Экономико-
математическая
библиотека

В. А. ЕМЕЛИЧЕВ
В. И. КОМЛИК

МЕТОД ПОСТРОЕНИЯ ПОСЛЕДОВАТЕЛЬНОСТИ ПЛАНОВ ДЛЯ РЕШЕНИЯ ЗАДАЧ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ

ЭКОНОМИКО-
МАТЕМАТИЧЕСКАЯ
БИБЛИОТЕКА

В. А. ЕМЕЛИЧЕВ, В. И. КОМЛИК

МЕТОД ПОСТРОЕНИЯ ПОСЛЕДОВАТЕЛЬНОСТИ ПЛАНОВ ДЛЯ РЕШЕНИЯ ЗАДАЧ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ

МОСКВА «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
1981

22.18
Е 60
УДК 519.6

Метод построения последовательности планов для решения задач дискретной оптимизации. Емеличев В. А., Комлик В. И.— М.: Наука. Главная редакция физико-математической литературы, 1981. — 208 с.

В книге дано систематическое изложение прикладных и теоретических проблем, связанных с применением разработанного авторами общего метода решения задач дискретной оптимизации. Этот метод оказался достаточно мощным средством решения широкого класса задач оптимального планирования и управления. С его помощью успешно решен ряд практических задач оптимального отраслевого планирования, а также задач оптимизации производственного планирования в АСУ. Наряду со строгостью изложения математических результатов, существенное место уделяется вопросам практического использования предлагаемых в книге подходов и их вычислительным аспектам.

Книга будет полезной для широкого круга научных работников, специализирующихся в области прикладной математики.

Табл. 55, илл. 1, библ. 119 назв.

Е $\frac{20205-091}{053(02)-81}$ 68-81. 1502000000

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ

Большинство моделей и методов, которые излагаются в этой книге, являются результатом обобщения опыта, накопленного авторами в процессе участия в практических работах по созданию автоматизированных систем управления предприятиями, а также в разработке оптимальных перспективных планов развития отраслей народного хозяйства.

Хорошо известно, что к задачам дискретной оптимизации, т. е. к задачам отыскания максимума или минимума некоторой функции на некотором дискретном множестве, формально сводятся многие оптимизационные задачи, имеющие важное народнохозяйственное значение. Это — задачи организации производства, управления отраслью, проектирования техники и др. Использование моделей дискретной оптимизации позволяет в ряде случаев учесть условия неделимости искомых объектов, описать дизъюнктивные ограничения на область их существования, ввести элементы комбинаторики и т. п.

В последние два десятилетия разработано большое количество разнообразных методов решения задач дискретной оптимизации, что отражено в многочисленной литературе по этой проблематике. Для целей практики несомненный интерес представляет поиск эффективных методов. В то же время потребности теории дискретной оптимизации диктуют необходимость создания общих схем решения.

Одной из универсальных схем решения задач оптимизации является схема последовательного анализа вариантов, общий формализм которой разработан в начале 60-х годов В. С. Михалевичем [72, 73]. Алгоритмы, реализующие эту схему, используют различные идеи перебора и анализа вариантов и успешно применяются к решению разнообразных дискретных задач оптимизации [14, 74, 75]. Частным случаем схемы последовательного анализа вариантов является динамическое программирование, в основе которого лежит принцип оптимальности Р. Беллмана [8]. Хотя на основе идей динамического программирования и разработаны алгоритмы решения обширного класса задач дискретной оптимизации (задачи о ранце, о коммивояжере, задачи целочисленного линейного программирования и др.), но на пути применения этих алгоритмов возникают значительные трудности при возрастании размерности задач. Эти трудности частично удается преодолеть при решении некоторых задач, используя метод множителей Лагранжа [9], а также методы последовательных приближений, разработанные Н. Н. Моисеевым [77—79] и его учениками.

Другой универсальной и гибкой схемой решения задач дискретного программирования является метод ветвей и границ, представляющий собой направленный перебор с отсеиванием неперспективных подмножеств планов. Этому методу посвящена обширная литература (см., например, [52, 61, 62, 108]).

В широком классе эвристических алгоритмов особо выделяются локальные алгоритмы, которые были введены и исследованы для решения задач дискретной природы Ю. И. Журавлевым [44—48]. Эти алгоритмы были использованы для построения минимальных нормальных форм алгебр логики [47, 49], для решения задач целочисленного линейного программирования [48, 110] и практических задач оптимального резервирования ресурса, распределенного во времени [102, 103].

Из комбинаторных методов, предназначенных для решения специальных задач нелинейного целочисленного программирования, следует выделить оригинальный метод последовательных расчетов, предложенный В. П. Черениным [101]. На основе метода последовательных расчетов В. Р. Хачатуровым разработан аппроксимационно-комбинаторный метод [99], успешно использовавшийся для решения прикладных задач.

Среди приближенных методов решения задач дискретной оптимизации упомянем здесь методы, предложенные И. В. Сергиенко и его сотрудниками [3, 4, 64, 65, 89, 89′, 90]. На основе этих методов разработаны специализированные пакеты прикладных программ для решения задач дискретной оптимизации.

Предлагаемый в данной книге единый подход к решению широкого класса задач дискретной оптимизации основан на построении и анализе определенной последовательности планов. Сущность этого метода состоит в замене исходной экстремальной задачи более простой, вспомогательной задачей, для которой мы умеем находить не только оптимальный план, но и следующие за ним в порядке ухудшения новой целевой функции планы. Построение таких планов продолжаем до выполнения критерия оптимальности — в результате получаем оптимальный план решения исходной задачи. Эта простая идея послужила источником многочисленных исследований, как теоретических, так и прикладных. Предложенная авторами общая схема — метод построения последовательности планов — объединяет такие, казалось бы различные, методы, как метод ветвей и границ и метод отсечения. В то же время на основе этого подхода удалось разработать эффективные алгоритмы решения прикладных задач. Успешно решен ряд практических задач оптимального планирования работы предприятий, а также задач развития и размещения производства.

Книга дает представление о теоретических основах, методах и приложениях одного из важных разделов математической теории планирования и управления — дискретной оптимизации.

В главе I излагаются модели дискретной оптимизации, к которым сводятся разнообразные задачи планирования и управления (в основном экономического характера). Рассматриваются задачи размещения и специализации производства, распределения капиталовложений, формирования производственной программы предприятия и др. Эта глава носит иллюстративный характер, так как детальное изложение данных вопросов требует написания отдельной книги.

Глава II посвящена обоснованию предлагаемого авторами метода. В основе этого метода лежит, как указывалось выше, идея построения последовательности планов вспомогательной задачи в порядке ухудшения оценочной функции (миноранты или мажоранты). Предлагаются процедуры упорядочения планов вспомогательной задачи и изложен метод решения общей задачи дискретного программирования.

В главе III описаны алгоритмы решения однопродуктовых задач размещения производства с учетом затрат на доставку сырья и готовой продукции, многопродуктовых и межотраслевых задач размещения. Алгоритмы иллюстрируются на конкретных числовых примерах. Обсуждаются вычислительные аспекты применения разработанных методов. Излагается опыт решения практических задач размещения производства.

Глава IV посвящена задачам целочисленного линейного программирования, к которым сводятся многочисленные практические задачи. Предлагаются алгоритмы их решения и приводятся результаты счета.

Глава V отведена для решения других задач дискретной оптимизации. Здесь подробно описаны алгоритмы решения задач размещения в непрерывной постановке. В основе этих алгоритмов лежит процедура построения последовательности выборок. Для задачи вогнутого программирования с сепарабельной функцией цели при линейных ограничениях предлагается алгоритм, состоящий в построении последовательности вершин многогранника ограничений в порядке неубывания линейной миноранты. Изложены также методы решения задач надежности, комплектации, космической навигации. Отдельный параграф посвящен решению задач минимизации на множестве подстановок. В частности, рассматриваются задачи минимизации линейной формы и скалярного произведения матриц, задача разрезания графа на подграфы с фиксированным числом вершин. Метод построения последовательности планов применяется и для нахождения множества неулучшаемых планов в дискретных многокритериальных задачах.

При написании данной книги авторы особое внимание стремились уделять вопросам практического внедрения предлагаемых в книге подходов и их вычислительным аспектам. Этим вопросам посвящена глава VI. Здесь дано описание пакета прикладных программ для решения задач целочисленного линейного программирования и решаемых с его помощью задач планирования горных работ, составления графика ремонта оборудования, расчета вариантов запуска сырья в производство.

Книга написана на основе лекций, прочитанных авторами в БГУ им. В. И. Ленина, Йенском университете им. Ф. Шиллера (ГДР), Белорусском государственном институте народного хозяйства им. В. В. Куйбышева и на XV Всесоюзной летней школе по методам оптимизации и проблемам управления (1975 г.). Авторам приятно выразить свою признательность Н. Н. Моисееву за инициативу написания этой книги.

В. А. Емеличев, В. И. Комлик

ГЛАВА I

ПРИКЛАДНЫЕ ЗАДАЧИ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ

В этой главе рассматривается широкий круг задач оптимизации, возникающих в процессе планирования и управления на уровне республики (размещение производительных сил), отрасли (распределение капитальных вложений, специализация производства) и предприятия (формирование производственной программы, выбор парка машин, загрузка оборудования).

В основном здесь приведены те модели, для решения которых разработаны алгоритмы на основе метода построения последовательности планов.

§ 1. Однопродуктовые задачи размещения производства

Проблема размещения и развития производительных сил различных районов является составным элементом перспективного отраслевого и регионального планирования. Наиболее известной задачей здесь является так называемая *задача размещения*, которая сводится к составлению такого плана реконструкции действующих и размещения новых предприятий, который бы обеспечивал удовлетворение спроса, заданного в территориальном разрезе, с минимальными народнохозяйственными затратами. В качестве критерия оптимальности при решении задач по размещению обычно принимается минимум затрат на производство и транспортировку продукции. Поэтому такие задачи называются *производственно-транспортными задачами размещения.*

В расчетах по оптимальным мощностям, размещению и специализации промышленных предприятий важное значение имеет учет транспортного фактора. Транспорт играет важную роль при определении оптимальных мощностей промышленных предприятий.

Известно, что с увеличением мощности предприятия уменьшаются себестоимость продукции и удельные капитальные вложения. Преимущества крупных предприятий нередко связаны с возможностью применения качественно иных, более эффективных технологических способов.

Вместе с тем увеличение мощности предприятий приводит к расширению зоны обслуживания готовой продукцией и, следовательно, к росту расходов. При этом величина транспортных расходов может достичь такого размера, что перекроет экономию, полученную за счет концентрации производства (увеличения мощности предприятий).

Наиболее простыми задачами размещения являются *однопродуктовые модели*. Рассмотрим несколько типичных моделей, возникающих в работе плановых органов на отраслевом уровне.

Имеются пункты потребления однородного продукта с заданными объемами потребления, т. е. рассчитаны потребности в продукте в территориальном разрезе. Продукт может производиться в нескольких пунктах производства, причем для каждого из этих пунктов известны мощности действующих предприятий и задан набор возможных вариантов их развития с учетом факторов обеспечения трудовыми ресурсами, энергоснабжения, наличия строительных площадок и др. Намечены пункты возможного размещения новых предприятий, для них также заданы возможные варианты развития. Для каждого такого варианта вычисляются затраты на производство продукции с учетом себестоимости и удельных капитальных вложений, составляющие суммарные приведенные затраты, в которые могут быть включены расходы на доставку сырья. Кроме того, рассчитаны затраты на перевозку единицы продукции из каждого возможного пункта производства в каждый пункт потребления. Задача состоит в выборе для каждого пункта производства такого объема производства и в определении такого плана перевозок, чтобы суммарные затраты на производство и транспортировку были минимальными при условии полного удовлетворения потребителей.

Переведем задачу на формальный язык.

Введем обозначения:

i — номер пункта размещения действующего предприятия или возможного размещения нового ($i = \overline{1, m}$);

j — номер пункта потребления продукции ($j = \overline{1, n}$);

b_j — потребность в готовой продукции j-го пункта потребления;

x_i — искомая мощность предприятия в i-м пункте;

$f_i(x_i)$ — суммарные приведенные затраты, связанные с размещением в i-м пункте предприятия мощности x_i. Эти затраты определяются по формуле

$$f_i(x_i) = c_i(x_i) + EK_i(x_i) + T_i(x_i),$$

где $c_i(x_i)$ — годовые эксплуатационные затраты без учета расходов на доставку сырья, $K_i(x_i)$ — единовременные капитальные вложения на строительство новых и реконструкцию действующих предприятий, E — нормативный коэффициент эффективности капитальных вложений (для различных отраслей он меняется в пределах от 0,12 до 0,2), $T_i(x_i)$ — затраты на доставку сырья в i-й пункт размещения (очевидно, что все перечисленные затраты зависят от величины мощности предприятия и от пункта размещения, поэтому всюду эти затраты представлены как функции от x_i и снабжены индексом i);

c_{ij} — транспортные затраты по доставке единицы готовой продукции из i-го пункта производства в j-й пункт потребления;

x_{ij} — искомый объем поставок готовой продукции из i-го пункта производства в j-й пункт потребления;

$\Gamma_i = \{a_i^1, a_i^2, \ldots, a_i^{s_i}\}$ — набор возможных мощностей для i-го пункта размещения.

Исходная информация, необходимая для решения задачи, может быть представлена в виде таблицы 1.

Задача 1.1. Задача состоит в определении для каждого i-го пункта производства такого объема производства x_i из заданного набора Γ_i и такого плана перевозок x_{ij} готовой продукции в каждый пункт потребления, чтобы суммарные производственно-транспортные затраты

$$\sum_{i=1}^{m} f_i(x_i) + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1.1}$$

Таблица 1

Возможные пункты производства A_i	Возможные объемы производства a_i^h	Приведенные затраты $f_i(a_i^h)$	Пункты потребления B_j			
			B_1	B_2	...	B_n
			Объемы потребления b_j			
			b_1	b_2	...	b_n
A_1	a_1^1 a_1^2 ... $a_1^{s_1}$	$f_1(a_1^1)$ $f_1(a_1^2)$ ... $f_1(a_1^{s_1})$	c_{11}	c_{12}	...	c_{1n}
A_2	a_2^1 a_2^2 ... $a_2^{s_2}$	$f_2(a_2^1)$ $f_2(a_2^2)$ ... $f_2(a_2^{s_2})$	c_{21}	c_{22}	...	c_{2n}
...	...	...	...	...	...	...
A_m	a_m^1 a_m^2 ... $a_m^{s_m}$	$f_m(a_m^1)$ $f_m(a_m^2)$ ... $f_m(a_m^{s_m})$	c_{m1}	c_{m2}	...	c_{mn}

были минимальными и удовлетворялись потребности b_j каждого потребителя в готовой продукции.

Таким образом, требуется минимизировать функцию (1.1) при следующих ограничениях.

1. Продукция каждого предприятия должна полностью вывозиться:

$$\sum_{j=1}^{n} x_{ij} = x_i, \quad i = \overline{1, m}. \tag{1.2}$$

2. Потребность каждого потребителя в готовой продукции должна быть полностью удовлетворена:

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = \overline{1, n}. \tag{1.3}$$

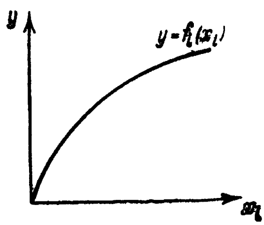
3. Мощность предприятия в i-м пункте может принимать одно из значений заданного набора:

$$x_i \in \Gamma_i = \{a_i^1, a_i^2, \ldots, a_i^{s_i}\}, \quad i = \overline{1, m}. \qquad (1.4)$$

4. Объемы перевозок продукции должны быть неотрицательными числами:

$$x_{ij} \geqslant 0, \quad i = \overline{1, m}, \quad j = \overline{1, n}. \qquad (1.5)$$

Сформулированная задача является типичной задачей дискретного программирования в силу того, что неизвестные x_i $(i = \overline{1, m})$ могут принимать значения только из заданных конечных наборов Γ_i. Так как функция приведенных затрат $f_i(x_i)$ является нелинейной, а точнее, выпуклой вверх (см. рисунок), то задача (1.1)—(1.5) есть задача нелинейного программирования. Вид этой функции говорит о том, что производственные затраты хотя и возрастают с увеличением мощности предприятия, но темпы этого роста падают.



Это объясняется снижением себестоимости продукции при концентрации производства. В случае, когда условие (1.4) не является обязательным, т. е. когда мощность предприятия может изменяться непрерывно и принимать любое значение, получаем задачу (1.1)—(1.3), (1.5) размещения в непрерывной постановке.

На практике часто возникают задачи размещения, в которых не требуется, чтобы суммарный спрос $b = \sum_{j=1}^{n} b_j$ выражался в виде суммы $b = \sum_{i=1}^{m} x_i$, где $x_i \in \Gamma_i$ $(i = \overline{1, m})$. Это возможно при нарушении баланса, т. е. либо при нарушении условий (1.2), либо — условий (1.3). В этих случаях могут быть сформулированы следующие две задачи размещения производства.

Задача 1.2. Требуется минимизировать функцию (1.1) при условиях (1.3)—(1.5) и условии

$$\sum_{j=1}^{n} x_{ij} \leqslant x_i, \quad i = \overline{1, m}$$

(т. е. когда объем вывозимой продукции не превышает мощности каждого предприятия).

Задача 1.3. Требуется минимизировать функцию (1.1) при условиях (1.2), (1.4), (1.5) и условии

$$\sum_{i=1}^{m} x_{ij} \geqslant b_j, \quad j=\overline{1, n}$$

(т. е. когда объем ввозимой продукции не меньше потребления).

§ 2. Задача размещения с двойным транспортом

Одним из недостатков предыдущих моделей является то, что затраты $T_i(x_i)$ на доставку сырья в пункты размещения определяются отдельно для каждого пункта, т. е. не учитываются сложные взаимосвязи между поставщиками и потребителями сырья. Все это упрощает реальную ситуацию и приводит к нахождению лишь приближенного решения реальной задачи размещения.

Рассматриваемая ниже модель позволяет, в отличие от предыдущих, более полно учитывать затраты на доставку сырья и состоит в следующем. Известны пункты по добыче сырья и запасы сырья. Кроме того, известны мощности действующих предприятий по выпуску готовой продукции, возможные варианты их развития и реконструкции. Намечены пункты возможного размещения новых предприятий. Для каждого из этих пунктов задан набор мощностей и соответствующие приведенные затраты, а также потребности в сырье. Известны пункты потребления готовой продукции и объемы потребления. Заданы транспортные расходы как по доставке единицы готовой продукции от пунктов производства до пунктов потребления, так и по доставке единицы сырья от пунктов добычи до пунктов производства.

Задача состоит в определении для каждого пункта производства такого объема производства из заданного набора мощностей и такого плана перевозок сырья и готовой продукции, чтобы суммарные производственно-транспортные затраты были минимальными и удовлетворялись спрос каждого потребителя на готовую продукцию, а также потребность в сырье каждого пункта производства.

Эта задача предназначена для решения однопродуктовых (или сводимых к ним) задач, где транспортные расходы на доставку сырья и готовой продукции имеют значительный удельный вес, а соотношение удельных транспортных затрат приближается к единице (цельномолочная, комбикормовая, отдельные отрасли деревообрабатывающей промышленности, промышленности строительных материалов и т. д.).

Переведем задачу на формальный язык. Введем обозначения:

i — номер пункта размещения действующего предприятия или возможного размещения нового ($i = \overline{1, m}$);

j — номер пункта потребления готовой продукции ($j = \overline{1, n}$);

k — номер пункта добычи сырья ($k = \overline{1, p}$);

b_j — потребность в готовой продукции j-го пункта потребления;

$\bar{b}_k$ — запасы сырья в k-м пункте добычи;

x_i — искомая мощность предприятия в i-м пункте размещения;

$d_i(x_i)$ — потребность предприятия мощности x_i в сырье;

$f_i(x_i)$ — приведенные затраты, связанные с размещением в i-м пункте предприятия мощности x_i. Эти затраты определяются по формуле

$$f_i(x_i) = c_i(x_i) + EK_i(x_i),$$

где $c_i(x_i)$ — годовые эксплуатационные затраты без учета расходов на доставку сырья, $K_i(x_i)$ — единовременные капитальные вложения на строительство новых и реконструкцию действующих предприятий, E — нормативный коэффициент эффективности капитальных вложений;

c_{ij} — транспортные затраты по доставке единицы готовой продукции из i-го пункта размещения в j-й пункт потребления;

x_{ij} — искомый объем поставок готовой продукции из i-го пункта производства в j-й пункт потребления;

$\bar{c}_{ki}$ — транспортные затраты по перевозке единицы сырья из k-го пункта добычи в i-й пункт производства;

$\bar{x}_{ki}$ — искомый объем перевозок сырья из k-го пункта добычи в i-й пункт производства;

$\Gamma_i = \{a_i^1, a_i^2, \ldots, a_i^{s_i}\}$ — набор мощностей для i-го пункта размещения.

Каждому набору типовых мощностей $\Gamma_i=\{a_i^1, a_i^2, \ldots$ $\ldots, a_i^{s_i}\}$ ставится в соответствие набор объемов потребления сырья

$$d(a_i^1),\ d(a_i^2),\ \ldots,\ d(a_i^{s_i}).$$

Задача состоит в определении для каждого i-го пункта размещения такого объема производства x_i из заданного набора мощностей Γ_i и такого плана перевозок сырья $\bar{x}_{ki}$ и готовой продукции x_{ij}, чтобы суммарные производственно-транспортные затраты

$$\sum_{i=1}^{m} f_i(x_i) + \sum_{k=1}^{p} \sum_{i=1}^{m} \bar{c}_{ki}\bar{x}_{ki} + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}x_{ij} \qquad (2.1)$$

были минимальными и удовлетворялся спрос b_j каждого потребителя на готовую продукцию, а также потребность в сырье каждого i-го пункта производства. Таким образом, требуется минимизировать функцию (2.1) при следующих ограничениях.

1. Продукция каждого предприятия должна полностью вывозиться:

$$\sum_{j=1}^{n} x_{ij} = x_i, \quad i = \overline{1, m}.$$

2. Потребность каждого потребителя в готовой продукции должна быть полностью удовлетворена:

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = \overline{1, n}.$$

3. Потребность каждого предприятия в сырье должна быть полностью удовлетворена:

$$\sum_{k=1}^{p} \bar{x}_{ki} = d(x_i), \quad i = \overline{1, m}.$$

Таблица 2

Возможные пункты производства A_i	Возможные объемы производства a_i^h	Объемы потребления сырья $d(a_i^h)$	Приведенные затраты $f_i(a_i^h)$	Пункты потребления B_j			
				B_1	B_2	...	B_n
				Объемы потребления b_j			
				b_1	b_2	...	b_n
A_1	a_1^1 a_1^2 ... $a_1^{s_1}$	$d(a_1^1)$ $d(a_1^2)$ ... $d(a_1^{s_1})$	$f_1(a_1^1)$ $f_1(a_1^2)$ ... $f_1(a_1^{s_1})$	c_{11}	c_{12}	...	c_{1n}
A_2	a_2^1 a_2^2 ... $a_2^{s_2}$	$d(a_2^1)$ $d(a_2^2)$ ... $d(a_2^{s_2})$	$f_2(a_2^1)$ $f_2(a_2^2)$ ... $f_2(a_2^{s_2})$	c_{21}	c_{22}	...	c_{2n}
...	...	...	...	...	...	...	...
A_m	a_m^1 a_m^2 ... $a_m^{s_m}$	$d(a_m^1)$ $d(a_m^2)$ ... $d(a_m^{s_m})$	$f_m(a_m^1)$ $f_m(a_m^2)$ ... $f_m(a_m^{s_m})$	c_{m1}	c_{m2}	...	c_{mn}

Таблица 3

Пункты добычи сырья C_k	Возможные пункты производства A_i			
	A_1	A_2	...	A_m
C_1	$\bar{c}_{11}$	$\bar{c}_{12}$	...	$\bar{c}_{1m}$
C_2	$\bar{c}_{21}$	$\bar{c}_{22}$	...	$\bar{c}_{2m}$
...	...	...	...	...
C_p	$\bar{c}_{p1}$	$\bar{c}_{p2}$	...	$\bar{c}_{pm}$

4. Количество сырья, вывозимого из каждого пункта добычи, не должно превосходить запасов:

$$\sum_{i=1}^{m} \bar{x}_{ki} \leqslant \bar{b}_k, \quad k=\overline{1, p}.$$

5. Мощности предприятий в каждом пункте могут принимать значения только из заданного набора:

$$x_i \in \Gamma_i = \{a_i^1, a_i^2, \ldots, a_i^{s_i}\}, \quad i=\overline{1, m}.$$

6. Объемы поставок сырья и готовой продукции должны быть неотрицательными:

$$x_{ij} \geqslant 0, \quad \bar{x}_{ki} \geqslant 0, \quad i=\overline{1, m}, \quad j=\overline{1, n}, \quad k=\overline{1, p}.$$

Необходимая для решения задачи информация может быть задана таблицами 2 и 3.

Решение задач оптимального развития и размещения отраслей позволяет определить: пункты размещения и размеры предприятий; соотношение между новым строительством и реконструкцией действующих предприятий; предприятия, подлежащие закрытию в связи с их неэффективностью; сырьевые зоны и зоны сбыта готовой продукции предприятий; потребные капитальные вложения по направлениям и объектам.

§ 3. Многопродуктовые задачи размещения производства

В отраслевом перспективном планировании широко применяются многопродуктовые задачи размещения. В отличие от однопродуктовых задач размещения, в которых специализация предприятий предполагается заранее известной, многопродуктовые модели размещения позволяют устанавливать специализацию и оптимальные размеры предприятий, которые должны быть построены в перспективном периоде, и определять районы их размещения. Существует много различных модификаций многопродуктовых задач размещения, в той или иной степени отличающихся друг от друга. В настоящем параграфе рассматриваются две наиболее часто встречающиеся задачи.

Задача 3.1. Пусть задано m пунктов производства r видов продуктов. Каждый i-й пункт производства характеризуется множеством вариантов специализации:

$$\Gamma_i = \{A_{i1}, A_{i2}, \ldots, A_{is_i}\}, \quad i = \overline{1, m},$$

а k-й вариант специализации i-го пункта A_{ik} задается r-мерным вектором:

$$A_{ik} = (a_{ik}^1, a_{ik}^2, \ldots, a_{ik}^r),$$

компоненты

$$a_{ik}^l, \quad l = \overline{1, r},$$

которого являются объемами производства l-го продукта в i-м пункте. Для каждого варианта специализации A_{ik} заданы производственные затраты $\varphi_i(A_{ik})$, связанные с реализацией этого варианта в i-м пункте. Известно, кроме того, n пунктов потребления. Потребность j-го пункта задается вектором

$$B_j = (b_j^1, b_j^2, \ldots, b_j^r),$$

где b_j^l — потребность j-го пункта в l-м продукте. Известны затраты c_{ij}^l на перевозку единицы l-го продукта из i-го пункта производства в j-й пункт потребления.

Задача состоит в определении для каждого i-го пункта $(i = \overline{1, m})$ такого варианта специализации

$$X_i = (x_i^1, x_i^2, \ldots, x_i^r)$$

из заданного набора Γ_i и такого плана перевозок

$$x_{ij}^l, \quad i = \overline{1, m}, \quad j = \overline{1, n}, \quad l = \overline{1, r},$$

чтобы суммарные производственно-транспортные затраты

$$\sum_{i=1}^{m} \varphi_i(X_i) + \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{l=1}^{r} c_{ij}^l x_{ij}^l \tag{3.1}$$

были минимальными и удовлетворялся спрос

$$b_j^l, \quad j = \overline{1, n}, \quad l = \overline{1, r},$$

каждого потребителя.

Таким образом, требуется минимизировать функцию (3.1) при следующих ограничениях.

1. Из пунктов производства продукции должно вывозиться не больше, чем производится:

$$\sum_{j=1}^{m} x_{ij}^{l} \leqslant x_i^l, \quad i=\overline{1, m}, \quad l=\overline{1, r}.$$

2. Потребности в продукции должны удовлетворяться полностью:

$$\sum_{i=1}^{m} x_{ij}^{l}=b_j^l, \quad j=\overline{1, n}, \quad l=\overline{1, r}.$$

3. Варианты развития предприятий выбираются из заданных наборов:

$$X_i=(x_i^1, x_i^2, \ldots, x_i^r) \in \Gamma_i, \quad i=\overline{1, m}.$$

4. Объемы перевозок должны быть неотрицательными:

$$x_{ij}^l \geqslant 0, \quad i=\overline{1, m}, \quad j=\overline{1, n}, \quad l=\overline{1, r}.$$

Задача 3.2. Сформулируем *многопродуктовую задачу размещения* для случая, когда варианты специализации задаются по каждому продукту отдельно.

Пусть задано m пунктов производства r продуктов. Каждый i-й пункт производства характеризуется множеством вариантов специализации по l-му продукту:

$$A_i^l=\left(a_{i1}^l, a_{i2}^l, \ldots, a_{is_i^l}^l\right), \quad i=\overline{1, m}, \quad l=\overline{1, r}.$$

Для каждого варианта специализации a_{ij}^l заданы производственные затраты $\varphi_i^l(a_{ij}^l)$, связанные с выпуском l-го продукта в i-м пункте в количестве a_{ij}^l.

Пусть, кроме того, известно n пунктов сосредоточения сырья, причем для каждого k-го пункта сосредоточения сырья известны запасы сырья b_k $(k=\overline{1, n})$. Для производства каждого продукта используется сырье, причем для производства l-го продукта в количестве x_i^l необходимо $\lambda^l x_i^l$ единиц сырья, где λ^l — коэффициент перевода, представляющий собой количество сырья, необходимое для выпуска единицы l-го продукта. Пусть c_{ki} — стоимость перевозки единицы сырья из k-го пункта сосредоточения сырья в i-й пункт производства. Кроме того, известна потребность c^l $(l=\overline{1, r})$ в l-м продукте.

Задача состоит в определении для каждого i-го пункта ($i=\overline{1, m}$) такого объема производства l-го вида продукции x_i^l из заданного набора A_i^l и такого плана перевозок сырья x_{ki} ($k=\overline{1, n}$, $i=\overline{1, m}$), чтобы суммарные производственно-транспортные затраты

$$\sum_{i=1}^{m} \sum_{l=1}^{r} \varphi_i^l(x_i^l) + \sum_{k=1}^{n} \sum_{i=1}^{m} c_{ki}x_{ki} \tag{3.2}$$

были минимальными и удовлетворялась потребность c^l ($l=\overline{1, r}$) в l-м продукте.

Таким образом, требуется найти минимум функции (3.2) при следующих условиях.

1. Потребности предприятий в сырье должны удовлетворяться:

$$\sum_{k=1}^{n} x_{ki} = \sum_{l=1}^{r} \lambda^l x_i^l, \quad i=\overline{1, m}.$$

2. Сырья должно вывозиться не больше, чем добываться:

$$\sum_{i=1}^{m} x_{ki} \leqslant b_k, \qquad k=\overline{1, n}.$$

3. Потребность в продукции должна удовлетворяться:

$$\sum_{i=1}^{m} x_i^l = c^l, \qquad l=\overline{1, r}.$$

4. Объемы производства продукции должны принимать значения из заданных наборов:

$$x_i^l \in A_i^l, \quad i=\overline{1, m}, \quad l=\overline{1, r}.$$

5. Объемы перевозок должны быть неотрицательными:

$$x_{ki} \geqslant 0, \quad i=\overline{1, m}, \quad k=\overline{1, n}.$$

§ 4. Межотраслевые задачи внутрирайонного размещения промышленного производства

Одним из важнейших направлений совершенствования территориального планирования является оптимизация комплексного развития социалистического производства союзных республик и экономических районов СССР.

Под комплексным развитием понимается взаимосвязанное, взаимообусловленное и пропорциональное развитие всех отраслей материального производства и сферы обслуживания населения. В промышленности оно осуществляется путем размещения предприятий в виде промузлов. Размещение предприятий в одном территориально-промышленном комплексе, в особенности на одной промышленной площадке, как об этом свидетельствует современная практика проектирования и строительства промузлов, позволяет получить значительный экономический эффект по сравнению с обособленным размещением этих предприятий (агломерационный эффект).

Анализ агломерационного эффекта по факторам показывает, что экономия капитальных вложений в промузле достигается за счет:

1) создания объектов вспомогательного и обслуживающего хозяйства (энергетических, транспортных, водо- и воздухоснабжения, канализации и т. д.) и системы инженерных сетей и коммуникаций, в результате чего уменьшаются площади и объемы зданий и сооружений, протяженность коммуникаций, происходит укрупнение агрегатов и уменьшается потребность в оборудовании, сокращается количество работающих;

2) блокирования производственных зданий, предприятий и изменения условий их строительства;

3) концентрации объемов строительно-монтажных работ в пределах промузла (снижается себестоимость строительства, уменьшается потребность во временных зданиях и сооружениях);

4) создания генерального плана промузла (сокращается территория, уменьшается объем работ по подготовке и благоустройству территории).

Экономия текущих затрат в промузле достигается за счет высвобождения работников и снижения затрат на заработную плату в первую очередь на общеузловых объектах вспомогательного и обслуживающего производства, уменьшения амортизационных расходов, снижения себестоимости продукции общеузловых объектов, сокращения затрат на текущий ремонт зданий и оборудования и на их эксплуатацию.

Приведенный выше экономический эффект представляет лишь часть агломерационного эффекта, а именно ту его часть, которая достигается благодаря объединению любых промышленных предприятий, потребляющих воду, электро- и теплоэнергию и т. д. (эффект от простой агломерации производств). Дополнительный эффект от совместного размещения технологически взаимосвязанных производств выражается в экономии за счет комплексного использования сырья, снижения транспортных расходов и может быть учтен путем корректировки доли экономии единовременных и текущих затрат при совместном строительстве.

Размещение предприятий в виде территориально-промышленных комплексов позволяет более эффективно решать проблемы социально-экономического характера — формировать рациональную систему расселения и обслуживания населения. Поэтому в экономико-математической модели внутрирайонного размещения производства должна учитываться возможность совместного решения и этих проблем. В первую очередь это касается учета в качестве пунктов размещения городов, имеющих наиболее благоприятные условия для промышленного и гражданского строительства, способных выполнять функции региональных центров — центров обслуживания окружающего городского и сельского населения.

Изложенное выше позволяет сформулировать несколько межотраслевых задач внутрирайонного размещения промышленного производства, в которых в той или иной мере учитываются приведенные выше факторы.

Задача 4.1. Исходя из потребностей в продукции и общесоюзной специализации района определены объемы производства продукции промышленности на перспективный период. Произведен отбор городов первоочередного размещения промышленных предприятий. Установлен перечень предприятий, подлежащих строительству или реконструкции. По каждому из них известны объемы потребления местных ресурсов (воды, энергии, строительных площадок). Требуется найти вариант размещения, который минимизирует затраты, связанные с размещением предприятий, при соблюдении ограничений на ресурсы.

Математически задача формулируется следующим образом. В m пунктах возможного размещения $(i = \overline{1, m})$ нужно разместить n предприятий $(j = \overline{1, n})$. Известен объем потребления a_{kj} $(k = \overline{1, l};\ j = \overline{1, n})$ ресурса k-го вида j-м предприятием, а также объемы r_{ik} ресурса k-го вида в i-м пункте. Кроме того, известны затраты c_{ij}, связанные с размещением j-го предприятия в i-м пункте. Задача состоит в нахождении варианта размещения, минимизирующего суммарные затраты, связанные с размещением предприятий, при соблюдении ограничений по ресурсам.

Введем переменные

$$x_{ij} = \begin{cases} 1, & \text{если } j\text{-е предприятие размещается в } i\text{-м пункте,} \\ 0 & \text{в противном случае.} \end{cases}$$

Математически задача заключается в минимизации функции

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

при выполнении следующих условий.

1. Каждое предприятие должно быть размещено в одном из пунктов:

$$\sum_{i=1}^{m} x_{ij} = 1, \quad j = \overline{1, n}.$$

2. Для всех пунктов размещения должны соблюдаться ограничения по каждому виду ресурсов:

$$\sum_{j=1}^{n} a_{kj} x_{ij} \leqslant r_{ik}, \quad i = \overline{1, m}, \quad k = \overline{1, l}.$$

3. $\quad x_{ij} = 0$ или $1, \quad i = \overline{1, m}, \quad j = \overline{1, n}.$

Задача 4.2. В m пунктах возможного размещения $(i = \overline{1, m})$ нужно построить предприятия по выпуску v видов продукции $(t = \overline{1, v})$. Предполагается, что каждое предприятие выпускает только один вид продукции. Для каждого i-го пункта заданы наборы возможных мощностей предприятий по выпуску t-го продукта

$$\Gamma_{it}=\{a_{it}^1,\ a_{it}^2,\ \ldots,\ a_{it}^{s_{it}}\},\quad i=\overline{1,\ m},\quad t=\overline{1,\ v},$$

и соответствующие им затраты $\varphi_{it}(x_{it})$, связанные с размещением в i-м пункте предприятия мощности x_{it}. Кроме того, для каждого предприятия мощности x_{it} заданы величины $p_{tk}(x_{it})$ $(t=\overline{1,\ v},\ k=\overline{1,\ l})$, характеризующие потребность в ресурсах k-го вида. Наличие ресурса k-го вида в i-м пункте задано и выражается величиной r_{ik} $(i=\overline{1,m}, k=\overline{1,l})$.

Задача состоит в определении для каждого i-го пункта такого объема производства t-го вида продукции x_{it} из заданного набора A_{it} и такого плана перевозок x_{ijt} $(i=\overline{1,\ m},\ j=\overline{1,\ n},\ t=\overline{1,\ v})$, чтобы суммарные производственно-транспортные затраты

$$\sum_{i=1}^{m}\sum_{t=1}^{v}\varphi_{it}(x_{it})+\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{t=1}^{v}c_{ijt}x_{ijt} \tag{4.1}$$

были минимальными и удовлетворялся спрос b_{jt} $(j=\overline{1,\ n},\ t=\overline{1,\ v})$ каждого потребителя при соблюдении ограничений на ресурсы.

Таким образом, требуется найти минимум функции (4.1) при следующих ограничениях.

1. Продукция каждого предприятия должна полностью вывозиться:

$$\sum_{j=1}^{n}x_{ijt}=x_{it},\quad i=\overline{1,\ m},\quad t=\overline{1,\ v}.$$

2. Потребности потребителей в продукции должны быть полностью удовлетворены:

$$\sum_{i=1}^{m}x_{ijt}=b_{jt},\quad j=\overline{1,\ n},\quad t=\overline{1,\ v}.$$

3. Расход ресурсов в каждом пункте размещения не должен превосходить их наличия:

$$\sum_{t=1}^{v}p_{tk}(x_{it})\leqslant r_{ik},\quad i=\overline{1,\ m},\quad k=\overline{1,\ l}.$$

4. Мощности предприятий могут принимать значения только из заданных наборов:

$$x_{it} \in \Gamma_{it}, \quad i = \overline{1, m}, \quad t = \overline{1, v}.$$

5. Объемы перевозок должны быть неотрицательными числами:

$$x_{ijt} \geqslant 0, \quad i = \overline{1, m}, \quad j = \overline{1, n}, \quad t = \overline{1, v}.$$

Задача 4.3. Исходя из потребностей в продукции установлен перечень предприятий, подлежащих строительству.

1. По каждому из них известны: объемы потребления воды, энергии, потребность в трудовых ресурсах, площадках и т. д.

2. Обоснована система рационального расселения населения, и произведен отбор городов первоочередного размещения промышленных предприятий. По каждому из них определены наборы возможных объемов производства местных ресурсов и соответствующие им затраты.

3. Определен экономический эффект от совместного размещения промышленных предприятий, который дифференцирован в зависимости от отраслей специализации промышленного узла (комплекса), технологической взаимосвязанности предприятий, количества входящих в промузел предприятий.

Требуется найти такой вариант размещения промышленных предприятий, который обеспечивает минимальные суммарные затраты на развитие производства и формирование производственной и социальной инфраструктуры городов при соблюдении соответствующих условий и ограничений (преимущественное развитие региональных центров, соотношение мужского и женского труда).

Математически задача формулируется следующим образом. В m пунктах возможного размещения ($i = \overline{1, m}$) нужно разместить n предприятий ($j = \overline{1, n}$). Естественно, что каждое предприятие может быть размещено только в одном пункте, т. е.

$$\sum_{i=1}^{m} x_{ij} = 1, \quad j = \overline{1, n}. \tag{4.2}$$

Неизвестные x_{ij} в выражении (4.2) могут принимать значение 0 или 1, т. е.

$$x_{ij}=0 \quad \text{или} \quad 1, \quad i=\overline{1, m}, \quad j=\overline{1, n}, \tag{4.3}$$

причем

$$x_{ij}=\begin{cases} 1, & \text{если } j\text{-е предприятие размещается в } i\text{-м пункте,} \\ 0 & \text{в противном случае.} \end{cases}$$

Если a_{jk} — объем потребления k-го ресурса j-м предприятием, то в i-м пункте размещения будет потреблено $\sum\limits_{j=1}^{n} a_{jk}x_{ij}$ единиц k-го ресурса. Так как для каждого i-го пункта размещения заданы наборы возможных объемов производства k-го ресурса

$$\Gamma_{ik}=\left\{a_{ik}^{1},\ a_{ik}^{2},\ \ldots,\ a_{ik}^{l_{ik}}\right\},$$

то ограничения по ресурсам запишутся в следующем виде:

$$\sum_{j=1}^{n} a_{jk}x_{ij} \leqslant y_{ik}, \quad i=\overline{1, m}, \quad k=\overline{1, s}; \tag{4.4}$$

$$y_{ik} \in \Gamma_{ik}, \quad i=\overline{1, m}, \quad k=\overline{1, s}. \tag{4.5}$$

Для любого пункта размещения имеются, кроме того, ограничения по росту населения. Если D_j — число работающих на j-м предприятии, а γ — средняя численность семьи работающего, то эти ограничения запишутся в виде

$$\gamma \sum_{j=1}^{n} D_j x_{ij} \leqslant V_i, \quad i=\overline{1, m}, \tag{4.6}$$

где V_i — максимально допустимый прирост населения в i-м пункте размещения.

При размещении новых предприятий необходимо также следить за тем, чтобы не возникало диспропорции в составе мужского и женского населения в пунктах размещения. Если обозначить через $\underline{\alpha}_i$ и $\bar{\alpha}_i$ соответственно нижнюю и верхнюю границу соотношения мужского и женского труда в i-м пункте, то разумно размещать предприятия так, чтобы выполнялись неравенства

$$\underline{\alpha}_i \leqslant \frac{\sum_{j=1}^{n} (D_j - d_j)\, x_{ij}}{\sum_{j=1}^{n} d_j x_{ij}} \leqslant \bar{\alpha}_i, \quad i = \overline{1, m}, \tag{4.7}$$

где d_j — число женщин, работающих на i-м предприятии.

Естественно, что коэффициенты $\underline{\alpha}_i$, $\bar{\alpha}_i$ рассчитываются с учетом сложившейся в i-м пункте размещения половозрастной структуры населения. Таким образом, ограничения задачи внутрирайонного размещения предприятий задаются условиями (4.2)—(4.7).

Целевая функция должна учитывать приведенные затраты на производство продукции, эффект от агломерации, а также затраты на производство ресурсов и создание социальной инфраструктуры. Если обозначить через c_j себестоимость годового объема производства продукции j-го предприятия, через K_{ij} — сметную стоимость строительства j-го предприятия в i-м пункте, то приведенные затраты выразятся величиной

$$\sum_{i=1}^{m} \sum_{j=1}^{n} (c_j + E_j K_{ij})\, x_{ij},$$

где E_j — нормативный коэффициент эффективности капиталовложений для j-го предприятия.

Экономический эффект от совместного размещения зависит от количества размещаемых предприятий и степени технологической взаимосвязанности предприятий в образуемом ими комплексе. По технологическому признаку предприятия разбиты на группы $J_1, J_2, \ldots$ $\ldots, J_t$, причем $J_p \cap J_k = \varnothing$ $(k \neq p)$. В качестве показателя, характеризующего степень технологической взаимосвязанности предприятий i-го пункта размещения по p-й группе, берется отношение объема валового выпуска предприятий p-й группы к общему валовому выпуску продукции, т. е.

$$\beta_i^p = \sum_{j \in J_p} B_j x_{ij} \Big/ \sum_{j=1}^{n} B_j x_{ij}.$$

Здесь B_j — валовой выпуск продукции j-м предприятием в стоимостном выражении. В качестве показателя, характеризующего степень технологической взаимосвязанности предприятий i-го пункта, берется максимальная из величин β_i^p, а именно число

$$\beta_i^{p_i} = \left(\max_{p=\overline{1,t}} \sum_{j=J_p} B_j x_{ij}\right) \Big/ \sum_{j=1}^{n} B_j x_{ij}.$$

Экономический эффект от агломерации получается за счет снижения себестоимости продукции предприятий, относящихся к p_i-й группе и за счет экономии затрат на строительство вспомогательных и подсобных производств.

Зная степень технологической взаимосвязанности предприятий p_i-й группы $\left(\beta_i^{p_i}\right)$, а также число предприятий p_i-й группы, размещаемых в i-м пункте $\left(\sum_{j \in J_{p_i}} x_{ij}\right)$, можно определить $\delta_i^{p_i} = \delta_i^{p_i}\left(\beta_i^{p_i}, \sum_{j \in J_{p_i}} x_{ij}\right)$ — величину, характеризующую долю снижения себестоимости продукции предприятий p_i-й группы, размещаемых в i-м пункте.

Аналогичным образом, зная степень технологической взаимосвязанности предприятий $\left(\beta_i^{p_i}\right)$ i-го пункта и число размещаемых в этом пункте предприятий $\left(\sum_{j=1}^{n} x_{ij}\right)$ можно определить долю $\rho_i^{p_i} = \rho_i^{p_i}\left(\beta_i^{p_i}, \sum_{j=1}^{n} x_{ij}\right)$ экономии затрат на строительство вспомогательных служб в i-м пункте.

Таким образом, экономия приведенных затрат от совместного размещения предприятий в i-м пункте определяется величиной

$$\psi_i(x_{i1}, x_{i2}, \ldots, x_{in}) = \delta_i^{p_i} \sum_{j \in J_{p_i}} c_j x_{ij} + \rho_i^{p_i} \sum_{j=1}^{n} k_j K_{ij} x_{ij},$$

где k_j — удельный вес затрат капиталовложений на создание вспомогательных и подсобных производств для j-го предприятия.

Приведенные затраты на создание социальной инфраструктуры на v жителей в i-м пункте задаются функцией $\varphi_i(v)$, а приведенные затраты на производстве y_{ik} единиц k-го вида местного ресурса в i-м пункте задаются функцией $f_i(y_{ik})$. Таким образом, целевая функция задачи внутрирайонного размещения предприятий запишется в виде

$$\sum_{i=1}^{m}\left[\sum_{k=1}^{s} f_{ik}(y_{ik}) + \varphi_i\left(\gamma \sum_{j=1}^{n} D_j x_{ij}\right) + \right.$$

$$\left. + \sum_{j=1}^{n} (c_j + E_j K_{ij}) x_{ij} - \psi_i(x_{i1}, x_{i2}, \ldots, x_{in})\right]. \qquad (4.8)$$

Следовательно, математически задача внутрирайонного размещения предприятий состоит в минимизации функции (4.8) при ограничениях (4.2)—(4.7).

§ 5. Задачи специализации литейного производства

Развитие литейного производства, являющегося важнейшей заготовительной базой машиностроения, требует создания специализированных заводов, цехов и участков, обеспечивающих отливками машиностроительные предприятия экономического района или группы экономических районов. Специализация литейного производства позволяет существенно улучшить использование производственных мощностей за счет интенсификации технологических процессов, внедрения высокопроизводительного оборудования, комплексной механизации и автоматизации производства. С увеличением объема производства однородной продукции обеспечивается достижение высоких технико-экономических показателей работы заводов, цехов и участков, в частности, по себестоимости и съему литья с единицы площади. В зависимости от реального состояния дел в литейном производстве возможны различные постановки задачи специализации. В некоторых случаях целесообразно учитывать издержки на транспортировку сырья, и тогда возникает задача нахождения оптимальной специализации литейного производства по критерию совокупных (производственно-транспортных) затрат с учетом имеющихся производственных площадей. В других случаях возникает потребность в решении аналогичной задачи по критерию выпуска продукции, причем специализация производства должна достигаться в первую очередь за счет более полного использования и наращивания мощностей действующих предприятий.

Остановимся на этих задачах более подробно.

Задача 5.1. Снижение себестоимости единицы литья в результате специализации требует, как правило, определенных капиталовложений. Поэтому при обосновании плана специализации необходимо исследовать зависимости удельных капиталовложений от уровня концентрации производства на каждом предприятии по каждому виду отливок.

Специализация литейного производства приводит к расширению кооперированных связей, что увеличивает объем перевозок литья. В связи с этим в большинстве случаев при расчетах экономической эффективности специализации нельзя игнорировать увеличение издержек на транспортировку литья. Себестоимость производства отливок и удельные капиталовложения, с одной стороны, и затраты на транспортировку литья к потребителю, с другой, оказывают противоположное влияние на экономическую эффективность разных вариантов, так как чем выше специализация производства, тем больше радиусы перевозок. Естественно поставить вопрос о выборе такой специализации, при которой достигаются наименьшие суммарные затраты (производственные и транспортные).

Сущность задачи состоит в следующем. В экономическом районе имеются литейные цехи (заводы), выпускающие отливки разных видов. Известно расположение машиностроительных заводов и их потребности в отливках. Определены транспортные расходы на доставку единицы литья от каждого литейного цеха к каждому потребителю. Установлена зависимость производственных затрат от объема литья каждого вида в каждом цехе (предполагается, что в производственные затраты включены удельные капиталовложения с учетом коэффициента эффективности).

Требуется определить оптимальную специализацию цехов и кооперированные связи с тем, чтобы удовлетворить спрос потребителей на отливки необходимых видов. В качестве критерия оптимальности принимается минимум суммарных производственно-транспортных затрат.

Пусть имеются:

m — количество видов литья, отливаемого в цехах экономического района $(i = \overline{1, m})$;

n — число литейных цехов в экономическом районе $(j=\overline{1, n})$;

l — число потребителей литья в экономическом районе $(k=\overline{1, l})$.

Искомые неизвестные обозначим:

x_{ij} — количество литья (т) i-го вида, отливаемое в j-м цехе;

x_{ij}^k — объем перевозок литья (т) i-го вида от j-го цеха к k-му потребителю.

Обозначим известные величины:

d_i^k — потребность k-го пункта в i-м виде литья (т);

a_i — общая потребность экономического района в i-м виде литья (т);

b_j — общий вес (т) всех видов литья, которое отливалось в j-м цехе до специализации;

c_{ij}^k — стоимость (руб.) транспортировки единицы i-го вида литья от j-го цеха к k-му потребителю;

$\varphi_{ij}(x_{ij})$ — функция, выражающая зависимость производственных затрат (руб.) на выпуск литья i-го вида от уровня концентрации в j-м цехе.

В принятых обозначениях задача оптимальной специализации литейного производства по критерию совокупных затрат математически формулируется следующим образом. Необходимо определить такую специализацию цехов x_{ij}, а также такие объемы перевозок x_{ij}^k, при которых целевая функция

$$L=\sum_{i=1}^{m}\sum_{j=1}^{n}\varphi_{ij}(x_{ij})+\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l}c_{ij}^k x_{ij}^k$$

принимает минимальное значение при следующих ограничениях.

1. Потребности в каждом виде литья в экономическом районе должны удовлетворяться:

$$\sum_{j=1}^{n}x_{ij}=a_i, \quad i=\overline{1, m}.$$

2. Общее количество всех видов литья, отливаемого в j-м цехе, должно сохраняться:

$$\sum_{i=1}^{m}x_{ij}=b_j, \quad j=\overline{1, n}.$$

3. Потребность k-го потребителя в i-м виде литья должна удовлетворяться:

$$\sum_{j=1}^{n} x_{ij}^{k} = d_{i}^{k}, \quad i=\overline{1, m}, \quad k=\overline{1, l}.$$

4. Выпуск литья i-го вида в j-м цехе должен быть равен количеству литья этого вида, отправляемого из этого цеха во все пункты потребления:

$$\sum_{k=1}^{l} x_{ij}^{k} = x_{ij}, \quad i=\overline{1, m}, \quad j=\overline{1, n}.$$

5. $x_{ij}^{k} \geqslant 0, \quad i=\overline{1, m}, \quad j=\overline{1, n}, \quad k=\overline{1, l}.$

Очевидно, для разрешимости этой задачи необходимо выполнение равенств

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j, \quad a_i = \sum_{k=1}^{l} d_i^k, \quad i=\overline{1, m}.$$

В результате решения указанной задачи будет получен план специализации, дающий минимум совокупных затрат. Так как этот минимум достигается концентрацией производства однородных отливок, а общее количество выпускаемого литья остается прежним, то за счет увеличения съема литья с единицы производственной площади происходит высвобождение производственной площади. Высвобождаемые площади можно условно считать вновь построенными, а их стоимость нужно добавить к общему экономическому эффекту.

Задача 5.2. Специализацию литейного производства целесообразно проводить в первую очередь за счет наиболее полного использования и наращивания мощностей действующих предприятий. Повышение уровня концентрации производства однородных отливок за счет сокращения номенклатуры выпускаемой продукции в каждом цехе позволяет без больших капиталовложений увеличить выпуск литья в действующих цехах. Поэтому естественно поставить задачу об оптимальном распределении производства отливок между действующими цехами с целью обеспечения максимального выпуска литья с соблюдением его комплектности. Будем предполагать, что удельный вес транспортных расходов в сумме совокупных затрат незначителен.

Сформулируем задачу следующим образом. Требуется распределить заданную программу производства $\{a_1, a_2, \ldots, a_m\}$ между цехами так, чтобы площадь, необходимая для изготовления отливок всех видов, была минимальной, что равносильно требованию максимального высвобождения производственной площади. Высвобождение площади позволяет увеличить выпуск литья сверх заданного плана.

Таким образом, если через $p_{ij}(x_{ij})$ обозначить производственную площадь (м^2), необходимую для выпуска x_{ij} тонн i-го вида литья в j-м цехе, то, с учетом прежних обозначений, задача состоит в достижении минимума занимаемой производственной площади

$$P = \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij}(x_{ij})$$

при следующих ограничениях:

$$\sum_{j=1}^{n} x_{ij} = a_i, \quad i = \overline{1, m},$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = \overline{1, n},$$

$$x_{ij} \geqslant 0, \quad i = \overline{1, m}, \quad j = \overline{1, n}.$$

Изложенные задачи являются задачами нелинейного программирования с непрерывными переменными и, на первый взгляд, не укладываются в рамки дискретных моделей. Однако следует учесть, что функции $\varphi_{ij}(x_{ij})$ и $p_{ij}(x_{ij})$ являются вогнутыми (см. рисунок), а поэтому и функции цели L и P в задачах 5.1 и 5.2 также вогнутые. Поскольку область определения каждой задачи есть выпуклый многогранник, то, как известно, минимум вогнутой функции достигается в вершине этого многогранника. Следовательно, обе задачи можно рассматривать как задачи дискретной оптимизации (на конечном множестве вершин многогранника).

§ 6. Формирование производственной программы предприятия из портфеля заказов

Затраты большинства предприятий на оплату потребляемых материальных ценностей составляют значительную долю всех расходов на производство. Для большинства предприятий стоимость комплектующих изделий, сырья и материалов составляет от 60% до 80% всех затрат на производство.

Очень часто предприятие испытывает дефицит по отдельным видам ресурсов. В этих условиях приходится решать сложные задачи оптимизации по составлению плана предприятия, наилучшим образом использующего имеющиеся материальные ресурсы. В одном случае при определении такого плана ставится цель — получение максимальной прибыли, в другом — максимальное выполнение плана по номенклатуре. От успешного решения этих вопросов во многом зависит ритмичность производства, что в конечном итоге влияет на рентабельность предприятия.

Задача получения оптимального плана с учетом вышеизложенных особенностей состоит в следующем. Предприятие к концу года имеет портфель заказов. Все заказы выполнить невозможно из-за дефицита некоторых видов материалов. При этом известны нормы расхода каждого вида материала на изготовление каждого заказа (изделия). Известны также запасы материалов и прибыль, которую получает предприятие от выполнения каждого заказа. Кроме того, каждому изделию в зависимости от приоритетности его выпуска ставится в соответствие определенный «вес» (например, по десятибалльной системе).

Требуется выбрать из портфеля заказов вариант плана производства на год (или квартал), учитывающий заданную приоритетность различных видов изделий и обеспечивающий, например, максимальную прибыль при соблюдении заданных ограничений по материалам.

Переведем задачу на формальный язык. Итак, имеется номенклатура изделий (портфель заказов) из n наименований ($j = \overline{1, n}$). Каждому изделию (заказу) в зависимости от приоритетности его выпуска ставится в соответствие определенный «вес» v_j. Эти «веса» позволяют соблюдать необходимую очередность выпуска изделий. Известны нормы a_{ij} расхода i-го ($i = \overline{1, m}$) материала на изготовление j-го изделия. Задано наличие материалов i-го вида K_i. Известны затраты труда t_j основных рабочих на изготовление j-го изделия, а также прибыль c_j и суммарная сдельная расценка r_j. Исходные данные приведены в таблице 4.

Таблица 4

Материал	Изделие				Запасы
	1	2	...	n	
1	a_{11}	a_{12}	...	a_{1n}	K_1
2	a_{21}	a_{22}	...	a_{2n}	K_2
...	...	...	...	...	...
m	a_{m1}	a_{m2}	...	a_{mn}	K_m
«Вес»	v_1	v_2	...	v_n	
Прибыль	c_1	c_2	...	c_n	
Затраты труда	t_1	t_2	...	t_n	
Сдельная расценка	r_1	r_2	...	r_n	

Необходимо составить оптимальный план (программу) производства. При составлении такого плана целевая функция может быть задана по-разному в зависимости от конкретной производственной ситуации.

Введем переменные

$$x_j = \begin{cases} 1, & \text{если } j\text{-е } (j=\overline{1, n}) \text{ изделие включается} \\ & \text{в программу,} \\ 0 & \text{в противном случае.} \end{cases}$$

Наиболее типичными являются следующие задачи оптимизации плана предприятия.

Задача 6.1. Выбрать вариант плана производства, учитывающий заданную приоритетность различных видов продукции и обеспечивающий тем самым максимальную «взвешенную» прибыль при соблюдении заданных ограничений по материалам:

$$\sum_{j=1}^{n} v_j c_j x_j \to \max,$$

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant K_i, \qquad i=\overline{1, m},$$

$$x_j = 0 \quad \text{или} \quad 1, \quad j=\overline{1, n}.$$

Задача 6.2. Выбрать вариант плана производства, учитывающий заданную приоритетность различных видов продукции и обеспечивающий максимальную «взвешенную» прибыль при соблюдении заданных ограничений по материалам и расценочному фонду основных рабочих:

$$\sum_{j=1}^{n} v_j c_j x_j \to \max,$$

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant K_i, \quad i = \overline{1, m},$$

$$\sum_{j=1}^{n} r_j x_j \leqslant R,$$

$$x_j = 0 \quad \text{или} \quad 1, \quad j = \overline{1, n}$$

(здесь R — расценочный фонд основных рабочих).

Задача 6.3. Выбрать вариант плана производства, учитывающий заданную приоритетность различных видов продукции и обеспечивающий максимальную «взвешенную» номенклатуру при соблюдении заданных ограничений по материалам:

$$\sum_{j=1}^{n} v_j x_j \to \max,$$

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant K_i, \quad i = \overline{1, m},$$

$$x_j = 0 \quad \text{или} \quad 1, \quad j = \overline{1, n}.$$

Задача 6.4. Выбрать вариант плана производства, учитывающий заданную приоритетность различных видов продукции и обеспечивающий максимальную «взвешенную» номенклатуру при соблюдении заданных ограничений по материалам и расценочному фонду основных рабочих:

$$\sum_{j=1}^{n} v_j x_j \to \max,$$

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant K_i, \quad i = \overline{1, m},$$

$$\sum_{j=1}^{n} r_j x_j \leqslant R,$$

$$x_j = 0 \text{ или } 1, \quad j = \overline{1, n}.$$

З а д а ч а 6.5. Выбрать вариант производства, учитывающий заданную приоритетность различных видов продукции и обеспечивающий максимальную «взвешенную» номенклатуру при соблюдении заданных ограничений по материалам и прибыли:

$$\sum_{j=1}^{n} v_j x_j \to \max,$$

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant K_i, \quad i=\overline{1, m},$$

$$\sum_{j=1}^{n} c_j x_j \geqslant P,$$

$$x_j = 0 \quad \text{или} \quad 1, \quad j=\overline{1, n},$$

где P — запланированная прибыль предприятия.

З а д а ч а 6.6. Выбрать вариант плана производства, учитывающий заданную приоритетность различных видов продукции и обеспечивающий минимальные «взвешенные» затраты труда основных рабочих при соблюдении заданных ограничений по материалам и прибыли:

$$\sum_{j=1}^{n} \frac{1}{v_j} t_j x_j \to \min,$$

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant K_i, \quad i=\overline{1, m},$$

$$\sum_{j=1}^{n} c_j x_j \geqslant P,$$

$$x_j = 0 \quad \text{или} \quad 1, \quad j=\overline{1, n}.$$

Результаты решения перечисленных задач используются:

1. Планово-экономическим отделом — для разработки и обоснования проекта месячного плана завода по объему производства.

2. Производственно-диспетчерским отделом — для составления оперативных графиков выпуска продук-

ции и формирования заделов на всех стадиях производственного процесса.

3. Отделом материально-технического снабжения — для составления графика распределения имеющихся материальных ресурсов между цехами и участками предприятия.

4. Главным экономистом завода — для оценки ожидаемых технико-экономических показателей производственно-хозяйственной деятельности предприятия по объему производства, производительности труда, расходу фонда заработной платы, себестоимости, прибыли, рентабельности.

5. Директором — для принятия решений, относящихся к оперативной ориентации коллектива завода в области выполнения производственного плана.

В заключение этого параграфа рассмотрим еще одну модель формирования плана производства по критерию загрузки оборудования. Если стремиться к максимальной загрузке имеющегося на предприятии оборудования, то задачу формирования производственной программы можно сформулировать следующим образом.

Задача 6.7. Имеется m видов оборудования $(i=\overline{1, m})$. Заданы максимально допустимая $b_i^{\max}$ и соответственно минимально допустимая $b_i^{\min}$ загрузки i-го вида оборудования (в процентах).

Портфель заказов состоит из n изделий $(j=\overline{1, n})$. Каждое j-е изделие характеризуется вектором

$$A_j=(a_{1j},\ a_{2j},\ \ldots,\ a_{mj}),$$

где a_{ij} — процент загрузки i-го вида оборудования j-м изделием. Известна прибыль c_j, которую получает предприятие от выполнения j-го заказа. Запланированная прибыль предприятия составляет величину P.

Требуется из портфеля заказов выбрать такие заказы, которые максимизировали бы загрузку оборудования при соблюдении ограничений по прибыли и имеющемуся в распоряжении предприятия парку оборудования.

Введем переменные

$$x_j=\begin{cases}1, & \text{если } j\text{-е изделие включается в программу;}\\ 0 & \text{в противном случае.}\end{cases}$$

Тогда математически задача состоит в максимизации функции

$$\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}x_j,$$

выражающей собой суммарную загрузку всех видов оборудования в процентах, при соблюдении условий

$$\begin{gathered} b_1^{\min} \leqslant \sum_{j=1}^{n} a_{1j}x_j \leqslant b_1^{\max}, \\ b_2^{\min} \leqslant \sum_{j=1}^{n} a_{2j}x_j \leqslant b_2^{\max}, \\ \cdots\cdots\cdots\cdots \\ b_m^{\min} \leqslant \sum_{j=1}^{n} a_{mj}x_j \leqslant b_m^{\max}, \end{gathered} \tag{6.1}$$

$$\sum_{j=1}^{n} c_j x_j \geqslant P, \tag{6.2}$$

$$x_j = 0 \quad \text{или} \quad 1, \quad j = \overline{1, n}. \tag{6.3}$$

Каждое из неравенств системы (6.1) означает выполнение заданных пределов для процента загрузки каждого вида оборудования. Неравенство (6.2) есть требование соблюдения запланированной прибыли. Альтернатива (6.3), т. е. включение или невключение изделия в план, превращает нашу задачу в задачу дискретной оптимизации.

§ 7. Задачи загрузки оборудования

При планировании работы предприятия (цеха) возникают различные задачи, связанные с рациональным использованием имеющего оборудования. Эти задачи очень разнообразны и зависят от целей, которых необходимо достигнуть на определенных этапах работы предприятия. В качестве таких целей может быть выбрано в одних случаях достижение минимальных затрат на выполнение задания, в других случаях задание необходимо выполнить в кратчайшие сроки.

Задача 7.1. Одной из наиболее распространенных является задача закрепления изделий по видам оборудования с целью достижения минимума затрат на выполнение намеченной программы при соблюдении ограничений на фонды времени работы оборудования. Эта задача состоит в следующем.

Пусть имеется m изделий ($i=\overline{1, m}$), которые могут изготавливаться на n видах оборудования. Каждое i-е изделие должно изготавливаться только на одном виде оборудования, причем должны быть изготовлены все изделия. Известно время t_{ij} изготовления i-го изделия на j-м виде оборудования, а также затраты c_{ij} на изготовление i-го изделия на j-м виде оборудования. Каждый j-й вид оборудования располагает фондом времени b_j. Требуется найти способ изготовления изделий, минимизирующий затраты на производство при соблюдении ограничений на фонд времени работы оборудования.

Введем переменные

$$x_{ij}=\begin{cases}1, & \text{если } i\text{-е изделие изготавливается на } j\text{-м} \\ & \text{виде оборудования,} \\ 0 & \text{в противном случае.}\end{cases}$$

Таким образом, задача состоит в минимизации суммарных затрат

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

при следующих условиях.

1. Каждое изделие должно быть изготовлено:

$$\sum_{j=1}^{n} x_{ij}=1, \quad i=\overline{1, m}.$$

2. Наличный фонд времени работы каждого вида оборудования не должен быть превышен:

$$\sum_{i=1}^{m} t_{ij}x_{ij}\leqslant b_j, \quad j=\overline{1, n}.$$

3. $x_{ij}=0$ или 1, $\quad i=\overline{1, m}, \quad j=\overline{1, n}.$

Задача 7.2. Эта задача отличается от предыдущей тем, что из каждой группы изделий, подлежащих изготовлению, для обработки должно быть выбрано только по одному изделию. Задача может быть сформулирована следующим образом. Имеется m изделий ($i=\overline{1, m}$), которые могут обрабатываться на n видах оборудования ($j=\overline{1, n}$). Все изделия разбиты на k групп $I_s \subset I=\{1, 2, \ldots, m\}$ таким образом, что различные группы не содержат одинаковых изделий, т. е.

$$I_{s_1} \cap I_{s_2}=\varnothing \quad \text{при} \quad s_1 \neq s_2,$$

$$\bigcup_{s=1}^{k} I_s=I.$$

Известны затраты c_i на изготовление i-го изделия, а также величины a_{ij}, характеризующие расход ресурса j-го вида на изготовление i-го изделия. Заданы, кроме того, ограничения b_j по ресурсу j-го вида. Необходимо выбрать для обработки по одному изделию из каждой группы таким образом, чтобы достигался минимум затрат на обработку при соблюдении ограничений по ресурсам.

Введем переменные

$$x_i=\begin{cases} 1, & \text{если } i\text{-е изделие обрабатывается,} \\ 0 & \text{в противном случае.} \end{cases}$$

Тогда математически задача может быть сформулирована как задача минимизации суммарных затрат

$$\sum_{i=1}^{m} c_i x_i$$

при следующих ограничениях.

1. Расход ресурсов не должен превышать их наличия:

$$\sum_{i=1}^{m} a_{ij} x_i \leqslant b_j, \quad j=\overline{1, n}.$$

2. Из каждой группы изделий для обработки должно выбираться только одно:

$$\sum_{i \in I_s} x_i=1, \quad s=\overline{1, k}.$$

3. $$x_i=0 \quad \text{или} \quad 1, \quad i=\overline{1, m}.$$

Задача 7.3. При составлении плана выполнения работ в предыдущих задачах учитывались ограничения по ресурсам за весь период планирования. Рассмотрим более общую задачу, в некоторой мере учитывающую динамику выполнения работ (которая позволяет учитывать ограниченность ресурсов в различные периоды времени) и состоящую в следующем.

Имеются объемы n различных видов работ

$$R=(r_1,\ r_2,\ \dots,\ r_n),$$

которые должны быть выполнены в плановом периоде. Величины r_j определяются исходя из конкретного содержания j-й работы. Рассмотрим процесс выполнения работ R в дискретном времени $t=1,\ 2,\ \dots$ Единицей измерения может быть месяц, сутки или час. Величина Δr_j, показывающая, какую часть работы можно выполнить за единицу времени, есть интенсивность выполнения j-го вида работы. Будем предполагать, что $u_i = r_j/\Delta r_j$ — целое число. При выполнении работ R будут затрачиваться различные виды ограниченных ресурсов. Число различных видов ресурсов есть s $(i=\overline{1,\ s})$. В каждую единицу времени запас всякого ресурса ограничен.

Задача ставится следующим образом: учитывая ограниченность ресурсов и интенсивность выполнения работ, выполнить намеченную программу в кратчайший срок.

Введем обозначения:

b_i^t — фонд ресурсов i-го вида $(i=\overline{1,s})$ в единицу времени с номером t;

d_{ij}^t — количество ресурса i-го вида, которое затрачивается на выполнение объема работ Δr_j в единицу времени с номером t;

$$x_j^t=\begin{cases}1, & \text{если } j\text{-й вид работы выполняется в период} \\ & \text{времени } t, \\ 0 & \text{в противном случае.}\end{cases}$$

В этих обозначениях задача равномерной загрузки оборудования формулируется следующим образом. Требуется найти такую последовательность векторов

$$X(t)=(x_1^t,\ x_2^t,\ \dots,\ x_n^t),\qquad t=\overline{1,\ T},$$

чтобы достигался минимум числа T (намеченная программа должна выполняться в кратчайший срок) при соблюдении следующих условий.

1. Затраты ресурсов каждого i-го вида в любой период времени t не превышают фонда ресурсов:

$$\sum_{j=1}^{n} d_{ij}^t x_j^t \leqslant b_i^t, \quad i=\overline{1, s}, \quad t=\overline{1, T}.$$

2. Намеченная программа должна быть выполнена полностью:

$$\sum_{t=1}^{T} x_j^t = u_j, \quad j=\overline{1, n}.$$

3. $\quad x_j^t = 0 \quad$ или $\quad 1, \quad j=\overline{1, n}, \quad t=\overline{1, T}.$

Иногда по технологическим причинам начатая работа с номером j_0 не может быть прервана до ее завершения. В этом случае вводятся дополнительно следующие ограничения логического вида:

4. $\quad$ Если $x_{j_0}^{t-1} = 1, \quad \sum_{q=1}^{t-1} x_{j_0}^q < u_{j_0}, \quad$ то $\quad x_{j_0}^t = 1.$

§ 8. Задачи стандартизации

Важнейшей чертой современного производства является стремление к комплектации готовых изделий из унифицированных и стандартных блоков. Унификация и стандартизация производства, являющиеся неотъемлемой чертой научно-технического прогресса, требуют решения ряда вопросов по выбору оптимальной номенклатуры выпускаемых изделий (блоков) с учетом их взаимозаменяемости.

В общих терминах задачу оптимальной стандартизации можно сформулировать следующим образом.

Задача 8.1. Известно, что промышленность может выпускать n различных видов деталей. Известны потребности b_j ($j = \overline{1, n}$) в каждом виде деталей. Задана матрица квалификаций $A = \|a_{ij}\|_{m \times n}$, содержащая информацию о возможности замены одних деталей другими: a_{ij} означает, что одна деталь i-го вида ($i \geqslant j$) может заменить a_{ij} штук деталей j-го вида. Если, не нарушая общности, предположить, что деталь с бóльшим номером заменяет несколько деталей с меньшим номером, то матрица квалификаций будет иметь вид

Детали	1	2	3	...	n
1	$a_{11}=1$				
2	a_{21}	$a_{22}=1$			
3	a_{31}	a_{32}	$a_{33}=1$		
...	...	...	...	...	...
n	a_{n1}	a_{n2}	a_{n3}	...	$a_{nn}=1$

Задана функция $f_i(x_i)$ стоимости производства деталей i-го вида $(i=\overline{1, n})$, которая имеет вид

$$f_i(x_i)=\begin{cases} d_i x_i + c_i, & \text{если} \quad x_i > 0, \\ 0, & \text{если} \quad x_i = 0, \end{cases}$$

где d_i, $c_i > 0$. Известно, кроме того, что по требованиям стандартизации можно одновременно выпускать не более s различных видов деталей. Требуется определить, какие виды изделий и в каких количествах (с учетом заменяемости) необходимо произвести, чтобы удовлетворить потребности с минимальными затратами на производство.

Для того чтобы записать экономико-математическую модель нашей задачи, введем переменную x_{ij}, означающую количество деталей i-го вида, производимых для замены деталей j-го вида. В этих обозначениях количество произведенных деталей i-го вида составит величину

$$x_i=\sum_{j=1}^{i} x_{ij}, \quad i=\overline{1, n}.$$

Требование стандартизации, т. е. выпуска одновременно не более s $(s<n)$ различных видов деталей, запишется следующим образом:

$$\sum_{i=1}^{n} \operatorname{sign} x_i \leqslant s,$$

где функция $\operatorname{sign} x_i$ определяется по формуле

$$\operatorname{sign} x=\begin{cases} 1, & \text{если} \quad x>0, \\ 0, & \text{если} \quad x=0. \end{cases}$$

Потребность в деталях j-го вида характеризуется величиной

$$\sum_{i=j}^{n} a_{ij} x_{ij}.$$

Итак, модель задачи оптимальной стандартизации деталей имеет вид

$$\sum_{i=1}^{n} f_i(x_i) \to \min, \tag{8.1}$$

$$\sum_{i=j}^{n} a_{ij} x_{ij} \geqslant b_j, \qquad j = \overline{1, n} \tag{8.2}$$

(т. е. потребность в деталях должна быть удовлетворена);

$$\sum_{i=1}^{n} \operatorname{sign} x_i \leqslant s; \tag{8.3}$$

$$\sum_{j=1}^{i} x_{ij} = x_i, \quad i = \overline{1, n}; \tag{8.4}$$

$$x_{ij} \geqslant 0, \quad i = \overline{1, n}, \quad j = \overline{1, n}. \tag{8.5}$$

Так как функции $f_i(x_i)$ монотонно возрастают, то условия (8.2) можно заменить на

$$\sum_{i=j}^{n} a_{ij} x_{ij} = b_j, \quad j = \overline{1, n}. \tag{8.2'}$$

Задача 8.2. В другой интерпретации задача (8.1)—(8.5) известна под названием *задачи выбора оптимального парка строительных машин.*

Известны объемы b_j $(j = \overline{1, n})$ каждого j-го вида работы в установленную единицу времени (например, год). Задана матрица квалификаций

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix},$$

содержащая информацию о возможности и эффективности выполнения j-го вида работы машиной i-го вида, т. е. a_{ij} — производительность (за год) машины i-го вида на j-м виде работы. Числа a_{ij} неотрицательны, причем $a_{ij}=0$, если машина i-го вида не может выполнить j-й вид работы.

Функция издержек $f_i(x_i)$ на аренду и содержание x_i машин i-го вида имеет тот же вид, что и в задаче 8.1. Требуется определить такой парк машин, который с минимальными издержками выполнит все работы.

Через x_{ij} обозначим количество машин i-го вида, которые будут (в течение года) выполнять работу j-го вида. Тогда количество машин i-го вида, выполняющих разнообразные работы, составит величину

$$x_{i1}+x_{i2}+\ldots+x_{in}=\sum_{j=1}^{n} x_{ij}.$$

Исходные данные и искомые величины задачи представлены в виде таблицы 5.

Таблица 5

Машины	Работы				Количество машин
	1	2	...	n	
1	a_{11} x_{11}	a_{12} x_{12}	...	a_{1n} x_{1n}	$\sum_{j=1}^{n} x_{1j}$
2	a_{21} x_{21}	a_{22} x_{22}	...	a_{2n} x_{2n}	$\sum_{j=1}^{n} x_{2j}$
...	...	...	...	...	...
m	a_{m1} x_{m1}	a_{m2} x_{m2}	...	a_{mn} x_{mn}	$\sum_{j=1}^{n} x_{mj}$
Объемы работ	b_1	b_2	...	b_n	

Производительность всех машин (всех видов), выполняющих работу j-го вида, составит величину

$$a_{1j}x_{1j}+a_{2j}x_{2j}+\ldots+a_{mj}x_{mj}=\sum_{i=1}^{m} a_{ij}x_{ij}.$$

Таким образом, задача выбора оптимального парка машин принимает следующий вид.

1. Минимизировать суммарные издержки на аренду и содержание всех машин:

$$\sum_{i=1}^{m} f_i\left(\sum_{j=1}^{n} x_{ij}\right) \to \min.$$

2. Выполнить объем всех запланированных работ:

$$\sum_{i=1}^{m} a_{ij}x_{ij} = b_j, \quad j = \overline{1, n}.$$

3. $$x_{ij} \geqslant 0, \quad i = \overline{1, m}, \quad j = \overline{1, n}.$$

Обе рассмотренные задачи являются задачами вогнутого программирования при линейных ограничениях. Как уже отмечалось в § 5, минимум целевой функции в этом случае достигается в вершине многогранника условий. И поэтому эти задачи можно рассматривать как задачи дискретной оптимизации. Кроме того, по смыслу задач необходимо, чтобы вершины многогранника были целочисленными. Как будет показано в § 1 гл. V, для этого достаточно, чтобы числа b_j/a_{ij} $(i = \overline{1, m}, j = \overline{1, n})$ были целыми.

§ 9. Распределение капитальных вложений по стройкам и годам планового периода

Повышение эффективности капитальных вложений, сокращение сроков строительства, своевременный ввод в действие производственных мощностей имеют большое значение в практике планирования. При планировании капитальных вложений по стройкам отрасли необходимо учитывать определенные зависимости между выделяемыми по периодам объемами капитальных вложений, количеством строек, нормативной продолжительностью их строительства, объемами строительно-монтажных работ и т. д. Распыление капиталовложений ведет к тому, что в стадии строительства находится гораздо больше строек, чем это целесообразно при выделенных ресурсах. Это приводит к превышению фактических сроков строительства против нормативных, к большим объемам незавершенного строительства. В таких условиях задача оптимального распределения капиталовложений приобретает первостепенное значение. Некоторые из таких задач и рассматриваются в настоящем параграфе.

Задачи 9.1 и 9.2 являются однопродуктовыми задачами распределения капиталовложений по годам планового периода. Они состоят в определении начала строительства каждой стройки с учетом ограничений на капиталовложения, строительно-монтажные работы, задания по вводу производственных мощностей и основных фондов. Различие задач состоит в том, что в первой из них эти характеристики предполагаются заданными в виде вариантов строительства по годам планового периода, а во второй они задаются в соответствии с действующими нормами продолжительности строительства. В качестве критериев оптимальности могут быть выбраны: максимум вводимых за плановый период фондов, максимум количества заканчиваемых в плановом периоде строек, минимум незавершенного строительства. Задача 9.3 является многопродуктовой задачей. В ней предусматривается возможность учета ограничений на ввод мощностей по многим видам продукции. Остановимся подробнее на каждой из них.

Задача 9.1. Задано n строек, для которых в плановый период (T лет) будут выделяться различные объемы капиталовложений. Известны общие лимиты K_t $(t=\overline{1, T})$ на капиталовложения по годам планового периода, лимиты S_t $(t=\overline{1, T})$ на строительно-монтажные работы и задания W_t по вводу мощностей на каждый год $t=\overline{1, T}$.

Пусть для i-й стройки заданы варианты (в количестве m_i штук) распределения капиталовложений по годам планового периода

$$\gamma_i^j=\left(k_{i1}^j,\ k_{i2}^j,\ \ldots,\ k_{iT}^j\right), \quad j=\overline{1, m_i}.$$

Таким образом, в первый год по j-му варианту развития i-й стройке выделяется k_{i1}^j единиц капиталовложений, во второй год — k_{i2}^j и т. д.

Различные варианты выделения капиталовложений для i-й стройки возникают в связи с тем, что стройка начинается либо в первый год, либо во второй и т. д. Каждый вариант γ_i^j определяет соответствующие объемы строительно-монтажных работ по годам планового периода $\left(s_{i1}^j,\ s_{i2}^j,\ \ldots,\ s_{iT}^j\right)$ и вариант ввода мощностей $\left(w_{i1}^j,\ w_{i2}^j,\ \ldots,\ w_{iT}^j\right)$. Величина вводимых фондов при выборе для i-й стройки j-го варианта распределения капиталовложений равна c_{ij}.

Задача заключается в выборе для каждой стройки варианта выделения капиталовложений, при котором максимизируются фонды, вводимые за плановый период.

Введем переменные

$$x_{ij}=\begin{cases}1, & \text{если для } i\text{-й стройки выбирается } j\text{-й вариант,}\\ 0 & \text{в противном случае.}\end{cases}$$

Тогда задача оптимального распределения капиталовложений по критерию вводимых фондов сводится к нахождению максимума функции $\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m_i} c_{ij}x_{ij}$ при выполнении следующих условий.

1. Соблюдение лимитов на капиталовложения по каждому году:

$$\sum_{i=1}^{n}\sum_{j=1}^{m_i} k_{it}^{j}x_{ij}\leqslant K_t,\quad t=\overline{1, T}.$$

2. Соблюдение лимитов на строительно-монтажные работы по каждому году:

$$\sum_{i=1}^{n}\sum_{j=1}^{m_i} s_{it}^{j}x_{ij}\leqslant S_t,\quad t=\overline{1, T}.$$

3. Выполнение заданий по вводу мощностей в каждом году:

$$\sum_{i=1}^{n}\sum_{j=1}^{m_i} w_{it}^{j}x_{ij}\geqslant W_t,\quad t=\overline{1, T}.$$

4. Для каждой стройки должен быть выбран один вариант развития:

$$\sum_{j=1}^{m_i} x_{ij}=1,\quad i=\overline{1, m}.$$

5. $x_{ij}=0$ или 1, $j=\overline{1, m_i}$, $i=\overline{1, n}$.

Иногда вместо максимизации вводимых фондов необходимо максимизировать количество заканчиваемых в плановый период строек. Для решения этой задачи достаточно положить $c_{ij}=1$, если по j-му варианту i-я стройка заканчивается в плановый период (за T лет), и $c_{ij}=0$ в противном случае.

При такой постановке задачи можно ввести еще одно ограничение — выполнение заданий на ввод основных фондов в каждом году:

$$\sum_{i=1}^{n} \sum_{j=1}^{m_i} c_{it}^{j} x_{ij} \geqslant C_t, \qquad t=\overline{1, T},$$

где C_t — задания по вводу основных фондов в t-м году, c_{it}^{j} — нормативное распределение вводимых фондов по i-й стройке.

В другой возможной постановке критерием оптимальности является минимизация незавершенного строительства. В этом случае задача состоит в минимизации функции

$$\sum_{i=1}^{n} \sum_{j=1}^{m_i} d_{ij} x_{ij}$$

при ранее сформулированных ограничениях. Здесь d_{ij} — объем незавершенного строительства к концу планируемого периода при выборе для i-й стройки j-го варианта распределения капиталовложений.

Задача 9.2. Возможна такая постановка задачи распределения капиталовложений, при которой для каждой стройки задаются не варианты распределения капиталовложений по годам планового периода, а лишь нормативная продолжительность строительства и соответствующие показатели по годам.

Тогда задача формулируется следующим образом. Задано n строек. Для i-й стройки известно распределение капиталовложений

$$k_{i1}, k_{i2}, \ldots, k_{i\tau_i}, \qquad i=\overline{1, n},$$

и соответствующие ему объемы строительно-монтажных работ

$$s_{i1}, s_{i2}, \ldots, s_{i\tau_i}, \qquad i=\overline{1, n},$$

по годам строительства. Здесь τ_i — нормативная продолжительность строительства i-й стройки. Этому распределению капиталовложений и строительно-монтажных работ соответствуют определенные объемы ввода производственных мощностей

$$w_{i1}, w_{i2}, \ldots, w_{i\tau_i}, \qquad i=\overline{1, n},$$

и величины вводимых фондов

$$c_{i1}, c_{i2}, \ldots, c_{i\tau_i}, \qquad i=\overline{1, n}.$$

Известны лимиты по годам планового периода на капиталовложения K_t $(t=\overline{1, T})$ и на строительно-монтажные работы S_t $(t=\overline{1, T})$, а также задания по вводу производственных мощностей W_t $(t=\overline{1, T})$ и фондов C_t $(t=\overline{1, T})$.

Задача состоит в максимизации количества заканчиваемых в плановом периоде строек при соблюдении указанных ограничений.

Введем переменные

$$x_{it}=\begin{cases}1, & \text{если } i\text{-я стройка начинается в } t\text{-м году планового периода,}\\ 0 & \text{в противном случае.}\end{cases}$$

Определим множества

$$I_t=\{i\mid\tau_i\leqslant T-t+1\},\quad t=\overline{1, T}.$$

Здесь I_t — множество номеров строек, заканчиваемых за плановый период, если строительство их начато в t-м году. В принятых обозначениях задача оптимального распределения капиталовложений заключается в максимизации функции $\sum\limits_{t=1}^{T}\sum\limits_{i\in I_t}x_{it}$ при следующих ограничениях:

$$\sum_{t=1}^{m}\sum_{i=1}^{n}k_{i,\,m-t+1}x_{it}\leqslant K_m,\qquad m=\overline{1, T};$$

$$\sum_{t=1}^{m}\sum_{i=1}^{n}s_{i,\,m-t+1}x_{it}\leqslant S_m,\qquad m=\overline{1, T};$$

$$\sum_{t=1}^{m}\sum_{i=1}^{n}w_{i,\,m-t+1}x_{it}\geqslant W_m,\qquad m=\overline{1, T};$$

$$\sum_{t=1}^{m}\sum_{i=1}^{n}c_{i,\,m-t+1}x_{it}\geqslant C_m,\qquad m=\overline{1, T};$$

$$\sum_{t=1}^{T}x_{it}\leqslant 1,\qquad i=\overline{1, n};$$

$$x_{it}=0\ \text{ или }\ 1,\quad i=\overline{1, n},\quad t=\overline{1, T}.$$

Приведенные здесь условия, хотя и записанные в несколько иной форме, имеют прежний смысл.

Задача 9.3. Естественным обобщением предыдущих задач является *многопродуктовая задача распределения капиталовложений.* В этом случае объемы ввода производственных мощностей заданы по нескольким видам продукции.

Чтобы сформулировать задачу математически, введем обозначения:

$I = \{\overline{1, n}\}$ — множество индексов строек;

T — количество лет планируемого периода;

T_i — множество лет возможного начала строительства для i-й стройки;

$\bar{I}$ — множество индексов строек, начало строительства которых предусмотрено директивными документами в планируемом периоде;

τ_i — нормативная продолжительность строительства i-й стройки;

k_{iq} — объем капиталовложений по i-й стройке в q-м году строительства $(i \in I, q = \overline{1, \tau_i})$;

s_{iq} — объем строительно-монтажных работ по i-й стройке в q-м году строительства $(i \in I, q = \overline{1, \tau_i})$;

w_{ijq} — объем вводимых мощностей j-го вида по i-й стройке в q-м году строительства $(i \in I, j = \overline{1, r}, q = \overline{1, \tau_i})$;

c_{iq} — объем вводимых основных производственных фондов по i-й стройке в q-м году строительства $(i \in I, q = \overline{1, \tau_i})$;

K_t — лимит на капиталовложения в t-м году планового периода $(t = \overline{1, T})$;

S_t — лимит на строительно-монтажные работы в t-м году планового периода $(t = \overline{1, T})$;

W_{tj} — задание по объему вводимых производственных мощностей j-го вида в t-м году планового периода $(t = \overline{1, T}, j = \overline{1, r})$;

C_t — задание по объему вводимых основных фондов в t-м году планового периода.

Введем переменные

$$x_{it} = \begin{cases} 1, & \text{если } i\text{-я стройка начинается в } t\text{-м году планового периода,} \\ 0 & \text{в противном случае.} \end{cases}$$

Определим множества

$$I_t = \{i \mid \tau_i \leqslant T - t + 1\}, \quad t = \overline{1, T}.$$

В принятых обозначениях задача сводится к максимизации функции

$$\sum_{t=1}^{T} \sum_{i \in I_t} x_{it},$$

представляющей собой количество вводимых строек в течение планового периода, при следующих ограничениях:

$$\sum_{t=1}^{m} \sum_{i \in I} k_{i,\, m-t+1} x_{it} \leqslant K_m, \qquad m = \overline{1, T};$$

$$\sum_{t=1}^{m} \sum_{i \in I} s_{i,\, m-t+1} x_{it} \leqslant S_m, \qquad m = \overline{1, T};$$

$$\sum_{t=1}^{m} \sum_{i \in I} w_{i,\, j,\, m-t+1} x_{it} \geqslant W_{mj}, \qquad m = \overline{1, T}, \quad j = \overline{1, r};$$

$$\sum_{t=1}^{m} \sum_{i \in I} c_{i,\, m-t+1} x_{it} \geqslant C_m, \qquad m = \overline{1, T};$$

$$\sum_{t \in T_i} x_{it} = 1, \quad \forall i \in \bar{I};$$

$$\sum_{t \in T_i} x_{it} \leqslant 1, \quad \forall i \in I;$$

$$x_{it} = 0, \quad \forall i \in I, \quad t \notin T_i;$$

$$x_{it} = 0 \ \text{или} \ 1, \quad \forall i \in I, \quad t = \overline{1, T}.$$

§ 10. Задача надежности

Рассматривая работу технической системы, необходимо учитывать возможность выхода из строя различных узлов этой системы, так как это приводит к частичному, а часто и к полному прекращению работы всей системы. Вместе с тем современное развитие техники, комплексная автоматизация производственных процессов требуют от многих устройств безупречного выполнения их функций в течение всего периода работы. Естественным поэтому является стремление к тому, чтобы устройства, выполняющие особо ответственные функции, были максимально безотказными в работе. В связи с этим весьма актуальной является проблема повышения надежности различных устройств.

Задачу надежности устройства во многих случаях можно формулировать как задачу построения надежного устройства из менее надежных компонент на основе дублирования. Если устройство состоит из некоторого числа последовательных ступеней, то его надежность, т. е. вероятность безотказной работы, можно принять равной произведению вероятностей безотказной работы отдельных ступеней. Увеличение надежности отдельной ступени происходит за счет параллельного размещения компонент-дублеров. При этом приходится учитывать практические ограничения по стоимости, объему, весу, потребляемой мощности и т. д. Задача состоит в конструировании максимально надежного устройства с учетом указанных линейных ограничений.

Введем обозначения:

$1+x_j$ — число компонент-дублеров, используемых в j-й ступени;

$p_j(x_j)$ — вероятность безотказной работы j-й ступени, когда в ней используется x_j+1 компонента;

a_{ij} — i-я характеристика j-й ступени (например, вес, объем, стоимость и т. д.);

b_i — ограничение по i-й характеристике.

В принятых обозначениях задача состоит в максимизации функции

$$P(X)=P(x_1,\ x_2,\ \ldots,\ x_n)=\prod_{j=1}^{n} p_j(x_j)$$

при условиях

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i, \qquad i=\overline{1,\ m},$$

$$x_j \geqslant 0, \quad x_j \text{ — целые числа}, \ j=\overline{1,\ n}.$$

ГЛАВА II

МЕТОД ПОСТРОЕНИЯ ПОСЛЕДОВАТЬЛЬНОСТИ ПЛАНОВ

Для решения изложенных в главе I задач ниже предлагается единый подход, в основе которого лежит идея упорядочения (построения последовательности) планов. Метод, получивший название *метода построения последовательности планов*, впервые был успешно применен в 1965 г. к решению задач размещения производства, а позднее — к решению многих практических задач дискретной оптимизации.

§ 1. Общая схема метода

Пусть задача A состоит в нахождении такого элемента p^* конечного множества P, на котором достигается минимум функции $F(x)$, определенной на этом множестве:

$$F(p^*) = \min_{x \in P} F(x).$$

Элементы множества P, среди которых производится поиск, называются *планами задачи* A, а план p^* — *оптимальным планом задачи* A.

Метод построения последовательности планов применим к задаче A, если выполняются следующие условия:

1) можно найти конечное расширение R множества P ($P \subseteq R$) и функцию $Q(x)$ (миноранту), определенную на R, такую, что $Q(x) \leqslant F(x)$ для всех $x \in P$;

2) можно построить алгоритм φ, который на k-м шаге ($k = 1, 2, \ldots$) находит элемент $r_k \in R$, обладающий свойством

$$Q(r_k) = \min_{x \in R_k} Q(x), \quad k \geqslant 1, \tag{1.1}$$

где $R_k = R_{k-1} \setminus r_{k-1}$ ($k \geqslant 2$), причем $R_1 = R$.

Таким образом, должен существовать алгоритм φ построения последовательности $r_1, r_2, \ldots$ элементов расширения R в порядке неубывания миноранты $Q(x)$.

При обосновании нашего метода важное значение имеет следующий критерий.

Критерий оптимальности. *Если существует такое натуральное число k, что*

$$\begin{gathered} P_k = \{r_1, r_2, \ldots, r_k\} \cap P \neq \varnothing, \\ Q(r_k) \geqslant \min_{x \in P_k} F(x) = F(p_k), \end{gathered} \qquad (1.2)$$

то p_k — оптимальный план задачи A.

Доказательство. Случай, когда $P_k = P$, тривиален. Предположим, что $P_k \neq P$. В силу условия (1.1) для любого $x \in P \setminus P_k$ имеем неравенство $Q(r_k) \leqslant Q(x)$. Отсюда, согласно (1.2) и принимая во внимание, что $Q(x)$ — миноранта функции $F(x)$, получаем

$$F(p_k) \leqslant F(x), \qquad x \in P \setminus P_k.$$

Кроме того, благодаря (1.2) имеем

$$F(p_k) \leqslant F(x) \quad \text{для} \quad x \in P_k.$$

Следовательно, $F(p_k) = \min_{x \in P} F(x)$, что и требовалось доказать.

Из доказанного следует, что для решения задачи A необходимо в процессе работы алгоритма упорядочения φ следить за выполнением критерия оптимальности (1.2). В этом и заключается сущность метода ψ построения последовательности планов, k-й шаг которого состоит в том, что с помощью алгоритма φ находится элемент r_k, а также множество P_k. Если $P_k = \varnothing$, то переходим к $(k+1)$-му шагу. Если $P_k \neq \varnothing$, то проверяем, выполняется ли условие (1.2). Если да, то процесс обрывается и p_k — оптимальный план задачи A. Если нет, то переходим к следующему шагу.

Таким образом, решение задачи A по методу построения последовательности планов приводит к следующей схеме расчета.

1. Конструируется конечное расширение R исходного множества P.

2. На множестве R задается функция $Q(x)$, являющаяся минорантой функции $F(x)$ на множестве P.

3. Формируется итеративный (но конечный) алгоритм φ, который на каждом шаге с номером k определяет оптимальный план r_k задачи минимизации функции $Q(x)$ на множестве R_k, образованном из расширения R путем исключения предыдущих оптимумов $r_1, r_2, \ldots, r_{k-1}$. Таким образом, алгоритм φ определяет последовательность оптимумов $r_1, r_2, \ldots$ миноранты $Q(x)$ на последовательности вложенных друг в друга множеств

$$R = R_1 \supset R_2 \supset R_3 \supset \ldots$$

4. На каждом k-м шаге метода ψ проверяется выполнение критерия оптимальности. Если критерий оптимальности выполняется, то процесс обрывается и p_k есть оптимальный план исходной задачи A. В противном случае переходим к следующему, $(k+1)$-му, шагу метода ψ.

В последующих параграфах при рассмотрении конкретных задач оптимизации мы будем показывать, как целесообразно строить расширение R, миноранту $Q(x)$, а также последовательность элементов $r_1, r_2, \ldots$ множества R, удовлетворяющую условию (1.1). Подробного описания алгоритма решения исходной задачи, как правило, приводить не будем. Как мы уже условились, исходную задачу будем называть задачей A, а вспомогательную задачу минимизации миноранты $Q(x)$ на множестве R — задачей $\bar{A}$.

Эффективность изложенного метода зависит от выбора расширения R и миноранты $Q(x)$. В простейшем случае миноранта $Q(x)$ может быть задана сразу на всем множестве R до начала работы метода. Однако если в процессе работы есть возможность уточнять миноранту на каждом шаге, т. е. находить функцию $Q_k(x)$, удовлетворяющую условиям

$$Q_1(x) \leqslant Q_{k-1}(x) \leqslant Q_k(x), \quad x \in R_k,$$
$$Q_k(x) \leqslant F(x), \quad x \in R_k \cap P,$$

то критерий оптимальности может сработать раньше.

Важной особенностью метода построения последовательности планов является то, что он применим не только для точного, но и для приближенного решения задачи A. Это становится очевидным при рассмотрении следующей теоремы.

Теорема. *Пусть на k-м шаге работы метода ψ $P_k \neq \varnothing$, а условие* (1.2) *не выполняется. Тогда*

$$Q(r_k) \leqslant F(p^*) \leqslant F(p_k).$$

Доказательство. Неравенство $F(p^*) \leqslant F(p_k)$ очевидно, так как p^* — оптимальный план задачи A. Докажем, что $Q(r_k) \leqslant F(p^*)$. Рассмотрим два возможных случая.

1) $p_k = p^*$, т. е. p_k — оптимальный план задачи A. Тогда по условию теоремы

$$F(p^*) = F(p_k) > Q(r_k).$$

2) $p_k \neq p^*$; значит, $p^* \in P \setminus P_k$. Отсюда по условию (1.1) получаем

$$F(p^*) \geqslant Q(p^*) \geqslant Q(r_k).$$

Теорема доказана.

В качестве следствия из этой теоремы получаем, что числа $Q(r_k)$ и $F(p_k)$ являются границами минимального значения целевой функции задачи A. При этом с каждым шагом, т. е. с возрастанием числа k, нижняя граница $Q(r_k)$ не убывает, а верхняя $F(p_k)$ не возрастает. Иначе говоря, в процессе работы метода построения последовательности планов можно оценить максимальное отклонение лучшего из полученных элементов от оптимального, так как справедливо неравенство

$$F(p_k) - F(p^*) \leqslant F(p_k) - Q(r_k).$$

Это позволяет остановить процесс поиска оптимального плана, как только достигнута заданная точность расчетов, не дожидаясь, пока сработает критерий оптимальности.

До сих пор мы рассматривали применение метода построения последовательности планов для решения задач дискретной оптимизации, заключающихся в минимизации функционала на конечном множестве. Ясно, что подобная схема может быть сконструирована и для задачи A максимизации функции $F(x)$ на конечном множестве P.

Метод построения последовательности планов применим к решению этой задачи, если

1) можно найти конечное расширение R множества P ($R \supseteq P$) и функцию $Q(x)$, определенную на R

и являющуюся мажорантой функции $F(x)$, т. е. $Q(x) \geqslant F(x)$ для $x \in P$;

2) можно построить алгоритм φ, который на k-м шаге ($k = 1, 2, \ldots$) находит элемент $r_k \in R$, обладающий свойством

$$Q(r_k) = \max_{x \in R_k} Q(x), \quad k \geqslant 1, \tag{1.3}$$

где $R_1 = R$, $R_k = R_{k-1} \setminus r_{k-1}$ ($k = 2, 3, \ldots$).

В этом случае критерий оптимальности можно сформулировать следующим образом.

Критерий оптимальности. *Если существует такое натуральное число k, что*

$$\begin{gathered} P_k = \{r_1, r_2, \ldots, r_k\} \cap P \neq \varnothing, \\ Q(r_k) \leqslant \max_{x \in P_k} F(x) = F(p_k), \end{gathered} \tag{1.4}$$

то p_k — оптимальный план задачи A.

Доказательство того, что элемент последовательности, построенный с помощью алгоритма φ и удовлетворяющий условиям (1.4), является оптимальным планом задачи A, аналогично приведенному выше доказательству для задачи минимизации.

Модификация общей схемы. Сущность этой модификации состоит в построении последовательности элементов некоторого расширения исходного множества с пропуском заведомо неоптимальных планов.

Новую модификацию ψ_0 нашей схемы изложим на примере решения задачи A минимизации функции $F(x)$ на конечном множестве P.

Метод ψ_0 применим к задаче A, если можно построить метод φ_0, который на k-м шаге находит элемент $r_k \in R$, обладающий свойством

$$Q(r_k) = \min_{x \in R_k} Q(x), \qquad k \geqslant 1,$$

где

$R_1 = R$ — конечное или бесконечное расширение множества P ($R \supseteq P$);

$Q(x)$ — функция, определенная на R и являющаяся минорантой функции $F(x)$ на P, т. е. $Q(x) \leqslant F(x)$ для всех $x \in P$;

$$R_k = R_{k-1} \setminus R(r_{k-1}), \quad r_{k-1} \in R(r_{k-1}) \subset R \quad (k \geqslant 2);$$
причем если $r_{k-1} \in P$, то

$$F(r_{k-1}) = \min_{x \in R(r_{k-1}) \cap P} F(x),$$

а если $r_{k-1} \notin P$, то $R(r_{k-1}) \cap P = \varnothing$.

Очевидно, что для последовательности $r_1, r_2, \ldots$, которая строится с помощью метода φ_0, справедлив критерий оптимальности. Поэтому описание k-го шага метода ψ_0 решения задачи A отличается от описания k-го шага метода ψ тем, что символ φ следует заменить на φ_0. Таким образом, метод ψ_0 отличается от ψ тем, что теперь на каждом шаге от множества R отсекается не один элемент, а несколько (целое множество).

§ 2. Схема упорядочения планов

Основой метода построения последовательности планов является процедура φ упорядочения планов задачи $\bar{A}$, состоящей в минимизации миноранты $Q(x)$ на множестве R. Как правило, подобные процедуры состоят в нахождении и элиминации (исключении) элементов множества R. Возможны различные способы элиминации элементов множества R. Обратимся к изучению одного из них, основанного на ветвлении множества R на дерево подмножеств.

Элиминация. Пусть $\{R_1, R_2, \ldots, R_t\}$ — разбиение некоторого подмножества множества R:

$$\bigcup_{i=1}^{t} R_i \subset R,$$

$$R_i \cap R_j = \varnothing, \quad i \neq j, \quad i, j = \overline{1, t},$$

и пусть $r_0 \in \bigcup_{i=1}^{t} R_i$. Элиминацию элемента r_0 из разбиения $\{R_1, R_2, \ldots, R_t\}$ определим с помощью описанного ниже алгоритма ξ. В частном случае, когда $t = 1$, получаем понятие элиминации r_0 из множества R_1. Результат элиминации r_0 из $\{R_1, R_2, \ldots, R_t\}$ будем обозначать через $\{R_1, R_2, \ldots, R_t\}_{r_0}$.

Пусть $R_i^1 = R_i \ (i = \overline{1, t_1}, \ t_1 = t)$.

Алгоритм ξ. k-й шаг ($k \geqslant 1$). Среди множеств $R_1^k, R_2^k, \ldots, R_{t_k}^k$ находим множество $R_{\nu(k)}^k$, содержащее элемент r_0. Если $R_{\nu(k)}^k \setminus r_0 \neq \varnothing$, то по некоторому правилу разбиваем $R_{\nu(k)}^k$ на непересекающиеся непустые подмножества:

$$R_{\nu(k)}^k = \bigcup_{i=1}^{\mu(k)} R_{\nu(k),\, i}^k, \quad \mu(k) \geqslant 2.$$

Не подвергшиеся разбиению и вновь полученные множества, т. е.

$$R_1^k, R_2^k, \ldots, R_{\nu(k)-1}^k, R_{\nu(k)+1}^k, \ldots, R_{t_k}^k,$$

$$R_{\nu(k),\,1}^k, R_{\nu(k),\,2}^k, \ldots, R_{\nu(k),\,\mu(k)}^k,$$

заново обозначаем через $R_1^{k+1}, R_2^{k+1}, \ldots, R_{t_{k+1}}^k$ и переходим к $(k+1)$-му шагу. Если $R_{\nu(k)}^k = \{r_0\}$, то

$$\{R_1, R_2, \ldots, R_t\}_{r_0} = \\ = \left\{R_1^k, R_2^k, \ldots, R_{\nu(k)-1}^k, R_{\nu(k)+1}^k, \ldots, R_{t_k}^{k+1}\right\},$$

и алгоритм ξ закончен.

Алгоритм ξ есть не что иное, как обычная схема ветвления, на каждом шаге которой разбивается множество, содержащее элемент r_0. Лишь на последнем шаге, когда появляется множество, состоящее из одного элемента r_0, оно удаляется из разбиения. Таким образом, результат элиминации элемента r_0 из разбиения $\{R_1, R_2, \ldots, R_t\}$ есть разбиение множества $\bigcup_{i=1}^{t} R_i \setminus r_0$ на непересекающиеся подмножества $R_1', R_2', \ldots, R_{t'}'$.

Построение последовательности. Будем предполагать, что мы владеем алгоритмом ξ элиминации и умеем находить минимум функции $Q(x)$ на подмножествах, получающихся в результате элиминации. Тогда построение последовательности $r_1, r_2, \ldots$ планов задачи A будем осуществлять по следующей схеме.

Пусть $R_1^1 = R$ $(t_1 = 1)$.

Алгоритм φ. k-й шаг ($k \geqslant 1$). Если $\bigcup\limits_{i=1}^{t_k} R_i^k \neq \varnothing$, то находим $\min\limits_{1 \leqslant i \leqslant t_k} \min\limits_{x \in R_i^k} Q(x) = Q(r_k)$. Элиминируем с помощью алгоритма ξ элемент r_k из разбиения $\{R_1^k, R_2^k, \ldots, R_{t_k}^k\}$, получаем

$$\{R_1^k, R_2^k, \ldots, R_{t_k}^k\}_{r_k} = \{R_1^{k+1}, R_2^{k+1}, \ldots, R_{t_{k+1}}^{k+1}\}$$

и переходим к $(k+1)$-му шагу. Если $\bigcup\limits_{i=1}^{t_k} R_i^k = \varnothing$, то последовательность планов задачи $\bar{A}$ построена.

Так как на каждом k-м шаге работы алгоритма φ происходит элиминация из разбиения $\{R_1^k, R_2^k, \ldots, R_{t_k}^k\}$ элемента r_k такого, что

$$Q(r_k) = \min_{x \in \bigcup\limits_{i=1}^{t_k} R_i^k} Q(x),$$

то алгоритм φ строит нужную последовательность $r_1, r_2, \ldots$ планов задачи $\bar{A}$, т. е. последовательность, обладающую свойством

$$Q(r_1) = \min_{x \in R} Q(x),$$

$$Q(r_k) = \min_{x \in R \setminus \{r_1, r_2, \ldots, r_{k-1}\}} Q(x), \quad k = 2, 3, \ldots$$

Построение последовательности $r_1, r_2, \ldots$ можно осуществлять, оперируя не с подмножествами, а лишь с их наилучшими представителями.

В последующих параграфах именно такое описание алгоритмов и будет принято.

§ 3. Упорядочение планов одной вспомогательной задачи

Как отмечалось в § 1, метод построения последовательности планов применим для решения задачи A, если можно сконструировать алгоритм φ, который позволяет эффективно упорядочивать планы вспомогательной задачи $\bar{A}$. В настоящем параграфе приводятся алгоритмы упорядочения планов часто встречающейся в дальнейшем изложении вспомогательной задачи.

Пусть задача $\bar{A}$ состоит в минимизации сепарабельной функции

$$Q(x_1, x_2, \ldots, x_n) = g_1(x_1)\,\omega g_2(x_2)\,\omega \ldots \omega g_n(x_n),$$

заданной на множестве R векторов $X = (x_1, x_2, \ldots, x_n)$, удовлетворяющих условиям

$$\sum_{i=1}^{n} x_i = b,$$

$$x_i \in \Gamma_i = \{a_{i1}, a_{i2}, \ldots, a_{is_i}\}, \quad i = \overline{1, n},$$

где $b > 0$, $a_{ij} \geqslant 0$ — целые числа. Здесь ω — операция сложения или умножения.

Для решения этой задачи можно применить эффективную процедуру динамического программирования. Для этого надо воспользоваться следующим функциональным уравнением:

$$f_k(z) = \min_{x_k \in \Gamma_k} [g_k(x_k)\,\omega f_{k-1}(z - x_k)], \quad k = \overline{2, n},$$

где $0 \leqslant z \leqslant b$, причем $f_1(z) = g_1(z)$ для $0 \leqslant z \leqslant b$. Здесь $f_k(z)$ — оптимальное значение функционала задачи $\bar{A}$, в которой индекс n и параметр b заменены соответственно на k и z.

Прежде чем перейти к изложению алгоритма построения последовательности планов задачи $\bar{A}$, введем некоторые необходимые в дальнейшем определения и обозначения.

План $\bar{X} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ задачи $\bar{A}$ назовем *h-оптимальным*, если

$$Q(\bar{X}) = g_{h+1}(\bar{x}_{h+1})\,\omega \ldots \omega g_n(\bar{x}_n)\,\omega f_h\left(b - \sum_{i=h+1}^{n} \bar{x}_i\right),$$

и будем обозначать его

$$(*, *, \ldots, *, \bar{x}_{h+1}, \ldots, \bar{x}_n).$$

Таким образом, $\bar{X} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ — h-оптимальный план задачи $\bar{A}$, если $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_h)$ — оптимальный план задачи минимизации функции

$$g_1(x_1)\,\omega g_2(x_2)\,\omega \ldots \omega g_h(x_h)$$

при условиях

$$\sum_{i=1}^{h} x_i = b - \sum_{i=h+1}^{n} \bar{x}_i,$$

$$x_i \in \Gamma_i, \quad i = \overline{1, h}.$$

Непосредственно из определения h-оптимального плана следует, что

$$Q(*, *, \ldots, *, \bar{x}_{h+1}, \ldots, \bar{x}_n) = \min_{X \in R(*, \ldots, *, \bar{x}_{h+1}, \ldots, \bar{x}_n)} Q(X), \quad (*)$$

где

$$R(*, \ldots, *, \bar{x}_{h+1}, \ldots, \bar{x}_n) =$$

$$= \{X \mid X \in R, \quad x_i = \bar{x}_i, \quad i = \overline{h+1, n}\},$$

т. е. $R(*, \ldots, *, \bar{x}_{h+1}, \ldots, \bar{x}_n)$ — множество всех тех планов задачи $\bar{A}$, у которых $x_i = \bar{x}_i$ $(i = \overline{h+1, n})$.

Теперь можно сформулировать процедуру построения последовательности $r_1, r_2, \ldots$ планов задачи $\bar{A}$ следующим образом.

Алгоритм ф. 1-й шаг. Среди множества W_1 всевозможных $(n-1)$-оптимальных планов

$$(*, *, \ldots, *, a_{n1}),$$
$$(*, *, \ldots, *, a_{n2}),$$
$$\ldots\ldots\ldots$$
$$(*, *, \ldots, *, a_{ns_n})$$

находим такой план

$$r_1 = (*, \ldots, *, x_n^1) = (x_1^1, x_2^1, \ldots, x_n^1),$$

что

$$Q(r_1) = \min_{X \in W_1} Q(X).$$

k-й шаг $(k = 2, 3, \ldots)$. Множество W_{k-1} преобразуем во множество W_k по следующему правилу. Оставляем без изменения все планы множества W_{k-1}, кроме плана

$$r_{k-1} = (*, \ldots, *, x_p^{k-1}, \ldots, x_n^{k-1}) =$$

$$= (x_1^{k-1}, x_2^{k-1}, \ldots, x_n^{k-1}),$$

который заменяем множеством $O(r_{k-1})$, состоящим из следующих планов:

$$\left(*,\ *,\ \ldots,\ *,\ \tilde{x}_{q-1},\ x_q^{k-1},\ \ldots,\ x_p^{k-1},\ \ldots,\ x_n^{k-1}\right)$$

$$\forall x_{q-1} \in \left\{x_{q-1} \,\middle|\, x_{q-1} \in \Gamma_{q-1},\ x_{q-1} \neq x_{q-1}^{k-1}\right\},\quad q = \overline{2,\ p}.$$

Далее, находим такой план r_k, что

$$Q(r_k) = \min_{X \in W_k} Q(X),$$

и переходим к следующему шагу.

Приступим к обоснованию того, что алгоритм φ строит требуемую последовательность планов задачи $\bar{A}$. В процессе работы алгоритма φ множество W_k получается из множества W_{k-1} путем замены элемента r_{k-1} множеством $O(r_{k-1})$, т. е.

$$W_k = (W_{k-1} \setminus r_{k-1}) \cup O(r_{k-1}).$$

При этом, как видно из построения, множество $O(r_{k-1})$ обладает свойством

$$\bigcup_{X \in O(r_{k-1})} R(X) = R(r_{k-1}) \setminus r_{k-1}.$$

Отсюда, вводя обозначение $R_k = \bigcup\limits_{X \in W_k} R(X)$, получаем

$$R_k = \left\{\bigcup_{X \in W_{k-1} \setminus r_{k-1}} R(X)\right\} \cup \left\{\bigcup_{X \in O(r_{k-1})} R(X)\right\} =$$

$$= \left\{\bigcup_{X \in W_{k-1} \setminus r_{k-1}} R(X)\right\} \cup \{R(r_{k-1}) \setminus r_{k-1}\} = R_{k-1} \setminus r_{k-1}.$$

Таким образом, на k-м шаге алгоритма φ из множества R_{k-1} элиминируется элемент r_{k-1}, т. е.

$$R_1 = R,\quad R_k = R \setminus \{r_1,\ r_2,\ \ldots,\ r_{k-1}\},\quad k = 2, 3, \ldots$$

В силу (*) справедливо равенство $Q(X) = \min\limits_{\bar{X} \in R(X)} Q(\bar{X})$ для всякого $X \in W_k$. Следовательно, имеем

$$Q(r_k) = \min_{X \in W_k} Q(X) = \min_{X \in W_k} \min_{\bar{X} \in R(X)} Q(\bar{X}) = \min_{\bar{X} \in R_k} Q(\bar{X}).$$

Тем самым доказано, что в процессе работы алгоритма φ строится такая последовательность планов $r_1, r_2, \ldots$, что

$$Q(r_k) = \min_{\bar{X} \in R_k} Q(\bar{X}),\quad k = 1, 2, \ldots,$$

т. е. алгоритм φ строит требуемую последовательность планов.

Проиллюстрируем работу алгоритма на примере.

Пример. Необходимо минимизировать функцию

$$Q(X)=g_1(x_1)+g_2(x_2)+g_3(x_3)$$

при условиях

$$x_1+x_2+x_3=5,$$
$$x_1, x_2, x_3 \in \{0, 1, 2, 3\}.$$

Функции $g_1(x_1)$, $g_2(x_2)$, $g_3(x_3)$ заданы таблицей 1.

Таблица 1

x	0	1	2	3
$g_1(x_1)$	0	150	280	400
$g_2(x_2)$	0	140	290	410
$g_3(x_3)$	0	130	270	420

Решим задачу при помощи функциональных уравнений динамического программирования. Процесс решения представим в виде подробных таблиц, которые нам в дальнейшем понадобятся при построении последовательности планов. В таблице 2 приведены значе-

Таблица 2

z	0	1	2	3	4	5
$f_1(z)$	0	150	280	400	—	—

ния функции $f_1(z)=g_1(z)$. В этой и последующих таблицах символ «—» означает, что функция для соответствующего значения не определена. В таблице 3

Таблица 3

z / x_2	0	1	2	3	4	5
0	0	150	280	400	—	—
1	—	140	290	420	540	—
2	—	—	290	440	570	690
3	—	—	—	410	560	690
$f_2(z)$	0	140	280	400	540	690

Таблица 4

z / x_3	0	1	2	3	4	5
0	0	140	280	400	540	690
1	—	130	270	410	530	670
2	—	—	270	410	550	670
3	—	—	—	420	560	700
$f_3(z)$	0	130	270	400	530	670

приведены значения функции $g_2(x_2)+f_1(z-x_2)$, а в последней строке — значения функции

$$f_2(z)=\min_{x_2\in\{0,\ 1,\ 2,\ 3\}}[g_2(x_2)+f_1(z-x_2)].$$

В таблице 4 приведены значения функции $g_3(x_3)+$ $+f_2(z-x_3)$, а в последней строке — значения функции

$$f_3(z)=\min_{x_3\in\{0,\ 1,\ 2,\ 3\}}[g_3(x_3)+f_2(z-x_3)].$$

1-й шаг. Строим множество W_1 всех 2-оптимальных планов. В нашем случае

$$W_1=\{B_1^1,\ B_2^1,\ B_3^1,\ B_4^1\},$$

$$B_1^1=(*,\ *,\ 0)=(3,\ 2,\ 0),\quad B_2^1=(*,\ *,\ 1)=(3,\ 1,\ 1),$$

$$B_3^1=(*,\ *,\ 2)=(3,\ 0,\ 2),\quad B_4^1=(*,\ *,\ 3)=(2,\ 0,\ 3),$$

$$Q(B_1^1)=690,\quad Q(B_2^1)=670,$$

$$Q(B_3^1)=670,\quad Q(B_4^1)=700.$$

Элементы множества W_1 перенумерованы так, что верхний индекс означает номер шага, а нижний указывает на порядковый номер элемента в множестве. Такого же правила нумерации планов мы будем придерживаться и в последующих шагах алгоритма.

Среди элементов множества W_1 находим план с минимальным значением целевой функции. В нашем случае $r_1 = B_2^1$ и $Q(r_1) = Q(B_2^1) = 670$. Переходим ко 2-му шагу.

2-й шаг. Преобразуем множество W_1 во множество W_2. Для этого план $r_1 = B_2^1$ заменяем следующей системой планов:

$$O(r_1) = \{(*, 2, 1), \quad (*, 3, 1)\}.$$

Таким образом,

$$W_2 = \{B_1^2, B_2^2, B_3^2, B_4^2, B_5^2\},$$

$$B_1^2 = B_1^1 = (*, *, 0) = (3, 2, 0), \quad B_2^2 = B_3^1 = (*, *, 2) = (3, 0, 2),$$

$$B_3^2 = B_4^1 = (*, *, 3) = (2, 0, 3), \quad B_4^2 = (*, 2, 1) = (2, 2, 1),$$

$$B_5^2 = (*, 3, 1) = (1, 3, 1), \quad Q(B_1^2) = 690, \quad Q(B_2^2) = 670,$$

$$Q(B_3^2) = 700, \quad Q(B_4^2) = 700, \quad Q(B_5^2) = 690.$$

Находим план $r_2 = B_2^2$ и $Q(r_2) = Q(B_2^2) = 670$. Переходим к 3-му шагу.

3-й шаг. Строим множество W_3. Для этого план $r_2 = B_2^2$ заменяем следующей системой планов:

$$O(r_2) = \{(*, 1, 2), \ (*, 2, 2), \ (*, 3, 2)\}.$$

Получаем:

$$W_3 = \{B_1^3, B_2^3, B_3^3, B_4^3, B_5^3, B_6^3, B_7^3\},$$

$$B_1^3 = B_1^2 = (*, *, 0) = (3, 2, 0), \quad B_2^3 = B_3^2 = (*, *, 3) = (2, 0, 3),$$

$$B_3^3 = B_4^2 = (*, 2, 1) = (2, 2, 1), \quad B_4^3 = B_5^2 = (*, 3, 1) = (1, 3, 1),$$

$$B_5^3 = (*, 1, 2) = (2, 1, 2), \qquad B_6^3 = (*, 2, 2) = (1, 2, 2),$$

$$B_7^3 = (*, 3, 2) = (0, 3, 2),$$

$$Q(B_1^3) = 690, \quad Q(B_2^3) = 700,$$

$$Q(B_3^3) = 700, \quad Q(B_4^3) = 690,$$

$$Q(B_5^3) = 690, \quad Q(B_6^3) = 690, \quad Q(B_7^3) = 680.$$

Находим

$$r_3 = B_7^3, \quad Q(r_3) = Q(B_7^3) = 680.$$

Итак, мы построили три лучших плана исходной задачи

$$r_1 = (3, 1, 1), \quad r_2 = (3, 0, 2), \quad r_3 = (0, 3, 2)$$

со значениями целевой функции

$$Q(r_1) = 670, \quad Q(r_2) = 670, \quad Q(r_3) = 680.$$

Если работу алгоритма продолжить, то будут построены все планы исходной задачи в порядке неубывания целевой функции. Все они приведены в таблице 5.

Таблица 5

r_i \ x_i	x_1	x_2	x_3	$Q(r_i)$
r_1	3	1	1	670
r_2	3	0	2	670
r_3	0	3	2	680
r_4	3	2	0	690
r_5	2	3	0	690
r_6	1	3	1	690
r_7	2	1	2	690
r_8	1	2	2	690
r_9	2	2	1	700
r_{10}	2	0	3	700
r_{11}	1	1	3	710
r_{12}	0	2	3	710

Замечание. Аналогичный алгоритм легко сформулировать и для случая, когда в условиях задачи $\bar{A}$ равенство $\sum_{i=1}^{n} x_i = b$ заменено неравенством $\sum_{i=1}^{n} x_i \geqslant b$. Тогда изменяется лишь функциональное уравнение, оно принимает следующий вид:

$$f_k(z) = \min_{x_k \in \Gamma_k(z)} [g_k(x_k) \omega f_{k-1}(\max(0, z - x_k))], \quad k = \overline{2, n},$$

где

$$\Gamma_k(z)=\left\{x_k \mid x_k\in\Gamma_k,\quad z-x_k\leqslant\sum_{i=k+1}^{n}\max_{1\leqslant j\leqslant s_i} a_{ij}\right\},$$
$$0\leqslant z\leqslant b,$$

причем

$$f_1(z)=\min_{\substack{x_1\geqslant z\\ x_1\in\Gamma_1}} g_1(x_1).$$

§ 4. Построение последовательности систем представителей

Пусть заданы m множеств $J_i=\{1, 2, \ldots, s_i\}$ $(i=\overline{1, m})$. Последовательность чисел $r=\{j_1, j_2, \ldots, j_m\}$ такую, что $j_i\in J_i$ $(i=\overline{1, m})$, будем называть *системой представителей* (*с. п.*) множеств J_i $(i=\overline{1, m})$. Множество всех с. п. будем обозначать через W. При решении многих задач дискретной оптимизации методом построения последовательности планов возникает задача построения последовательности с. п. в порядке неубывания некоторой функции. В частности, в дальнейшем нам понадобится строить последовательность с. п., когда функция определена следующим образом.

Пусть заданы множества чисел

$$\Gamma_i=\{c_{i1}, c_{i2}, \ldots, c_{is_i}\},\quad i=\overline{1, m}.$$

На множестве W определим функцию

$$Q(r)=\sum_{i=1}^{m} c_{ij_i},\quad r=\{j_1, \ldots, j_m\}\in W.$$

Рассмотрим перестановки

$$\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^{s_i}\quad (i=\overline{1, m})$$

чисел множеств J_i $(i=\overline{1, m})$ такие, что

$$c_{i\alpha_i^1}\leqslant c_{i\alpha_i^2}\leqslant\ldots\leqslant c_{i\alpha_i^{s_i}}.$$

Очевидно, с. п. $r_1=(\alpha_1^1, \alpha_2^1, \ldots, \alpha_m^1)$ является оптимальной системой задачи минимизации функции $Q(r)$ на множестве W, т. е.

$$Q(r_1)=\min_{r\in W} Q(r).$$

Пусть $r = (j_1, \ldots, j_{p-1}, \alpha_p^q, \alpha_{p+1}^1, \ldots, \alpha_m^1) \in W$, где $1 \leqslant q \leqslant s_p$, $1 \leqslant p \leqslant m$. Определим $O(r)$ следующим образом:

$$O(r) = \begin{cases} \bigcup\limits_{i=0}^{m-p} r^{(i)}, & \text{если} \quad q < s_p; \\ \bigcup\limits_{i=1}^{m-p} r^{(i)}, & \text{если} \quad q = s_p,\ p < m; \\ \varnothing, & \text{если} \quad q = s_p,\ p = m, \end{cases} \tag{4.1}$$

где

$$r^{(0)} = (j_1, \ldots, j_{p-1}, \alpha_p^{q+1}, \alpha_{p+1}^1, \ldots, \alpha_m^1),$$

$$r^{(i)} = (j_1, \ldots, j_{p-1}, \alpha_p^q, \alpha_{p+1}^1, \ldots, \alpha_{p+i-1}^1, \alpha_{p+i}^2, \alpha_{p+i+1}^1, \ldots, \alpha_m^1), \quad i = \overline{1, m-p}.$$

В частности,

$$O(r_1) = \bigcup_{i=1}^{m} r^{(i)}, \tag{4.2}$$

где $r^{(i)} = (\alpha_1^1, \alpha_2^1, \ldots, \alpha_{i-1}^1, \alpha_i^2, \alpha_{i+1}^1, \ldots, \alpha_m^1)$.

Алгоритм φ построения последовательности с. п. описывается следующим образом.

Алгоритм φ. 1-й шаг. Строим с. п. $r_1 = (\alpha_1^1, \alpha_2^1, \ldots, \alpha_m^1)$.

k-й шаг ($k = 2, 3, \ldots$). Строим множество с. п. W_k по одному из следующих правил.

Правило 1. $W_k = \bigcup\limits_{i=1}^{k-1} O(r_i) \setminus \{r_1, r_2, \ldots, r_{k-1}\}$.

Правило 2. $W_k = (W_{k-1} \setminus r_{k-1}) \cup O(r_{k-1})$, причем $W_1 = r_1$.

Далее, находим такую систему представителей r_k, что

$$Q(r_k) = \min_{r \in W_k} Q(r), \tag{4.3}$$

и переходим к $(k+1)$-му шагу. Нетрудно убедиться, что множества W_k, построенные по правилам 1 и 2, совпадают.

Перейдем к обоснованию того, что алгоритм φ строит требуемую последовательность с. п. Рассмотрим

$$r^* = (j_1^*, \ldots, j_{p-1}^*, \alpha_p^q, \alpha_{p+1}^1, \ldots, \alpha_m^1),$$
$$1 \leqslant q \leqslant s_p, \quad 1 \leqslant p \leqslant m.$$

Положим

$$R(r^*) = \{r \mid j_i = j_i^*, \quad i = \overline{1, p-1}, \quad j_p \geqslant \alpha_p^q, \quad r \in W\},$$

т. е. $R(r^*)$ — множество с. п., у которых $j_i = j_i^*$ $(i = \overline{1, p-1})$, а $j_p \in \{\alpha_p^q, \alpha_p^{q+1}, \ldots, \alpha_p^{s_p}\}$. Естественно считать, что $R(r_1) = W$.

Очевидно, что в этих обозначениях справедливо соотношение

$$\min_{r \in R(r^*)} Q(r) = Q(r^*), \tag{4.4}$$

т. е. r^* — лучший представитель множества $R(r^*)$, построенного по описанному правилу.

Из определения множества $R(r)$, а также из соотношений (4.1) и (4.2) видно, что множество $O(r_{k-1})$ обладает свойством

$$\bigcup_{r \in O(r_{k-1})} R(r) = R(r_{k-1}) \setminus r_{k-1}.$$

Отсюда, вводя обозначение $R_k = \bigcup\limits_{r \in W_k} R(r)$ и учитывая правила построения множества W_k, получаем

$$R_k = \Bigl\{\bigcup_{r \in W_{k-1} \setminus r_{k-1}} R(r)\Bigr\} \cup \Bigl\{\bigcup_{r \in O(r_{k-1})} R(r)\Bigr\} =$$
$$= \Bigl\{\bigcup_{r \in W_{k-1} \setminus r_{k-1}} R(r)\Bigr\} \cup \Bigl\{R(r_{k-1}) \setminus r_{k-1}\Bigr\} =$$
$$= \Bigl\{\bigcup_{r \in W_{k-1}} R(r)\Bigr\} \setminus r_{k-1} = R_{k-1} \setminus r_{k-1}.$$

Таким образом, на k-м шаге работы алгоритма φ из множества R_{k-1} элиминируется элемент r_{k-1}, и в силу $R_1 = \bigcup\limits_{r \in W_1} R(r) = R(r_1) = W$ имеем

$$R_k = W \setminus \{r_1, r_2, \ldots, r_{k-1}\}. \tag{4.5}$$

Так как соотношение (4.4) справедливо для всякого $r^* \in W_k$, то, учитывая соотношение (4.3), получаем

$$Q(r_k) = \min_{r^* \in W_k} Q(r^*) = \min_{r^* \in W_k} \min_{r \in R(r^*)} Q(r) = \min_{r^* \in R_k} Q(r^*).$$

Следовательно, в силу соотношения (4.5)

$$Q(r_k) = \min_{r \in W \setminus \{r_1, \ldots, r_{k-1}\}} Q(r),$$

т. е. алгоритм φ строит последовательность с. п. в порядке неубывания функции $Q(r)$.

Проиллюстрируем работу алгоритма на примере.

Пример. Покажем, как строить последовательность с. п. в порядке неубывания функции $Q(r)$ для случая, когда

$$\Gamma_1 = \{10, 7, 4\}, \quad \Gamma_2 = \{3, 6, 2, 10, 15\}, \quad \Gamma_3 = \{6, 5, 8, 1\}.$$

В нашем примере

$$\alpha_1^1 = 3, \quad \alpha_1^2 = 2, \quad \alpha_1^3 = 1, \quad \alpha_2^1 = 3, \quad \alpha_2^2 = 1, \quad \alpha_2^3 = 2,$$

$$\alpha_2^4 = 4, \quad \alpha_2^5 = 5, \quad \alpha_3^1 = 4, \quad \alpha_3^2 = 2, \quad \alpha_3^3 = 1, \quad \alpha_3^4 = 3.$$

1-й шаг. Строим множество W_1. В нашем случае

$$W_1 = \{c_1^1\}, \quad c_1^1 = r_1, \; r_1 = (\alpha_1^1, \alpha_2^1, \alpha_3^1) = (3, 3, 4), \quad Q(r_1) = 7.$$

2-й шаг. Строим множества

$$W_2 = (W_1 \setminus r_1) \cup O(r_1) = O(r_1) = \{c_1^2, c_2^2, c_3^2\},$$

$$c_1^2 = (\alpha_1^2, \alpha_2^1, \alpha_3^1) = (2, 3, 4), \quad c_2^2 = (\alpha_1^1, \alpha_2^2, \alpha_3^1) = (3, 1, 4),$$

$$c_3^2 = (\alpha_1^1, \alpha_2^1, \alpha_3^2) = (3, 3, 2),$$

$$Q(c_1^2) = 10, \quad Q(c_2^2) = 8, \quad Q(c_3^2) = 11.$$

Поскольку наименьшее значение функции достигается на с. п. c_2^2, то $r_2 = c_2^2$. Переходим к 3-му шагу.

3-й шаг. Строим множество W_3. Для этого $r_2 = c_2^2$ заменяем множеством

$$O(r_2) = \{(\alpha_1^1, \alpha_2^3, \alpha_3^1), \; (\alpha_1^1, \alpha_2^2, \alpha_3^2)\} = \{(3, 2, 4), (3, 1, 2)\}.$$

Получаем:

$$W_3=\{c_1^3,\ c_2^3,\ c_3^3,\ c_4^3\},$$

$$c_1^3=c_1^2=(2,\ 3,\ 4),\quad c_2^3=c_3^2=(3,\ 3,\ 2),$$

$$c_3^3=(3,\ 2,\ 4),\qquad c_4^3=(3,\ 1,\ 2),$$

$$Q(c_1^3)=10,\quad Q(c_2^3)=11,$$

$$Q(c_3^3)=11,\quad Q(c_4^3)=12.$$

Следовательно, $r_3=c_1^3$, $Q(r_3)=10$.

Итак, мы построили три лучших системы представителей:

$$r_1=(\alpha_1^1,\ \alpha_2^1,\ \alpha_3^1)=(3,\ 3,\ 4),$$

$$r_2=(\alpha_1^1,\ \alpha_2^2,\ \alpha_3^1)=(3,\ 1,\ 4),$$

$$r_3=(\alpha_1^2,\ \alpha_2^3,\ \alpha_3^1)=(2,\ 3,\ 4),$$

со значениями функции $Q(r_1)=7$, $Q(r_2)=8$, $Q(r_3)=10$. Если работу алгоритма продолжить, то можно выписать последовательность всех систем представителей.

§ 5. Вычислительные аспекты построения последовательности планов

При практической реализации процесса упорядочения планов вспомогательной задачи $\bar{A}$, изложенной в § 3, естественно возникает вопрос о рациональной организации вычислительного процесса. В настоящем параграфе речь идет о сокращении объема запоминаемой в процессе работы алгоритма φ информации.

Точки решетки. Для построения последовательности планов необходимо на каждом шаге алгоритма φ находить минимум функции $Q(X)$ на подмножествах, получаемых в результате элиминации очередного плана, т. е. решать задачу минимизации функции

$$Q(X)=g_1(x_1)\,\omega g_2(x_2)\,\omega\ \ldots\ \omega g_n(x_n) \tag{5.1}$$

при условиях

$$\sum_{i=1}^{n} x_i=b, \tag{5.2}$$

$$x_i\in\Gamma_i=\{a_{i1},\ a_{i2},\ \ldots,\ a_{is_i}\},\quad i=\overline{1,\ n}, \tag{5.3}$$

где b, a_{ij} — целые положительные числа.

Рассмотрим некоторые вычислительные особенности решения подобных задач. Напомним, что задача (5.1)—(5.3) может быть решена при помощи функционального уравнения

$$f_k(z) = \min_{x_k \in \Gamma_k} [g_k(x_k)\,\omega f_{k-1}(z - x_k)], \quad k = \overline{2, n}, \tag{5.4}$$

$$f_1(z) = g_1(z).$$

Для этого необходимо знать значения функций $f_k(z)$ ($k = \overline{1, n}$) в промежуточных точках $0 \leqslant z \leqslant b$. Так как все исходные параметры задачи, т. е. числа b, a_{ij}, целые, то достаточно в качестве промежуточных точек, которые мы в дальнейшем будем называть *точками решетки*, брать лишь целые числа $0, 1, 2, \ldots, b$. Но и в этом случае при больших значениях b объем информации, которую необходимо хранить, решая задачи с помощью уравнений (5.4), огромен. Все это делает актуальной проблему сокращения количества точек решетки до такого числа, которое приемлемо с точки зрения машинной реализации.

Оказывается, что в большинстве случаев для нахождения оптимального плана задачи (5.1)—(5.3) нет необходимости в качестве точек решетки брать все целые числа от 0 до b.

Определение. Число d будем называть *нормальным*, если существуют такие числа $a_{i\lambda_i} \in \Gamma_i$ ($i = \overline{1, n}$) и натуральное число k, что

$$\sum_{i=1}^{k} a_{i\lambda_i} = d, \quad d + \sum_{i=k+1}^{n} a_{i\lambda_i} = b.$$

Нуль и b также будем считать нормальными числами. Множество всех нормальных чисел обозначим через N.

Связь между нормальными числами и точками решетки устанавливается следующей теоремой.

Теорема. *Если в качестве точек решетки взять все нормальные числа, то использование функционального уравнения* (5.4) *позволяет найти решение задачи* (5.1)—(5.3).

Доказательство. Пусть $\alpha = (a_{1t_1}, a_{2t_2}, \ldots$ $\ldots, a_{nt_n})$—оптимальный план задачи $\bar{A}$. Очевидно, что при нахождении этого плана с помощью функционального уравнения (5.4) необходимо, чтобы числа

$$a_p=\sum_{i=1}^{p} a_{it_i},\quad p=\overline{1, n},$$

являлись точками решетки. Поэтому достаточно показать, что числа a_p $(p=\overline{1, n})$ являются нормальными числами. Так как $\alpha=\left(a_{1t_1}, \ldots, a_{nt_n}\right)$ — план задачи $\bar{A}$, то имеет место равенство

$$\sum_{i=1}^{n} a_{it_i}=b,$$

т. е.

$$a_p+\sum_{i=p+1}^{n} a_{it_i}=b,\quad p=\overline{1, n-1}.$$

Отсюда следует, что числа a_p $(p=\overline{1, n-1})$ — нормальные. Так как число $a_n=b$ нормально по определению, то теорема доказана.

Прежде чем приступить к изложению процедуры построения нормальных чисел, введем вспомогательное понятие γ-преобразования произвольной последовательности $M=(\mu_1, \mu_2, \ldots, \mu_p)$.

При $\mu_p>0$ последовательность M заменяем s последовательностями

$$\begin{gathered}(\mu_1, \mu_2, \ldots, \mu_p, \mu_p-\gamma_1),\\ (\mu_1, \mu_2, \ldots, \mu_p, \mu_p-\gamma_2),\\ \cdots\cdots\cdots\cdots\\ (\mu_1, \mu_2, \ldots, \mu_p, \mu_p-\gamma_s).\end{gathered}$$

При $\mu_p=0$ последовательность M оставляем без изменения.

При $\mu_p<0$ последовательность M вычеркиваем.

Такое преобразование будем называть *γ-преобразованием* последовательности M с помощью чисел $\gamma_1, \gamma_2, \ldots, \gamma_s$.

Алгоритм построения точек решетки. i-й шаг $(i=\overline{1, n})$. Применим γ-преобразование к каждой последовательности, построенной к началу i-го шага с помощью чисел

$$a_{n-i+1,1},\quad a_{n-i+1,2}, \ldots, a_{n-i+1,s_{n-i+1}},$$

причем к началу 1-го шага имеется одна последовательность, состоящая из числа b.

$(n+1)$-й шаг. Из последовательностей, построенных после n шагов, выбираются такие последовательности

$$M_1 = (\mu_1^1, \mu_1^2, \ldots, \mu_1^{t_1}),$$
$$M_2 = (\mu_2^1, \mu_2^2, \ldots, \mu_2^{t_2}),$$
$$\cdots\cdots\cdots\cdots$$
$$M_p = (\mu_p^1, \mu_p^2, \ldots, \mu_p^{t_p}),$$

у которых $\mu_1^{t_1} = \mu_2^{t_2} = \ldots = \mu_p^{t_p} = 0$ и $\{1, 2, \ldots \ldots, n - t_s + 1\} \in \varepsilon$ $(s = \overline{1, p})$, где $\varepsilon = \{i \mid 0 \in \Gamma_i\}$.

$(n+2)$-й шаг. Находим множества чисел $\bigcup\limits_{l=1}^{p} M_l = N'$.

Покажем, что множество N', построенное в результате описанной процедуры, состоит из нормальных чисел. Действительно, пусть $\mu_s^l \in N'$ — элемент последовательности M_s, стоящий на l-м месте. Следовательно, найдутся такие $a_{i\lambda_i} \in \Gamma_i$ $(i = n - t_s + 2, \ldots \ldots, n - l + 1)$, что

$$\mu_s^l - \sum_{i=n-t_s+2}^{n-l+1} a_{i\lambda_i} = \mu_s^{t_s} = 0, \quad \{1, 2, \ldots, n - t_s + 1\} \in \varepsilon,$$

т. е.

$$0 \in \Gamma_i \quad (i = \overline{1, n - t_s + 1}).$$

Отсюда имеем

$$\mu_s^l = \sum_{i=1}^{n-l} a_{i\lambda_i}.$$

Но по построению

$$\mu_s^l = b - \sum_{i=n-l+2}^{n} a_{i\lambda_i}.$$

Таким образом, число μ_s^l является нормальным, и поэтому множество N' состоит из нормальных чисел.

Кроме того, покажем, что произвольное нормальное число принадлежит множеству N'. Действительно, для произвольного $a \in N$ существуют, по определению, такие числа $a_{i\lambda_i} \in \Gamma_i$ $(i = \overline{1, n})$ и число k, что

$$a = \sum_{i=1}^{k} a_{i\lambda_i}, \quad a + \sum_{i=k+1}^{n} a_{i\lambda_i} = b.$$

Вместе с тем последовательность

$$\left(b,\ b - a_{n\lambda_n},\ \ldots,\ b - \sum_{i=k+1}^{n} a_{i\lambda_i},\ \ldots,\ b - \sum_{i=1}^{n} a_{i\lambda_i}\right)$$

содержит нормальное число a. С другой стороны, эта последовательность сама содержится в множестве последовательностей $M_1, M_2, \ldots, M_p$, так как $b - \sum_{i=1}^{n} a_{i\lambda_i} = 0$. Следовательно, $a \in N'$.

Таким образом, показано, что построенное множество N' совпадает с множеством нормальных чисел.

Хранение информации о планах. Другим вопросом, связанным с рациональной организацией процесса φ построения последовательности планов вспомогательной задачи, рассмотренной в § 3, является вопрос хранения информации о планах, которые порождаются h-оптимальными планами.

Внимательно присмотревшись к алгоритму φ, замечаем, что h-оптимальный план, являющийся лучшим на некотором шаге, заменяется системой планов, насчитывающей до $\sum_{i=1}^{h+1} (s_i - 1)$ планов. Это объясняется тем, что каждая p-я компонента $(p = \overline{1, h+1})$ плана может заменяться поочередно всеми элементами множества Γ_i. Оказывается, из всех этих планов можно запоминать лишь один, снабдив его дополнительной информацией. Для этого вводится понятие (h, g)-оптимального плана.

Пусть $1 \leqslant h \leqslant n$. Зафиксируем величины $\bar{x}_i \in \Gamma_i$ $(i = \overline{h+2, n})$. Пусть $\left(x_1^{(g)}, \ldots, x_{h+1}^{(g)}\right)$ — оптимальный план задачи

$$g_1(x_1)\,\omega\, g_2(x_2)\,\omega\, \ldots\, \omega\, g_{h+1}(x_{h+1}) \to \min,$$

$$\sum_{i=h+2}^{n} \bar{x}_i + \sum_{i=1}^{h+1} x_i \geqslant b,$$

$$x_i \in \Gamma_i, \quad i = \overline{1, h},$$

$$x_{h+1} \in \Gamma_{h+1}^{(g)}\left(\bar{x}_{h+2}, \ldots, \bar{x}_n\right).$$

Здесь множество $\Gamma_{h+1}^{(g)}(\bar{x}_{h+2}, \ldots, \bar{x}_n)$ определяется с помощью рекуррентного соотношения

$$\Gamma_{h+1}^{(g)}(\bar{x}_{h+2}, \ldots, \bar{x}_n) = \Gamma_{h+1}^{(g-1)}(\bar{x}_{h+2}, \ldots, \bar{x}_n) \setminus \{x_{h+1}^{(g-1)}\},$$

где $x_{h+1}^{(g-1)}$ — последняя (по порядку) компонента в оптимальном плане $(x_1^{(g-1)}, \ldots, x_{h+1}^{(g-1)})$ задачи

$$g_1(x_1)\,\omega g_2(x_2)\,\omega \ldots \omega g_{h+1}(x_{h+1}) \to \min,$$

$$\sum_{i=h+2}^{n} \bar{x}_i + \sum_{i=1}^{h} x_i \geqslant b,$$

$$x_i \in \Gamma_i, \quad i = \overline{1, h},$$

$$x_{h+1} \in \Gamma_{h+1}^{(g-1)}(\bar{x}_{h+2}, \ldots, \bar{x}_n),$$

причем

$$\Gamma_{h+1}^{(1)}(\bar{x}_{h+2}, \ldots, \bar{x}_n) = \left\{ x_{h+1} \mid x_{h+1} \in \Gamma_{h+1}, \quad \sum_{i=h+2}^{n} \bar{x}_i + x_{h+1} + \sum_{i=1}^{h} \max_{1 \leqslant j \leqslant s_i} a_{ij} \geqslant b \right\}.$$

План $X = (x_1^{(g)}, \ldots, x_h^{(g)}, x_{h+1}^{(g)}, \bar{x}_{h+2}, \ldots, \bar{x}_n)$ задачи $\bar{A}$ будем называть (h, g)*-оптимальным* $(1 \leqslant h \leqslant n-1, 1 \leqslant g \leqslant s_{h+1})$ и обозначать его через ${}_g[\bar{x}_{h+2}, \ldots, \bar{x}_n)$. В частности, $(0, 1)$-оптимальный план, который следует обозначать через ${}_1[\varnothing) = (x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)})$, является оптимальным планом задачи $\bar{A}$.

Для (h, g)-оптимального плана $X = {}_g[\bar{x}_{h+2}, \ldots, \bar{x}_n)$ определим множество $O(X)$ по следующему правилу:

$$O(X) = \{X^{(1)}, X^{(2)}, \ldots, X^{(h+1)}\},$$

где

$$X^{(1)} = {}_{g+1}[\bar{x}_{h+2}, \ldots, \bar{x}_n),$$

$$X^{(k)} = {}_g[x_{h-k+1}^{(g)}, \ldots, x_{h+1}^{(g)}, \bar{x}_{h+2}, \ldots, \bar{x}_n), \quad k = \overline{2, h+2}.$$

Теперь можно сформулировать процедуру построения последовательности планов $r_1, r_2, \ldots$ следующим образом.

Алгоритм φ. 1-й шаг. Полагаем

$$W_1 = \{{}_1[\varnothing)\}, \quad r_1 = {}_1[\varnothing) = (x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}).$$

k-й шаг ($k = 2, 3, \ldots$). Образуем множество W_k по правилу

$$W_k = (W_{k-1} \setminus r_{k-1}) \cup O(r_{k-1}).$$

Находим r_k такой, что

$$Q(r_k) = \min_{X \in W_k} Q(X),$$

и переходим к $(k+1)$-му шагу.

Заметим, что множество W_k можно строить также и по следующему правилу:

$$W_k = \bigcup_{i=1}^{k-1} O(r_i) \setminus \{r_1, r_2, \ldots, r_{k-1}\}.$$

Обоснование того, что приведенный алгоритм строит требуемую последовательность планов, аналогично приведенному в § 3.

Проиллюстрируем работу алгоритма на примере из § 3.

Пример. 1-й шаг. Полагаем

$$W_1 = \{B_1^1\} = \{{}_1[\varnothing)\}, \quad r_1 = {}_1[\varnothing).$$

Здесь

$$B_1^1 = {}_1[\varnothing) = (3, 1, 1), \quad Q(B_1^1) = 670.$$

2-й шаг. Образуем множество $W_2 = (W_1 \setminus r_1) \cup \cup O(r_1)$. Так как

$$O(r_1) = \{{}_2[\bar{x}_3 = 2), \quad {}_3[\bar{x}_2 = 3,\ \bar{x}_3 = 1)\},$$

то

$$W_2 = \{B_1^2, B_2^2\},$$

где

$$B_1^2 = {}_2[\bar{x}_3 = 2) = (3, 0, 2), \quad Q(B_1^2) = 670,$$

$$B_2^2 = {}_2[\bar{x}_2 = 3,\ \bar{x}_3 = 1) = (1, 3, 1), \quad Q(B_2^2) = 690.$$

Находим план r_2. В нашем случае

$$r_2 = B_1^2 = (3, 0, 2), \quad Q(r_2) = 670.$$

Переходим к 3-му шагу.

3-й шаг. Образуем множество $W_3 = (W_2 \setminus r_2) \cup \cup O(r_2)$. Так как

$$Q(r_2) = \{_3[\bar{x}_3 = 0),\ _2[\bar{x}_2 = 3,\ \bar{x}_3 = 2)\},$$

то

$$W_3 = \{B_1^3,\ B_2^3,\ B_3^3\},$$

где

$$B_1^3 = B_2^2 = {}_2[\bar{x}_2 = 3,\ \bar{x}_3 = 1) = (1,\ 3,\ 1), \quad Q(B_1^3) = 690,$$
$$B_2^3 = {}_3[\bar{x}_3 = 0) = (3,\ 2,\ 0), \quad Q(B_2^3) = 690,$$
$$B_3^3 = {}_2[\bar{x}_2 = 3,\ \bar{x}_3 = 2) = (0,\ 3,\ 2), \quad Q(B_3^3) = 680.$$

Находим план $r_3 = B_3^3 = (0,\ 3,\ 2)$ и значение функции $Q(r_3) = 680$.

Продолжая работу алгоритма, можно построить все планы исходной задачи.

ГЛАВА III

РЕШЕНИЕ ЗАДАЧ РАЗМЕЩЕНИЯ ПРОИЗВОДСТВА

§ 1. Решение однопродуктовых задач размещения

В настоящем параграфе излагаются алгоритмы решения однопродуктовых задач размещения в вариантной постановке.

Задача с балансом. Напомним (см. § 1 гл. I), что однопродуктовая задача размещения с балансом (задача A) состоит в минимизации суммарных производственно-транспортных затрат

$$\sum_{i=1}^{m} f_i(x_i) + \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{1.1}$$

при ограничениях

$$\sum_{j=1}^{n} x_{ij} = x_i, \quad i=\overline{1, m}, \tag{1.2}$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j=\overline{1, n}, \tag{1.3}$$

$$x_i \in \Gamma_i = \left\{a_i^1, a_i^2, \ldots, a_i^{s_i}\right\}, \quad i=\overline{1, m}, \tag{1.4}$$

$$x_{ij} \geqslant 0, \quad i=\overline{1, m}, \quad j=\overline{1, n}. \tag{1.5}$$

Из условий (1.2) и (1.3) вытекает общий баланс производства и потребления

$$\sum_{i=1}^{m} x_i = \sum_{j=1}^{n} b_j. \tag{1.6}$$

В дальнейшем сумму $\sum\limits_{j=1}^{n} b_j$ будем обозначать через b. Вектор $X=(x_1, x_2, \ldots, x_m)$, компоненты которого удовлетворяют условиям (1.4) и (1.6), будем называть *вариантом размещения задачи* A.

Таким образом, задача A состоит в нахождении такого варианта размещения, который минимизирует функцию

$$F(X)=\sum_{i=1}^{m} f_i(x_i)+T(X),$$

где

$$T(X)=\min\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

при условиях

$$\sum_{i=1}^{m} x_{ij}=b_j, \quad \sum_{j=1}^{n} x_{ij}=x_i, \quad x_{ij}\geqslant 0,$$
$$i=\overline{1, m}, \quad j=\overline{1, n}.$$

Поэтому задача A принимает вид

$$F(X)\to\min,$$
$$\sum_{i=1}^{m} x_i=b,$$
$$x_i\in\Gamma_i, \quad i=\overline{1, m}.$$

Эта задача является частным случаем задачи, рассмотренной в § 3 гл. II.

В качестве вспомогательной задачи $\bar{A}$ возьмем задачу минимизации функции

$$Q(X)=\sum_{i=1}^{m} g_i(x_i)$$

при условиях

$$\sum_{i=1}^{m} x_i=b,$$
$$x_i\in\Gamma_i, \quad i=\overline{1, m}.$$

Здесь $g_i(x_i)=f_i(x_i)+t_i(x_i)$, где

$$t_i(x_i)=\min\sum_{j=1}^{n} c_{ij}x_{ij} \tag{1.7}$$

при условиях

$$\sum_{j=1}^{n} x_{ij} = x_i, \quad x_{ij} \leqslant b_j, \; x_{ij} \geqslant 0, \; i = \overline{1, m}, \; j = \overline{1, n}. \qquad (1.8)$$

Покажем, что функция $Q(X)$ является минорантой функции $F(X)$, т. е. что для любого варианта размещения $X = (x_1, x_2, \ldots, x_m)$ справедливо неравенство $F(X) \geqslant Q(X)$. Действительно, пусть $(x_{11}, x_{12}, \ldots, x_{mn})$ — оптимальное решение транспортной задачи, соответствующее варианту размещения $X = (x_1, x_2, \ldots, x_m)$, а $(\bar{x}_{11}, \bar{x}_{12}, \ldots, \bar{x}_{mn})$ — совокупность оптимальных решений задач (1.7)—(1.8) при $i = \overline{1, m}$. Очевидно, что

$$\sum_{j=1}^{n} c_{ij} x_{ij} \geqslant \sum_{j=1}^{n} c_{ij} \bar{x}_{ij}, \quad i = \overline{1, m}.$$

Следовательно,

$$F(X) - Q(X) = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} c_{ij} x_{ij} - \sum_{j=1}^{n} c_{ij} \bar{x}_{ij} \right) \geqslant 0,$$

что и требовалось доказать.

Как строить последовательность вариантов размещения $r_1, r_2, \ldots$ вспомогательной задачи $\bar{A}$ в порядке неубывания миноранты $Q(X)$, показано в § 3 гл. II.

Метод ψ *решения однопродуктовой задачи* A *размещения производства.* 1-й шаг. При помощи алгоритма φ находим план r_1 вспомогательной задачи $\bar{A}$ и вычисляем величину $L_1 = F(r_1)$.

k-й шаг ($k = 2, 3, \ldots$). Находим план r_k и сравниваем числа $Q(r_k)$ и L_{k-1}. Если $Q(r_k) \geqslant L_{k-1}$, то работа алгоритма заканчивается и оптимальным вариантом размещения является вариант задачи A, соответствующий величине L_{k-1}. Если $Q(r_k) < L_{k-1}$, то вычисляем величину $F(r_k)$ и сравниваем числа $F(r_k)$ и L_{k-1}. При $L_{k-1} \leqslant F(r_k)$ полагаем $L_k = L_{k-1}$ и переходим к $(k+1)$-му шагу. При $L_{k-1} > F(r_k)$ полагаем $L_k = F(r_k)$ и переходим к $(k+1)$-му шагу.

Пример. Пусть заданы три пункта возможного размещения предприятий и четыре пункта потребления выпускаемой продукции. Все исходные данные представлены в таблице 1.

Таблица 1

Возможные пункты производства	Возможные объемы производства a_i^h	Приведенные затраты $f_i\left(a_i^h\right)$	Пункты потребления			
			1	2	3	4
			Объемы потребления b_j			
			4	6	10	5
1	0 5 10	0 200 250	2	4	3	5
2	0 10 15	0 150 200	6	2	4	4
3	0 5 15	0 130 170	3	5	5	6

На предварительном этапе работы алгоритма строим функции $g_i(x)$, решая задачи (1.7)—(1.8) при $i=1, 2, 3$ (см. таблицу 2; прочерк в таблице означает, что для данного x функция $g_i(x)$ не определена).

Таблица 2

x	0	5	10	15
$g_1(x)$	0	211	276	—
$g_2(x)$	0	—	178	248
$g_3(x)$	0	147	—	237

Таким образом, вспомогательная задача состоит в данном случае в минимизации функции

$$Q(X)=g_1(x_1)+g_2(x_2)+g_3(x_3)$$

при условиях

$$x_1 + x_2 + x_3 = 25,$$

$$x_1 \in \{0, 5, 10\}, \quad x_2 \in \{0, 10, 15\}, \quad x_3 \in \{0, 5, 15\}.$$

Вспомогательную задачу решаем при помощи функциональных уравнений динамического программирования (см. § 3 гл. II). Результаты представим в виде подробных таблиц, которые в дальнейшем понадобятся при описании алгоритма решения исходной задачи.

В таблице 3 приведены значения функции $f_1(z) = g_1(z)$. В таблице 4 приведены значения функции

Таблица 3

z	0	5	10	15	20	25
$f_1(z)$	0	211	276	—	—	—

Таблица 4

x_2 \ z	0	5	10	15	20	25
0	0	211	276	—	—	—
10	—	—	178	389	454	—
15	—	—	—	248	459	524
$f_2(z)$	0	211	178	248	454	524

$g_2(x_2) + f_1(z - x_2)$, а в последней строке — значение функции

$$f_2(z) = \min_{x_2 \in \{0, 10, 15\}} [g_2(x_2) + f_2(z - x_2)].$$

В таблице 5 приведены значения функции $g_3(x_3) + f_2(z - x_3)$, а в последней строке — функции

$$f_3(z) = \min_{x_3 \in \{0, 5, 15\}} [g_3(x_3) + f_2(z - x_3)].$$

Таблица 5

x_3 \ z	0	5	10	15	20	25
0	0	211	178	248	454	524
5	—	147	358	325	395	601
15	—	—	—	237	448	415
$f_3(z)$	0	147	178	237	395	415

1-й шаг. Строим множество W_1 всех 2-оптимальных планов:

$$W_1 = \{B_1^1, B_2^1, B_3^1\},$$

$$B_1^1 = (*, *, 0) = (10, 15, 0),$$

$$B_2^1 = (*, *, 5) = (10, 10, 5),$$

$$B_3^1 = (*, *, 15) = (0, 10, 15).$$

Вычисляем значения

$$Q(B_1^1) = 524, \quad Q(B_2^1) = 601, \quad Q(B_3^1) = 415.$$

Находим план $r_1 = B_3^1$ и соответствующее ему значение миноранты $Q(r_1) = 415$.

Вычисляем величину $F(r_1)$. Для этого необходимо решить следующую транспортную задачу, заданную таблицей 6. В таблице в правых нижних углах клеток стоят объемы перевозок для оптимального плана

Таблица 6

x_i \ a_j	4	6	10	5
0	2	4	3	5
10	6	2 $\boxed{6}$	4	4 $\boxed{4}$
15	3 $\boxed{4}$	5	5 $\boxed{10}$	6 $\boxed{1}$

транспортной задачи. Так как транспортные затраты равны 96, то $L_1 = F(r_1) = 416$.

2-й шаг. Преобразуем множество W_1 в множество W_2. Для этого план $r_1 = B_3^1$ заменяем системой планов

$$O(r_1) = \{(*, 0, 15)\}.$$

Получаем

$$W_1 = \{B_1^2, B_2^2, B_3^2\},$$

$$B_1^2 = B_1^1 = (*, *, 0) = (10, 15, 0),$$

$$B_2^2 = B_2^1 = (*, *, 5) = (10, 10, 5),$$

$$B_3^2 = (*, 0, 15) = (10, 0, 5),$$

$$Q(B_1^2) = 524, \quad Q(B_2^2) = 601, \quad Q(B_3^2) = 513.$$

Находим план $r_2 = B_3^2$ и $Q(r_2) = 513$. Так как $Q(r_2) > L_1$ $(513 > 416)$, то r_1 — оптимальный план задачи, в котором $x_1 = 0$, $x_2 = 10$, $x_3 = 15$, а объемы перевозок x_{ij} заданы в таблице 6.

Задачи с небалансом. На практике часто возникают задачи размещения, для которых число $b = \sum_{j=1}^{n} b_j$ не выражается в виде суммы $b = \sum_{i=1}^{m} x_i$, где $x_i \in \Gamma_i$ $(i = \overline{1, m})$. Это приводит к нарушению либо условий (1.2), либо условий (1.3). Таким образом, появляются еще две задачи.

Задача A_1. Требуется минимизировать функцию

$$\sum_{i=1}^{m} f_i(x_i) + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

при условиях (1.3)—(1.5) и

$$\sum_{j=1}^{n} x_{ij} \leqslant x_i, \quad i = \overline{1, m}. \tag{1.9}$$

Задача A_2. Требуется минимизировать функцию

$$\sum_{i=1}^{m} f_i(x_i) + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

при условиях (1.2), (1.4), (1.5) и

$$\sum_{i=1}^{m} x_{ij} \geqslant b_j, \quad j = \overline{1, n}. \tag{1.10}$$

Таким образом, для задач A_1 и A_2 выполняется условие

$$\sum_{i=1}^{m} x_i \geqslant b. \tag{1.11}$$

Ранее в этом параграфе мы определили понятие варианта размещения для задачи A. Аналогично, *вариантом размещения задач* A_1 *и* A_2 будем называть вектор $X = (x_1, x_2, \ldots, x_m)$, компоненты которого удовлетворяют условиям (1.4), (1.11). Очевидно, задача A_i $(i = 1, 2)$ состоит в нахождении такого варианта размещения, который минимизирует функцию

$$F_k(X) = \sum_{i=1}^{m} f_i(x_i) + T_i(X),$$

где $T_k(X) = \min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$ при условиях (1.7), (1.5), (1.9) для $k = 1$ и условиях (1.2), (1.5), (1.10) для $k = 2$. Поэтому задачи A_1 и A_2 принимают следующий вид.

Задача A_1.

$$F_1(X) \to \min,$$

$$\sum_{i=1}^{m} x_i \geqslant b,$$

$$x_i \in \Gamma_i, \quad i = \overline{1, m}.$$

Задача A_2.

$$F_2(X) \to \min,$$

$$\sum_{i=1}^{m} x_i \geqslant b,$$

$$x_i \in \Gamma_i, \quad i = \overline{1, m}.$$

Эти задачи являются частным случаем задач, рассмотренных в § 3 гл. II.

В качестве вспомогательной задачи $\bar{A}_k$ $(k = 1, 2)$ возьмем задачу минимизации функции

$$Q_k(X) = \sum_{i=1}^{m} g_i(x_i)$$

при условиях

$$\sum_{i=1}^{m} x_i \geqslant b, \quad x_i \in \Gamma_i \quad i = \overline{1, m}.$$

Здесь для задачи A_1

$$g_i(x_i) = f_i(x_i) + \sum_{j=1}^{n} d_j(x_i)/m,$$

где

$$d_j(x_i) = \min \sum_{\gamma=1}^{m} c_{\gamma j} x_{\gamma j}$$

при условиях

$$\sum_{\gamma=1}^{m} x_{\gamma j} = b_j, \quad 0 \leqslant x_{\gamma j} \leqslant \max_{1 \leqslant \beta \leqslant s_\gamma} a_\gamma^\beta,$$

$$\gamma = \overline{1, m}, \quad \gamma \neq i, \quad 0 \leqslant x_{ij} \leqslant x_i,$$

а для задачи A_2

$$g_i(x_i) = f_i(x_i) + e_i(x_i),$$

где

$$e_i(x_i) = x_i \min_{1 \leqslant j \leqslant n} c_{ij}.$$

Очевидно, что функции $Q_1(X)$ и $Q_2(X)$ являются минорантами для функций $F_1(X)$ и $F_2(X)$ соответственно.

Алгоритм решения задач A_1 и A_2 описывается аналогично алгоритму решения задачи A.

§ 2. Решение задачи размещения с двойным транспортом

В настоящем параграфе излагается алгоритм решения однопродуктовой задачи размещения в вариантной постановке с двойным транспортом. Напомним (см. § 2 гл. I), что задача размещения с двойным транспортом (задача A) состоит в минимизации функции

$$\sum_{i=1}^{m} f_i(x_i) + \sum_{k=1}^{\bar{n}} \sum_{i=1}^{m} \bar{c}_{ki}\bar{x}_{ki} + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}x_{ij}$$

при ограничениях

$$\sum_{j=1}^{n} x_{ij} = x_i, \qquad i = \overline{1, m},$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \qquad j = \overline{1, n},$$

$$\sum_{k=1}^{\bar{n}} \bar{x}_{ki} = d(x_i), \quad i = \overline{1, m},$$

$$\sum_{i=1}^{m} \bar{x}_{ki} = \bar{b}_k, \qquad k = \overline{1, n},$$

$$x_i \in \Gamma_i = \left\{a_i^1, a_i^2, \ldots, a_i^{s_i}\right\}, \quad i = \overline{1, m},$$

$$x_{ij} \geqslant 0, \quad \bar{x}_{ki} \geqslant 0, \quad i = \overline{1, m}, \quad j = \overline{1, n}, \quad k = \overline{1, \bar{n}}.$$

Будем предполагать, что задача A имеет решение, т. е. существуют такие мощности $x_i \in \Gamma_i$ $(i = \overline{1, m})$, что

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad \sum_{i=1}^{m} d_i(x_i) \leqslant \sum_{k=1}^{\bar{n}} \bar{b}_k.$$

Вариантом размещения задачи A будем называть вектор $X = (x_1, x_2, \ldots, x_m)$, компоненты которого удовлетворяют условиям

$$\sum_{i=1}^{m} x_i = \sum_{j=1}^{n} b_j, \quad x_i \in \Gamma_i, \quad i = \overline{1, m}.$$

Таким образом, задача A состоит в нахождении такого варианта размещения, который минимизирует функцию

$$F(X)=\sum_{i=1}^{m} f_i(x_i)+T(X)+\overline{T}(X),$$

где $T(X)=\min\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$ при условиях

$$\sum_{i=1}^{m} x_{ij}=b_j,\quad \sum_{j=1}^{n} x_{ij}=x_i,\quad x_{ij}\geqslant 0,$$
$$i=\overline{1,\,m},\quad j=\overline{1,\,n},$$

а $\overline{T}(X)=\min\sum_{k=1}^{\bar{n}}\sum_{i=1}^{m}\bar{c}_{ki}\bar{x}_{ki}$ при условиях

$$\sum_{i=1}^{m}\bar{x}_{ki}\leqslant\bar{b}_k,\quad \sum_{k=1}^{\bar{n}}\bar{x}_{ki}=d(x_i),$$
$$\bar{x}_{ki}\geqslant 0,\quad k=\overline{1,\,\bar{n}},\quad i=\overline{1,\,m}.$$

В качестве вспомогательной задачи $\bar{A}$ возьмем задачу минимизации функции

$$Q(X)=\sum_{i=1}^{m} g_i(x_i)$$

при условиях

$$\sum_{i=1}^{m} x_i=b,\quad x_i\in\Gamma_i,\quad i=\overline{1,\,m},$$

где $b=\sum_{j=1}^{n} b_j$, а функции $g_i(x_i)$ $(i=\overline{1,\,m})$ строятся по следующему правилу:

$$g_i(x_i)=f_i(x_i)+t_i(x_i)+\bar{t}_i(x_i).$$

Здесь $t_i(x_i)=\min\sum_{j=1}^{n} c_{ij}x_{ij}$ при условиях

$$\sum_{j=1}^{n} x_{ij}=x_i,\quad x_{ij}\leqslant b_j,\quad x_{ij}\geqslant 0,\quad j=\overline{1,\,n},$$

а $\bar{t}_i(x_i)=\min\sum_{k=1}^{\bar{n}}\bar{c}_{ki}\bar{x}_{ki}$ при условиях

$$\sum_{k=1}^{\bar{n}}\bar{x}_{ki}=d(x_i),\quad \bar{x}_{ki}\leqslant\bar{b}_k,\quad \bar{x}_{ki}\geqslant 0,\quad k=\overline{1,\,\bar{n}}.$$

Нетрудно показать (аналогично § 1), что функция $Q(X)$ является минорантой функции $F(X)$, т. е. для любого варианта размещения $X=(x_1, x_2, \ldots, x_m)$ выполняется неравенство $Q(X)\leqslant F(X)$.

Описание алгоритма решения задачи размещения с двойным транспортом аналогично приведенному в § 1. Проиллюстрируем его работу на примере.

Пример. Заданы три пункта возможного размещения предприятий. Для каждого из них заданы:

1) возможные объемы производства, определяемые наборами Γ_i;

2) соответствующие объемам производства потребности в сырье;

3) приведенные затраты $f_i(x_i)$.

Известны четыре пункта потребления (с объемами потребления b_j) и матрица транспортных затрат на перевозку единицы готовой продукции от пунктов возможного размещения до пунктов потребления.

Кроме того, известны пять пунктов добычи сырья, для каждого из которых определены запасы сырья, задана матрица транспортных затрат по перевозке единицы сырья от каждого пункта добычи до каждого пункта возможного размещения предприятий. Вся информация по задаче скомпонована в таблицах 7 и 8.

На предварительном этапе работы алгоритма строим функции $g_i(x_i)$ (см. таблицу 9). В каждой

Таблица 7

Возможные пункты производства	Возможные объемы производства a_i^h	Объемы потребления сырья $d\left(a_i^h\right)$	Приведенные затраты $f_i\left(a_i^h\right)$	Пункты потребления			
				1	2	3	4
				Объемы потребления b_j			
				4	6	10	5
1	0 5 10	0 10 20	0 150 180	2	4	3	5
2	0 10 15	0 20 30	0 150 200	6	2	4	4
3	0 5 15	0 10 30	0 130 170	3	5	5	6

Таблица 8

Пункты добычи сырья	Возможные пункты производства			Запасы сырья $\bar{b}_k$
	1	2	3	
1	1	3	6	15
2	4	5	3	20
3	2	4	3	10
4	6	1	2	5
5	5	7	4	10

Таблица 9

x	0	5	10	15
$g_1(x)$	0	171	231	—
$g_2(x)$	0	—	228	338
$g_3(x)$	0	172	—	322

клетке таблицы 9 проставлена сумма приведенных затрат и минимальных транспортных расходов как на перевозку данного объема готовой продукции x из пункта производства к потребителям, так и на перевозку соответствующего объема сырья $d(x_i)$ к заданному пункту производства из пунктов добычи сырья. При этом минимальные транспортные расходы определяются решением двух однострочных транспортных задач. Например, для клетки, расположенной на пересечении второй строки и четвертого столбца решаются две однострочные транспортные задачи, задаваемые таблицами 10 и 11.

Таблица 10

x_2 \ a_j	4	6	10	5
15	6	2 $\boxed{6}$	4 $\boxed{9}$	4

Таблица 11

$d(x_2)$ \ $\bar{b}_k$	15	20	10	5	10
30	3 $\boxed{15}$	5	4 $\boxed{10}$	1 $\boxed{5}$	7

Следовательно, вспомогательная задача в настоящем случае состоит в минимизации функции

$$Q(X) = g_1(x_1) + g_2(x_2)$$

при условиях

$$x_1 + x_2 + x_3 = 25,$$

$$x_1 \in \{0, 5, 10\}, \quad x_2 \in \{0, 10, 15\}, \quad x_3 \in \{0, 5, 15\}.$$

Функции $g_i(x_i)$ $(i = 1, 2, 3)$ заданы таблицей 9.

Как и в § 1, при построении последовательности планов нам понадобится информация, получаемая в процессе решения вспомогательной задачи. Представим ее в виде таблиц. В таблице 12 приведены значения функции $f_1(z) = g_1(z)$, в таблице 13 — значения функций $g_2(x_2) + f_1(z - x_2)$ и $f_2(z)$, а в таблице 14 — функций $g_3(x_3) + f_2(z - x_3)$ и $f_3(z)$.

Таблица 12

z	0	5	10	15	20	25
$f_1(z)$	0	171	231	—	—	—

Таблица 13

x_2 \ z	0	5	10	15	20	25
0	0	171	231	—	—	—
10	—	—	228	399	459	—
15	—	—	—	338	509	569
$f_2(z)$	0	171	228	338	459	569

Таблица 14

z / x_3	0	5	10	15	20	25
0	0	171	228	338	459	569
5	—	172	343	400	510	631
15	—	—	—	322	493	550
$f_3(z)$	0	171	231	322	459	550

1-й шаг: Строим множество W_1 всех 2-оптимальных планов:

$$W_1 = \{B_1^1, B_2^1, B_3^1\},$$

$$B_1^1 = (*, *, 0) = (10, 15, 0),$$

$$B_2^1 = (*, *, 5) = (10, 10, 5),$$

$$B_3^1 = (*, *, 15) = (0, 10, 15),$$

$$Q(B_1^1) = 569, \quad Q(B_2^1) = 631, \quad Q(B_3^1) = 550.$$

Находим $r_1 = B_3^1$, $Q(r_1) = 550$ и вычисляем величину $F(r_1)$. Для нахождения $F(r_1)$ необходимо решить две транспортные задачи, задаваемые таблицами 15 и 16. Так как транспортные затраты равны соответственно 96 и 140, то $L_1 = F(r_1) = 556$.

Таблица 15

a_j / x_i	4	6	10	5
0	2	4	3	5
10	6	2 $\boxed{6}$	4	4 $\boxed{4}$
15	3 $\boxed{4}$	5	5 $\boxed{10}$	6 $\boxed{1}$

Таблица 16

$d(x_i)$ \ $\bar{b}_k$	15	20	10	5	10
0	1	4	2	6	5
20	3 $\boxed{15}$	5	4	1 $\boxed{5}$	7
30	6	3 $\boxed{20}$	3 $\boxed{10}$	2	4

2-й шаг. Строим множество $W_2 = (W_1 \setminus r_1) \cup O(r_1)$. Так как $O(r_1) = \{(*, 0, 15)\}$, то

$$W_2 = \{B_1^2, B_2^2, B_3^2\},$$

$$B_1^2 = B_1^1 = (*, *, 0) = (10, 15, 0),$$

$$B_2^2 = B_2^1 = (*, *, 5) = (10, 10, 5),$$

$$B_3^2 = (*, 0, 15) = (10, 0, 15),$$

$$Q(B_1^2) = 569, \quad Q(B_2^2) = 631, \quad Q(B_3^2) = 553.$$

Находим $r_2 = B_3^2$, $Q(r_2) = 553$. Так как $Q(r_2) < L_1 = 556$, то необходимо сравнить величины $F(r_2)$ и L_1. Для вычисления $F(r_2)$ необходимо решить две транспортные задачи, условия которых задаются таблицами 17 и 18.

Таблица 17

x_i \ a_j	4	6	10	5
10	2	4	3 $\boxed{10}$	5
0	6	2	4	4
15	3 $\boxed{4}$	5 $\boxed{6}$	5	6 $\boxed{5}$

Таблица 18

$d(x_i)$ \ b_k	15	20	10	5	10
20	1 $\boxed{15}$	4	2	6	5
0	3	5	4	1	7
30	6	3 $\boxed{20}$	3 $\boxed{5}$	2 $\boxed{5}$	4

Транспортные затраты равны соответственно 102 и 110. Поэтому $F(r_2) = 562$. Сравниваем величины $F(r_2)$ и L_1. Так как $F(r_2) > L_1$, то полагаем $L_2 = L_1 = 556$ и переходим к 3-му шагу.

3-й шаг. Строим множество $W_3 = (W_2 \setminus r_2) \cup O(r_2)$. Поскольку $O(r_2) = \varnothing$, то

$$W_3 = \{B_1^3, B_2^3\},$$

$$B_1^3 = B_1^2 = (*, *, 0) = (10, 15, 0),$$

$$B_2^3 = B_2^2 = (*, *, 5) = (10, 10, 5),$$

$$Q(B_1^3) = 569, \quad Q(B_2^3) = 631.$$

Находим $r_3 = B_1^3$, $Q(r_3) = 569$. Так как $Q(r_3) = 569 > > L_2 = 556$, то план r_1, соответствующий величине L_2, и будет оптимальным с компонентами $x_1 = 0$, $x_2 = 10$, $x_3 = 15$ и суммарными затратами $F(r_1) = 556$. Объемы x_{ij} и $\bar{x}_{kl}$ транспортных перевозок заданы в таблицах 15 и 16.

§ 3. Решение многопродуктовых задач размещения

Многопродуктовая задача (задача A) размещения производства (см. § 3 гл. I) заключается в минимизации функции

$$\sum_{i=1}^{m} f_i(X_i) + \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{l=1}^{r} c_{ij}^l x_{ij}^l \qquad (3.1)$$

при ограничениях

$$\sum_{j=1}^{n} x_{ij}^{l} \leqslant x_{i}^{l}, \quad i=\overline{1, m}, \quad l=\overline{1, r}, \tag{3.2}$$

$$\sum_{i=1}^{m} x_{ij}^{l} = b_{j}^{l}, \quad j=\overline{1, n}, \quad l=\overline{1, r}, \tag{3.3}$$

$$X_i = (x_i^1, x_i^2, \ldots, x_i^r) \in \Gamma_i =$$

$$= \{A_{i1}, A_{i2}, \ldots, A_{is_i}\}, \quad i=\overline{1, m}, \tag{3.4}$$

$$x_{ij}^{l} \geqslant 0, \quad i=\overline{1, m}, \quad j=\overline{1, n}, \quad l=\overline{1, r}, \tag{3.5}$$

где $A_{ik}=(a_{ik}^1, a_{ik}^2, \ldots, a_{ik}^r)$ $(k=\overline{1, s_i})$, а параметры a_{ik}^l, b_j^l — целые положительные числа.

Из условий (3.2) и (3.3) следует, что

$$\sum_{i=1}^{m} X_i \geqslant B. \tag{3.6}$$

Здесь $B=(b^1, b^2, \ldots, b^r)=\sum_{j=1}^{n} B_j$, где $B_j=(b_j^1, b_j^2, \ldots, b_j^r)$. Вектор $r=(X_1, X_2, \ldots, X_m)$, компоненты которого удовлетворяют условиям (3.4) и (3.6), будем называть *вариантом размещения задачи* A.

В этих обозначениях задача A состоит в нахождении такого варианта размещения, который минимизирует функцию

$$F(X_1, X_2, \ldots, X_m) = \sum_{i=1}^{m} f_i(X_i) + \min_{S} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{l=1}^{r} c_{ij}^{l} x_{ij}^{l},$$

где множество S определяется условиями (3.2), (3.3), (3.5). Поэтому задача A принимает вид

$$F(X_1, X_2, \ldots, X_m) \to \min,$$

$$\sum_{i=1}^{m} X_i \geqslant B,$$

$$X_i \in \Gamma_i, \quad i=\overline{1, m}.$$

Прежде чем приступить к формулировке вспомогательной задачи и описанию алгоритма, введем некоторые определения. *Образом вектора* $r=(X_1, X_2, \ldots, X_m)$, где $X_i \in \Gamma_i$ $(i=\overline{1, m})$, по t-му продукту назовем вектор $r^t=(x_1^t, x_2^t, \ldots, x_m^t)$. Очевидно, что по образу $r^t = \left(a^t_{1t_1}, a^t_{2t_2}, \ldots, a^t_{mt_m}\right)$ можно однозначно восстановить прообраз $r = \left(A_{1t_1}, A_{2t_2}, \ldots, A_{mt_m}\right)$.

В качестве расширения множества задачи $\bar{A}$ и берется множество образов по какому-либо продукту.

Рассмотрим функцию $\sum_{i=1}^{m} g_i(X_i)$. Здесь

$$g_i(X_i) = f_i(X_i) + \sum_{j=1}^{n} d_j(X_i)/m,$$

где

$$d_j(X_i) = \sum_{l=1}^{r} d_j^l(x_i^l), \quad d_j^l(x_i^l) = \min \sum_{p=1}^{m} c_{pj}^l x_{pj}^l$$

при условиях

$$\sum_{p=1}^{m} x_{pj}^l = b_j^l,$$

$$0 \leqslant x_{pj}^l \leqslant \max_{1 \leqslant k \leqslant s_p} a_{pk}^l,$$

$$p = \overline{1, m}, \quad p \neq i, \quad 0 \leqslant x_{ij}^l \leqslant x_i^l.$$

Нетрудно показать, что

$$\sum_{i=1}^{m} g_i(X_i) \leqslant F(X_1, X_2, \ldots, X_m)$$

для любого плана $r = (X_1, X_2, \ldots, X_m)$ задачи A, т. е. $\sum_{i=1}^{m} g_i(X_i)$ можно взять в качестве миноранты.

Из предыдущих рассуждений вытекает, что вспомогательная задача $\bar{A}$ состоит в минимизации функции

$$Q(X_1, X_2, \ldots, X_m) = \sum_{i=1}^{m} g_i(X_i)$$

при условиях

$$\sum_{i=1}^{m} x_i^t \geqslant b^t,$$

$$x_i^t \in \left\{a_{i1}^t, a_{i2}^t, \ldots, a_{is_i}^t\right\}, \quad i = \overline{1, m},$$

$$X_i \in \Gamma_i, \quad i = \overline{1, m}.$$

Напомним, что здесь $\left(x_1^t, x_2^t, \ldots, x_m^t\right)$ — образ вектора $(X_1, X_2, \ldots, X_m)$. Алгоритм, описанный в § 3 гл. II, позволяет строить последовательность планов $r_1^t, r_2^t, \ldots$ задачи $\bar{A}$ в порядке неубывания функции $Q(X_1, \ldots, X_m)$. Если из этой последовательности удалить планы, прообразы которых не удовлетворяют условию (3.6), то останется последовательность $r_1, r_2, \ldots$ вариантов размещения такая, что

$$Q(r_k) = \min_{r \in R_k} Q(r), \quad k \geqslant 1,$$

$$R_k = R_{k-1} \setminus r_{k-1}, \quad R_1 = R,$$

где R — множество вариантов размещения.

Следовательно, алгоритм ψ решения многопродуктовой задачи размещения состоит в следующем.

Алгоритм ψ. На подготовительном этапе алгоритма выбираем l-й продукт, для которого достигается

$$\max_{1 \leqslant l \leqslant r} \min_{\Omega_l} \sum_{i=1}^{m} \omega_i \left(x_i^l\right).$$

Здесь $\omega_i\left(a_{ik}^l\right) = g_i\left(A_{ik}\right)$, Ω_l — множество векторов $\left(x_1^l, x_2^l, \ldots, x_m^l\right)$, определенное условиями

$$\sum_{i=1}^{m} x_i^l \geqslant b^l, \quad x_i^l \in \left\{a_{i1}^l, a_{i2}^l, \ldots, a_{is_i}^l\right\}, \quad i = \overline{1, m}.$$

Полагаем $L_0 = \infty$.

k-й шаг ($k \geqslant 1$). При помощи алгоритма φ находим план r_k^t и сравниваем числа $Q\left(r_k^t\right)$ и L_{k-1}. Если $Q\left(r_k^t\right) \geqslant L_{k-1}$, то работа алгоритма заканчивается и оптимальным вариантом размещения задачи A является вариант, соответствующий величине L_{k-1}. Если $Q\left(r_k^t\right) < L_{k-1}$, то проверяем, принадлежит ли прообраз r_k вектора r_k^t множеству R. При $r_k \notin R$ переходим к $(k+1)$-му шагу. При $r_k \in R$ вычисляем величину $F(r_k)$ и сравниваем числа $F(r_k)$ и L_{k-1}. Для $L_{k-1} \leqslant F(r_k)$ полагаем $L_k = L_{k-1}$ и переходим к $(k+1)$-му шагу. Для $L_{k-1} > F(r_k)$ полагаем $L_k = F(r_k)$ и переходим к $(k+1)$-му шагу.

Проиллюстрируем работу алгоритма на примере. Для простоты рассмотрим многопродуктовую задачу размещения без транспорта,

Пример. Заданы три пункта возможного размещения предприятий по производству трех видов продукции. Варианты специализации и соответствующие производственные затраты заданы в таблице 19 для первого пункта, в таблице 20 — для второго пункта, в таблице 21 — для третьего. Потребность в продукции задана вектором $B = (b^1, b^2, b^3) = (25, 15, 5)$.

Таблица 19

A_{1k}	Номер продукта			$f_1(X_1)$
	1	2	3	
A_{11}	0	0	0	0
A_{12}	5	10	5	150
A_{13}	10	10	0	180

Таблица 20

A_{2k}	Номер продукта			$f_2(X_2)$
	1	2	3	
A_{21}	0	0	0	0
A_{22}	10	5	15	150
A_{23}	15	15	0	200

Таблица 21

A_{3k}	Номер продукта			$f_3(X_3)$
	1	2	3	
A_{31}	0	0	0	0
A_{32}	5	0	15	130
A_{33}	15	0	5	170

В качестве вспомогательной задачи возьмем задачу минимизации функции

$$Q(X_1, X_2, X_3) = g_1(x_1^1) + g_2(x_2^1) + g_3(x_3^1)$$

при условиях

$$x_1^1 + x_2^1 + x_3^1 \geqslant 25,$$

$$x_1^1 \in \{0, 5, 10\}, \quad x_2^1 \in \{0, 10, 15\}, \quad x_3^1 \in \{0, 5, 15\}.$$

Здесь

$$g_1(x_1^1) = g_1(X_1), \quad g_2(x_2^1) = g_2(X_2), \quad g_3(x_3^1) = g_3(X_3).$$

Исходные данные для этой задачи задаются таблицей 22

Таблица 22

x	0	5	10	15
$g_1(x)$	0	150	180	—
$g_2(x)$	0	—	150	200
$g_3(x)$	0	130	—	170

Как и в §§ 1, 2, при построении последовательности планов нам понадобится информация, получаемая в процессе решения вспомогательной задачи. Представим ее в виде таблиц. В таблице 23 приведены значения функции $f_1(z) = g_1(z)$, в таблице 24 — значения функций $g_2(x_2^1) + f_1(z - x_2^1)$ и $f_2(z)$, а в таблице 25 — функций $g_3(x_3^1) + f_2(z - x_3^1)$ и $f_3(z)$.

Таблица 23

z	0	5	10	15	20	25
$f_1(z)$	0	150	180	—	—	—

Таблица 24

x_2^1 \ z	0	5	10	15	20	25
0	0	150	180	—	—	—
10	150	150	150	300	330	—
15	200	200	200	200	350	380
$f_2(z)$	0	150	150	200	330	380

Таблица 25

x_3^1 \ z	0	5	10	15	20	25
0	0	150	150	200	330	380
5	130	130	280	280	330	460
15	170	170	170	170	320	320
$f_3(z)$	0	130	150	170	320	320

1-й шаг. Строим множество всех 2-оптимальных планов:

$$W_1 = \{B_1^1,\ B_2^1,\ B_3^1\},$$

$$B_1^1 = (*,\ *,\ 0) = (10,\ 15,\ 0),$$

$$B_2^1 = (*,\ *,\ 5) = (10,\ 10,\ 5),$$

$$B_3^1 = (*,\ *,\ 15) = (0,\ 10,\ 15),$$

$$Q(B_1^1) = 380, \quad Q(B_2^1) = 450, \quad Q(B_3^1) = 320.$$

Находим план $r_1 = B_3^1$. Так как прообраз плана r_1 не является планом исходной многопродуктовой задачи, переходим ко 2-му шагу.

2-й шаг. Строим множество $W_2 = (W_1 \setminus r_1) \cup O(r_1)$. Так как

$$O(r_1) = \{(*,\ 0,\ 15),\ (*,\ 15,\ 15),\ (5,\ 10,\ 15),\ (10,\ 10,\ 15)\},$$

то

$$W_2 = \{B_1^2\ B_2^2,\ B_3^2,\ B_4^2,\ B_5^2,\ B_6^2\},$$

$$B_1^2 = B_1^1 = (*,\ *,\ 0) = (10,\ 15,\ 0),$$

$$B_2^2 = B_2^1 = (*,\ *,\ 5) = (10,\ 10,\ 5),$$

$$B_3^2 = (*,\ 0,\ 15) = (10,\ 0,\ 15),$$

$$B_4^2 = (*,\ 15,\ 15) = (0,\ 15,\ 15),$$

$$B_5^2 = (5,\ 10,\ 15), \quad B_6^2 = (10,\ 10,\ 15),$$

$$Q(B_1^2) = 380, \quad Q(B_2^2) = 450, \quad Q(B_3^2) = 350,$$

$$Q(B_4^2) = 370, \quad Q(B_5^2) = 470, \quad Q(B_6^2) = 490.$$

Находим план $r_2 = B_3^2$ и $Q(r_2) = 350$. Так как прообраз плана $r_2 = B_3^2 = (10, 0, 15)$, т. е. план

$$X_1 = A_{13} = \begin{pmatrix} 10 \\ 10 \\ 0 \end{pmatrix}, \quad X_2 = A_{21} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad X_3 = A_{33} = \begin{pmatrix} 15 \\ 0 \\ 5 \end{pmatrix},$$

удовлетворяет условиям задачи, т. е. при его реализации производится необходимое количество продукции всех видов, то он и является оптимальным.

§ 4. Решение межотраслевой задачи размещения

В § 4 гл. I приведены модели некоторых межотраслевых задач внутрирайонного размещения предприятий. В настоящем параграфе приводится алгоритм решения одной из них (задачи 4.2), которая состоит в определении для каждого из пунктов размещения $(i = \overline{1, m})$ такого объема производства t-го вида продукции x_{it} из заданного набора Γ_{it} и такого плана перевозок x_{ijt} $(i = \overline{1, m},\ j = \overline{1, n},\ t = \overline{1, v})$, чтобы суммарные производственно-транспортные затраты

$$\sum_{i=1}^{m} \sum_{t=1}^{v} \varphi_{it}(x_{it}) + \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{t=1}^{v} c_{ijt} x_{ijt} \tag{4.1}$$

были минимальными и удовлетворялся спрос b_{jt} $(j = \overline{1, n}, t = \overline{1, v})$ каждого потребителя при соблюдении ограничений на ресурсы.

Таким образом, в задаче A требуется найти минимум функции (4.1) при следующих ограничениях:

$$\sum_{j=1}^{n} x_{ijt} \leqslant x_{it}, \quad i = \overline{1, m}, \quad t = \overline{1, v}, \tag{4.2}$$

$$\sum_{i=1}^{m} x_{ijt} = b_{jt}, \quad j = \overline{1, n}, \quad t = \overline{1, v}, \tag{4.3}$$

$$\sum_{t=1}^{v} p_{tk}(x_{it}) \leqslant r_{ik}, \quad i = \overline{1, m}, \quad k = \overline{1, l}, \tag{4.4}$$

$$x_{it} \in \Gamma_{it} = \left\{a_{it}^1, \ldots, a_{it}^{s_{it}}\right\}, \quad i = \overline{1, m}, \quad t = \overline{1, v}, \tag{4.5}$$

$$x_{ijt} \geqslant 0, \quad i = \overline{1, m}, \quad j = \overline{1, n}, \quad t = \overline{1, v}. \tag{4.6}$$

Задача (4.1)—(4.6) является *задачей дискретного программирования с блочной структурой*. В частном случае, когда $v = 1$ и отсутствует ограничение (4.4), она превращается в однопродуктовую производственно-транспортную задачу размещения, рассмотренную в § 1.

Обозначим через $F(X_1, X_2, \ldots, X_m)$ сумму

$$\sum_{i=1}^{m} \sum_{t=1}^{v} \varphi_{it}(x_{it}) + \sum_{t=1}^{v} T(X_t).$$

Здесь

$$X_t = (x_{1t}, x_{2t}, \ldots, x_{mt}), \quad t = \overline{1, v},$$

а $T(X_t) = \min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ijt} x_{ijt}$ при условиях

$$\sum_{j=1}^{n} x_{ijt} = x_{it}, \quad i = \overline{1, m},$$

$$\sum_{i=1}^{m} x_{ijt} = b_{jt}, \quad j = \overline{1, n},$$

$$x_{ijt} \geqslant 0, \quad i = \overline{1, m}, \quad j = \overline{1, n}.$$

Таким образом, задача A принимает вид

$$F(X_1, X_2, \ldots, X_v) \to \min,$$

$$\sum_{i=1}^{m} x_{it} = b_t, \quad t = \overline{1, v},$$

$$x_{it} \in \Gamma_{it}, \quad i = \overline{1, m}, \quad t = \overline{1, v},$$

$$\sum_{t=1}^{v} p_{tk}(x_{it}) \leqslant r_{ik}, \quad i = \overline{1, m}, \quad k = \overline{1, l},$$

где $b_t = \sum_{j=1}^{n} b_{jt} \quad (t = \overline{1, v})$.

В качестве вспомогательной задачи $\bar{A}$ возьмем задачу минимизации функции

$$Q(X_1, X_2, \ldots, X_v) = \sum_{i=1}^{m} \sum_{t=1}^{v} g_{it}(x_{it})$$

при условиях

$$\sum_{i=1}^{m} x_{it} = b_t, \quad t = \overline{1, v},$$

$$x_{it} \in \Gamma_{it}, \quad i = \overline{1, m}, \quad t = \overline{1, v}.$$

Здесь

$$g_{it}(x_{it}) = \varphi_{it}(x_{it}) + u_{it}(x_{it}),$$

где

$$u_{it}(x_{it}) = \min \sum_{j=1}^{n} c_{ijt} x_{ijt}$$

при условиях

$$\sum_{j=1}^{n} x_{ijt} = x_{it}, \quad 0 \leqslant x_{ijt} \leqslant b_{jt}, \quad j = \overline{1, n}.$$

Нетрудно показать, что функция $Q(r)$ является минорантой функции $F(r)$ на множестве планов задачи $\bar{A}$.

Очевидно, задача $\bar{A}$ имеет решение, если разрешима каждая из задач $\bar{A}_t$ $(t = \overline{1, v})$, состоящая в минимизации функции

$$Q_t(X_t) = \sum_{i=1}^{m} g_{it}(x_{it})$$

при ограничениях

$$\sum_{i=1}^{m} x_{it} = b_t, \quad x_{it} \in \Gamma_{it}, \quad i = \overline{1, m}.$$

В дальнейшем будем предполагать, что каждая из задач $\bar{A}_t$ $(t = \overline{1, v})$ имеет решение.

Ранее было показано (см. § 1 гл. II), как можно строить последовательность планов $X_t^1, X_t^2, \ldots$ задачи $\bar{A}_t$ в порядке неубывания функций $Q_t(X_t)$ таких, что

$$Q_t(X_t^k) = \min_{X \in R_t^k} Q_t(X), \quad k = 1, 2, \ldots,$$

где $R_t^k = R_t^{k-1} \setminus X_t^{k-1}$ $(k = 2, 3, \ldots)$, R_t^1 — множество планов задачи $\bar{A}_t$.

Имея последовательности планов $X_t^1, X_t^2, \ldots$ $(t = \overline{1, v})$ задач $\bar{A}_t$ $(t = \overline{1, v})$, можно, пользуясь алгоритмом построения систем представителей (см. § 4 гл. II), строить последовательность планов $r_1, r_2, \ldots$ задачи $\bar{A}$ в порядке неубывания миноранты $Q(r)$. Для этого нужно положить

$$\Gamma_1 = \left\{Q_1(X_1^1),\ Q_1(X_1^2),\ \ldots,\ Q_1\left(X_1^{|R_1^1|}\right)\right\},$$
$$\Gamma_2 = \left\{Q_2(X_2^1),\ Q_2(X_2^2),\ \ldots,\ Q_2\left(X_2^{|R_2^1|}\right)\right\},$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$\Gamma_v = \left\{Q_v(X_v^1),\ Q_v(X_v^2),\ \ldots,\ Q_v\left(X_v^{|R_v^1|}\right)\right\}.$$

Тогда для с.п. $c = (j_1, j_2, \ldots, j_v)$ функция $Q(c)$ определяется следующим образом:

$$Q(c) = \sum_{t=1}^{v} Q_t\left(X_t^{j_t}\right).$$

Соответствующий с. п. $c = (j_1, j_2, \ldots, j_v)$ план задачи $\bar{A}$ будем обозначать

$$r_c = \left\{X_1^{j_1},\ X_2^{j_2},\ \ldots,\ X_v^{j_v}\right\}.$$

Тогда последовательности систем представителей $c_1, c_2, \ldots$, полученной с помощью изложенного в § 4 гл. II алгоритма, будет соответствовать последовательность $r_1, r_2, \ldots$ планов задачи $\bar{A}$, причем $r_1 = r_{c_1}$, $r_2 = r_{c_2}, \ldots$

Замечание. При построении последовательности $c_1, c_2, \ldots$ нет необходимости (да и нецелесообразно) полностью находить множества

$$\Gamma_t = \left\{Q_t(X_t^1),\ Q_t(X_t^2),\ \ldots,\ Q_t\left(X_t^{|R_t^1|}\right)\right\}.$$

В процессе построения последовательности систем представителей планы $X_t^1, X_t^2, \ldots$ задач $\bar{A}_t$ нужно строить по мере необходимости.

Алгоритм ψ решения межотраслевой задачи размещения состоит в следующем.

Алгоритм ψ. Полагаем $L_0 = \infty$.

k-й шаг $(k \geqslant 1)$. Находим план r_k задачи $\bar{A}$ и сравниваем числа $Q(r_k)$ и L_{k-1}. При $Q(r_k) \geqslant L_{k-1}$ работа алгоритма заканчивается и решением задачи A является план, соответствующий величине L_{k-1}. При $Q(r_k) < L_{k-1}$ проверяем, удовлетворяет ли план r_k условиям (4.4). Если не удовлетворяет, то полагаем $L_k = L_{k-1}$ и переходим к $(k+1)$-му шагу. Если удовлетворяет, то вычисляем величину $F(r_k)$ и сравниваем числа $F(r_k)$ и L_{k-1}. Для $F(r_k) \geqslant L_{k-1}$ полагаем $L_k = L_{k-1}$ и переходим к $(k+1)$-му шагу. Для $F(r_k) < L_{k-1}$ полагаем $L_k = F(r_k)$ и переходим к $(k+1)$-му шагу.

Для иллюстрации работы алгоритма рассмотрим пример.

Пример. Заданы три пункта размещения $(m = 3)$. В каждом из них возможно размещение предприятий, выпускающих два вида продукции $(v = 2)$ и потребляющих один вид ресурсов $(l = 1)$. Возможные мощности предприятий, затраты, связанные с размещением соответствующих мощностей, и потребности в ресурсе заданы для первого продукта в таблице 26, а для второго — в таблице 27.

Таблица 26

Возможные пункты производства	Возможные объемы производства a_{i1}^h	Приведенные затраты $\varphi_{i1}(a_{i1}^h)$	Потребность в ресурсе $p_{11}(a_{i1}^h)$
1	0 3 5	0 20 30	0 4 6
2	0 2 4	0 10 30	0 3 5
3	0 3 5	0 5 15	0 4 6

Известно наличие ресурса в каждом из пунктов размещения, т. е. заданы величины

$$r_{11} = 10, \quad r_{21} = 10, \quad r_{31} = 6.$$

Таблица 27

Возможные пункты производства	Возможные объемы производства a_{i2}^h	Приведенные затраты $\varphi_{i2}\left(a_{i2}^h\right)$	Потребность в ресурсе $p_{21}\left(a_{i2}^h\right)$
1	0 6	0 10	0 3
2	0 6 10	0 20 35	0 3 5
3	0 4 10	0 5 40	0 2 5

Следовательно, по ресурсам должны выполняться условия:

$$p_{11}(x_{11}) + p_{21}(x_{12}) \leqslant 10,$$

$$p_{11}(x_{21}) + p_{21}(x_{22}) \leqslant 10,$$

$$p_{11}(x_{31}) + p_{21}(x_{32}) \leqslant 6.$$

Кроме того, заданы четыре пункта потребления готовой продукции ($n = 4$), размер спроса каждого потребителя в видах продукции и матрицы транспортных затрат по перевозке единицы продукции соответствующего вида от каждого пункта возможного размещения предприятий до каждого потребителя. Эти исходные данные приведены в таблицах 28, 29.

Таблица 28

Возможные пункты производства	Пункты потребления			
	1	2	3	4
	Объемы потребления b_{j1}			
	5	2	2	3
1	2	5	9	2
2	3	4	6	7
3	2	6	5	8

Таблица 29

Возможные пункты производства	Пункты потребления			
	1	2	3	4
	Объемы потребления b_{j2}			
	3	6	5	2
1	6	3	3	6
2	2	8	4	4
3	4	7	1	5

На предварительном этапе работы алгоритма строим функции $g_{it}(x_{it})$ (см. таблицы 30, 31). Пользуясь алгоритмом, изложенным в § 1 гл. II, строим

Таблица 30

x	0	2	3	4	5
$g_{11}(x)$	0	—	26	—	40
$g_{21}(x)$	0	16	—	42	—
$g_{31}(x)$	0	—	11	—	25

Таблица 31

x	0	4	6	10
$g_{12}(x)$	0	—	28	—
$g_{22}(x)$	0	—	43	69
$g_{32}(x)$	0	9	—	67

последовательности планов X_i^1, X_i^2. В нашем случае

$$X_1^1=(5,\ 2,\ 5),\qquad X_1^2=(5,\ 4,\ 3),\qquad X_1^3=(3,\ 4,\ 5),$$
$$Q(X_1^1)=81,\qquad Q(X_1^2)=93,\qquad Q(X_1^3)=103,$$
$$X_2^1=(6,\ 6,\ 4),\qquad X_2^2=(6,\ 0,\ 10),\qquad X_2^3=(6,\ 10,\ 0),$$
$$Q(X_2^1)=75,\qquad Q(X_2^2)=95,\qquad Q(X_2^3)=97.$$

Следовательно,

$$\Gamma_1=\{81,\ 93,\ 103\},\quad \Gamma_2=\{75,\ 95,\ 97\},$$
$$\alpha_1^1=1,\quad \alpha_1^2=2,\quad \alpha_1^3=3,\quad \alpha_2^1=1,\quad \alpha_2^2=2,\quad \alpha_2^3=3.$$

Полагаем $L_0=\infty$.

1-й шаг. Строим множество $W_1=\{c_1^1\}$, где $c_1^1=(1,\ 1)$. Отсюда

$$c_1=c_1^1,\quad Q(c_1)=156.$$

Следовательно,

$$r_1=r_{c_1}=(X_1^1,\ X_2^1).$$

Так как $Q(r_1)<L_0$ и план r_1 не удовлетворяет ограничениям по ресурсам, то полагаем $L_1=\infty$ и переходим ко 2-му шагу.

2-й шаг. Строим множество $W_2=(c_1^2,\ c_2^2)$, где $c_1^2=(2,\ 1)$, $c_2^2=(1,\ 2)$. Находим

$$Q(c_1^2)=168,\quad Q(c_2^2)=176.$$

Следовательно,

$$c_2=c_1^2,\quad r_2=r_{c_2}=(X_1^2,\ X_2^1).$$

Так как план $r_2(X_1^2=(5,\ 4,\ 3),\ X_2^1=(6,\ 6,\ 4))$ удовлетворяет ограничениям по ресурсам, вычисляем величину $F(r_2)$. Для этого необходимо решить транспортные задачи, информация для которых задана таблицами 32 и 33 (соответственно для первого и для второго продукта). Получаем

$$F(r_2)=30+30+5+36+10+20+5+40=176.$$

Таблица 32

x_{i1} \ b_{j1}	5	2	2	3
5	2 $\boxed{2}$	5	9	2
4	3 $\boxed{2}$	4 $\boxed{2}$	6	7 $\boxed{3}$
3	2 $\boxed{1}$	6	5 $\boxed{2}$	8

Таблица 33

x_{i2} \ b_{j2}	3	6	5	2
6	6	3 $\boxed{6}$	3	6
6	2 $\boxed{3}$	8	4 $\boxed{1}$	4 $\boxed{2}$
4	4	7	1 $\boxed{4}$	3

Так как $F(r_2) < L_1 = \infty$, то $L_2 = 171$ и переходим к 3-му шагу.

3-й шаг. Строим множество $W_3 = (c_1^3, c_2^3, c_3^3)$, где

$$c_1^3 = c_2^2 = (1, 2), \qquad c_2^3 = (3, 1), \qquad c_3^3 = (2,2),$$

$$Q(c_1^3) = 176, \quad Q(c_2^3) = 178, \quad Q(c_3^3) = 188.$$

Отсюда $c_3 = c_1^3$, $Q(c_3) = 176$. Следовательно,

$$r_3 = r_{c_3} = \{X_1^1, X_2^2\}.$$

Так как $Q(r_3) > L_2$, то работа алгоритма заканчивается. Решением задачи является план, соответствующий величине L_2, т. е. план $r_2 = (X_1^2, X_2^1)$, где $X_1^2 = (5, 4, 3)$, $X_2^1 = (6, 6, 4)$. Следовательно, в оптимальном варианте размещения объем производства первого вида продукции составляет 5 единиц для первого пункта размещения, 4 — для второго, 3 — для третьего, а для второго продукта эти величины равны 6, 6 и 4 единицам соответственно.

Транспортные перевозки, соответствующие оптимальному варианту размещения, заданы в таблицах 32, 33.

§ 5. Вычислительные аспекты и опыт решения практических задач

Метод построения последовательности планов реализован на ЭВМ и применялся для решения многочисленных практических производственно-транспортных задач размещения производства. Решение производственно-транспортных задач размещения связано с направленным перебором вариантов r_1, r_2, ... размещения до выполнения неравенства

$$L_{k-1} = \min\{F(r_1), \ldots, F(r_{k-1})\} \leqslant Q(r_k).$$

Для нахождения варианта r_k необходимо построить множество W_k. Это множество получается из множества W_{k-1} путем замены плана r_{k-1} множеством планов $O(r_{k-1})$ по правилу

$$W_k = (W_{k-1} \setminus r_{k-1}) \cup O(r_{k-1}).$$

Следовательно,

$$W_k = \{W_1 \cup O(r_1) \cup \ldots \cup O(r_{k-1})\} \setminus \{r_1, r_2, \ldots, r_{k-1}\}.$$

Итак, для нахождения r_k необходимо запомнить планы r_1, r_2, ..., r_{k-1} и множество W_1. На основании этой информации можно построить множество W_k и, следовательно, найти следующий план r_k. Ясно, что с возрастанием числа итераций объем запоминаемой информации растет лишь линейно.

Метод позволяет находить также и приближенное решение задачи. При этом (что очень важно в практике) всегда можно оценить сверху отклонение лучшего из полученных вариантов от оптимального, так как после s шагов работы алгоритма это отклонение не будет превышать величины $L_s - Q(r_s)$. Поскольку

$$L_l - Q(r_l) \leqslant L_l - Q(r_{l-1}),$$

то это отклонение на каждом шаге не возрастает.

Практические задачи размещения производства (как, впрочем, и другие экономические задачи) таковы, что математическая формализация их, как правило, отражает не все факторы, влияющие на выбор оптимального варианта, а только те, которые считаются основными и которые удается формализовать. Таким образом, математическая модель задачи оптимального размещения производства неполно отражает истинное положение вещей и представляет собой так называемую упрощающую идеализацию реальной задачи. Если учесть неизбежную неточность исходных данных, то из вышеизложенного следует, что оптимальный вариант размещения производства, полученный как результат решения математической модели (математический оптимальный вариант), может оказаться с экономической точки зрения не самым лучшим. Естественно, однако, считать, что экономический оптимальный вариант отличается от математического незначительно, так как в противном случае выбранный критерий оптимальности не выполнял бы своего назначения. Поэтому наряду с математическим оптимальным вариантом следует выдавать еще и ряд других, близких к нему по критерию оптимальности. Из этих вариантов специалист, принимающий решение, выбирает наилучший с учетом игнорированных в модели условий, практического опыта, интуиции и т. п.

Изложенные в предыдущих параграфах алгоритмы позволяют находить не только оптимальный вариант, но и ряд следующих за ним по значению целевой функции, т. е. строить такую последовательность планов $r_1^*, r_2^*, \ldots$ задачи размещения, что

$$Q(r_k^*) = \min_{r \in P_k} F(r),$$

где $P_k = P_{k-1} \setminus r_{k-1}^*$ $(k=2, 3, \ldots)$, P_1 — множество всех планов задачи.

Например, если на k-м шаге метода ψ (см. § 1) найден оптимальный план r_1^*, т. е. выполняется критерий оптимальности

$$Q(r_k) \geqslant L_{k-1} = \min\{F(r_1), \ldots, F(r_{k-1})\} = F(r_1^*),$$

то для нахождения r_2^* следует положить

$$L_k = \min_{r \in \{r_1, \ldots, r_{k-1}\} \setminus r_1^*} F(r)$$

и работу метода вновь продолжить до выполнения критерия оптимальности. Действуя подобным образом, можно построить также планы r_3^*, r_4^*, ...

Опыт решения задач размещения в ряде организаций показал эффективность предложенного метода. В НИИЭМП при Госплане БССР решались реальные задачи перспективного развития отраслей народного хозяйства БССР (машиностроение, строительная индустрия, легкая и пищевая промышленность и др.). Среднее время решения задач с параметрами $m = 30$, $n = 60$, $s_i = 8$ составляет около 30 минут машинного времени на ЭВМ Минск-22. Статистика решения 50 различных задач размещения размерности $m = 10 \div 20$, $n = 30 \div 40$, $s_i = 4 \div 6$ показала, что время решения этих задач колеблется от 2 до 20 минут. При решении таких задач на ЭВМ Минск-32 время счета в 4—5 раз меньше. Накоплен определенный опыт успешного решения задач размещения и на ЭВМ Урал-14 (ГлавНИВЦ Госплана УССР). Например, решение задач размерности $m = 18$, $n = 100$, $s_i = 4$ требует около 30 минут машинного времени ЭВМ Урал-14. (Для сравнения заметим, что решение тех же задач по программе, реализующей метод ветвей и границ, требует, как правило, в 2—3 раза больше машинного времени.)

Накоплен определенный опыт решения задач размещения с двойным транспортом (см. § 2) для различных отраслей народного хозяйства. Так, задачи с параметрами $m = 18$, $n = 60$, $l = 25$, $s_i = 8$ были решены в среднем за 35 минут на ЭВМ Минск-22.

Следует отметить, что в большинстве случаев при решении задач размещения методом построения последовательности планов основное время тратится на доказательство оптимальности плана, как правило, получающегося на первых шагах работы метода.

ГЛАВА IV

РЕШЕНИЕ ЗАДАЧ ЦЕЛОЧИСЛЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Важнейшим классом задач математического программирования являются задачи целочисленного линейного программирования (ЦЛП). К ним сводятся многие практические задачи оптимизации, в частности большинство известных комбинаторных задач. В настоящей главе на основе общей схемы метода построения последовательности планов излагаются алгоритмы решения задач ЦЛП, при этом особое внимание уделяется важному подклассу задач, когда переменные принимают только значения 0 или 1.

§ 1. Задача целочисленного линейного программирования

Рассмотрим задачу A максимизации линейной формы

$$F(X)=\sum_{j=1}^{n} c_j x_j \tag{1.1}$$

при условиях

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i, \quad i=\overline{1, m}, \tag{1.2}$$

$$0 \leqslant x_j \leqslant d_j, \quad j=\overline{1, n}, \tag{1.3}$$

$$x_j \text{ — целое число}, \quad j=\overline{1, n}, \tag{1.4}$$

где параметры $b_i, d_j, c_j > 0$, $a_{ij} \geqslant 0$ — целые числа *).

*) Случай $c_j < 0$ сводится к рассматриваемому заменой $\bar{x}_j = d_j - x_j$. Коэффициенты a_{ij} предполагаются неотрицательными для упрощения изложения (см. [55]).

Если все условия (1.2) сложить с весами $u_1, u_2, \ldots$ $\ldots, u_m$, то получим

$$\sum_{j=1}^{n} a_j x_j \leqslant b,$$

где

$$a_j = \sum_{i=1}^{m} a_{ij} u_i, \quad b = \sum_{i=1}^{m} b_i u_i.$$

Ясно, что множество планов, определяемое этим «суррогатным» ограничением и условиями (1.3), (1.4), содержит в себе множество планов исходной задачи.

В дальнейшем рассматривается случай, когда в качестве весов выбираются числа

$$u_i = 0, \quad i \neq t, \quad u_t = 1,$$

где t — номер, для которого достигается $\min\limits_{1 \leqslant i \leqslant m} L_i$, где $L_i = \max\limits_{X \in \Omega_i} F(X)$. Здесь Ω_i — множество, заданное ограничениями (1.3) и

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i. \tag{1.5}$$

Нахождение L_i, т. е. максимума функции $F(X)$ на множестве Ω_i, не представляет трудностей, так как справедлива следующая

Теорема. *Пусть*

$$\frac{c_1}{a_{i1}} \geqslant \frac{c_2}{a_{i2}} \geqslant \ldots \geqslant \frac{c_n}{a_{in}}. \tag{1.6}$$

Тогда оптимальный план $X^0 = (x_1^0, x_2^0, \ldots, x_n^0)$ *задачи максимизации функции* $F(X)$ *на множестве* Ω_i *определяется по формулам*

$$\begin{gathered} x_j^0 = \begin{cases} d_j, & \text{если } j \leqslant r_0, \\ 0, & \text{если } j > r_0 + 1, \end{cases} \\ x_{r_0+1}^0 = \frac{b_i - \sum\limits_{j \leqslant r} a_{ij} d_j}{a_{i,\, r_0+1}}, \end{gathered} \tag{1.7}$$

где $r_0 = \max\left\{ r \,\middle|\, \sum\limits_{j \leqslant r} a_{ij} d_j \leqslant b_i \right\}$.

Доказательство. Случай $\sum\limits_{j=1}^{n} a_{ij} d_j \leqslant b_i$ тривиален, так как $x_j^0 = d_j$ $(j = \overline{1, n})$.

Если $\sum\limits_{j=1}^{n} a_{ij}d_j > b_i$, то оптимальный план задачи принадлежит множеству

$$P' = \left\{X \mid \sum_{j=1}^{n} a_{ij}x_j = b_i,\ 0 \leqslant x_j \leqslant d_j,\ j = \overline{1, n}\right\}.$$

Поэтому для доказательства теоремы достаточно показать, что для любого плана $X' = (x_1', x_2', \ldots, x_n') \in P'$ выполняется неравенство $F(X^0) \geqslant F(X')$, где $X^0 = (x_1^0, x_2^0, \ldots, x_n^0)$ — план, определяемый формулами (1.7).

Мы имеем $F(X^0) - F(X') = \sum\limits_{j=1}^{n} c_j \Delta x_j$, где $\Delta x_j = x_j^0 - x_j'$ $(j = \overline{1, n})$. Здесь $\Delta x_j \geqslant 0$ для $j = \overline{1, r_0}$ и $\Delta x_j \leqslant 0$ для $j = \overline{r_0 + 2, n}$ $(0 \leqslant r_0 \leqslant n - 1)$.

Возможны два случая:

1) $\Delta x_{r_0+1} \geqslant 0$. Тогда

$$F(X^0) - F(X') = \sum_{j=1}^{r_0+1} c_j \Delta x_j - \sum_{j=r_0+2}^{n} c_j |\Delta x_j|.$$

Так как справедливы неравенства (1.6), то

$$c_j \geqslant \frac{c_{r_0+1}}{a_{i, r_0+1}} a_{ij}, \quad j = \overline{1, r_0 + 1},$$

$$c_j \leqslant \frac{c_{r_0+1}}{a_{i, r_0+1}} a_{ij}, \quad j = \overline{r_0 + 2, n}.$$

Следовательно,

$$F(X^0) - F(X') \geqslant$$

$$\geqslant \frac{c_{r_0+1}}{a_{i, r_0+1}} \left(\sum_{j=1}^{r_0+1} a_{ij} \Delta x_j - \sum_{j=r_0+2}^{n} a_{ij} |\Delta x_j| \right) =$$

$$= \frac{c_{r_0+1}}{a_{i, r_0+1}} \sum_{j=1}^{n} a_{ij} \Delta x_j = \frac{c_{r_0+1}}{a_{i, r_0+1}} \left(\sum_{j=1}^{n} a_{ij} x_j^0 - \sum_{j=1}^{n} a_{ij} x_j' \right) = 0,$$

что и требовалось доказать.

2) $\Delta x_{r_0+1} < 0$. Здесь доказательство проводится аналогично, с той лишь разницей, что $1 \leqslant r_0 \leqslant n-1$. Если бы $r_0 = 0$, то $a_{i1}d_1 > b_i$, т. е.

$$x_1^0 = b_i/a_{i1}, \quad x_j^0 = 0, \quad j = \overline{2, n},$$

и, следовательно, $\Delta x_1 = 0$. Доказательство теоремы закончено.

Пример 1. Максимизировать функцию

$$6x_1 + 4x_2 + x_3$$

при ограничениях

$$x_1 + 2x_2 + 3x_3 \leqslant 5,$$

$$0 \leqslant x_j \leqslant 2, \quad j = 1, 2, 3.$$

В нашем случае переменные упорядочены нужным образом, так как $6/1 > 4/2 > 1/3$.

Из теоремы следует, что для нахождения оптимального плана этой задачи надо последовательно присваивать переменным максимально возможные значения. В данном случае можно положить $x_1 = 2$, $x_2 = 3/2$, $x_3 = 0$. Следовательно, значение функции для оптимального плана равно 18.

В качестве вспомогательной задачи $\bar{A}$ возьмем задачу максимизации функции

$$F(X) = Q(X)$$

на множестве, задаваемом ограничениями

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i,$$

$$0 \leqslant x_j \leqslant d_j, \quad j = \overline{1, n},$$

$$x_j \text{ — целое число}, \quad j = \overline{1, n}.$$

Введем понятие h-оптимального плана задачи $\bar{A}$ следующим образом. План $\bar{X} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ задачи $\bar{A}$ будем называть h-оптимальным и обозначать $(*, *, \ldots, \bar{x}_{h+1}, \ldots, \bar{x}_n)$, если

$$F(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n) =$$
$$= c_{h+1}\bar{x}_{h+1} + \ldots + c_n\bar{x}_n + f_h\left(b_i - \sum_{j=h+1}^{n} a_{ij}\bar{x}_j\right).$$

Здесь $f_h(z)$ — оптимальное значение линейной формы задачи $\bar{A}$, в которой индекс n и параметр b_t заменены соответственно на h и z. Таким образом, $\bar{X} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ — h-оптимальный план задачи $\bar{A}$, если $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_h)$ — оптимальный план задачи максимизации линейной формы

$$\sum_{j=1}^{h} c_j x_j$$

при условиях

$$\sum_{j=1}^{n} a_{tj} x_j \leqslant b_t - \sum_{j=h+1}^{n} a_{tj} \bar{x}_j,$$

$$0 \leqslant x_j \leqslant d_j, \quad j = \overline{1, h},$$

$$x_j \text{ — целое число}, \quad j = \overline{1, n}.$$

Используя понятие h-оптимального плана задачи $\bar{A}$, можно следующим образом описать алгоритм φ построения последовательности планов $r_t^1, r_t^2, \ldots$ задачи $\bar{A}$ в порядке невозрастания функции (1.1).

Алгоритм φ. 1-й шаг. Среди множества W_t^1 $(n-1)$-оптимальных планов

$$(*, *, \ldots, *, l), \quad l = 0, 1, \ldots, \min\left(d_n, \left[\frac{b_t}{a_{tn}}\right]\right),$$

задачи $\bar{A}$ находим такой план

$$r_t^1 = \left(*, *, \ldots, *, s_n^1\right) = \left(s_1^1, s_2^1, \ldots, s_n^1\right),$$

что

$$F\left(r_t^1\right) = \max_{r \in W_t^1} F(r).$$

k-й шаг $(k = 2, 3, \ldots)$. Множество планов W_t^{k-1} преобразуем в множество W_t^k по следующему правилу. Оставляем без изменения все планы множества W_t^{k-1}, кроме плана

$$r_t^{k-1} = \left(*, *, \ldots, *, s_p^{k-1}, \ldots, s_n^{k-1}\right) = \left(s_1^{k-1}, s_2^{k-1}, \ldots, s_n^{k-1}\right),$$

который заменяем системой планов

$$\left(*,\ *,\ \ldots,\ *,\ s_{q-1},\ s_q^{k-1},\ \ldots,\ s_p^{k-1},\ \ldots,\ s_n^{k-1}\right),$$

где

$$q=2,\ 3,\ \ldots,\ p,$$

$$s_{q-1}=0,\ 1,\ \ldots,\ s_{q-1}^{k-1}-1,\ s_{q-1}^{k-1}+1,\ \ldots$$

$$\ldots,\ \min\left(d_{q-1},\ \left[\frac{b_t-\sum\limits_{j=q}^{n} a_{tj}s_j^{k-1}}{a_{t,\,q-1}}\right]\right).$$

Далее, находим такой план r_t^k, что $F\left(r_t^k\right)=\max\limits_{r\in W_t^k} F(r)$.

Обоснование того, что изложенный алгоритм строит требуемую последовательность планов, аналогично ранее приведенному (см. § 3 гл. II). Так как в данном случае мажоранта совпадает с целевой функцией, то алгоритм решения задачи A состоит в построении последовательности планов задачи $\bar{A}$ в порядке невозрастания функции $F(X)$ до получения плана задачи A, который и будет оптимальным.

Проиллюстрируем работу алгоритма на примере.

Пример 2. Необходимо минимизировать функцию

$$F(X)=6x_1+4x_2+x_3$$

при условиях

$$x_1+2x_2+3x_3\leqslant 5,$$

$$2x_1+x_2+x_3\leqslant 4,$$

$$0\leqslant x_j\leqslant 2,\quad x_j \text{ — целое число},\quad j=1,\ 2,\ 3.$$

На предварительном этапе находим $L_t=\min\limits_{i=1,2} L_i$. В данном случае $t=2$, $L_2=\min\{18,\ 14\}=14$. Следовательно, вспомогательная задача $\bar{A}$ состоит в максимизации функции

$$F(X)=6x_1+4x_2+x_3$$

при условиях

$$2x_1+x_2+x_3\leqslant 4,$$

$$x_j\in\{0,\ 1,\ 2\},\quad j=1,\ 2,\ 3.$$

Для удобства представим задачу $\bar{A}$ в виде задачи $\bar{A}_1$ максимизации функции

$$Q(X) = g_1(x_1) + g_2(x_2) + g_3(x_3)$$

при условиях

$$x_1 + x_2 + x_3 \leqslant 4,$$

$$x_1 \in \{0, 2, 4\}, \quad x_2 \in \{0, 1, 2\}, \quad x_3 \in \{0, 1, 2\},$$

где функции $g_i(x_i)$ заданы таблицей 1. Очевидно, что задачи $\bar{A}_1$ и $\bar{A}$ эквивалентны. Действительно, плану $\bar{x}_1$, $\bar{x}_2$, $\bar{x}_3$ задачи $\bar{A}_1$ соответствует план $x_1 = \bar{x}_1/2$, $x_2 = \bar{x}_2$, $x_3 = \bar{x}_3$ задачи $\bar{A}$.

Как и в предыдущих параграфах, при построении последовательности планов нам понадобится информация, получаемая в процессе решения задачи $\bar{A}_1$ методом функциональных уравнений динамического программирования. Представим ее в виде таблиц. В таблице 2 приведены значения функции $f_1(z) = g_1(z)$, в таблице 3 приведены значения функций $g_2(x_2) + f_1(z - x_2)$ и $f_2(z)$, а в таблице 4 — функций $g_3(x_3) + f_2(z - x_3)$ и $f_3(z)$.

Таблица 1

x	0	1	2	4
$g_1(x)$	0	—	6	12
$g_2(x)$	0	4	8	—
$g_3(x)$	0	1	2	—

Таблица 2

z	0	1	2	3	4
$f_1(z)$	0	0	6	6	12

Таблица 3

x_2 \ z	0	1	2	3	4
0	0	0	6	6	12
1	—	4	4	10	10
2	—	—	8	8	14
$f_2(z)$	0	4	8	10	14

Таблица 4

x_3 \ z	0	1	2	3	4
0	0	4	8	10	14
1	—	1	5	9	11
2	—	—	2	6	10
$f_3(z)$	0	4	8	10	14

1-й шаг. Строим множество W_2^1 всех 2-оптимальных планов

$$W_2^1 = \{B_1^1, B_1^2, B_1^3\} = \{(*, *, 0), (*, *, 1), (*, *, 2)\}.$$

Вычисляем значения целевой функции:

$$F(B_1^1) = 14, \quad F(B_1^2) = 11, \quad F(B_1^3) = 10.$$

Среди планов множества W_2^1 находим лучший план r_2^1. В данном случае

$$r_2^1 = B_1^1 = (*, *, 0) = (2, 2, 0),$$

т. е. $\bar{x}_1 = 2$, $\bar{x}_2 = 2$, $\bar{x}_3 = 0$.

Этому плану соответствует план $x_1 = 1$, $x_2 = 2$, $x_3 = 0$ задачи $\bar{A}$. Так как он удовлетворяет условиям исходной задачи, то он и является оптимальным планом задачи. Следовательно, в оптимальном плане задачи

$$x_1 = 1, \quad x_2 = 2, \quad x_3 = 0, \quad F(r_2^1) = 14.$$

§ 2. Задача целочисленного линейного программирования с булевыми переменными

Рассмотрим задачу A максимизации линейной формы

$$F(X)=\sum_{j=1}^{n} c_j x_j \tag{2.1}$$

при условиях

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \quad i=\overline{1, m}, \tag{2.2}$$

$$x_j=0 \quad \text{или} \quad 1, \quad j=\overline{1, n}. \tag{2.3}$$

Будем предполагать, что $c_j \geqslant 0$. Этого всегда можно добиться заменой переменных $\bar{x}_j = 1 - x_j$, если $c_j < 0$, и $\bar{x}_j = x_j$, если $c_j \geqslant 0$. Обозначим через s_i максимум суммы $\sum_{j=1}^{n} x_j$ при условиях (2.3) и

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i.$$

Легко видеть, что

$$s_i=\max\left\{p \,\middle|\, \sum_{j\leqslant p} a_{ij} \leqslant b_i\right\}, \tag{2.4}$$

где $a_{i1} \leqslant a_{i2} \leqslant \ldots \leqslant a_{in}$. В случае, когда s_i не существует, задача не имеет решения. Если все s_i $(i=\overline{1, m})$, вычисленные по формуле (2.4), существуют, то для любого плана X задачи A имеет место соотношение

$$\sum_{j=1}^{n} x_j \leqslant s = \min_{1\leqslant i\leqslant m} s_i.$$

Поэтому в качестве вспомогательной задачи $\bar{A}$ рассмотрим задачу максимизации целевой функции $F(X)$ при условиях (2.3) и

$$\sum_{j=1}^{n} x_j \leqslant s.$$

Прежде чем излагать алгоритм построения последовательности планов задачи $\bar{A}$, введем необходимые понятия и обозначения. Для пары (u, v), где

$$u \subset \{\overline{1, n}\}, \quad v \subset \{\overline{1, n}\}, \quad u \cap v = \varnothing,$$
$$u \cup v = \{1, 2, \ldots, |u| + |v|\},$$

определим число $s(u, v)$, которое не меньше числа ненулевых компонент любого вектора

$$(x_{|u|+|v|+1}, x_{|u|+|v|+2}, \ldots, x_n),$$

удовлетворяющего условиям

$$\sum_{j=|u|+|v|+1} a_{ij} x_j \leqslant b_i - \sum_{j \in v} a_{ij}, \quad i = \overline{1, m},$$
$$x_j = 0 \quad \text{или} \quad 1, \quad j \in N \setminus \{u \cup v\}.$$

Заметим, что $s = s(\varnothing, \varnothing)$. Если число $s(u, v)$ существует, то пару (u, v) будем называть *допустимой*. Для допустимой пары (u, v) определим:

1) план $X(u, v) = (x_1^0, x_2^0, \ldots, x_n^0)$, компоненты которого вычисляются по формуле

$$x_j^0 = \begin{cases} 1, & \text{если} \\ & j \in v \cup \{\overline{|u| + |v| + 1, |u| + |v| + s(u, v)}\}, \\ 0, & \text{если} \quad j \in u \cup \{\overline{|u| + |v| + s(u, v) + 1, n}\} \end{cases}$$

2) множество пар

$$O(u, v) = \{(u_1, v_1), (u_2, v_2), \ldots, (u_{s(u,v)}, v_{s(u,v)})\}$$

таких, что

$$u_i = u \cup \{|u| + |v| + i\}, \quad i = \overline{1, s(u, v)},$$
$$v_1 = v,$$
$$v_i = v \cup \{\overline{|u| + |v| + 1, |u| + |v| + i - 1}\}, \; i = \overline{2, s(u, v)}.$$

При $s(u, v) = 0$ полагаем $O(u, v) = \varnothing$. Через $O^*(u, v)$ обозначим множество всех допустимых пар (u_i, v_i) из $O(u, v)$. Очевидно, если $O(u, v) = \varnothing$, то $O^*(u, v) = \varnothing$. С допустимой парой (u, v) отождествим множество $R(u, v)$ тех планов X из R, для которых

$$x_j = \begin{cases} 1, & \text{если} \quad j \in v, \\ 0, & \text{если} \quad j \in u, \end{cases}$$
$$\sum_{j=1}^{n} x_j \leqslant s(u, v) + |v|.$$

Опишем теперь алгоритм φ построения последовательности планов $r_1, r_2, \ldots$ задачи $\bar{A}$.

Алгоритм φ. На подготовительном этапе работы:

а) упорядочиваем переменные так, что $c_1 \geqslant \geqslant c_2 \geqslant \ldots \geqslant c_n \geqslant 0$;

б) полагаем $W_1 = (u_1^1, v_1^1)$, где $u_1^1 = v_1^1 = \varnothing$ (заметим, что с парой (u_1^1, v_1^1) отождествляется множество

$$R(u_1^1, v_1^1) = R(\varnothing, \varnothing) = R).$$

k-й шаг ($k = 1, 2, \ldots$). Из множества пар W_k находим пару (u_k^*, v_k^*) такую, что

$$F(X(u_k^*, v_k^*)) = \max_{(u, v) \in W_k} F(X(u, v)).$$

Полагаем $r_k = X(u_k^*, v_k^*)$, образуем множество W_{k+1} путем замены в W_k пары (u_k^*, v_k^*) множеством пар $O^*(u_k^*, v_k^*)$ и переходим к $(k+1)$-му шагу.

Обоснование того, что алгоритм φ строит последовательность планов $r_1, r_2, \ldots$ задачи $\bar{A}$ в порядке невозрастания $F(X)$, опустим, поскольку оно проводится аналогично приведенному в § 3 гл. II. Так как в данном случае мажоранта совпадает с функцией цели, то первый план этой последовательности, удовлетворяющий условиям (2.2), и будет оптимальным планом задачи A.

Пусть Ω_i — множество, заданное условиями

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i, \quad 0 \leqslant x_j \leqslant 1, \quad j = \overline{1, n}.$$

Тогда число

$$F^* = \min_{1 \leqslant i \leqslant m} \max_{X \in \Omega_i} F(X)$$

является верхней границей функции $F(X)$ при условиях (2.2), (2.3), т. е. $F^* \geqslant F(X)$ для каждого $X \in P$. Поэтому, прежде чем проверять принадлежность плана r_k множеству P, следует убедиться, что $F(r_k) \leqslant F^*$.

Следовательно, алгоритм ψ решения исходной задачи A состоит в следующем.

Алгоритм ψ. На предварительном шаге находим величину F^*.

k-й шаг ($k \geqslant 1$). При помощи алгоритма φ строим план r_k. Если $F(r_k) \geqslant F^*$, то переходим к $(k+1)$-му шагу. Если $F(r_k) \leqslant F^*$, то проверяем, является ли r_k планом исходной задачи A. Если да, то r_k — оптимальный план исходной задачи. Если нет — переходим к $(k+1)$-му шагу.

Проиллюстрируем работу алгоритма на примере.

Пример. Необходимо максимизировать функцию

$$15x_1 + 14x_2 + 10x_3 + 8x_4 + 5x_5$$

при ограничениях

$$3x_1 + 2x_2 + 2x_3 + 3x_4 + 2x_5 \leqslant 10,$$

$$3x_1 + 4x_2 + 2x_3 + 3x_4 + 2x_5 \leqslant 10,$$

$$x_j = 0 \quad \text{или} \quad 1, \quad j = \overline{1,\ 5}.$$

Так как переменные уже упорядочены, то на подготовительном шаге работы алгоритма вычисляем F^*. Для этого находим

$$F_1 = 47, \quad F_2 = 41^1/_3.$$

Следовательно, $F^* = 41^1/_3$. Таким образом, значение функции, соответствующее оптимальному плану исходной задачи, не может превосходить величины $41^1/_3$.

Строим множество $W_1 = \{(u_1^1,\ v_1^1)\}$, где $u_1^1 = v_1^1 = \varnothing$. Вычисляем $F(u_1^1,\ v_1^1) = 47$.

1-й шаг. Из множества W_1 выбираем пару $(u_1^*,\ v_1^*)$ с максимальным значением функции. В данном случае это пара $(u_1^1,\ v_1^1)$ со значением функции $F(u_1^*,\ v_1^*) = F(u_1^1,\ v_1^1) = 47$. Следовательно,

$$r_1 = X(u_1^*,\ v_1^*).$$

Так как $F(r_1) > F^* = 41^1/_3$, то строим множество

$$W_2 = \{(u_2^1,\ v_2^1),\ (u_2^2,\ v_2^2),\ (u_2^3,\ v_2^3),\ (u_2^4,\ v_2^4)\},$$

где

$$u_2^1 = \{1\}, \quad v_2^1 = \varnothing, \quad u_2^2 = \{2\}, \quad v_2^2 = \{1\},$$

$$u_2^3 = \{3\}, \quad v_2^3 = \{1,\ 2\}, \quad u_2^4 = \{4\}, \quad v_2^4 = \{1,\ 2,\ 3\},$$

вычисляем величины

$$F\left(u_2^1,\ v_2^1\right)=37,\quad F\left(u_2^2,\ v_2^2\right)=38,\quad F\left(u_2^3,\ v_2^3\right)=42,$$
$$F\left(u_2^4,\ v_2^4\right)=44$$

и переходим ко 2-му шагу.

2-й шаг. Находим пару $\left(u_2^*,\ v_2^*\right)=\left(u_2^4,\ v_2^4\right)$ со значением функции $F\left(u_2^*,\ v_2^*\right)=44$. Следовательно,

$$r_2=X\left(u_2^*,\ v_2^*\right).$$

Так как $F(r_2)>F^*$, то строим множество

$$W_3=\left\{\left(u_3^1,\ v_3^1\right),\ \left(u_3^2,\ v_3^2\right),\ \left(u_3^3,\ v_3^3\right),\ \left(u_3^4,\ v_3^4\right)\right\},$$

где

$$\left(u_3^1,\ v_3^1\right)=\left(u_2^1,\ v_2^1\right),\quad \left(u_3^2,\ v_3^2\right)=\left(u_2^2,\ v_2^2\right),$$
$$\left(u_3^3,\ v_3^3\right)=\left(u_2^3,\ v_2^3\right),\quad u_3^4=\{4,\ 5\},\quad v_3^4=\{1,\ 2,\ 3\},$$

вычисляем $F\left(u_3^4,\ v_3^4\right)=39$ и переходим к 3-му шагу.

3-й шаг. Находим пару

$$\left(u_3^*,\ v_3^*\right)=\left(u_3^3,\ v_3^3\right),\quad F\left(u_3^*,\ v_3^*\right)=42.$$

Следовательно,

$$r_3=X\left(u_3^*,\ v_3^*\right).$$

Так как $F(r_3)\geqslant F^*$, то строим множество

$$W_4=\left\{\left(u_4^1,\ v_4^1\right),\ \left(u_4^2,\ v_4^2\right),\ \left(u_4^3,\ v_4^3\right),\ \left(u_4^4,\ v_4^4\right),\ \left(u_4^5,\ v_4^5\right)\right\},$$

где

$$\left(u_4^1,\ v_4^1\right)=\left(u_3^1,\ v_3^1\right),\quad \left(u_4^2,\ v_4^2\right)=\left(u_3^2,\ v_3^2\right),$$
$$\left(u_4^3,\ v_4^3\right)=\left(u_3^4,\ v_3^4\right),\quad u_4^4=\{3,\ 4\},\quad v_4^4=\{1,\ 2\},$$
$$u_4^5=\{3,\ 5\},\quad v_4^5=\{1,\ 2,\ 4\}.$$

Далее вычисляем $F\left(u_4^4,\ v_4^4\right)=34$, $F\left(u_4^5,\ v_4^5\right)=37$ и переходим к 4-му шагу.

4-й шаг. Находим пару

$$\left(u_4^*,\ v_4^*\right)=\left(u_4^3,\ v_4^3\right),\quad F\left(u_4^*,\ v_4^*\right)=39.$$

Следовательно,

$$r_4=X\left(u_4^*,\ v_4^*\right).$$

Так как $F(r_4)=39<F^*=41^1/_3$, то проверяем, является ли вектор $r_1=(1, 1, 1, 0, 0)$ планом исходной задачи. Оказывается, да. Следовательно, r_1 — оптимальный план решаемой задачи.

§ 3. Многомерная задача о ранце с булевыми переменными

В двух предыдущих параграфах при решении задач целочисленного линейного программирования применялся тот вариант общей схемы, при котором множество планов задачи расширялось, а мажорантой являлась сама целевая функция. При этом первый элемент строящейся последовательности, являющийся планом исходной задачи, будет ее оптимальным планом. В то же время несомненный интерес представляет такой вариант общей схемы, при котором строятся лишь планы исходной задачи ($R=P$), поскольку в этом случае на каждом шаге можно оценить отклонение лучшего из полученных планов от оптимального. Естественно, что при этом необходимо задать мажоранту, отличную от целевой функции.

Такой подход удается осуществить для решения *многомерной задачи A о ранце с булевыми переменными*:

$$F(X)=\sum_{j=1}^{n} c_j x_j \to \max, \tag{3.1}$$

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \quad i=\overline{1, m}, \tag{3.2}$$

$$0 \leqslant x_j \leqslant 1, \quad j=\overline{1, n}, \tag{3.3}$$

$$x_j \text{ — целое число}, \quad j=\overline{1, n}, \tag{3.4}$$

где $a_{ij} \geqslant 0$, $b_i > 0$, c_j — целые числа. Не исключая общности, можно считать, что $c_j > 0$ для любого j (в противном случае j-я компонента оптимального плана равна 0).

Ниже предлагается два алгоритма решения этой задачи, в основе которых лежит идея построения планов задачи A в порядке невозрастания мажоранты, представляющей собой верхнюю оценку значения функции $F(X)$ на подмножествах множества P. Особенностью второго алгоритма является то, что он строит последовательность планов с пропуском заведомо неоптимальных.

Первый алгоритм (ψ_1). Совокупность индексов $v \subset N = \{1, 2, \ldots, n\}$ будем называть *допустимой* и писать $v \in D$, если

$$\sum_{j \in v} a_{ij} \leqslant b_i, \quad i = \overline{1, m}.$$

Очевидно, что $v = \varnothing$ — допустимая совокупность.

Пусть пара индексов (u_0, v_0) такова, что

$$u_0 \subset N, \quad v_0 \subset N, \quad u_0 \cap v_0 = \varnothing, \quad v_0 \in D,$$
$$u_0 \cup v_0 = \{1, 2, \ldots, |u_0| + |v_0|\}.$$

Такую пару будем называть *допустимой*. Определим для нее план

$$X(u_0, v_0) = (x_1^0, \ldots, x_n^0) \in P,$$

компоненты которого вычисляются по формулам

$$x_j^0 = \begin{cases} 0 & \text{для} \quad j \in u_0, \\ 1 & \text{для} \quad j \in v_0, \end{cases}$$

$$x_r^0 = \begin{cases} 1, & \text{если} \quad \{j \mid j < r,\ x_j^0 = 1\} \cup \{r\} \in D, \\ 0 & \text{в противном случае,} \end{cases}$$

где $r = \overline{|u_0| + |v_0| + 1, n}$.

Кроме того, введем обозначения

$$\bar{u}_0 = \{j \mid j \notin u_0,\ x_j^0 = 0\}, \quad \bar{v}_0 = \{j \mid j \notin v_0,\ x_j^0 = 1\}.$$

Для допустимой пары (u_0, v_0) определим множество пар

$$W(u_0, v_0) = \{(u_1, v_1), (u_2, v_2), \ldots, (u_{|\bar{v}_0|}, v_{|\bar{v}_0|})\}$$

по следующему правилу:

$$v_1 = v_0,$$
$$v_p = v_{p-1} \cup \{j'_{p-1}\}, \quad p = \overline{2, |\bar{v}_0|},$$
$$u_p = \{j \mid j \in u_0 \cup \bar{u}_0,\ j < j'_p\} \cup \{j'_p\}, \quad p = \overline{1, |\bar{v}_0|},$$

где $\{j'_1, j'_2, \ldots, j'_{|\bar{v}_0|}\} = \bar{v}_0$, причем $j'_1 < j'_2 < \ldots < j'_{|\bar{v}_0|}$.

Очевидно, что каждая пара множества $W(u_0, v_0)$ — допустимая. Будем считать, что $W(u_0, v_0) = \varnothing$, если $\bar{v}_0 = \varnothing$.

Через $P(u_0, v_0)$ обозначим множество планов задачи (3.1)—(3.4) с дополнительным условием

$$x_j^0 = \begin{cases} 0 & \text{для} \quad j \in u_0, \\ 1 & \text{для} \quad j \in v_0. \end{cases} \tag{3.5}$$

Тогда справедливо соотношение

$$\bigcup_{(u, v) \in W(u_0, v_0)} P(u, v) = P(u_0, v_0) \setminus X(u_0, v_0). \tag{3.6}$$

Пусть

$$F(u_0, v_0) = \min_{1 \leqslant i \leqslant m} F_i(u_0, v_0),$$

где $F_i(u_0, v_0)$ — максимальное значение функции $F(X)$ при условиях (3.3), (3.5) и $\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i$. Решение этой задачи не представляет труда (см. § 1).

Сформулируем алгоритм φ_1 построения последовательности планов задачи A.

Алгоритм φ_1. Пусть $W_1 = \{(u_1^1, v_1^1)\}$, где $u_1^1 = v_1^1 = \varnothing$.

k-й шаг ($k = 1, 2, \ldots$). Из множества W_k выбираем пару (u_k^*, v_k^*) такую, что

$$F(u_k^*, v_k^*) = \max_{(u, v) \in W_k} F(u, v).$$

Находим план $r_k = X(u_k^*, v_k^*)$. Строим множество пар

$$W_{k+1} = \{W_k \setminus (u_k^*, v_k^*)\} \cup W(u_k^*, v_k^*)$$

и переходим к $(k+1)$-му шагу алгоритма φ_1.

Из алгоритма φ_1 и соотношения (3.6) вытекает равенство

$$\bigcup_{(u, v) \in W_{k+1}} P(u, v) = P \setminus \{r_1, r_2, \ldots, r_k\}, \tag{3.7}$$

$$k = 1, 2, \ldots$$

Так как $F(u_k^*, v_k^*) \geqslant F(r_k)$ ($k = 1, 2, \ldots$), то функция $Q(r_k) = F(u_k^*, v_k^*)$ является мажорантой функции

$F(X)$, причем для любой пары $(u, v) \in W_k$ справедливо равенство

$$F(u, v) = Q(X(u, v)) = \max_{X \in P(u, v)} Q(X).$$

Теперь, учитывая (3.7), можно утверждать, что алгоритм φ_1 строит требуемую последовательность планов.

Заметим, что при построении множества W_{k+1} из него разумно удалять пары (u, v), для которых выполняется неравенство

$$F(u, v) \leqslant \max_{1 \leqslant i \leqslant k} F(r_i).$$

Поэтому алгоритм ψ_1 нахождения оптимального плана задачи A можно сформулировать следующим образом.

Алгоритм ψ_1. Пусть $W_1' = W_1$.

k-й шаг ($k = 1, 2, \ldots$). Выбираем такую пару (u_k^*, v_k^*), что

$$F(u_k^*, v_k^*) = \max_{(u, v) \in W_k'} F(u, v).$$

Находим $r_k = X(u_k^*, v_k^*)$ и строим

$$W_{k+1}' = \{(u, v) \mid (u, v) \in (W_k' \setminus (u_k^*, v_k^*)\} \cup \cup W(u_k^*, v_k^*), \quad F(u, v) \geqslant \max_{1 \leqslant i \leqslant k} F(r_i) = F(r_k^*)\}.$$

Если $W_{k+1}' = \varnothing$, то r_k^* — оптимальный план. Если $W_{k+1}' \neq \varnothing$, то переходим к $(k+1)$-му шагу.

Проиллюстрируем работу алгоритма на примере.

Пример. Максимизовать

$$F(X) = 6x_1 + 4x_2 + x_3$$

при условиях

$$x_1 + 2x_2 + 3x_3 \leqslant 5,$$
$$2x_1 + x_2 + x_3 \leqslant 4,$$
$$x_j = 0 \quad \text{или} \quad 1, \quad j = 1, 2, 3.$$

На подготовительном шаге работы алгоритма строим множество $W_1' = \{(u_1^1, v_1^1)\}$, где $u_1^1 = v_1^1 = \varnothing$, $F(u_1^1, v_1^1) = 10^2/_3$.

1-й шаг. Из множества W_1' выбираем пару (u_1^*, v_1^*) с максимальным значением $F(u, v)$. В нашем случае $(u_1^*, v_1^*) = (u_1^1, v_1^1)$. Строим план $r_1 = X(u_1^*, v_1^*) = (1, 1, 0)$. Следовательно,

$$Q(r_1) = 10^2/_3, \quad F(r_1) = 10.$$

Так как $Q(r_1) > F(r_1)$, то строим множество пар

$$W_2 = \{(u_2^1, v_2^1), (u_2^2, v_2^2)\},$$

где

$$u_2^1 = \{1\}, \quad v_2^1 = \varnothing, \quad u_2^2 = \{2\}, \quad v_2^2 = \{1\},$$

$$F(u_2^1, v_2^1) = 5, \quad F(u_2^2, v_2^2) = 7.$$

Поскольку $F(u_2^1, v_2^1) = 5 < F(r_1) = 10$ и $F(u_2^2, v_2^2) = 7 < F(r_1)$, то $W_2' = \varnothing$. Следовательно, $r_1 = (1, 1, 0)$ — оптимальный план исходной задачи.

Второй алгоритм (ψ_2). Введем *отношение порядка* для двух векторов $X' = (x_1', x_2', \ldots, x_n')$ и $X'' = (x_1'', x_2'', \ldots, x_n'')$. Будем считать, что $X' \leqslant X''$, если $x_j' \leqslant x_j''$ $(j = \overline{1, n})$. Если среди последних неравенств есть хотя бы одно строгое, то будем говорить, что X' *меньше, чем* X'' $(X' < X'')$.

Очевидно, если X_0 — план задачи A, то всякий меньший вектор — тоже план задачи A. Кроме того, так как все c_j больше нуля, то

$$F(X_0) > F(X) \quad \text{для} \quad X \leqslant X_0. \tag{3.8}$$

Поэтому при построении последовательности планов задачи A в порядке ухудшения мажоранты можно не конструировать планы, которые меньше уже полученных, так как среди них нет оптимального плана.

Пусть пара (u_0, v_0) такова, что

$$u_0, v_0 \subset N, \quad u_0 \cap v_0 = \varnothing,$$
$$N \setminus \{u_0 \cup v_0\} = \{j_1, j_2, \ldots, j_q\},$$

где $j_1 < j_2 < \ldots < j_q$. Пусть

$$J_k = \{j_1, j_2, \ldots, j_k\}, \quad k = \overline{1, q}, \quad J_0 = \varnothing.$$

Определим план $X(u_0, v_0)=(x_1^0, \ldots, x_n^0)$ задачи A, компоненты которого вычисляются по формулам

$$x_j^0=\begin{cases} 0 & \text{для } j\in u_0,\\ 1 & \text{для } j\in v_0,\end{cases}$$

$$x_{j_k}^0=\begin{cases} 1, & \text{если } v_0\cup\{j\mid j\in J_{k-1},\ x_j^0=1\}\cup\{j_k\}\in D,\\ 0 & \text{в противном случае,}\end{cases}$$

где $k=\overline{1, q}$. Для этого плана введем обозначение

$$\bar{u}_0=\{j\mid j\in N\setminus u_0,\ x_j^0=0\}.$$

Для допустимой пары (u_0, v_0) определим множество пар

$$W(u_0, v_0)=\{(u_1, v_1), (u_2, v_2), \ldots, (u_{|\bar{u}_0|}, v_{|\bar{u}_0|})\}$$

по правилам:

$$u_1=u_0,$$

$$u_p=u_{p-1}\cup\{j'_{p-1}\},\quad p=\overline{2, |\bar{u}_0|},$$

$$v_p=v_0\cup\{j'_p\},\quad p=\overline{1, |\bar{u}_0|},$$

где $\{j'_1, j'_2, \ldots, j'_{|\bar{u}_0|}\}=\bar{u}_0$, причем $j'_1<j'_2<\ldots<j'_{|\bar{u}_0|}$.

Естественно считать, что $W(u_0, v_0)=\varnothing$, если $\bar{u}_0=\varnothing$.

Введем множество допустимых пар

$$W^*(u_0, v_0)=\{(u, v)\mid (u, v)\in W(u_0, v_0),\ v\in D\}.$$

Как и в предыдущем пункте, через $P(u_0, v_0)$ обозначим множество планов задачи (3.1)—(3.4) с дополнительным условием (3.5). Тогда имеет место следующее соотношение:

$$\bigcup_{(u, v)\in W^*(u_0, v_0)} P(u, v)=$$
$$=P(u_0, v_0)\setminus\{X\mid X\leqslant X(u, v),\ X\in P(u_0, v_0)\}. \qquad (3.9)$$

Сформулируем алгоритм φ_2 построения последовательности планов задачи A.

Алгоритм φ_2. Пусть $W_1=\{(u_1^1, v_1^1)\}$, где $u_1^1=v_1^1=\varnothing$.

k-й шаг ($k=1, 2, \ldots$). Находим

$$\max_{(u, v)\in W_k} F(u, v)=F(u_k^*, v_k^*)$$

и план $r_k = X(u_k^*, v_k^*)$. Строим множество пар

$$W_{k+1} = \{W_k \setminus (u_k^*, v_k^*)\} \cup W^*(u_k^*, v_k^*)$$

и переходим к $(k+1)$-му шагу.

Из (3.9) и алгоритма φ_2 получаем

$$\bigcup_{(u, v) \in W_{k+1}} P(u, v) = P \setminus \bigcup_{i=1}^{k} P(r_i), \quad k = 1, 2, \ldots, \qquad (3.10)$$

где $P = P(u_1^*, v_1^*)$ — множество планов задачи A,

$$P(r_i) = \{X \mid X \leqslant r_i, \ X \in P(u_i^*, v_i^*)\}.$$

При этом, в силу (3.8), имеем

$$F(r_k) = \max_{X \in P(r_k)} F(X), \quad k = 1, 2, \ldots \qquad (3.11)$$

Так как $F(u_k^*, v_k^*) \geqslant F(r_k)$ $(k = 1, 2, \ldots)$, то функция

$$Q(r_k) = F(u_k^*, v_k^*)$$

является мажорантой функции $F(X)$, причем для любой пары $(u, v) \in W_k$ справедливо равенство

$$F(u, v) = Q(X(u, v)) = \max_{X \in P(u, v)} Q(X).$$

Теперь, учитывая (3.10) и (3.11), можно утверждать, что алгоритм φ_2 строит требуемую последовательность. Поэтому для последовательности $r_1, r_2, \ldots$ справедлив критерий оптимальности.

Итак, k-й шаг $(k = 1, 2, \ldots)$ алгоритма ψ_2 решения задачи A состоит в том, что с помощью алгоритма φ_2 находим план и проверяем, выполняется ли условие

$$Q(r_k) \leqslant \max_{1 \leqslant i \leqslant k} F(r_i) = F(r_k^*).$$

Если да, то r_k^* — оптимальный план задачи A. Если нет, то переходим к следующему шагу.

Проиллюстрируем работу алгоритма на примере предыдущего пункта.

Пример. Максимизировать функцию

$$F(X) = 6x_1 + 4x_2 + x_3$$

при условиях

$$x_1+2x_2+3x_3\leqslant 5,$$
$$2x_1+x_2+x_3\leqslant 4,$$
$$x_j=0 \quad \text{или} \quad 1, \quad j=1,\ 2,\ 3.$$

На предварительном шаге работы алгоритма строим множество $W_1=\{(u_1^1,\ v_1^1)\}$, где $u_1^1=v_1^1=\varnothing$. Вычисляем $F(u_1^1,\ v_1^1)=10^2/_3$.

1-й шаг. Из множества W_1 выбираем пару $(u_1^*,\ v_1^*)$ с максимальным значением функции $F(u,\ v)$. Для данного примера это будет пара $(u_1^1,\ v_1^1)$ со значением функции $F(u_1^*,\ v_1^*)=F(u_1^1,\ v_1^1)=10^2/_3$. Строим план

$$r_1=X(u_1^*,\ v_1^*)=(1,\ 1,\ 0).$$

Следовательно, $Q(r_1)=10^2/_3$ и $F(r_1)=10$. Так как $Q(r_1)>F(r_1)$, то строим множество $W_2=\{(u_2^1,\ v_2^1)\}$, где $u_2^1=\varnothing$, $v_2^1=\{3\}$. Вычисляем величину $F(u_2^1,\ v_2^1)=9$ и переходим ко 2-му шагу.

2-й шаг. Находим пару $(u_2^*,\ v_2^*)=(u_2^1,\ v_2^1)$. Строим план

$$r_2=X(u_2^*,\ v_2^*)=(1,\ 0,\ 1), \quad F(r_2)=7, \quad Q(r_2)=9.$$

Так как $Q(r_2)<\max\{F(r_1),\ F(r_2)\}$, то $r_1=(1,\ 1,\ 0)$ — оптимальный план задачи.

§ 4. Вычислительные аспекты

Известно, что для решения задачи, рассмотренной в § 1, при $m=1$ может быть эффективно использован метод динамического программирования. Однако по мере увеличения числа ограничений трудоемкость метода и требования, предъявляемые к памяти вычислительной машины, резко возрастают в связи с необходимостью вычисления функций m переменных. При решении задачи с двумя линейными ограничениями оказалось удобнее ввести множитель Лагранжа и использовать принцип оптимальности на каждом шаге, соответствующем выбранному значению множителя Лагранжа. Предлагаемый в § 1 алгоритм позволяет, на наш взгляд, в некоторой мере преодолеть трудности, связанные с размерностью, так как он состоит в направленном переборе планов вспомогательной задачи с одним линейным ограничением до получения плана исходной задачи, который и будет оптимальным. Таким образом, решение многомерной задачи сводится к решению ряда одномерных задач. Кроме этого, объем запоминаемой информации возрастает незначительно по сравнению с одномерной задачей.

Действительно, множество W_t^k планов вспомогательной одномерной задачи получается из множества W_t^{k-1} путем замены плана r_t^{k-1} множеством планов $O\left(r_t^{k-1}\right)$, которое всегда можно восстановить, имея план r_t^{k-1}. Следовательно, для построения k-го плана последовательности $r_t^1, r_t^2, \ldots$ необходимо иметь множество W_t^1 и предыдущие планы последовательности $r_t^1, r_t^2, \ldots, r_t^{k-1}$.

В § 3 для решения многомерной задачи о ранце с булевыми переменными предлагаются алгоритмы решения, в которых в качестве мажоранты берется верхняя оценка значений функции $F(X)$ на подмножествах множества P, а множество P не расширяется ($R=P$). Таким образом, в процессе работы алгоритмов строятся планы исходной задачи в порядке невозрастания оценочной функции. К достоинствам этих алгоритмов можно отнести то, что, строя лишь планы исходной задачи, легко оценить отклонение лучшего из полученных планов от оптимального, так как на k-м шаге работы для оптимального значения F^* функции задачи A имеют место двусторонние оценки

$$\max_{1 \leqslant l \leqslant k} F(r_l) \leqslant F^* \leqslant Q(r_k).$$

Это обстоятельство особенно важно при решении реальных задач, для которых зачастую достаточно найти приближенное решение с заданной точностью ($\delta\%$). В этом случае работу алгоритмов можно остановить, как только выполнено неравенство

$$\frac{F\left(u_k^*, v_k^*\right) - \max\limits_{1 \leqslant l \leqslant k} F(r_l)}{F\left(u_k^*, v_k^*\right)} \cdot 100 \leqslant \delta.$$

Алгоритмы, изложенные в § 3, реализованы на ЭВМ Минск-32. Решались практические задачи, возникшие при разработке автоматизированных систем управления для ряда предприятий. В таблице 5

Таблица 5

$m \times n$	Время счета	Количество итераций		Количество единиц в оптимальном плане
		а	б	
5×140	1′ 36″	28	21	91
5×145	2′ 51″	44	44	41
5×150	46″	9	9	64
5×155	18′	245	85	77
5×160	3′	73	27	144
5×165	40″	8	3	77
5×170	2′	35	35	96
5×175	6′	76	70	79
5×180	2′	41	4	62
8×162	35′	287	101	96
10×98	11′	70	12	52
10×100	37′	156	1	35
10×106	10′	39	4	61
10×110	2′	677	84	43
10×112	24′	289	126	65
10×120	2′ 45″	21	21	40
10×124	12′	40	36	76
10×130	30″	2	2	42
10×131	18′	50	41	55
10×140	50″	8	8	40
10×160	2′ 20″	19	13	51
10×180	20″	3	3	62
11×175	34′	301	97	102
15×75	14′	110	46	21
15×90	9′	378	4	46
15×95	24′	757	1	39
15×120	9″	6	5	38
15×130	2″	1	1	43
20×80	11″	13	12	32
30×161	18′	143	11	22
50×30	2′	37	37	6
50×40	10′	92	59	8
60×30	4′	15	12	7
60×40	12′	420	247	10
70×30	1′	8	7	5
70×40	9′	347	33	8

Прод. табл. 5

$m \times n$	Время счета	Количество итераций		Количество единиц в оптимальном плане
		а	*б*	
70×60	8′	48	22	30
80×30	2′	55	27	6
90×30	5′	69	12	6
90×50	5′	455	12	8

приведены некоторые результаты расчетов по решению задач формирования квартальных и месячных планов производства. Столбцы *а* и *б* таблицы указывают, соответственно, общее число итераций и число итераций до получения оптимального плана.

Разработанные программы позволяют находить как точное, так и приближенное решение, отличающееся от оптимального на наперед заданную величину.

Приведенные в таблице 5 данные позволяют говорить о хорошей сходимости алгоритмов. При этом оптимальный план строится как правило достаточно быстро. Однако алгоритмы продолжают работать для того, чтобы можно было убедиться в оптимальности уже построенного плана.

Некоторые результаты экспериментов на ЕС ЭВМ по решению задач ЦЛП методом построения последовательности планов приведены в гл. VI.

ГЛАВА V

РЕШЕНИЕ СПЕЦИАЛЬНЫХ ЗАДАЧ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ

§ 1. Задачи вогнутого программирования

Известно, что минимум вогнутой функции на выпуклом многограннике достигается в одной из вершин; поэтому такие задачи вогнутого программирования можно рассматривать как задачи дискретной оптимизации, для решения которых применима общая схема метода построения последовательности планов.

Вспомогательная задача. Рассмотрим задачу A минимизации вогнутой функции $F(X)$ на многограннике Ω, заданном ограничениями

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = \overline{1, n}, \tag{1.1}$$

$$x_{ij} \geqslant 0, \quad i = \overline{1, m}, \quad j = \overline{1, n}. \tag{1.2}$$

Нетрудно показать, что план $X^0 = \left\| x_{ij}^0 \right\|_{m \times n}$ задачи A будет вершиной многогранника Ω тогда и только тогда, когда он содержит n положительных компонент. Другими словами, план X^0 является вершиной тогда и только тогда, когда для любого j $(j = \overline{1, n})$ существует $1 \leqslant i_j \leqslant m$ такое, что $x_{i_j j}^0 = b_j$, $x_{ij}^0 = 0$ $(\forall i \neq j)$. Таким образом, многогранник Ω является целочисленным при любых целых $b_1, b_2, \ldots, b_n$.

Отсюда следует, что существует взаимно однозначное соответствие между множеством вершин и множеством W систем представителей множеств $I_j = \{1, 2, \ldots, m\}$ $(j = \overline{1, n})$ (см. § 4 гл. II). Поэтому задача A эквивалентна задаче

$$\min_{r \in W} F(X_r),$$

где X_r — вершина, соответствующая системе представителей $r = \{i_1, i_2, \ldots, i_n\}$.

Итак, задача A может быть решена путем построения последовательности систем представителей в порядке неубывания некоторой миноранты $Q(r)$ до выполнения критерия оптимальности.

Так как в нашем случае все множества I_j $(j=\overline{1, n})$ совпадают, то построение последовательности с. п. сводится к построению последовательности n-выборок. Назовем *n-выборкой из множества M* упорядоченное n-множество элементов из M. Обозначим через R множество всех n-выборок $I = \{i_1, i_2, \ldots, i_n\}$ из множества $M = \{1, 2, \ldots, m\}$. Следовательно, задача отыскания минимума функции $F(X)$ на многограннике Ω эквивалентна задаче отыскания

$$\min_{I \in R} F(X_I), \tag{1.3}$$

где X_I — вершина, соответствующая выборке I. Построение последовательности выборок множества R в порядке неубывания миноранты $Q(I)$ легко проводится в случае, когда функция $Q(I)$ имеет вид

$$Q(I) = \sum_{j=1}^{n} c_{i_j j},$$

где $\| c_{ij} \|_{m \times n}$ — некоторая $(m \times n)$-матрица.

Пусть перестановки $(\alpha_j^1, \alpha_j^2, \ldots, \alpha_j^m)$ чисел множества M таковы, что $c_{\alpha_j^1 j} \leqslant c_{\alpha_j^2 j} \leqslant \ldots \leqslant c_{\alpha_j^m j}$ $(j = \overline{1, n})$. Тогда выборка $I_1 = \{\alpha_1^1, \alpha_2^1, \ldots, \alpha_n^1\}$ удовлетворяет условию

$$Q(I_1) = \min_{I \in R} Q(I),$$

т. е. является оптимальной.

Введем обозначение

$$O(I_1) = \bigcup_{j=1}^{n} I^{(j)},$$

где

$$I^{(j)} = (\alpha_1^1, \alpha_2^1, \ldots, \alpha_{j-1}^1, \alpha_j^2, \alpha_{j+1}^1, \ldots, \alpha_n^1).$$

Пусть

$$I = (i_1, \ldots, i_{p-1}, \alpha_p^q, \alpha_{p+1}^1, \ldots, \alpha_n^1) \in R,$$

где $1 \leqslant q \leqslant m$, $1 \leqslant p \leqslant n$.

Определим множество $O(I)$ следующим образом:

$$O(I)=\begin{cases}\bigcup\limits_{j=0}^{n-p} I^{(j)}, & \text{если } q<m,\ p\leqslant n,\\ \bigcup\limits_{j=1}^{n-p} I^{(j)}, & \text{если } q=m,\ p<n,\\ \varnothing, & \text{если } q=m,\ p=n,\end{cases}$$

где

$$I^{(0)}=\left(i_1,\ \ldots,\ i_{p-1},\ \alpha_p^{q+1},\ \alpha_p^1,\ \ldots,\ \alpha_n^1\right),$$

$$I^{(j)}=\left(i_1,\ \ldots,\ i_{p-1},\ \alpha_p^q,\ \alpha_{p+1}^1,\ \ldots,\ \alpha_{p+j-1}^1,\right.$$

$$\left.\alpha_{p+j}^2,\ \alpha_{p+j+1}^1,\ \ldots,\ \alpha_n^1\right),$$

$$j=\overline{1,n-p}.$$

В этих обозначениях алгоритм φ построения последовательности выборок в порядке неубывания функции $Q(I)$ описывается следующим образом.

Алгоритм φ. 1-й шаг. Строим выборку $I_1=\left(\alpha_1^1,\ \ldots\right.$ $\left.\ldots,\ \alpha_n^1\right)$.

k-й шаг ($k=2,\ 3,\ \ldots$). Строим множество выборок W_k по одному из следующих правил.

Правило 1. $W_k=\bigcup\limits_{j=1}^{k-1} O(I_j)\setminus\{I_1,\ I_2,\ \ldots,\ I_{k-1}\}$

Правило 2. $W_k=(W_{k-1}\setminus I_{k-1})\cup I_{k-1}$, причем $W_1=I_1$.

Далее, находим такую выборку I_k, что

$$Q(I_k)=\min_{I\in W_k} Q(I),$$

и переходим к следующему шагу.

Обоснование того, что алгоритм φ строит требуемую последовательность выборок, проводится аналогично § 4 гл. II.

Остановимся теперь на решении конкретных задач, для которых можно построить миноранты $Q(I)$ указанного выше вида.

Обобщенная задача размещения. Минимизировать функцию

$$F(X)=\sum_{i=1}^{m} f_i\left(\sum_{j=1}^{n} a_{ij}x_{ij}\right)+\sum_{i=1}^{m}\sum_{j=1}^{n} e_{ij}x_{ij} \tag{1.4}$$

на многограннике Ω, заданном ограничениями

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j=\overline{1, n}, \tag{1.5}$$

$$x_{ij} \geqslant 0, \quad i=\overline{1, m}, \quad j=\overline{1, n}, \tag{1.6}$$

где $a_{ij} \geqslant 0$, $b_j > 0$, $f_i(\sigma_i)$ — вогнутые функции, причем $f_i(0) = 0$.

Сформулированная задача превращается в известную задачу размещения производства, если $a_{ij} = 1$; b_j — потребность j-го потребителя; e_{ij} — затраты на перевозку единицы продукции из i-го пункта производства j-му потребителю; x_{ij} — объем перевозок из i-го пункта производства j-му потребителю; $f_i(x_i)$ — затраты на производство $x_i = \sum_{j=1}^{n} x_{ij}$ единиц продукции в i-м пункте производства.

Эту задачу можно также трактовать как задачу размещения предприятий по переработке сырья. Тогда a_{ij} показывает, сколько единиц «эталонного» сырья заменяет одна единица реального сырья j-й группы, если сырье перерабатывается на i-м предприятии. В другой интерпретации a_{ij} показывает выход готовой продукции на i-м предприятии из единицы сырья j-й группы.

Ранее было показано, что задача (1.4)—(1.6) эквивалентна задаче (1.3), поэтому для ее решения можно использовать алгоритм φ построения последовательности выборок.

Покажем, как можно строить миноранту $Q(I)$. Для множества $N' \subset N = \{\overline{1, n}\}$ положим

$$\sigma_i[N'] = \sum_{j \in N'} \sigma_i[j], \quad i=\overline{1, m},$$

где $\sigma_i[j] = a_{ij}b_j$. Естественно считать, что $\sigma_i[\varnothing] = 0$. Так как областью изменения каждой суммы

$$\sigma_i = \sum_{j=1}^{n} a_{ij}x_{ij}, \quad i=\overline{1, m},$$

является интервал $[0, \sigma_i[N]]$, то, учитывая вогнутость каждой функции $f_i(\sigma_i)$, в качестве миноранты на множестве R можно брать сумму

$$Q(I)=\sum_{j=1}^{n} c_{i_j j},$$

где

$$c_{ij}=\frac{f_i(\sigma_i[N])}{\sigma_i[N]}\sigma_i[j]+e_{ij}b_j, \quad i=\overline{1,m}, \quad j=\overline{1,n}.$$

Задача стандартизации. Рассмотрим задачу минимизации функции

$$F(X)=\sum_{i=1}^{m} f_i\left(\sum_{j=1}^{n} x_{ij}\right) \tag{1.7}$$

на множестве, заданном ограничениями

$$\sum_{i=1}^{m} a_{ij}x_{ij}=b_j, \quad j=\overline{1,n}, \tag{1.8}$$

$$x_{ij}\geqslant 0, \quad i=\overline{1,m}, \quad j=\overline{1,n}, \tag{1.9}$$

где $f_i(x)$ — монотонно возрастающие вогнутые функции, причем

$$f_i(0)=0, \quad i=\overline{1,m}, \quad a_{ij}\geqslant 0, \quad b_j>0.$$

Так как $f_i(x)$ $(i=\overline{1,m})$ — монотонно возрастающие вогнутые функции, то задача (1.7)—(1.9) эквивалентна задаче

$$\min_{I\in R^*} F(X_I), \tag{1.10}$$

где

$$R^*=\{I\mid I=\{i_1, \ldots, i_n\}\in R, \quad a_{i_j j}>0, \quad j=\overline{1,n}\},$$

а X_I — план задачи (1.7)—(1.9), соответствующий выборке I, т. е. матрица порядка $m\times n$, в каждом j-м столбце которой содержится по одной положительной компоненте $x_{i_j j}=b_j/a_{i_j j}$ (остальные компоненты равны нулю). Решение задачи (1.10) осуществляется путем построения последовательности выборок множества R^* в порядке неубывания миноранты

$$Q(I)=\sum_{j=1}^{n} c_{i_j j},$$

где

$$c_{ij}=\frac{b_j f_i\left(\sum\limits_{j\in N_i} b_j/a_{ij}\right)}{a_{ij}\sum\limits_{j\in N_i} b_j/a_{ij}},$$

$$N_i=\{j \mid j\in\overline{\{1,n\}},\ a_{ij}>0\}.$$

§ 2. Локальный подход

К вопросу построения последовательности планов в порядке неубывания миноранты можно подойти не только путем ветвления исходного множества (или его расширения), но и путем рассмотрения соседних элементов, т. е. осуществляя локальный подход. Этот подход мы изложим на примере решения задачи минимизации суммы вогнутых функций $F(X)=\sum\limits_{i=1}^{n} f_i(x_i)$ на выпуклом многограннике Ω. Таким образом, среди линейных неравенств, с помощью которых задан многогранник Ω, либо существуют неравенства вида $0\leqslant x_i\leqslant d_i$ $(i=\overline{1,n})$, либо они являются следствием других неравенств. Каждая функция $f_i(x_i)$ вогнута на интервале $[0, d_i]$.

Известно, что минимум вогнутой функции $F(X)$, если он существует, достигается по крайней мере в одной из вершин многогранника Ω. Таким образом, исходным множеством P, на котором задана функция $F(X)$, будем считать множество вершин $V(\Omega)$ многогранника Ω.

Не ограничивая общности, можно предположить, что многогранник Ω невырожден и, следовательно, любая вершина имеет n соседних вершин. Пусть $O(X_0)$— множество вершин, соседних с X_0. В качестве миноранты можно взять линейную форму

$$Q(X)=\sum_{i=1}^{n}\frac{f_i(d_i)-f_i(0)}{d_i}x_i+\sum_{i=1}^{n} f_i(0).$$

Теперь опишем алгоритм φ построения последовательности вершин многогранника Ω.

Алгоритм φ. На подготовительном этапе находим оптимальную вершину X_1, решая задачу минимизации линейной формы $Q(X)$ на множестве Ω. Пусть $W_1 = \{X_1\}$.

k-й шаг ($k \geqslant 1$). Находим план X_k такой, что

$$Q(X_k) = \min_{X \in W_k} Q(X), \tag{2.1}$$

строим множество

$$W_{k+1} = \bigcup_{i=1}^{k} O(X_i) \setminus \{X_1, X_2, \ldots, X_k\}$$

и переходим к следующему шагу.

Нетрудно показать, что последовательность $X_1, X_2, \ldots$ вершин многогранника Ω, построенная с помощью алгоритма φ, обладает свойством

$$Q(X_k) = \min_{X \in D_k} Q(X), \quad k \geqslant 1, \tag{2.2}$$

где $D_k = D_{k-1} \setminus X_{k-1}$ ($k \geqslant 2$), $D_1 = P = V(\Omega)$. Действительно, пусть для некоторого k ($k \geqslant 2$) равенство (2.2) неверно, т. е. существует вершина $X_0 \neq X_i$ ($i = \overline{1, k}$) такая, что

$$\min_{X \in D_k} Q(X) = Q(X_0) < Q(X_k). \tag{2.3}$$

Тогда на основании симплекс-метода можно утверждать существование последовательности соседних вершин, расположенных в порядке невозрастания функции $Q(X)$, начинающейся в X_0 и заканчивающейся в X_1:

$$X_0 = X^{(1)}, X^{(2)}, \ldots, X^{(p)} = X_1,$$
$$Q(X^{(i)}) \geqslant Q(X^{(j)}), \quad 1 \leqslant i < j \leqslant p. \tag{2.4}$$

Очевидно, что $X_0 \notin P_{k-1}$, а $X_1 \in P_{k-1}$, где $P_{k-1} = \{X_1, X_2, \ldots, X_{k-1}\}$. Поэтому существует такое число q ($1 \leqslant q \leqslant p-1$), что $X^{(q)} \notin P_{k-1}$, а $X^{(q+1)} \in P_{k-1}$. Отсюда, так как $X^{(q)} \in O(X^{(q+1)})$, то $X^{(q)} \in W_k$. С другой стороны, в силу (2.1), (2.3), (2.4) имеем

$$Q(X^q) < \min_{X \in W_k} Q(X).$$

Полученное противоречие доказывает, что алгоритм φ строит требуемую последовательность вершин.

Аналогичным образом легко доказать, что $W_k \neq \varnothing$ для $k \leqslant m$, где m — число вершин многогранника Ω.

Иными словами, алгоритм φ позволяет строить последовательность, состоящую из всех m вершин многогранника.

Таким образом, алгоритм ψ решения исходной задачи состоит в следующем.

Алгоритм ψ. s-й шаг ($s = 1, 2, \ldots$). С помощью алгоритма φ находим план X_k и проверяем, выполняется ли условие

$$Q(X_s) \geqslant \min_{i=\overline{1,\, s}} F(X_i) = F(X_s^*).$$

Если да, то X_s^* — оптимальный план. Если нет, то переходим к следующему шагу.

Замечание 1. Известно, что для перехода в соседнюю вершину необходимо ввести в базис новую переменную. Очевидно, что в нашем случае при построении $O(X_i)$ для того, чтобы не строить лишние вершины, это введение должно осуществляться лишь тогда, когда происходит неубывание линейной формы $Q(X)$.

Замечание 2. Очевидно, что с помощью предложенного метода можно решить задачу целочисленного линейного программирования с булевыми переменными:

$$L(X) = \sum_{j=1}^{n} c_j x_j \to \max,$$

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \quad i = \overline{1, m}, \tag{2.5}$$

$$0 \leqslant x_j \leqslant 1, \quad j = \overline{1, n}, \tag{2.6}$$

$$x_j \text{ — целое}, \quad j = \overline{1, n}. \tag{2.7}$$

Действительно, пусть Ω — многогранник, определенный условиями (2.5) и (2.6), а Ω' — множество планов нашей задачи, т. е. множество, заданное условиями (2.5)—(2.7). Ясно, что $\Omega' \subset V(\Omega)$. Следовательно, при построении с помощью алгоритма φ последовательности вершин многогранника Ω в порядке невозрастания функции $L(X)$ необходимо лишь следить за выполнением условия целочисленности (2.7). Первый целочисленный план в этой последовательности и будет оптимальным планом задачи.

§ 3. Решение задачи надежности

Напомним, что задача надежности A состоит в максимизации функции

$$F(X)=\prod_{j=1}^{n} p_j(x_j) \tag{3.1}$$

при условиях

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i, \quad i=\overline{1, m}, \tag{3.2}$$

$$x_j=0, 1, 2, \ldots, \; j=\overline{1, n}, \tag{3.3}$$

где $a_{ij}>0$, $b_i>0$ $(i=\overline{1, m}, j=\overline{1, n})$.

В качестве вспомогательной задачи $\bar{A}$ возьмем задачу максимизации функции (3.1) на множестве R, задаваемом ограничениями

$$\sum_{j=1}^{n} a_{tj}x_j \leqslant b_t, \quad x_j=0, 1, 2, \ldots, \; j=\overline{1, n},$$

где t — фиксированный произвольный индекс $(1 \leqslant t \leqslant m)$.

Такую задачу легко решить методом динамического программирования. Основное функциональное уравнение для решения задачи $\bar{A}$ имеет вид

$$f_k(z)=\max_{0\leqslant x_k\leqslant[z/a_{tk}]} [p_k(x_k) f_{k-1}(z-a_{tk}x_k)], \quad k=\overline{2, n},$$

где $0 \leqslant z \leqslant b_t$, причем $f_1(z)=p_1([z/a_{t1}])$.

Введем понятие h-оптимального плана задачи $\bar{A}$. План $\bar{X}=(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ задачи $\bar{A}$ будем называть h-оптимальным $(1 \leqslant h \leqslant n)$ и обозначать $(*, *, \ldots, *, \bar{x}_{h+1}, \ldots, \bar{x}_n)$, если

$$F(\bar{x}_1, \ldots, \bar{x}_n)=\prod_{j=h+1}^{n} p_j(x_j) f_h\left(b_t-\sum_{j=h+1}^{n} a_{tj}\bar{x}_j\right).$$

Здесь $f_h(z)$ — оптимальное значение целевой функции задачи $\bar{A}$, в которой индекс n и параметр b_t заменены соответственно на h и z. Таким образом, план $\bar{X}=(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ является h-оптимальным планом задачи $\bar{A}$, если $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_h)$ — оптимальный план задачи максимизации функции

$$\prod_{j=1}^{h} p_j(x_j)$$

при условиях

$$\sum_{j=1}^{h} a_{tj}x_j \leqslant b_t - \sum_{j=h+1}^{n} a_{tj}\bar{x}_j,$$

$$x_j = 0, 1, 2, \ldots, \quad j = \overline{1, h}.$$

Используя понятие h-оптимального плана задачи $\bar{A}$, можно описать алгоритм φ построения последовательности планов задачи $\bar{A}$ в порядке невозрастания целевой функции, т. е. такой последовательности $X_1, X_2, \ldots$, что

$$F(X_k) = \max_{X \in R_k} F(X), \quad k = 1, 2, \ldots,$$

где $R_k = R_{k-1} \setminus X_{k-1}$ $(k = 2, 3, \ldots)$, $R_1 = R$.

Алгоритм φ. 1-й шаг. Среди множества W_1 $(n-1)$-оптимальных планов

$$(*, *, \ldots, l), \quad l = 0, 1, 2, \ldots, \left[\frac{b_t}{a_{tn}}\right]$$

задачи $\bar{A}$ находим такой план

$$X_1 = (*, *, \ldots, *, s_n^1) = (s_1^1, s_2^1, \ldots, s_n^1),$$

что $F(X_1) = \max\limits_{X \in W_1} F(X)$.

k-й шаг $(k = 2, 3, \ldots)$. Множество W_{k-1} преобразуем в множество W_k по следующему правилу. Оставляем без изменения все планы множества W_{k-1}, кроме плана

$$X_{k-1} = (*, *, \ldots, *, s_p^{k-1}, \ldots, s_n^{k-1}) = (s_1^{k-1}, s_2^{k-1}, \ldots, s_n^{k-1}),$$

который заменяем системой планов

$$(*, *, \ldots, s_{q-1}, s_q^{k-1}, \ldots, s_p^{k-1}, \ldots, s_n^{k-1}),$$

где

$$q = 2, 3, \ldots, p,$$

$$s_{q-1} = 0, 1, 2, \ldots, s_{q-1}^{k-1} - 1, s_{q-1}^{k-1} + 1, \ldots, \left[\frac{b_t - \sum\limits_{j=q}^{n} a_{tj}x_j^{k-1}}{a_{t,q-1}}\right].$$

Далее, находим такой план X_k, что

$$F(X_k) = \max_{X \in W_k} F(X),$$

и переходим к следующему шагу.

Обоснование того, что алгоритм φ строит требуемую последовательность планов, проводится аналогично § 3 гл. II.

Так как в данном случае мажоранта совпадает с целевой функцией, то решением задачи A будет первый план последовательности $X_1, X_2, \ldots$, удовлетворяющий всем условиям задачи A. Проиллюстрируем работу алгоритма на примере.

Пример. Максимизировать функцию

$$F(X) = p_1(x_1)\, p_2(x_2)\, p_3(x_3)$$

при условиях

$$x_1 + 2x_2 + 3x_3 \leqslant 5,$$
$$2x_1 + x_2 + x_3 \leqslant 4,$$
$$x_j \in \{0, 1, 2\}, \quad j = \overline{1, 3},$$

где функции $p_i(x_i)$ заданы таблицей 1.

Таблица 1

x	0	1	2
$p_1(x)$	1/2	3/4	5/6
$p_2(x)$	1/3	1/2	2/3
$p_3(x)$	2/3	4/5	6/7

В качестве вспомогательной задачи $\bar{A}$ возьмем задачу максимизации функции $F(X)$ при условиях

$$2x_1 + x_2 + x_3 \leqslant 4, \quad x_j \in \{0, 1, 2\}, \quad j = 1, 2, 3.$$

Для удобства представим задачу $\bar{A}$ в виде задачи $\bar{A}_1$ максимизации функции

$$Q(X) = g_1(x_1)\, g_2(x_2)\, g_3(x_3)$$

при условиях

$$x_1 + x_2 + x_3 \leqslant 4,$$

$$x_1 \in \{0, 2, 4\}, \quad x_2 \in \{0, 1, 2\}, \quad x_3 \in \{0, 1, 2\},$$

где функции $g_i(x_i)$ заданы таблицей 2.

Таблица 2

x	0	1	2	4
$g_1(x)$	1/2	—	3/4	5/6
$g_2(x)$	4/3	1/2	2/3	—
$g_3(x)$	2/3	4/5	6/7	—

Очевидно, что задачи $\bar{A}_1$ и $\bar{A}$ эквивалентны и плану $(\bar{x}_1, \bar{x}_2, \bar{x}_3)$ задачи $\bar{A}_1$ соответствует план $x_1 = \bar{x}_1/2$, $x_2 = \bar{x}_2$, $x_3 = \bar{x}_3$ задачи $\bar{A}$.

При построении последовательности планов нам понадобится информация, получаемая в процессе решения задачи $\bar{A}_1$ методом функциональных уравнений динамического программирования. В таблице 3 приведены значения функции $f_1(z) = g_1(z)$, в таблице 4

Таблица 3

z	0	1	2	3	4
$f_1(z)$	1/2	1/2	3/4	3/4	5/6

Таблица 4

x_2 \ z	0	1	2	3	4
0	1/6	1/6	1/4	1/4	5/18
1	—	1/4	1/4	3/8	3/8
2	—	—	1/3	1/3	1/2
$f_2(z)$	1/6	1/4	1/3	3/8	1/2

приведены значения функций $g_2(x_2)f_1(z-x_2)$ и $f_2(z)$, а в таблице 5 — функций $g_3(x_3)f_2(z-x_3)$ и $f_3(z)$.

Таблица 5

x_3 \ z	0	1	2	3	4
0	1/9	1/6	2/9	1/4	1/3
1	—	2/15	1/5	4/15	3/10
2	—	—	1/7	3/4	2/7
$f_3(z)$	1/9	1/6	2/9	4/15	1/3

1-й шаг. Строим множество W_1 всех 2-оптимальных планов

$$W_1=\{B_1^1, B_1^2, B_1^3\}=\{(*, *, 0), (*, *, 1), (*, *, 2)\}.$$

Вычисляем значения целевой функции

$$F(B_1^1)=1/3, \quad F(B_1^2)=3/10, \quad F(B_1^3)=2/7.$$

Среди планов множества W_1 находим лучший план X_1. В данном случае

$$X_1=B_1^1=(*, *, 0)=(2, 2, 0),$$

т. е.

$$\bar{x}_1=2, \quad \bar{x}_2=2, \quad \bar{x}_3=0.$$

Этому плану соответствует план $x_1=1, x_2=2, x_3=0$ задачи $\bar{A}$, который удовлетворяет условиям исходной задачи и, следовательно, является оптимальным.

§ 4. Решение задачи комплектации

Унификация и стандартизация производства дает возможность комплектовать готовую продукцию из набора стандартных блоков. В этой связи возникает следующая важная практическая задача планирования выпуска изделий: какое количество изделий каждого вида следует выпускать, а также какое количество блоков каждого вида следует выпускать, чтобы добиться максимального экономического эффекта, учитывая, что определенные блоки могут рассматриваться как готовая продукция? При этом при планировании необходимо исходить из наличных производственных мощностей, а также из ограниченности имеющихся ресурсов.

Рассмотрим модель комплектации n видов изделий из m видов блоков на r заводах с использованием s видов ресурсов:

$$\sum_{j=1}^{n} F_j x_j + \sum_{i=1}^{m} p_i y_i \to \max,$$

$$\sum_{j=1}^{n} d_{ij} x_j + y_i = \sum_{l=1}^{r} v_{il}, \quad i = \overline{1, m},$$

$$x_j = \sum_{l=1}^{r} z_{jl}, \quad j = \overline{1, n},$$

$$\sum_{i \in G_l} a_{ik} v_{il} + \sum_{j \in H_l} e_{kj} z_{jl} \leqslant b_{kl}, \quad k = \overline{1, s}, \quad l = \overline{1, r},$$

$$0 \leqslant x_j \leqslant v_j, \quad 0 \leqslant y_i \leqslant u_i, \quad j = \overline{1, n}, \ i = \overline{1, m},$$

$$0 \leqslant v_{il} \leqslant V_{il}, \ 0 \leqslant z_{jl} \leqslant M_{jl}, \ i = \overline{1, m}, \ j = \overline{1, n}, \ l = \overline{1, r},$$

x_j, y_i, v_{il}, z_{jl} — целые числа.

Здесь искомые величины:

x_j — количество изделий j-го вида, собираемых на заводах;

y_i — количество блоков i-го вида, реализуемых «врассыпную»;

v_{il} — количество блоков i-го вида, производимых на l-м заводе;

z_{jl} — количество изделий j-го вида, собираемых на l-м заводе.

Известные величины:

F_j — экономический эффект от реализации одного изделия j-го вида;

p_i — экономический эффект от реализации одного блока i-го вида;

d_{ij} — количество блоков i-го вида, необходимое для комплектации одного изделия j-го вида;

a_{ik} — количество ресурса k-го вида, необходимое для изготовления одного блока i-го вида;

e_{kj} — количество ресурса k-го вида, необходимое для сборки одного изделия j-го вида;

b_{kl} — запас ресурса k-го вида на l-м заводе;

H_l — множество видов изделий, собираемых на l-м заводе, $H_l \subset \{\overline{1, n}\}$;

G_l — множество видов блоков, производимых на l-м заводе, $G_l \subset \{\overline{1, m}\}$;

V_{il}, M_{jl} — предельные мощности l-го завода по выпуску, соответственно, блоков и изделий;

v_j, u_i — верхние границы выпуска соответственно изделий и блоков, реализуемых «врассыпную».

Особенностью рассматриваемой задачи является то, что правые части некоторых из ограничений содержат переменные, явно не входящие в целевую функцию.

Для упрощения дальнейшего изложения сформулируем более общую задачу, которая содержит в себе в качестве частного случая рассмотренную модель комплектации.

Задача A.

$$F(X) = \sum_{j=1}^{n} f_j(x_j) \to \max, \tag{4.1}$$

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant \sum_{l=1}^{r} b_{il} y_l, \quad i = \overline{1, m}, \tag{4.2}$$

$$\sum_{l=1}^{r} c_{kl} y_l \leqslant d_k, \quad k = \overline{1, s}, \tag{4.3}$$

$$0 \leqslant x_j \leqslant v_j, \quad j = \overline{1, n}, \tag{4.4}$$

$$0 \leqslant y_l \leqslant u_l, \quad l = \overline{1, r}, \tag{4.5}$$

$$x_j \text{ — целые числа}, \quad j = \overline{1, n}, \tag{4.6}$$

$$y_l \text{ — целые числа}, \quad l = \overline{1, r}, \tag{4.7}$$

где коэффициенты a_{ij}, c_{kl}, b_{il}, d_k — целые неотрицательные числа.

Запишем задачу A в несколько иной форме. Через U обозначим множество векторов $Y = (y_1, y_2, \ldots, y_r)$, удовлетворяющих условиям (4.3), (4.5), (4.7), а через $P(Y)$ — множество векторов $X = (x_1, x_2, \ldots, x_n)$, которые удовлетворяют условиям (4.2), (4.4), (4.6) при фиксированном Y. Ясно, что задача A эквивалентна задаче

$$\max_{X \in P} F(X), \tag{4.8}$$

где $P = \bigcup_{Y \in U} P(Y)$.

Прежде чем перейти к изложению схемы решения задачи (4.8), заметим, что для построения последовательности $Y_1, Y_2, \ldots$ элементов множества U в порядке невозрастания функции

$$L(Y) = \sum_{i=1}^{m} \sum_{l=1}^{r} b_{il} y_l \tag{4.9}$$

следует строить последовательность $\bar{Y}_1, \bar{Y}_2, \ldots$ элементов множества W, задаваемого ограничениями (4.5), (4.7) и

$$\sum_{k=1}^{s} \sum_{l=1}^{r} c_{kl} y_l \leqslant \sum_{k=1}^{s} d_k, \tag{4.10}$$

в порядке невозрастания функции $L(Y)$ и из нее удалять элементы, не удовлетворяющие условию (4.3).

Очевидно, что справедливо неравенство

$$\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_j \leqslant L(Y_1). \tag{4.11}$$

Следовательно, в качестве расширения множества P планов задачи (4.8) можно взять множество R, определяемое условиями (4.4), (4.6), (4.11). Поэтому решение задачи (4.8) состоит в построении последовательности $X_1, X_2, \ldots$ элементов множества R в порядке невозрастания $F(X)$ до получения элемента X^*, принадлежащего множеству P, который и будет оптимальным планом задачи (4.8).

Критерий принадлежности элементов последовательности $X_1, X_2, \ldots$ множеству P формулируется следующим образом. Вектор $X = (x_1, x_2, \ldots, x_m)$ принадлежит множеству P тогда и только тогда, когда существует индекс $q \geqslant 1$ такой, что

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant \sum_{l=1}^{r} b_{il} y_l^q, \quad i = \overline{1, m},$$

где $(y_1^q, y_2^q, \ldots, y_r^q) = Y_q = Y(X)$. Следовательно, оптимальный план исходной задачи A имеет вид

$$\{X^*, Y^*\} = (x_1^*, x_2^*, \ldots, x_n^*, y_1^*, y_2^*, \ldots, y_r^*),$$

где $Y^* = Y(X^*) = (y_1^*, y_2^*, \ldots, y_r^*)$.

Если в задаче A не налагать условия целочисленности на переменные y_l, то получим задачу комплектации в непрерывной постановке (задача B). К решению этой задачи метод построения последовательности планов применяется несколько иначе, чем к решению задачи A. В качестве вспомогательной задачи можно взять задачу максимизации $F(X)$ на множестве R^*, задаваемом условиями (4.4), (4.6) и

$$\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}x_j \leqslant L(Y^*),$$

где Y^* — оптимальный план задачи максимизации функции (4.9) при условиях (4.3), (4.5).

Алгоритм решения задачи B состоит в построении последовательности $X_1, X_2, \ldots$ элементов множества R^* в порядке невозрастания $F(X)$ до получения элемента $X_p = (x_1^p, \ldots, x_n^p)$ $(p \geqslant 1)$, для которого существует неотрицательное решение $Y_p = (y_1^p, \ldots, y_r^p)$ системы неравенств (4.3), (4.5) и

$$\sum_{j=1}^{n} a_{ij}x_j^p \leqslant \sum_{l=1}^{r} b_{il}y_l, \quad i = \overline{1, m}.$$

Оптимальным планом задачи B будет вектор

$$(X_p, Y_p) = (x_1^p, x_2^p, \ldots, x_n^p, y_1^p, y_2^p, \ldots, y_r^p).$$

§ 5. Экстремальные задачи на множестве подстановок

Пусть H — непустое подмножество симметрической группы S_n подстановок, действующих на множестве $N = \{1, 2, \ldots, n\}$. Пусть задана функция $F(h)$ на множестве подстановок H. Задача минимизации (максимизации) функции $F(h)$ на множестве подстановок

H состоит в нахождении такой подстановки $h_0 \in H$, что

$$F(h_0) \leqslant F(h) \quad (F(h_0) \geqslant F(h))$$

для всякой подстановки $h \in H$.

Подобные задачи возникают всюду, где требуется определить наилучший в том или ином смысле порядок выполнения некоторой конечной системы операций. С каждым возможным порядком естественным образом ассоциируется перестановка $(i_1, i_2, \ldots, i_n)$ чисел $1, 2, \ldots, n$, указывающая последовательность выполнения операций. В свою очередь, каждой перестановке $(i_1, i_2, \ldots, i_n)$ соответствует подстановка $\begin{pmatrix} 1 & 2 & \ldots & n \\ i_1 & i_2 & \ldots & i_n \end{pmatrix}$ из симметрической группы S_n. Таким образом, класс экстремальных задач на множествах подстановок содержит такие классические задачи оптимального упорядочения, как задача о назначениях, задачи теории расписаний, задача о бродячем торговце, о круговой расстановке станков и др.

В настоящем параграфе мы приведем примеры применения метода построения последовательности планов к решению некоторых экстремальных задач на множестве подстановок.

1. Минимизация линейной формы. Пусть $\pi = \langle a, b\rangle$ — пара n-мерных вещественных векторов $a = (a_1, a_2, \ldots, a_n)$ и $b = (b_1, b_2, \ldots, b_n)$, h — произвольная подстановка из симметрической группы S_n подстановок, действующих на множестве $N = \{1, 2, \ldots \ldots, n\}$. Положим

$$F_\pi(h) = \sum_{i=1}^{n} a_i b_{h(i)},$$

где $h = \begin{pmatrix} 1 & 2 & \ldots & n \\ h(1) & h(2) & \ldots & h(n) \end{pmatrix}$.

Пусть $H \subseteq S_n$.

Задача A. Требуется найти в H такую подстановку h_0, что

$$F_\pi(h_0) = \min_{h \in H} F_\pi(h).$$

Ясно, что решение этой задачи можно осуществить путем построения последовательности подстановок множества S_n в порядке неубывания функции $F_\pi(h)$

до получения подстановки, принадлежащей множеству H.

Прежде чем приступить к описанию алгоритма построения указанной последовательности, введем некоторые необходимые в дальнейшем обозначения и определения.

Для пары (G, M) такой, что $G=\{g_1, g_2, \ldots, g_k\} \subset N$, $M = \{m_1, m_2, \ldots, m_l\} \subset N$, $G \cap M = \varnothing$, $|G| < n$, определим множество подстановок

$$P(G, M)= \begin{cases} \{h \in S_n \mid h(i)=g_i, \quad i=\overline{1, |G|}, \quad h(|G|+1) \notin M\}, & \text{если} \quad G \neq \varnothing, \\ \{h \in S_n \mid h(1) \notin M\}, & \text{если} \quad G=\varnothing. \end{cases}$$

Через $h(G, M)$ будем обозначать подстановку, оптимальную на множестве $P(G, M)$, т. е. такую, что

$$F_\pi(h(G, M)) = \min_{h \in P(G, M)} F_\pi(h).$$

Далее будет показано, как строить такую подстановку.

Паре (G, M) и произвольной подстановке $h^0 \in P(G, M)$ поставим в соответствие множество пар

$$O(G, M, h^0)= \begin{cases} \bigcup\limits_{\nu=1}^{n-|G|-1} (G_\nu, M_\nu), & \text{если } |G| \leqslant n-2, \ |M| \leqslant n-|G|-2; \\ \bigcup\limits_{\nu=2}^{n-|G|-1} (G_\nu, M_\nu), & \text{если } |G| \leqslant n-3, \ |M| = n-|G|-1; \\ \varnothing & \text{в остальных случаях,} \end{cases}$$

где

$$M_\nu = \begin{cases} M \bigcup \{h^0(|G|+1)\} & \text{при} \quad \nu=1, \\ \{h^0(|G|+\nu)\} & \text{при} \quad \nu=\overline{2, n-|G|-1}, \end{cases}$$

$$G_\nu = \begin{cases} G & \text{при} \quad \nu=1, \\ \{g_1, g_2, \ldots, g_k, h^0(|G|+1), \ldots, h^0(|G|+\nu-1)\} & \\ & \text{при} \quad \nu=\overline{2, n-|G|-1}. \end{cases}$$

Опишем теперь алгоритм φ построения последовательности подстановок $h^{(1)}, h^{(2)}, \ldots$ вспомогательной задачи $\bar{A}$ в порядке неубывания целевой функции.

Алгоритм φ. На подготовительном шаге образуем множество $W_1 = \{(\varnothing, \varnothing)\}$.

k-й шаг ($k = 1, 2, \ldots$). Определим пару $(G_{r_0(k)}, M_{r_0(k)}) \in W_k$ такую, что

$$F_\pi(h(G_{r_0(k)}, M_{r_0(k)})) = \min_{(G, M) \in W_k} F_\pi(h(G, M)).$$

Полагаем $h^{(k)} = h(G_{r_0(k)}, M_{r_0(k)})$, образуем множество

$$W_{k+1} = \{W_k \setminus (G_{r_0(k)}, M_{r_0(k)})\} \bigcup O\left(G_{r_0(k)}, M_{r_0(k)}, h^{(k)}\right). \quad (5.1)$$

Если $W_{k+1} = \varnothing$, алгоритм заканчивает работу, в противном случае переходим к следующему шагу.

Займемся обоснованием алгоритма φ, т. е. покажем, что он строит последовательность подстановок $h^{(1)}, h^{(2)}, \ldots$, удовлетворяющую условиям

$$F_\pi(h^{(k)}) = \min_{h \in M_k} F_\pi(h), \quad k = 1, 2, \ldots,$$

где $M_1 = S_n$, $M_k = M_{k-1} \setminus \{h^{(k-1)}\}$ $(k = 2, 3, \ldots)$.

Непосредственно из определения множества $O(G, M, h^0)$ имеем

$$P(G_i, M_i) \bigcap P(G_j, M_j) = \varnothing,$$

если

$$(G_i, M_i) \neq (G_j, M_j), \quad (G_i, M_i), (G_j, M_j) \in O(G, M, h^0).$$

Кроме того, $h^0 \notin P\ (G, M)$ для любой пары $(G, M) \in O(G, M, h^0)$. Поэтому получаем

$$P(G, M) \setminus \{h^0\} = \bigcup_{(G, M) \in O(G, M, h^0)} P(G, M).$$

Отсюда, учитывая (5.1), имеем

$$S_n \setminus \bigcup_{i=1}^{k-1} \{h^i\} = \bigcup_{(G, M) \in W_k} P(G, M), \quad k = 2, 3, \ldots$$

Следовательно, для каждого $k \geqslant 1$

$$\min_{h \in M_k} F_\pi(h) = \min_{h \in S_n \setminus \bigcup_{i=1}^{k-1} \{h^{(i)}\}} F_\pi(h) =$$

$$= \min_{h \in \bigcup_{(G, M) \in W_k} P(G, M)} F_\pi(h) = \min_{(G, M) \in W_k} \min_{h \in P(G, M)} F_\pi(h) =$$

$$= \min_{(G, M) \in W_k} F_\pi(h(G, M)) = F_\pi(h^{(k)}),$$

что и требовалось доказать.

Покажем теперь, как находить подстановку $h(G, M)$, т. е. подстановку, оптимальную на множестве $P(G, M)$. Хорошо известен следующий результат: если

$$a_1 \leqslant a_2 \leqslant \ldots \leqslant a_n,$$
$$b_1 \geqslant b_2 \geqslant \ldots \geqslant b_n,$$

то

$$\min_{h \in S_n} F_\pi(h) = F_\pi(h_0),$$

где

$$h_0 = h(\varnothing, \varnothing) = \begin{pmatrix} 1 & 2 & \ldots & n \\ 1 & 2 & \ldots & n \end{pmatrix}.$$

В предположении, что числа $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n$ упорядочены указанным образом, очевидно, что

$$h(G, M) = \begin{pmatrix} 1 & 2 & \ldots & |G| & |G|+1 & |G|+2 & \ldots & n \\ g_1 & g_2 & \ldots & g_{|G|} & g & i_{|G|+2} & \ldots & i_n \end{pmatrix},$$

где

$$g = \min\{i \mid i \in N \setminus (G \cup M)\},$$
$$\{i_{|G|+2}, \ldots, i_n\} = (N \setminus G) \setminus \{g\},$$
$$i_{|G|+2} < i_{|G|+3} < \ldots < i_n.$$

Проиллюстрируем работу алгоритма ψ на примере.

Пример. В качестве множества H рассмотрим множество всех полных циклов C симметрической группы подстановок S_6. Пусть

$$a = (a_1, a_2, a_3, a_4, a_5, a_6) = (2, 3, 6, 8, 9, 11),$$
$$b = (b_1, b_2, b_3, b_4, b_5, b_6) = (7, 6, 5, 10, 3, 2).$$

Рассматриваемая задача состоит в нахождении подстановки $h_0 \in C$ такой, что

$$F(h_0) = \min_{h \in C} F(h).$$

Образуем множество $W_1 = \{(\varnothing, \varnothing)\}$.

1-й шаг. Находим пару $(G_{r_0(1)}, M_{r_0(1)}) = (\varnothing, \varnothing) \in W_1$ и полагаем

$$h^{(1)} = h(\varnothing, \varnothing) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 1 & 2 & 3 & 5 & 6 \end{pmatrix}.$$

Так как $h^{(1)} \notin C$, то образуем множество пар

$$W_2 = \{(\varnothing;\ 4),\ (4;\ 1),\ (4,\ 1;\ 2),\ (4,\ 1,\ 2;\ 3)\}.$$

Значения целевой функции на подстановках $h(G, M)$, где $(G, M) \in W_2$, равны соответственно 169, 169, 168, 168.

2-й шаг.

$$(G_{r_0(2)}, M_{r_0(2)}) = (4,\ 1;\ 2) \in W_2,$$

$$h^{(2)} = h(4,\ 1;\ 2) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 1 & 3 & 2 & 5 & 6 \end{pmatrix} \notin C.$$

Образуем множество

$$W_3 = \{(\varnothing;\ 4),\ (4;\ 1),\ (4,\ 1,\ 2;\ 3),\ (4,\ 1;\ 2,\ 3)\}.$$

Значения целевой функции соответственно равны 169, 169, 168, 174.

3-й шаг. Определяем пару

$$(G_{r_0(3)}, M_{r_0(3)}) = (4,\ 1,\ 2;\ 3) \in W_3,$$

$$h^{(3)} = h(4,\ 1,\ 2;\ 3) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 1 & 2 & 5 & 3 & 6 \end{pmatrix}.$$

Так как $h^{(3)} \notin C$, то образуем множество

$$W_4 = \{(\varnothing;\ 4),\ (4;\ 1),\ (4,\ 1;\ 2,\ 3),\ (4,\ 1,\ 2;\ 3,\ 5),\ (4,\ 1,\ 2,\ 5;\ 3)\}.$$

Значения целевой функции соответственно равны 169, 169, 174, 170, 174.

4-й шаг.

$$(G_{r_0(4)}, M_{r_0(4)}) = (\varnothing;\ 4) \in W_4,$$

$$h^{(4)} = h(\varnothing;\ 4) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 3 & 5 & 6 \end{pmatrix} \notin C.$$

Образуем множество

$$W_5 = \{(4;\ 1),\ (4,\ 1;\ 2,\ 3),\ (4,\ 1,\ 2;\ 3,\ 5),\ (\varnothing;\ 4,\ 1),\ (4,\ 1,\ 2,\ 5;\ 3)\}.$$

Значения целевой функции соответственно равны 169, 174, 170, 173, 174.

5-й шаг.

$$\left(G_{r_0(5)},\ M_{r_0(5)}\right) = (4;\ 1) \in W_5,$$

$$h^{(5)} = h(4;\ 1) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 2 & 1 & 3 & 5 & 6 \end{pmatrix} \notin C.$$

Образуем множество

$$W_6 = \{(4,\ 1;\ 2,\ 3),\ (4,\ 1,\ 2;\ 3,\ 5),\ (\varnothing;\ 4,\ 1),\ (4;\ 1,\ 2),\ (4,\ 1,\ 2,\ 5;\ 3)\}.$$

Значения целевой функции соответственно равны 174, 171, 173, 174, 174.

6-й шаг.

$$\left(G_{r_0(6)},\ M_{r_0(6)}\right) = (4,\ 1,\ 2;\ 3,\ 5) \notin W_6,$$

$$h^{(6)} = h(4,\ 1,\ 2;\ 3,\ 5) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 1 & 2 & 6 & 3 & 5 \end{pmatrix}.$$

Так как $h^{(6)} \in C$, то алгоритм заканчивает работу. Минимум линейной форме на множестве всех полных циклов доставляет подстановка $h_0 = (1,\ 4,\ 6,\ 5,\ 3,\ 2)$, $F(h_0) = 171$.

2. Минимизация скалярного произведения двух матриц. Пусть $A = \| a_{ij} \|$, $B = \| b_{ij} \|$ — квадратные матрицы порядка n, все элементы которых положительны, h — подстановка из S_n. Определим функцию $F(h)$ следующим образом:

$$F(h) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{h(i)\,h(j)}.$$

Задача минимизации скалярного произведения двух матриц состоит в нахождении такой подстановки h_0, что

$$F(h_0) = \min_{h \in H} F(h).$$

Рассмотрим функцию

$$Q(h)=\sum_{i=1}^{n} a_i^{(p)}\, b_{h(i)}^{(p)},$$

где

$$a_i^{(p)}=\left(\sum_{j=1}^{n} a_{ij}^p\right)^{1/p},\quad b_i^{(p)}=\left(\sum_{j=1}^{n} b_{ij}^q\right)^{1/q},\quad i=\overline{1, n},$$

$$1/p+1/q=1,\quad p<1.$$

Докажем, что $Q(h)$ является минорантой функции $F(h)$. Для этого воспользуемся следующим результатом.

Неравенство Гёльдера. Если a_i, $b_i \geq 0$, $1/p+1/q=1$, $p<1$ $(p \neq 0)$, то

$$\sum_{i=1}^{n} a_i b_i \geqslant \left(\sum_{i=1}^{n} a_i^p\right)^{1/p} \left(\sum_{i=1}^{n} b_i^q\right)^{1/q}.$$

Неравенство заменяется на противоположное, если $p>1$.

Применяя это неравенство, получаем

$$F(h)=\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}\, b_{h(i)\ h(j)} \geqslant$$

$$\geqslant \sum_{i=1}^{n}\left(\sum_{j=1}^{n} a_{ij}^p\right)^{1/p}\left(\sum_{j=1}^{n} b_{h(i)\ h(j)}^q\right)^{1/q} =$$

$$=\sum_{i=1}^{n}\left(\sum_{j=1}^{n} a_{ij}^p\right)^{1/p}\left(\sum_{j=1}^{n} b_{h(i)\,j}^q\right)^{1/q}=\sum_{i=1}^{n} a_i^{(p)}\, b_{h(i)}^{(p)}$$

для любой подстановки $h \in S_n$.

Так как миноранта является линейной формой, то построение последовательности подстановок множества S_n в порядке неубывания функции $Q(h)$ можно осуществить с помощью алгоритма, изложенного в предыдущем пункте.

3. Минимизация суммы линейных форм. Пусть, как и ранее, $\pi=\langle a, b\rangle$ — пара n-мерных вещественных векторов $a=(a_1, a_2, \ldots, a_n)$, $b=(b_1, b_2, \ldots, b_n)$ и $F_\pi(h)=\sum_{i=1}^{n} a_i\, b_{h(i)}$. Пусть теперь задано r пар векторов

$$\pi_j=\langle a^{(j)}, b^{(j)}\rangle, \quad j=\overline{1, r},$$

$$a^{(j)}=(a_{1j}, a_{2j}, \ldots, a_{nj}), b^{(j)}=(b_{1j}, b_{2j}, \ldots, b_{nj}), \quad j=\overline{1, r};$$

$$\Omega=(\pi_1, \pi_2, \ldots, \pi_r).$$

Определим функцию

$$F_\Omega(h)=\sum_{j=1}^{r} F_{\pi_j}(h).$$

Задача минимизации суммы линейных форм состоит в нахождении для некоторого непустого подмножества $H \subseteq S_n$ такой подстановки h_0, что

$$F_\Omega(h_0)=\min_{h\in H} F_\Omega(h).$$

При решении этой задачи в качестве расширения можно взять все множество S_n, а в качестве миноранты — функцию

$$Q_\Omega(h)=F_{\pi^*}(h)=\sum_{i=1}^{n} a_i^* b_{h(i)}^*.$$

Здесь $\pi^*=\langle a^*, b^*\rangle$ — пара n-мерных векторов с координатами

$$a_i^*=\left(\sum_{j=1}^{r} a_{ij}^p\right)^{1/p}, \quad b_i^*=\left(\sum_{j=1}^{r} b_{ij}^q\right)^{1/q}, \quad i=\overline{1, n},$$

где $1/p+1/q=1$, $p<1$.

То, что линейная форма $Q_\Omega(h)$ является минорантой $F_\Omega(h)$, непосредственно следует из неравенства Гёльдера.

4. Разрезание графа на подграфы с фиксированным числом вершин. При проектировании технических устройств важной является задача компоновки модулей в определенные конструктивные единицы, которая сводится к следующей задаче.

Пусть Γ — полный n-вершинный неориентированный граф, каждому ребру (i, j) которого приписан вес a_{ij}, а $P=\{k_1, k_2, \ldots, k_r\}$ — набор целых положительных чисел таких, что $\sum_{s=1}^{r} k_s=n$.

Требуется разрезать граф Γ на r подграфов с числом вершин $k_1, k_2, \ldots, k_r$ в каждом из них и максимальным суммарным весом соединительных ребер. Эта задача, в свою очередь, сводится к нахождению подстановки из S_n, максимизирующей функцию

$$F(A, P, h) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{h(i)\ h(j)},$$

где $A = \|a_{ij}\|_{n\times n}$ — симметрическая матрица весов графа Г, а матрица B имеет вид

$$B = \| b_{ij} \|_{n\times n} = \begin{vmatrix} E_1 & 0 & \dots & 0 \\ 0 & E_2 & \dots & 0 \\ . & . & . & . \\ 0 & 0 & \dots & E_r \end{vmatrix}.$$

Здесь E_i — матрица порядка $k_i \times k_i$, состоящая из единиц.

Действительно, любая подстановка h из S_n определяет такое разрезание графа Г на r подграфов, при котором s-й подграф содержит вершины множества $h^{-1}(I_s)$, где

$$I_s = \left\{ \sum_{i=1}^{s-1} k_i + 1, \dots, \sum_{i=1}^{s} k_i \right\}, \quad s = \overline{1, r}.$$

Обратно, любое разрезание графа Г, s-й подграф которого содержит множество вершин M_s, задает множество подстановок $\{h \in S_n \mid h^{-1}(I_s) = M_s,\ s = \overline{1, r}\}$, каждая из которых определяет данное разрезание.

Для решения сформулированной задачи можно воспользоваться подходом, описанным ранее в пункте 2.

§ 6. Решение многокритериальных задач

Большинство практических задач принятия решений имеют несколько противоречивых критериев, образующих вектор эффективности. Поэтому нахождение множества неулучшаемых планов является важным этапом решения многих практических задач многокритериальной оптимизации.

Так в задаче формирования производственной программы предприятия из портфеля заказов (§ 6, гл. 1) фактически имеются четыре критерия, четыре цели:

максимизация прибыли и номенклатуры, минимизация затрат труда и максимизация загрузки оборудования. В силу противоречивости этих критериев нет возможности найти план, оптимальный сразу по всем критериям, и поэтому возникает потребность в нахождении таких планов, каждый из которых нельзя было бы улучшить ни по одному из критериев, не ухудшив по другим. Такие планы мы будем называть *неулучшаемыми* (в литературе они иногда называются *оптимальными по Парето*).

В настоящем параграфе описывается алгоритм построения множества неулучшаемых планов, основанный на методе построения последовательности планов. Остановимся вначале на вопросах структуры этого множества. Прежде всего условимся, что каждый критерий минимизируется. Очевидно, что это не исключает общности, поскольку, если $f(x) \to \max$, то $-f(x) \to$ $\to \min$.

Пусть качество планов некоторого конечного множества M оценивается m-мерной вектор-функцией эффективности

$$f(x) = \{f_1(x), f_2(x), \ldots, f_m(x)\},$$

где $f_i(x)$ $(i = \overline{1, m})$ являются заданными действительными функциями от $x \in M$. План $x_0 \in M$ называется *неулучшаемым в* M, если не существует такого плана $x_1 \in M$, что

$$f_i(x_1) \leqslant f_i(x_0), \quad i = \overline{1, m},$$

причем хотя бы одно из неравенств строгое. Множество всех неулучшаемых в M планов обозначим через M^*. Подмножество P множества M будем называть *отделимым по функции* $q(x)$, если имеет место неравенство

$$\max_{x \in P} q(x) < \min_{x \in M \setminus P} q(x).$$

Очевидно, что для всякого плана $x_0 \in M$ множество $M_i(x_0) = \{x \mid f_i(x) \leqslant f_i(x_0)\}$ является отделимым по функции $f_i(x)$ при любом $i = \overline{1, m}$.

Теорема. *Пусть* $M_1, M_2, \ldots, M_m$ — *подмножества множества* M, *отделимые, соответственно, по* *функциям* $f_1(x), f_2(x), \ldots, f_m(x)$. *Если* $\bigcap_{i=1}^{m} M_i \neq \varnothing$, *то*

1) $$M^* \subset \bigcup_{i=1}^{m} M_i;$$

2) $$\left\{\bigcap_{i=1}^{m} M_i\right\} \cap M^* \neq \varnothing.$$

Доказательство. 1) Достаточно показать, что произвольный план $x_1 \notin \bigcup_{i=1}^{m} M_i$ может быть улучшен. Действительно, так как $x_1 \in M \setminus M_i$ $(i = \overline{1, m})$, то

$$f_i(x_1) \geqslant \min_{x \in M \setminus M_i} f_i(x), \quad i = \overline{1, m}.$$

Вместе с тем для всякого плана $x_0 \in M_0 = \bigcap_{i=1}^{m} M_i$ справедливы неравенства

$$f_i(x_0) \leqslant \max_{x \in M_i} f_i(x), \quad i = \overline{1, m}.$$

Поэтому на основании условий теоремы имеем

$$f_i(x_0) < f_i(x_1), \quad i = \overline{1, m}.$$

Следовательно, план x_1 может быть улучшен планом x_0, что доказывает первую часть теоремы.

2) Пусть x_0 — некоторый фиксированный план множества

$$M_0 = \bigcap_{i=1}^{m} M_i.$$

Рассмотрим множество $W(x_0) = \bigcap_{i=1}^{m} M_i(x_0)$. Для любого плана $x_1 \in W(x_0)$ справедливы соотношения

$$f_i(x_1) \leqslant \max_{x \in M_i} f_i(x), \quad i = \overline{1, m}.$$

Поэтому в силу отделимости множеств $M_1, M_2, \ldots$ $\ldots, M_m$ имеем

$$f_i(x_1) < \min_{x \in M \setminus M_i} f_i(x), \quad i = \overline{1, m}.$$

Отсюда вытекает, что $x_1 \in M_0$. Следовательно, $W(x_0) \subset M_0$. Вместе с тем очевидно, что всякий план

$x_2 \in W(x_0)$, неулучшаемый в $W(x_0)$, будет неулучшаемым и в M. Поэтому $W(x_0) \cap M^* \neq \varnothing$, и, следовательно, $M_0 \cap M^* \neq \varnothing$. Теорема доказана.

Следствие. $M^* \subset \bigcup_{i=1}^{m} M_i(x)$ *для любого плана* $x \in M$.

Из следствия, в частности, вытекает, что все неулучшаемые планы содержатся и в множестве $N = \bigcup_{i=1}^{m} M_i(x^*)$, где x^* — план, доставляющий минимум функции $f_1(x)$ на множестве M.

Приводимый ниже алгоритм, основанный на построении последовательности планов, и осуществляет путем направленного перебора выделение всех неулучшаемых планов из множества N.

Алгоритм построения множества неулучшаемых планов. На подготовительном этапе находим план x^*, для которого выполняются соотношения

$$f_i(x^*) = \min_{x \in R_i} f_i(x), \quad i = \overline{1, m},$$

где

$$R_1 = M,$$
$$R_i = \{\bar{x} \mid \bar{x} \in R_{i-1}, f_i(\bar{x}) = \min_{x \in R_{i-1}} f_i(x)\},$$
$$i = \overline{2, m}.$$

Полагаем $M_0^* = \varnothing$.

i-й этап ($i = \overline{1, m}$). Полагаем $M_i^1 = M$, $N_i^0 = \varnothing$.

k-й шаг ($k \geqslant 1$). Строим множество

$$P_{ik} = \{\bar{x} \mid f_i(\bar{x}) = \min_{x \in M_i^k} f_i(x)\}.$$

Из множества P_{ik} выделяем множество планов P_{ik}^*, неулучшаемых в P_{ik}. Из планов множества P_{ik}^* таких, что $f_j(x) > f_j(x^*)$ $(j = \overline{1, i-1})$, выбираем планы, которые не могут быть улучшены ни одним из планов множества

$$\left\{ x \mid x \in \left(\bigcup_{j=0}^{i-1} M_j^* \right) \cup \left(\bigcup_{s=0}^{k-1} N_i^s \right), \right.$$
$$\left. f_i(x) \leqslant f_i^k = \min_{x \in M_i^k} f_i(x) \right\},$$

и формируем из них множество N_i^k. Полагаем $N_i^k = \varnothing$, если таких планов нет.

Сравниваем числа f_i^k и $f_i(x^*)$. Если $f_i^k < f_i(x^*)$, то полагаем $M_i^{k+1} = M_i^k \setminus P_{ik}$ и переходим к $(k+1)$-му шагу. Если $f_i^k = f_i(x^*)$, то полагаем $M_i^* = \bigcup_{s=1}^{k} N_i^s$. Сравниваем числа i и m. Если $i < m$, то переходим к $(i+1)$-му этапу. Если $i = m$, то алгоритм заканчивает работу и

$$M^* = \bigcup_{i=1}^{m} M_i^*.$$

В случае двухкритериальной задачи, для которой качество планов оценивается функциями $f_1(x)$ и $f_2(x)$, изложенный выше алгоритм значительно упрощается и состоит в следующем.

Алгоритм для случая двух критериев. На предварительном этапе работы находим план x^*, для которого выполняются соотношения

$$f_2(x^*) = \min_{x \in M} f_2(x), \quad f_1(x^*) = \min_{x \in R^*} f_1(x),$$

где

$$R^* = \{\bar{x} \mid f_2(\bar{x}) = \min_{x \in M} f_2(x)\}.$$

Очевидно, что план x^* является неулучшаемым в M. Полагаем $f_2^0 = \infty$, $M_1^1 = M$.

k-й шаг ($k \geqslant 1$). Строим множество

$$P_{1k} = \{\bar{x} \mid f_1(\bar{x}) = \min_{x \in M_1^k} f_1(x)\}$$

и из него выделяем подмножество

$$P_{1k}^* = \{\bar{x} \mid f_2(\bar{x}) = \min_{x \in P_{1k}} f_2(x)\}.$$

Сравниваем величины $g_2^k = \min_{x \in P_{1k}} f_2(x)$ и f_2^{k-1}. Если $g_2^k < f_2^{k-1}$, то полагаем $N_1^k = P_{1k}^*$, $f_2^k = g_2^k$. Если $g_2^k \geqslant f_2^{k-1}$, то полагаем $N_1^k = \varnothing$, $f_2^k = f_2^{k-1}$.

Сравниваем числа $f_1^k = \min\limits_{x \in M_1^k} f_1(x)$ и $f_1(x^*)$. Если $f_1^k < f_1(x^*)$, то полагаем $M_1^{k+1} = M_1^k \setminus P_{1k}$ и переходим к $(k+1)$-му шагу. Если $f_1^k = f_1(x^*)$, то алгоритм заканчивает работу и

$$M^* = \bigcup_{s=1}^{k} N_1^s.$$

Проиллюстрируем работу алгоритма для двухкритериальной задачи на примере.

Пример. Пусть множество M векторов (x_1, x_2, x_3) задается ограничениями

$$x_1 + x_2 + x_3 = 5, \quad x_j \in \{0, 1, 2, 3\}, \quad j = \overline{1, 3}.$$

Качество планов оценивается двумерной вектор-функцией

$$f(x) = \{f_1(x), f_2(x)\},$$

где

$$f_1(x) = g_1^1(x_1) + g_2^1(x_2) + g_3^1(x_3), \quad f_2(x) = \\ = g_1^2(x_1) + g_2^2(x_2) + g_3^2(x_3),$$

а функции $g_s^l(x_s)$ задаются таблицей 6.

Таблица 6

x	0	1	2	3
$g_1^1(x)$	0	15	28	40
$g_2^1(x)$	0	14	29	41
$g_3^1(x)$	0	13	27	42
$g_1^2(x)$	0	20	27	29
$g_2^2(x)$	0	15	19	24
$g_3^2(x)$	0	16	18	25

На предварительном этапе работы алгоритма находим план x^*. Для нахождения x^* необходимо методом динамического программирования решить задачу минимизации $f_2(x)$ при условиях

$$x_1+x_2+x_3=5,\ x_1,\ x_2,\ x_3 \in \{0,\ 1,\ 2,\ 3\}.$$

В нашем случае $x^* = (0,\ 3,\ 2)$ и, следовательно, $f_1^0 = f_1(x^*) = 68$.

Полагаем $f_2^0 = \infty$.

1-й шаг. С помощью алгоритма, изложенного в § 3 гл. II, строим множество $P_{11}=\{r_1,\ r_2\}$, где $r_1=(3,\ 1,\ 1)$, $r_2=(3,\ 0,\ 2)$. Так как $f_2(r_1)=60$, $f_2(r_2)=47$, то $P_{11}^*=\{r_2\}$.

Сравниваем величины $g_2^1 = f_2(r_2) = 47$ и $f_2^0 = \infty$. Так как $g_2^1 < f_2^0$, то полагаем $N_1^1 = P_{11}^* = \{r_2\}$, $f_2^1 = 47$ и переходим к сравнению чисел $f_1^1 = 67$ и $f_1^0 = 68$. Так как $f_1^1 < f_1^0$, то переходим ко 2-му шагу.

2-й шаг. Строим множество $P_{12}=\{r_3\}$, $r_3=(0,\ 2,\ 3)$ и выделяем множество $P_{12}^* = \{r_3\}$. Так как $g_2^2 = f_2(r_3) = 42 < f_2^1 = 47$, то полагаем $N_1^2 = P_{12}^* = \{r_3\}$, $f_2^2 = g_2^2 = 42$ и переходим к сравнению чисел $f_1^2 = 68$ и $f_1^0 = 68$. Так как $f_1^2 = f_1^0$, то алгоритм заканчивает работу и

$$M^* = N_1^1 \cup N_1^2 = \{r_2,\ r_3\}.$$

Таким образом, в нашем примере множество неулучшаемых планов состоит из двух планов:

$$r_2=(3,\ 0,\ 2),\quad r_3=(0,\ 3,\ 2).$$

§ 7. Задача оптимальной коррекции траектории

Актуальной задачей космической навигации является задача минимизации суммарного импульса линейной идеальной коррекции траектории летательных аппаратов.

Пусть совокупность корректируемых параметров траектории задается m-мерным вектором ξ. Известны возможные моменты $t_1,\ t_2,\ \ldots,\ t_n$ коррекции траектории; i-я коррекция определяется: величиной импульса x_i, единичным n_i-мерным (обычно $n_i = 3$) вектором z_i, задающим направление импульса, и матрицей A_i размера $m \times n_i$, характеризующей влияние i-го импульса на вектор корректируемых параметров. В результате коррекций необходимо обеспечить переход траектории в состояние, характеризуемое m-мерным вектором b.

Задача оптимальной коррекции состоит в минимизации суммарного импульса и сводится к решению следующей задачи математического программирования:

$$\sum_{i=1}^{n} x_i \to \min,$$

$$\sum_{i=1}^{n} x_i A_i z_i = b,$$

$$x_i \geqslant 0, \quad i = \overline{1, n},$$

$$|z_i| = 1, \quad i = \overline{1, n}.$$

Существует [68] эффективный алгоритм решения этой задачи, являющийся модификацией симплекс-метода, причем оптимальный план задачи содержит не более m отличных от нуля векторов коррекции $v_i = x_i z_i$. Для приложений целесообразно рассмотреть задачу коррекции при условии, что число корректирующих импульсов не превышает заданного числа s $(s < m)$.

Моменты коррекции, как правило, не могут быть произвольно выбраны из заданного дискретного множества $\{t_1, t_2, \ldots, t_n\}$, что обусловлено необходимостью завершения коррекции в определенные сроки, уточнения траектории после коррекции и другими причинами. Поэтому будем считать, что совокупность индексов $y = \{j_1, j_2, \ldots, j_s\}$, определяющих s моментов $(t_{j_1}, t_{j_2}, \ldots, t_{j_s})$ коррекции, принадлежит некоторому множеству Y, содержащему, очевидно, не более чем C_n^s элементов.

Задача оптимальной коррекции траектории при ограничении на число импульсов сводится к определению

$$\min_{y \in Y} \omega(y),$$

где $\omega(y) = \min\limits_{G(y)} \sum\limits_{i \in y} x_i$. Здесь множество $G(y)$ определяется условиями

$$\sum_{i \in y} x_i A_i z_i = b,$$

$$x_i \geqslant 0, \quad i \in y,$$

$$|z_i| = 1, \quad i \in y.$$

Таким образом, получаем дискретную задачу нахождения оптимального набора из s моментов коррекции.

Покажем, что в качестве миноранты функции $\omega(y)$ на множестве Y можно взять функцию

$$Q(y)=\Big(\sum_{i\in y} u_i\Big)^{-1/2},$$

где

$$u_i=\frac{b^T A_i A_i^T b}{(b^T b)^2}.$$

Прежде всего рассмотрим задачу минимизации функции $\sum_{i\in y} x_i^2$ на области $P(y)$, определяемой условиями

$$\sum_{i\in y} x_i a_i = b,$$

$$x_i \geqslant 0, \quad i\in y.$$

Используя множители Лагранжа, можно получить неравенство

$$\min_{P(y)} \sum_{i\in y} x_i^2 \geqslant \frac{(b^T b)^2}{\sum_{i\in y} (b^T a_i)^2}.$$

Отсюда, принимая во внимание очевидное неравенство

$$(\omega(y))^2 \geqslant \min_{G(y)} \sum_{i\in y} x_i^2,$$

имеем

$$(\omega(y))^2 \geqslant \frac{(b^T b)^2}{\max\Big\{\sum_{i\in y} (b^T A_i z_i)^2 \,\Big|\, z_i^T z_i = 1\Big\}} = \frac{(b^T b)^2}{\sum_{i\in y} b^T A_i A_i^T b},$$

что и требовалось доказать.

Рассмотрим вопрос о построении последовательности наборов $y_1, y_2, \ldots$ в порядке неубывания миноранты $Q(y)$. Поскольку миноранта $Q(y)$ монотонно возрастает с убыванием величины $\sum_{i\in y} u_i$, то для построения последовательности $y_1, y_2, \ldots$ в порядке неубывания $Q(y)$ достаточно строить последовательность векторов $r_1, r_2, \ldots,$ компоненты которых удовлетворяют условиям

$$\sum_{i=1}^{n} x_i = s,$$

$$x_i = 0 \quad \text{или} \quad 1, \quad i = \overline{1, n},$$

в порядке невозрастания функции

$$L(X) = \sum_{i=1}^{n} u_i x_i.$$

Любому плану $(x_1, x_2, \ldots, x_n)$ этой задачи соответствует такой набор индексов $(j_1, j_2, \ldots, j_s)$, что $x_{j_k} = 1 (k = \overline{1, s})$. (Для построения этой последовательности следует воспользоваться алгоритмом, изложенным в § 2 гл. IV.)

ГЛАВА VI

ЗАДАЧИ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ В АСУ

В настоящей главе описывается пакет прикладных программ решения задач целочисленного линейного программирования (ППП ЦЛП), в котором реализованы метод построения последовательности планов, аддитивный алгоритм Балаша, локально-стохастический алгоритм. Приводится общая схема локально-стохастических алгоритмов и ее применение к решению задачи ЦЛП с булевыми переменными и неотрицательными коэффициентами. Описаны некоторые конкретные оптимизационные задачи в АСУ, которые решались с помощью ППП ЦЛП.

Рассмотренная ниже задача оперативного управления качеством добываемой руды входит в состав АСУ «Беларуськалий». Задача составления графика ремонта оборудования в энергосистеме внедрена в Белорусском филиале энергетического института им. Г. М. Кржижановского для решения задач оптимального планирования ремонтных работ на тепловых электростанциях. Задача по расчету вариантов запуска сырья в производство с целью выполнения планируемых показателей предприятия входит в комплекс задач автоматизированных систем управления производством.

ППП ЦЛП использовался также для решения задачи оптимизации потребления материальных ресурсов при выборе портфеля заказов и формировании плана производства (см. § 6 гл. I) на Бердянском заводе «Азовкабель», а задачи оптимального распределения годовой производственной программы предприятия по месяцам — на Климовском заводе текстильного машиностроения, Барановичском заводе автоматических линий и на других предприятиях станкостроительной промышленности.

§ 1. Пакет прикладных программ решения задач целочисленного программирования

Пакет прикладных программ решения задач целочисленного линейного программирования (ППП ЦЛП) предназначен для решения задач целочисленного линейного программирования вида

$$\sum_{j=1}^{n} c_j x_j \to \text{ext}, \tag{1.1}$$

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \quad i = \overline{1, m}, \tag{1.2}$$

$$\sum_{j=1}^{n} d_{kj} x_j = q_k, \quad k = \overline{1, p}, \tag{1.3}$$

$$\sum_{j=1}^{n} l_{sj} x_j \geqslant r_s, \quad s = \overline{1, t}, \tag{1.4}$$

$$x_j = 0 \text{ или } 1, \quad j = \overline{1, n}. \tag{1.5}$$

В пакете используются алгоритмы, основанные на идеях метода построения последовательности планов, а также аддитивный алгоритм Балаша и локально-стохастический алгоритм. Пакет позволяет решать задачи вида $\sum_{j=1}^{n} c_j x_j \to \max$ при ограничениях (1.2) и (1.5) и задачи вида $\sum_{j=1}^{n} c_j x_j \to \min$ при ограничениях (1.4) и (1.5) с числом переменных до 1000 и числом ограничений до 500 (при условии, что $a_{ij}, l_{sj} \geqslant 0$, $b_i, r_s > 0$).

При решении задач вида (1.1)—(1.5) максимальное число переменных — 200, максимальное число ограничений — 200. При решении задач ЦЛП с числом переменных до 500 пакет обеспечивает получение оптимального решения. При решении задач с числом переменных более 500 находится приближенное решение при помощи локально-стохастических алгоритмов.

В ППП ЦЛП имеются программные средства, позволяющие удалять избыточные неравенства из системы ограничений задачи и тем самым уменьшать размерность решаемой задачи. Пакет позволяет также строить тестовые задачи ЦЛП любой размерности с заданным оптимальным планом, что дает возможность экспериментально исследовать эффективность различных методов решения задач ЦЛП. В пакете обеспечивается автоматизация подготовки матрицы задачи непосредственно из информационной базы АСУП, а также выбор алгоритма решения конкретной задачи. Другими словами, пользователю пакета нет необходимости в алгоритмизации вычислительного процесса на этапе формирования матрицы задачи, а также в написании оригинальных программ подготовки исходных данных задачи из данных информационной базы АСУП. Пользователю достаточно описать информационную базу, состав и структуру экономико-математической модели задачи и указать формальные соотношения между данными информационной базы и элементами матрицы задачи.

Эта особенность пакета полезна при использовании его для решения реальных задач оптимизации в АСУП, когда от объекта к объекту меняются состав и структура информационной базы, меняется экономико-математическая модель и, следовательно, появляется необходимость в создании каждый раз новых программ, учитывающих все эти особенности. Программы, реализующие алгоритмы, составлены на языке ассемблер, а сервисные программы — на PL/1.

ППП ЦЛП может эксплуатироваться на любой модели ЕС ЭВМ, имеющей следующий минимальный комплект технических средств:

оперативная память — не менее 256 К байт;
накопитель на магнитных дисках — 3;
устройство ввода с перфокарт — 1;
алфавитно-цифровое печатающее устройство — 1;
пультовая пишущая машинка — 1.

Получение оптимального решения задачи ЦЛП с числом ограничений 90—120 и числом переменных 150—250 на ЭВМ ЕС-1022 требует от 20 до 45 минут, включая ввод с магнитного диска матрицы задачи и печать результатов.

Нахождение приближенного решения задач с числом переменных 1000 и числом ограничений 10 при помощи локально-стохастических алгоритмов требует около часа машинного времени при точности решения 92%.

§ 2. Локально-стохастические алгоритмы

Общая схема. Общую схему локально-стохастических алгоритмов изложим на примере задачи дискретной оптимизации

$$\max\{L(X) \mid X \in G \subset Z_n\}, \tag{2.1}$$

где Z_n — множество всех n-мерных векторов с целочисленными неотрицательными координатами, G — область ограничений задачи (множество планов).

Суть схемы состоит в следующем. С помощью некоторой стохастической процедуры (как правило, с адаптацией к исходным данным задачи) производится выбор исходного плана задачи — вектора X^0 и определяется окрестность этого плана, состоящая из относительно небольшого числа других планов задачи. Среди планов, принадлежащих окрестности, находится оптимальный, который будем называть *локально-оптимальным планом задачи.* Процесс случайного выбора исходного плана и перехода от него к локально-оптимальному осуществляется многократно. Из двух локально-оптимальных планов, полученных на предыдущем и текущем шагах поиска, каждый раз запоминается лучший в смысле значения целевой функции, обеспечивая тем самым получение *субоптимального* плана задачи (2.1) — наилучшего среди множества построенных локально-оптимальных планов.

Основные характеристики локально-стохастических алгоритмов решения задачи (2.1), такие, как скорость сходимости, трудоемкость выполнения одного шага поиска или трудоемкость построения локального оптимума, существенно зависят от правила задания метрики в Z_n и величины радиуса окрестностей. В предлагаемых алгоритмах метрика введена путем определения расстояния $r(X, Y)$ между любыми точками $X, Y \in Z_n$ по формуле

$$r(X, Y) = \sum_{j=1}^{n} |x_j - y_j|.$$

Окрестностью k-го порядка $O_k(X^0)$ точки X^0 назовем множество точек $X \in Z_n$, удовлетворяющих неравенству $r(X^0, X) \leqslant k$. План X^0 будем называть *локально-оптимальным планом k-го порядка,* если

$$L(X^0) \geqslant L(X) \quad \text{для всех} \quad X \in O_k(X^0) \cap G.$$

Локально-стохастический алгоритм будем называть *алгоритмом k-го порядка*, если любому найденному с его помощью плану обеспечивается локальная оптимальность k-го порядка.

Важным элементом рассматриваемых локально-стохастических алгоритмов является процедура G_α построения случайной перестановки $I=(j_1, j_2, \ldots, j_n)$ элементов исходной числовой последовательности $I_0=(j_1^0, j_2^0, \ldots, j_n^0)$.

Процедура G_α, $\alpha \in \{1, 2, \ldots\}$.

1. Имеем исходную последовательность $I_0=(j_1^0, \ldots \ldots, j_n^0)$, полагаем $i:=n$.

2. С помощью датчика случайных чисел, равномерно распределенных на интервале $]0, 1[$, получаем случайное число ξ и вычисляем $\eta:=1-\xi^\alpha$.

3. Полагаем $k:=[\eta\cdot i+1]$, где символ $[a]$ обозначает целую часть числа a.

4. Меняем местами элементы последовательности I_{n-i}, стоящие на i-м и k-м местах, получая в результате перестановку I_{n-i+1}.

5. Полагаем $i:=i-1$. Если $i>1$, то переходим к п. 2, а если $i=1$, то случайная перестановка $I:=I_{n-1}$ получена.

Задача ЦЛП с булевыми переменными и неотрицательными параметрами. Рассмотрим локально-стохастические алгоритмы $A(1, \alpha, \beta)$ и $A(2, \alpha, \beta)$ соответственно первого и второго порядка для решения задачи Z:

$$L(X)=\sum_{j=1}^{n} c_j x_j \to \max,$$

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \quad i=\overline{1, m},$$

$$x_j=0 \text{ или } 1, \quad j=\overline{1, n}.$$

Здесь $a_{ij}\geqslant 0$, $b_i>0$, $c_j\geqslant 0$ $(i=\overline{1,m}, j=\overline{1,n})$, α — значение параметра процедуры G_α, а $\beta\in\{1, 2, 3, 4\}$ — порядковый номер одной из приводимых ниже формул расчета так называемых «весовых коэффициентов» v_j $(j=\overline{1, n})$, определяющих способ адаптации алгоритма к исходным параметрам задачи.

Опишем общую схему алгоритма $A(1, \alpha, \beta)$. На нулевом шаге производится адаптация алгоритма к исходным данным, т. е. в зависимости от значения параметра β рассчитываются «весовые коэффициенты» v_j $(j=\overline{1, n})$:

$$1)\ v_j=c_j, \qquad 3)\ v_j=c_j/(a_j^+-a_j^-),$$
$$2)\ v_j=c_j/a_j^+, \qquad 4)\ v_j=c_j\left(\sum_{i=1}^{m}\frac{a_{ij}}{b_i}\right)^{-1},$$

где $a_j^+=\max_i(a_{ij}/b_i)$, $\quad a_j^-=\min_i(a_{ij}/b_i)$.

Запоминается адаптационная последовательность $I=(j_1^0, j_2^0, \ldots, j_n^0)$, для которой выполняются неравенства $v_{j_1^0}\geqslant v_{j_2^0}\geqslant\ldots\geqslant v_{j_n^0}$.

На s-м $(s\geqslant 1)$ шаге поиска с помощью процедуры G_α генерируется случайная перестановка $I=(j_1, j_2, \ldots, j_n)$ элементов последовательности I_0. Далее, строим план $X^0=(x_1^0, x_2^0, \ldots, x_n^0)$, компоненты которого определяются по формуле

$$x_{j_r}^0=\begin{cases}1, & \text{если } \sum\limits_{l<r} a_{ij_l}\, x_{j_l}^0+a_{ij_r}\leqslant b_i, \quad i=\overline{1, m},\\ 0 & \text{в противном случае,}\end{cases}$$

где $r=1, 2, \ldots, n$.

Нетрудно показать, что план X^0 задачи Z является локально-оптимальным планом 1-го порядка.

Перейдем к рассмотрению алгоритма $A(2, \alpha, \beta)$ 2-го порядка решения задачи. Он состоит из двух процедур — алгоритма 1-го порядка, предназначенного для отыскания локально-оптимальных планов 1-го порядка, и процедуры P_2 перехода от найденного локально-оптимального плана 1-го порядка задачи Z к локально-оптимальному плану 2-го порядка. Отметим, что с помощью процедуры P_2 мы стараемся улучшить полученный локально-оптимальный план 1-го порядка путем одновременной замены значений двух компонент, одна из которых 0, а другая 1.

Рассмотрим алгоритм P_2. Пусть X^0 — найденный локально-оптимальный план 1-го порядка.

Алгоритм P_2. Пусть

$$c_1\geqslant c_2\geqslant\ldots\geqslant c_n, \quad \Delta_i:=b_i-\sum_{j=1}^{n}a_{ij}\,x_j^0, \quad i=\overline{1, m}.$$

1. Полагаем $k := 1$, признак $\gamma := 0$.

2. Если $x_k = 0$, то полагаем $l := n$ и переходим к 3, иначе — к 7.

3. Если $\overset{0}{x_l} = 1$, то переходим к 5, иначе — к 4.

4. Полагаем $l := l - 1$. Если $l > k$, то переходим к 3, иначе — к 7.

5. Вычисляем $\bar{\Delta}_i := \Delta_i - a_{ik} + a_{il}$ $(i = \overline{1, m})$. Если хотя бы для одного i окажется $\bar{\Delta}_i < 0$, то возвращаемся к 4.

6. Реализуем имеющуюся возможность улучшения плана — полагаем $\overset{0}{x_k} := 1, \overset{0}{x_l} := 0$, признак $\gamma := 1$; запоминаем новые невязки $\Delta_i := \bar{\Delta}_i$ $(i = \overline{1, m})$.

7. Полагаем $k := k + 1$. Если $k < n$, то переходим к 2.

8. Если признак $\gamma = 1$, то переходим к 1, иначе — к 9.

9. Полагаем $j := 1$.

10. Если $x_j^0 = 1$, то переходим к 12.

11. Проверяем выполнимость неравенств $\Delta_i - a_{ij} \geqslant \geqslant 0$ $(i = \overline{1, m})$. Если все m неравенств выполняются, то полагаем $x_j^0 := 1$, запоминаем новые невязки $\Delta_i := = \Delta_i - a_{ij}$ $(i = \overline{1, m})$.

12. Полагаем $j := j + 1$. Если $j < n$, то переходим к 10, иначе работу заканчиваем.

Нетрудно доказать, что применение процедуры P_2 к любому из локально-оптимальных планов 1-го порядка задачи Z приводит к локально-оптимальному плану 2-го порядка. Локально-стохастические алгоритмы 1-го и 2-го порядков для общей задачи целочисленного программирования строятся на принципах, аналогичных рассмотренным в алгоритмах 1-го и 2-го порядков для задачи Z.

В результате машинных экспериментов выявлено, что эффективность каждого из алгоритмов $A(1, \alpha, \beta)$ и $A(2, \alpha, \beta)$ зависит от количества единиц k в искомом оптимальном плане. Чем больше величина $|k - n/2|$, тем более эффективно работают алгоритмы. Наиболее эффективными оказались алгоритмы $A(2, 3, 2)$ и $A(2, 3,4)$. При этом алгоритм $A(2, 3, 4)$ по времени нахождения точного решения задач с числом переменных до 70 сравним с методом построения последовательности планов и в большинстве случаев превосходит различные модификации алгоритма Балаша и метод ветвей и границ.

§ 3. Задача оперативного управления качеством добываемой руды

Геологические и горнотехнические условия разработки месторождений калийных солей обусловливают многообразие качественного состава руды по шахтным полям, горизонтам и горным участкам. Следствием этого являются значительные колебания качества руды, поступающей на переработку. Вместе с тем важнейшим требованием обогатительной фабрики к качеству руды является его стабильность. Колебания качества перерабатываемой руды (характеризуемого в первую очередь содержанием в руде хлорида калия) приводит к снижению показателей работы обогатительной фабрики, выражающемуся в увеличении удельного расхода реагентов и менее полном извлечении хлорида калия из руды.

Основная функция системы оперативного управления качеством добываемой руды и заключается в обеспечении стабилизации качества руды, поступающей на обогатительную фабрику, путем оперативного планирования и управления производством.

Задача, которую мы здесь рассматриваем, состоит в составлении такого расписания работы каждого из забоев на очередную смену, которое позволяло бы стабилизировать во времени качество потока руды, поступающей на обогатительную фабрику, и заключается в минимизации максимального отклонения среднего содержания полезной компоненты в руде от плановых показателей с учетом горизонтов, на которых расположены забои. Естественно, при этом должны учитываться ограничения на пропускные способности конвейеров и стволов шахты в каждый момент времени, а также необходимость выполнения плана по добыче руды в каждом из забоев.

Математическая модель. Задача R оптимального управления работой забоев шахты (управления качеством добываемой калийной руды) состоит в минимизации функции

$$L(X) = \max_{j \in M} \max_{u \in U} |\alpha_{ju} - \alpha_u^*| \qquad (3.1)$$

при ограничениях

$$\sum_{i \in N_u} g_{ij} x_{ij} \leqslant P_u^j, \quad \forall u \in U, \quad \forall j \in M = \{\overline{1, m}\}; \qquad (3.2)$$

$$\sum_{i \in J_k} g_{ij} x_{ij} \leqslant R_k^j, \quad \forall j \in M, \quad \forall k \in K; \qquad (3.3)$$

$$\sum_{j=1}^{m} g_{ij} x_{ij} \geqslant H_i, \quad \forall i \in N = \{\overline{1, n}\}; \qquad (3.4)$$

$$x_{ij} = 0 \text{ или } 1, \ \forall i \in N, \quad \forall j \in M. \qquad (3.5)$$

Здесь:

M — множество интервалов планирования;

U — множество горизонтов рудника;

N — множество забоев рудника;

K — множество магистральных конвейеров рудника;

N_u — множество забоев, расположенных на u-м горизонте;

J_k — множество забоев, отгружающих руду на k-й магистральный конвейер;

g_{ij} — производственные возможности i-го забоя по отгрузке руды в j-й интервал времени;

$x_{ij} = 1$, если i-й забой в j-м интервале времени отгружает руду; $x_{ij} = 0$ в противном случае;

$\alpha_{ju} = \left(\sum_{i \in N_u} \alpha_i g_{ij} x_{ij}\right) / \sum_{i \in N_u} g_{ij} x_{ij}$ — средневзвешенная величина качества руды (процентное содержание в руде хлорида калия), поступающей к стволу u-го горизонта;

α_i — процентное содержание хлорида калия в руде i-го забоя;

α_u^* — плановое процентное содержание хлорида калия в руде, поступающей с u-го горизонта;

P_u^j — пропускная способность ствола u-го горизонта в j-й интервал времени;

R_k^j — пропускная способность k-го магистрального конвейера в j-й интервал времени;

H_i — плановое задание на смену по добыче руды для i-го забоя.

Целевая функция (3.1) стабилизирует содержание хлорида калия в добываемой руде. Ограничения (3.2) и (3.3) регламентируют мощность потока руды в соответствии с пропускной способностью стволов и магистральных конвейеров шахты. Ограничения (3.4) контролируют выполнение каждым забоем плановых заданий за смену.

Алгоритм решения задачи. Для решения задачи R разработан специальный вариант локально-стохастического алгоритма, суть которого состоит в следующем. Всякий его шаг состоит из m итераций, на каждой из которых формируется один столбец искомого плана $X^0 = \{x^0_{ij}\}_{n\times m}$, т. е. составляется график включения забоев в работу или прекращения ими работы в каждый j-й интервал времени в зависимости от производственных возможностей забоев в данный интервал времени и степени выполнения сменных заданий за предшествующие интервалы времени.

На каждой j-й итерации $(j = \overline{1, m})$ порядок вычисления элементов j-го столбца определяется случайной перестановкой $I = (i_1, i_2, \ldots, i_n)$ чисел $1, 2, \ldots, n$. На первой итерации в качестве I выбирается случайная равновероятная перестановка чисел $1, 2, \ldots, n$. На каждой последующей итерации в качестве перестановки I выбирается случайная перестановка элементов числовой последовательности $I^0 = (i^0_1, i^0_2, \ldots$ $\ldots, i^0_n)$, получаемая с помощью процедуры G_3. Числовая последовательность I^0, формируемая на предшествующей итерации, есть последовательность номеров забоев, упорядоченных по убыванию невязок:

$$\Delta_i = \left(H_i - \sum_{r=1}^{j-1} g_{ir}\, x^0_{ir}\right) / H_i, \quad i = \overline{1, n}.$$

Невязки Δ_i $(i = \overline{1, n})$ определяют недовыполнения плановых заданий забоями за $j - 1$ интервалов времени, предшествующих рассматриваемому j-му интервалу.

Введенная таким образом адаптация поиска при появлении «узких мест» существенно увеличивает вероятность построения матрицы X^0, которая удовлетворяет ограничениям (3.4).

Более детально алгоритм состоит в следующем.

Алгоритм. На подготовительном шаге полагаем начальное значение целевой функции равным ее заведомо наихудшему значению, например $L^* = \infty$,

k-й шаг ($k \geqslant 1$).

1. Определяем начальные значения так называемых приоритетных коэффициентов $\Delta_i^0 = 0$ $(i = \overline{1, n})$, генерируем случайную равновероятную перестановку $I = (i_1, i_2, \ldots, i_n)$ чисел $1, 2, \ldots, n$.

2. Полагаем номер итерации $j = 1$, начальные значения искомых неизвестных $x_{ij}^0 = 0$ $(i = \overline{1, n}, j = \overline{1, m})$.

3. В порядке $i = i_1, i_2, \ldots, i_n$ заменяем значения x_{ij}^0 на единицы, если такая замена не нарушает ограничений (3.2) и (3.3). Если эти ограничения нарушаются, то значения переменных оставляем без изменения.

4. Вычисляем невязки

$$\Delta_i = \left(H_i - \sum_{s=1}^{j} g_{is} x_{is}^0\right) \Big/ H_i, \quad i = \overline{1, n},$$

и определяем адаптационную последовательность $I^0 = (i_1^0, \ldots, i_n^0)$ такую, что $\Delta_{i_1^0} \geqslant \Delta_{i_2^0} \geqslant \ldots \geqslant \Delta_{i_n^0}$; полагаем $j := j + 1$.

5. Если $j \leqslant m$, то с помощью процедуры G_3 генерируем случайную перестановку $I = (i_1, \ldots, i_n)$ элементов адаптационной последовательности I^0 и переходим к 3; иначе полагаем $k := k + 1$ и переходим к 6.

6. Если $\Delta_{i_1^0} < 0$, то план, удовлетворяющий ограничениям (3.2)—(3.5), получен — переходим к 7; иначе полагаем $\Delta_i^0 := \Delta_i^0 + \Delta_i$ $(i = \overline{1, n})$ и переходим к 9.

7. Если $L(X^0) \leqslant L^*$, то полагаем $L^* = L(X^0)$, запоминаем полученный план $W^* = X^0$.

8. Если запланированное число шагов не выполнено, то переходим к 1, иначе переходим к 10.

9. Если запланированное число шагов не выполнено, то с помощью процедуры G_3 генерируем случайную перестановку $I = (i_1, \ldots, i_n)$ номеров $i_1^0, i_2^0, \ldots, i_n^0$ таких, что $\Delta_{i_1^0}^0 \geqslant \ldots \geqslant \Delta_{i_n^0}^0$, и переходим к 2; иначе переходим к 10.

10. Алгоритм работу окончил, W^* — субоптимальный план со значением целевой функции L^*.

Из способа построения планов задачи R легко следует, что каждый план, получаемый с помощью описанного алгоритма, является локально-оптимальным планом 1-го порядка.

§ 4. Задача составления графика ремонта оборудования в энергосистеме

Частью общей проблемы управления энергосистемой, проблемы оптимизации режимов работы основного оборудования, является планирование его ремонта. Сложность данной проблемы предопределила целесообразность ее разделения на ряд взаимосвязанных, последовательно решаемых задач составления годового графика ремонтов, выбора состава включенного оборудования, экономичного распределения нагрузки. Среди перечисленных задач задача планирования ремонтов решается первой, и результаты ее решения существенно влияют на эффективность функционирования энергосистемы в целом. В эксплуатационных условиях возможны ситуации, когда требуется оперативно изменять рассчитанный предварительно график ремонтов (аварии, стихийные бедствия и др.). Большое число агрегатов, необходимость учета разных, в том числе противоречивых, ограничений диктуют необходимость использования современных методов решения оптимизационных задач и применения ЭВМ.

Задача заключается в составлении графика ремонта оборудования, который определяет целесообразные (по критерию минимальных затрат) сроки вывода в ремонт того или иного агрегата. Математически задачу можно сформулировать как задачу дискретной оптимизации. С этой целью разделим планируемый период (год) на равные интервалы (месяцы или декады). Выбор масштаба измерения времени, естественно, зависит от продолжительности ремонтов. Необходимо, чтобы продолжительность ремонтов каждого вида оборудования измерялась целым числом временных интервалов.

Пусть параметр τ_i — искомый номер интервала, в котором i-й агрегат подлежит выводу в ремонт. Введем вспомогательный параметр x_{ij}, полагая его равным единице, если i-й агрегат в j-й период времени находится в ремонте, и равным нулю в противном случае. Тогда математическую модель можно записать следующим образом:

$$L(X)=\sum_{i=1}^{n}\sum_{j=1}^{m}c_{ij}x_{ij}\to\min \tag{4.1}$$

при ограничениях

$$\sum_{j=\tau_i}^{\tau_i+t_i-1}x_{ij}=t_i,\quad i=\overline{1,n}, \tag{4.2}$$

$$a_i\leqslant\tau_i\leqslant b_i,\quad i=\overline{1,n}, \tag{4.3}$$

$$\sum_{i\in N^k}x_{ij}\leqslant n_j^k,\quad j=\overline{1,m},\quad k=\overline{1,s}, \tag{4.4}$$

$$\sum_{i\in N_\nu^k}x_{ij}\leqslant 1,\quad j=\overline{1,m},\quad k=\overline{1,s},\quad \nu=\overline{1,p}, \tag{4.5}$$

$$\sum_{i=1}^{n}P_ix_{ij}\leqslant P^j,\qquad j=\overline{1,m}, \tag{4.6}$$

$$\sum_{i=1}^{n}W_ix_{ij}\leqslant W^j,\qquad j=\overline{1,m}, \tag{4.7}$$

$$x_{ij}=0 \text{ или } 1,\quad i=\overline{1,m},\quad j=\overline{1,n}, \tag{4.8}$$

$$\tau_i \text{ — целое число},\quad i=\overline{1,m}. \tag{4.9}$$

Исходные данные описанной модели имеют следующий смысл:

c_{ij} — снижение уровня эффективности работы энергосистемы при ремонте i-го агрегата в j-й период времени;

t_i — длительность ремонта i-го агрегата (в интервалах);

a_i, b_i — допустимые временные границы начала ремонта i-го агрегата;

N^k — множество агрегатов k-го типа (котлы, турбины и т. д.);

n_j^k — допустимое количество одновременного ремонта агрегатов k-го типа в j-й период;

N_ν^k — множество агрегатов k-го типа на ν-й станции;

P_i — численность персонала, требуемая для ремонта i-го агрегата;

P^j — общая численность ремонтного персонала в j-й период времени, которым располагает энергосистема;

W_i — снижение мощности энергосистемы при выводе в ремонт i-го агрегата;

W^j — допустимое снижение мощности энергосистемы в j-й период времени.

В этих обозначениях целевая функция (4.1) определяет затраты, связанные с ремонтом всех агрегатов системы. Ограничения (4.2), (4.3) обеспечивают непрерывность ремонта каждого агрегата в заданных временных границах. С помощью технологических ограничений (4.4) контролируется общее количество одновременно ремонтируемых в энергосистеме агрегатов каждого типа. Условия (4.5) не допускают одновременного ремонта двух и более однотипных агрегатов на одной станции. Ресурсные ограничения (4.6) обеспечивают запланированные ремонты обслуживающим персоналом, а ограничения (4.7) не позволяют превышать в каждый период времени допустимое снижение мощности энергосистемы.

В основе локально-стохастического алгоритма решения задачи (4.1)—(4.9) лежат те же идеи, что и в вышеописанном алгоритме решения задачи. Схема работы этого алгоритма состоит в том, что на каждом шаге выполняем следующее.

1. Случайным образом генерируем начальный план, т. е. любой график ремонта, удовлетворяющий ограничениям (4.2), (4.3), (4.8), (4.9).

2. Исходя из начального плана, пытаемся построить допустимый план, т. е. график ремонта, удовлетворяющий всем ограничениям задачи. Если такой план построен, то переходим к п. 3. Если нет, то в конце каждого шага запоминаем лучший из «плохих» планов (одно или несколько ограничений не выполняется). Критерием выбора такого плана является минимизация наибольшей из относительных невязок ограничений (4.4)—(4.7). По истечении запланированного числа шагов поиска в случае, когда ни одного допустимого плана не найдено, печатается лучший из «плохих» планов. Если он не удовлетворяет заказчика, то счет продолжается (переход к п. 1).

3. Фиксируем радиус $k = 1$ и строим локально-оптимальный план k-го порядка. После этого либо алгоритм заканчивает свою работу и лучший из полученных локально-оптимальных планов берем в качестве исходного графика ремонта, либо продолжаем работу, переходя к п. 4.

4. Увеличиваем радиус k и переходим к следующему шагу.

Вычислительная реализация данного алгоритма на ЭВМ ЕС-1022 показала его высокую эффективность.

Алгоритм апробирован на реальной задаче расчета графика капитального ремонта оборудования на год (с разбивкой по месяцам) для энергосистемы с установленной мощностью порядка 5 тыс. Мвт. Число ремонтируемых агрегатов — 67 (турбины, парогенераторы, водогрейные котлы). Среднее время решения задачи равно 40 мин.

С небольшими модификациями описанные выше модель и алгоритм были использованы на производственном объединении «Беларуськалий» для оптимизации календарных планов ремонта горных машин (добычных комбайнов). Решена серия задач составления графиков планово-предупредительных ремонтов ($m = 12$, $n = 40$). Среднее время решения равно 10 мин (ЕС-1020).

§ 5. Расчет вариантов запуска сырья в производство

Ввиду сложности и специфичности технологического процесса изготовления бриллиантов чрезвычайно трудно определить, какие из имеющихся в наличии партий исходного сырья (алмазов) необходимо запустить в производство для переработки, чтобы обеспечивался нормальный режим работы производственных участков и гарантировалось выполнение планируемых показателей предприятия. Необоснованно завышенный объем запускаемого в производство сырья приводит к использованию участками более дорогостоящего и менее трудоемкого при обработке крупного сырья и к накоплению в незавершенном производстве полуфабрикатов, менее выгодных с точки зрения их переработки (наиболее трудоемких и менее ценных).

Предлагаемая здесь модель разработана и внедрена для месячного планирования запуска сырья в производство на ряде ювелирных фабрик страны. Для того чтобы сформулировать задачу математически, введем необходимые обозначения. Пусть n — число партий сырья, которые могут обрабатываться в плановом периоде. Исходными данными являются:

v_j — общий вес j-й партии сырья в каратах;

c_j — стоимость одного карата сырья j-й партии;

s_j — стоимость одного карата готовой продукции j-й партии;

α_j — процент выхода готовой продукции из j-й партии сырья;

k_j — количество алмазов в j-й партии.

Для предприятия планируются следующие показатели:

V — общий вес в каратах готовой продукции, которую необходимо выпустить в плановом периоде;

v — средний вес одного кристалла сырья;

c — средняя стоимость одного карата сырья;

α — средний процент выхода готовой продукции;

s — средняя стоимость одного карата готовой продукции.

Необходимо выбрать к запуску такие партии сырья, чтобы добиться максимума стоимости готовой продукции при достижении планируемых предприятию показателей. Если ввести переменную x_j, равную единице в случае, когда j-я партия запускается в производство, и равную нулю в противном случае, то задача сводится к максимизации функции

$$\sum_{j=1}^{n} v_j \alpha_j s_j x_j$$

при соблюдении ограничений

1) по общему весу готовой продукции:

$$\sum_{j=1}^{n} v_j \alpha_j x_j \geqslant 100\ V;$$

2) по среднему весу одного кристалла сырья:

$$\sum_{j=1}^{n} v_j x_j \leqslant v \sum_{j=1}^{n} k_j x_j;$$

3) по средней стоимости одного карата сырья:

$$\sum_{j=1}^{n} c_j v_j x_j \leqslant c \sum_{j=1}^{n} v_j x_j;$$

4) по среднему проценту выхода готовой продукции:

$$\sum_{j=1}^{n} v_j \alpha_j x_j \geqslant \alpha \sum_{j=1}^{n} v_j x_j;$$

5) по средней стоимости одного карата готовой продукции:

$$\sum_{j=1}^{n} v_j \alpha_j s_j x_j \geqslant s \sum_{j=1}^{n} v_j \alpha_j x_j;$$

6) $$x_j = 0 \text{ или } 1, \quad j = \overline{1, n}.$$

Промышленная реализация этой модели позволяет строить такие варианты графиков запуска сырья в обработку (производство), при которых значительно сокращаются объемы незавершенного производства и обеспечивается выполнение установленных предприятию плановых заданий на основные технико-экономические показатели.

§ 6. Распределение производственной программы предприятия по плановым периодам

При заданной годовой или квартальной производственной программе, т. е. при заданном количестве выпускаемых изделий каждого типа, предприятиям часто предоставляется право свободного распределения ее по месяцам при условии равенства суммы месячных программ годовой или квартальной программе. Тогда и возникает задача распределения производственной программы предприятия по месяцам, или, другими словами, задача формирования единого плана-графика выпуска изделий всеми подразделениями предприятия по месяцам.

Указанная задача особенно актуальна для предприятий с единичным и мелкосерийным характером производства, где большая номенклатура и малая серийность выпускаемых изделий вызывают частые изменения состава и количественного соотношения различных изделий в годовом или квартальном плане выпуска, что, в свою очередь, приводит к необходимости оперативного пересмотра плана-графика. Обеспечение планомерного выполнения всех заказов при ритмичной работе подразделений предприятия, равномерном использовании всех ресурсов, минимальном количестве переналадок оборудования и их равномерном распределении в течение планируемого периода, концентрации выпуска изделий одного наименования в минимально возможном отрезке времени — все это должно быть учтено в едином плане-графике выпуска изделий.

Перед плановыми службами предприятия ежемесячно возникает проблема выбора из общего номенклатурного перечня такого набора изделий, который бы обеспечивал рациональное использование ресурсов и равномерное распределение выпуска изделий в течение года или квартала.

Изделия в программе выпуска разнятся друг от друга по структуре трудовых затрат, а также по времени, необходимому для их изготовления. Одни, например, требуют больших объемов токарных и сверлильных работ, другие — слесарных, третьи — фрезерных и т. д. Цикл изготовления одного изделия, как правило, превышает один месяц, и поэтому необходимо учитывать наличие свободных ресурсов на весь период изготовления изделий. Таким образом, если не обеспечить правильного набора месячных программ выпуска изделий, то неизбежно возникнут резкие скачки в загрузке различных подразделений, различных групп оборудования, что приведет к перегрузке отдельных видов станков и к простою других.

Исходными данными для решения этой задачи являются:

— ведомость характеристик изделий, которая содержит данные о плане выпуска предприятием каждого изделия, оптовую цену каждого изделия, данные о длительности изготовления изделия, о трудоемкости изготовления изделия на различных группах оборудования;

— ведомость характеристик лимитирующих групп оборудования, которая содержит данные о действительных фондах времени групп оборудования на весь период планирования с указанием допустимых отклонений фактического фонда времени от действительного;

— план выпуска продукции предприятием в стоимостном выражении на весь период планирования с разбивкой по месяцам.

В результате решения задачи необходимо получить план-график выпуска изделий с указанием сроков начала и окончания изготовления каждого изделия, с указанием стоимости выпускаемой в каждом месяце продукции, а также с указанием фактической загрузки подразделений в течение планируемого периода.

Чтобы сформулировать экономико-математическую модель, введем обозначения:

$I = \{1, 2, \ldots, n\}$ — множество индексов изделий, выпускаемых предприятием;

$J = \{1, 2, \ldots, m\}$ — множество индексов видов оборудования, используемых при выпуске изделий;

T — период планирования (в месяцах);

τ_i — цикл (продолжительность) изготовления i-го изделия ($\tau_i \leqslant T$);

f_{ij}^t — затраты времени j-го вида оборудования в t-й месяц ($t = \overline{1, \tau_i}$) обработки i-го изделия; $f_{ij}^t = 0$ при $t > \tau_i$;

F_{jt} — фонд времени работы j-го оборудования в t-й месяц ($t = \overline{1, T}$);

c_i — оптовая цена i-го изделия;

P_t — план выпуска продукции предприятием на t-й месяц в стоимостном выражении ($t = \overline{1, T}$).

Введем переменные

$$x_{it} = \begin{cases} 1, & \text{если обработка } i\text{-го изделия} \\ & \text{начинается в } t\text{-й месяц;} \\ 0 & \text{в противном случае.} \end{cases}$$

В принятых обозначениях загрузка j-го вида оборудования в t-й месяц равна величине

$$k_{jt} = \sum_{q=1}^{t} \sum_{i=1}^{n} f_{ij}^{t-q+1} x_{iq}.$$

Тогда разность между максимальной и минимальной загрузками оборудования в t-м месяце составит

$$k_t = \max_{1 \leqslant j \leqslant m} k_{jt} - \min_{1 \leqslant j \leqslant m} k_{jt}.$$

Если в качестве критерия оптимальности принять равномерность загрузки по месяцам, то задача распределения производственной программы предприятия заключается в минимизации функции

$$\max_{1 \leqslant t \leqslant T} k_t$$

при следующих ограничениях.

1. Загрузка оборудования каждого вида в каждом месяце не должна превышать фонда времени работы:

$$\sum_{q=1}^{t} \sum_{i=1}^{n} f_{ij}^{t-q+1} x_{iq} \leqslant F_{jt}, \quad j=\overline{1, m}, \quad t=\overline{1, T}.$$

2. План выпуска продукции по месяцам предприятием выполняется:

$$\sum_{i=1}^{n} c_i \, x_{i,\, t-\tau_i+1} \geqslant P_t, \quad t=\overline{1, T}$$

(по определению считаем, что $x_{iq}=0$, если $q \leqslant 0$).

3. Каждое изделие должно быть изготовлено в плановом периоде:

$$\sum_{t=1}^{T-\tau_i+1} x_{it}=1, \quad i=\overline{1, n},$$

$$\sum_{t=1}^{T} x_{it}=1, \quad i=\overline{1, n};$$

4. $x_{it}=0$ или 1, $i=\overline{1, n}$, $t=\overline{1, T}$.

БИБЛИОГРАФИЧЕСКИЙ КОММЕНТАРИЙ

Список литературы, приведенный в этой книге, не претендует на полноту, поскольку авторы не ставили перед собой цели дать обзор всех работ по дискретной оптимизации.

Сначала сделаем замечания общего характера по темам, имеющим непосредственное отношение к рассматриваемым здесь вопросам.

В настоящее время имеется несколько специальных монографий, посвященных дискретной оптимизации, из которых упомянем книги А. А. Корбута и Ю. Ю. Финкельштейна [62], Т. Саати [88], Т. Ху [100], М. М. Ковалева [52], М. Шоха [118], Р. Гарфинкеля и Г. Немхаузера [107] и И. Пилера [117].

Хороший обзор литературы по дискретным экстремальным задачам, опубликованной за последние 10 лет, дал В. К. Леонтьев в работе [66]. Достаточно подробная библиография по этой тематике содержится в [111, 112]. Некоторые сведения об эффективности комбинаторных методов в дискретной оптимизации можно найти в книге [92].

В практике перспективного отраслевого и регионального планирования большое распространение получили задачи размещения производства как в дискретной, так и в непрерывной постановке. В последнее время методы решения таких задач успешно разрабатывают А. Е. Бахтин [7], В. Р. Хачатуров [98, 99], А. П. Уздемир [94, 95], и др. (см., например, [74, 91]).

В связи с задачами стандартизации укажем работы В. Т. Дементьева, Э. Х. Гимади, В. Л. Береснева [10, 17], в которых изучены экстремальные задачи выбора непрерывных параметров изделий, подлежащих выпуску.

Широкому комплексу вопросов, которые необходимо решать при разработке и внедрении автоматизированных систем управления (АСУ), посвящены книги В. М. Глушкова [18, 19], Н. П. Федоренко [96], А. Г. Аганбегяна и др. [1].

Среди работ, посвященных вопросам технологии разработки пакетов прикладных программ, укажем статьи В. М. Глушкова и др. [20, 21], А. А. Самарского и др. [51].

Зачастую для математических моделей реальных процессов, отличающихся большой размерностью, не удается создать эффективный алгоритм нахождения точного решения. Более того, как показано С. В. Яблонским [104] и Ю. И. Журавлевым [45], а позднее С. А. Куком [63] и Р. М. Карпом [50], для некоторых комбинаторных задач вообще не существует эффективных точных алгоритмов. Поэтому для решения таких задач применяются приближенные алгоритмы. Некоторые сьедения о приближенных методах решения задач дискретного программирования можно найти в [97].

В последнее время для решения оптимизационных задач все шире стали применяться методы, основанные на стохастических процедурах поиска решений. Среди них отметим методы стохастической аппроксимации Ю. М. Ермольева [43] и методы случайного поиска Л. А. Растригина [85, 86] и И. Б. Моцкуса [80], позволившие построить эффективные алгоритмы решения разнообразных прикладных задач.

Теперь перейдем к замечаниям по главам.

Глава I

§ 4 основан на работе Я. М. Александровича, В. И. Комлика, Г. К. Ладышева и Г. В. Чечетта [2], а § 5 — на работе В. А. Емеличева и В. И. Шлефрина [26].

В постановке и решении задач распределения капитальных вложений по стройкам и годам планового периода (§ 9) принимали участие В. А. Буклей, В. А. Павликов и Н. М. Пастухов (см. [40]).

Глава II

Метод построения последовательности планов был разработан авторами в 1964—1965 гг. для решения задач отраслевого планирования и результаты были доложены на Всесоюзной конференции по математическому оптимальному программированию в 1965 г. [87]. При изложении общей схемы метода авторы следуют работе [34]. Приведенная здесь схема упорядочения планов, основанная на нахождении и исключении элементов множества R, изложена в работе [36]. Описание алгоритма упорядочения планов (§ 3) опирается на работы [23, 28, 53, 55]. Некоторые варианты общей схемы изложены в книге М. М. Ковалева [52]. Один из этих вариантов есть не что иное, как аппроксимационно-комбинаторный метод В. Р Хачатурова [99], а другой близок к принципу расширения М. Шоха [118] и методу погружения В. П. Булатова [12].

Следует отметить, что в нашу схему ψ_0 укладываются (см. [36, 37]) такие известные методы, как методы отсечения Р. Гомори [109] в целочисленном программировании и Дж. Келли [113] в выпуклом программировании, метод фильтра Э. Балаша [105], метод ветвей и границ Лэнд и Дойг [116] для решения задач целочисленного линейного программирования и его общая схема, предложенная Балашем [106].

Глава III

Первые результаты по применению метода построения последовательности планов к решению однопродуктовых задач размещения изложены в работе авторов [23], дальнейшее их развитие содержится в [25]. Вопросы решения задачи размещения с двойным транспортом рассматриваются в работе В. И. Комлика и Г. В. Чечетта [56]. Подробное изложение алгоритмов решения многопродуктовых задач размещения можно найти в [29, 53], а межотраслевой задачи размещения — в [58]. Вычислительные аспекты и опыт решения практических задач размещения содержатся в работах [25, 40, 53].

Глава IV

Вопросы применения метода построения последовательности планов к решению общей задачи ЦЛП рассматривались авторами в работе [55]. Изложению алгоритма решения задачи ЦЛП с булевыми переменными посвящены работы [30, 31, 54]. Результаты машинных экспериментов по решению задач ЦЛП приведены в [38, 40, 54, 76, 83]. В работах [38, 83] дано сравнение результатов с результатами других авторов.

Глава V

Методы решения задач стандартизации в дискретной постановке изучались В. А. Емеличевым, М. М. Ковалевым и Э. Гирлихом (см [39, 114, 115]). Изложение § 1 следует работам В. А. Емеличева и М. М. Ковалева [32, 35]. Изложенный в § 2 локальный подход к построению последовательности планов предложен В. А. Емеличевым в [27]. § 3 основан на работе авторов [24], а § 4— на работе Буй Кат Тыонга [11].

Систематическое исследование экстремальных задач на подстановках было начато Д. А. Супруненко и его учениками в 60-х годах. Как показал Д. А. Супруненко [93], многие экстремальные задачи комбинаторного типа могут быть сведены к проблеме нахождения экстремальных значений линейной формы на некоторых множествах подстановок. Применению метода построения последовательности планов к решению экстремальных задач на подстановках посвящены статьи В. И. Комлика и В. Е. Микульского [57, 70, 71], Н. Н. Метельского [69], Н. Е. Гайкова [15, 16]. При изложении задачи минимизации линейной формы авторы следовали работе [57]. Алгоритм решения задачи минимизации скалярного произведения двух матриц предложен в [71], задачи минимизации суммы линейных форм — в [70], а задачи разрезания графа на подграфы с фиксированным числом вершин — в [15].

Качественному изучению структуры множества неулучшаемых планов посвящена работа В. И. Комлика и В. И. Куриленкова [59]. При изложении алгоритмов построения множества неулучшаемых планов (§ 6) авторы следовали в основном статье [60]. § 7 основан на статье В. Ц. Бахшияна [6].

Глава VI

§ 1. Пакет прикладных программ ЦЛП разработан В. А. Павлечко. Работа ППП ЦЛП обеспечивается как под управлением СУПЕРВИЗОРА ОС/ЕС и ДОС/ЕС, так и под управлением пакета прикладных программ для технико-экономического планирования [81].

§ 2. Локально-стохастические алгоритмы решения задач ЦЛП предложены в работе В. А. Емеличева, М. М. Ковалева, А. М. Кононенко [33] и развивались в работах В. А. Пирьяновича [42, 83], на основе которых и изложен этот параграф.

§ 3. В постановке и решении задачи оперативного управления качеством добываемой руды участвовали Л. В. Федулов, Н. А. Лепешинский, В. А. Пирьянович, М. М. Ковалев, П. Д. Кудрявцев, В. В. Кацубо, А. К. Гец [67, 84].

§ 4. Изложение данного параграфа следует работе В. А. Пирьяновича и Д. А. Гольбина [82]. Основными исполнителями работы по созданию системы календарного планирования предупредительных ремонтов на производственном объединении «Беларуськалий» являлись Л. В. Федулов, М. М. Ковалев, Ф. С. Бабицкая [5].

§ 5. В решении задач расчета вариантов запуска сырья в производство участвовали В. В. Михолан, В. А. Павлечко, В. В. Полонский, В. М. Громыко [76].

§ 6. В разработке экономико-математической модели задачи оптимального распределения годовой производственной программы по плановым периодам и метода ее реализации принимали участие В. А. Павлечко, В. А Емеличев, Е. И. Велесько, В. И. Комлик, Н. И. Матафонова, В. А. Герман, А. А. Илюкович, Н. Н. Кандауров [13, 41].

ЛИТЕРАТУРА

1. Аганбегян А. Г., Багриновский К. А., Гранберг А. Г. Система моделей народнохозяйственного планирования. — М.: Мысль, 1972.
2. Александрович Я. М., Комлик В. И., Ладышев Г. К., Чечетт Г. В. К вопросу внутрирайонного размещения промышленного производства. — В кн.: Математические модели и методы в автоматизированных системах, Минск, НИИЭМП при Госплане БССР, 1975, вып. 1.
3. Архипова Т. Т., Сергиенко И. В. О формализации и решении некоторых задач организации вычислительного процесса в системах обработки данных. — Кибернетика, 1973, № 5.
4. Архипова Т. Т., Сергиенко И. В. Метод возможных направлений и метод вектора спада в ЦЛП. — Кибернетика, 1975, № 5.
5. Бабицкая Ф. С., Ковалев М. М., Федулов Л. В. Оптимизация календарных планов ремонта горных машин. — В кн.: Вопросы математического обеспечения в автоматизированных системах, Минск, НИИЭМП при Госплане БССР, 1979.
6. Бахшиян Б. Ц. Комбинаторный метод решения задач оптимальной коррекции траектории при ограничении на число импульсов. — Космич. исслед., 1976, 14, № 4.
7. Бахтин А. Е. Методы решения задач территориально-производственного планирования. — Новосибирск: Наука, 1976.
8. Беллман Р. Динамическое программирование. — М.: ИЛ, 1960.
9. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. — М.: Наука, 1965.
10. Береснев В. Л., Гимади Э. Х., Дементьев В. Т. Экстремальные задачи стандартизации. — Новосибирск: Наука, 1978.
11. Буй Кат Тыонг. К задаче комплектации.— Вестник БГУ. Серия 1, 1976, № 1.
12. Булатов В. П. Методы погружения в задачах оптимизации. — Новосибирск: Наука, 1977.
13. Велесько Е. И., Герман В. А., Илюкович А. А., Кандауров Н. Н., Матафонова Н. И., Павлечко В. А. Календарное планирование производственной программы предприятия с мелкосерийным производством. — Приборы и системы управления, 1976, № 3.
14. Волкович В. Л., Волошин А. Ф. Об одной схеме метода последовательного анализа и отсеивания вариантов. — Кибернетика, 1978, № 4.

15. Гайков Н. Е. Разрезание графа на подграфы с фиксированным числом вершин. — Изв. АН БССР. Серия физ.-мат. наук, 1977, № 2.
16. Гайков Н. Е. Приближенное решение общей задачи о назначениях и задачи минимизации скалярного произведения матриц. — Изв. АН БССР. Серия физ.-мат. наук, 1976, № 2.
17. Гимади Э. Х., Дементьев В. Т. Некоторые задачи выбора оптимальных параметров ряда и методы их решения (Задачи стандартизации). — В кн.: Проблемы кибернетики, М., 1973, вып. 27.
18. Глушков В. М. Введение в АСУ. — М.: Техника, 1974.
19. Глушков В. М. Макроэкономические модели и принципы построения ОГАС. — М.: Статистика, 1975.
20. Глушков В. М., Вельбицкий И. В., Стогний А. А. Единый автоматно-лингвистический подход к технологии разработки пакетов прикладных программ. — В кн.: Тезисы докладов международной конференции «Структура и организация пакетов программ». Тбилиси, 1976.
21. Глушков В. М., Капитонова Ю. В., Каспшицкая М. Ф., Сергиенко И. В. Алгоритмическое обеспечение пакета программ Вектор-1, предназначенное для решения одного класса задач проектирования ЭВМ. — Кибернетика, 1978, № 4.
22. Емеличев В. А. Об одной задаче вогнутого программирования. — Изв. АН БССР. Серия физ.-мат. наук, 1965, № 3.
23. Емеличев В. А., Комлик В. И. Применение динамического программирования к решению задачи размещения. — ДАН БССР, 1966, **10**, № 10.
24. Емеличев В. А., Комлик В. И. О проблеме надежности. — Изв. АН БССР. Серия физ.-мат. наук, 1968, № 5.
25. Емеличев В. А., Комлик В. И. О некоторых задачах размещения. — ДАН БССР, 1968, **12**, № 10.
26. Емеличев В. А., Шлефрин В. И. Некоторые экстремальные задачи специализации литейного производства. — Экономика и математические методы, 1968, **4**, вып. 1.
27. Емеличев В А. Вогнутое программирование с сепарабельной функцией цели при линейных ограничениях. — Изв. АН БССР. Серия физ.-мат. наук, 1969, № 6.
28. Емеличев В. А., Комлик В. И. Решение задач дискретного программирования методом построения последовательности планов. — ДАН СССР, 1969, **188**, № 2.
29. Емеличев В. А., Комлик В. И. К многопродуктовой задаче размещения. — Вестник БГУ. Серия 1, 1970, № 1.
30. Емеличев В. А., Комлик В. И., Краверсий И. М. Алгоритм решения задач целочисленного линейного программирования методом построения последовательности планов. — ДАН БССР, 1970, **14**, № 11.
31. Емеличев В. А. К задачам дискретной оптимизации. — ДАН СССР, 1970, **192**, № 5.
32. Емеличев В. А., Ковалев М. М. Решение некоторых задач вогнутого программирования методом построения последовательности планов. I. — Изв. АН БССР. Серия физ.-мат. наук, 1970, № 6.
33. Емеличев В. А., Ковалев М. М., Кононенко А. М. Два приближенных метода решения задач целочисленного

линейного программирования с булевыми переменными. — В кн.: Автоматизированные системы управления, Минск, 1971, вып. 5.
34. Емеличев В. А. К теории дискретной оптимизации. — ДАН СССР, 1971, 198, № 2.
35. Емеличев В. А., Ковалев М. М. Решение некоторых задач вогнутого программирования методом построения последовательности планов. II. — Изв. АН БССР. Серия физ.-мат. наук, 1971, № 2.
36. Емеличев В. А. Дискретная оптимизация: Последовательностные схемы решения. I. — Кибернетика, 1971, № 6.
37. Емеличев В. А. Дискретная оптимизация. Последовательностные схемы решения. II. — Кибернетика, 1972, № 2.
38. Емеличев В. А., Краверский И. М. Машинный эксперимент по решению задач целочисленного линейного программирования методом построения последовательности планов. — ЖВМ и МФ, 1973, 13, № 2.
39. Емеличев В. А., Ковалев М. М. Гирлих Э. Некоторые математические вопросы оптимальной стандартизации. — 18. Intern. Wiss. Koll. TH Ilmenau, 1973.
40. Емеличев В. А., Комлик В. И. Математическое обеспечение класса оптимизационных задач в АСПР. — В кн.: Тезисы докладов Всесоюзного семинара-совещания по проблеме «Вопросы разработки математического обеспечения АСПР». Минск, 1973.
41. Емеличев В. А., Велесько Е. И., Комлик В. И., Илюкович А. А., Павлечко В. А. Распределение производственной программы предприятия по плановым периодам. — В кн.: Материалы Всесоюзной научной конференции «Математическое обеспечение автоматизированных систем управления». М., 1975.
42. Емеличев В. А., Пирьянович В. А. Об одном методе построения субоптимального плана задачи целочисленного линейного программирования. — В кн.: Вопросы математического обеспечения в автоматизированных системах. Минск, 1979.
43. Ермольев Ю. М. Методы стохастического программирования. — М.: Наука, 1976.
44. Журавлев Ю. И. Об одном классе алгоритмов над конечными множествами. — ДАН СССР, 1963, 151, № 5.
45. Журавлев Ю. И. Оценки сложности алгоритмов построения минимальных дизъюнктивных нормальных форм для функций алгебры логики. — В кн.: Дискретный анализ, Новосибирск, 1964, вып. 3.
46. Журавлев Ю. И. Оценка сложности локальных алгоритмов для некоторых экстремальных задач на конечных множествах. — ДАН СССР, 1964, 158, № 5.
47. Журавлев Ю. И. Локальные алгоритмы вычисления информации. I. — Кибернетика, 1965, № 1.
48. Журавлев Ю. И., Финкельштейн Ю. Ю. Локальный алгоритм для задач линейного целочисленного программирования. — В кн.: Проблемы кибернетики, 1965, вып. 14.
49. Журавлев Ю. И. Локальные алгоритмы вычисления информации. II. — Кибернетика, 1966, № 2.

50. Карп Р. М. Сводимость комбинаторных проблем. — В кн.: Кибернетический сборник. Новая серия, 1975, вып. 12.
51. Карпов В. Я., Корягин Д. А., Самарский А. А. Принципы разработки пакетов прикладных программ для задач математической физики. — ЖВМ и МФ, 1978, 18, № 2.
52. Ковалев М. М. Дискретная оптимизация (целочисленное программирование). — Минск.: БГУ, 1977.
53. Комлик В. И. Об одной многопродуктовой задаче размещения. — Изв. АН БССР. Серия физ.-мат. наук, 1967, № 4.
54. Комлик В. И., Емеличев В. А. Решение задачи целочисленного программирования с переменными 0—1. — ДАН БССР, 1967, 11, № 12.
55. Комлик В. И., Емеличев В. А. Об одной задаче целочисленного программирования. — Изв. АН БССР. Серия физ.-мат. наук, 1968, № 4.
56. Комлик В. И., Чечетт Г. В. Решение задачи размещения с учетом затрат на доставку сырья. — В кн.: Математические модели, алгоритмы и программы задач размещения производства для ЭВМ Минск-22, Минск, НИИЭМП при Госплане БССР, 1971.
57. Комлик В. И., Микульский В. Е. Об одном подходе к минимизации линейной формы на множестве подстановок. — В кн.: Экономико-математические модели и численные методы оптимизации, Минск, НИИЭМП при Госплане БССР, 1972.
58. Комлик В. И., Чечетт Г. В. Об одной межотраслевой задаче размещения предприятий. — В кн.: Экономико-математические модели и численные методы оптимизации, Минск, НИИЭМП при Госплане БССР, 1972.
59. Комлик В. И., Куриленков В. И. О структуре множества неулучшаемых решений. — Изв. АН БССР. Серия физ.-мат. наук, 1976, № 4.
60. Комлик В. И., Куриленков В. И., Ромашкина Н. В. Об одном алгоритме построения множества неулучшаемых планов. — В кн.: Автоматизированные системы плановых расчетов в республиканских плановых органах. Минск, НИИЭМП при Госплане БССР, 1977, вып. 9.
61. Корбут А. А., Сигал И. Х., Финкельштейн Ю. Ю. Метод ветвей и границ. — Math. Operationsforsch, Statist. Ser. «Optimization», 1977, 8, № 2.
62. Корбут А. А., Финкельштейн Ю. Ю. Дискретное программирование. — М.: Наука, 1969.
63. Кук С. А. Сложность процедур вывода теорем. — В кн.: Кибернетический сборник. Новая серия, 1975, вып. 12.
64. Лебедева Т. Т., Рощин В. А., Сергиенко И. В. Один подход к решению задач математического программирования со смешанными переменными. — Кибернетика, 1977, № 2.
65. Лебедева Т. Т., Сергиенко И. В. Метод направляющих окрестностей. — ДАН УССР. Серия А, 1978, № 8.
66. Леонтьев В. К. Дискретные экстремальные задачи. — В кн.: Теория вероятностей, математическая статистика, теоретическая кибернетика. Сер. «Итоги науки и техники», том 16, М., 1979.
67. Лепешинский Н. А., Федулов Л. В., Пирьянович В. А., Кацубо В. В. Разработка математических мо-

делей, алгоритмов и программ по задачам подсистемы планирования и управления горными работами (Управление качеством и оперативное управление). — М.: ВНТИЦ, 1978, № В 681214. — Научный отчет.
68. Лидов М. Л. Математическая аналогия между некоторыми оптимальными задачами коррекции траекторий и выбора состава измерений и алгоритмы их решения. — Космич. исслед., 1971, 9, № 5.
69. Метельский Н. Н. Об экстремальных значениях линейной формы на некоторых множествах. — Изв. АН БССР. Серия физ.-мат. наук, 1972, № 5.
70. Микульский В. Е. Об экстремуме суммы линейных форм на множестве подстановок. — ДАН БССР, 1974, 18, № 10.
71. Микульский В. Е. Об экстремальных значениях скалярного произведения матриц. — Изв. АН БССР. Серия физ.-мат. наук, 1975, № 3.
72. Михалевич В. С. Последовательные алгоритмы оптимизации и их применение. I. — Кибернетика, 1965, № 1.
73. Михалевич В. С. Последовательные алгоритмы оптимизации и их применение. II. — Кибернетика, 1965, № 2.
74. Михалевич В. С. и др. Вычислительные методы оптимальных проектных решений. — Киев: Наукова думка, 1977.
75. Михалевич В. С., Ермольев Ю. М., Шкурба В. В., Шор Н. З. Сложные системы и решение экстремальных задач. — Кибернетика, 1967, № 5.
76. Михолан В. В., Павлечко В. А., Полонский В. В. Опыт применения некоторых алгоритмов целочисленного линейного программирования при решении задач в АСУ производства бриллиантов. — В кн.: Автоматизированные системы управления, Минск, 1975, вып. 3 (21).
77. Моисеев Н. Н. Численные методы в теории оптимальных систем. — М.: Наука, 1971.
78. Моисеев Н. Н. Элементы теории оптимальных систем. — М.: Наука, 1975.
79. Моисеев Н. Н., Иванилов Ю. П., Столярова Е. М. Методы оптимизации. — М.: Наука, 1968.
80. Моцкус И. Б. Многоэкстремальные задачи в проектировании. — М.: Наука, 1967.
81. Павлечко В. А. Реализация пакета прикладных программ для технико-экономического планирования. — В кн.: Вычислительная техника социалистических стран, 1979, № 5.
82. Пирьянович В. А., Гольбин Д. А. Применение локально-стохастической оптимизации для планирования ремонтов в энергосистеме. — Вестник БГУ. Серия 1, 1979, № 1.
83. Пирьянович В. А. Машинные эксперименты по решению задач целочисленного линейного программирования с помощью локально-стохастических алгоритмов. — ЖВМ и МФ, 1979, 19, № 1.
84. Пирьянович В. А., Ковалев М. М., Федулов Л. В., Кудрявцев П. Д. Оптимизация отгрузки руды с горных участков на калийном руднике. — Изв. вузов (Горный журнал), 1977, № 5.
85. Растригин Л. А. Статистические методы поиска. — М.: Наука, 1968.

86. Растригин Л. А. Случайный поиск с линейной тактикой. — Рига. Зинатне, 1971.
87. Розенталь Ю. М., Комлик В. М., Емеличев В. А. Опыт решения задач по определению оптимальных мощностей и размещению предприятий с учетом нелинейности функционала и целочисленности решения. — В кн.: Тезисы докладов Всесоюзной конференции по математическому оптимальному программированию, Новосибирск, 1965.
88. Саати Т. Целочисленные методы оптимизации и связанные с ними экстремальные проблемы. — М.: Мир, 1973.
89. Сергиенко И. В. Вопросы разработки одного подхода к решению дискретных задач оптимизации в системах обработки данных и автоматизированных системах управления. — Управляющие системы и машины, 1974, № 6.
89′. Сергиенко И. В., Лебедева Т. Т., Рощин В. А. Приближенные методы решения дискретных задач оптимизации. — Киев: Наукова думка, 1980.
90. Сергиенко И. В. Применение метода вектора спада для решения задач целочисленного программирования. — Управляющие системы и машины, 1975, № 3.
91. Система моделей оптимального планирования. — М.: Наука, 1975.
92. Современное состояние теории исследования операций. — М.: Наука, 1979.
93. Супруненко Д. А. О значениях линейной формы на множестве подстановок. — Кибернетика, 1968, № 2.
94. Уздемир А. П. Динамическое размещение производств. I. — Автоматика и телемеханика, 1979, № 11.
95. Уздемир А. П. Динамическое размещение производств. II. — Автоматика и телемеханика, 1979, № 12.
96. Федоренко Н. П. Некоторые вопросы теории и практики планирования и управления. — М.: Наука, 1979.
97. Финкельштейн Ю. Ю. Приближенные методы и прикладные задачи дискретного программирования. — М.: Наука, 1976.
98. Хачатуров В. Р., Астахов Н. Д. Динамические задачи размещения (модели и методы решения). — Экономика и матем. методы, 1976, XII, № 1.
99. Хачатуров В. Р. Аппроксимационно-комбинаторный метод и некоторые его приложения. — ЖВМ и МФ, 1974, 14, № 6.
100. Ху Т. Целочисленное программирование и потоки в сетях. — М.: Мир, 1974.
101. Черенин В. П. Решение некоторых комбинаторных задач оптимального планирования методом последовательных расчетов (Материалы к конф. по опыту и перспективам применения матем. методов и ЭВМ в планировании). — Новосибирск: Изд-во СО АН СССР, 1962.
102. Щербина О. А. О квазиблочных экономико-математических моделях.— В кн.: Экономико-математическое моделирование развития отраслей и транспорта, Киев, ИК АН УССР, 1978.
103. Щербина О. А. Математическая модель оптимального предварительного резервирования гостиничных номеров. — В кн.: Сборник работ по математической кибернетике, М., ВЦ АН СССР, 1976, вып. 1.

104. Яблонский С. В. Об алгоритмических трудностях синтеза минимальных контактных схем. — В кн.: Проблемы кибернетики, М., 1959, вып. 2.
105. Balas E. Discrete programming by the filter method. — Operations Research, 1967, **15**, № 5.
106. Balas E. Note on the branch-and-baund principle. — Operations Research, 1968, **16**, № 2.
107. Garfinkel R. S., Nemhauser G. L. Integer programming. — New York — London — Sydney — Toronto: J. Willey and Sons, 1972.
108. Geoffrion A. M., Marsten R. E. Integer programming algorithms: a framework and state of the art survey. — Management Science, 1972, **18**, № 9.
109. Gomory R. E. Outline of an algorithm for integer solution to linear programs. — Bull. Amer. Math. Soc., 1958, **64**, № 5.
110. Guba W. Ein maximaler lokaler Algorithmus für Aufgaben der ganzzahliegen linearen Optimierung. — Elektronische Informationsverarbeitung und Kybernetik, 1975, **11**, № 10—12.
111. Integer programming and related areas. A classified bibliography. — Lect. Notes. Econ. and Math. Syst., 1976, № 128.
112. Integer programming and related areas. A classified bibliography. — Lect. Notes. Econ. and Math. Syst., 1979, № 145.
113. Kelly J. E. The cutting plane method for solving convex programs. — J. Soc Industr. Appl. Math., 1960, **8**, № 4.
114. Kowaljov M., Girlich F. Über einige Aufgaben der optimalen Standartisierung. — Die Technik, 1976, № 9.
115. Kowaljov M., Girlich F. Zum Problem der optimalen Standartisierung. — Math. Operationsforsch. Statist. (Ser. Optimization), 1977, **8**, № 1.
116. Land A. H., Doig A. G. An automatic method of solving discrete programming problems. — Econometrica, 1960, **28**, № 3.
117. Piehler J. Ganzzahliege lineare Optimirung (Methoden und Probleme). — Leipzig: Teubner, 1970.
118. Schoch M. Das Erweiterungsprinzip und seine Anwendung zur Entwicklung von Algorithmen für die Lösung kombinatorischer Optimierungsaufgaben. — Berlin: VEB Deutscher Verlag der Wissenschaften, 1976.

Емеличев
Владимир Алексеевич

Комлик
Владимир Иванович

МЕТОД ПОСТРОЕНИЯ
ПОСЛЕДОВАТЕЛЬНОСТИ ПЛАНОВ
ДЛЯ РЕШЕНИЯ ЗАДАЧ
ДИСКРЕТНОЙ ОПТИМИЗАЦИИ

Серия: «Экономико-математическая библиотека»

Редактор Е. Ю. Ходан

Техн. редактор И. Ш. Аксельрод

Корректоры Т. С. Плетнева, Л. С. Сомова

ИБ № 11859

Сдано в набор 12.11.80. Подписано к печати 10.07.81. Т-20089. Формат 84×108$^{1}/_{32}$. Бумага тип. № 1. Литературная гарнитура. Высокая печать. Условн. печ. л. 10,92. Уч.-изд. л. 10,61. Тираж 6000 экз. Заказ № 888. Цена 1 р. 10 к.

Издательство «Наука»
Главная редакция физико-математической литературы
117071, Москва, В-71, Ленинский проспект, 15

Ленинградская типография № 2 головное предприятие ордена Трудового Красного Знамени Ленинградского объединения «Техническая книга» им. Евгении Соколовой Союзполиграфпрома при Государственном комитете СССР по делам издательств, полиграфии и книжной торговли.
198052, г. Ленинград, Л-52, Измайловский проспект, 29.

1 р. 10 к.