных задач (первая публикация алгоритма [142]; см. также [140, 143]). Этот метод особенно удобен в вычислительном отношении, ибо он требует лишь выполнения действий сложения и вычитания. В литературе (см., например, Симоннар [205]) эти методы иногда называются соответственно «циклическим» и «дискретным» алгоритмами Гомори, мы также будем пользоваться этими терминами. Кроме того, Гомори распространил первый из своих алгоритмов на частично целочисленные задачи линейного программирования («смешанный» алгоритм Гомори, см. [141]). Важно отметить, что доказана также конечность этих трех алгоритмов *).

Нетривиальность проблемы отсечения хорошо иллюстрируется на примере следующего приема формирования отсекающих ограничений, предложенного Данцигом [104] для полностью целочисленной задачи линейного программирования вскоре после первой работы Гомори. Этот прием основан на следующем простом факте. Если имеется нецелочисленное

*) Более подробно о конечности алгоритмов Гомори сказано в следующем параграфе.

оптимальное решение задачи линейного программирования x_j , $j \in B$, (B — множество индексов, нумерующих базисные переменные), то все ее целочисленные решения должны удовлетворять неравенству

$$\sum_{j \in B} x_j \geq 1. \quad (3.1)$$

Полученное же решение этому неравенству не удовлетворяет, ибо $x_j = 0$ для $j \notin B$. Таким образом, соответствующее (3.1) уравнение определяет отсекающую плоскость.

Этот процесс формирования дополнительных ограничений существенно проще способа Гомори. Однако конечность алгоритма с ограничениями вида (3.1) Данцигу установить не удалось. Этот вопрос был исследован Гомори и Гофманом [148]. Они выяснили, что такой алгоритм не будет конечным (или даже вообще не будет сходиться к оптимальному решению), если нарушены некоторые необходимые условия. Таким образом, для весьма широкого класса задач этот упрощенный прием оказывается непригодным.

Развитие методов отсекающего в последние годы шло как по пути их усовершенствования и детализации, так и по пути их распространения на более сложные классы задач. Мартин [188] предложил модификацию «циклического» алгоритма Гомори, дающую значительное ускорение сходимости (см. обзор Балинского [78]). Ю. Ю. Финкельштейн [54, 55] предложил специализированный алгоритм для решения задач целочисленного линейного программирования с булевыми переменными; им же предложен прием для усиления отсекающих ограничений Гомори (см. [56]). В работах Бен-Израэля и Чарнса [87], Гарриса [152] и Юнга [219] предложены «прямые» алгоритмы, реализующие метод отсекающего (в них используется прямой симплекс-метод в отличие от алгоритмов Гомори, в которых используется двойственный симплекс-метод). Нельсон [193] предложил прямой алгоритм для решения задач типа задачи покрытия. В работах Дальтона [100] и Дальтона и Ллевелина [101] «смешанный» алгоритм Гомори обобщается на частично дискретную задачу линейного программирования. Гарвер [122] предлагает алгоритм, практически совпадающий с «дискретным» алгоритмом Гомори. Гловер [134, 135] предлагает некоторые новые реализации метода отсекающего. Алгоритм для решения некоторых специальных задач с булевыми переменными предложен Хэли [153], однако в общем случае этот алгоритм не приводит к оптимальному решению. Хэлди и Айзексон [151] предложил реализацию «циклического» метода Гомори, дающую ускорение сходимости. Некоторое усовершенствование алгоритма Гомори (применительно к задачам по-

крытия) дано в работе Джонсона [164]. Келли [170] предложил применить метод отсекающих плоскостей для решения задач выпуклого (нецелочисленного) программирования; этот метод может быть перенесен на решение задач выпуклого целочисленного программирования, что и было сделано Куртисом [98]. Основываясь на «циклическом» алгоритме Гомори, Пихлер [197] предложил его обобщение на частично целочисленную задачу линейного программирования. Обобщение «циклического» алгоритма Гомори на частично целочисленную задачу линейного программирования дано также в работе Циммермана [220]. Ван Слайк и Уэкс [206] предложили прием «диагонализация» симплекс-таблицы, позволяющий модифицировать «циклический» и «дискретный» алгоритмы Гомори. Некоторые предложения по формированию отсекающих плоскостей в алгоритме Гомори содержатся в работе Сринивасана [208]. Витцгалл [218] перенес идею «дискретного» алгоритма Гомори на полностью целочисленную задачу с линейными и «параболическими» ограничениями:

$$a_{00} - L_0(X) - b_1(L_1(X))^2 - \dots - b_k(L_k(X))^2 \geq 0;$$

здесь $L_s(X) = a_{s1}x_1 + \dots + a_{sn}x_n$ — линейно независимые однородные линейные формы, $s=0, 1, \dots, k$. Новый подход при реализации идеи отсекающих плоскостей предложил А. А. Вотьяков [2*].

Интересный вопрос о двойственных оценках для задач целочисленного линейного программирования исследовали Гомори и Баумоль [147]; это исследование продолжили Алькали и Клеворик [71]. Впрочем, пока неясно, имеются ли на этом пути какие-либо перспективы для создания эффективных вычислительных методов.

Вторая из указанных выше групп методов отличается от первой тем, что в ней максимально используется конечность проблемы, ее комбинаторный характер. Естественно, что методы этой группы по своему характеру довольно разнородны; все они в какой-то мере используют идею перебора. Впервые метод такого рода был предложен в работе Лэнд и Дойг [180] в 1960 г. Позднее Литтл и др. авторы [185] использовали весьма близкую идею для решения задачи о бродячем торговце; их подход оказался весьма перспективным. Названный авторами «методом ветвей и границ», этот алгоритм породил ряд вариантов («метод последовательного отделения и оценивания» [90]), а главное — привлек внимание к самой идее «метода ветвей и границ» и тем самым стимулировал работы в этом направлении.

На близких к работе Литтла и др. авторов идеях основан независимо предложенный в 1963 г. Фором и Мальгранжем [116] «булев» метод для решения целочисленных задач;

изложению и уточнению этого метода посвящены также работы Ле Гарфа и Мальгранжа [121] и Фора и Мальгранжа [115].

В несколько другом направлении развиваются идеи метода «ветвей и границ» в «аддитивном» алгоритме Балаша для решения целочисленных задач с булевыми переменными [76, 73] и его модификациях [72, 74, 75]. В заметке Гловера и Зайонца [137] изложены некоторые предложения по усовершенствованию алгоритма Балаша и указано на возможность его эффективного применения к некоторым специальным задачам, в частности — к задаче раскроя. Гловер [136] предложил также усовершенствованный вариант алгоритма Балаша. Подход, близкий к подходу Балаша, развит в работах Бертье, Роя и Нгиема [90] и Бертье и Роя [89]. Модификацию метода Балаша предложил Джеффрион [126].

Хотя интенсивное развитие комбинаторных методов для решения дискретных задач началось совсем недавно, сейчас в этой области имеется уже немало работ. И. Ф. Клебанов [26] предложил модификацию «метода ветвей и границ» для решения задачи оптимального размещения однопродуктового производства. И. В. Романовский указал способ наиболее выгодной круговой расстановки станков [49]. Бил [83] дал алгоритм типа «ветвей и границ» для решения невыпуклой нелинейной задачи специального вида. Алгоритмы для решения задач теории расписаний предлагаются в работах Брукса и Уайта [94], Ломницкого [186], Брауна и Ломницкого [95] и Игнела и Шрейга [156]. Некоторое обобщение алгоритма Литтла и др. авторов дано в работе Черни [97]; в частности, им рассматривается задача коммивояжера с многократным заходом в начальный город. Развитие подхода Лэнд и Дойг дано в работах Дэкина [99] и Дрибека [111]. Задачу размещения складов исследовали Эфроимсон и Рэй [113]. Алгоритм для решения задачи расстановки оборудования дали Джевитт и Плитер [124]. Методы решения квадратичной задачи о назначениях предложили Гилмор [128] и Лаулер [182]. Для чисто комбинаторных задач методы типа «ветвей и границ» развивали Голомб и Баумерт [138]. Комбинаторный метод для синтеза логических сетей предложили Карп и др. авторы [167]. Задача о назначениях с целевой функцией, зависящей от назначения пар переменных, решалась Лэнд [179]. Литтл [184] разработал метод «ветвей и границ» применительно к задаче о синхронизации сигналов уличного движения. Томпсон [211] предложил алгоритм для решения целочисленных задач линейного программирования, близкий по идее к алгоритму Лэнд и Дойг; существенным преимуществом алгоритма Томпсона является заметное снижение требований к размеру памяти.

Задачи, близкие к задачам дискретного программирования, исследовали Ивэнеску, Розенберг и Рудеану (см. [157, 158, 160, 161, 162]). Изложение большинства результатов в этой области дано в монографии Ивэнеску «Псевдодобулево программирование и его приложения» [159].

Радо [198, 199] предложил комбинаторный алгоритм для решения задач линейного программирования с дополнительными логическими условиями.

Для решения некоторых комбинаторных задач оптимального планирования В. П. Черенин [60, 61] предложил применить разработанный им метод последовательных расчетов. Модификация метода последовательных расчетов применительно к некоторым задачам размещения производства дана в работах В. П. Черенина и В. Р. Хачатурова [62, 63, 3*]. Для решения некоторых задач дискретного программирования может быть применен метод динамического программирования. Это относится, например, к задаче о ранце, упомянутой в § 2 (подробное изложение дано у Е. Г. Гольштейна и Д. Б. Юдина [81]), а также к задаче коммивояжера (см. Беллман [85] и Хелд и Карп [154]).

К ряду задач дискретного программирования могут быть применены методы последовательной оптимизации (см. статьи В. С. Михалевича [38, 39] и В. С. Михалевича и В. В. Шкурбы [40]). Эти методы имеют в значительной мере комбинаторный характер.

К методам второй группы в некоторых отношениях близки локальные алгоритмы, исследуемые в дискретном анализе (см. работы Ю. И. Журавлева [16, 14, 15]). Идеи локального подхода были применены к задачам дискретного программирования в работах Ю. И. Журавлева и Ю. Ю. Финкельштейна [17] и Ю. Ю. Финкельштейна [57].

Промежуточное в каком-то смысле положение между первой и второй группой методов занимает предложенный Бендерсом [86] метод, использующий идею декомпозиции задачи.

Наконец, вне этих двух групп методов находится предложенный недавно Гомори [144] алгоритм для решения некоторых целочисленных задач линейного программирования. Дальнейшее развитие этих идей, а также новый подход к построению отсечений даны в [5*].

Третья группа методов — это методы случайного поиска и другие приближенные методы. Возникновение приближенных методов стимулировалось двумя причинами: 1) существующие точные методы пока что недостаточно совершенны, так что при решении больших задач возникают значительные трудности; 2) необходимо решать некоторые оперативные задачи, для которых более ценным является быстрое получение приближен-

ного решения, чем получение точного решения через значительный промежуток времени.

Решению дискретных задач с помощью случайного поиска допустимого решения с последующей «локальной» оптимизацией посвящена работа Рейтера и Шермана [200] (смысл будет разъяснен в § 6). О применении случайного поиска с локальной оптимизацией для решения задач теории расписаний сообщается в обзоре В. В. Шкурбы [65]. Одну задачу теории расписаний решали этим методом Л. И. Фейгин [53] и Ешке [163].

Недавно И. И. Пятецкий-Шапиро, В. А. Волконский, Л. В. Левина и А. Поманский [47] предложили новый алгоритм для решения задач целочисленного линейного программирования с бинарными переменными с помощью случайного поиска.

Применению методов случайного поиска для решения дискретных задач посвящена также работа Пэйджа [19].

Некоторые приближенные приемы для решения дискретной задачи размещения производства предложены в работах И. В. Гирсанова и Б. Т. Поляка [7], И. З. Кагановича [21], Н. В. Карпеевой и Т. А. Косенко [25]. Приближенным методам решения транспортной задачи с фиксированными доплатами посвящены работы А. А. Корбута и В. В. Малинникова [30, 31], Балинского [77], Куна и Баумоля [174]. Приближенный метод решения задачи линейного программирования с булевыми переменными, основанный на рассмотрении эквивалентной многоэкстремальной задачи, предложил К. А. Багриновский [1*].

Вопрос о вычислительной эффективности методов дискретного программирования будет освещен в § 7.

§ 4. МЕТОДЫ ОТСЕЧЕНИЯ

Рассмотрим целочисленную задачу линейного программирования: максимизировать

$$x_0 = \sum_{j=1}^n c_j x_j \quad (4.1)$$

при условиях

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \dots, m, \quad (4.2)$$

$$x_j \geq 0, \quad j = 1, \dots, n, \quad (4.3)$$

$$x_j — \text{целое для всех } j \in J. \quad (4.4)$$

Здесь $J \subset R = \{1, \dots, n\}$.

Мы начнем с описания первого («циклического») алгоритма Гомори [139, 140, 145, 146]. Поскольку «циклический» алгоритм Гомори предназначен для решения полностью целочисленных задач линейного программирования, мы положим $J = R$. Кроме того, будем считать, что все c_j — целые, так что целевая функция x_0 тоже должна быть целочисленной.

Решим задачу (4.1) — (4.3), т. е. исходную задачу без требования целочисленности (4.4). Если задача (4.1) — (4.3) имеет решения, то не имеет решения и исходная задача (4.1) — (4.4). Если решение задачи (4.1) — (4.3) удовлетворяет условию целочисленности (4.4), то оно является одновременно и решением исходной задачи (4.1) — (4.4).

Предположим теперь, что найденное решение задачи (4.1) — (4.3) не удовлетворяет условию целочисленности.

Пусть N — множество индексов небазисных переменных, соответствующее решению задачи (4.1) — (4.3). Выпишем развернутую симплекс-таблицу, т. е. выразим все переменные x_j , $j = 1, \dots, n$, и целевую функцию x_0 через переменные, входящие в множество N :

$$x_i = a_{i0} + \sum_{j \in N} a_{ij} (-x_j), \quad i = 0, 1, \dots, n. \quad (4.5)$$

Пусть x_{i_1} — одна из переменных, значение которой в найденном решении задачи (4.1) — (4.3) (т. е. число $a_{i_1 0}$) не является целым.

Будем обозначать через $[\alpha]$ целую часть и через $\{\alpha\}$ — дробную часть произвольного числа α . Это означает, что

$$\alpha = [\alpha] + \{\alpha\}, \quad (4.6)$$

где

$$[\alpha] — \text{целое число} \quad (4.7)$$

и

$$0 \leq \{\alpha\} < 1. \quad (4.8)$$

Пользуясь этими обозначениями, получаем из (4.5)

$$x_i + \sum_{j \in N} [a_{ij}] x_j - [a_{i0}] = \{a_{i0}\} - \sum_{j \in N} \{a_{ij}\} x_j. \quad (4.9)$$

Заметим теперь, что левая часть (4.9) должна быть целым числом и обозначим это целое число через $(-z_i)$. Тогда

$$\sum_{j \in N} \{a_{ij}\} x_j = \{a_{i0}\} + z_i$$

и при $i=i_1$ получаем

$$\sum_{j \in N} \{a_{i,j}\} x_j = \{a_{i,0}\} + z_i. \quad (4.10)$$

Докажем теперь, что целочисленная переменная z_i удовлетворяет условию $z_i \geq 0$. Допустим, что $z_i < 0$. В силу целочисленности z_i это означает, что

$$z_i \leq -1. \quad (4.11)$$

Из (4.11), (4.10) получаем

$$\sum_{j \in N} \{a_{i,j}\} x_j < 0, \quad (4.12)$$

что невозможно, в силу (4.3) и (4.8).

Итак, целочисленная переменная

$$z_i = -\{a_{i,0}\} + \sum_{j \in N} -\{a_{i,j}\} (-x_j) \quad (4.13)$$

должна удовлетворять условию неотрицательности. Однако в найденном решении задачи (4.1) — (4.3) $z_i = -\{a_{i,0}\} < 0$, т. е. условие неотрицательности z_i не удовлетворяется. Введение целочисленной неотрицательной переменной z_i по формуле (4.13) и дает нам дополнительное ограничение Гомори. Присоединяя это ограничение к оптимальной симплекс-таблице задачи (4.1) — (4.3), мы получаем новую задачу линейного программирования, к которой снова применяется тот же прием, и т. д.

Конечность «циклического» алгоритма Гомори может быть доказана при следующих условиях:

1) Множество допустимых решений нецелочисленной задачи (4.2) — (4.3) ограничено.

2) Строка i_1 , по которой строится дополнительное ограничение, выбирается по правилу:

$$i_1 = \min_{i=0, 1, \dots, n} \{i \mid \{a_{i,0}\} \neq 0\}. \quad (4.14)$$

3) Нецелочисленная задача линейного программирования (4.1) — (4.3) (и все последующие вспомогательные задачи линейного программирования) решается не по критерию максимизации x_0 , а по критерию лексикографической максимизации вектора $(x_0, x_1, \dots, x_n)^*$.

«Смешанный» алгоритм Гомори [141] является обобщением «циклического» алгоритма на частично целочисленную задачу

* Лексикографическая максимизация может быть проведена при помощи лексикографического двойственного симплекс-метода.

линейного программирования, т. е., задачу (4.1)–(4.4), в которой не все переменные должны быть целыми ($J \neq R = \{1, \dots, n\}$). Отличие «смешанного» алгоритма от «циклического» состоит только в правиле построения дополнительных ограничений. Итак, пусть $J \neq R = \{1, \dots, n\}$.

Введем обозначение

$$a'_{i,l} = \begin{cases} a_{ij}, & j \in N \setminus J, \\ a_{ij} + b_j, & j \in N \cap J. \end{cases} \quad (4.15)$$

Здесь b_j —произвольное целое число. Из (4.5), (4.15) и (4.6) получим:

$$x_i - [a_{i0}] - \sum_{j \in N \cap J} b_j x_j = \{a_{i0}\} - \sum_{j \in N} a'_{ij} x_j. \quad (4.16)$$

Обозначим левую часть (4.16) через $(-z_i)$ и заметим, что она должна быть целой, если только $i \in J$. Итак

$$\sum_{j \in N} a'_{ij} x_j = \{a_{i0}\} + z_i, \quad (4.17)$$

где z_i —целое, если $i \in J$,

Предположим теперь (как и при рассмотрении «циклического» алгоритма), что

$$i_1 \in J \quad (4.18)$$

и что $a_{i_1 0}$ —нецелое, т. е., что

$$0 < \{a_{i_1 0}\} < 1. \quad (4.19)$$

Тогда z_{i_1} в формуле (4.17) должно быть целым, причем возможны два случая

а) $z_{i_1} \geq 0$, и следовательно

$$\sum_{j \in N} a'_{i_1 j} x_j \geq \{a_{i_1 0}\}. \quad (4.20)$$

б) $z_{i_1} < 0$, в силу целочисленности z_{i_1} , имеем $z_{i_1} \leq -1$, и следовательно

$$\sum_{j \in N} a'_{i_1 j} x_j \leq \{a_{i_1 0}\} - 1. \quad (4.21)$$

Введем обозначения

$$N^+ = \{j \mid j \in N; a'_{i_1 j} \geq 0\}, \quad (4.22)$$

$$N^- = \{j \mid j \in N; a'_{i_1 j} < 0\}. \quad (4.23)$$

В случае а) получаем

$$\sum_{j \in N^+} a'_{i_1 j} x_j \geq \sum_{j \in N} a'_{i_1 j} x_j \geq \{a_{i_1 0}\}. \quad (4.24)$$

В случае б) получаем

$$\sum_{j \in N^-} a'_{i,j} x_j \leq \sum_{j \in N} a'_{i,j} x_j \leq \{a_{i,0}\} - 1, \quad (4.25)$$

т. е. (с учетом (4.19))

$$\frac{\{a_{i,0}\}}{\{a_{i,0}\} - 1} \sum_{j \in N^-} a'_{i,j} x_j \geq \{a_{i,0}\}. \quad (4.25')$$

Заметим теперь, что в обоих случаях верны следующие неравенства:

$$\sum_{j \in N^+} a'_{i,j} x_j \geq 0 \quad (4.26)$$

и

$$\frac{\{a_{i,0}\}}{\{a_{i,0}\} - 1} \sum_{j \in N^-} a'_{i,j} x_j \geq 0. \quad (4.27)$$

В случае а) из (4.24) и (4.27), а в случае б) из (4.25') и (4.26) получаем

$$\sum_{j \in N^+} a'_{i,j} x_j + \frac{\{a_{i,0}\}}{\{a_{i,0}\} - 1} \sum_{j \in N^-} a'_{i,j} x_j \geq \{a_{i,0}\}. \quad (4.28)$$

Этому неравенству удовлетворяют все допустимые решения (4.1)–(4.4), но не удовлетворяет решение задачи (4.1)–(4.3). Заметим, что правая часть (4.28) положительна, а все коэффициенты при переменных x_j в левой части — неотрицательны. Очевидно, что отсекающее неравенство (4.28) будет тем сильнее, чем меньше коэффициенты в левой части (4.28). Подбирая соответствующим образом целые числа b_j в (4.15), получаем окончательно следующее дополнительное ограничение для «смешанного» алгоритма Гомори:

$$\sum_{j \in N} a''_{i,j} x_j \geq \{a_{i,0}\}, \quad (4.29)$$

где коэффициенты $a''_{i,j}$ вычисляются следующим образом:

$$a''_{i,j} = \begin{cases} a_{i,j}, & \text{если } a_{i,j} \geq 0, \quad j \notin J, \\ \frac{a_{i,0}}{a_{i,0} - 1} a_{i,j}, & \text{если } a_{i,j} < 0, \quad j \in J, \\ \{a_{i,j}\}, & \text{если } \{a_{i,j}\} \leq \{a_{i,0}\}, \quad j \in J, \\ \frac{a_{i,0}}{a_{i,0} - 1} (\{a_{i,j}\} - 1), & \text{если } \{a_{i,j}\} > \{a_{i,0}\}, \quad j \in J \end{cases} \quad (4.30)$$

Конечность «смешанного» алгоритма Гомори может быть доказана при выполнении тех же условий, что и для «циклического» алгоритма, но с добавлением еще одного условия:

$$c_j = 0 \text{ для всех } j \in J. \quad (4.31)$$

Так как мы предполагаем, что все c_j — целые, то из (4.31) следует, что целевая функция x_0 должна быть целой. Если же условие (4.31) не выполнено (т. е. целочисленность целевой функции не гарантируется), то, как показал Уайт [217], не гарантируется не только конечность алгоритма Гомори, но даже и сходимость к решению задачи (4.1) — (4.4).

Известно, что при решении с помощью симплекс-метода задачи линейного программирования значительных размеров можно получить не только неточный, но и грубо ошибочный результат за счет накапливающихся ошибок округления. Эта опасность сильно возрастает при решении с помощью метода отсечения задачи целочисленного линейного программирования: во-первых, приходится многократно применять симплекс-метод (увеличивается продолжительность вычислений), во-вторых, возникает возможность принять целое число за нецелое и наоборот (что может привести, например, к ошибкам при вычислении коэффициентов дополнительного ограничения).

Чтобы устранить эти осложнения, связанные с ошибками округления, Гомори предложил «дискретный» алгоритм [142, 143] для решения полностью целочисленной задачи линейного программирования. Для того, чтобы «дискретный» алгоритм начал работать, надо уметь построить начальную симплекс-таблицу, полностью целочисленную и (лексикографически) двойственно допустимую (в общем случае это довольно сложно, но для некоторых классов задач начальная таблица строится относительно просто). Если начальная таблица является допустимой, то она дает решение задачи (4.1) — (4.4). Если же начальная таблица не является допустимой, т. е. среди чисел a_{i0} есть отрицательные, то строится дополнительное ограничение. Рассмотрим произвольное $\lambda > 0$. Тогда из (4.5) получаем

$$\frac{x_i}{\lambda} = \frac{a_{i0}}{\lambda} + \sum_{j \in N} \frac{a_{ij}}{\lambda} (-x_j). \quad (4.32)$$

Из (4.32) мы имеем

$$\frac{x_i}{\lambda} + \sum_{j \in N} \left\{ \frac{a_{ij}}{\lambda} \right\} x_j - \left\{ \frac{a_{i0}}{\lambda} \right\} = \left[\frac{a_{i0}}{\lambda} \right] + \sum_{j \in N} \left[\frac{a_{ij}}{\lambda} \right] (-x_j). \quad (4.33)$$

Заметим, что правая часть (4.33) должна быть целой, следовательно должна быть целой и левая часть (4.33). Допустим теперь, что

$$\frac{x_i}{\lambda} + \sum_{j \in N} \left\{ \frac{a_{ij}}{\lambda} \right\} x_j - \left\{ \frac{a_{i0}}{\lambda} \right\} < 0. \quad (4.34)$$

Выпишем очевидные неравенства

$$\begin{aligned} \frac{x_i}{\lambda} + \sum_{j \in N} \left\{ \frac{a_{ij}}{\lambda} \right\} x_j &\geq 0, \\ -1 &< -\left\{ \frac{a_{i0}}{\lambda} \right\} \leq 0 \end{aligned} \quad (4.35)$$

и получим

$$-1 < \frac{x_i}{\lambda} + \sum_{j \in N} \left\{ \frac{a_{ij}}{\lambda} \right\} x_j - \left\{ \frac{a_{i0}}{\lambda} \right\} < 0. \quad (4.36)$$

Но (4.36) не может иметь места, так как левая часть (4.33) должна быть целой. Значит, неравенство (4.34) не может иметь места, т. е. левая (а значит и правая) часть (4.33) должна быть неотрицательной:

$$z_i = \left[\frac{a_{i0}}{\lambda} \right] + \sum_{j \in N} \left[\frac{a_{ij}}{\lambda} \right] (-x_j) \geq 0. \quad (4.37)$$

Допустим теперь, что $i = i_1$ и $a_{i_1 0} < 0$. Тогда неравенство (4.37) является отсекающим, т. е. ему удовлетворяют все допустимые решения задачи (4.1) — (4.4), но не удовлетворяет псевдоплан, соответствующий начальной таблице. Выбирая строку, соответствующую z_{i_1} , в качестве ведущей, подбираем λ так, чтобы ведущий элемент (выбранный по правилам лексикографического двойственного симплекс-метода) был равен -1 , и проделываем одну итерацию лексикографического двойственного симплекс-метода. Снова получаем полностью целочисленную и лексикографически двойственно допустимую таблицу. Если она не является допустимой, то применяем к ней тот же прием и т. д.

Для того чтобы «дискретный» алгоритм Гомори был конечным, достаточно, чтобы

1) множество допустимых решений нецелочисленной задачи (4.2) — (4.3) было ограниченным,

2) строка i_1 , по которой строится дополнительное ограничение, выбиралась бы из условия

$$i_1 = \min_{i=0,1,\dots,n} \{i | a_{i0} < 0\}. \quad (4.38)$$

Мартин [188] предложил модификацию «циклического» алгоритма Гомори, дающую существенное ускорение сходимости. В исходной задаче числа c_j , a_{ij} , b_i предполагаются целыми. Элементы последовательно преобразуемых симплекс-таблиц записываются без округления (иначе говоря, поскольку эти элементы являются рациональными числами, отдельно записываются числитель и знаменатель).

В начале решается нецелочисленная задача (4.1) — (4.3). Если полученное решение не удовлетворяет условию целочисленности (4.4), то выбирается строка i_1 , в которой $a_{i_1,0}$ — нецелое и записывается условие целочисленности переменной x_{i_1} (если считать переменные x_j целыми ($j \in N$))

$$z_{i_1} = -\{a_{i_1,0}\} + \sum_{j \in N} -\{a_{i_1,j}\}(-x_j) \text{ целое } \geq 0. \quad (4.39)$$

Это — дополнительное ограничение «циклического» алгоритма Гомори. В качестве ведущей строки условно выбирается строка, соответствующая z_{i_1} и по правилам лексикографического двойственного симплекс-метода выбирается ведущий столбец k . Затем ограничение (4.39) преобразуется в ограничение

$$z'_{i_1} = -\{ta_{i_1,0}\} + \sum_{j \in N} -\{ta_{i_1,j}\}(-x_j) \text{ целое } \geq 0. \quad (4.39')$$

Здесь число t ищется с помощью алгоритма Евклида из условия

$$\{ta_{i_1,k}\} = \min_{\substack{z=1,2,\dots \\ \{za_{i_1,k}\} > 0}} \{za_{i_1,k}\}. \quad (4.40)$$

В качестве ведущей строки выбирается строка, соответствующая z_{i_1} , а в качестве ведущего столбца — ранее выбранный столбец k , и прodelывается одна итерация двойственного симплекс-метода. Вновь полученная симплекс-таблица проверяется на целочисленность соответствующего ей (псевдо) плана и если условие целочисленности не выполняется, то к ней применяется тот же прием и т. д. — до тех пор, пока (псевдо) план, соответствующий очередной симплекс-таблице, не окажется целочисленным. Если эта симплекс-таблица окажется допустимой и двойственно-допустимой, то получено решение задачи (4.1) — (4.4). Если же это не так, то полученная задача линейного программирования решается и получается допустимая и лексикографически двойственно-допустимая симплекс-таблица.

К этой таблице снова применяется та же процедура, что и к оптимальной симплекс-таблице задачи (4.1) — (4.3) и т. д.

Куртийо [98] распространил «дискретный» алгоритм Гомори на задачи выпуклого целочисленного программирования.

Выше мы уже отмечали, что его подход сочетает идеи метода отсекающих плоскостей Келли [170] для задач нелинейного программирования и метода Гомори. В случае линейности целевой функции и ограничений метод Куртийо переходит в метод Гомори.

Ю. Ю. Финкельштейном [54, 55] была предложена новая реализация метода отсечения (названная им *B*-алгоритмом), специально приспособленная для задач целочисленного линейного программирования с булевыми переменными, т. е. для задач следующего вида:
максимизировать

$$\sum_{j=1}^N c_j x_j \quad (4.41)$$

при условиях

$$\sum_{j=1}^N a_{ij} x_j = b_i, \quad i = 1, \dots, m, \quad (4.42)$$

$$x_j \geq 0, \quad j = 1, \dots, N, \quad (4.43)$$

$$x_j \leq 1, \quad j = 1, \dots, n, \quad (4.44)$$

$$x_j - \text{целые}, \quad j = 1, \dots, n. \quad (4.45)$$

Здесь $n \leq N$.

Построение дополнительных ограничений в *B*-алгоритме основано на следующем факте. Пусть (x'_1, \dots, x'_n) — лексикографический максимум вектора (x_1, \dots, x_n) при условиях (4.42)–(4.44), причем x'_r — первая по номеру нецелочисленная компонента. Тогда ограничению

$$\sum_{j=1}^{r-1} [x_j(1 - x'_j) + (1 - x_j)x'_j] + (1 - x_r) \geq 1 \quad (4.46)$$

удовлетворяют все планы (x_1, \dots, x_N) , удовлетворяющие условиям (4.42)–(4.45), но не удовлетворяет ни один из планов, доставляющих лексикографический максимум вектору (x_1, \dots, x_n) .

§ 5. КОМБИНАТОРНЫЕ МЕТОДЫ

Методы этой группы в той или иной мере используют конечность числа допустимых решений дискретной задачи. Разумеется, полный перебор всех возможных решений является, вообще говоря, нереализуемым, и сущность всех описываемых методов состоит в целенаправленном сокращении этого перебора.

Важная и наиболее известная серия методов этой группы может быть объединена под названием «методов ветвей и гра-

ниц». Последнее название было дано в работе Литтла и др. авторов [185] для метода решения задачи о бродячем торговце. Впервые же метод подобного рода был предложен в работе Лэнд и Доиг [180] в 1960 г. Опишем коротко его идею.

Рассматривается целочисленная (или частично целочисленная) задача вида (4.1)—(4.4). Как и в методах отсечения, процесс начинается с решения соответствующей ей линейной задачи (4.1)—(4.3) (без учета требований целочисленности). Если полученное оптимальное решение $X^0 = (x_0^0, x_1^0, \dots, x_n^0)$ этим требованиям не удовлетворяет, то значение целевой функции $x_0^0 = \Delta^0$ дает верхнюю границу для искомого целочисленного оптимума, т. е. $\max x_0 \leq \Delta^0$. Пусть некоторая переменная x_{i_0} , где $i_0 \in J$, не получила в решении X^0 целочисленного значения. В оптимальном целочисленном решении значение x_{i_0} должно быть либо уменьшено по крайней мере до $\lfloor x_{i_0}^0 \rfloor$, либо увеличено по крайней мере до $\lceil x_{i_0}^0 \rceil + 1$. Проще всего найти верхнюю x_{i_0}'' и нижнюю x_{i_0}' (целочисленные) границы допустимого значения для x_{i_0} можно, решив две задачи линейного программирования, заключающиеся в максимизации и минимизации x_{i_0} при условиях (4.2)—(4.3). Теперь для каждого фиксированного целочисленного значения $x_{i_0} = k$ в найденном отрезке $[x_{i_0}', x_{i_0}'']$ можно искать $\max x_0$ путем решения задачи линейного программирования с ограничениями (4.2)—(4.3) и дополнительным ограничением $x_{i_0} = k$. Все эти возможности можно суммировать в виде некоторого дерева задач, в котором исходная вершина V^0 соответствует решению X^0 задачи (4.1)—(4.3), а каждая из соединенных с ней ветвью вершин $V \binom{i_0}{k}$ соответствует решению задачи линейного программирования $L \binom{i_0}{k}$: максимизировать x_0 при ограничениях (4.2)—(4.3) и дополнительном ограничении $x_{i_0} = k$. Каждой из вершин $V \binom{i_0}{k}$ приписывается граница $\Delta \binom{i_0}{k}$, равная максимальному значению x_0 , найденному при решении задачи $L \binom{i_0}{k}$. Очевидно, что $\Delta \binom{i_0}{k} \leq \Delta^0$ для всех значений k .

Если решение $X \binom{i_0}{k}$ задачи $L \binom{i_0}{k}$ удовлетворяет условию целочисленности (4.4) для $k = k'$, и $x_0 \binom{i_0}{k'} = \max_k x_0 \binom{i_0}{k}$, то

$X \left(\begin{smallmatrix} i_0 \\ k' \end{smallmatrix} \right)$ является решением исходной задачи (4.1) — (4.4). Если же это не так, то может возникнуть необходимость в дальнейшем «разветвлении» дерева из некоторых вершин $V \left(\begin{smallmatrix} i_0 \\ k \end{smallmatrix} \right)$, не удовлетворяющих условию целочисленности (4.4). Пусть, например, в вершине $V \left(\begin{smallmatrix} i_0 \\ k' \end{smallmatrix} \right)$ компонента $x_{i_1} \left(\begin{smallmatrix} i_0 \\ k' \end{smallmatrix} \right)$ не удовлетворяет условию целочисленности (4.4) ($i_1 \in J$).

Тогда, аналогично тому, как это делалось раньше, мы максимизируем (и минимизируем) x_{i_1} при ограничениях (4.2) — (4.3) и дополнительном ограничении $x_{i_0} = k'$ и находим верхнюю x'_{i_1} и нижнюю x''_{i_1} (целочисленные) границы допустимого изменения x_{i_1} .

Совершенно аналогично определяются и вершины $V \left(\begin{smallmatrix} i_0 & i_1 \\ k' & k \end{smallmatrix} \right)$, соединяемые ветвями с $V \left(\begin{smallmatrix} i_0 \\ k' \end{smallmatrix} \right)$.

В дальнейшем может возникнуть необходимость в новых разветвлениях и рассмотрении вершины вида $V \left(\begin{smallmatrix} i_0 & i_1 & \dots & i_p \\ k_0 & k_1 & \dots & k_p \end{smallmatrix} \right)$, соответствующего ей решения $X \left(\begin{smallmatrix} i_0 & i_1 & \dots & i_p \\ k_0 & k_1 & \dots & k_p \end{smallmatrix} \right)$ и границы $\Delta \left(\begin{smallmatrix} i_0 & i_1 & \dots & i_p \\ k_0 & k_1 & \dots & k_p \end{smallmatrix} \right)$. Здесь $i_r \in J$, $r=0, 1, \dots, p$, k_r — целое ($r=0, 1, \dots, p$), $X \left(\begin{smallmatrix} i_0 & i_1 & \dots & i_p \\ k_0 & k_1 & \dots & k_p \end{smallmatrix} \right)$ — решение задачи (4.1) — (4.3) при дополнительных ограничениях

$$x_{i_2} = k_r, \quad r=0, 1, \dots, p \quad \text{и} \quad \Delta \left(\begin{smallmatrix} i_0 & i_1 & \dots & i_p \\ k_0 & k_1 & \dots & k_p \end{smallmatrix} \right) = x_0 \left(\begin{smallmatrix} i_0 & i_1 & \dots & i_p \\ k_0 & k_1 & \dots & k_p \end{smallmatrix} \right).$$

Возможность уменьшения объема вычислений (по сравнению с полным перебором) основывается на следующих соображениях:

1) Невозрастание границ:

$$\Delta^0 \geq \Delta \left(\begin{smallmatrix} i_0 \\ k_0 \end{smallmatrix} \right) \geq \Delta \left(\begin{smallmatrix} i_0 & i_1 \\ k_0 & k_1 \end{smallmatrix} \right) \geq \dots \quad (5.1)$$

Это дает возможность разумным образом выбирать вершину, в которой будет произведено очередное «разветвление». Естественно в качестве такой вершины выбрать вершину с наибольшей границей (из числа вершин, из которых еще не производилось разветвление).

2) Возможность отбрасывания «бесперспективных» вершин и проверка оптимальности решения. Это производится следующим образом.

Рассмотрим две вершины

$$V \begin{pmatrix} i_0 i'_1 \dots i'_p \\ k'_0 k'_1 \dots k'_p \end{pmatrix}, V \begin{pmatrix} i_0 i''_1 \dots i''_q \\ k''_0 k''_1 \dots k''_q \end{pmatrix}.$$

Допустим, что: а) $X \begin{pmatrix} i_0 i'_1 \dots i'_p \\ k'_0 k'_1 \dots k'_p \end{pmatrix}$ удовлетворяет условию целочисленности (4.4), б) $X \begin{pmatrix} i_0 i''_1 \dots i''_q \\ k''_0 k''_1 \dots k''_q \end{pmatrix}$ не удовлетворяет условию целочисленности (4.4), в) $\Delta \begin{pmatrix} i_0 i'_1 \dots i'_p \\ k'_0 k'_1 \dots k'_p \end{pmatrix} > \Delta \begin{pmatrix} i_0 i''_1 \dots i''_q \\ k''_0 k''_1 \dots k''_q \end{pmatrix}$.

Тогда любое допустимое решение задачи (4.1)—(4.4), удовлетворяющее условию: $x_{i_0} = k_0$; $x_{i'_r} = k'_r$, $r = 1, \dots, p$, не может быть оптимальным решением задачи (4.1)—(4.4).

Мы не будем более детально описывать метод Лэнд и Дойг. Ясно, что в реализации этого метода для полностью и частично целочисленных (или дискретных) задач линейного программирования никакой разницы нет. Кроме того, нет надобности в специальном доказательстве его конечности и достижения именно оптимального решения. С другой стороны, представляется, что этот метод целесообразно применять для задач с небольшим количеством целочисленных (или дискретных) переменных, поскольку объем вычислений и требуемые размеры памяти весьма быстро возрастают с увеличением размера задачи. Следует, впрочем, указать, что в упомянутой выше работе [21] Томпсон дал алгоритм, основанный на только что изложенных соображениях и существенно более экономный в отношении требований к памяти.

По-видимому, наиболее перспективное направление для развития идей метода Лэнд и Дойг было указано в упомянутой выше работе Литтла и др. авторов [185] — это максимальное использование специфической структуры задачи для построения специализированных правил «разветвления» и вычисления границ.

В алгоритме Лэнд и Дойг используются не только комбинаторные соображения, но и аппарат линейного программирования (следует напомнить, что этот аппарат является основным рабочим инструментом методов отсечения). В алгоритме Литтла и др. авторов комбинаторные соображения сочетаются в сущности с рассмотрением специальной задачи линейного программирования — задачи о назначениях, соответствующей в некотором смысле рассматриваемой ими задаче о бродячем торговце. Таким образом, метод Лэнд и Дойг и ме-

тод Литтла и др. авторов занимают как бы промежуточное положение между методами отсечения и «чисто комбинаторными» методами, среди которых следует в первую очередь отметить метод Фора и Мальгранжа и метод Балаша. Как метод Фора и Мальгранжа, так и метод Балаша были вначале предложены для решения задач целочисленного линейного программирования с бинарными (булевыми) переменными, принимающими значения 0 или 1.

Первым из них был «булев» метод Фора и Мальгранжа, изложенный в работе [116]. Рассматривается целочисленная задача линейного программирования с булевыми переменными:

максимизировать

$$x_0 = \sum_{j=1}^n c_j x_j \quad (5.2)$$

при условиях

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i=1, \dots, m, \quad (5.3)$$

$$x_j = 0 \text{ или } 1, \quad j=1, \dots, n. \quad (5.4)$$

Здесь c_j — целые числа, $j=1, \dots, n$. Не нарушая общности, можно считать, что все $c_j \geq 0$ (в противном случае для всех j таких, что $c_j < 0$, можно было бы сделать замену $x_j = 1 - x_j$; мы считаем, что такая замена уже сделана).

Одновременно с каждой из переменных x_j рассматривается переменная $x'_j = 1 - x_j$. С помощью вновь введенных переменных каждое из ограничений (5.3) переписывается в следующем виде

$$\sum_{j \in N_i^+} a_{ij} x_j + \sum_{j \in N_i^-} a_{ij} (1 - x'_j) \leq b_i,$$

где

$$N_i^+ = \{j \mid j=1, 2, \dots, n; a_{ij} \geq 0\}, \quad N_i^- = \{j \mid j=1, 2, \dots, n; a_{ij} < 0\},$$

или

$$\sum_{j \in N_i^+} a_{ij} x_j + \sum_{j \in N_i^-} a'_{ij} x'_j \leq b'_i, \quad i=1, \dots, m. \quad (5.3')$$

Здесь

$$a'_{ij} = -a_{ij} > 0, \quad b'_i = b_i - \sum_{j \in N_i^-} a_{ij}.$$

Запись (5.3) в виде (5.3') удобна тем, что все коэффициенты a_{ij} и a'_{ij} , входящие в (5.3'), неотрицательны.

Очевидно, что если $b'_i < 0$ для некоторого $i=i_1$, то задача

(5.2)—(5.4) не имеет ни одного допустимого решения. Поэтому можно считать, что все $b'_i > 0$.*)

Сначала некоторым способом ищется допустимое решение $X^0 = (x_0^0, x_1^0, \dots, x_n^0)$ задачи (5.2), (5.3'), (5.4). Затем к условиям (5.3') добавляется условие $x_0 \geq x_0^0 + 1$ и ищется допустимое решение X^1 новой задачи. Затем к условиям (5.3') добавляется условие $x_0 \geq x_0^1 + 1$ и снова ищется допустимое решение X^2 вновь полученной задачи и т. д.

Поскольку условие $x_0 \geq t$ может быть переписано в виде

$$\sum_{j=1}^n c_j x_j \leq \sum_{j=1}^n c_j - t, \quad (5.3'')$$

то все сводится к нахождению допустимого решения (удовлетворяющего условиям вида (5.3') и (5.4)).

Пусть $(\bar{x}_1, \dots, \bar{x}_k)$ — некоторый набор значений переменных x_1, \dots, x_k ($1 \leq k \leq n$), удовлетворяющий условию (5.4). Введем обозначения:

$$N_i^+(\bar{x}_1, \dots, \bar{x}_k) = \{j \mid j \in N_i^+ \cap \{1, \dots, k\}; \bar{x}_j = 1\} \quad (5.5)$$

и

$$N_i^-(\bar{x}_1, \dots, \bar{x}_k) = \{j \mid j \in N_i^- \cap \{1, \dots, k\}; \bar{x}_j = 0\}. \quad (5.6)$$

Назовем набор $(\bar{x}_1, \dots, \bar{x}_k)$ условно допустимым, если выполняются неравенства

$$\sum_{j \in N_i^+(\bar{x}_1, \dots, \bar{x}_k)} a_{ij} + \sum_{j \in N_i^-(\bar{x}_1, \dots, \bar{x}_k)} a'_{ij} \leq b'_i, \quad i = 1, \dots, m. \quad (5.7)$$

Назовем набор $(\bar{x}_1, \dots, \bar{x}_k)$ недопустимым, если $(\bar{x}_1, \dots, \bar{x}_k, x_{k+1}, \dots, x_n)$ не является допустимым решением (т. е. не удовлетворяет условиям (5.3')—(5.4)) ни при каких значениях (x_{k+1}, \dots, x_n) .

Ясно, что непосредственная проверка набора $(\bar{x}_1, \dots, \bar{x}_k)$ на недопустимость — дело сложное, в то время как проверка на условную допустимость очень проста. Основная идея алгоритма Фора и Мальгранжа заключается в сокращении направленного перебора (при поиске допустимого решения) за счет выявления недопустимости некоторых наборов $(\bar{x}_1, \dots, \bar{x}_k)$ (желательно — при возможно меньшем k) с помощью проверки

*Если $b'_{i_1} = 0$ для некоторого $i = i_1$, то можно считать, что $x_j = 0$, если $a_{i_1 j} > 0$ и что $x'_j = 0$ для всех $j \in N_{i_1}^-$. Ясно, что этот случай тоже можно не рассматривать.

некоторых других наборов на условную допустимость. При этом используются следующие утверждения:

1) Для того чтобы набор (x_1, \dots, x_k) был недопустимым, достаточно, чтобы он не был условно допустимым.

2) Для того чтобы набор (x_1, \dots, x_k) был недопустимым, необходимо и достаточно, чтобы были недопустимыми два набора $(x_1, \dots, x_k, 0)$ и $(x_1, \dots, x_k, 1)$.

Мы не будем давать детального описания «булевого» алгоритма Фора и Мальгранжа. Обобщение этого алгоритма на общую целочисленную задачу линейного программирования дано в работе Ле Гарфа и Мальгранжа [121]. Работа Фора и Мальгранжа [115] не содержит существенно новых идей, однако в ней четко формализовано содержание работ [116] и [121]; изложение ясное и лаконичное.

Другой метод для задач с булевыми переменными (так называемый «аддитивный алгоритм») был развит Балашем (см. заметку [76] и подробное изложение в [73]). Балаш рассматривает следующую задачу:

минимизировать

$$x_0 = \sum_{j=1}^n c_j x_j \quad (5.8)$$

при условиях

$$\sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = 1, \dots, m. \quad (5.9)$$

$$x_j = 0 \text{ или } 1, \quad j = 1, \dots, n. \quad (5.10)$$

Как и при рассмотрении алгоритма Фора и Мальгранжа, мы, не ограничивая общности, считаем, что $c_j \geq 0, j = 1, \dots, n$.

В начале алгоритма Балаша все переменные полагают равными нулю. Если $(x_1, \dots, x_n) = (0, \dots, 0)$ является допустимым решением (т. е. удовлетворяет условиям (5.9)), то очевидно, что $(0, \dots, 0)$ является искомым оптимальным решением задачи (5.8)–(5.10). Если же $(0, \dots, 0)$ не является допустимым решением, то проводится систематический процесс последовательного придания некоторым переменным значения 1.

Для этого проводится анализ ряда вершин возникающего дерева вариантов. Произвольной его вершине $V(j_1, \dots, j_q)$ соответствует набор (x_1, \dots, x_n) , для которого

$$\bar{x}_j = \begin{cases} 1, & \text{если } j = j_s, \quad 1 \leq s \leq q, \\ 0 & \text{в противном случае,} \end{cases} \quad (5.11)$$

Вершина называется допустимой, если соответствующий ей набор удовлетворяет условиям (5.9). В противном случае она называется недопустимой. При разветвлении вершины $V(j_1, \dots, j_q)$ возникают вершины вида $V(j_1, \dots, j_q, j_{q+1})$.

Уменьшение объема вычислений (по сравнению с полным перебором) достигается следующим образом:

1) Сформулированы правила, по которым в ряде случаев можно определить, что недопустимая вершина $V(j_1, \dots, j_q)$ является «бесперспективной», т. е. что все вершины вида $V(j_1, \dots, j_q, j_{q+1}, \dots, j_p)$ при любых p ($q+1 \leq p \leq n$) также являются недопустимыми. Эти правила близки к аналогичным соображениям Фора и Мальгранжа.

2) Если $V(j_1, \dots, j_q)$ — допустимая вершина, то можно не анализировать все вершины, для которых значение функционала

$$\sum_{j=1}^n c_j x_j \geq \sum_{r=1}^q c_{j_r}$$

3) Если $V(j_1, \dots, j_q)$ — некоторая вершина, то значение функционала в вершине $V(j_1, \dots, j_q, \dots, j_p)$ ($q+1 \leq p \leq n$) может только возрасти:

$$\sum_{r=1}^p c_{j_r} \geq \sum_{r=1}^q c_{j_r} \text{ при } p > q. \quad (5.11)$$

Как и в методе Лэнд и Дойг, каждой вершине, в которой еще возможно разветвление, приписывается некоторым образом оценка (граница).

Если $V(j_1, \dots, j_q)$ — допустимая вершина, то оценка равна значению функционала $\sum_{r=1}^q c_{j_r}$. Если же $V(j_1, \dots, j_q)$ — недопустимая вершина, то оценка векторная и имеет вид (N, C) . Здесь C — оценка снизу для значения функционала на вершинах вида $V(j_1, \dots, j_p)$, $p > q$, а N — оценка «недопустимости» вершины $V(j_1, \dots, j_q)$ (например, максимальная невязка в (5.9); если же уже показано, что вершина $V(j_1, \dots, j_q)$ — «бесперспективная», то $N = +\infty$).

Если в некоторый момент минимум оценки функционала на (уже вычисленных) допустимых вершинах достигается на вершине $V(j_1, \dots, j_q)$ и не превосходит минимума оценки функционала на (уже вычисленных) неразветвленных недопустимых вершинах, то вершина $V(j_1, \dots, j_q)$ дает искомое решение задачи (5.8) — (5.10). Если же это не так, то производится разветвление наиболее «перспективной» из недопустимых вершин («перспективность» определяется сравнением оценок (N, C)).

Мы не будем входить в дальнейшие подробности. Балаш опубликовал ряд работ, посвященных дальнейшему развитию «аддитивного» алгоритма: в [74] дано обобщение на общую целочисленную задачу линейного программирования и на целочисленную задачу нелинейного программирования, в [75] дан улучшенный вариант «аддитивного» алгоритма для бинарных задач, в [76] результаты работы [75] переносятся на частично целочисленную задачу линейного программирования.

Комбинаторный подход (так называемый «локальный» алгоритм), позволяющий снизить перебор при решении задач целочисленного линейного программирования, был предложен в работе Ю. И. Журавлева и Ю. Ю. Финкельштейна [17].

Эти идеи были развиты Ю. Ю. Финкельштейном [57] применительно к квазиблочным задачам дискретного программирования с сепарабельной целевой функцией. Рассматривалась следующая задача:

максимизировать

$$\sum_{j=1}^n f_j(x_j) \quad (5.12)$$

при условиях

$$\varphi_i(x_{j_{i1}}, \dots, x_{j_{ip_i}}) = 0, \quad i = 1, \dots, m, \quad (5.13)$$

$$x_j \in B_j = \{b_{j1}, \dots, b_{jq_j}\}, \quad j = 1, \dots, n. \quad (5.14)$$

Здесь $\{j_{i1}, \dots, j_{ip_i}\} \subset N = \{1, \dots, n\}$. На функции f_j и φ_i не накладывается никаких требований. Предполагается лишь, что вычисление каждой из этих функций при заданном значении аргументов производится достаточно быстро.

Рассматривается матрица $\|h_{ij}\|$.

$$h_{ij} = \begin{cases} 1, & \text{если } i \in \{j_{i1}, \dots, j_{ip_i}\}, \\ 0 & \text{в противном случае.} \end{cases} \quad (5.15)$$

Предполагается, что матрица $\|h_{ij}\|$ имеет квазиблочную структуру. Это означает, что множество индексов $M = \{1, \dots, m\}$ можно разбить на непустые непересекающиеся множества U_1, \dots, U_k таким образом, что только множества U_l и U_{l+1} , $l = 1, \dots, k-1$, могут зацепляться по $\|h_{ij}\|$. Два множества U_r и U_s не зацепляются по $\|h_{ij}\|$, если

$$\left(\sum_{i \in U_r} h_{ij} \right) \left(\sum_{i \in U_s} h_{ij} \right) = 0, \quad j = 1, \dots, n. \quad (5.16)$$

Обозначим $N_l = \left\{ \sum_{i \in U_l} h_{ij} \geq 1 \right\}$. Предложен алгоритм, позволяю-

ший найти решение задачи (5.12) — (5.14) не более чем за $\sum_{i=1}^k \left(\prod_{j \in N_i} q_j \right)$ элементарных операций, в то время как объем полного перебора равен $\prod_{j=1}^n q_j$. Разумеется, при этом возрастают требования к объему памяти.

Трудности, встречающиеся при решении дискретных задач, делают особенно актуальными поиски методов решения специальных задач. Одним из интересных шагов в этом направлении представляется работа В. П. Черенина [60, 61], в которых развивается «метод последовательных расчетов». Изложим его основную идею.

В методе последовательных расчетов рассматривается задача минимизации функции $p(\omega)$, заданной на семействе всех подмножеств некоторого конечного множества I . Предполагается, что p удовлетворяет следующему условию (своеобразный аналог условия выпуклости):

$$p(\delta) + p(\gamma) \leq p(\delta \cup \gamma) + p(\delta \cap \gamma) \quad (5.17)$$

для всех $\delta, \gamma \subset I$. Множество α называется точкой глобального минимума $p(\omega)$ на множестве I , если $p(\omega) \geq p(\alpha)$ для любого $\omega \subset I$. Метод основан на следующей теореме: функция $p(\omega)$, удовлетворяющая условию (5.17), монотонно убывает вплоть до α и монотонно возрастает после α на любой последовательности подмножеств множества I , обладающих тем свойством, что каждое последующее множество содержит предыдущее, отличаясь от него ровно на один элемент.

Из этой теоремы следует, что если для каких-либо $\omega_1 \subset \omega_2 \subset I$ вычислены значения $p(\omega_1)$ и $p(\omega_2)$, то в случае $p(\omega_1) < p(\omega_2)$ можно отбросить все $2^{m-|\omega_2|}$ подмножеств—вариантов $\omega \supset \omega_2$, в случае $p(\omega_1) > p(\omega_2)$ можно отбросить все $2^{|\omega_1|}$ вариантов $\omega \subset \omega_1$ (здесь через $|\omega|$ обозначено число элементов множества ω). Указанные условия отбраковки позволяют последовательно исключать из рассмотрения большие группы вариантов, заведомо не дающих минимума $p(\omega)$.

§ 6. МЕТОДЫ СЛУЧАЙНОГО ПОИСКА И ДРУГИЕ ПРИБЛИЖЕННЫЕ МЕТОДЫ

Наиболее распространенной является следующая методика применения случайного поиска: 1) Производится случайный выбор допустимого решения. 2) Полученное допустимое реше-

ние улучшается до получения «локального» оптимума (см. ниже). Эта процедура повторяется многократно и из всех полученных «локальных» оптимумов выбирается наилучший по значению функционала.

Первый этап (случайный выбор допустимого решения) может быть организован достаточно эффективно лишь в том случае, когда множество допустимых решений легко обозримо. Например, для задачи о бродячем торговце множество допустимых решений — это множество перестановок чисел $(1, 2, \dots, n)$. В общем же случае поиск допустимого решения — задача весьма нелегкая.

Переходим ко второму этапу. Введем понятие окрестности. Окрестность G_X определяется для каждого допустимого решения X и сама тоже состоит из допустимых решений. Единственное (практическое) требование к понятию окрестности состоит в том, что окрестность не должна содержать слишком много точек, так что полный перебор всех точек окрестности может быть (практически) осуществлен. Например, в задаче о коммивояжере окрестностью G_X перестановки $X = (i_1, \dots, i_n)$ можно назвать все перестановки, которые получаются из (i_1, \dots, i_n) транспозицией двух индексов.

Допустимое решение X назовем «локальным» максимумом, если значение функционала $f(X)$ удовлетворяет условию

$$f(X) \geq \max_{Y \in G_X} f(Y). \quad (6.1)$$

Второй этап решения протекает следующим образом: 1-ый шаг. Производится полный перебор всех $X \in G_{X^0}$ (X^0 — допустимое решение, полученное на первом этапе). Если X^0 является «локальным» максимумом, то процесс окончен. Если же

$$f(X^1) = \max_{X \in G_{X^0}} f(X) > f(X^0),$$

то переходим ко второму шагу.

k -ый шаг ($k \geq 2$). Производится полный перебор всех $X \in G_{X^{k-1}}$. Если X^{k-1} является «локальным» максимумом, то процесс окончен. Если же

$$f(X^k) = \max_{X \in G_{X^{k-1}}} f(X) > f(X^{k-1}),$$

то переходим к $(k + 1)$ -му шагу.

Выше мы уже упоминали некоторые работы, в которых применялся случайный поиск с локальной оптимизацией. Наиболее полное описание метода дано у Рейтера и Шермана [200]; там же дана реализация метода применительно к задаче коммивояжера.

Другой метод случайного поиска приближенного решения задачи целочисленного линейного программирования с булевыми переменными дан в работе И. И. Пятецкого — Шапиро, В. А. Волконского, Л. В. Левиной и А. Поманского [47].

Рассматривается следующая задача: максимизировать

$$\sum_{j=1}^n c_j x_j \quad (6.2)$$

при условиях

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 1, \dots, m, \quad (6.3)$$

$$x_j = 0, \text{ или } 1, \quad j = 1, \dots, n. \quad (6.4)$$

Все коэффициенты a_{ij} , c_j , b_i неотрицательны.

Решение ищется следующим образом:

а) Фиксируется b_0 и ищется вектор (x_1, \dots, x_n) , удовлетворяющий условиям (6.3), (6.4) и дополнительному условию

$$\sum_{j=1}^n c_j x_j \geq b_0. \quad (6.5)$$

Поиск вектора (x_1, \dots, x_n) производится описываемым ниже итеративным методом.

б) Увеличиваем b_0 и повторяем ту же процедуру. Процесс заканчивается, когда система неравенств (6.3) — (6.5) перестает решаться за фиксированное число итераций.

Центр тяжести данной работы лежит в методе итераций, который применяется следующим образом. Начальный набор $\{x_j^0\}$ выбирается произвольно. Пусть на k -ом шаге получен набор $\{x_j^{(k)}\}$. Вычисляем

$$\Delta_0 = \max \left[\left(b_0 - \sum_{j=1}^n c_j x_j^{(k)} \right) / b_0, 0 \right] \quad (6.6)$$

и

$$\Delta_i = \max \left[\left(\sum_{j=1}^n a_{ij} x_j^{(k)} - b_i \right) / b_i, 0 \right]. \quad (6.7)$$

Затем с использованием случайного выбора независимо друг от друга меняются компоненты вектора $\{x_j^{(k)}\}$ с одной и той же вероятностью p , равной

$$p = \min(c, \max_{0 \leq i \leq m} \Delta_i). \quad (6.8)$$

(В проводившихся численных экспериментах принималось $c =$

$= 1/2$). При этом получается новый набор $\{x_j^{(k+1)}\}$. Когда все Δ_i обращаются в нуль—итеративный процесс окончен.

Приближенные методы, не использующие идею случайного поиска, строились также для некоторых частных задач. Назовем здесь лишь неоднородную транспортную задачу, уже упоминавшуюся выше. Она заключается в минимизации

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij}(x_{ij}) \quad (6.9)$$

при обычных условиях транспортного типа

$$x_{ij} \geq 0, \quad \sum_{j=1}^n x_{ij} \leq a_i, \quad \sum_{i=1}^m x_{ij} = b_j, \quad (6.10)$$

где

$$c_{ij}(x_{ij}) = \begin{cases} 0, & x_{ij} = 0, \\ c_{ij}x_{ij} + d_{ij}, & x_{ij} > 0. \end{cases} \quad (6.11)$$

Балинский [77] (см. также обзор А. А. Корбуа [28]) указал для этой задачи эквивалентную ей целочисленную задачу, заключающуюся в минимизации

$$\sum_{i=1}^m \sum_{j=1}^n (c_{ij}x_{ij} + d_{ij}y_{ij}) \quad (6.12)$$

при условиях (6.10) и дополнительных условиях

$$y_{ij} = 0 \text{ или } 1, \quad x_{ij} \leq M_{ij}y_{ij}, \quad (6.13)$$

где

$$M_{ij} = \min \{a_i, b_j\}. \quad (6.14)$$

Исходя из некоторых легко проверяемых свойств решения задачи (6.12), (6.13), (6.10), Балинский предлагает аппроксимировать исходную задачу обычной транспортной задачей с целевой функцией

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij}^1 x_{ij}, \quad (6.15)$$

где

$$c_{ij}^1 = c_{ij} + \frac{d_{ij}}{M_{ij}}. \quad (6.16)$$

Этот прием легко пояснить и геометрически.

В работах А. А. Корбуа и В. В. Малинникова [30, 31] предложен прием, позволяющий улучшить получаемое методом Балинского приближенное решение. Его идея заключается в следующем. В любом базисном плане $\|x_{ij}^1\|$ аппроксими-

рующей транспортной задачи, как известно, всегда существует по крайней мере два таких x_{ij}^1 , что $x_{ij}^1 = M_{ij}$. Для соответствующих пар i, j затраты в исходной и аппроксимирующей задачах совпадают и равны $c_{ij}x_{ij}^1 + d_{ij}$. Для остальных пар затраты в аппроксимирующей задаче оказываются заниженными. Поэтому принимается, что искать улучшения плана следует путем перераспределений только на тех парах i, j , для которых $x_{ij}^1 < M_{ij}$, причем значения затрат в соответствующих клетках следует пересчитать.

Для этого в качестве первого приближения к решению задачи (6.9) — (6.11) составляется и решается транспортная задача с коэффициентами целевой функции (6.16) (т. е. осуществляется метод Балинского). В полученном решении отмечаются те x_{ij}^1 , для которых $x_{ij}^1 = M_{ij}$, т. е. $x_{ij}^1 = a_i$ или $x_{ij}^1 = b_j$ (соответствующие строки i и столбцы j назовем висячими рядами). Затем висячие ряды вычеркиваются и производится пересчет чисел a_i и b_j : если $x_{ij}^1 = a_i$, вычеркивается строка i и полагается $b_j^1 = b_j - x_{ij}^1$; если $x_{ij}^1 = b_j$, вычеркивается столбец j и полагается $a_i^1 = a_i - x_{ij}^1$. Для остальных рядов принимается $a_i^1 = a_i$, $b_j^1 = b_j$. Находятся величины

$$M_{ij}^1 = \min \{a_i^1, b_j^1\}, \quad (6.14')$$

вычисляются

$$c_{ij}^2 = c_{ij} + \frac{d_{ij}}{M_{ij}^1} \quad (6.16')$$

и снова решается транспортная задача (уже меньшего размера) с исходными данными a_i^1, b_j^1, c_{ij}^2 . Для оптимального решения этой задачи снова повторяется вычеркивание висячих рядов, пересчет данных и т. д. На каждом цикле этого (разумеется, конечного) процесса в окончательный план заносятся те x_{ij} , которые оказались в висячих рядах.

Деталей проведения этого процесса мы касаться не будем. Основная тонкость состоит в том, что он может оказаться не монотонным; в связи с этим в алгоритме могут появиться некоторые несложные ветвления.

§ 7. ВЫЧИСЛИТЕЛЬНЫЙ ОПЫТ И РЕАЛИЗАЦИЯ АЛГОРИФМОВ НА ЭВМ

Важнейшей в дискретном программировании является проблема численной реализации методов. В настоящее время эта область остается почти целиком экспериментальной —

многие наблюдавшиеся при машинных экспериментах явления до сих пор не получили теоретического объяснения.

Мы начнем с методов отсечения как исторически первых и наиболее освоенных в вычислительном отношении. Речь будет идти прежде всего о методе Гомори. Сразу же отметим, что несмотря на гарантированную его сходимость за конечное число шагов, поведение машинных реализаций отличается большой «непредсказуемостью»: для значительной части решавшихся задач оптимальное решение не удавалось получить даже после многих тысяч итераций (см., например, Женюи [125]), в то время как другие задачи решались за несколько десятков итераций. По-видимому, прямой связи с размерами задачи здесь нет: неудачи были зафиксированы даже для небольших задач (до десятка переменных), а успехи — для сравнительно крупных (несколько десятков переменных). Причины этого не выяснены; подмечено лишь, что необходимое число итераций имеет тенденцию к возрастанию с увеличением числа переменных, ростом порядка коэффициентов задачи и увеличением заполненности ее матрицы. Весьма любопытно при этом то, что иногда удавалось существенно сократить число необходимых итераций путем простой перестановки уравнений и переменных. Далее, отмечены случаи, когда решение небольшой задачи теории расписаний (3 машины, 6 деталей) в форме целочисленной задачи линейного программирования потребовало свыше 3000 итераций, что более чем вчетверо превышает объем работы по прямому перебору всех $6!$ вариантов.

Первые машинные эксперименты по методу Гомори относятся к 1959 г.; Гомори [146] сообщает об успешном решении нескольких задач с числом переменных до 15. В статье Гасса [123] приводятся сведения о трех программах для машины IBM-704, законченных к лету 1960 г. В них использовался второй алгоритм метода Гомори; допустимые размеры задач: $n \leq 100$, $m + n < 200$. Мы не будем характеризовать программы, созданные за несколько последующих лет, так как существенный прогресс был достигнут лишь после осуществления предложенной Мартином [188] модификации метода Гомори («ускоренный евклидов алгоритм»). На ее основе была написана программа, позволяющая решать задачи с $m \leq 1600$, $n \leq 2200$. Например, задача с $m = 80$, $n = 2400$ была решена в начале 1965 г. на машине IBM-7094 за 24 мин. Максимальный размер решенных с ее помощью задач был $m = 215$, $n = 2600$. Правда, последние данные относятся к довольно специфическим задачам типа задач о покрытии (заполненность матрицы 15%; большинство ненулевых элементов равно единице).

Эти задачи потребовали добавления очень малого количества отсекающих ограничений (менее 10).

Наиболее подробно накопленный по методу Гомори вычислительный опыт отражен в обзорной статье Балинского [78], а также в работах Сринивасана [208], Хэлди и Айзексона [151]. Некоторые сведения по этому вопросу имеются также в работах Балинского и Куондта [79], Била [82], Дзя [108], д'Эзопо и Лефковица [114], Миллера, Таккера и Землина [191], Стори и Вагнера [209].

В работе Бауэра [80] приведена программа «дискретного» алгорифма Гомори на языке АЛГОЛ; однако в этой программе имеются ошибки. Усовершенствованный и исправленный вариант этой программы дан Лангмааком [181].

Для методов, отнесенных нами ко второй группе, вычислительный опыт является неизмеримо более бедным. Во многих случаях отсутствуют даже сведения об их программировании для ЭВМ. Известно, например, что метод Лэнд и Дойг подвергся машинному опробованию только в 1964 г.; данными о результатах этого эксперимента мы пока не располагаем. Метод «ветвей и границ» Литтла и др. [185] для решения задачи коммивояжера был опробован на машине IBM-7090; среднее по 100 задачам время решения для 30 пунктов составило около 1 мин; блок-схема программы приведена в [185]. Имеются также сведения о достаточной эффективности машинной реализации метода Фора и Мальгранжа [116]; блок-схема соответствующей программы приведена в работе Ле Гарфа и Мальгранжа [121].

Некоторые результаты ручного экспериментального счета по методу Балаша охарактеризованы в статьях [73, 4*]. По результатам эксперимента алгоритм оказался эффективным для задач с количеством переменных $n \leq 30$. При дальнейшем увеличении n его эффективность быстро уменьшается.

В статьях В. П. Черенина [60] и В. Р. Хачатурова [3*] имеются сведения о результатах машинного решения некоторых задач средних размеров при помощи разработанного им «метода последовательных расчетов».

Результаты машинного эксперимента по решению методом случайного поиска с локальной оптимизацией задачи коммивояжера (с количеством городов до 57) приведены в работе Рейтера и Шермана [200].

В статье И. И. Пятецкого-Шапиро, В. А. Волконского, Л. В. Левиной и А. Поманского [47] описаны первые результаты машинного эксперимента по разработанному ими методу случайного поиска для целочисленных задач линейного про-

граммирования. Так, в задаче размером 3×17 при истинном значении максимума 14,91 приближенное его значение 14,76 было получено за 166 итераций.

§ 8. ОБЗОРНЫЕ СТАТЬИ И МОНОГРАФИИ

Впервые в монографической литературе дискретному программированию было уделено место в книге Симоннара [205]; посвященные этому вопросу главы имеются в монографиях Данцига [10], Баумоля [3], Е. Г. Гольштейна и Д. Б. Юдина [8], С. И. Зуховицкого и Л. И. Авдеевой [20], Йокша [165], Вайда [212].

Первая обзорная статья, посвященная целочисленному линейному программированию, была написана Данцигом [105]. Хотя эта статья опубликована в 1960 г., она и сейчас не потеряла своего значения (ее содержание в значительной степени повторено в монографии Данцига [10]). Отметим также обзорный доклад Форте [118]. Обзорная статья Ламбера [178] к настоящему времени уже несколько устарела, а обзор Панаже [196] полностью перекрывается другими обзорными статьями.

Достаточно подробным является обзор А. А. Корбута [28] (библ. 84 назв.) и особенно обзор Балинского [78] (библ. 104 назв.). Сжатостью и высоким уровнем изложения отличается обзор Биля [82] (библ. 27 назв.). В работе Ю. И. Волкова и В. И. Хохлюка [5] дано подробное изложение алгоритма Гомори, метода Лэнд и Дойг, а также некоторых вопросов, до сих пор вообще не освещавшихся в литературе на русском языке (модификация Мартина алгоритма Гомори и «прямой» алгоритм Бен-Израэля и Чарнса).

Обзор Лаулера и Вуда [183] посвящен специально методам «ветвей и границ» (библ. 49 назв.)

БИБЛИОГРАФИЯ

1. Атлас Б. А., Гдалевич С. С., Выбор оптимального комплекса технических средств доставки грузов. Экон. и матем. методы, 1966, 2, № 3, 394—405 (РЖМат, 1966, 11В297)
2. Барткус А. Т., Приближенное решение некоторых комбинаторных задач линейного программирования методом дихотомии. Ж. вычисл. матем. и матем. физики, 1964, 4, № 4, 776—780 (РЖМат, 1964, 12В291)
3. Баумоль У., Экономическая теория и исследование операций. Перев. с англ., М., «Прогресс», 1965 (РЖМат, 1966, 9В173К)

4. **Борисова Т. Н.**, Влашек З., Карманов В. Г., Поляк Б. Т., Некоторые методы решения задач о размещении. В сб. Вычислительные методы и программирование, т. 3. М., Изд. МГУ, 1965, 441—451 (РЖМат, 1965, 1В255)
5. **Волков Ю. И.**, Хохлаук В. И., Методы решения целочисленных задач линейного программирования. В сб. Матем. модели и методы оптимальн. планир. Новосибирск, Изд. «Наука», 1966, 5—35 (РЖМат, 1967, 1В310)
6. **Гейл Д.**, Теория линейных экономических моделей. Перев. с англ., М., ИЛ, 1963 (РЖМат, 1964, 1В402К)
7. **Гирсанов И. В.**, Поляк Б. Т., Математические методы решения задачи о размещении. В сб. Пробл. оптимальн. планир., проектир. и упр. произ-вом М., Изд. МГУ, 1963, 288—300 (РЖМат, 1964, 1В387)
8. **Гольштейн Е. Г.**, Юдин Д. Б., Новые направления в линейном программировании. М., Изд. «Сов. Радио», 1966 (РЖМат, 1966, 9В184К)
9. **Гофман А. Дж.**, Краскал Дж. Б., Целочисленные граничные точки выпуклых многогранников. В сб. Линейные неравенства и смежные вопросы. М., Изд. ин. лит., 1959, 325—347
10. **Данциг Дж.**, Линейное программирование, его применения и обобщения. Перев. с англ., М., «Прогресс», 1966 (РЖМат, 1966, 1В245К)
11. —, Орден А., Вольф Ф., Обобщенный симплексный метод минимизации линейной формы при ограничениях в виде линейных неравенств. В сб. Методы решения общ. задачи линейн. программир. Вып. 2. М., Госстатиздат, 1963, 35—46 (РЖМат, 1964, 3В335)
12. **Демьянов В. Ф.**, Худяков Л. Ю., Одна задача целочисленного выпуклого программирования. Изв. АН СССР. Техн. кибернетика 1965, № 6, 3—12
13. —, —, Решение одной целочисленной задачи квадратичного программирования. Прикл. матем. и механ., 1965, 29, № 1, 158—161 (РЖМат, 1965, 8В185)
14. **Журавлев Ю. И.**, Локальные алгоритмы вычисления информации. I. Кибернетика, 1965, № 1, 12—19 (РЖМат, 1967, 1В174)
15. —, Локальные алгоритмы вычисления информации. II. Кибернетика, 1966, № 2, 1—11 (РЖМат, 1967, 3В233)
16. —, Об одном классе алгоритмов над конечными множествами. Докл. АН СССР, 1963, 151, № 5, 1025—1028 (РЖМат, 1964, 4В237)
17. —, Финкельштейн Ю. Ю., Локальные алгоритмы для задач линейного целочисленного программирования. В сб. Пробл. кибернетики. Вып. 14. М., «Наука», 1965, 289—295 (РЖМат, 1966, 4В144)
18. **Зайчик М. Л.**, Метод решения одного класса целочисленных распределительных задач. Экон. и матем. методы, 1966, 2, № 2, 225—234 (РЖМат, 1966, 9В195)
19. **Залесский А. Е.**, Сведение некоторых комбинаторных задач к целочисленному линейному программированию. Вестн. АН БССР. Сер. физ.-матем. н., Изв. АН БССР. Сер. физ.-матем. н., 1965, № 3, 24—28 (РЖМат, 1966, 3В271)
20. **Зуховицкий С. И.**, Авдеева Л. И., Линейное и выпуклое программирование. Справочное руководство. М., «Наука», 1964 (РЖМат, 1965, 4В204К)
21. **Каганович И. З.**, Оптимальная мощность и размещение предприятий молочной промышленности Эстонии. В сб. Матем. методы и пробл. размещения произ-ва, М., Экономиздат, 1963, 24—46 (РЖМат, 1964, 8В410)
22. —, Рейснер М., Об одной задаче целочисленного программирования. Изв. АН ЭстССР. Серия обществ. н., 1965, 14, № 2, 223—231 (РЖМат, 1966, 3В270)
23. **Каменецкий Л. Е.**, Корбут А. А., Применение метода линейного

- программирования для решения задачи развития Канско-Ачинского бассейна. Уголь, 1965, № 2, 29—31
24. Каплан Г. А., Выбор структуры системы телемеханики для комплексов с сосредоточенными объектами. Экон. и матем. методы, 1966, 2, № 3, 439—445 (РЖМат, 1966, 12В299)
 25. Карпеева И. В., Косенко Т. А., Задача размещения мясоперерабатывающих предприятий в Горно-Алтайской автономной области на 1970 г. Экон. и матем. методы, 1966, 2, № 3, 454—459
 26. Клебанов И. Ф., Комбинаторный метод решения одного класса задач размещения предприятий. Гос. план. комис. БССР. Вычисл. центр. Минск, 1965 (РЖМат, 1966, 7В268 К)
 27. Корбут А. А., Неоднородные задачи размещения (типа транспортной). В сб. Применение матем. при размещ. производит. сил, М., «Наука», 1964, 57—61 (РЖМат, 1965, 3В276)
 28. —, Целочисленные задачи линейного программирования. В сб. Экон.-матем. методы. Вып. 2. М., «Наука», 1965, 141—186 (РЖМат, 1965, 12В260)
 29. —, Каменецкий Л. Е., Вайсбурд Б. М., Применение методов линейного программирования при проектировании Канско-Ачинского угольного бассейна. Изв. ВУЗ'ов, Горн. журнал, 1964, № 6, 48—51
 30. —, Малинников В. В., Неоднородные задачи о размещении типа транспортной. Тр. Ленингр. инж.-экон. ин-та, 1966, вып. 58, 194—198 (РЖМат, 1967, 1В338)
 31. —, —, Приближенное решение некоторых неоднородных моделей размещения. Экон. и матем. методы, 1965, 1, № 3, 425—441 (РЖМат, 1965, 12В262)
 32. Коробков В. К., О некоторых целочисленных задачах линейного программирования. В сб. Пробл. кибернетики, Вып. 14. М., «Наука», 1965, 297—299 (РЖМат, 1965, 11В205)
 33. Линейные неравенства и смежные вопросы. Сб. перев. с англ., М., Изд. ин. лит., 1959
 34. Лурье А. Л., О некоторых задачах календарного планирования. В сб. Пробл. кибернетики, вып. 7. М., Физматгиз, 1962, 201—208 (РЖМат, 1964, 2В255)
 35. Майстрова Т. Л., О применении линейного программирования для минимизации нормальных форм булевых функций. «Тр. Междунар. симп. по теории релейн. устройств и конечн. автоматов (ИФАК)», 1962. Синтез релейных структур», М., 1965, 245—260 (РЖМат, 1966, 6В242)
 36. Медницкий В. Г., Оптимальный раскрой полосы. Экон. и матем. методы, 1965, 1, № 2, 282—288 (РЖМат, 1965, 11В204)
 37. Минц Л. Е., Финкельштейн Ю. Ю. Применение математических методов и электронно-вычислительных машин для решения задач размещения промышленности. В сб. Планир. и экон.-матем. методы, М., «Наука», 1964, 256—277 (РЖМат, 1964, 7В473)
 38. Михалевич В. С., Последовательные алгоритмы оптимизации и их применение. I. Кибернетика, 1965, № 1, 45—56 (РЖМат, 1966, 1В96)
 39. —, Последовательные алгоритмы оптимизации и их применение. II. Последовательные правила для опытов с детерминированными исходами. Кибернетика, 1965, № 2, 85—88 (РЖМат, 1966, 5В68).
 40. —, Шкурба В. В., Последовательные схемы оптимизации в задачах упорядочения выполнения работ. Кибернетика, 1966, № 2, 34—40 (РЖМат, 1966, 11В279)
 41. Москаленко М. П., Кошечева Н. С., Определение оптимального варианта расстановки рыболовецких судов по районам промысла в течение года. Научн. тр. НГУ. Сер. экон., 1965, вып. 5, 33—39 (РЖМат, 1966, 1В274)

42. Моцкус И. Б., Методы целочисленнолинейного программирования некоторых нелинейных задач оптимального проектирования. Изв. АН СССР. Энерг. и трансп., 1963, № 4, 410—416 (РЖМат, 1964, 5В362)
43. —, О применении метода Монте—Карло для решения многоэкстремальных и комбинаторных задач. В сб. «Общ. вопр. примен. вероятност. и статист. методов». Вып. 4. Киев, Гостехиздат УССР, 1963, 30—41 (РЖМат, 1965, 3В198)
44. Мудров В. И., Один способ решения задачи коммивояжера с помощью целочисленного линейного программирования (задача нахождения гамильтоновых путей в полном графе). Ж. вычисл. матем. и матем. физ., 1963, 3, № 6, 1137—1139 (РЖМат, 1965, 8В323)
45. —, Определение гамильтоновых путей кратчайшей длины в полном графе методами целочисленного программирования. Изв. АН СССР, Техн. киберн., 1965, № 2, 3—8 (РЖМат, 1965, 10В227)
46. Поляк Р. А., Об одной неоднородной транспортной задаче. В сб. Матем. модели и методы оптимальн. планир., Новосибирск, «Наука», 1966, 109—115 (РЖМат, 1967, 1В313)
47. Пятецкий—Шапиро И. И., Волконский В. А., Левина Л. В., Ломанский А., Об одном итеративном методе решения задач целочисленного программирования. Докл. АН СССР, 1966, 169, № 6, 1289—1292 (РЖМат, 1967, 3В325)
48. Райкин А. Л., Оптимизация избыточности при наличии ограничений. Автоматика и телемеханика, 1965, 26, № 2, 388—398
49. Романовский И. В., Задача о наивыгоднейшей круговой расстановке станков. Экон. и матем. методы, 1966, 2, № 4, 578—581 (РЖМат, 1967, 2В301)
50. Рыбашов М. В., Дудников Е. Е., Сведение задачи о нахождении допустимого целочисленного (булева) решения к задаче вогнутого программирования. Изв. АН СССР. Техн. кибернетика, 1966, № 2, 39 (РЖМат, 1966, 9В191)
51. Сарчимелия Р. А., Алгоритм целочисленного решения систем линейных алгебраических уравнений и его применение в задачах целочисленного линейного программирования. Экон. и матем. методы, 1965, 1, № 5, 760—771 (РЖМат, 1966, 2В263)
52. Толчан А. Я., Об одном способе оптимизации структуры сети связи. В сб. Пробл. передачи информ. Вып. 11, М., АН СССР, 1963, 133—146 (РЖМат, 1964, 4В345)
53. Фейгин Л. И., Об одной задаче теории расписаний. Труды ЦНИИКА, вып. 8, М., 1964
54. Финкельштейн Ю. Ю., Алгоритм для решения задач целочисленного линейного программирования с булевыми переменными. Экон. и матем. методы, 1965, 1, № 5, 746—759 (РЖМат, 1966, 2В265)
55. —, Об одном алгоритме для решения задач целочисленного линейного программирования с булевыми переменными. Докл. АН СССР, 1965, 164, № 5, 993—996 (РЖМат, 1966, 5В236)
56. —, О построении дополнительных ограничений при решении задач целочисленного линейного программирования. Изв. АН СССР. Техн. кибернетика, 1965, № 3, 43—44 (РЖМат, 1966, 3В270)
57. —, О решении задач дискретного программирования специального вида. Экон. и матем. методы, 1965, 1, № 2, 262—270 (РЖМат, 1965, 11В206)
58. Форд Л. Р., Фалкерсон Д. Р., Потoki в сетях. Перев. с англ., М., «Мир», 1966 (РЖМат, 1966, 11В249К)
59. Хеллер И., Томпкинс Ч. Б., Обобщение одной теоремы Данцига. В сб. Линейные неравенства и смежные вопросы, М., Изд. ин. лит., 1959, 348—354

60. Черенин В. П., Решение некоторых комбинаторных задач оптимального планирования методом последовательных расчетов. Материалы к конф. по опыту и перспективам примен. матем. методов и ЭВМ в планировании. Новосибирск, 1962 (РЖМат, 1965, 2В410К)
61. —, Решение некоторых комбинаторных задач оптимального планирования методом последовательных расчетов. (Научно-методические материалы Экон.-матем. семинара. Лабор. экон.-матем. методов АН СССР, Вычисл. центр АН СССР, вып. 2). М., 1962 (РЖМат, 1964, 11В261)
62. —, Хачатуров В. Р., Решение методом последовательных расчетов одного класса задач о размещении производства. В сб. Экон.-матем. методы. Вып. 2. М., «Наука», 1965 (РЖМат, 1966, 1В272)
63. —, —, Решение методом последовательных расчетов одного класса задач о размещении производства. В сб. Примен. матем. методов и ЭВМ в экон. исслед. Ташкент, «Наука», 1965, 112—114 (РЖМат, 1966, 1В271)
64. Шембель В. И., Задача распределения сверхурочных часов при проектировании сезонных предприятий. Экон. и матем. методы, 1965, 1, № 4, 576—587 (РЖМат, 1966, 4В147)
65. Шкурба В. В., Вычислительные схемы решения задач теории расписаний. Кибернетика, 1965, № 3, 72—76 (РЖМат, 1966, 2В303)
66. —, Некоторые математические задачи производственного планирования. Научн. тр. Моск. инж.-экон. ин-та, 1964, вып. 21, 146—155 (РЖМат, 1965, 12В263)
67. —, Теория расписаний. I. Комбинаторные задачи. Семинар «Экон. киберн. и исслед. опер.», Киев; 1964
68. —, Теория расписаний. II. Общие подходы и методы моделирования. Семинар «Экон. киберн. и исслед. опер.», Киев, 1964
69. Юдин Д. Б., Методы количественного анализа сложных систем. I. Изв. АН СССР. Техн. кибернетика, 1965, № 1, 3—13 (РЖМат, 1967, 1В415)
70. —, Методы количественного анализа сложных систем. II. Изв. АН СССР. Техн. кибернетика, 1966, № 1 (РЖМат, 1967, 3В148)
71. Alcaly R. E., Klevorick A. K., A note on the dual prices of integer programs. *Econometrica*, 1966, 34, № 1, 206—214 (РЖМат, 1966, 8В165)
72. Balas E., Adaptation de l'algorithme additif accéléré à la programmation linéaire mixte. C. r. Acad. sci., 1966, АВ262, № 14, А831—А834 (РЖМат, 1966, 11В251)
73. —, An additive algorithm for solving linear programs with zero-one variables. *Operat. Res.*, 1965, 13, № 4, 517—546 (РЖМат, 1966, 4В140)
74. —, Extension de l'algorithme additif à la programmation en nombres entiers et à la programmation non linéaire. C. r. Acad. sci., 1964, 258, № 21, 5136—5139 (РЖМат, 1965, 6В174)
75. —, La méthode du filtre (l'algorithme additif accéléré) pour les programmes linéaires en variables bivalentes. C. r. Acad. sci., 1966, АВ262, № 13, А766—А769 (РЖМат, 1966, 11В250)
76. —, Un algorithme additif pour la résolution des programmes linéaires en variables bivalentes. C. r. Acad. sci., 1964, 258, № 15, 3817—3820 (РЖМат, 1965, 6В173)
77. Balinski M. L., Fixed-cost transportation problems. *Naval Res. Log. Quart.*, 1961, 8, № 1, 41—54 (РЖМат, 1962, 4В304)
78. —, Integer programming: methods, uses, computation. *Manag. Sci.* 1965, 12, № 3, 253—313 (РЖМат, 1966, 8В162)
79. —, Quandt R. E., On an integer program for a delivery problem. *Operat. Res.*, 1964, 12, № 2, 300—304 (РЖМат, 1965, 1В258)

80. **Bauer F. L.**, Algorithm 153 Gomory. Communications of the ACM, 1963, 6, № 2, 68
81. **Beale E. M. L.**, A method of solving linear programming problems when some but not all of the variables must take integral values. Statistical Techniques Research Group, Princeton, N. J., 1958
82. —, Survey of integer programming. Operat. Res. Quart., 1965, 16, № 2, 219—228 (PJKMar, 1966, 3B266)
83. —, Two transportation problems. Actes 3-e Confinternat. rech. opérationnelle. Oslo. 1963, Paris—London, 1964, 780—787 (PJKMar, 1965, 7B162)
84. **Bellman R.**, Comment on Dantzig's paper on discrete variable extremum problems. Operat. Res., 1957, 5, № 5, 723—724 (PJKMar, 1958, № 7, 6019)
85. —, Dynamics programming treatment of the traveling salesman problem. J. Assoc. Comput. Machinery, 1962, 9, № 1, 61—63 (PJKMar, 1962, 11B367)
86. **Benders J. F.**, Partitions procedures for solving mixed-variables programming problems. Numer. Math., 1962, 4, № 3, 238—252 (PJKMar, 1964, 1B389)
87. **Ben-Israel A.**, **Charnes A.**, On some problems of diophantine programming. Cahiers Centre d'Études de Rech. Opérat., 1962, 4, № 4, 215—288
88. **Berge C.**, Quelques réflexions sur les programmes linéaires en nombres entiers. Conf. 6 Congr. Internat. Inst. Manag. Sci., Paris, 1959. London—Paris—New York—Los Angeles, 1—4 (PJKMar, 1962, 3B367)
89. **Bertier P.**, **Roy B.**, Procédure de résolution pour une classe de problèmes pouvant avoir un caractère combinatoire. Cahiers Centre d'Études de Rech. Opérat., 1964, 6, 202—208
90. —, —, **Ng Hiem Ph. T.**, Programmes linéaires en nombres entiers et procédure S. E. P. Metra, 1965, 4, № 3
91. **Bessière F.**, Sur la recherche du nombre chromatique d'un graphe par un programme linéaire en nombres entiers. Rev. franç. rech. opérat., 1965, 9, № 35, 143—148 (PJKMar, 1966, 4B143)
92. **Boot J. G. G.**, **Theil H.**, A procedure for integer maximization of a definite quadratic function. Actes 3-e Conf. internat. rech. opérationnelle. Oslo 1963. Paris—London, 1964, 667—682 (PJKMar, 1965, 3B211)
93. **Breuer M. A.**, Implementation of threshold nets by integer linear programming. IEEE Trans. Electron. Comput., 1965, 14, № 6, 950—952 (PJKMar, 1966, 9B194)
94. **Brooks G. H.**, **White Ch. R.**, An algorithm for finding optimal or near optimal solutions to the production scheduling problem. Industr. Engineering, 1965, 16, № 1, 34—40
95. **Brown A. P. G.**, **Lomnicki Z. A.**, Some applications of the «branch—and—bound» algorithm to the machine scheduling problem. Operat. Res. Quart., 1966, 17, № 2, 173—186 (PJKMar, 1967, 1B312)
96. **Camion P.**, Une méthode de résolution par l'algebre de Boole des problèmes combinatoires ou interviennent des entiers. Cahiers Centre d'Études de Rech. Opérat. (Bruxelles), 1960, 2, 234—289
97. **Černý M.**, Algorithmus řešení okružních dopravních úloh. Ekon.-mat. obzor, 1965, № 4, 391—405 (PJKMar, 1966, 7B267)
98. **Courtillot M.**, Sur la résolution des programmes a solutions entières. Rev. franç. traitement inform. chiffres, 1965 8, № 2, 81—94 (PJKMar, 1966, 3B269)
99. **Dakin R. J.**, A tree search algorithm for mixed integer programming problems. Comput. J., 1965, 8, 250—255
100. **Dalton R. E.**, Discrete programming with Gomory cuts. Doct. diss.

- Raleigh, N. C. State Univ., 1964, Dissert. Abstr., 1965, 26, № 2, 1060 (PЖMar, 1966, 8B163D)
101. —, Llewellyn R. W., An extension of the Gomory mixed-integer algorithm to mixed-discrete variables. *Manag. Sci.*, 1966, 12, № 7, 562—575
 102. Dantzig G. B., A machine-job scheduling model. *Manag. Sci.*, 1960, 6, № 2, 191—196 (PЖMar, 1961, 6B100)
 103. —, Discrete variable extremum problems. *Operat. Res.*, 1957, 5, № 2, 266—277 (PЖMar, 1958, 6018)
 104. —, Note on solving linear programs in integers. *Naval Res. Log. Quart.*, 1959, 6, № 1, 75—76 (PЖMar, 1961, 4B97)
 105. —, On the significance of solving linear programming problems with some integer variables. *Econometrica*, 1960, 28, № 1, 30—44 (PЖMar, 1961, 4B96)
 106. —, Fulkerson D. R., Johnson S. M. On a linear programming combinatorial approach to the traveling-salesman problem. *Operat. Res.*, 1959, 7, № 1, 58—66 (PЖMar, 1961, 1B125)
 107. —, —, —, Solution of a large scale traveling salesman problem. *Operat. Res.*, 1954, 2, № 3, 393—410
 108. Day R. D., On optimal extracting from a multiple file data storage system: an application of integer programming. *Operat. Res.*, 1965, 13, № 3, 472—494 (PЖMar, 1966, 5B240)
 109. Dinkelbach W., On convex integer programming. *Colloq. on applic. mathem. to econ.*, Budapest, 1963, Hung. Acad. Sci., 1965, 75—78 (PЖMar, 1966, 4B158).
 110. —, Steffens F., Gemischt ganzzahlige lineare Programme zur Lösung gewisser Entscheidungsprobleme. *Unternehmensforschung*, 1961, 5, № 1, 3—14 (PЖMar, 1962, 3B368)
 111. Driebeck N. J., An algorithm for the solution of mixed integer programming problems. *Manag. Sci.*, 1966, 12, 576—587 (PЖMar, 1967, 2B305)
 112. Edwards C., Using discrete programming. *Agric. Econ. Res.*, 1963, 15, № 2, 49—60 (PЖMar, 1964, 5B451)
 113. Efromson M. A., Ray T. L., A branch—bound algorithm for plant location. *Operat. Res.*, 1966, 14, № 3, 361—368 (PЖMar, 1967, 1B311)
 114. D'Esopo D. A., Leikowitz B. Note on an integer linear programming model for determining a minimum embarkation fleet. *Naval Res. Log. Quart.*, 1964, 11, № 1, 79—82
 115. Faure R., Malgrange Y., Nová metoda celočíselného lineárního programování. *Econ.-mat. obzor*, 1966, 2, № 1, 11—20 (PЖMar, 1966, 4B193)
 116. —, —, Une méthode booléenne pour la résolution des programmes linéaires en nombres entiers. *Gestion*, 1963, 6, 250—260 (PЖMar, 1964, 4B347)
 117. Fortet R., Applications de l'algèbre de Boole en recherche opérationnelle. *Rev. franç. rech. operat.* (Bruxelles), 1960, 2, 47—126
 118. —, Les programmes linéaires en nombres entiers. *Math. Progr. écon.*, Paris, 1964, 71—91 (PЖMar, 1964, 1B256)
 119. Freeman R. J., Computational experience with the Balas Integer Programming Algorithm. *Rand Corp.*, P—3241, October 1965
 120. Fulkerson D. R., Flow networks and combinatorial operations research. *Amer. Math. Monthly*, 1966, 73, № 2, 115—138 (PЖMar, 1966, 9B188)
 121. Le Garff A., Malgrange Y., Résolution des programmes linéaires à valeurs entières par une méthode booléenne «compacte», *Actes 3-e conf. internat. rech. opérationnelle*. Oslo. 1963, Paris—London, 1964, 695—701 (PЖMar, 1965, 3B195)
 122. Garver L. L., Power generation scheduling by integer programming—development of theory. *Power Apparatus and Systems*, 1963, № 64

123. **Gass S. I.**, Recent developments in linear programming. *Advances in Computers*, vol. 2, New York—London, Acad. Press, 1961, 295—377 (PЖMar, 1962, 12B377)
124. **Gavett J. W.**, **Plyter N. W.** The optimal assignment of facilities to locations by branch and bound. *Operat. Res.*, 1966, 14, № 2, 210—232 (PЖMar, 1967, 1B308)
125. **Genuys F.**, Application de la programmation linéaire en nombres entiers à un problème de découpe. *Inform. Process.*, 1962, Amsterdam, N. Holland Publ. Co., 1963, 195—197 (PЖMar, 1964, 3B333)
126. **Geoffrion A.**, Integer programming by implicit enumeration and Balas method. *Rand Corp.*, RM—4783—PR, February, 1966
127. **Giglio R. J.**, **Wagner H. W.**, Approximative solutions to the three machine scheduling problem. *Operat. Res.*, 1964, 12, № 2, 305—324 (PЖMar, 1965, 1B311)
128. **Gilmore P. C.**, Optimal and suboptimal algorithms for the quadratic assignment problem. *J. Soc. Industr and Appl. Math.*, 1962, 10, № 2, 305—313 (PЖMar, 1964, 5B363)
129. —, **Gomory R. E.**, A linear programming approach to the cutting stock problem. *Operat. Res.*, 1961, 9, № 6, 849—859 (PЖMar, 1963, 12B556)
130. —, —, A linear programming approach to the cutting stock problem. Part II. *Operat. Res.*, 1963, 11, № 6, 863—888 (PЖMar, 1965, 4B214)
131. —, —, A solvable case of the traveling salesman problem. *Proc. Nat. Acad. Sci. U. S. A.*, 1964, 51, № 2, 178—181 (PЖMar, 1964, 11B262)
132. —, —, Multistage cutting stock problems of two and more dimensions. *Operat. Res.*, 1965, 13, № 1, 94—120 (PЖMar, 1966, 4B139)
133. —, —, Sequencing a one-state variable machine: a solvable case of the traveling salesman problem. *Operat. Res.*, 1964, 12, № 5, 655—679 (PЖMar, 1965, 4B216)
134. **Glover F.**, A bound escalation method for the solution of integer programs. *Cahiers du centre d'Etudes de Recherche Opérat.* (Bruxelles), 1964, 6, 131—168
135. —, A hybrid-dual integer programming algorithm. *Cahiers Centre Etudes Rech. Opérat.*, 1965, 7, 5—23
136. —, A multiphase-dual algorithm for the zero-one integer programming problem. *Operat. Res.*, 1965, 13, № 6, 879—919 (PЖMar, 1966, 7B265)
137. —, **Zionts S.**, A note on the additive algorithm of Balas. *Operat. Res.* 1965, 13, № 4, 546—549 (PЖMar, 1966, 4B142)
138. **Golomb S. W.**, **Baumert L. D.**, Backtrack programming. *J. Assoc. Comput. Machinery*, 1965, 12, 516—524 (PЖMar, 1966, 8B286)
139. **Gomory R. E.**, An algorithm for integer solutions to linear programs. Princeton—IBM Mathematics Research Project, Technical Report № 1, November 17, 1958
140. —, An algorithm for integer solutions to linear programs. *Recent Advances Math. Program.* New York—San Francisco—Toronto—London, McGraw—Hill Book. Co., Inc., 1963, 269—302 (PЖMar, 1965, 3B194)
141. —, An algorithm for the mixed integer problem. *Rand Corp.*, P—1885, Santa Monica, California, February 22, 1960
142. —, An all-integer programming algorithm IBM Research Center, 1960, January, Research Report RC—189
143. —, An all-integer integer programming algorithm. *Industrial Scheduling*, Prentice Hall, Englewood Cliffs, New Jersey, 1963, 193—206
144. —, On the relation between integer and noninteger solutions to linear programs. *Proc. Nat. Acad. Sci. U. S. A.*, 1965, 53, № 2, 260—265 (PЖMar, 1965, 10B224)
145. —, Outline of an algorithm for integer solution to linear programs.

- Bull. Amer. Math. Soc., 1958, 64, № 5, 275—278 (PЖMar, 1959, № 7, 7256)
146. —, Solving linear programming problems in integers. Proc. Sympos. Appl. Math., 1960, X, 211—215 (PЖMar, 1962, 2B451)
 147. —, Baumol W. J., Integer programming and pricing. *Econometrica*, 1960, 28, № 3, 521—550 (PЖMar, 1962, 7B325)
 148. —, Hoffman A. J., On the convergence of an integer programming process. *Naval Res. Log. Quart.*, 1963, 10, № 2, 121—123 (PЖMar, 1964, 6B390)
 149. Grosvenor D. D., Uses of integer programming in problems of optimization and their computational aspects. *Doct. diss. Iowa State Univ. Sci. and Technol.*, 1963. *Dissert. Abstr.*, 1964, 24, № 10, 4211 (PЖMar, 1965, 3B196D).
 150. Hadley G., *Nonlinear and dynamic programming* Reading Mass.—London, Addison—Wesley Publ. Co., 1964 (PЖMar, 1965, 8B183K)
 151. Haldi J., Isaacson L. M., A computer code for integer solutions to linear programs. *Operat. Res.*, 1965, 13, № 6, 946—959 (PЖMar, 1966, 7B264)
 152. Harris P. M. J., An algorithm for solving mixed integer programmes. *Operat. Res. Quart.*, 1964, 15, № 2, 117—132 (PЖMar, 1965, 7B165)
 153. Healy W. C., Jr., Multiple choice programming (A procedure for linear programming with zero-one variables). *Operat. Res.*, 1964, 12, № 1, 122—138, (PЖMar, 1965, 9B193)
 154. Held M., Karp R. M., A dynamic programming approach to sequencing problems. *J. Soc. Industr. and Appl. Math.*, 1962, 10, № 1, 196—210 (PЖMar, 1963, 3B388)
 155. Hess S. W., Weaver J. B., Siefeldt H. J., Whelan J. N., Zitlau P. A., Nonpartisan political redistricting by computer. *Operat. Res.*, 1965, 13, № 6, 998—1006 (PЖMar, 1966, 9B202)
 156. Ignall E., Schrage L., Application of the branch and bound technique to some flowshop scheduling problems. *Operat. Res.*, 1965, 13, № 3, 400—412 (PЖMar, 1966, 3B312)
 157. Ivănescu P. L., Outline of the pseudo-boolean method for integer polynomial programming. *Bull. Acad. polon sci. Ser. sci. math. astron. et phys.*, 1964, 12, № 11, 685—686 (PЖMar, 1966, 6B243)
 158. —, Programmation polynomiale en nombres entières. *C. r. Acad. sci.*, 1964, 258, № 2, 424—427 (PЖMar, 1964, 11B267)
 159. —, Pseudo-boolean programming and applications. *Lect. Notes in Math.* 1965, № 9 (PЖMar, 1967, 1B309)
 160. —, Pseudo-boolean programming with special restraints. Applications to graph theory. *Electron. Informationsverarb. und Kybernet.*, 1965, 1, № 3, 165—183 (PЖMar, 1966, 9B198)
 161. —, Rosenberg I., Rudeanu S. Asupra determinării minimelor funcțiilor pseudobooleene. *Studii și cercetări mat. Acad. PPR*, 1963, 14, № 3, 359—364 (PЖMar, 1965, 6B176)
 162. —, —, Problems of pseudo-boolean programming. *Colloq. applic. math. to econ.* Budapest, 1963, Budapest, Hung. Acad. Sci., 1965, 141—144 (PЖMar, 1966, 2B266)
 163. Jaeschke G., Das Reihenfolgeproblem für Erzeugnisse mit gleichem Ablaufplan (RF1). *Electron. Rechenanlag.*, 1964, 6, № 3, 137—144 (PЖMar, 1965, 5B218)
 164. Johnson E. L., Network flows, graphs, and integer programming. *Doct. diss. Berkeley, Univ. Calif.*, 1965. *Dissert. Abstr.*, 1965, 26, № 2, 1063—1064 (PЖMar, 1966, 10B201D)
 165. Joksch H. C., *Lineares Programmieren*. Tübingen, J. C. B. Mohr (Paul Siebeck), 1962 (PЖMar, 1963, 8B372K).
 166. Karp R. M., Minimum-redundancy coding for the discrete noiseless

- channel. IRE Transactions of the professional group in information theory, 1961, vol. IT-7, № 1, 27—38
167. —, McFarlin F. E., Roth J. P., Wilts J. R., A computer program for the synthesis of combinatorial switching circuits. Proc. Second Annual Sympos., on Switching Circuit Theory and Logical Design. AIEE Publication S—134(1961)
 168. Kataoka S., A method of discrete programming problem. J. Operat. Res. Japan., 1958, 2, № 1, 27—42 (PЖMar, 1962, 1B321).
 169. Kelley J. E., Techniques for storage allocation algorithms. Commun. Assoc. Comput. Machinery, 1961, 4, № 10, 449—454 (PЖMar, 1962, 10B350)
 170. —, The cutting-plane method for solving convex programs. J. Soc. Industr. Appl. Math., 1960, 8, № 4, 703—712 (PЖMar, 1962, 11B364)
 171. Kondor G., Comments on the solution of the integer linear programming problem. Colloq. applic. math. to econ. Budapest, 1963, Budapest, 1965, (PЖMar, 1966, 2B264)
 172. Krekó B., On a discrete programming problem. Colloq. Foundat. Math.. Math. Mech. and Applic., Tihany, 1962, Budapest, 1965, 269—270 (PЖMar, 1966, 2B267)
 173. Krelle W., Ganzzahlige Programmierungen: Theorie und Anwendung in der Praxis. Unternehmensforschung, 1958, 2, № 4, 161—175
 174. Kühn H. W., Baumol W. J., An approximative algorithm for the fixed-charges transportation problem. Naval Res. Log. Quart., 1962, 9, № 1, 1—15 (PЖMar, 1963, 5B396)
 175. Künzi H. P., Oettli W., Integer quadratic programming. Recent Advances Math. Program., New York—San Francisco—Toronto—London, McGraw—Hill Book Co., Inc., 1963, 303—308 (PЖMar, 1965, 3B210)
 176. —, —, Une méthode de résolution de programmes quadratiques en nombres entiers sous des fonctions objectives strictement convexes. C. r. Acad. sci. Paris, 1961, 252, № 10, 1415—1417 (PЖMar, 1962, 4B317)
 177. Lambert F., Programmes en nombres entiers et programmes mixtes. Metra, 1962, 1, № 1, 93—112, 120, 124, 128 (PЖMar, 1963, 12B552)
 178. —, Programmes linéaires mixtes. Cahiers Centre d'Etudes Rech. Opérat., 1960, 2, № 1-2, 47—126
 179. Land A. H., A problem of assignment with interrelated costs. Operat. Res. Quart., 1963, 14, № 2, 185—199 (PЖMar, 1964, 5B359)
 180. —, Doig A. G., An automatic method of solving discrete programming problems. Econometrica, 1960, 28, № 3, 497—520 (PЖMar, 1962, 4B305)
 181. Langmaak H., Algorithm 263 Gomory 1. Communications of the ACM, 1965, 8, № 10, 601—602
 182. Lawler E. L., The quadratic assignment problem. Manag. Sci., 1963, 9, № 4, 586—599 (PЖMar, 1964, 8B319)
 183. —, Wood D. E., Branch-and-bound methods: a survey. Operat. Res., 1966, 14, № 4, 699—717 (PЖMar, 1967, 3B324)
 184. Little J. D. C., The synchronization of traffic signals by mixed-integer linear programming. Operat. Res., 1966, 14, № 4, 568—594 (PЖMar, 1967, 3B327)
 185. —, Murty K. G., Sweeney D. W., Karel C., An algorithm for the traveling salesman problem. Operat. Res., 1963, 11, № 6, 972—989 (PЖMar, 1964, 8B321)
 186. Lomnicki Z. A., A «branch-and-bound» algorithm for the exact solution of the three-machine scheduling problem. Operat. Res. Quart., 1965, 16, № 1, 89—100 (PЖMar, 1965, 10B253)
 187. Markovitz H. M., Manne A. S., On the solution of discrete program-

- ming problems. *Econometrica*, 1957, 25, № 1, 84—110
188. **Martin G. T.**, An accelerated Euclidean algorithm for integer linear programming. *Recent Advances Math. Program.*, New—York—San Francisco—Toronto—London, McGraw—Hill Book Co., Inc., 1963, 311—317 (PЖMar, 1965, 3B267)
 189. **McCluskey J. F.**, Error correcting codes—a linear programming approach. *Bell System Techn. J.*, 1959, 38, 1485—1512
 190. —, Minimization of Boolean functions. *Bell. System. Techn. J.*, 1956, 35, 1417—1444
 191. **Miller C. E.**, **Tucker A. W.**, **Zemlin R. A.**, Integer programming formulation of traveling salesman problems. *J. Assoc. Comput. Machinery*, 1960, 7, № 4, 326—329 (PЖMar, 1962, 1B327)
 192. **Morlet E.**, Codage de la résolution de programmes linéaires mixtes. *Cahiers Centre d'Etudes Rech. Opérat.*, 1960, 2, № 2, 127—160
 193. **Nelson L. D.**, On a special class of problems in integer programming. *Doct. diss. Ohio State Univ.*, 1965. *Dissert. Abstrs.*, 1965, 26, № 3, 1671 (PЖMar, 1966, 8B164D)
 194. **Oyahia A.**, Programmes linéaires a variables discrètes. *Rev. franç. rech. opérat.*, 1962, 6, № 22, 55—75 (PЖMar, 1963, 5B398)
 195. **Page E. S.**, On Monte Carlo methods in congestion problems. I. Searching for an optimum in discrete situations. *Operat. Res.*, 1965, 13, № 2, 291—299 (PЖMar, 1966, 2B304)
 196. **Panaget J.—P.**, Programmes a variables entières. Programmes a variables mixtes. *Prévisions, calcul et réalités*, Paris, 1965, 181—195 (PЖMar, 1965, 9B191)
 197. **Piehler J.**, Zur Anwendung der Methode von Gomory auf gemischt-ganzzahlige lineare Optimierung. *Math. und Wirtschaft*, 1964, 2, 48—53
 198. **Radó F.**, Programme liniara cu conditii logice. *Comun. Acad. RPR*, 1963, 13, № 12, 1039—1042 (PЖMar, 1964, 7B405)
 199. —, Un algorithme pour résoudre certains problèmes de programmation mathématique. *Mathematica (RPR)* 1964, 6, № 1, 105—116 (PЖMar, 1966, 5B237)
 200. **Reiter S.**, **Sherman G.**, Discrete optimizing. *J. Soc. Industr. Appl. Math.*, 1965, 13, № 3, 864—889 (PЖMar, 1966, 5B238)
 201. **Roth A. P.** Minimization over boolean trees. *IBM J. Research Developm.*, 1960, 4, 543—558
 202. —, **Karp R. M.**, Minimization over boolean Graphs. *IRH J. Research Developm.*, 1962, 6, 227—238 (PЖMar, 1964, 7B283)
 203. —, **Wagner E. G.**, Algebraic topological methods for the synthesis of switching systems. Part III: Minimization of non-singular boolean trees. *IBM J. Resant. Developm.*, 1959, 3, 326—344
 204. **Schooner B. A.**, The incorporation of step functions and ramp functions into a linear programming model. *Operat. Res.*, 1964, 12, № 5, 773—777 (PЖMar, 1965, 4B210)
 205. **Simonnard M.**, *Programmation linéaire*. Paris, Dunod, 1962, (PЖMar, 1963, 5B388K)
 206. **Van Slyke R.**, **Wets R.**, On diagonalization methods in integer programming. *Abstract. Recent Advances Mathem. program.*, New York—San Francisco—Toronto—London, McGraw—Hill Book Co., 1963, 309—310
 207. **Spielberg K.**, On the fixed charge transportation problems. ; «*Assoc. Comput. Mach. Proc. 19th Nat. Conf.—Philadelphia, Pa.*, 1964. New—York, N. Y., 1964, Al. 1/1—Al. 1/13 (PЖMar, 1966, 9B196)
 208. **Šrinivasan A. V.**, An investigation of some computational aspects of integer programming. *J. Assoc. Comput. Machinery*, 1965, 12, № 4, 525—535 (PЖMar, 1966, 6B244)
 209. **Story A. E.**, **Wagner H. M.**, Computational experience with integer

- programming for job-shop scheduling. Industrial scheduling, Prentice Hall, Englewood Cliffs, N. J., 1963
210. **Terebesi M.**, Bemerkungen zum Verfahren von Gomory zur Bestimmung ganzzahliger Lösungen von linearen Programmen. Unternehmensforschung, 1961, 5, № 4, 197—215 (PЖMat, 1963, 10B325)
 211. **Thompson G. L.**, The stopped simplex method. I. Basic theory for mixed integer programming; integer programming. Rev. franç. rech. opérat., 1964, 8, № 31, 159—182 (PЖMat, 1966, 6B241)
 212. **Vajda S.**, Mathematical programming. Reading, Mass., Addison-Wesley Publ., Co., 1961 (PЖMat, 1963, 3B412K)
 213. **Vietorisz T.**, Industrial development planning models with economies of scale and indivisibilities. Papers Res. Sci. Assoc., 1963 (1964), 12, 157—192
 214. **Wagner H. M.**, An integer linear programming model for machine scheduling. Naval Res. Log. Quart., 1959, 6, № 2, 131—140 (PЖMat, 1961, 4B98)
 215. **Walker R. S.**, An enumerative technique for a class of combinatorial problems. Am. Math. Soc. Sympos. Appl. Math. Proc., 1960, 10, 91—94
 216. **Wegstein J. M.**, Algorithms. Commun. Assoc. Comput. Machinery., 1963, 6, № 2, 67—69 (PЖMat, 1963, 10B386)
 217. **White W. W.**, On Gomory's mixed integer algorithm. Senior Thesis, Princeton University, May 1, 1961
 218. **Witzgall V. C.**, An all-integer programming algorithm with parabolic constraints. J. Soc. Industr. and Appl. Math., 1963, 11, № 4, 855—871 (PЖMat, 1965, 1B259)
 219. **Young R. D.**, A primal (all-integer) integer programming algorithm. J. Res. Nat. Bur. Standards, 1965, B69, № 3, 213—250 (PЖMat, 1966, 9B190)
 220. **Zimmerman K.**, Zobecnění Gomoryho algoritmu na případ, kdy je některé proměnné v úloze lineárního programování musí být celočíselné. Ekon.-mat. obzor., 1965, 1, № 2, 164—181 (PЖMat, 1965, 12B261).

Дополнение к библиографии

- 1*. **Багриновский К. А.**, Об одном способе минимизации выпуклой функции на невыпуклом множестве. Научн. тр. Новосиб. ун-та. Сер. экон., 1966, вып. 8, 67—78
- 2*. **Вотяков А. А.**, Алгоритм решения обобщенной целочисленной задачи линейного программирования. В сб. Математические методы в некоторых задачах оптимального планирования. Свердловск, изд-во Наука, 1967, 73—82
- 3*. **Хачатуров В. Р.**, Определение оптимального и всех близких к нему вариантов размещения предприятий с ограниченными сверху объемами производства. Изв. АН КазССР. Сер. физ.-матем., 1967, № 3, 38—43
- 4*. **Freeman R. J.**, Computational experience with a «Balasian» integer programming algorithm. Operat. Res., 1966, 14, № 5, 935—941
- 5*. **Gomory R. E.**, Faces of an integer polyhedron. Proc. Nat. Acad. Sci. U. S. A., 1967, 57, № 1, 16—18