

→ КОНСТРУКТИВНЫЕ МЕТОДЫ ОПТИМИЗАЦИИ



КОНСТРУКТИВНЫЕ МЕТОДЫ ОПТИМИЗАЦИИ

Р. Габасов
Ф.М. Кириллова
А.И. Тятюшкин

1

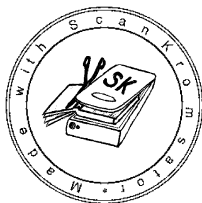
**Р. ГАБАСОВ,
Ф. М. КИРИЛЛОВА,
А. И. ТЯТЮШКИН**

КОНСТРУКТИВНЫЕ МЕТОДЫ ОПТИМИЗАЦИИ

Часть I

ЛИНЕЙНЫЕ ЗАДАЧИ

**Минск
Издательство «Университетское»
1984**



Scan AAW

Рекомендовано
Советом факультета прикладной математики
Белгосуниверситета имени В. И. Ленина

Рецензент — Б. Н. Пшеничный,
доктор физико-математических наук

Габасов Р., Кириллова Ф. М., Тятюшкин А. И. Конструктивные методы оптимизации. Ч. 1. Линейные задачи.— Мн.: Изд-во «Университетское», 1984.—214 с.

Рассматриваются статические линейные задачи общего вида. Для их решения построены алгоритмы различных классов. В основу прямых точных алгоритмов положен созданный ранее в Минске адаптивный метод. Исследованы модификации метода (конечная, μ -адаптивная, с блочной заменой, многошаговая) с повышенной эффективностью. Разработаны конечные двойственные точные алгоритмы. Построена новая теория безопорных методов. Излагается новый подход к созданию приближенных методов, на итерациях которых допускается нарушение ограничений. Много внимания уделяется алгоритмам решения больших задач. Новые алгоритмы программно реализованы на ЭВМ, приведены результаты массовых численных экспериментов.

Рассчитана на специалистов, занятых решением разнообразных прикладных оптимизационных задач. Ее материал можно использовать в учебных целях при подготовке специалистов по прикладной математике.

Библиогр. 44 назв. Ил. 4. Табл. 30. Прилож. с. 201—211.

Г $\frac{1502000000-004}{\text{М 317-84}}$ 25—83

ОГЛАВЛЕНИЕ

Предисловие	4
<i>Глава 1. ОБЩИЕ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИ-</i> <i>РОВАНИЯ</i>	6
§ 1. Адаптивный метод	9
§ 2. Вариант метода с пересчетом оценок	33
§ 3. Мультипликативная форма	37
§ 4. Адаптивный алгоритм для канонической задачи	43
§ 5. Алгоритм с симплексной нормировкой	47
§ 6. Конечные модификации	50
§ 7. μ -адаптивный метод	74
§ 8. Блочный метод	90
§ 9. Многошаговые методы	96
§ 10. Безопорные методы	110
§ 11. Приближенные методы	116
§ 12. Особенности реализации алгоритмов линейно- го программирования на ЭВМ	122
<i>Глава 2. БОЛЬШИЕ ЗАДАЧИ</i>	129
§ 1. Задачи с большим числом переменных	131
§ 2. Задачи с большим числом ограничений	140
§ 3. Декомпозиция опоры канонической задачи	147
§ 4. Адаптивный метод с декомпозицией опоры	161
§ 5. Прикладные задачи	169
§ 6. Адаптивный метод решения динамической транс- портной задачи	174
§ 7. Вычислительная технология решения больших задач оптимизации	194
Комментарии к главе 1	197
Комментарии к главе 2	199
Приложение	201
§ 1. Программа адаптивного метода с оптимальной заменой элементов опоры	201
§ 2. Программа опорного метода	202
§ 3. Программа адаптивного метода с декомпози- цией опоры	203
Литература	212

ПРЕДИСЛОВИЕ

Оптимизационные задачи продолжают оставаться одним из основных объектов прикладной математики. Их значение в течение последних 30—35 лет неуклонно растет. В разные периоды развития современных методов оптимизации на первый план выходили задачи разных типов, но общая тенденция состояла в постоянном расширении класса порожденных практическими запросами моделей и в постепенном переходе от их качественного изучения к построению конструктивных алгоритмов. В качественной теории к настоящему времени разработана весьма общая теория необходимых условий оптимальности. С точки зрения классических методов непрямого типа это означает, что создана база для численных методов решения исключительно широкого класса задач. Однако практика развития современных численных методов математического программирования показывает, что на пути не прямых методов встречаются серьезные трудности. Первым результатом в области методов прямого типа для решения неклассических экстремальных задач, который оказал большое влияние на развитие численных методов оптимизации, явился симплекс-метод линейного программирования.

Линейные модели широко используются в качественной и конструктивной теориях всех разделов математики. Они с момента создания симплекс-метода привлекались и для построения численных методов нелинейного программирования [39]. Как показывает опыт работы [34], модели и методы линейного программирования эффективны и в оптимальном управлении.

В каждой конкретной сфере приложений линейных моделей возникают специфические особенности. От их полного учета во многом зависит эффективность метода решения всей задачи. Пример [34] и другие применения линейных моделей показывают, что на современном этапе развития методов оптимизации стала весьма актуальной проблема разработки хороших алгоритмов решения специальных задач линейного программирования, имеющих большие размеры.

Этой проблеме посвящена и данная монография. При этом основным предметом исследований в ней являются большие задачи, возникающие в теории оптимального управления. В идейном плане монография непосредственно примыкает к работам [5, 6, 8], в которых разработан новый подход к конструктивной теории линейного программирования. В данной работе этот подход развивается дальше, дополняется новыми результатами. Методы, разработанные в [5, 6, 8], детализируются и испытываются на ряде численных машинных экспериментов.

Теоретическая часть монографии основана на результатах Р. Габасова, Ф. М. Кирилловой, их сотрудников. Основная работа, связанная с реализацией и испытанием алгоритмов, выполнена А. И. Тятюшкиным. Дополнительные эксперименты проведены авторами соответствующих разделов (они перечислены ниже).

Кратко о содержании монографии. В гл. 1 сначала детально излагаются некоторые ключевые методы из [5, 6, 8] с тем, чтобы любой программист смог по приведенным описаниям составить программу для ЭВМ. При этом в отличие от [5, 6, 8] приводится несколько форм алгоритмов, удобных для разных целей. Затем разрабатываются обобщения построенных ранее методов, которые оказываются конечными методами решения любой задачи линейного программирования. Приведенные здесь правила выхода из цикла принципиально отличаются от известных [26]. Надо подчеркнуть, что новый материал выходит за рамки теоретических рассуждений. Подобные вопросы в последнее время привлекают внимание многих ученых в связи с их возросшим прикладным значением. В остальной части главы изложены методы, не вошедшие в [5, 6, 8]. К ним, в частности, относятся многшаговые и приближенные методы.

Вторая глава посвящена алгоритмам решения больших задач, в которых матрица условий не имеет специальной структуры. Здесь излагается несколько общих принципов построения алгоритмов, не опирающихся на специфику задач. Подобные алгоритмы интересны как дежурные средства для решения (возможно, не всегда лучшим образом) задач, специфика которых пока четко не описана. Как и в гл. 1, основные методы разработаны до деталей, испытаны на многих задачах. Глава завершается изучением нескольких прикладных задач.

Монография снабжена приложением, в котором приводятся тексты программ основных алгоритмов. В библиографии указаны лишь цитируемые работы.

Данная монография, состоящая из пяти частей, как и ранее изданная в трех частях монография по методам линейного программирования, ни в коей мере не претендует на энциклопедичность. Она содержит описание, обоснование и результаты экспериментальной проверки на ЭВМ разнообразных алгоритмов, основанных на едином подходе к проблемам оптимизации, который был предложен Р. Габасовым и Ф. М. Кирилловой в начале 70-х годов. В ней отражен многолетний опыт работы участников Минского семинара по конструктивным методам оптимального управления, функционирующего на базе кафедры методов оптимального управления факультета прикладной математики Белорусского ордена Трудового Красного Знамени государственного университета им. В. И. Ленина и лаборатории теории процессов управления Института математики АН БССР.

В книгу включены результаты В. С. Глушенкова (гл. 1 § 1 п. 14, § 7), О. И. Костюковой (гл. 1 § 6, 11, гл. 2 § 6), А. А. Сенько (гл. 1 § 1 пп. 11—13), Фам Тхе Лонга (гл. 1 § 8, 9, гл. 2 § 3 п. 7), Е. И. Шилкиной (гл. 1 § 10). В подготовке рукописи принимали участие А. В. Гуминский и А. В. Покатаев. Всем указанным товарищам авторы выражают искреннюю благодарность.

А в т о р ы

Глава 1. ОБЩИЕ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Линейные оптимизационные задачи известны в некоторых общих формах. С теоретической точки зрения они эквивалентны между собой, поскольку к любой из них может быть сведена каждая линейная задача оптимизации. При численном решении прикладных задач предпочтение отдается той форме, которая ближе к математической модели задачи и не требует дополнительных преобразований, увеличивающих ее размеры. В первые годы развития теории линейного программирования удобной была нормальная (стандартная) форма с ограничениями-неравенствами

$$c'x \rightarrow \max, Ax \leq b, x \geq 0.$$

Численные методы, построенные по образцу симплекс-метода, опираются, как правило, на каноническую форму (с основными ограничениями-равенствами)

$$c'x \rightarrow \max, Ax = b, x \geq 0. \quad (1)$$

В машинных программах симплекс-алгоритмов любая линейная задача

$$c'x \rightarrow \max, A_{(1)}x \leq b_{(1)}, A_{(2)}x = b_{(2)}, A_{(3)}x \geq b_{(3)}, \quad (2)$$

$$x_{(1)} \geq 0, x_{(2)} \leq 0, x = (x_{(1)}, x_{(2)}, x_{(3)})$$

автоматически сводится к канонической форме (1) несмотря на то, что при этом теряется специфика ограничений (2), увеличиваются размеры задачи. Это, естественно, влияет и на эффективность численного решения. Однако в начальный период развития симплексных методов указанный путь был популярным в силу своей универсальности и того благоприятного факта, что размеры решаемых в то время прикладных задач не служили пре-

пятствием для реализации стандартных методов на ЭВМ.

В современных машинных программах некоторые частные задачи из (2), например,

$$c'x \rightarrow \max, Ax=b, 0 \leq x \leq d,$$

уже не сводятся к (1). Для их решения используется модификация симплекс-алгоритма, существенно увеличивающая эффективность алгоритма. Имеются и другие задачи из (2), для которых разработаны соответствующие варианты симплекс-алгоритма.

Среди общих задач линейного программирования с середины 70-х годов стала выделяться задача в естественной форме (интервальная задача)

$$c'x \rightarrow \max, b_* \leq Ax \leq b^*, d_* \leq x \leq d^*.$$

Она положена и в основу алгоритмов данной главы. Последние базируются на подходе, развитом в монографии [8].

Напомним, несмотря на то, что в реальных расчетах симплекс-метод проявил себя с наилучшей стороны, с момента возникновения не прекращалась работа по его совершенствованию. Большие усилия были направлены на обоснование новых правил выбора ведущего столбца, разработку блочных способов замены базисной матрицы, при этом неизменным оставался основной элемент метода — базисный план. Итерации по базисным планам обладают определенными достоинствами, но это не единственный метод организации точных методов. Более того, обязательное соблюдение базисности промежуточных планов можно рассматривать как ограничение на метод, не вызванное исходной задачей оптимизации. С начала 70-х годов в Минске ведутся работы по созданию точных методов, не опирающихся на базисные планы. Результаты, полученные до 1977 г., вошли в монографию [5]. Дальнейшие разработки в этом направлении отражены в данной главе. Однако здесь содержатся не только новые теоретические результаты, но и результаты численных экспериментов, объем которых к моменту написания монографии [8] был еще невелик.

В настоящее время накоплен достаточно обширный экспериментальный материал как по методам монографии [8], так и по новым методам, включенным в данную книгу. По существу, каждый новый принцип построения метода, каждый новый элемент итерации теперь прохо-

дит проверку в численном эксперименте. Работа эта очень трудоемкая, но, по нашему мнению, чрезвычайно актуальная на современном этапе развития методов оптимизации. Многочисленные теоретические работы, содержащие только общие рекомендации, но не подкрепленные четким описанием области их эффективных приложений, не столько способствуют, сколько тормозят прогресс методов оптимизации из-за возможности появления ложных направлений исследования.

Приведем краткое описание содержания гл. 1.

В § 1 излагается алгоритм адаптивного метода, предложенного в [8]. Приводятся результаты новых численных экспериментов. Версия адаптивного алгоритма с пересчетом оценок описана в § 2. Третий параграф посвящен построению мультипликативного варианта адаптивного метода, который от основного варианта отличается формой представления в памяти ЭВМ опорной матрицы.

В § 4, 5 строятся два алгоритма для решения канонической задачи (в предыдущих параграфах рассматривалась интервальная задача). Сначала описывается вариант адаптивного метода, а затем прямой метод с симплексной нормировкой, являющийся непосредственным развитием симплекс-метода на небазисные планы. По-другому эта задача решается в [5].

Следующие два параграфа (§ 6, 7) содержат результаты по обоснованию конечных алгоритмов. В § 6 конечность достигается за счет использования вырожденных сопровождающих двойственных планов. Более общий случай исследуется в § 7. Проблема конечности (другими словами, задача избежания заикливания) из плоскости теоретических рассуждений постепенно переходит в плоскость конструктивных построений. Она широко обсуждается в специальной литературе. Опасность заикливания и связанное с этим явление замедление сходимости становятся реальной угрозой при решении практических задач.

Вопросы повышения эффективности адаптивного алгоритма рассматриваются в § 8, 9. Сначала строится вариант с блочной заменой опоры, затем — несколько многошаговых вариантов. Предварительные эксперименты подтвердили перспективность обоих направлений развития адаптивного метода.

Новому изложению безопорных методов решения общей задачи линейного программирования посвящен § 10.

Здесь обсуждаются различные варианты реализации общей схемы, предлагаются новые алгоритмы решения вспомогательных задач. Перспективна идея использования субоптимальных планов вспомогательных задач.

Приближенные методы решения общей задачи разрабатываются в § 11. Они основаны на идеях векторной оптимизации и включают в себя ряд известных методов с использованием модифицированной функции Лагранжа.

Глава завершается обсуждением вопросов численной реализации алгоритмов на ЭВМ.

§ 1. Адаптивный метод

Адаптивный метод возник в результате длительной работы над развитием симплекс-метода [8]. В данной монографии он положен в основу всех излагаемых методов. Поэтому первый параграф посвящен детальному изложению основных элементов адаптивного метода. Все операции метода аккуратно выписаны, формулы доведены до такой формы, которая удобна для составления алгоритмов, программ (см. приложение). С момента создания адаптивного метода в Институте математики АН БССР, в Белгосуниверситете им. В. И. Ленина, в ИВЦ СО АН СССР постоянно ведутся численные эксперименты, в которых этот метод всесторонне испытывается, сравнивается с другими. Многие результаты этих экспериментов содержатся в соответствующих местах данной монографии. Часть из них помещена и в первом параграфе.

1. Постановка задачи. Основные обозначения. Рассмотрим интервальную задачу линейного программирования

$$c'x \rightarrow \max, \quad b_* \leq Ax \leq b^*, \quad d_* \leq x \leq d^*. \quad (1)$$

Здесь c, x, d_*, d^* — n -векторы; b_*, b^* — m -векторы; $A = A(I, J)$ — $m \times n$ -матрица, $I = \{1, 2, \dots, m\}$ — множество индексов основных ограничений (строк матрицы A), $J = \{1, 2, \dots, n\}$ — множество индексов переменных (столбцов матрицы A).

Квадратную матрицу $A_{\text{оп}} = A(I_{\text{оп}}, J_{\text{оп}})$, составленную из элементов a_{ij} , $i \in I_{\text{оп}} \subset I$, $j \in J_{\text{оп}} \subset J$, матрицы A , будем называть опорной матрицей, если $\det A_{\text{оп}} \neq 0$, $I_{\text{оп}}$ — множество индексов опорных строк, $J_{\text{оп}}$ — множество индексов опорных столбцов, $|I_{\text{оп}}| = |J_{\text{оп}}|$. Обозначим далее $I_{\text{н}} = I \setminus I_{\text{оп}}$; $J_{\text{н}} = J \setminus J_{\text{оп}}$ — множество индексов неопорных

строк, столбцов соответственно; $a'_i = A(i, J)$ — i -я строка A ; $I_*(x) = \{i \in I: A(i, J)x = b_{*i}\}$, $I^*(x) = \{i \in I: A(i, J)x = b_i^*\}$, $\tilde{I}(x) = \{i \in I: b_{*i} < A(i, J)x < b_i^*\}$, $J_*(x) = \{j \in J: x_j = d_{*j}\}$, $J^*(x) = \{j \in J: x_j = d_j^*\}$, $\tilde{J}(x) = \{j \in J: d_{*j} < x_j < d_j^*\}$.

Вектор x , на котором выполняются основные ($b_* \leq Ax \leq b^*$) и прямые ($d_* \leq x \leq d^*$) ограничения, называется планом задачи (1). План x^0 , доставляющий целевой функции $c'x$ максимальное значение, назовем оптимальным. Субоптимальный (ε -оптимальный) план определяется неравенством $c'x^0 - c'x^\varepsilon \leq \varepsilon$. Пара $\{x, A_{\text{оп}}\}$ из плана и опоры задачи (1) называется опорным планом. Опорный план считается невырожденным, если он не вырожден по основным ($b_*(I_{\text{н}}) < A(I_{\text{н}}, J)x(J) < b^*(I_{\text{н}})$) и прямым ($d_*(J_{\text{оп}}) < x(J_{\text{оп}}) < d^*(J_{\text{оп}})$) ограничениям.

Необходимо создать алгоритм, который для любого $\varepsilon \geq 0$ при любом начальном опорном плане $\{x, A_{\text{оп}}\}$ за конечное число итераций строит ε -оптимальный план.

2. Формула приращения. Пусть $\{x, A_{\text{оп}}\}$ — опорный план задачи (1). Подсчитаем на нем $\omega_* = b_* - Ax$, $\omega^* = b^* - Ax$ — нижний и верхний векторы невязок, $u' = u'(I_{\text{оп}}) = c'(J_{\text{оп}})A_{\text{оп}}^{-1}$ — вектор потенциалов, $\Delta' = \Delta'(J) = u'A(I_{\text{оп}}, J) - c'(J)$ — вектор оценок. Тогда при любом другом плане $\bar{x} = x + \Delta x$ приращение $c'\bar{x} - c'x$ целевой функции

$$c'\Delta x = [u'A(I_{\text{оп}}, J) - \Delta'(J)]\Delta x(J).$$

Введя обозначение $z(I_{\text{оп}}) = A(I_{\text{оп}}, J)\Delta x(J)$ и учитывая, что $\Delta(J_{\text{оп}}) = 0$, получим

$$c'(J)\Delta x(J) = u'(I_{\text{оп}})z(I_{\text{оп}}) - \Delta'(J_{\text{н}})\Delta x(J_{\text{н}}). \quad (2)$$

3. Критерий оптимальности. Анализ формулы (2) позволяет получить условия, при которых любые допустимые вариации переменных не увеличивают значений целевой функции.

Теорема 1. Соотношения

$$u(I_{* \text{оп}}) \leq 0, u(I_{\text{оп}}^*) \geq 0, u(\tilde{I}_{\text{оп}}) = 0, \quad (3)$$

$$\Delta(J_{* \text{н}}) \geq 0, \Delta(J_{\text{н}}^*) \leq 0, \Delta(\tilde{J}_{\text{н}}) = 0$$

достаточны, а в случае невырожденности и необходимы для оптимальности опорного плана $\{x, A_{\text{оп}}\}$.

4. Критерий субоптимальности. При нарушении соотношений (3) найдется такое допустимое приращение Δx , что $c'\Delta x > 0$. Допустимые приращения Δx удовлетворяют неравенствам

$$\begin{aligned} d_*(J_H) - x(J_H) &\leq \Delta x(J_H) \leq d^*(J_H) - x(J_H), \\ \omega_*(I_{оп}) &\leq z(I_{оп}) \leq \omega^*(I_{оп}). \end{aligned} \quad (4)$$

Максимальное значение приращения целевой функции (2) при ограничении (4) достигается при

$$\begin{aligned} \Delta x_j &= \begin{cases} d_{*j} - x_j, & \text{если } \Delta_j > 0, \\ d_j^* - x_j, & \text{если } \Delta_j < 0, \\ 0, & \text{если } \Delta_j = 0, \end{cases} \quad j \in J_H, \\ z_i &= \begin{cases} \omega_i^*, & \text{если } u_i > 0, \\ \omega_{*i}, & \text{если } u_i < 0, \\ 0, & \text{если } u_i = 0, \end{cases} \quad i \in I_{оп}, \end{aligned} \quad (5)$$

и равно

$$\begin{aligned} \beta &= \sum_{\Delta_j > 0} \Delta_j (x_j - d_{*j}) + \sum_{\Delta_j < 0} \Delta_j (x_j - d_j^*) + \\ &+ \sum_{u_i < 0} u_i \omega_{*i} + \sum_{u_i > 0} u_i \omega_i^*. \end{aligned} \quad (6)$$

Поскольку неравенства (4) не учитывают неопорные основные и опорные прямые ограничения на допустимые приращения, то $c'x^0 - c'x \leq \beta$, т. е. β — оценка субоптимальности плана x .

Рассмотрим задачу, двойственную к (1):

$$\begin{aligned} \psi(\lambda) &= b^*s - b^*t + d^*w - d^*v \rightarrow \min, \\ A'y + w - v &= c, \quad y = s - t, \quad s \geq 0, \quad t \geq 0, \quad w \geq 0, \quad v \geq 0. \end{aligned} \quad (7)$$

По опоре $A_{оп} = A(I_{оп}, J_{оп})$ построим вектор $\lambda = \{y, s, t, v, w\}$ с компонентами

$$\begin{aligned} y(I_{оп}) &= u, \quad y(I_H) = 0, \quad s_i = y_i, \quad t_i = 0, \quad \text{если } y_i \geq 0; \\ s_i &= 0, \quad t_i = -y_i, \quad \text{если } y_i < 0, \quad i \in I; \\ v_j &= \Delta_j, \quad w_j = 0, \quad \text{если } \Delta_j \geq 0; \\ v_j &= 0, \quad w_j = -\Delta_j, \quad \text{если } \Delta_j < 0, \quad j \in J, \end{aligned} \quad (8)$$

который, как легко проверить, является планом задачи (7). С помощью прямого x и двойственного λ планов из (6), (8) получим

$$\beta = (c'x^0 - c'x) + (b^{*'}s - b_*'t - d_*'v + d^{*'}w - b^{*'}s^0 + b_*'t^0 + d_*'v^0 - d^{*'}w^0) = \beta_x + \beta_{\text{оп}}, \quad (9)$$

где $\beta_x = c'x^0 - c'x$ — мера неоптимальности плана x ; $\beta_{\text{оп}} = \psi(\lambda) - \psi(\lambda^0)$ — мера неоптимальности опоры.

Если плану x приписать опору $A_{\text{оп}}^0$, сопровождающий двойственный план которой является оптимальным двойственным базисным планом задачи (1), то $\beta_{\text{оп}} = 0$ и $\beta = \beta_x$, т. е. оценка субоптимальности станет точной: $c'x^0 - c'x = \beta$.

Теорема 2. При $\beta \leq \varepsilon$ опорный план $\{x, A_{\text{оп}}\}$ является ε -оптимальным планом задачи (1). Для каждого ε -оптимального плана x^ε существует такая опора $A_{\text{оп}}$, что оценка субоптимальности β опорного плана $\{x^\varepsilon, A_{\text{оп}}\}$ удовлетворяет неравенству $\beta \leq \varepsilon$.

5. Итерация. Построение подходящего направления. В адаптивном алгоритме для построения подходящего направления решается задача максимизации приращения целевой функции (2) при ограничениях (4). В отличие от максимизации производной, применяемой в симплекс-методе, задача (2), (4) полнее отражает цель и структуру ограничений исходной задачи (1).

Решение задачи (2), (4) строится по формуле (5). Используя эту формулу и обозначение $z(I_{\text{оп}}) = A(I_{\text{оп}}, J_{\text{оп}})\Delta x(J_{\text{оп}}) + A(I_{\text{оп}}, J_{\text{н}})\Delta x(J_{\text{н}})$, построим подходящее направление $l = \{l(J_{\text{оп}}), l(J_{\text{н}})\}$:

$$l_j = \begin{cases} d_j^* - x_j, & \text{если } \Delta_j < 0, \\ d_{*j} - x_j, & \text{если } \Delta_j > 0, \\ 0, & \text{если } \Delta_j = 0, \end{cases} \quad j \in J_{\text{н}}; \quad (10)$$

$$l(J_{\text{оп}}) = A_{\text{оп}}^{-1} \omega(I_{\text{оп}}) - A_{\text{оп}}^{-1} A(I_{\text{оп}}, J_{\text{н}}) l(J_{\text{н}}),$$

где

$$\omega(I_{\text{оп}}) = \{\omega_i, i \in I_{\text{оп}}\}, \quad \omega_i = \begin{cases} \omega_{*i}, & \text{если } u_i < 0, \\ \omega_i^*, & \text{если } u_i > 0, \\ 0, & \text{если } u_i = 0. \end{cases}$$

6. Итерация. Вычисление шага. В случае оптимальной опоры $A_{\text{оп}} (\beta_{\text{оп}} = 0)$ и невырожденного сопровождающего двойственного плана подходящее направление (10) за одну итерацию приводит к оптимальному плану $\bar{x} = x + l$. В общем случае неизвестно, оптимальна ли опора. Поэтому для построения \bar{x} вычислим шаг Θ вдоль $l(\bar{x} = x + \Theta l)$.

Поскольку условия (4) учтены при построении l , то они, очевидно, будут выполнены на \bar{x} при $\Theta \leq 1$. Найдем максимально допустимый шаг Θ по опорным прямым и неопорным основным ограничениям:

$$\Theta_{j_0} = \min \Theta_j, \quad j \in J_{\text{оп}}, \quad \Theta_j = \begin{cases} (d_{*j} - x_j)/l_j, & \text{если } l_j < 0, \\ (d_j^* - x_j)/l_j, & \text{если } l_j > 0, \\ \infty, & \text{если } l_j = 0; \end{cases} \quad (11)$$

$$\Theta_{i_0} = \min \Theta_i, \quad i \in I_{\text{н}}, \quad \Theta_i = \begin{cases} \omega_{*i}/a_i' l, & \text{если } a_i' l < 0, \\ \omega_i^*/a_i' l, & \text{если } a_i' l > 0, \\ \infty, & \text{если } a_i' l = 0. \end{cases} \quad (12)$$

Следовательно, максимально допустимый шаг Θ вдоль l , при котором $\bar{x} = x + \Theta l$ остается планом задачи (1),

$$\Theta = \min \{1, \Theta_{j_0}, \Theta_{i_0}\}. \quad (13)$$

При переходе от x к новому плану \bar{x} целевая функция получает приращение $\Theta c' l \geq 0$, причем $c' x^0 - c' \bar{x} = c' x^0 - c' x - \Theta c' l \leq \beta - \Theta c' l = \beta - \Theta(u' A(I_{\text{оп}}, J) - \Delta') l(J) = \beta + \Theta(\Delta' l - u' \omega(I_{\text{оп}})) = \beta - \Theta \beta = (1 - \Theta) \beta$, т. е. при $(1 - \Theta) \beta \leq \varepsilon$ план \bar{x} — ε -оптимальный, задача (1) решена. При $\bar{\beta} = (1 - \Theta) \beta > \varepsilon$ итерация продолжается: улучшается компонента $A_{\text{оп}}$ начальной информации $\{x, A_{\text{оп}}\}$.

7. Итерация. Замена сопровождающего двойственного плана. Опору $A_{\text{оп}}$ заменяем на опору $\bar{A}_{\text{оп}}$ такую, чтобы уменьшилось второе слагаемое в оценке субоптимальности (9). Уменьшения $\beta_{\text{оп}}$ можно добиться с помощью итерации двойственного метода.

Введем обозначения: $\Delta(J_{\text{н}} \cup I_{\text{оп}}) = \{\Delta(J_{\text{н}}), -u(I_{\text{оп}})\}$; $\kappa(J_{\text{н}} \cup I_{\text{оп}}) = \{x(J_{\text{н}}) + l(J_{\text{н}}), A(I_{\text{оп}}, J)[x(J) + l(J)]\}$; $\tilde{d}^*(J_{\text{н}} \cup I_{\text{оп}}) = \{d^*(J_{\text{н}}), b^*(I_{\text{оп}})\}$; $\tilde{d}_*(J_{\text{н}} \cup I_{\text{оп}}) = \{d_*(J_{\text{н}}), b_*(I_{\text{оп}})\}$. Подходящее направление для вектора $\Delta(J_{\text{н}} \cup I_{\text{оп}})$ в случае $\Theta = \Theta_{i_0} < 1$ вычисляется по формуле

$$\xi'(J_{\text{н}} \cup I_{\text{оп}}) = k e_{i_0}' \{A(I_{\text{н}}, J_{\text{н}}) - A(I_{\text{н}}, J_{\text{оп}}) A_{\text{оп}}^{-1} A(I_{\text{оп}}, J_{\text{н}}), A(I_{\text{н}}, J_{\text{оп}}) A_{\text{оп}}^{-1}\}, \quad (14)$$

$$k = \begin{cases} 1, & \text{если } a_{i_0}' \bar{x} = b_{i_0}^*, \\ -1, & \text{если } a_{i_0}' \bar{x} = b_{*i_0}, \end{cases}$$

и в случае $\Theta = \Theta_{j_0} < 1$ —

$$\xi'(J_{\text{н}} \cup I_{\text{оп}}) = k e_{j_0}' \{A_{\text{оп}}^{-1} A(I_{\text{оп}}, J_{\text{н}}), -A_{\text{оп}}^{-1}\}, \quad (15)$$

$$k = \begin{cases} 1, & \text{если } \bar{x}_{j_0} = d_{*j_0}, \\ -1, & \text{если } \bar{x}_{j_0} = d_{j_0}^*, \end{cases} \quad (15)$$

где e_i — единичный вектор с единицей на i -м месте. Начальная скорость убывания двойственной функции

$$\alpha_0 = \begin{cases} \kappa_{j_0} - d_{*j_0} & \text{при } k = 1, \\ d_{j_0}^* - \kappa_{j_0} & \text{при } k = -1, \end{cases} \quad j_0 \in J_{\text{оп}}, \text{ если } \Theta = \Theta_{j_0}; \quad (16)$$

$$\alpha_0 = \begin{cases} \kappa_{i_0} - b_{*i_0} & \text{при } k = -1, \\ b_{i_0}^* - \kappa_{i_0} & \text{при } k = 1, \end{cases} \quad i_0 \in I_{\text{н}}, \text{ если } \Theta = \Theta_{i_0}. \quad (17)$$

При движении вдоль направления ξ компоненты вектора $\Delta + \sigma \xi$ меняют знаки и скорость убывания двойственной функции уменьшается (по модулю):

$$\alpha_{k+1} = \alpha_k + \sum_{\xi_j < 0, \Delta_j^{(k)} = 0} \xi_j (\kappa_j - \tilde{d}_j^*) + \sum_{\xi_j > 0, \Delta_j^{(k)} = 0} \xi_j (\kappa_j - \tilde{d}_{*j}) \quad (18)$$

(из-за появления новых положительных слагаемых). При некотором $k = s$ получим $\alpha_s \geq 0$. Тогда имеем $\bar{\Delta} =$

$$= \Delta + \xi \sum_{k=0}^{s-1} \sigma_k. \text{ Индекс } j, \text{ с которым была вычислена по-}$$

следняя положительная добавка в (18), обозначим через j_* , если $j \in J_{\text{н}}$, а при $i \in I_{\text{оп}}$ — через i_* . Соответственно обозначим и двойственный шаг $\sigma = \sigma_{j_*}$ или $\sigma = \sigma_{i_*}$, значе-

ние которого равно $\sum_{k=0}^{s-1} \sigma_k$.

8. Итерация. Построение матрицы, обратной к новой опорной матрице.

1) Пусть $\Theta = \Theta_{i_0} < 1$, $\sigma = \sigma_{i_*}$. Тогда $\bar{I}_{\text{оп}} = (I_{\text{оп}} \setminus i_*) \cup i_0$, $\bar{J}_{\text{оп}} = J_{\text{оп}}$,

$$\bar{A}_{\text{оп}}^{-1} = A_{\text{оп}}^{-1} + A_{\text{оп}}^{-1} e_{i_*} [A(i_*, J_{\text{оп}}) - A(i_0, J_{\text{оп}})] A_{\text{оп}}^{-1} / \alpha, \quad (19)$$

$$\alpha = A(i_0, J_{\text{оп}}) A_{\text{оп}}^{-1} e_{i_*} = k \xi(i_*) \neq 0.$$

2) Пусть $\Theta = \Theta_{i_0} < 1$, $\sigma = \sigma_{j_*}$. Тогда $\bar{I}_{\text{оп}} = I_{\text{оп}} \cup i_0$, $\bar{J}_{\text{оп}} = J_{\text{оп}} \cup j_*$,

$$\begin{aligned} & \bar{A}_{\text{оп}}^{-1} = \\ & = \begin{Bmatrix} A_{\text{оп}}^{-1} + A_{\text{оп}}^{-1} A(I_{\text{оп}}, j_*) A(i_0, J_{\text{оп}}) A_{\text{оп}}^{-1} / \alpha, & -A_{\text{оп}}^{-1} A(I_{\text{оп}}, j_*) / \alpha \\ -A(i_0, J_{\text{оп}}) A_{\text{оп}}^{-1} / \alpha, & 1/\alpha \end{Bmatrix}, \end{aligned} \quad (20)$$

$$\alpha = A(i_0, j_*) - A(i_0, J_{\text{оп}}) A_{\text{оп}}^{-1} A(I_{\text{оп}}, j_*) = k\xi(j_*) \neq 0.$$

3) Пусть $\Theta = \Theta_{j_0} < 1$, $\sigma = \sigma_{i_*}$. Тогда $\bar{I}_{\text{оп}} = I_{\text{оп}} \setminus i_*$, $\bar{J}_{\text{оп}} = J_{\text{оп}} \setminus j_0$,

$$\begin{aligned} \bar{A}_{\text{оп}}^{-1} &= A_{\text{оп}}^{-1}(I_{\text{оп}} \setminus i_*, J_{\text{оп}} \setminus j_0) - A_{\text{оп}}^{-1}(i_*, \\ &J_{\text{оп}} \setminus j_0) A_{\text{оп}}^{-1}(I_{\text{оп}} \setminus i_*, j_0) / \alpha, \end{aligned} \quad (21)$$

$$\alpha = A_{\text{оп}}^{-1}(i_*, j_0) = -k\xi(i_*) \neq 0.$$

4) Пусть $\Theta = \Theta_{j_0} < 1$, $\sigma = \sigma_{j_*}$. Тогда $\bar{I}_{\text{оп}} = I_{\text{оп}}$, $\bar{J}_{\text{оп}} = (J_{\text{оп}} \setminus j_0) \cup j_*$,

$$\bar{A}_{\text{оп}}^{-1} = A_{\text{оп}}^{-1} - A_{\text{оп}}^{-1} [A(I_{\text{оп}}, j_*) - A(I_{\text{оп}}, j_0)] e'_{j_0} A_{\text{оп}}^{-1} / \alpha, \quad (22)$$

$$\alpha = e'_{j_0} A_{\text{оп}}^{-1} A(I_{\text{оп}}, j_*) = k\xi(j_*) \neq 0.$$

На основе изложенных результатов построим вычислительную схему адаптивного метода.

9. Алгоритм. Пусть x — заданный план задачи (1), т. е. $b_* \leq Ax \leq b^*$, $d_* \leq x \leq d^*$. В случае, когда x не является планом, применяется стартовый алгоритм, который изложен в § 6. Если дополнительно известна опора, то вычислим векторы $u' = c'(J_{\text{оп}}) A_{\text{оп}}^{-1}$ и $\Delta' = u' A(I_{\text{оп}}, J) - c'$ и, полагая $it = 0$, перейдем в п. 2 следующего алгоритма.

1. Если информации об $A_{\text{оп}}$ нет, то положим $I_{\text{оп}} = \emptyset$, $I_{\text{н}} = \{1, 2, \dots, m\}$; $J_{\text{оп}} = \emptyset$, $J_{\text{н}} = \{1, 2, \dots, n\}$; размерность опоры $r = 0$; счетчик итераций $it = 0$; $\Delta_j = -c_j$, $j = \overline{1, n}$.

2. Вычислим векторы невязок $\omega_* = b_* - Ax$, $\omega^* = b^* - Ax$.

3. По формуле (6) найдем оценку субоптимальности β , при этом параллельно вычислим l_j , $j \in J_{\text{н}}$. При $\beta \leq \varepsilon$ перейдем к п. 21.

4. По формуле (10) вычислим $l(J_{\text{оп}})$.

5. По формулам (11) — (13) вычислим шаг Θ и одновременно найдем индекс i_0 или j_0 .

6. Построим новый план $\bar{x} = x + \Theta l$ и псевдоплан $\kappa = x + l$.

7. Положим $it = it + 1$. Вычислим значения прямой и двойственной целевых функций: $y = c'x$,

$$w = \sum_{\substack{u_i < 0, \\ i \in I_{\text{оп}}}} u_i b_{*i} + \sum_{\substack{u_i > 0, \\ i \in I_{\text{оп}}}} u_i b_i^* - \sum_{\substack{\Delta_j < 0, \\ j \in J_{\text{н}}}} \Delta_j d_j^* - \sum_{\substack{\Delta_j > 0, \\ i \in J_{\text{н}}}} \Delta_j d_{*j}.$$

Печать: it , y , w .

8. Проверим неравенство $(1 - \Theta)\beta \leq \varepsilon$. Да \rightarrow п. 21, нет \rightarrow п. 9.

9. Если $\Theta = \Theta_{j_0}$, то по формулам (15), (16) вычислим подходящее направление $\xi(J_H \cup I_{оп})$ и значение α_0 . Переход к п. 11.

10. Если $\Theta = \Theta_{i_0}$, то вычислим $\xi(J_H \cup I_{оп})$ по формуле (14), а значение α_0 — по формуле (17).

11. Положим $\Delta(I_{оп}) = -u(I_{оп})$, $\kappa(I_{оп}) = A(I_{оп}, J) \times \times (x(J) + l(J))$, $\sigma = 0$.

12. Вычислим $\sigma_{\tilde{j}} = \min \sigma_j$, $\sigma_j = -\Delta_j/\xi_j$ при $\Delta_j \xi_j < 0$, $j \in J_H \cup I_{оп}$.

13. Найдем $\bar{\Delta}_j = \Delta_j + \sigma_{\tilde{j}} \xi_j$, $j \in J_H \cup I_{оп}$, $\sigma = \sigma + \sigma_{\tilde{j}}$.

14. Если $\bar{\Delta}_j = 0$, $j \in J_H \cup I_{оп}$, то положим $\alpha = \alpha + \xi_j(\kappa_j - \gamma)$,

$$\gamma = \begin{cases} d_j^* & \text{при } \xi_j < 0, \\ d_{*j} & \text{при } \xi_j > 0, \text{ если } j \in J_H, \end{cases}$$

и

$$\gamma = \begin{cases} b_j^* & \text{при } \xi_j < 0, \\ b_{*j} & \text{при } \xi_j > 0, \text{ если } j \in I_{оп}. \end{cases}$$

15. Проверим неравенство $\alpha < 0$. Да \rightarrow п. 12, нет \rightarrow п. 16.

16. Если $\tilde{j} \in I_{оп}$, то положим $i_* = \tilde{j}$, $\sigma_{i_*} = \sigma_{\tilde{j}}$; если $\tilde{j} \in J_H$, то $j_* = \tilde{j}$, $\sigma_{j_*} = \sigma_{\tilde{j}}$.

17. Если $\Theta = \Theta_{i_0} < 1$, $\sigma = \sigma_{i_*}$, то $\bar{I}_{оп} = (I_{оп} \setminus i_*) \cup i_0$, $\bar{J}_{оп} = J_{оп}$, $u(i_0) = k\sigma_{i_*}$; $\bar{A}_{оп}^{-1}$ вычислим по формуле (19). Переход в п. 2.

18. Если $\Theta = \Theta_{i_0} < 1$, $\sigma = \sigma_{j_*}$, то $\bar{I}_{оп} = I_{оп} \cup i_0$, $\bar{J}_{оп} = J_{оп} \cup j_*$, $r = r + 1$, $u(i_0) = k\sigma_{j_*}$; $\bar{A}_{оп}^{-1}$ вычислим по формуле (20). Переход в п. 2.

19. Если $\Theta = \Theta_{j_0} < 1$, $\sigma = \sigma_{i_*}$, то $\bar{I}_{оп} = I_{оп} \setminus i_*$, $\bar{J}_{оп} = J_{оп} \setminus j_0$, $\Delta(j_0) = k\sigma_{i_*}$, $u(i_*) = u(r)$, $r = r - 1$; $\bar{A}_{оп}^{-1}$ вычислим по формуле (21). Переход в п. 2.

20. Если $\Theta = \Theta_{j_0} < 1$, $\sigma = \sigma_{j_*}$, то $\bar{I}_{оп} = I_{оп}$, $\bar{J}_{оп} = (J_{оп} \setminus j_0) \cup j_*$, $\Delta(j_0) = k\sigma_{j_*}$; $\bar{A}_{оп}^{-1}$ вычислим по формуле (22). Переход в п. 2.

21. Выдача результатов: it , $c'x$, ω , x , ω_* , ω^* .

Программная реализация изложенного алгоритма приведена в приложении (§ 1).

10. Первая группа численных экспериментов. Алгоритм реализован на языках ФОРТРАН и АЛГОЛ (ГДР и БЭСМ-АЛГОЛ). На ЭВМ БЭСМ-6 проведен ряд экспе-

риментов. Параметры a_{ij} , c_j , d_{*j} , d_j^* , b_{*i} , b_i^* , $i = \overline{1, m}$, $j = \overline{1, n}$, задачи (1) генерировались различными способами. В частности, с помощью библиотечных процедур UR и RANDOM генерировались случайные числа, равномерно распределенные на отрезках $[-1, 1]$ и $[0, 1]$, которые затем линейно отображались на заданные отрезки $[a, b]$. Границы отрезков выбирались так, чтобы множество планов не было пустым: $a_{ij} \in [-100, 100]$, $c_j \in [-100, 100]$, $d_{*j} \in [-100, 0]$, $d_j^* \in [0, 100]$, $b_{*i} \in [-100, 0]$, $b_i^* \in [0, 100]$. Оптимальный план в полученных задачах, как правило, имел опору небольших размеров, меньше $m/2$. Задачи, в которых число активных ограничений достигало значения $k \in [m/2, m]$, были получены с помощью генератора

```

'PROCEDURE' GENER1(A, B, N, J, K, CS);
'VALUE' A, B, K; 'REAL' A, B; 'INTEGER' N, J, K;
'ARRAY' CS; 'BEGIN' 'REAL' F; 'INTEGER' I; F:=3.14;
'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
'BEGIN' CS[I]:=(I * J * (K * K))/F - ENTIER (I * J * (K * K)/F);
CS[I]:=A + SIN (F * CS[I]) * (B - A)
'END'
'END' GENER1;

```

Назначение этой процедуры — построить n -вектор CS, компоненты которого выбираются из отрезка $[a, b]$, а их значения зависят от номера компоненты i , номера задачи j и от параметра k , который принимает разные значения при генерировании векторов задачи. Например, при $k=1$ генерируется первая строка матрицы A , при $k=2$ — вторая и т. д. При $k=m+1$ генерируется вектор c .

Параметры процедуры GENER1 имеют следующее назначение: A, B — границы отрезка, из которого выбираются элементы вектора; N — размерность вектора; J — номер задачи (по его значению различаются параметры разных задач); K — параметр, определяющий некоторый вектор в данной задаче; CS — имя массива, куда заносится сгенерированный вектор.

Чтобы получить N -вектор с первой задачи, достаточно выполнить оператор $GENER1(-100, 100, N, 1, M+1, C)$.

Эксперимент 1. Цель — ознакомиться с процессом решения задачи.

В табл. 1 приведены (с точностью до 0,1) значения параметров задачи 10×20 , сгенерированных процедурой GENER1 на указанных выше отрезках при $j=200$. Полагалось $k=1, m$ — для строк A ; $k=m+1, m+2, m+3$,

b_*	A									
-38.7	64.0	87.8	-48.6	29.1	99.2	-0.6	-14.5	97.1	40.6	-63.9
-3.1	29.1	97.1	72.5	-33.3	-29.2	74.6	96.4	26.8	-95.7	32.1
-8.3	40.7	100	43.7	-95.7	37.8	99.9	46.4	-91.7	34.9	99.7
-51.9	-33.3	25.8	70.4	95.6	98.3	78.4	38.0	-18.2	-83.7	-48.6
-84.1	80.0	56.9	-56.3	94.7	26.0	-14.8	100	-10.9	22.6	95.6
-26.9	-95.7	-91.7	-87.7	-83.7	-79.7	-75.7	-71.8	-67.8	-63.9	-60.0
-0.4	-88.0	-76.1	-64.2	-52.5	-40.9	-29.5	-18.4	-7.6	2.8	12.9
-15.8	95.6	-18.2	61.6	49.1	-0.6	90.5	-80.0	98.8	-36.8	72.5
-66.5	34.9	99.2	59.3	-63.9	6.2	92.7	78.4	-29.2	-25.8	80.0
-68.5	94.7	-10.9	54.1	59.3	-18.4	96.4	-91.7	92.7	-3.8	48.9
c	-96.0	-92.0	-88.0	-84.0	-80.0	-76.0	-72.1	-68.1	-64.2	-60.3
d_*	-13.8	-12.7	-97.8	-14.8	-11.7	-95.8	-15.9	-10.8	-93.8	-16.9
x^1	0	0	0	0	0	0	0	0	0	0
d^*	23.8	46.2	65.9	81.9	93.2	99.1	99.4	94.0	83.1	67.5

$m+4$, $m+5$ — для векторов c , b_* , d_* , d^* , b_* соответственно. Начальный план $x = \{x_i = 0, i = \overline{1, n}\}$. Изменение значений прямой и двойственной целевых функций на итерациях адаптивного алгоритма показано в табл. 2. Оптимальный план $x^* = \{-13.76, -12.69, -97.84, 27.35, -11.73, 47.16, -13.36, -10.81, 68.90, 54.97, 48.08, 0.04, -7.55, 12.28, 27.82, -10.84, -8.26, -87.87, -20.44, -7.49\}$. Опора достигла максимальных размеров (10×10) . Сравнивая x^* с d_* и d^* , замечаем, что десять его компонент принимают промежуточные значения ($d_{*j} < x_j < d_j^*$), остальные (неопорные) равны граничным значениям, т. е. x^* — невырожденный базисный план.

Эксперимент 2. Цель — сравнить адаптивный и симплексный методы по числу итераций и по времени счета.

Задачи вида (1) размеров 20×30 генерировались процедурой GENER1 при $j = \overline{100, 114}$, а задачи размеров 30×40 — с помощью библиотечной процедуры UR при $j = 103, 104, \dots, 112$. При этом в задачах 30×40 вектор b_* полагался равным $-\infty$, т. е. основные условия имели вид $Ax \leq b$. Каждая задача решалась двумя методами. Процесс решения начинался с плана $x = 0$.

При решении задач вида (1) симплекс-методом использовалась программа, реализующая метод с обратной матрицей применительно к задаче с двухсторонними ограничениями:

$$c'x \rightarrow \max, Ax - z = 0, d_* \leq x \leq d^*, b_* \leq z \leq b^*,$$

где z — m -вектор дополнительных переменных. Начальный базис состоял из единичных векторов.

Табл. 1

A										b*
81.9	72.5	-84.0	54.3	92.7	-33.3	16.5	100	12.9	-29.2	8.1
97.8	70.4	-37.1	-25.5	76.5	95.6	22.6	-91.7	35.1	98.3	16.1
49.1	-87.7	31.9	99.5	51.8	-83.7	28.8	99.2	54.4	-79.7	23.9
12.9	61.5	91.6	99.8	85.0	49.1	-3.8	-67.8	-64.2	-0.6	31.6
-52.2	54.1	81.9	-95.7	78.2	59.3	-60.3	93.7	29.1	-18.4	39.1
-56.0	-52.1	-48.3	-44.4	-40.6	-36.8	-33.0	-29.2	-25.5	-21.8	46.3
22.6	31.9	40.7	48.9	56.7	63.8	70.4	76.4	81.7	86.4	53.3
35.1	16.2	83.5	-60.3	100	-56.0	81.7	19.7	31.9	74.6	59.8
91.7	3.1	-60.3	61.5	98.8	32.1	-96.0	37.8	99.5	56.9	66.1
64.0	-25.8	97.8	-83.7	90.4	3.1	43.5	68.5	-33.3	98.8	71.8
-56.4	-52.4	-48.6	-44.7	-40.9	-37.1	-33.3	-29.5	-25.8	-22.1	c
-9.9	-91.8	-18.1	-9.1	-89.9	-19.2	-8.3	-87.9	-20.4	-7.5	d*
0	0	0	0	0	0	0	0	0	0	x ¹
48.1	25.9	2.2	21.8	44.4	64.4	80.8	92.5	98.9	99.6	d*

Адаптивный метод применялся непосредственно к задаче (1). Начальная опора имела размеры 0×0 .

Результаты эксперимента приведены в табл. 3. В задачах, сформированных с помощью GENER1 (опора достигает размеров $k \in [m/2, m]$), адаптивный алгоритм эффективнее симплексного в 2—3 раза; в задачах, сгенерированных с помощью UR (размер опоры не превосходит $m/2$), — в 10 раз.

Расчеты проводились по программе, написанной на языке АЛГОЛ-БЭСМ (ДОС ДИСПАК).

Эксперимент 3. Цель — изучить характеристики адаптивного алгоритма при решении задач, по структуре и размерам, близких к реальным.

Табл. 2

Номер итерации	c'x	$\psi(\lambda)$	Номер итерации	c'x	$\psi(\lambda)$
1	2.3213	44108.925	9	46.6159	59.0448
2	11.1057	42105.443	10	46.8633	58.6458
3	11.4691	41919.542	11	46.9173	51.6202
4	46.5518	31370.523	12	47.1981	50.9567
5	46.5564	63.7572	13	48.0823	50.5163
6	46.5597	63.2809	14	48.3769	50.4411
7	46.5830	61.9724	15	50.1071	50.1071
8	46.6024	59.2809			

Номер задачи	GENER1				UR			
	Симплекс		Адаптивный		Симплекс		Адаптивный	
	Число итераций	Время решения, с	Число итераций	Время решения, с	Число итераций	Время решения, с	Число итераций	Время решения, с
1	46	18	22	6	118	78	12	3
2	55	21	30	10	112	74	41	11
3	41	16	28	9	91	57	19	5
4	49	18	29	10	57	37	12	3
5	43	15	26	8	89	56	38	9
6	45	16	26	9	106	72	32	8
7	48	18	25	8	100	64	21	5
8	44	16	27	8	119	79	14	4
9	41	17	28	9	85	55	33	8
10	44	15	24	7	107	72	39	10

Задавалась разреженная матрица. Число ее ненулевых элементов составляло от 0,5 до 20 %. Такие матрицы с помощью процедуры GENER2 генерируются в файл, длина которого равна числу ненулевых элементов. Ненулевой элемент со своим номером строки хранится в одной ячейке. Элементы располагаются по столбцам, начала которых запоминаются в массиве целых чисел. Использовалась следующая процедура:

```

'PROCEDURE' GENER2 (K1, M, A, NCA);
'INTEGER' K1, M; 'INTEGER' 'ARRAY' NCA;
'ARRAY' A; 'BEGIN' 'INTEGER' I, I1, K, K2, J1; 'BOOLEAN' LOG;
'REAL' R2; NCA[1]:=1; K:=0; K2:=M/(K1+1); K2:=R2*K1;
K2:=M-K2;
'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
'BEGIN' LOG:= 'TRUE';
'FOR' I:=1 'STEP' 1 'UNTIL' K1 'DO'
'BEGIN' J1:=R2*I; J1:=J1+K; 'IF' LOG 'AND' J1>J 'THEN'
'BEGIN' 'IF' J1-R2-K<J 'THEN' J1:=J; LOG:= 'FALSE'
'END'; I1:=NCA[J]+I; A[I1]:=SIN(J**2+I)*200;
PLA031(A[I1], J1)
'END' I
NCA[J+1]:=NCA[J]+K1;

```

```

'IF' K<K2 'THEN' K:=K+1 'ELSE' K:=0
'END' J
'END' GENER2;

```

Эта процедура генерирует элементы столбцов со значениями из отрезка $[-200, 200]$.

Смысл параметров процедуры: $K1$ — число ненулевых элементов матрицы; M — число строк матрицы; $A[1:AK]$ — файл для хранения матрицы, AK — число ненулевых элементов; $NCA[1:N+1]$ указывает адреса в файле, с которых начинаются столбцы.

```

Векторы  $c, d_*, d^*, b_*, b^*$  генерировались процедурой
'PROCEDURE' VEK (K, A, B, C, N);
'VALUE' K, A, B; 'REAL' A, B; 'INTEGER' K, N;
'ARRAY' C;
'BEGIN' 'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
'BEGIN' C[J]:= (A+B)/2 + SIN (K ** 2 + J) * (B-A)/2;
PLA031 (C[J], J)
'END'
'END' VEK;

```

Эта процедура генерирует n -вектор с компонентами из $[a, b]$. Параметр K принимает значения 1, 2, 3, 4, 5 при генерировании векторов c, d_*, d^*, b_*, b^* , элементы которых выбираются из отрезков $[-100, 100]$, $[-100, 0]$, $[0, 100]$, $[-5000, 0]$, $[0, 5000]$, соответственно.

Для решения сгенерированных задач применялась программа на языке АЛГОЛ-ГДР (МС «ДУБНА»), реализующая адаптивный алгоритм с использованием внешней памяти (МБ и МД) для хранения файла A и обратной матрицы $A_{оп}^{-1}$. Результаты эксперимента приведены в табл. 4.

Табл. 4

$m \times n$	30×45					70×100			100×150			200×250
K1	3	4	5	6	7	1	2	3	2	3	4	1
Число итераций	24	28	29	33	32	43	45	72	76	78	84	111
Время решения, с	12	15	15	18	17	56	64	120	202	225	275	720

Значительный рост времени решения задачи с увеличением ее размеров объясняется тем, что программа, по

которой проводились расчеты, ориентирована на файл, хранящий ненулевые элементы матрицы только по столбцам. При вызове опорных элементов строки в этом случае необходимо списывать из внешней памяти все опорные столбцы, выбирая из них по одному элементу нужной строки. Эта операция выполнялась бы значительно быстрее, если бы матрица A хранилась и по столбцам и по строкам, что, разумеется, потребовало бы дополнительной памяти.

Обратная матрица в данной программе хранится полностью и занимает k^2 ($k \leq m$) слов.

Все операции с матрицами проводились посредством вызова их по частям в буферную память объемом в 1024 слова. С ростом размеров матриц увеличивается число обменов с внешней памятью, а следовательно, и время решения задачи.

11. Модификация двойственного опорного метода. В адаптивном методе, изложенном в пп. 1—8, при замене опоры (п. 7) использовался двойственный опорный метод [5,33] (для задачи (1)). В связи с этим возник вопрос о построении новых эффективных двойственных методов решения задачи (1).

Особенность предлагаемой модификации состоит в том, что движение во множестве двойственных планов, приводящее к уменьшению двойственной целевой функции (7), происходит не по элементарным направлениям (приведенным координатным осям), а по градиенту целевой функции (вдоль линейной комбинации элементарных направлений).

Вектор $\{y, s, t, v, w\}$, удовлетворяющий (7), назовем двойственным планом задачи (1). Вектор $\delta = A'y - c$, вычисленный по двойственному плану,— коплан задачи (7). Будем считать, что векторы s, t, v, w согласованы с y, δ :

$$\begin{aligned} s_i &= y_i, \quad t_i = 0, \quad \text{если } y_i \geq 0; \\ s_i &= 0, \quad t_i = -y_i, \quad \text{если } y_i < 0, \quad i \in I; \\ v_j &= \delta_j, \quad w_j = 0, \quad \text{если } \delta_j \geq 0; \\ v_j &= 0, \quad w_j = -\delta_j, \quad \text{если } \delta_j < 0, \quad j \in J. \end{aligned} \tag{23}$$

Совокупность $\{\delta, A_{оп}\}$ коплана и опоры задачи (7) называется опорным копланом, который считается невырожденным, если $y_i \neq 0, i \in I_{оп}$, и $\delta_j \neq 0, j \in J_{н}$.

По опорному коплану $\{\delta, A_{оп}\}$ построим соответствующий ему псевдоплан κ :

$$\begin{aligned}
\kappa_j &= d_j^* \text{ при } \delta_j < 0; \kappa_j = d_{*j} \text{ при } \delta_j \geq 0, j \in J_{\text{оп}}; \\
\omega_i &= b_i^* \text{ при } y_i > 0; \omega_i = b_{*i} \text{ при } y_i \leq 0, i \in I_{\text{оп}}; \\
\kappa(J_{\text{оп}}) &= A_{\text{оп}}^{-1} [\omega(I_{\text{оп}}) - A(I_{\text{оп}}, J_{\text{н}}) \kappa(J_{\text{н}})]; \\
\omega(I_{\text{н}}) &= A(I_{\text{н}}, J) \kappa(J).
\end{aligned} \tag{24}$$

Теорема 3 (критерий оптимальности). Соотношения

$$\begin{aligned}
d_{*j} &\leq \kappa_j \leq d_j^* \text{ при } \delta_j = 0; \kappa_j = d_{*j} \text{ при } \delta_j > 0; \\
\kappa_j &= d_j^* \text{ при } \delta_j < 0, j \in J_{\text{оп}}; b_{*i} \leq \omega_i \leq b_i^* \\
&\text{при } y_i = 0;
\end{aligned} \tag{25}$$

$$\omega_i = b_{*i} \text{ при } y_i < 0; \omega_i = b_i^* \text{ при } y_i > 0, i \in I_{\text{н}}$$

достаточны, а в случае невырожденности и необходимы для оптимальности опорного коплана.

Псевдоплан κ в этом случае является оптимальным планом прямой задачи (1).

Теорема 4 (достаточное условие субоптимальности). Если для псевдоплана κ и вектора ω , построенных по формулам (24), выполняются соотношения

$$\begin{aligned}
d_{*j} &\leq \kappa_j \leq d_j^*, j \in J_{\text{оп}}; b_{*i} \leq \omega_i \leq b_i^*, i \in I_{\text{н}}; \\
&\sum_{\delta_j > 0, j \in J_{\text{оп}}} (\kappa_j - d_{*j}) \delta_j + \sum_{\delta_j < 0, j \in J_{\text{оп}}} (\kappa_j - d_j^*) \delta_j + \\
&+ \sum_{y_i < 0, i \in I_{\text{н}}} (b_{*i} - \omega_i) y_i + \sum_{y_i > 0, i \in I_{\text{н}}} (b_i^* - \omega_i) y_i \leq \varepsilon,
\end{aligned}$$

то псевдоплан κ является ε -оптимальным планом задачи (1) (т. е. $c'x^0 - c'\kappa \leq \varepsilon$).

Опишем процедуру улучшения опорного коплана. Пусть $\{\delta, A_{\text{оп}}\}$ — опорный коплан, $\bar{\delta} = \delta + \Delta\delta$ — другой коплан, соответствующий двойственному плану $\{\bar{y}, \bar{s}, \bar{t}, \bar{v}, \bar{w}\}$, $\bar{y} = y + \Delta y$, $\bar{s} = s + \Delta s$, $\bar{t} = t + \Delta t$, $\bar{v} = v + \Delta v$, $\bar{w} = w + \Delta w$, для которого также выполняется условие согласования (23).

Используя (24) и условие согласования (23), получим формулу приращения двойственной целевой функции задачи (7):

$$\begin{aligned}
b^{*'} \Delta s - b_{*}' \Delta t - d_{*}' \Delta v + d^{*'} \Delta w &= (A\kappa - \omega)' \Delta y + b^{*'} \Delta s - \\
- b_{*}' \Delta t - d_{*}' \Delta v + d^{*'} \Delta w &= \Delta y' A\kappa - d_{*}' \Delta v + d^{*'} \Delta w - \Delta y' \omega +
\end{aligned}$$

$$\begin{aligned}
& + b^* \Delta s - b_* \Delta t = \sum_{\substack{\delta_j > 0, \bar{\delta}_j < 0, \\ j \in J_{\text{оп}}}} [(\kappa_j - d_j^*) \bar{\delta}_j - (\kappa_j - d_{*j}) \delta_j] + \\
& + \sum_{\substack{\delta_j < 0, \bar{\delta}_j > 0, \\ j \in J_{\text{оп}}}} [(\kappa_j - d_{*j}) \bar{\delta}_j - (\kappa_j - d_j^*) \delta_j] + \\
& + \sum_{\substack{\delta_j > 0, \bar{\delta}_j > 0, \\ j \in J_{\text{оп}}}} (\kappa_j - d_{*j}) \Delta \delta_j + \sum_{\substack{\delta_j < 0, \bar{\delta}_j < 0, \\ j \in J_{\text{оп}}}} (\kappa_j - d_j^*) \Delta \delta_j + \\
& + \sum_{\substack{\delta_j = 0, \bar{\delta}_j > 0, \\ j \in J_{\text{н}}}} (\kappa_j - d_{*j}) \bar{\delta}_j + \sum_{\substack{\delta_j = 0, \bar{\delta}_j < 0, \\ j \in J_{\text{н}}}} (\kappa_j - d_j^*) \bar{\delta}_j + \\
& + \sum_{\substack{\delta_j < 0, \bar{\delta}_j > 0, \\ j \in J_{\text{н}}}} (d_j^* - d_{*j}) \bar{\delta}_j + \sum_{\substack{\delta_j > 0, \bar{\delta}_j < 0, \\ j \in J_{\text{н}}}} (d_{*j} - d_j^*) \bar{\delta}_j + \quad (26) \\
& + \sum_{\substack{y_i = 0, \bar{y}_i > 0, \\ i \in I_{\text{оп}}}} (b_i^* - \omega_i) \bar{y}_i + \sum_{\substack{y_i = 0, \bar{y}_i < 0, \\ i \in I_{\text{оп}}}} (b_{*i} - \omega_i) \bar{y}_i + \\
& + \sum_{\substack{y_i < 0, \bar{y}_i > 0, \\ i \in I_{\text{оп}}}} (b_i^* - b_{*i}) \bar{y}_i + \sum_{\substack{y_i > 0, \bar{y}_i < 0, \\ i \in I_{\text{оп}}}} (b_{*i} - b_i^*) \bar{y}_i + \\
& + \sum_{\substack{y_i > 0, \bar{y}_i > 0, \\ i \in I_{\text{н}}}} (b_i^* - \omega_i) \Delta y_i + \sum_{\substack{y_i < 0, \bar{y}_i < 0, \\ i \in I_{\text{н}}}} (b_{*i} - \omega_i) \Delta y_i + \\
& + \sum_{\substack{y_i > 0, \bar{y}_i < 0, \\ i \in I_{\text{н}}}} [(b_{*i} - \omega_i) \bar{y}_i - (b_i^* - \omega_i) y_i] + \\
& + \sum_{\substack{y_i < 0, \bar{y}_i > 0, \\ i \in I_{\text{н}}}} [(b_i^* - \omega_i) \bar{y}_i - (b_{*i} - \omega_i) y_i].
\end{aligned}$$

Если соотношения (25) не выполняются, то приращение (26) можно сделать отрицательным (в невырожденном случае).

В предлагаемом методе на каждой итерации будем варьировать k , $k \geq 1$, переменных, включающих опорные компоненты коплана δ и неопорные компоненты вектора y .

Построим подмножества $J_{\text{оп}}^0 \subset J_{\text{оп}}$ и $I_{\text{н}}^0 \subset I_{\text{н}}$, $|J_{\text{оп}}^0| + |I_{\text{н}}^0| = k$, следующим образом. Выберем опорные переменные коплана δ и неопорные переменные вектора y , на которых не выполняется критерий оптимальности (25), и для каждой из них подсчитаем числа $\gamma_j^{\text{оп}}$, $j \in J_{\text{оп}}$; $\gamma_i^{\text{н}}$, $i \in I_{\text{н}}$:

$$\gamma_j^{\text{оп}} = \begin{cases} \kappa_j - d_{*j}, & \text{если } \delta_j > 0 \text{ или } \delta_j = 0 \text{ и } \kappa_j < d_{*j}, \\ \kappa_j - d_j^*, & \text{если } \delta_j < 0 \text{ или } \delta_j = 0 \text{ и } \kappa_j > d_j^*, \end{cases}$$

$$\gamma_i^{\text{н}} = \begin{cases} \omega_i - b_{*i}, & \text{если } y_i < 0 \text{ или } y_i = 0 \text{ и } \omega_i < b_{*i}, \\ \omega_i - b_i^*, & \text{если } y_i > 0 \text{ или } y_i = 0 \text{ и } \omega_i > b_i^*. \end{cases}$$

Положим

$$J_{\text{оп}}^0 = \{j : j \in J_{\text{оп}}, |\gamma_j^{\text{оп}}| \geq \mu\}, \quad I_{\text{н}}^0 = \{i : i \in I_{\text{н}}, |\gamma_i^{\text{н}}| \geq \mu\}, \quad (27)$$

где $\mu, \mu \geq 0$, — параметр метода.

При построении множеств $J_{\text{оп}}^0$ и $I_{\text{н}}^0$ можно использовать и другие принципы, например, в число варьируемых переменных включать определенный процент или конкретное число переменных, на которых не выполняется критерий оптимальности (25).

Подходящее направление для коплана δ и вектора y построим по следующему правилу:

$$\begin{aligned} t_j^1 &= 0, \quad j \in J_{\text{оп}} \setminus J_{\text{оп}}^0; \quad t_j^1 = -\gamma_j^{\text{оп}}, \quad j \in J_{\text{оп}}^0; \\ t_i^2 &= 0, \quad i \in I_{\text{н}} \setminus I_{\text{н}}^0; \quad t_i^2 = \gamma_i^{\text{н}}, \quad i \in I_{\text{н}}^0; \\ t^{2'}(I_{\text{оп}}) &= [t^{1'}(J_{\text{оп}}) - t^{2'}(I_{\text{н}}) A(I_{\text{н}}, J_{\text{оп}})] A_{\text{оп}}^{-1}; \\ t^{1'}(J_{\text{н}}) &= t^{2'}(I) A(I, J_{\text{н}}). \end{aligned} \quad (28)$$

Начальная скорость убывания двойственной целевой функции задачи (7) вдоль подходящего направления (28)

$$\alpha_0 = - \sum_{j \in J_{\text{оп}}^0} (\gamma_j^{\text{оп}})^2 - \sum_{i \in I_{\text{н}}^0} (\gamma_i^{\text{н}})^2.$$

Максимальный шаг σ вдоль подходящего направления (28) находим по правилу длинного двойственного шага (см. п. 7). Новые коплан $\bar{\delta}$ и вектор \bar{y} равны $\bar{\delta} = \delta + \sigma t^1$, $\bar{y} = y + \sigma t^2$.

Опишем правила построения новой опоры $\bar{A}_{\text{оп}} = A(\bar{I}_{\text{оп}}, \bar{J}_{\text{оп}})$. Вычислим $\gamma^0 = \max \{ \max_{j \in J_{\text{оп}}^0} |\gamma_j^{\text{оп}}|, \max_{i \in I_{\text{н}}^0} |\gamma_i^{\text{н}}| \}$.

Пусть $\gamma^0 = \gamma_{j_0}^{\text{оп}}$, $j_0 \in J_0^{\text{оп}}$. Если $\sigma = \sigma_{j_*}$, $j_* \in J_{\text{н}}$, то $\bar{I}_{\text{оп}} = I_{\text{оп}}$, $\bar{J}_{\text{оп}} = (J_{\text{оп}} \setminus j_0) \cup j_*$. Если $\sigma = \sigma_{i_*}$, $i_* \in I_{\text{оп}}$, то $\bar{I}_{\text{оп}} = I_{\text{оп}} \setminus i_*$, $\bar{J}_{\text{оп}} = J_{\text{оп}} \setminus j_0$. Пусть $\gamma^0 = \gamma_{i_0}^{\text{н}}$, $i_0 \in I_{\text{н}}$. Если $\sigma = \sigma_{j_*}$, $j_* \in J_{\text{н}}$, то $\bar{I}_{\text{оп}} = I_{\text{оп}} \cup i_0$, $\bar{J}_{\text{оп}} = J_{\text{оп}} \cup j_*$. Если $\sigma = \sigma_{i_*}$, $i_* \in I_{\text{оп}}$, то $\bar{I}_{\text{оп}} = (I_{\text{оп}} \setminus i_*) \cup i_0$, $\bar{J}_{\text{оп}} = J_{\text{оп}}$. При $\sigma = \sigma_{j_*}$, $j_* \in J_{\text{оп}}$, или $\sigma = \sigma_{i_*}$, $i_* \in I_{\text{н}}$, опора сохраняется.

12. Блочная замена опоры. Решение задачи (1) адаптивным методом с элементарной заменой опоры (АЭ) описано в пп. 1—8. В адаптивном методе с блочной заменой опоры (АБ), как и в АЭ, на каждой итерации осуществляется переход от опорного плана $\{x, A_{\text{оп}}\}$ к новому опорному плану $\{\bar{x}, \bar{A}_{\text{оп}}\}$, на котором $c'x \leq c'\bar{x}$, $\bar{x} = x + \Theta_0 l$, где l — подходящее направление подъема, Θ_0 — максимально допустимый шаг вдоль l . В отличие от АЭ в АБ для получения $\bar{A}_{\text{оп}}$ решается вспомогательная задача.

Выберем некоторые подмножества I_0 , J_0 ($I_0 \subset I_{\text{н}}$, $J_0 \subset J_{\text{оп}}$). Среди всевозможных опор $\bar{A}_{\text{оп}} = A(I_{\text{оп}}, \bar{J}_{\text{оп}})$, таких, что $I_{\text{оп}} \cap I_0 = \emptyset$, $J_{\text{оп}} \supset J_0$, найдем ту, при которой оценка субоптимальности $\bar{\beta}$ опорного плана $\{\bar{x}, \bar{A}_{\text{оп}}\}$ минимальна. Требования $J_0 \subset \bar{J}_{\text{оп}}$ и $I_0 \cap \bar{I}_{\text{оп}} = \emptyset$ эквивалентны условиям

$$w(J_0) = v(J_0) = 0, \quad y(I_0) = s(I_0) = t(I_0) = 0. \quad (29)$$

Таким образом, поиск опоры $\bar{A}_{\text{оп}}$ сводится к поиску оптимального базисного плана вспомогательной задачи, полученной из двойственной задачи (7) с учетом дополнительных условий (29):

$$\begin{aligned} f(r) = b^{*'}(\tilde{I})s(\tilde{I}) - b'_*(\tilde{I})t(\tilde{I}) - d'_*(\tilde{J})v(\tilde{J}) + \\ + d^{*'}(\tilde{J})w(\tilde{J}) \rightarrow \min, \end{aligned} \quad (30)$$

$$\begin{aligned} A'(\tilde{I}, J_0)y(\tilde{I}) = c(J_0), \quad A'(\tilde{I}, \tilde{J})y(\tilde{I}) + w(\tilde{J}) - \\ - v(\tilde{J}) = c(\tilde{J}), \end{aligned}$$

$$y(\tilde{I}) = s(\tilde{I}) - t(\tilde{I}), \quad s(\tilde{I}) \geq 0, \quad t(\tilde{I}) \geq 0, \quad v(\tilde{J}) \geq 0, \quad w(\tilde{J}) \geq 0,$$

где $\tilde{I} = I \setminus I_0$, $\tilde{J} = J \setminus J_0$, $r = \{y(\tilde{I}), s(\tilde{I}), t(\tilde{I}), v(\tilde{J}), w(\tilde{J})\}$.

Для решения задачи (30) в качестве начального базисного плана можно взять план r , составленный из компонент двойственного плана (8), сопровождающего опору

$A_{\text{оп}}$. Базис оптимального базисного плана $r^* = \{y^*(\tilde{I}), s^*(\tilde{I}), t^*(\tilde{I}), v^*(\tilde{J}), w^*(\tilde{J})\}$ задачи (30) и есть искомая опора $\bar{A}_{\text{оп}}$.

Для новой опоры $\bar{A}_{\text{оп}}$ оценка $\bar{\beta}_{\text{оп}} = \beta_{\text{оп}} - (f(r) - f(r^*))$. Следовательно, опора $\bar{A}_{\text{оп}}$ не хуже опоры $A_{\text{оп}}$, поскольку $f(r) \geq f(r^*)$. Оценка субоптимальности нового опорного плана $\{\bar{x}, \bar{A}_{\text{оп}}\}$ равна $\bar{\beta} = (1 - \Theta_0)\beta - (f(r) - f(r^*))$.

Естественно ожидать, что величина $f(r) - f(r^*)$ будет большей, если в множество $J_{\text{оп}} \setminus J_0$ войдут индексы j , для которых числа $x_j + l_j$ в наибольшей степени выходят за границы отрезка $[d_{*j}, d_j^*]$ (аналогично для $I_{\text{н}} \setminus I_0$).

Рассмотрим задачу, двойственную к (30):

$$c'(J)x(J) \rightarrow \max, \quad (31)$$

$$b_*(\tilde{I}) \leq A(\tilde{I}, J)x(J) \leq b^*(\tilde{I}); \quad d_*(\tilde{J}) \leq x(\tilde{J}) \leq d^*(\tilde{J}),$$

которую можно решать вместо (30), начиная с опорного плана $\{\bar{x}, A_{\text{оп}}\}$. Если на оптимальном опорном плане $\{x^*, \bar{A}_{\text{оп}}\}$ задачи (31) (на оптимальном псевдоплане задачи (30)) выполняются условия

$$d_{*j} \leq x_j^* \leq d_j^*, \quad j \in J_0; \quad b_{*i} \leq a_i' x_i^* \leq b_i^*, \quad i \in I_0, \quad (32)$$

то $\bar{A}_{\text{оп}}$ — оптимальная опора задачи (1) (т. е. величина $f(r) - f(r^*)$ максимально возможная) и x^* — ее оптимальный план. Следовательно, в множество J_0 имеет смысл включать те опорные компоненты плана \bar{x} , которые далеки от границы (аналогично для I_0):

$$\begin{aligned} J_0 &= \{j : j \in J_{\text{оп}}; \quad d_{*j} + \xi < \bar{x}_j < d_j^* - \xi\}, \\ I_0 &= \{i : i \in I_{\text{н}}; \quad b_{*i} + \xi < a_i' \bar{x} < b_i^* - \xi\}, \end{aligned} \quad (33)$$

или те, на которых при движении вдоль l прямые ограничения нарушаются в последнюю очередь (аналогично для I_0):

$$\begin{aligned} J_0 &= \{j : j \in J_{\text{оп}}; \quad \Theta_j > \Theta_0 + \xi\}, \\ I_0 &= \{i : i \in I_{\text{н}}; \quad \Theta_i > \Theta_0 + \xi\}, \end{aligned} \quad (34)$$

где ξ — параметр метода.

С другой стороны, от выбора опоры $\bar{A}_{\text{оп}}$ существенно зависит выбор подходящего направления на следующей итерации (для опорного плана $\{\bar{x}, \bar{A}_{\text{оп}}\}$). Пусть условия (32) не выполняются. Возьмем в качестве подходящего

направления вектор $x^* - \bar{x}$. Если коплан $\bar{\Delta}$, сопровождающий опору $\bar{A}_{\text{оп}}$, невырожденный, то этот вектор совпадет с вектором \bar{l} , построенным по формулам (10) для коплана $\bar{\Delta}$. При вырожденном коплане $\bar{\Delta}$ векторы $x^* - \bar{x}$ и \bar{l} могут не совпадать, однако в любом случае оба они являются оптимальными планами задачи максимизации функции (2) при ограничениях (4), построенной по опоре $\bar{A}_{\text{оп}}$. Из (31) видно, что максимально допустимый шаг $\bar{\Theta}_0$ вдоль направления $x^* - \bar{x}$ достигается либо на прямых ограничениях $j \in J_0$, либо на основных ограничениях $i \in I_0$, причем $\bar{\Theta}_0 > 0$. Это говорит о том, что в J_0 и I_0 следует включать далекие (на плане \bar{x}) от границ переменные и ограничения. При больших ξ в (33) множество $J_0 = I_0 = \emptyset$ и задача (30) ((31)) совпадают с исходной. При $\xi = 0$ и условии, что шаг $\bar{\Theta}_0$ на предыдущей итерации определился однозначно, имеем случай элементарной замены (см. п. 7). В АБ переход от $\{x, A_{\text{оп}}\}$ к $\{\bar{x}, \bar{A}_{\text{оп}}\}$ называется большой итерацией метода. Она состоит из малых итераций решения вспомогательной задачи (30) ((31)).

13. Вторая группа численных экспериментов. Рассматривалась интервальная задача линейного программирования (1). Сравнивались четыре метода АЭ, АБ, двойственный опорный (ДО) и модифицированный двойственный опорный (МДО) (см. п. 11) [33]. На ЭВМ ЕС-1030 проведен ряд экспериментов. Перечисленные методы реализованы в виде подпрограмм на языке ФОРТРАН-IV. Этими методами решались серии задач вида (1) при разных значениях m, n . Каждая серия состояла из пяти задач. Все задачи были поделены на два типа: $m \leq n$ (I), $m > n$ (II). Значения c_j, a_{ij} элементов c, A задач генерировались датчиком случайных чисел с равномерным распределением на отрезках $-10 \leq c_j \leq 10$, $-10 \leq a_{ij} \leq 10, i = \overline{1, m}, j = \overline{1, n}$. Компоненты $x_j^h, j = \overline{1, n}$, вектора x^h генерировались на отрезке $[-50, 50]$. По вектору x^h и матрице A вычислялись $a_i' x^h, i = \overline{1, m}$. Далее генерировались случайные числа $\gamma_i^1 \in [0, s_1]$ и $\gamma_i^2 \in [0, s_2]$. После этого полагалось $b_{*i} = a_i' x^h - \gamma_i^1, b_i^* = a_i' x^h + \gamma_i^2, i = \overline{1, m}$. Числа s_1, s_2 в серии выбирались разными и принимали одно из следующих значений: 10, 120, 200, 350. Аналогичным образом поступали при формировании прямых ограничений: $d_{*j} = x_j^h - \gamma_j^1, d_j^* = x_j^h + \gamma_j^2, j = \overline{1, n}$ (со значениями 30, 120, 200,

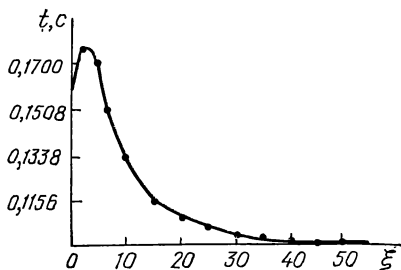
350 для s_1 и s_2). Для генерирования случайных чисел использовалась библиотечная подпрограмма RANDU. Сгенерированная описанным способом задача имела непустое множество планов, ее оптимальному плану, как правило, соответствовали опоры размеров, равных или близких k , где $k = \min(m, n)$.

Эксперимент 1. Цель — сравнить эффективность АЭ и АБ *) (по быстродействию), а также установить влияние на эффективность АБ величины параметра ξ , $\xi \geq 0$, от которого зависят размеры вспомогательной задачи (30) ((31)).

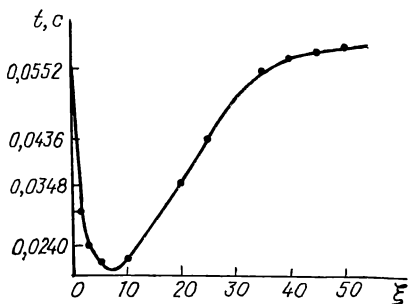
При формировании вспомогательной задачи для построения подмножеств I_0 и J_0 использовалось правило (33). Задача решалась двойственным опорным методом с оптимальной заменой опоры (с длинным шагом). В качестве начального плана выбирался сгенерированный вектор x^H . Начальная опора в обоих методах имела размеры 0×0 . Во всех сериях каждая задача методом АБ решалась 14 раз при следующих значениях параметра ξ : 0, 1,

3, 5, 7, 10, 15, 20, 25, 30, 35, 40, 45, 50 % от величины $d = \max \{ \max_{i \in I} (b_i^* - b_{*i}), \max_{j \in J} (d_j^* - d_{*j}) \}$.

Качественное исследование АБ показало, что для задач типа I АБ работает лучше при $\xi \geq 50\%$ d , хуже при $\xi \approx 1-3\%$ d . Для задач типа II этот метод лучше работает при $\xi \approx 3-10\%$ d , хуже при $\xi \geq 50\%$ d . На рис. 1 показана зависимость от ξ времени (t) решения методом



Р и с. 1



Р и с. 2

*) Результаты численного эксперимента по сравнению АЭ и симплекс-метода приведены в п. 10 и [8].

АБ задач типа I (в качестве примера взята серия задач 30×40), а на рис. 2 — задач типа II (серия задач 200×10).

В табл. 5 приведены результаты этого эксперимента: среднее время (стандартизированное по Колвилу) решения задач по сериям указанными методами и оптимальное значение параметра $\xi_{\text{опт}}$ (параметра АБ).

Табл. 5

$m \times n$	АЭ	АБ		$m \times n$	АЭ	АБ	
	Время	Время	$\xi_{\text{опт}}$		Время	Время	$\xi_{\text{опт}}$
30×40	0.1604	0.0957	50.0	400×20	0.3144	0.1112	6,3
10×90	0.0829	0.0499	50.0	600×20	0.5142	0.1309	3,0
90×10	0.0223	0.0119	4.6	270×30	0.6356	0.3055	7,3
200×10	0.0525	0.0177	6.3	600×30	1.4124	0.3806	3,7
300×10	0.0771	0.0191	3.7	900×30	1.7807	0.4542	4,0
150×20	0.1691	0.0825	10.0				

Эксперимент 2. Цель — сравнить эффективность ДО и МДО (по числу итераций и по времени счета), а также выявить эффективность МДО при различных значениях параметра метода.

Процесс решения задач ДО и МДО начинался с ко-плана $\delta = -c$ и вектора $y = 0$. Начальная опора имела размеры 0×0 . В выполненной программной реализации МДО для построения подмножеств $J_{\text{оп}}^0$ и $I_{\text{н}}^0$ использовалось правило (27). Каждая задача из серии решалась 12 раз при значениях параметра μ : 100, 97, 94, 90, 87, 85, 80, 75, 60, 50, 20, 0 % от величины γ^0 (см. п. 11). Наибольшая эффективность МДО получена при $\mu \approx 70$ —100 % γ^0 . Проведено исследование МДО для случая, когда компоненты подходящего направления на подмножествах $J_{\text{оп}}^0$ и $I_{\text{н}}^0$, в отличие от (28), определялись следующим образом:

$$t_j^1 = -\text{sign}(\gamma_j^{\text{оп}}), j \in J_{\text{оп}}^0; \quad t_i^2 = \text{sign}(\gamma_i^{\text{н}}), i \in I_{\text{н}}^0. \quad (35)$$

Время решения задачи при одинаковых значениях параметра μ и оптимальные значения μ для модификаций (28) и (35) не совпадали, но время при оптимальных значениях μ в том и другом случае было почти одинаково.

Анализ численного эксперимента показал, что МДО работает лучше ДО для задач типа I, а для задач типа II лучше работает ДО (табл. 6).

З а м е ч а н и е. Из результатов численных экспериментов, приведенных в данном пункте, видно, что для задач типа I самым эффективным из четырех рассмотренных методов оказался МДО, а для задач типа II — АБ.

Табл. 6

$m \times n$	ДО		МДО		
	Число итераций	Время	Число итераций	Время	$\mu_{\text{опт}}$
20×20	20.75	0.0210	20.25	0.0184	97.00
30×40	35.00	0.0941	32.75	0.0875	89.50
40×40	42.25	0.1678	39.50	0.1386	77.50
10×90	18.75	0.0471	13.25	0.0317	76.00
10×300	22.00	0.2919	15.67	0.2087	20.00
20×150	39.33	0.3239	33.00	0.2292	74.67
20×400	48.67	1.3926	41.67	1.0516	75.00
90×10	23.90	0.0231	25.00	0.0445	99.25
200×10	32.33	0.0607	30.00	0.1091	97.00
150×20	50.00	0.1603	50.00	0.2583	100.00
400×20	68.33	0.4631	58.00	0.7570	100.00

14. Третья группа численных экспериментов. С помощью датчика псевдослучайных чисел было сформировано 14 серий канонических задач для различных m и n [24], [29]. Каждая серия состояла из 10 задач. В основе генератора лежал следующий датчик:

```

FUNCTION Y(P)
P = SIN (137.* P + 2.)
K = P * 3 * 100
Y=K/10
RETURN
END

```

При генерировании каждой задачи вначале псевдослучайными числами заполнялись вектор c и матрица A . Для получения d_* , d^* и начального плана x^H генерировались тройки псевдослучайных чисел α_{j_1} , α_{j_2} , α_{j_3} , после чего по-

Табл. 7

Номер серии	m	n	Среднее число смен опоры в серии					
			1	2	3	4	5	6
1	5	10	2.6	3.4	4.2	8.3	3.9	2.9
2	5	30	8.1	18.1	22.5	26.5	20.9	7.1
3	5	50	11.2	34.8	43.3	46.8	38.6	10.7
4	5	90	13.8	63.1	77.0	87.6	70.8	11.5
5	7	10	2.3	2.9	2.8	10.5	2.8	3.0
6	7	30	10.5	18.9	25.9	32.9	25.3	9.8
7	7	50	13.7	37.9	45.7	55.5	45.6	13.6
8	7	90	18.4	67.2	88.6	109.9	80.3	16.1
9	8	30	11.3	22.9	21.1	35.4	21.8	9.5
10	8	50	14.4	39.6	44.3	60.9	43.0	14.9
11	8	90	19.7	66.2	92.1	105.0	84.4	16.7
12	10	30	12.2	21.9	22.6	33.7	24.8	10.5
13	10	50	18.8	47.5	47.8	59.7	42.9	16.4
14	10	90	27.9	73.8	92.4	112.6	91.6	23.0

Табл. 8

Номер задачи	Число смен опоры при решении задач 10×90						Номер задачи	Число смен опоры при решении задач 10×90					
	1	2	3	4	5	6		1	2	3	4	5	6
1	32	72	88	111	103	17	6	24	44	82	115	85	22
2	37	92	98	130	109	29	7	25	84	92	112	94	17
3	25	58	106	113	85	27	8	26	82	81	102	81	20
4	19	70	88	97	74	28	9	27	87	93	123	81	25
5	36	73	99	106	97	20	10	28	76	97	117	107	25

лагалось $d_{*j} = \min \{\alpha_{j_1}, \alpha_{j_2}, \alpha_{j_3}\}$, $d_j^* = \max \{\alpha_{j_1}, \alpha_{j_2}, \alpha_{j_3}\}$, $x_j^H = \alpha_{j_1} + \alpha_{j_2} + \alpha_{j_3} - d_{*j} - d_j^*$. Затем вычислялся вектор $b = Ax^H$. Начальная опора $\{n - m + 1, \dots, n\}$.

Методы сравнивались по числу смен опоры — одной из наиболее трудоемких операций в каждом точном алгоритме.

Сравнивались следующие методы: 1) прямой метод с адаптивной нормировкой и оптимальной заменой элементов опоры (длинный шаг); 2) прямой метод с адаптивной нормировкой и упрощенной заменой элементов опоры (короткий шаг); 3) прямой опорный метод с симплексной нормировкой [5]; 4) двухфазный симплекс-метод; 5) метод Чарнса [40]. (В отличие от метода 3 при выборе ведущего столбца предпочтение отдается столбцам, соответствующим не критическим компонентам текущего плана); 6) модифицированный прямой метод с адаптивной нормировкой и оптимальной заменой элементов опоры. (В отличие от метода 1 ведущая строка выбирается из условия максимума начальной скорости убывания двойственной целевой функции.)

Перечисленные методы реализованы в виде подпрограмм на языке FORTRAN-IV ЕС ЭВМ. Результаты экспериментов приведены в табл. 7, 8.

§ 2. Вариант метода с пересчетом оценок

Для задачи

$$c'x \rightarrow \max, \quad b_* \leq Ax \leq b^*, \quad d_* \leq x \leq d^*$$

рассматривается новая реализация адаптивного метода, в которой, как и в методе обратной матрицы (в одной из классических реализаций симплекс-метода), компоненты n -вектора оценок не запоминаются, а в случае необходимости заново вычисляются по формуле

$$\Delta_j = u' (I_{оп}) A (I_{оп}, j) - c_j, \quad j \in J_n; \quad u' (I_{оп}) = c' (J_{оп}) A_{оп}^{-1}.$$

Такая реализация позволяет при одном и том же объеме оперативной памяти решать задачи со значительно большим числом переменных, чем при использовании алгоритма § 1. Однако при этом вместо процедуры оптимальной замены опоры (основанной на вычислении «длинного» двойственного шага) используется процедура простой замены опоры (основанной на вычислении «короткого» двойственного шага). В излагаемой ниже реализации компоненты коплана на каждой итерации вычисляются дважды: для построения подходящего прямого направления и для вычисления двойственного шага. В силу указанных причин количество итераций (и время счета) в алгоритме § 1 меньше, чем в алгоритме данного параграфа. Поэтому новую реализацию целесообразно

применять лишь в случаях, когда ощущается недостаток оперативной памяти ЭВМ, а задача содержит много переменных.

1. Алгоритм. 1. Зададим план x и положим $I_{\text{оп}} = \emptyset$, $I_{\text{н}} = \{1, 2, \dots, m\}$, $J_{\text{оп}} = \emptyset$, $J_{\text{н}} = \{1, 2, \dots, n\}$, размерность опоры $r=0$, число итераций $it=0$.

2. Вычислим вектор потенциалов $u' (I_{\text{оп}}) = c' (I_{\text{оп}}) A_{\text{оп}}^{-1}$.

3. Положим $v(I) = \omega(I) = 0$, $\beta = 0$.

4. Организуем цикл по $j \in J = \{1, 2, \dots, n\}$, в котором выполним следующие операции: а) $\omega(I) = \omega(I) + A(I, j)x_j$; б) $\Delta = u'(I_{\text{оп}})A(I_{\text{оп}}, j) - c_j$; в) $l=0$, если $\Delta=0$; $l=d_{*j} - x_j$, если $\Delta > 0$; $l=d_j^* - x_j$, если $\Delta < 0$; г) $v(I) = v(I) + A(I, j)l$; д) $\beta = \beta + \Delta l$. (Здесь операции б) — д) выполняются только при $j \in J_{\text{н}}$, причем величины Δ и l скалярные.)

5. Вычислим вектор $w(I_{\text{оп}})$: $w_i = b_i^* - \omega_i$, если $u_i > 0$; $w_i = b_{*i} - \omega_i$, если $u_i < 0$; $w_i = 0$, если $u_i = 0$, $i \in I_{\text{оп}}$.

6. Найдем полное значение оценки субоптимальности $\beta = \beta + u'(I_{\text{оп}})w(I_{\text{оп}})$ и проверим неравенство $\beta \leq \varepsilon$. При выполнении неравенства перейдем на конец решения задачи, т. е. к п. 25, в противном случае продолжим ее решение.

7. Положим $it = it + 1$ и вычислим $l(J_{\text{оп}}) = A_{\text{оп}}^{-1} [w(I_{\text{оп}}) - v(I_{\text{оп}})]$.

8. Вычислим $v(I_{\text{н}}) = A(I_{\text{н}}, J)l(J)$. Для этого к найденному в п. 4 вектору $v(I) = A(I, J_{\text{н}})l(J_{\text{н}})$ добавим $A(I, J_{\text{оп}})l(J_{\text{оп}})$.

9. Найдем максимально допустимый шаг вдоль $l(J_{\text{оп}})$: положим $\Theta_{j_0} = 1$; в цикле по $j \in \{j \in J_{\text{оп}}: l_j \neq 0\}$ выполним операции: а) $\Theta = (d_{*j} - x_j)/l_j$, если $l_j < 0$; $\Theta = (d_j^* - x_j)/l_j$, если $l_j > 0$; б) если $\Theta < \Theta_{j_0}$, то положим $\Theta_{j_0} = \Theta$, $j_0 = j$.

10. Вычислим Θ_{i_0} — шаг по неопорным основным ограничениям. Положим $\Theta_{i_0} = \Theta_{j_0}$, $i_0 = 0$. Цикл по $i \in \{i \in I_{\text{н}}: v_i \neq 0\}$: а) $\Theta = (b_i^* - \omega_i)/v_i$, если $v_i > 0$; $\Theta = (b_{*i} - \omega_i)/v_i$, если $v_i < 0$; б) если $\Theta < \Theta_{i_0}$, то положим $\Theta_{i_0} = \Theta$, $i_0 = i$.

11. Если $i_0 = 0$, то $\Theta = \Theta_{j_0}$, в противном случае $\Theta = \Theta_{i_0}$. При $\Theta = 1$ положим $x = x + l$ и перейдем в п. 25.

12. В случае $i_0 = 0$ положим

$$k = \begin{cases} 1, & \text{если } l_{j_0} < 0, \\ -1, & \text{если } l_{j_0} > 0, \end{cases}$$

в противном случае

$$k = \begin{cases} 1, & \text{если } v_{i_0} > 0, \\ -1, & \text{если } v_{i_0} < 0. \end{cases}$$

13. Если $i_0 > 0$, то перейдем в п. 19.

14. Найдем σ_{j_*} . Положив $\sigma_{j_*} = \infty$, $j_* = 0$, для $j \in J_{\text{н}}$ выполним следующие операции: а) $\xi = k A_{\text{оп}}^{-1}(j_0, I_{\text{оп}}) A(I_{\text{оп}}, j)$; б) $\Delta = u'(I_{\text{оп}}) A(I_{\text{оп}}, j) - c_j$; в) если $\Delta = 0$, $x_j \neq d_{*j}$, $\xi > 0$ или $\Delta = 0$, $x_j \neq d_j^*$, $\xi < 0$, то положим $\sigma_{j_*} = 0$, $\alpha = k\xi$, $j_* = j$; г) при $\Delta \xi < 0$ вычислим $\sigma = -\Delta/\xi$, если $\sigma < \sigma_{j_*}$, то положим $\sigma_{j_*} = \sigma$, $j_* = j$, $\alpha = k\xi$; д) если $\Delta < 0$, то $l = d_{*j} - x_j$; если $\Delta > 0$, то $l = d_j^* - x_j$; $l = 0$ при $\Delta = 0$; е) вычислим неопорные компоненты нового плана $x_j = x_j + \Theta l$. Здесь Δ , l , ξ — скалярные величины, принимающие новые значения при каждом $j \in J_{\text{н}}$.

15. Вычислим опорные компоненты нового плана $x_j = x_j + \Theta l_j$, $j \in J_{\text{оп}}$ ($l(J_{\text{оп}})$ вычислено в п. 7). Если $(1 - \Theta)\beta < \varepsilon$, то x — субоптимальный план и, следовательно, перейдем в п. 25. Если $\sigma_{j_*} = 0$, то перейдем в п. 18, так как в этом случае $\sigma = \sigma_{j_*}$.

16. Найдем σ_{i_*} . Положив $i_* = 0$, $\sigma_{i_*} = \sigma_{j_*}$, для $i \in I_{\text{оп}}$ выполним следующие операции: а) $\xi = k A_{\text{оп}}^{-1}(j_0, i)$; б) если $u_i = 0$, $\xi > 0$, $\omega_i \neq b_i^*$ или $u_i = 0$, $\xi < 0$, $\omega_i \neq b_{*i}$, то положим $\sigma_{i_*} = 0$, $i_* = i$, $\alpha = k\xi$; в) при $u_i \xi < 0$ вычислим $\sigma = -u_i/\xi$, если $\sigma < \sigma_{i_*}$, то положим $\sigma_{i_*} = \sigma$, $i_* = i$, $\alpha = k\xi$.

17. Если $i_* > 0$ ($\sigma = \sigma_{i_*}$), то, положив $\bar{I}_{\text{оп}} = I_{\text{оп}} \setminus i_*$, $\bar{J}_{\text{оп}} = J_{\text{оп}} \setminus j_0$, вычислим новую матрицу $\bar{A}_{\text{оп}}^{-1} = A_{\text{оп}}^{-1}(I_{\text{оп}} \setminus i_*, J_{\text{оп}} \setminus j_0) - A_{\text{оп}}^{-1}(i_*, J_{\text{оп}} \setminus j_0) A_{\text{оп}}^{-1}(I_{\text{оп}} \setminus i_*, j_0)/\alpha$. (Для удобства удаленные строку и столбец следует поменять местами с последними (r -ми) строкой и столбцом. Аналогичную перестановку нужно сделать и в множествах $I_{\text{оп}}$ и $J_{\text{оп}}$.) Положив $r = r - 1$, перейдем в п. 2, т. е. начнем выполнение новой итерации.

18. При $j_* > 0$ ($\sigma = \sigma_{j_*}$) вычислим $\bar{A}_{\text{оп}}^{-1} = A_{\text{оп}}^{-1} - [A_{\text{оп}}^{-1} A(I_{\text{оп}}, j_*) - l_{j_0}] e'_{j_0} A_{\text{оп}}^{-1}/\alpha$ и, положив $\bar{J}_{\text{оп}} = (J_{\text{оп}} \setminus j_0) \cup j_*$, перейдем в п. 2.

19. Пусть $i_0 > 0$ ($\Theta = \Theta_{i_0}$), вычислим $v(I_{\text{оп}}) = A(i_0, J_{\text{оп}}) \times \times A_{\text{оп}}^{-1}$.

20. Вычислим σ_{j_*} . Положив $j_* = 0$, $\sigma_{j_*} = \infty$, для $j \in J_{\text{н}}$ выполним операции: а) $\xi = k [A(i_0, j) - v(I_{\text{оп}}) A(I_{\text{оп}}, j)]$; б) — е) из п. 14.

21. Вычислим опорные компоненты плана $x_j = x_j + \Theta l_j$, $j \in J_{\text{оп}}$. Если $(1 - \Theta)\beta \leq \varepsilon$, то перейдем в п. 25; если $\sigma_{j_*} = 0$, то — в п. 24.

22. Вычислим σ_{i_*} . Положив $\sigma_{i_*} = \sigma_{j_*}$, $i_* = 0$, для $i \in I_{\text{оп}}$ выполним операции: а) $\xi = -kv_i$; б) — в) из п. 16.

23. При $i_* > 0$ ($\sigma = \sigma_{i_*}$) вычислим $\bar{A}_{\text{оп}}^{-1} = A_{\text{оп}}^{-1} + A_{\text{оп}}^{-1} \times \times e_{i_*}' [e_{i_*}' - v(I_{\text{оп}})]/\alpha$ и, положив $\bar{I}_{\text{оп}} = (I_{\text{оп}} \setminus i_*) \cup i_0$, перейдем на новую итерацию, т. е. в п. 2.

24. При $j_* > 0$ ($\sigma = \sigma_{j_*}$) вычислим

$$\bar{A}_{\text{оп}}^{-1} = \begin{bmatrix} A_{\text{оп}}^{-1} + A_{\text{оп}}^{-1} A(I_{\text{оп}}, j_*) v(I_{\text{оп}})/\alpha; & -A_{\text{оп}}^{-1} A(I_{\text{оп}}, j_*)/\alpha \\ -v(I_{\text{оп}})/\alpha; & 1/\alpha \end{bmatrix}$$

и, положив $\bar{I}_{\text{оп}} = I_{\text{оп}} \cup i_0$, $\bar{J}_{\text{оп}} = J_{\text{оп}} \cup j_*$, $r = r + 1$, перейдем в п. 2.

25. Вывод результатов: x — субоптимальный план, $c'x$ — значение целевой функции на этом плане, it — число выполненных итераций.

З а м е ч а н и я. 1. Некоторые векторы, например $w(I_{\text{оп}})$ в п. 5, были введены для сокращения описания алгоритма. При его реализации компоненты этих векторов легко восстановить через другие.

2. Индексным множествам $I_{\text{оп}}$ и $J_{\text{оп}}$ в программе соответствуют файлы целых чисел; перебирая подряд элементы этих файлов, получаем индексы ($i \in I_{\text{оп}}$ или $j \in J_{\text{оп}}$), например, плана x .

3. Здесь $e_{i_0}' A_{\text{оп}}^{-1}$ и $A_{\text{оп}}^{-1} e_{j_0} - i_0$ — строка и j_0 — столбец матрицы $A_{\text{оп}}^{-1}$ соответственно и, конечно, умножение вектора на матрицу в этом случае выполнять не нужно.

4. В отличие от алгоритма § 1 в данном алгоритме не вводится дополнительно ни одного n -мерного вектора. Это, во-первых, приводит к дополнительным вычислениям на итерации: дважды вычисляется коплан (при этом вызывается по столбцам вся матрица A), во-вторых, увеличивает число итераций из-за необходимости выполнять короткий двойственный шаг (без полной минимизации приращения двойственной функции).

Табл. 9

$m \times n$		30×45			70×100			100×150			200×250
		3	4	5	1	2	3	2	3	4	1
Число итераций	§ 1	24	28	29	43	54	12	16	18	84	111
	§ 2	34	36	30	43	58	81	86	97	116	117
Время решения, с	§ 1	12	15	15	56	64	120	202	225	275	720
	§ 2	20	23	17	109	187	317	140	902	1182	1916

2. Численные эксперименты. Для сравнения алгоритмов § 1, 2 по числу итераций и по времени счета приведем результаты решения нескольких задач, сгенерированных так же, как и в эксперименте 3 § 1 (табл. 9). Программная реализация алгоритма § 2 была выполнена на языке АЛГОЛ-ГДР для БЭСМ-6.

§ 3. Мультипликативная форма

Одна из широко распространенных версий симплекс-метода, удобная с точки зрения экономии оперативной памяти для реализации на ЭВМ, называется мультипликативным методом. Изложению соответствующей версии адаптивного метода из § 1 посвящен данный параграф. Приводятся детали реализации, полезные программистам при работе над аналогичными вопросами для других методов.

1. Мультипликаторы. В соответствии с элементарными преобразованиями, которые выполняются над опорной матрицей на итерациях адаптивного алгоритма, мультипликативное представление обратной матрицы будет получено в следующем виде:

$$A_{оп}^{-1} = \eta_k \cdot \eta_{k-1} \cdot \dots \cdot \eta_2 \cdot \eta_1 \cdot \xi_1 \cdot \xi_2 \cdot \dots \cdot \xi_{v-1} \cdot \xi_v, \quad (1)$$

где $\eta_i, i = \overline{1, k}$, — левые мультипликаторы, отличающиеся от единичной матрицы одним столбцом; $\xi_j, j = \overline{1, v}$, — правые мультипликаторы, которые отличаются от единичной матрицы одной строкой.

Приведем формулы, по которым вычисляются мультипликаторы [1, 35], соответствующие преобразованиям (19) — (22) из § 1.

$$1) \text{ Замена строки: } \bar{I}_{оп} = (I_{оп} \setminus i_*) \cup i_0, \quad \bar{J}_{оп} = J_{оп}, \\ \bar{A}_{оп}^{-1} = A_{оп}^{-1} \xi,$$

$$\xi = \begin{bmatrix} E; & 0 \\ -h(I_{оп} \setminus i_*) / h(i_*); & 1/h(i_*) \end{bmatrix}, \\ h(I_{оп}) = A(i_0, J_{оп}) A_{оп}^{-1}. \quad (2)$$

$$2) \text{ Добавление строки и столбца: } \bar{I}_{оп} = I_{оп} \cup i_0, \quad \bar{J}_{оп} = J_{оп} \cup j_*,$$

$$\bar{A}_{оп}^{-1} = \eta A_{оп}^{-1} \xi, \quad \eta = \begin{bmatrix} E; & -g(J_{оп}) \\ 0; & 1 \end{bmatrix},$$

$$g(J_{\text{оп}}) = A_{\text{оп}}^{-1} A(I_{\text{оп}}, j_*), \quad (3)$$

$$\xi = \begin{bmatrix} E; & 0 \\ -h(I_{\text{оп}})/\alpha; & 1/\alpha \end{bmatrix}, \quad h(I_{\text{оп}}) = A(i_0, J_{\text{оп}}) A_{\text{оп}}^{-1}, \quad (4)$$

$$\alpha = A(i_0, j_*) - A(i_0, J_{\text{оп}}) g(J_{\text{оп}}).$$

3) Искключение строки и столбца: $\bar{I}_{\text{оп}} = I_{\text{оп}} \setminus i_*$, $\bar{J}_{\text{оп}} = J_{\text{оп}} \setminus j_0$,

$$\bar{A}_{\text{оп}}^{-1} = \eta A_{\text{оп}}^{-1} \xi, \quad \eta = \begin{bmatrix} E; & g(J_{\text{оп}} \setminus j_0) \\ 0; & 1/g(j_0) \end{bmatrix},$$

$$g(J_{\text{оп}}) = A_{\text{оп}}^{-1}(J_{\text{оп}}, i_*), \quad (5)$$

$$\xi = \begin{bmatrix} E; & 0 \\ -h(I_{\text{оп}} \setminus i_*)/h(i_*); & 1 \end{bmatrix}, \quad h(I_{\text{оп}}) = A_{\text{оп}}^{-1}(j_0, I_{\text{оп}}). \quad (6)$$

4) Замена столбца: $\bar{I}_{\text{оп}} = I_{\text{оп}}$, $\bar{J}_{\text{оп}} = (J_{\text{оп}} \setminus j_0) \cup j_*$,
 $\bar{A}_{\text{оп}}^{-1} = \eta \cdot A_{\text{оп}}^{-1}$,

$$\eta = \begin{bmatrix} E; & -g(J_{\text{оп}} \setminus j_0)/g(j_0) \\ 0; & 1/g(j_0) \end{bmatrix}, \quad g(J_{\text{оп}}) = A_{\text{оп}}^{-1} A(I_{\text{оп}}, j_0). \quad (7)$$

2. Основные процедуры для работы с мультипликаторами. Векторы g и h для мультипликаторов (2) — (7) вычисляются посредством умножения столбца или строки матрицы A (их опорных элементов) на $A_{\text{оп}}^{-1}$ слева или справа. Только в формулах (5), (6) умножение на $A_{\text{оп}}^{-1}$ выполняется для единичного вектора: $g = A_{\text{оп}}^{-1} e_{i_*}$, $h' = e_{j_0}' A_{\text{оп}}^{-1}$.

На итерациях адаптивного алгоритма матрица $A_{\text{оп}}^{-1}$ так же, как и в симплекс-методе, используется только при разложении заданных векторов по строкам или столбцам матрицы $A_{\text{оп}}$. Для этого опять-таки выполняется умножение вектора на $A_{\text{оп}}^{-1}$ справа или слева. Таким образом, нет необходимости вычислять матрицу $A_{\text{оп}}^{-1}$ по формуле (1). Для реализации алгоритма достаточно иметь вычислительную процедуру, которая организует умножение заданного вектора на матрицу $A_{\text{оп}}^{-1}$ в форме (1) как слева, так и справа:

```
'PROCEDURE' MULT (Q, AP, AL, NMP, NML, MP, ML, K);
'ARRAY' Q; 'REAL' AP, AL; 'INTEGER' 'ARRAY' NMP, NML;
'INTEGER' MP, ML, K; 'BEGIN' 'INTEGER' I, J, J1, J2, I2;
'REAL' R1, R2; 'ARRAY' P[1:K+1];
```

```

'FOR' I:=MP 'STEP'-1 'UNTIL' 1 'DO'
'BEGIN' I2:=NMP[I+1]-NMP[I]; R1:=0;
REXRD(P[I], P[I2], AP, NMP[I]);
'FOR' J:=1 'STEP' 1 'UNTIL' I2-1 'DO'
'BEGIN' PLA0R1 (P [J], J2); R1:= R1 + P[J] * Q [J2]
'END' J;
J:=P[I2]; Q[J]:=R1
'END' I;
'FOR' I:=1 'STEP' 1 'UNTIL' ML 'DO'
'BEGIN' I2:=NML[I+1]-NML[I];
REXRD(P[I], P[I2], AL, NML[I]); J2:=P[I2]; R1:=Q[J2];
'IF' ABS(R1) > E1 'THEN'
'BEGIN' 'FOR' J:=1 'STEP' 1 'UNTIL' I2-1 'DO'
'BEGIN' PLA0R1 (P[J], J1);
'IF' J1 = J2 'THEN' Q [J1] := P [J] * R1
'ELSE' Q [J1] := Q [J1] + P [J] * R1
'END' J
'END'
'END' I
'END' MULT;

```

Назначение параметров процедуры MULT: Q — вектор, который умножается на матрицу (1), результат заносится на место этого же вектора; AP, AL — файлы, где хранятся правые и левые мультипликаторы; NMP, NML — целые массивы для хранения начальных адресов правых и левых мультипликаторов в файлах AP и AL; MP, ML — число правых и левых мультипликаторов; K — размер опоры.

REXRD(P[I], P[I2], AP, NMP[I]) списывает правый мультипликатор из файла AP, начиная с AP[NMP[I]], в массив P. Мультипликатор имеет I2—1 ненулевых элементов в строке (или столбце) с номером, находящимся в P[I2].

PLA0R1(P[J], J2) из пяти последних разрядов P[J] выделяет индекс ненулевого элемента (P[J]) (например, номер столбца для строчного мультипликатора) и заносит в J2.

Процедура MULT выполняет умножение вектора на матрицу (1) слева или справа в зависимости от порядка расположения параметров. Например, если требуется выполнить умножение $c'(J_{\text{оп}})A_{\text{оп}}^{-1}$, то обращение к этой процедуре записывается так: MULT(C, AL, AP, NML, NMP, ML, MP, K), а при вычислении произведения вида $A_{\text{оп}}^{-1}Q(I_{\text{оп}})$ — MULT(Q, AP, AL, NMP, NML, MP, ML, K).

Поскольку в больших задачах ЛП матрица A , как правило, сильно разрежена (число ненулевых элементов менее 1 %), то и векторы h и g , по которым строятся мультипликаторы, будут содержать большое число нулевых элементов. Поэтому в файлы AL и AP следует записывать лишь ненулевые элементы и их индексы.

Пусть в Q находится вектор h (или g), с помощью которого нужно вычислить мультипликатор ξ (или η) и записать его в компактной форме в файл AP (или AL). Эту операцию можно выполнить с помощью следующей процедуры:

```
'PROCEDURE' SPMR (I0, Q, ML, NML, AL, B, K); 'ARRAY' Q;
'REAL' AL; 'INTEGER' I0, ML, K; 'INTEGER' 'ARRAY' NML;
'BOOLEAN' B; 'BEGIN' 'INTEGER' I, J, J1, I1; 'REAL' R1, R2;
'ARRAY' P[1:K+1]; R1:=1/Q[I0]; R2:='IF' B 'THEN' 1 'ELSE' R1;
J:=0; 'FOR' I:=1 'STEP' 1 'UNTIL' K 'DO'
'BEGIN' 'IF' ABS(Q[I]) < 10-10 'THEN' 'GO' 'TO' EK;
J:=J+1; P[J]:='IF' I=I0 'THEN' R2 'ELSE' -Q[I]*R1;
PLA031(P[J], I); EK:'END' I;
J1:=J+1; P[J1]:=I0; ML:=ML+1; I1:=NML[ML];
NML[ML+1]:=I1+J1; REXWT(P[I], P[J1], AL, I1)
'END' SPMR;
```

Здесь $I0$ — номер ведущей строки (столбца) мультипликатора; ML — число ранее записанных мультипликаторов; NML запоминает начальный адрес следующего мультипликатора, при $B='TRUE'$ на место ведущего элемента записывается 1, при $B='FALSE'$ — $1/Q[I0]$; K — размерность опоры; AL — файл, в который записываются мультипликаторы.

PLA031(P[J], I) запаковывает i -й элемент мультипликатора вместе с индексом i в одну ячейку.

REXWT(P[I], P[J1], AL, I1) переписывает информацию о мультипликаторе в файл AL, начиная с адреса $I1=NML[ML]$. При формировании и записи, например, правого мультипликатора ξ (2) достаточно с помощью оператора MULT получить вектор $H=h(I_{оп})$, $IZ=i_*$, а затем выполнить оператор SPMR(IZ, H, MP, NMP, AP, 'FALSE', K).

З а м е ч а н и е. Если число итераций велико, то даже при компактной записи длина файлов AL и AP, необходимая для запоминания мультипликаторов, может стать чрезмерно большой. Во избежание трудностей, связанных с этим обстоятельством, в программе следует предусмотреть процедуру повторения (обновления), которая по имеющейся совокупности опорных строк и столбцов формирует новые

файлы меньшей длины, так как в процессе их формирования будут записываться только мультипликаторы вида (3) и (4), соответствующие накоплению опоры.

Рассмотрим мультипликативный вариант адаптивного алгоритма, в котором, как и в алгоритме § 2, оценки Δ_j , $j \in J_{\text{н}}$, вычисляются через вектор потенциалов, но в отличие от алгоритма § 2 эти оценки запоминаются, что позволяет избежать их повторного вычисления и организовать оптимальную замену опоры.

3. Алгоритм. 1. Зададим план x и положим $I_{\text{оп}} = \emptyset$, $J_{\text{оп}} = \emptyset$, $I_{\text{н}} = \{1, 2, \dots, m\}$, $J_{\text{н}} = \{1, 2, \dots, n\}$, $it = 0$, $k = r = 0$, $ML = MP = 0$, $NML[1] = NMP[1] = 1$.

2. Вычислим вектор потенциалов $u'(I_{\text{оп}}) = c'(J_{\text{оп}})A_{\text{оп}}^{-1}$. Для этого при $U = c(J_{\text{оп}})$ выполним процедуру MULT (U , AL , AP , NML , NMP , ML , MP , K), после чего будем иметь $U = u'(I_{\text{оп}}) = c'(J_{\text{оп}})A_{\text{оп}}^{-1}$. Если $ML = MP = 0$, в MULT вычислений не будет.

3. Положив $v(I) = \omega(I) = 0$, $\beta = 0$, выполним в цикле по $j \in J = \{1, 2, \dots, n\}$ следующие операции: а) $\omega(I) = \omega(I) + A(I, j)x_j$; б) $\Delta_j = u'A(I_{\text{оп}}, j) - c_j$; в) $l_j = d_j^* - x_j$, если $\Delta_j < 0$; $l_j = d_{*j} - x_j$, если $\Delta_j > 0$; $l_j = 0$, если $\Delta_j = 0$; г) $v(I) = v(I) - A(I, j)l_j$; д) $\beta = \beta - \Delta_j l_j$. Здесь $\Delta(J)$ и $l(J)$ — n -векторы, поэтому значения Δ_j , l_j , $j \in J_{\text{н}}$, запоминаются.

4. Вычислим $w(I_{\text{оп}})$: $w_i = b_i^* - \omega_i$, если $u_i > 0$; $w_i = b_{*i} - \omega_i$, если $u_i < 0$; $w_i = 0$, если $u_i = 0$.

5. Досчитаем оценку субоптимальности $\beta = \beta + u'(I_{\text{оп}})w(I_{\text{оп}})$ и проверим неравенство $\beta \leq \varepsilon$. Да \rightarrow п. 19, нет \rightarrow п. 6.

6. Вычислим $l(J_{\text{оп}}) = A_{\text{оп}}^{-1}(w(I_{\text{оп}}) - v(I_{\text{оп}}))$. Для этого положим $L = w(I_{\text{оп}}) - v(I_{\text{оп}})$ и обратимся к процедуре MULT (L , AP , AL , NMP , NML , MP , ML , K).

7—10. Выполнив пп. 8—10 из алгоритма § 2, найдем шаг Θ .

11. Вычислим новый план $\bar{x}_j = x_j + \Theta l_j$, $j \in J$, и псевдоплан $\kappa_j = x_j + l_j$, $j \in J$; положим $it = it + 1$; если $(1 - \Theta)\beta \leq \varepsilon$, то перейдем в п. 19.

12. Пусть $\Theta = \Theta_{j_0}$. 1) При $\kappa_{j_0} < d_{*j_0}^*$ положим $k_1 = 1$, $\alpha = \kappa_{j_0} - d_{*j_0}^*$; 2) при $\kappa_{j_0} > d_{j_0}^*$ положим $k_1 = -1$, $\alpha = d_{j_0}^* - \kappa_{j_0}$; 3) по формуле (15) § 1 вычислим подходящее двойственное направление. Для этого, положив $P = p_i$, $p_i = 0$, $i = \overline{1, k}$, $i \neq j_0$, $p_{j_0} = 1$, выполним оператор MULT (P , AL ,

AP, NML, NMP, ML, MP, K). В результате получим вектор $p(I_{\text{оп}}) = e'_{j_0} A_{\text{оп}}^{-1} = A_{\text{оп}}^{-1}(j_0, I_{\text{оп}})$ и с его помощью построим $\xi(I_{\text{оп}} \cup J_{\text{н}}) = \{-k_1 p(I_{\text{оп}}), k_1 p(I_{\text{оп}}) A(I_{\text{оп}}, J_{\text{н}})\}$. Перейдем в п. 14.

13. Пусть $\Theta = \Theta_{i_0}$. 1) При $a'_{i_0} \kappa < b_{*i_0}$ положим $k_1 = -1$, $\alpha = a'_{i_0} \kappa - b_{*i_0}$; 2) при $a'_{i_0} \kappa > b_{*i_0}$ положим $k_1 = 1$, $\alpha = b_{*i_0} - a'_{i_0} \kappa$; 3) по формуле (14) § 1 вычислим подходящее направление для двойственного плана. Для этого, положив $Q = A(i_0, J_{\text{оп}})$, выполним оператор MULT(Q, AL, AP, NML, NMP, ML, MP, K). Тогда будем иметь $Q = q(I_{\text{оп}}) = A(i_0, J_{\text{оп}}) A_{\text{оп}}^{-1}$, с помощью которого вычислим $\xi(I_{\text{оп}} \cup J_{\text{н}}) = \{k_1 q(I_{\text{оп}}), k_1 [A(i_0, J_{\text{н}}) - q'(I_{\text{оп}}) A(I_{\text{оп}}, J_{\text{н}})]\}$.

14. Выполним оптимальный (в смысле минимума приращения двойственной функции) двойственный шаг. Это делается так же, как в пп. 11—16 алгоритма § 1. В результате получим индексы i_* и j_* для преобразования опоры.

15. Пусть $\Theta = \Theta_{i_0}$, $\sigma = \sigma_{i_*}$ (при $\Theta = \Theta_{j_0}$ перейдем в п. 17). Правый мультипликатор (2) $Q = q(I_{\text{оп}}) = A(i_0, J_{\text{оп}}) A_{\text{оп}}^{-1}$ уже вычислен в п. 13. Выполним его запись в файл AP:IZ = i_* , SPMR(IZ, Q, MP, NMP, AP, 'FALSE', K). Полагая $\bar{I}_{\text{оп}} = (I_{\text{оп}} \setminus i_*) \cup i_0$, перейдем в п. 2.

16. Пусть $\Theta = \Theta_{i_0}$, $\sigma = \sigma_{j_*}$. Тогда при $Q = A(I_{\text{оп}}, j_*)$ выполним оператор MULT(Q, AP, AL, NMP, NML, MP, ML, K). Получим $Q = q(I_{\text{оп}}) = A_{\text{оп}}^{-1} A(I_{\text{оп}}, j_*)$. При $P = A(i_0, J_{\text{оп}})$ выполним оператор MULT(P, AL, AP, NML, NMP, ML, MP, K). Получим $P = p(I_{\text{оп}}) = A(i_0, J_{\text{оп}}) A_{\text{оп}}^{-1}$. Положив $K = K + 1$, $Q[K] = 1$, запишем левый мультипликатор (3) в файл AL:SPMR(K, Q, ML, NML, AL, 'TRUE', K). Положив $P[K] = k_1 \xi_{j_*}$, запишем правый мультипликатор (4) в файл AP:SPMR(K, P, MP, NMP, AP, 'FALSE', K). Положив $r = r + 1$, $\bar{I}_{\text{оп}} = I_{\text{оп}} \cup i_0$, $\bar{J}_{\text{оп}} = J_{\text{оп}} \cup j_*$, перейдем на новую итерацию, т. е. в п. 2.

17. Пусть $\Theta = \Theta_{j_0}$, $\sigma = \sigma_{i_*}$. Положив $Q = q_i$, $q_i = 0$, $i = \overline{1, k}$, $i \neq i_*$, $q_{i_*} = 1$, выполним оператор MULT(Q, AP, AL, NMP, NML, MP, ML, K). Получив $Q = A_{\text{оп}}^{-1}(J_{\text{оп}}, i_*)$ — левый мультипликатор (5), запишем его в файл AL:J0 = j_0 , SPMR(J0, Q, ML, NML, AL, 'FALSE', K). Правый мультипликатор (2) $P = A_{\text{оп}}^{-1}(j_0, I_{\text{оп}})$ был вычислен в п. 12. Запишем его в файл AP:IZ = i_* , SPMR(IZ, P, MP,

NMP, AP, 'TRUE', K). Полагая $r = r - 1$, $\bar{I}_{оп} = I_{оп} \setminus i_*$, $\bar{J}_{оп} = J_{оп} \setminus j_0$, перейдем в п. 2.

18. Пусть $\Theta = \Theta_{j_0}$, $\sigma = \sigma_{j_*}$. Положив $Q = A(I_{оп}, j_*)$, выполним оператор MULT(Q, AP, AL, NMP, NML, MP, ML, K). Будем иметь $Q = A_{оп}^{-1} A(I_{оп}, j_*)$ — левый мультипликатор (7), запись которого в файл AL реализует оператор SPMR(J0, Q, ML, NML, AL, 'FALSE', K). Положив $\bar{J}_{оп} = (J_{оп} \setminus j_0) \cup j_*$, перейдем в п. 2.

19. Вывод результатов: x — субоптимальный план, $c'x$ — значение целевой функции, it — число итераций.

З а м е ч а н и е. Описанный алгоритм не включает процедуру повторения, которая работает по достижению большого числа левых или правых мультипликаторов ($ML > MAX \vee MP > MAX$). Эта процедура повторяет операции п. 16 при $j_* \in J_{оп}$, $i_0 \in I_{оп}$, $k = \overline{1, r}$ (не меняя r — достигнутый размер опоры), кроме операции получения $P[K] = \alpha$ (из (4)) и дополнительной операции для упорядочения опорных индексов.

4. Численный эксперимент. Некоторые примеры из табл. 5 были решены и с помощью мультипликативного алгоритма. Результаты приведены в табл. 10 для сравнения с алгоритмами § 1, 2.

Табл. 10

$m \times n$		70×100		100×150		200×250
k		1	2	1	2	1
Число итераций	§ 1	43	45	59	76	111
	§ 2	43	59	62	86	117
	§ 3	43	45	59	76	111
Время решения, с	§ 1	56	64	156	202	720
	§ 2	109	187	401	740	1916
	§ 3	67	75	180	242	876

§ 4. Адаптивный алгоритм для канонической задачи

1. Постановка задачи. Рассмотрим линейную задачу

$$c'x \rightarrow \max, Ax = b, d_* \leq x \leq d^*, \quad (1)$$

где c , x , d_* , d^* — n -векторы, b — m -вектор, A — $m \times n$ -

матрица, $A = A(I, J)$, причем $\text{rang } A = m$, $n > m$, $I = \{1, 2, \dots, m\}$, $J = \{1, 2, \dots, n\}$.

Пусть допустимое отклонение $\varepsilon \geq 0$ от максимума целевой функции, начальный план x и опора $A_{\text{оп}} = A(I, J_{\text{оп}})$, $J_{\text{оп}} \subset J$, заданы. Требуется построить алгоритм поиска ε -оптимального плана задачи (1): $c'x^0 - c'x \leq \varepsilon$, где x^0 — оптимальный план.

2. Критерий оптимальности и субоптимальности. Из уравнения $u'A_{\text{оп}} = c'(J_{\text{оп}})$ найдем вектор потенциалов $u' = c'(J_{\text{оп}}) A_{\text{оп}}^{-1}$. Вычислим оценки $\Delta_j = u'A(I, j) - c_j$, $j \in J_{\text{н}}$, $J_{\text{н}} = J \setminus J_{\text{оп}}$. Тогда приращение целевой функции, соответствующее допустимой вариации Δx ($A\Delta x = 0$), можно записать в виде

$$c'\Delta x = -\Delta'(J_{\text{н}})\Delta x(J_{\text{н}}). \quad (2)$$

Условие неположительности приращения (2) для допустимых Δx_j ($\Delta x_j \leq 0$, если $x_j = d_j^*$; $\Delta x_j \geq 0$, если $x_j = d_{*j}$; Δx_j — произвольное, если $d_{*j} < x_j < d_j^*$) сводится к соотношениям

$$\begin{aligned} \Delta_j \geq 0 \text{ при } x_j = d_{*j}, \Delta_j \leq 0 \text{ при } x_j = d_j^*, \Delta_j = 0 \\ \text{при } d_{*j} < x_j < d_j^*, \end{aligned} \quad (3)$$

составляющим достаточное условие оптимальности плана x .

Максимум приращения (2) при условиях

$$d_{*j} - x_j \leq \Delta x_j \leq d_j^* - x_j, \quad j \in J_{\text{н}}, \quad (4)$$

равен

$$\beta = - \sum_{\Delta_j > 0} \Delta_j (d_{*j} - x_j) - \sum_{\Delta_j < 0} \Delta_j (d_j^* - x_j). \quad (5)$$

Поскольку при вычислении β ограничения на $\Delta x(J_{\text{оп}})$ не были учтены, то $\beta \geq c'x^0 - c'x$, т. е. β — оценка сверху для приращения целевой функции. Таким образом, выполнение неравенства

$$\beta \leq \varepsilon \quad (6)$$

означает ε -оптимальность плана x : $c'x^0 - c'x \leq \varepsilon$.

3. Подходящее направление для плана x . Пусть при заданном ε план x не удовлетворяет условию (6). Построим подходящее направление $l(J_{\text{н}})$ из условия максимального приращения целевой функции (2) при допустимых приращениях (4) плана x :

$$l_j = d_j^* - x_j, \text{ если } \Delta_j < 0; \quad l_j = d_{*j} - x_j, \text{ если } \Delta_j > 0; \\ l_j = 0, \text{ если } \Delta_j = 0, j \in J_{\text{н}}. \quad (7)$$

Опорные компоненты $l_j, j \in J_{\text{оп}}$, найдем из равенства $Al=0$:

$$l(J_{\text{оп}}) = -A_{\text{оп}}^{-1}A(I, J_{\text{н}})l(J_{\text{н}}). \quad (8)$$

4. Вычисление шага. При вычислении нового плана $\bar{x} = x + \Theta l$, где l определено по формулам (7), (8), $\Theta \leq 1$, могут нарушиться только ограничения на опорные переменные. Вычислим максимально возможный шаг вдоль l :

$$\Theta = \min \{1, \Theta_{j_0}\}; \quad \Theta_{j_0} = \min_{j \in J_{\text{оп}}} \Theta_j, \quad \Theta_j = (d_j^* - x_j) / l_j, \\ \text{если } l_j > 0; \quad (9)$$

$$\Theta_j = (d_{*j} - x_j) / l_j, \text{ если } l_j < 0; \quad \Theta_j = \infty, \text{ если } l_j = 0.$$

Новый план $\bar{x} = x + \Theta l$ при $\Theta = 1$ будет оптимальным, при $(1 - \Theta)\beta \leq \varepsilon$ — ε -оптимальным.

5. Разложение оценки субоптимальности. Дальнейшее улучшение плана x возможно только при новой опоре. Операцию замены опоры проведем с помощью итерации в двойственной к (1) задаче:

$$b'y + d_*'w - d_*'v \rightarrow \min, \quad A'y + w - v \geq c, \quad w \geq 0, \quad v \geq 0. \quad (10)$$

Имея коплан Δ , сопровождающий опорный план $\{x, A_{\text{оп}}\}$, легко построить план задачи (10):

$$y' = u' = c'(J_{\text{оп}})A_{\text{оп}}^{-1}; \quad w_j = 0, \quad v_j = \Delta_j, \text{ если } \Delta_j \geq 0; \quad (11) \\ w_j = -\Delta_j, \quad v_j = 0, \text{ если } \Delta_j \leq 0.$$

Подставив двойственный план (11) в выражение (5) и воспользовавшись равенством $c'x^0 = b'y^0 + d_*'w^0 - d_*'v^0$, получим

$$\beta = \Delta'(J)x(J) - v'd_* + w'd_* = (y'A - c')x - v'd_* + w'd_* = \\ = c'x^0 - c'x + (b'y + d_*'w - d_*'v) - \\ - (b'y^0 - d_*'v^0 + d_*'w^0) = \beta_x + \beta_{\text{оп}}, \quad (12)$$

где β_x — мера неоптимальности плана x , $\beta_{\text{оп}}$ — мера неоптимальности опоры.

6. Двойственный шаг. Построим подходящее направление для коплана Δ . Положив $k = -1$, если $\bar{x}_{j_0} = d_{j_0}^*$; $k = 1$, если $\bar{x}_{j_0} = d_{*j_0}$, вычислим вектор

$$\xi(J_{\text{н}}) = ke'_{j_0}A_{\text{оп}}^{-1}A(I, J_{\text{н}}). \quad (13)$$

Двойственный шаг σ вдоль $\xi(J_H)$ найдем, минимизируя приращение двойственной функции $\beta_{оп}$ в (12).

Начальная скорость убывания двойственной функции вдоль ξ равна: $\alpha_0 = \kappa_{j_0} - d_{*j_0}$, если $k = 1$, $\alpha_0 = d_{j_0}^* - \kappa_{j_0}$, если $k = -1$, где $\kappa = x + l$.

При движении вдоль направления $\xi(J_H)$ скорость убывания двойственной функции остается постоянной, пока некоторые компоненты коплана $\bar{\Delta} = \Delta + \sigma \xi$ не обратятся в нуль. Если до этого момента скорость была равной α_k , то при дальнейшем движении она будет равна

$$\alpha_{k+1} = \alpha_k + \sum_{\xi_j < 0, \Delta_j^{(k)} = 0} \xi_j (\kappa_j - d_j^*) + \sum_{\xi_j > 0, \Delta_j^{(k)} = 0} \xi_j (\kappa_j - d_{*j}), \quad (14)$$

где $\Delta_j^{(k)}$ — новые нулевые компоненты, полученные при движении, соответствующем скорости α_k . Если на каком-то этапе получим $\alpha_s \geq 0$, то движение вдоль $\xi(J_H)$ следует прекратить. Пусть длительность движения при скорости α_k равна σ_k . Тогда новый коплан, полученный в результате движения вдоль $\xi(J_H)$, $\bar{\Delta}(J_H) = \Delta(J_H) + \sum_{k=0}^{s-1} \sigma_k \xi(J_H)$. Любой индекс j_* из множества $J_{H0} = \{j : j \in J_H, \bar{\Delta}_j = 0, \xi_j(\kappa_j - d_{*j}) > 0 \vee \xi_j(\kappa_j - d_j^*) > 0\}$, например, тот, с которым было вычислено последнее положительное слагаемое в (14), можно взять для преобразования опоры.

7. Преобразование опоры. В опоре $A_{оп}$ заменим столбец a_{j_0} на a_{j_*} , $\bar{J}_{оп} = (J_{оп} \setminus j_0) \cup j_*$. При этом оценка субоптимальности $\beta_{оп}$ уменьшится на величину $r = \sum_{k=0}^{s-1} |\alpha_k| \sigma_k$ и, следовательно, станет равной $\bar{\beta} = (1 - \Theta) \beta - r$. Если $\bar{\beta} \leq \varepsilon$, то x — ε -оптимальный план.

Матрица, обратная к $\bar{A}_{оп}(I, \bar{J}_{оп})$, вычисляется по формуле

$$\bar{A}_{оп}^{-1} = [E + (e_{j_0} - A_{оп}^{-1} A(I, j_*)) e'_{j_0} / \alpha] A_{оп}^{-1}, \quad \alpha = k \xi_{j_*}. \quad (15)$$

8. Алгоритм. 1. Вычислим вектор потенциалов $u' = c'(J_{оп}) A_{оп}^{-1}$ и найдем оценки $\Delta_j = u' A(I, j) - c_j$, $j \in J_H$.

2. По формуле (5) найдем оценку субоптимальности β . При $\beta \leq \varepsilon$ перейдем в п. 13.

3. Вычислим подходящее направление l по формулам (7), (8).

4. Максимально допустимый шаг Θ вдоль l найдем по формуле (9).

5. Вычислим новый план $\bar{x} = x + \Theta l$ и псевдоплан $\kappa = x + l$. При $(1 - \Theta)\beta \leq \varepsilon$ переходим в п. 13.

6. Положив $k = -1$, если $\bar{x}_{j_0} = d_{j_0}^*$; $k = 1$, если $\bar{x}_{j_0} = d_{*j_0}$, вычислим подходящее направление ξ для коплана Δ по формуле (13).

7. Вычислим начальную скорость убывания двойственной функции при движении вдоль ξ : $\alpha = \kappa_{j_0} - d_{*j_0}$, если $k = 1$; $\alpha = d_{j_0}^* - \kappa_{j_0}$, если $k = -1$.

8. Среди Δ_j , $j \in J_{\Pi}$, выявим нулевые компоненты и по формуле (14) найдем новое значение α . Если среди индексов j , по которым вычислена положительная добавка в (14), есть такие, что $d_{*j} < x_j < d_j^*$, то полагаем $j_* = j$. (В противном случае в качестве j_* берется любой индекс, с которым вычислена положительная добавка в (14).)

9. Если $\alpha \geq 0$, то переходим в п. 12, иначе — в п. 10.

10. Вычислим шаг $\sigma_k = \min\{-\Delta_j/\xi_j, \xi_j \Delta_j < 0, j \in J_{\Pi}\}$.

11. Пересчитаем коплан $\bar{\Delta}_j = \Delta_j + \sigma_k \xi_j$, $j \in J_{\Pi}$, и переходим в п. 8.

12. Преобразуем опору $\bar{J}_{\text{оп}} = (J_{\text{оп}} \setminus j_0) \cup j_*$, матрицу $A_{\text{оп}}^-$ пересчитаем по формуле (15). Положим $\Delta_{j_0} = k\sigma$, тем самым закончено построение нового коплана. Новую итерацию начинаем с п. 2.

13. Выдача результатов.

Результаты численного эксперимента приведены в § 5.

§ 5. Алгоритм с симплексной нормировкой

1. Схема метода. Рассмотрим прямой опорный метод решения

$$c'x \rightarrow \max, Ax = b, d_* \leq x \leq d^*, \quad (1)$$

задачи (1), отличающийся от изложенного в § 4 адаптивного алгоритма видом нормировочного условия, которое используется при построении подходящего направления. В данном параграфе максимум приращения (2) § 4

будем вычислять не при условии (4), а при симплексной нормировке $\sum_{j \in J_H} |\Delta x_j| = 1$.

Очевидно, что максимум функции $-\Delta'(J_H)l(J_H)$ при условии $\sum_{j \in J_H} |l_j| = 1$ достигается на векторе $l(J_H)$ с компонентами

$$\begin{aligned} l_{j_*} &= -\text{sign } \Delta_{j_*}, \quad l_j = 0, \quad j \in J_H, \quad j \neq j_*; \\ |\Delta_{j_*}| &= \max_{j \in J_H} |\Delta_j|. \end{aligned} \quad (2)$$

Опорные компоненты l вычисляются по формуле (8) § 4, шаг вдоль направления $l(J_{\text{оп}})$ вычисляется, как и в § 4, но при этом учитывается дополнительное условие $d_{*j_*} \leq x_{j_*} + \Theta l_{j_*} \leq d_{j_*}^*$.

Двойственный шаг в методе данного параграфа не выполняется, а для преобразования опоры используется индекс j_* , найденный в (2).

2. Алгоритм. 1. Вычислим вектор потенциалов $u' = c'(J_{\text{оп}})A_{\text{оп}}^{-1}$ и найдем оценки $\Delta_j = u'A(I, j) - c_j$, $j \in J_H$.

2. По формуле (5) § 4 найдем оценку субоптимальности β . Если $\beta \leq \varepsilon$, перейдем в п. 7.

3. Подходящее направление l для изменения плана найдем по формулам (8) § 4, (2).

4. Максимально допустимый шаг Θ вдоль $l(J_{\text{оп}})$ найдем по формуле (9) § 4. Положив $\Theta_{j_*} = d_{j_*}^* - x_{j_*}$ при $l_{j_*} = 1$; $\Theta_{j_*} = x_{j_*} - d_{*j_*}$ при $l_{j_*} = -1$, проверим неравенство $\Theta_{j_*} \leq \Theta$. Если неравенство выполнено, вычислим новый план $\bar{x} = x + \Theta_{j_*}l$ и перейдем в п. 2; если нет, перейдем в п. 5.

5. Вычислим новый план $\bar{x} = x + \Theta l$. При $(1 - \Theta)\beta \leq \varepsilon$ \bar{x} — субоптимальный план. Выдаем результаты.

6. Преобразуем опору $\bar{J}_{\text{оп}} = (J_{\text{оп}} \setminus j_0) \cup j_*$, пересчитаем $A_{\text{оп}}^{-1}$ по формуле (15) § 4 и перейдем в п. 1.

Для данного алгоритма в мультипликативном варианте так же, как и для алгоритма § 4, будет строиться последовательность только левых мультипликаторов вида (7) § 3.

Сравнивая трудоемкость итераций алгоритмов § 4, 5, замечаем, что в обоих алгоритмах необходимо выполнение основной операции — скалярное умножение m -вектора на все неопорные столбцы матрицы A . В алгоритме § 4 она выполняется при вычислении подходящего на-

правления $\xi(J_H)$ для коплана $\Delta(J_H)$, в алгоритме § 5 — при вычислении коплана $\Delta(J_H)$. Вторая из операций — пересчет $A_{оп}^{-1}$ — присутствует в обоих алгоритмах. Для реализации алгоритма § 4 требуется больший объем памяти для запоминания векторов $\Delta(J_H)$ и $\xi(J_H)$. Число итераций в алгоритме § 5 значительно больше, чем в алгоритме § 4, следовательно, решение задачи по алгоритму § 5 требует больше времени. Из приведенных ниже результатов экспериментов видно, что с ростом отношения n/m разница в числе итераций в алгоритмах § 4, 5 становится заметной.

3. Численные эксперименты. Задачи (1) для экспериментов генерировались процедурой GENER1, описанной в § 1. В пределах заданной размерности задачи различались с помощью параметра j , который принимал значения $j=33+7k$, $k=1, 2, 3, 4$. Элементы матрицы A генерировались при $i=\overline{1, m}$, где i — номер строки матрицы A со значениями на отрезке $[-100, 100]$; компоненты векторов c , d_* , d^* со значениями на отрезках $[-50, 50]$, $[-150, 0]$, $[0, 100]$ — при $i=m+1, m+3, m+4$ соответственно.

Эксперимент 1. Цель эксперимента — сравнить алгоритмы § 4, 5 по времени решения и числу итераций для различных значений m и n .

Вектор правых частей и начальный план брались равными нулю. Результаты эксперимента приведены в табл. 11.

Эксперимент 2. Цель эксперимента — сравнить алгоритмы § 4, 5 по времени решения и числу итераций при $n \gg m$.

Компоненты вектора генерировались при $i=m+2$ со значениями на отрезке $[20, 100]$. Для поиска начального

Табл. 11

$m \times n$		30×45				70×100			
Номер задачи		1	2	3	4	1	2	3	4
Число итераций	§ 4	18	15	12	13	48	53	38	54
	§ 5	21	21	18	18	58	56	51	58
Время решения, с	§ 4	3	2	2	2	30	32	26	32
	§ 5	3	3	3	3	35	35	30	36

опорного плана использовался алгоритм § 5. Как и на первой фазе симплекс-метода, решалась задача — $\sum_{i=1}^m x_{n+i} \rightarrow \max$, $Ax + x_u = b$, $d_* \leq x \leq d^*$, $x_u \geq 0$, $x_u = (x_{n+1}, \dots, x_{n+m})$. Для некоторых m и n , например, $m=30$, $n=45$, сгенерированные описанным способом задачи не имели планов (при решении задач первой фазы получались ненулевые искусственные переменные). В задачах с $n \gg m$ почти всегда находились планы, после чего решение задачи продолжалось с помощью алгоритмов § 4, 5. Время решения и число итераций для каждого алгоритма фиксировались с момента окончания первой фазы. Результаты эксперимента при $m=10$, $n=100$ приведены в табл. 12.

Табл. 12

Номер задачи		1	2	3	4	5	6	7	8	9	10
Число итераций	§ 4	34	26	26	40	28	33	52	29	22	40
	§ 5	123	113	125	121	119	128	133	116	132	127
Время решения, с	§ 4	4	4	4	5	3	4	5	4	4	5
	§ 5	8	7	8	7	7	7	7	7	8	7

§ 6. Конечные модификации

Адаптивный метод, изложенный в трех вариантах (см. § 1—3), конечен лишь при выполнении определенных условий невырожденности. Эти условия выполняются почти в каждой задаче (и далеко не просто записать задачу, для которой адаптивный метод приводит к циклу), поэтому на первый взгляд при разработке алгоритмов, нацеленных на решение прикладных задач, не имеет смысла заботиться об исключительных ситуациях, тратить усилия на построение специальных дополнений, обеспечивающих конечность алгоритмов. Подобные аргументы в свое время приводились и в теории симплекс-метода. Однако там уже давно разработаны конечные модификации. Безусловно, они имеют большое теоретическое значение, так как позволяют завершить теорию симплекс-метода. Нельзя принижать и практического значения конечных методов. Хотя в «чистом» виде зацикли-

вание из-за вырожденности в практических расчетах, по утверждению многих специалистов, пока не встречалось, но не исключено, что замедление сходимости, часто наблюдаемое при решении прикладных задач, вызывается ситуациями, близкими к вырожденным. В связи с этим в научной литературе повысился интерес к явлению за-цикливания. Например, в 1977 г. Р. Блэнд предложил новый конечный вариант симплекс-алгоритма. В данном параграфе излагается конечная модификация адаптивного метода, основанная на вырожденных планах, сопровождающих двойственные планы [12, 13]. Предварительно описываются два конечных варианта симплекс-алгоритма.

1. Конечные варианты симплекс-алгоритма. Во введении описан классический симплекс-метод решения канонической задачи

$$c'x \rightarrow \max, Ax=b, x \geq 0, \quad (1)$$

где $A=A(I, J)$, $I=\{1, 2, \dots, m\}$, $J=\{1, 2, \dots, n\}$. Как известно, симплекс-алгоритм конечен, если задача (1) невырождена, т. е. у любого ее базисного плана

$$x = \{x_6, x_H = 0\}, x_6 = x(J_6) = A_6^{-1}b \geq 0, \quad (2)$$

$$A_6 = A(I, J_6), J_6 \subset J, \det A_6 \neq 0, x_H = x(J_H), J_H = J \setminus J_6,$$

базисные переменные x_j , $j \in J_6$, отличны от нуля. В этом случае на каждой симплексной итерации положителен шаг

$$\Theta = \Theta_{i_0} = \min \Theta_i, i \in J_6, \Theta_i = x_i/x_{ij_0}, x_{ij_0} > 0, \quad (3)$$

$$\{x_{ij_0}, i \in J_6\} = A_6^{-1}a_{j_0}, \Delta_{j_0} < 0,$$

что ведет к строгому возрастанию целевой функции.

В вырожденных задачах на некоторых итерациях могут появиться шаги $\Theta=0$. В этих случаях происходит смена базиса, но не изменяется значение целевой функции. Многократное повторение подобной ситуации может привести к базису, который уже встречался на предыдущих итерациях. Иными словами, происходит за-цикливание и метод оказывается бесконечным.

Классический метод предотвращения за-цикливания состоит в дополнении правила (3) вычисления ведущего индекса i_0 . В вырожденных задачах минимум по $i \in J_6$ в (3) может достигаться не на одном i_0 , а на множестве индексов $J_6^1 = \{i_* \in J_6, \Theta_{i_*} = \min x_i/x_{ij_0}, i \in J_6, x_{ij_0} > 0\}$.

Для выбора среди J_6^1 ведущего индекса i_0 строится множество $J_6^2 = \{i_* \in J_6^1, \Theta_{i_*}^1 = \min x_{i1}/x_{i0}, i \in J_6^1\}$.

Если $|J_6^2| > 1$, то строятся множества J_6^3, J_6^4, \dots . Пусть построено множество J_6^k . Тогда $J_6^{k+1} = \{i_* \in J_6^k, \Theta_{i_*}^k = \min x_{ik}/x_{i0}, i \in J_6^k, x_{i0} > 0\}$. Для каждой задачи (1) и любого базисного плана (2) через конечное число шагов будет построено множество J_6^k с единственным элементом, который и берется в качестве ведущего i_0 .

Известно несколько обоснований описанной процедуры избежания заикливания (метод возмущений, лексикографический метод).

Конечный вариант симплекс-алгоритма, предложенный Р. Блэндом, связан с определенными правилами выбора как индекса j_0 ведущего столбца (выше этот индекс удовлетворял только необходимому условию $\Delta_{j_0} < 0$), так и индекса i_0 ведущей строки симплексной таблицы: $j_0 = \min \{j \in J_{\text{н}}: \Delta_j < 0\}$, $i_0 = \min \{i_* \in J_6: \Theta_{i_*} = \min x_{i0}/x_{i0}, x_{i0} > 0, i \in J_6\}$. Покажем, следуя В. С. Глушенкову, что при таком выборе ведущих элементов возникновение цикла невозможно.

Предположим противное: пусть при решении некоторой задачи методом Блэнда возник цикл. Разобьем множество J на непересекающиеся подмножества: $J_{\text{пб}}$ — множество индексов, принадлежащих J_6 на каждой итерации цикла, $J_{\text{пн}}$ — множество индексов, не принадлежащих J_6 ни на одной итерации цикла, $J_{\text{бн}} = J \setminus (J_{\text{пб}} \cup J_{\text{пн}})$. На любой итерации цикла выполнены соотношения $\Delta(J_{\text{пб}}) = 0$, $l(J_{\text{пн}}) = 0$. Пусть $j_* = \max \{j: j \in J_{\text{бн}}\}$, l^* — направление на итерации цикла, когда j_* выходит из базиса, Δ^* — оценки на итерации цикла, когда j_* входит в базис. Из правила Блэнда следует $\Delta^*(J_{\text{бн}} \setminus j_*) \geq 0$, $l^*(J_{\text{бн}} \setminus j_*) \geq 0$, $\Delta_{j_*}^* < 0$, $l_{j_*}^* < 0$. Тогда

$$\Delta^* l^* = \Delta_{j_*}^* l_{j_*}^* + \Delta^*(J_{\text{бн}} \setminus j_*) l^*(J_{\text{бн}} \setminus j_*) + \Delta^*(J_{\text{пб}}) l^*(J_{\text{пб}}) + \Delta^*(J_{\text{пн}}) l^*(J_{\text{пн}}) > 0.$$

С другой стороны, $\Delta^* l^* = (y^* A - c') l^* = -c' l^* < 0$. Полученное противоречие доказывает невозможность цикла в модификации Блэнда.

Приведем конечную модификацию симплекс-алгоритма, предложенную и обоснованную В. С. Глушенковым после ознакомления с материалом данного параграфа.

Рассмотрим задачу (1). Процесс решения задачи на-

чинается с некоторого базисного плана $\{x, J_6\}$. Положим $s = 0$, $J_{\text{бп}}^{(0)} = \emptyset$.

Опишем итерацию метода.

Шаг 1. По аналогии с классическим симплекс-методом строим вектор оценок $\Delta'(J_{\text{н}}) = c'(J_6)A_6^{-1}A(I, J_{\text{н}}) - c'(J_{\text{н}})$. Проверяем достаточное условие оптимальности $\Delta'(J_{\text{н}}) \geq 0$. Если оно выполнено, то решение задачи закончено. При $s = 0$ оптимальным планом задачи будет x , при $s > 0$ — план \bar{x} , $\bar{x}(J_{\text{бп}}^{(1)}) = x(J_{\text{бп}}^{(1)})$, $\bar{x}(J \setminus J_{\text{бп}}^{(1)}) = 0$. Если $\exists j \in J_{\text{н}}, \Delta_j < 0$, то переходим к следующему шагу.

Шаг 2. Выбираем произвольно $j_0 \in J_{\text{н}}, \Delta_{j_0} < 0$, и строим направление l улучшения плана x : $l_{j_0} = 1$, $l(J_{\text{н}} \setminus j_0) = 0$, $l(J_6) = -A_6^{-1}A(I, j_0)$.

Шаг 3. Вычисляем шаг Θ^0 вдоль l : $\Theta^0 = \min \{\Theta_j, j \in J_6 \setminus J_{\text{бп}}^{(s)}\}$, $\Theta_j = -x_j/l_j$, если $l_j < 0$, $\Theta_j = \infty$, если $l_j \geq 0$. При $\Theta^0 = 0$ переходим к шагу 5; при $\Theta^0 = \infty$ — к шагу 6. В случае $0 < \Theta^0 < \infty$ переход к следующему шагу.

Шаг 4. Заменяем базис: $J_6 = (J_6 \setminus j_*) \cup j_0$, где j_* — индекс, на котором достигается минимум при вычислении Θ^0 . Вычисляем новый план \bar{x} , $\bar{x}(J_{\text{бп}}^{(s)}) = x(J_{\text{бп}}^{(s)})$, $\bar{x}(J \setminus J_{\text{бп}}^{(s)}) = x(J \setminus J_{\text{бп}}^{(s)}) + \Theta^0 l(J \setminus J_{\text{бп}}^{(s)})$. Переход к шагу 1.

Шаг 5. Положим $J_{\text{бп}}^{(s+1)} = \{j \in J_6, x_j > 0\}$. Хотя бы одна компонента x_j , $j \in J_6 \setminus J_{\text{бп}}^{(s+1)}$, увеличивается на положительное число. Увеличиваем s на единицу и осуществляем переход к шагу 3.

Шаг 6. Если $s = 0$, то целевая функция исходной задачи не ограничена на множестве ее планов. Решение закончено. Если $s \geq 1$, то полагаем $x(J_6 \setminus J_{\text{бп}}^{(s)}) = 0$, уменьшаем s на единицу и осуществляем переход к шагу 3.

Теорема 1. При любых A, b, c и начальном базисном плане $\{x, J_6\}$ решение задачи (1) заканчивается через конечное число итераций.

Итерация заканчивается при переходе к шагу с меньшим номером, построении оптимального плана либо обнаружении неограниченности сверху значений целевой функции на множестве планов. Для доказательства конечности алгоритма достаточно показать конечность числа итераций, заканчивающихся на шагах 4, 5 или 6. Такие итерации назовем соответственно итерациями первого, второго и третьего типов.

Доказательству теоремы предпошлем несколько лемм.

Лемма 1. Величина s в процессе работы метода ограничена.

Доказательство. При замене базиса введенная компонента \bar{x}_{j_0} всегда будет положительной. Следовательно, общее число положительных компонент не уменьшается. При увеличении s число ненулевых компонент x_j , $j \in J_{\text{оп}}$, увеличивается. Поэтому справедливо строгое вложение $J_{\text{оп}}^{(0)} \subset J_{\text{оп}}^{(1)} \subset \dots \subset J_{\text{оп}}^{(s)} \forall s$. Но $J_{\text{оп}}^{(s)} \subset J_6 \forall s$, поэтому $s \leq m$. Лемма доказана.

Лемма 2. За итерацией второго типа не может следовать итерация третьего типа, а за итерацией третьего типа не может следовать итерация второго типа.

Доказательство. Итерация второго типа выполняется при $\Theta^0 = 0$, т. е. $\exists j \in J_6 \setminus J_{\text{оп}}^{(s)}, x_j = 0, l_j < 0$. На следующей итерации $j \notin J_{\text{оп}}^{(s+1)}, x_j \geq 0$, а значение l_j остается прежним. Следовательно, $\Theta_j < \infty$, что влечет за собой $\Theta^0 < \infty$. При этом переход к шагу 6 (итерация третьего типа) невозможен. Первая часть леммы доказана.

Итерация третьего типа выполняется при $\Theta_j = \infty$. Это означает, что $\forall j \in J_6 \setminus J_{\text{оп}}^{(s)}, l_j \geq 0$. Справедливо неравенство $x_j > 0 \forall j \in J_{\text{оп}}^{(s)}$. Значит, на следующей итерации $\Theta^0 > 0$ и переход к шагу 5 (итерация второго типа) невозможен. Лемма доказана.

Лемма 3. Число итераций первого типа конечно.

Доказательство. Предположим противное: пусть найдется задача, при решении которой бесконечно число итераций первого типа. Величина s ограничена. Следовательно, найдется хотя бы одно s_0 , при котором число итераций первого типа бесконечно. Без ограничения общности можно считать, что s_0 — минимальное из таких чисел. Покажем, что, начиная с некоторой итерации, все время $s \geq s_0$. Действительно, в противном случае при $s = s_0$ бесконечное число раз выполняется итерация третьего типа. В силу леммы 2 после этого каждый раз при $s < s_0$ должна выполняться итерация первого типа. Следовательно, бесконечное число итераций первого типа должно выполняться при $s < s_0$, что противоречит предположению о минимальности s_0 .

Рассмотрим задачу

$$c'z \rightarrow \max, \quad Az = b^{(s_0)}, \quad z(J \setminus J_{\text{оп}}^{(s_0)}) \geq 0, \quad (4)$$

где $\sigma(s_0)$ определяется по плану x сразу после перехода $s_0 - 1 \rightarrow s_0$. С каждой итерацией первого типа на уровне

s_0 осуществляется переход к новому базисному плану задачи (4) со строгим увеличением целевой функции. Поскольку число базисных планов у задачи (4) конечно, то и число итераций первого типа на уровне s_0 конечно. Получено противоречие с предположением о бесконечном числе итераций первого типа. Лемма доказана.

Лемма 4. Число итераций третьего типа конечно.

Доказательство. Согласно лемме 2 за каждой итерацией третьего типа следует итерация первого или третьего типов. Предположим, что число итераций третьего типа бесконечно. При этом возможны два случая: за каждой итерацией третьего типа следует конечное число (возможно 0) итераций третьего типа, а затем следует итерация первого типа; начиная с некоторой итерации, все последующие принадлежат к третьему типу. В первом случае в задаче бесконечно число итераций первого типа, что невозможно в силу леммы 3. Во втором случае s неограниченно уменьшается, что невозможно в силу $s \geq 0$. Следовательно, число итераций третьего типа конечно. Лемма доказана.

Лемма 5. Число итераций второго типа конечно.

Доказательство. Согласно леммам 3 и 4 число итераций первого и третьего типов конечно. Следовательно, если число итераций второго типа бесконечно, то, начиная с некоторой итерации, все последующие должны принадлежать ко второму типу. При этом s должно неограниченно увеличиваться, что противоречит лемме 1. Лемма доказана.

Доказательство теоремы. С одной стороны, итерации могут закончиться либо построением оптимального плана задачи, либо обнаружением неограниченности значений целевой функции, с другой — согласно леммам 3—5 число итераций каждого из возможных типов конечно. Следовательно, через конечное число итераций задача (1) при любом плане (2) будет решена. Теорема доказана.

Для иллюстрации метода рассмотрим пример, на котором заикливаются стандартный симплекс-метод:

$$\begin{aligned} 0,75x_5 - 20x_6 + 0,5x_7 - 6x_8 &\rightarrow \max, \\ x_1 + 0,25x_5 - 8x_6 - x_7 + 8x_8 &= 0, \\ x_2 + 0,5x_5 - 12x_6 - 0,5x_7 + 3x_8 &= 0, \\ x_3 - 0,5x_5 + 0,25x_6 + 0,25x_7 + 0,25x_8 &= 0, \\ x_4 + x_7 &= 1, \\ x_i &\geq 0, \quad i = \overline{1, 8}. \end{aligned}$$

Начальный план $x^H = \{0, 0, 0, 1, 0, 0, 0, 0\}$; начальный базис $J^H = \{1, 2, 3, 4\}$. Приведем последовательность симплексных таблиц, соответствующих итерациям метода. Исходной является табл. 13. В первый столбец зане-

Табл. 13

$J \backslash J_6$	1	2	3	4	5	6	7	8	s=0		s=1	
									x_6	Θ_i	x_6	Θ_i
1	1	0	0	0	0,25	—8	—1	9	0	0	1	4
2	0	1	0	0	0,5	—12	—0,5	3	0	0	1	2
3	0	0	1	0	0,5	0,25	0,25	0,25	0	∞	1	∞
4	0	0	0	1	0	0	1	0	1	∞	1*	
Δ	0	0	0	0	—0,75	20	—0,5	6				

сен список базисных индексов, во втором — восьмом столбцах содержатся коэффициенты разложения столбцов матрицы A по базисным векторам и компоненты вектора оценок, в последующих — компоненты базисного плана и числа Θ_i для различных s .

На первой итерации: $j_0=5$, $\Theta_1=\Theta_2=0$, $\Theta_3=\Theta_4=\infty$, $\Theta^0=\min \Theta_i=0$. Следовательно, эта итерация относится ко второму типу. Базис заменяться не будет, к таблице добавляется новый столбец x_6 при $s=1$, $J_{6n}^{(1)}=\{4\}$. Полагаем $x^{(1)}(J_5 \setminus J_{6n}^{(1)}) = x^{(0)} + \{1, 1, 1\}$, $x_4^{(1)} = x_4^{(0)}$. Индексы $j \in J_{6n}^s \setminus J_{6n}^{s-1}$ отмечаем звездочкой в правом верхнем углу клетки в столбце s , x_6 . Увеличиваем s на единицу.

На второй итерации: $\Theta_1=4$, $\Theta_2=2$, $\Theta_3=\infty$, $j_*=2$, $\Theta^0=2$ (см. столбец x_6 , $s=1$). Пересчитываем таблицу по правилу прямоугольника с ведущим элементом x_{25} и заменяем множество J_6 . Получаем симплексную таблицу (табл. 14).

На третьей итерации: $j_0=7$, $\Theta_1=\Theta_5=\Theta_3=\infty=\Theta^0$. Следовательно, эта итерация относится к третьему типу. Возвращаемся к $s=0$.

На четвертой итерации: $\Theta_1=\Theta_5=\Theta_3=\infty$, $\Theta_4=1$, $\Theta^0=1$, $j_*=4$. Заменяем базис, пересчитываем таблицу, получаем следующую симплекс-таблицу (табл. 15). Она удовлетво-

Табл. 14

$J \backslash J_6$	1	2	3	4	5	6	7	8	s=0		s=1	
									x_6	Θ_i	x_6	Θ_i
1	1	-0,5	0	0	0	-2	-0,75	7,5	0	∞	0,5	∞
5	0	2	0	0	1	-24	-1	6	0	∞	3	∞
3	0	1	1	0	0	-11,75	-0,25	3,25	0	∞	2	∞
4	0	0	0	1	0	0	1	0	1	1	1*	
Δ	0	1,5	0	0	0	2	-1,25	11,5				

Табл. 15

$J \backslash J_6$	1	2	3	4	5	6	7	8	s=0	
									x_6	Θ_i
1	1	-0,5	0	0,75	0	-2	0	7,5	0,75	
5	0	2	0	1	1	-24	0	6	1	
3	0	1	1	0,25	0	-11,75	0	3,25	0,25	
7	0	0	0	1	0	0	1	0	1	
Δ	0	1,5	0	1,25	0	2	0	10,5		

ряет критерию оптимальности. Оптимальный план $x^0 = \{0,75; 0; 0,25; 0; 1; 0; 1; 0\}$. Задача решена.

В заключение приведем последовательность базисов при решении данного примера стандартным симплекс-методом, методом возмущений, методом Блэнда и предлагаемым методом (s) (табл. 16).

2. Конечный вариант адаптивного метода. Рассмотрим общую задачу линейного программирования с двухсторонними прямыми ограничениями

$$c'x \rightarrow \max, Ax=b, d_* \leq x \leq d^*. \quad (5)$$

В § 1 описан адаптивный метод решения задачи (5), который состоит в последовательном преобразовании опор-

Номер базиса	Симплекс-метод	Метод возмущений	Метод Блэнда	Метод (s)
0	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4
1	5, 2, 3, 4	1, 5, 3, 4	5, 2, 3, 4 5, 6, 3, 4	1, 5, 3, 4
2	5, 6, 3, 4	1, 5, 3, 7	5, 6, 7, 4	1, 5, 3, 7
3	7, 6, 3, 4		5, 1, 7, 4	
4	7, 8, 3, 4		5, 1, 7, 2 5, 1, 7, 3	
5	1, 8, 3, 4			
6	1, 2, 3, 4			

ных планов $\{x, A_{\text{оп}}\}$ и основан на принципе уменьшения оценки субоптимальности $\beta(x, A_{\text{оп}}) = \Delta'(J_{\text{н}+})(x(J_{\text{н}+}) - d_*(J_{\text{н}+})) - \Delta'(J_{\text{н}-})(x(J_{\text{н}-}) - d^*(J_{\text{н}-}))$, где

$$\Delta = A'u - c, \quad u' = c'(J_{\text{оп}}) A_{\text{оп}}^{-1}, \quad J_{\text{н}} = J \setminus J_{\text{оп}}, \quad (6)$$

$$J_{\text{н}+} = \{j \in J_{\text{н}} : \Delta_j > 0\}, \quad J_{\text{н}-} = \{j \in J_{\text{н}} : \Delta_j < 0\}.$$

В классическом симплекс-методе (см. введение) мера успешности итерации определяется приращением целевой функции $c'x$. Эта мера зависит только от текущего базисного плана x и поэтому в симплекс-методе появляются дополнительные итерации всякий раз, когда не увеличивается $c'x$. Подобные явления наблюдаются [26] почти во всех прикладных задачах и связаны с вырожденностью планов. Мера успешности итерации адаптивного метода в силу указанного выше принципа определяется уменьшением оценки субоптимальности. Последняя, как известно (см. § 1), зависит как от плана, так и от опоры: $\beta(x, A_{\text{оп}}) = \beta(x) + \beta(A_{\text{оп}})$. Поэтому опорный план улучшается адаптивным методом в отличие от симплекс-метода и за счет замены плана, и за счет изменения опоры.

Цель настоящего и следующих пунктов — модифицировать метод § 1 на случай, когда он на опорном плане не приводит к уменьшению оценки субоптимальности. Две модификации симплекс-метода для вырожденных базисных планов, описанные в п. 1, направлены только на предупреждение зацикливания и не учитывают непосредственно исходной цели оптимизации в рассматриваемой задаче. Изложенный далее подход основан на том убеждении, что вырождение — один из признаков внутренней

специфики задачи. Поэтому следует не «бояться» вырожденных планов, а, напротив, приспособливать к ним метод оптимизации с учетом дополнительной информации, доступной на итерациях.

Пусть $\{x, A_{\text{оп}}\}$ — опорный план, оценка субоптимальности которого превосходит заданную точность приближения к оптимальному значению целевой функции: $\beta(x, A_{\text{оп}}) > \varepsilon$. Построим план $x^{(1)} = x + \Theta^{(1)} l^{(1)}$, где $l^{(1)} = l^{(1)}(J)$:

$$\begin{aligned} l^{(1)}(J_{\text{н}+}) &= d_*(J_{\text{н}+}) - x(J_{\text{н}+}), \quad l^{(1)}(J_{\text{н}-}) = d^*(J_{\text{н}-}) - x(J_{\text{н}-}), \\ l(J_{\text{н}0}) &= 0, \quad J_{\text{н}0} = J_{\text{н}} \setminus (J_{\text{н}+} \cup J_{\text{н}-}), \quad l^{(1)}(J_{\text{оп}}) = -A_{\text{оп}}^{-1} A(I, J_{\text{н}}) \times \\ &\times l(J_{\text{н}}); \quad \Theta^{(1)} = \min \{1, \Theta_{j_1}\}, \quad \Theta_{j_1} = \min \Theta_j, \quad j \in J_{\text{оп}}; \quad (7) \\ \Theta_j &= (d_{*j} - x_j) / l_j^{(1)} \quad \text{при } l_j^{(1)} < 0; \quad \Theta_j = (d_j^* - x_j) / l_j^{(1)} \\ &\quad \text{при } l_j^{(1)} > 0; \quad \Theta_j = \infty \quad \text{при } l_j^{(1)} = 0. \end{aligned}$$

Согласно § 1 имеем $\beta(x^{(1)}, A_{\text{оп}}) = (1 - \Theta^{(1)}) \beta(x, A_{\text{оп}})$.

Если $\beta(x^{(1)}, A_{\text{оп}}) \leq \varepsilon$, то решение задачи (5) прекращается на ε -оптимальном плане $x^{(1)}$. Эта часть итерации совпадает с аналогичной из § 1. Предлагаемая модификация относится к случаю $\beta(x^{(1)}, A_{\text{оп}}) > \varepsilon$.

Зададим параметр алгоритма $\eta > 0$, положим $s = 1$, $K = J_{\text{оп}}$, $J_{\text{н}0}^{(0)} = J_{\text{н}+} \cup J_{\text{н}-}$, $J_{\text{н}0}^{(0)} = J_{\text{н}0}$, $z_0(J_{\text{н}0}^{(0)}) = \Delta(J_{\text{н}0}^{(0)})$, $j_0 = \emptyset$, $\mu_1 = 1$ при $x_{j_1}^{(1)} = d_{*j_1}$; $\mu_1 = -1$ при $x_{j_1}^{(1)} = d_{j_1}^*$

и по правилам следующего пункта приступим к корректировке направления $l^{(1)}$, считая $j_1 \notin \{j_0\}$. Корректировка закончится построением или ε -оптимального плана x^ε (задача (1) решена), или опорного плана $\{\bar{x}, \bar{A}_{\text{оп}}\}$ с $\beta(\bar{x}, \bar{A}_{\text{оп}}) < \beta(x, A_{\text{оп}})$ (начало следующей итерации).

3. Процедура корректировки направления. Пусть при выбранном $\eta > 0$ для натурального числа s известны:

$$\begin{aligned} K, \quad K_{\text{н}} = J \setminus K, \quad J_{\text{н}0}^{(k)}, \quad k = \overline{0, s-1}, \quad J_{\text{н}0}^{(s-1)}; \quad z_k(J_{\text{н}0}^{(k)}), \quad k = \overline{0, s-1}; \\ j_0, \quad j_k, \quad \mu_k, \quad k = \overline{1, s}, \quad \Theta^{(s)}; \quad x^{(s)}; \quad l^{(s)}; \quad \beta(x^{(s)}, A(I, K)); \\ B(K, I) = [A(I, K)]^{-1}. \end{aligned} \quad (8)$$

Для $s = 1$ эта информация приведена в п. 2. Используем ее для корректировки направления $l^{(s)}$. Рассмотрим

отдельно две возможности: 1) $j_s \notin \{j_0, \dots, j_{s-1}\}$; 2) $j_s \in \{j_0, \dots, j_{s-1}\}$. В случае 1) положим

$$z_s(K_H) = \mu_s B(j_s, I) A(I, K_H) \quad (9)$$

и рассмотрим два подслучая: 1а) $J_{H0}^{(s-1)} = \emptyset$, 1б) $J_{H0}^{(s-1)} \neq \emptyset$.

В случае 1а) корректировка начинается с замены в

(8) множества $K: K \rightarrow \tilde{K}$. Для этого сначала подсчитаем

$$\sigma(j) = \begin{cases} -z_k(j)/z_s(j), & \text{если } z_k(j) z_s(j) < 0, \\ \infty & \text{в остальных случаях,} \end{cases} \quad (10)$$

где $j \in J_{H0}^{(k)}$, $k = s-1, \dots, 0$. В каждой k -й группе $\{\sigma(j), j \in J_{H0}^{(k)}\}$ числа $\sigma(j) < \infty$ упорядочим по возрастанию: $\sigma(j_{1,k}) \leq \sigma(j_{2,k}) \leq \dots \leq \sigma(j_{p_k,k})$. Далее положим

$$\alpha(0, s-1) = \mu_s (1 - \Theta^{(s)}) l_{j_s}^{(s)} \quad (11)$$

и введем рекуррентно числа $\alpha(i, k)$, $i = \overline{1, p_k}$, $k = s-1, s-2, \dots, 0$:

$$\alpha(i, k) = \alpha(i-1, k) + |z_s(j_{i,k})| (d_{j_{i,k}}^* - \overline{d_{*j_{i,k}}}),$$

считая $\alpha(0, k) = \alpha(p_{k+1}, k+1)$.

По построению, числа $\alpha(i, k)$ удовлетворяют неравенствам

$$\begin{aligned} \alpha(0, s-1) &< 0, \quad \alpha(p_0, 0) \geq 0; \\ \alpha(0, s-1) &\leq \alpha(1, s-1) \leq \alpha(2, s-1) \leq \dots \leq \alpha(p_{s-1}, s-1) \leq \\ &\leq \alpha(1, s-2) \leq \alpha(2, s-2) \leq \dots \leq \alpha(p_{s-2}, s-2) \leq \dots \\ &\leq \alpha(1, 0) \leq \alpha(2, 0) \leq \dots \leq \alpha(p_0, 0). \end{aligned}$$

Поэтому найдутся такие индексы v, s_* , что $\alpha(v, s_*) \geq 0$, $\alpha(v-1, s_*) < 0$ (при $v=1$ считаем, что $\alpha(0, s_*) = \alpha(p_{s_*+1}, s_*+1)$). После этого множество K в (8) заменяем на множество $\tilde{K} = (K \setminus j_s) \cup j_{v, s_*}$. Если при этом $s_* = 0$, $v > 1$, то корректировку заканчиваем без изменения $l^{(s)}$ построением нового опорного плана $\{\bar{x}, \bar{A}_{оп}\}$, $\bar{A}_{оп} = A(I, \bar{J}_{оп})$, $\bar{J}_{оп} = \tilde{K}$.

Согласно § 1 при замене опоры $A_{оп} \rightarrow \bar{A}_{оп}$ значение двойственной целевой функции на соответствующих сопровождающих планах уменьшается на величину $-\sum_{k=1}^{v-1} \alpha(k, 0) (\sigma(j_{k,0}) - \sigma(j_{k-1,0}))$, где $\sigma(j_{0,0}) = 0$. Следова-

тельно, для $\{\bar{x}, \bar{A}_{оп}\}$ выполняется неравенство $\beta(\bar{x}, \bar{A}_{оп}) < \beta(x, A_{оп})$, ибо

$$\beta(\bar{A}_{оп}) - \beta(A_{оп}) = \sum_{k=1}^{v-1} \alpha(k, 0) (\sigma(j_{k,0}) - \sigma(j_{k-1,0})) < 0. \quad (12)$$

При $s_* > 0$ ($s_* = 0$, $v = 1$) корректировку $l^{(s)}$ продолжаем, заменив данные (8) на следующие *):

$$s_*; \quad \tilde{K}, \quad \tilde{K}_H = J \setminus \tilde{K}, \quad \tilde{J}_{H0}^{(k)} = J_{H0}^{(k)}, \quad k = \overline{0, s_* - 1}; \quad \tilde{J}_{H0}^{(s_*-1)} = \\ = \tilde{K}_H \setminus \bigcup_{k=0}^{s_*-1} \tilde{J}_{H0}^{(k)}; \quad \tilde{z}_k(\tilde{J}_{H0}^{(k)}) = z_k(J_{H0}^{(k)}), \quad k = \overline{0, s_* - 1}; \quad \tilde{j}_0 = \emptyset,$$

$$\tilde{j}_k = j_k, \quad \tilde{\mu}_k = \mu_k, \quad k = \overline{1, s_*}, \quad \tilde{\Theta}^{(s_*)} = \Theta^{(s)}; \quad (13)$$

$$\tilde{x}^{(s_*)} = x^{(s)}; \quad \tilde{l}^{(s_*)} = l^{(s)}; \quad \beta(\tilde{x}^{(s_*)}, A(I, \tilde{K})) = \beta(x^{(s)},$$

$$A(I, K)); \quad \tilde{B}(\tilde{K}, I) = [A(I, \tilde{K})]^{-1}.$$

Рассмотрим случай 1б): $J_{H0}^{(s-1)} \neq \emptyset$. Положим $J_{H0}^{(s)} = \{j \in J_{H0}^{(s-1)} : z_s(j) = 0\}$, $J_{H+}^{(s)} = \{j \in J_{H0}^{(s-1)} : z_s(j) > 0\}$, $J_{H-}^{(s)} = \{j \in J_{H0}^{(s-1)} : z_s(j) < 0\}$, $J_{H0}^{(s)} = J_{H+}^{(s)} \cup J_{H-}^{(s)}$, построим вектор $l^{(s+1)} = l^{(s+1)}(J)$:

$$l^{(s+1)}(K_H \setminus J_{H0}^{(s)}) = (1 - \Theta^{(s)}) l^{(s)}(K_H \setminus J_{H0}^{(s)}); \\ l^{(s+1)}(J_{H+}^{(s)}) = d_*(J_{H+}^{(s)}) - x^{(s)}(J_{H+}^{(s)}); \quad l^{(s+1)}(J_{H-}^{(s)}) = d^*(J_{H-}^{(s)}) - \\ - x^{(s)}(J_{H-}^{(s)}); \quad l^{(s+1)}(K) = -B(K, I) A(I, K_H) l^{(s+1)}(K_H)$$

и найдем число

$$\alpha(0, s-1) = \mu_s l_{j_s}^{(s+1)}. \quad (15)$$

В зависимости от параметра $\eta > 0$ может реализоваться одна из следующих возможностей:

$$1б_1) \quad \alpha(0, s-1) < 0;$$

$$1б_2) \quad \alpha(0, s-1) \geq 0, \quad \gamma = (d_{j_s}^* - d_{*j_s}) \beta(x^{(s)}, A(I, K)) / |l_{j_s}^{(s+1)}| \geq \\ \geq \min \{\beta(x^{(s)}, A(I, K)), \eta\};$$

$$1б_3) \quad \alpha(0, s-1) \geq 0, \quad \gamma < \min \{\beta(x^{(s)}, A(I, K)), \eta\}.$$

) Вся информация, связанная с номерами $k = \overline{s_ + 1, s}$, во внимание не принимается.

В случае 1б₁) по правилам случая 1а) находим индексы v , s_* и заменяем множество K на \tilde{K} , используя вместо (11) выражение (15). При $s_* = 0$, $v > 1$ корректировка заканчивается опять без изменения $l^{(s)}$ построением нового опорного плана $\{\bar{x}, \bar{A}_{\text{оп}}\}$, $\bar{x} = x^{(s)}$, $\bar{A}_{\text{оп}} = A(I, \bar{J}_{\text{оп}})$, $\bar{J}_{\text{оп}} = \tilde{K}$, для которого будет выполняться неравенство $\beta(\bar{x}, \bar{A}_{\text{оп}}) < \beta(x, A_{\text{оп}})$ в силу (12). При $s_* > 0$ ($s_* = 0$, $v = 1$) в случае 1б₁) корректировку $l^{(s)}$ продолжим, заменив исходные данные (8) на (13).

Пусть реализовался случай 1б₂). Вектор $l^{(s)}$ заменяем на вектор $l^{(s+1)}$ из (14) (корректируем направление $l^{(s)}$) и строим новый план $x^{(s+1)} = x^{(s)} + \Theta^{(s+1)} l^{(s+1)}$, где $\Theta^{(s+1)}$ — максимально допустимый шаг вдоль $l^{(s+1)}$: $\Theta^{(s+1)} = \min\{1, \Theta_{j_{s+1}}\}$, $\Theta_{j_{s+1}} = \min \Theta_j$, $j \in K$, $\Theta_j = (d_{*j} - x_j^{(s)}) / l_j^{(s+1)}$ при $l_j^{(s+1)} < 0$; $\Theta_j = (d_j^* - x_j^{(s)}) / l_j^{(s+1)}$ при $l_j^{(s+1)} > 0$; $\Theta_j = \infty$ при $l_j^{(s+1)} = 0$. Для $\{x^{(s+1)}, A(I, K)\}$ имеем $\beta(x^{(s+1)}, A(I, K)) = (1 - \Theta^{(s+1)}) \beta(x^{(s)}, A(I, K))$. Если $\beta(x^{(s+1)}, A(I, K)) \leq \varepsilon$, то корректировка заканчивается на ε -оптимальном плане $x^* = x^{(s+1)}$. При $\beta(x^{(s+1)}, A(I, K)) > \varepsilon$ корректировка продолжается с исходными данными: $s+1$; K , K_H , $J_{H0}^{(k)}$, $k = \overline{0, s}$; $J_{H0}^{(s)}$; $z_k(J_{H0}^{(k)})$, $k = \overline{0, s}$; j_0, j_k, μ_k , $k = \overline{1, s+1}$, $\Theta^{(s+1)}$; $x^{(s+1)}$; $l^{(s+1)}$; $\beta(x^{(s+1)}, A(I, K))$; $B(K, I)$, где $\mu_{s+1} = 1$ при $x_{i_{s+1}}^{(s+1)} = d_{*j_{s+1}}$; $\mu_{s+1} = -1$ при $x_{i_{s+1}}^{(s+1)} = d_{j_{s+1}}^*$.

Рассмотрим случай 1б₃). В исходных данных (8) заменяем: множество K на $\tilde{K} = (K \setminus j_s) \cup j_*$, где j_* — любой элемент из $J_{H0}^{(s)} \neq \emptyset$, множества K_H , $J_{H0}^{(s)}$, $J_{H0}^{(s)}$ на множества $\tilde{K}_H = J \setminus \tilde{K}$, $\tilde{J}_{H0}^{(s)} = j_s$, $\tilde{J}_{H0}^{(s)} = J_{H0}^{(s-1)} \setminus j_*$; вектор $z_s(J_{H0}^{(s)})$ на вектор $\tilde{z}_s(\tilde{J}_{H0}^{(s)}) = -\mu_s$. Далее строим направление $l^{(s+1)}$ по формулам (14). После этого обязательно окажемся в условиях случая 1б₂), по правилам которого и продолжаем корректировку.

Рассмотрим, наконец, случай 2): $j_s \in \{j_0, \dots, j_{s-1}\}$. Он может реализоваться только при $s > 1$ (см. п. 2). Положим $\bar{x} = x^{(s)}$, $\bar{A}_{\text{оп}} = A(I, \bar{J}_{\text{оп}})$, $\bar{J}_{\text{оп}} = K$. Корректировку заканчиваем построением нового опорного плана $\{\bar{x}, \bar{A}_{\text{оп}}\}$, для которого $\beta(\bar{x}, \bar{A}_{\text{оп}}) \leq \beta(x, A_{\text{оп}}) - \eta$.

Докажем последнее утверждение. Пусть $j_s = j_s^-, \bar{s} < s$. Тогда согласно проведенным построениям в силу того, что $x_j^{(s)} = x_j^{(\bar{s})} + \Theta l_j^{(\bar{s}+1)}, j \in \bigcup_{k=0}^{\bar{s}} J_{H0}^{(k)}$, где $\Theta = (d_{j_s^*}^* - d_{*j_s^-}) / |l_{j_s^-}^{(\bar{s}+1)}|$,

имеем $c' x^{(s)} = c' x^{(\bar{s})} + \Theta \beta(x^{(\bar{s})}, A(I, K)) > c' x^{(\bar{s})} + \eta > c' x + \eta$. Из разложения (9) § 1 с учетом последнего неравенства следует справедливость утверждения.

Лемма 6. При любых исходных данных процедура корректировок направления состоит из конечного числа шагов.

Доказательство. Данным (8) поставим в соответствие набор вспомогательных задач

$$\begin{aligned} \mu_k x_{j_k} &\rightarrow \max, \\ A(I, K \cup J_{H0}^{(k-1)}) x(K \cup J_{H0}^{(k-1)}) &= b - A(I, J_*^{(k-1)}) \times \\ &\times (x(J_*^{(k-1)}) + l(J_*^{(k-1)})); \\ d_{*j} &\leq x_j \leq d_j^*, j \in J_{H0}^{(k-1)} \cup K \setminus \{j_0, \dots, j_{k-1}\}, \\ J_*^{(k-1)} &= J \setminus (K \cup J_{H0}^{(k-1)}) \end{aligned} \quad (16)$$

с двойственными базисными копланами $\Delta_k(K \cup J_{H0}^{(k-1)})$:

$$\Delta_k(K) = 0, \Delta_k(J_{H0}^{(k-1)}) = z_k(J_{H0}^{(k-1)}), k = \overline{1, s}. \quad (17)$$

Процедура корректировок, описанная выше, осуществляется двумя способами: без замены множества K (при этом увеличиваются номера $s: s \rightarrow s+1$ (см. случай 1б₂)) и с заменой множества $K: K \rightarrow \tilde{K}$. При этом номер s либо уменьшается: $s \rightarrow s_* < s$ (см. случай 1а), 1б₁), либо остается прежним *): $s \rightarrow s_* = s$ (см. случай 1б₃)).

Проследим, как преобразования данных (8), соответствующие двум способам корректировок, сказываются на наборе вспомогательных задач (16) и их двойственных базисных копланах (17).

Если корректировка идет первым способом, то исходные данные (8) заменяются на данные (13). Очевидно, что задачи (16) с номерами $k = \overline{1, s}$ и их двойственные базисные копланы (17), соответствующие данным (8) и (13), совпадают. Следовательно, при первом способе

) В дальнейшем не будем отдельно рассматривать случаи $s \rightarrow s_ < s$ и $s \rightarrow s$, считая, что $s \rightarrow s_* \leq s$.

корректировки вспомогательные задачи не решаются, а только увеличивается их число.

Легко проверить, что и при втором способе корректировки вспомогательные задачи (16) с номерами $k=\overline{1, s_*}$ совпадают для новых и старых данных. Однако в этом случае двойственный базисный коплан s_* -й задачи, соответствующей новым данным, будет лучше по значению двойственной целевой функции s_* -й задачи (16), чем коплан, соответствующий старым данным.

Поскольку $s < |J|$, то первый способ не может повторяться подряд бесконечное число раз. Через конечное число корректировок обязательно встретится хотя бы одна корректировка, осуществляемая вторым способом, и при этом произойдет улучшение двойственного базисного плана в s_* -й задаче (16). Поскольку число двойственных базисных копланов в каждой задаче (16) конечно, то и второй способ не может повторяться бесконечно. Таким образом, через конечное число итераций реализуется ситуация, когда надо закончить процедуру корректировок (см. случай $s_*=0$, $v>1$, либо случай 2)). Лемма доказана.

Теорема 2. При любых $A, b, c, d_*, d^*, m < \infty, n < \infty, \varepsilon \geq 0, \eta > 0$ для каждого начального опорного плана $\{x, A_{\text{оп}}\}$ через конечное число итераций будет построен ε -оптимальный план задачи (5).

Доказательство. Согласно описанному выше алгоритму каждая итерация включает в себя процедуру корректировок направления, конец которой совпадает с концом итерации. Из сказанного выше следует, что процедура корректировок заканчивается в следующих ситуациях:

- 1) если в случаях 1а), 1б₁) выполняются условия $s_*=0, v>1$;
- 2) если в случае 1б₂) имеет место неравенство $\beta(x^{(s+1)}, A(I, K)) \leq \varepsilon$;
- 3) если реализовался случай 2).

Если процедура корректировок закончилась ситуацией 1), то двойственный базисный план задачи (5), сопровождающий опору $\bar{A}_{\text{оп}}$, лучше двойственного базисного плана, сопровождающего опору $A_{\text{оп}}$. В силу того, что число двойственных базисных планов задачи (5) конечно, ситуация 1) может повториться лишь конечное число раз.

Ситуация 2) может встретиться только один раз, ибо она означает, что исходная задача решена.

Рассмотрим ситуацию 3). Выше показано, что в этом случае целевая функция задачи (5) увеличивается не менее чем на $\eta > 0$. Множество планов задачи (5) компактно, следовательно, целевая функция задачи (5) ограничена сверху на множестве планов: $c'x \leq M$ для $\forall x \in \{x : Ax = b, d_* \leq x \leq d^*\}$. Значит, ситуация 3) может повториться не более чем $[M/\eta]$ раз.

Из сказанного следует, что для решения задачи (5) необходимо конечное число итераций (процедур корректировок). Теорема доказана.

4. Конечный двойственный точный алгоритм линейного программирования. Рассмотрим задачу

$$f(\delta) = b'y - d'_*v + d'^*w \rightarrow \min, \quad A'y - v + w = c, \quad (18) \\ v \geq 0, \quad w \geq 0,$$

которая является двойственной к задаче (5). В линейном программировании наряду с прямыми методами большую роль играют двойственные методы. Известны дополнительные правила к классическому двойственному симплекс-методу, гарантирующие конечность этого метода для любой задачи (5). Однако эти правила выхода из цикла не связаны непосредственно с задачей оптимизации (5). Построим конечный двойственный алгоритм, близкий в идейном отношении к прямому алгоритму, изложенному в пп. 2, 3.

Пусть $\{\delta, A_{оп}\}$, $A_{оп} = A(I, J_{оп})$, $J_{оп} \subset J$, — начальный опорный коплан [5] задачи (5). Обозначим: $J_{н} = J \setminus J_{оп}$, $J_{н0}^{(0)} = \{j \in J_{н} : \delta_j = 0\}$, $J_{н+}^{(0)} = \{i \in J_{н} : \delta_j > 0\}$, $J_{н-}^{(0)} = \{j \in J_{н} : \delta_j < 0\}$, $J_{н0}^{(0)} = J_{н+}^{(0)} \cup J_{н-}^{(0)}$. Построим соответствующий псевдоплан $\kappa^{(1)} = \kappa^{(1)}(J) : \kappa^{(1)}(J_{н+}^{(0)} \cup J_{н0}^{(0)}) = d_*(J_{н+}^{(0)} \cup J_{н0}^{(0)})$, $\kappa^{(1)}(J_{н-}^{(0)}) = d^*(J_{н-}^{(0)})$, $\kappa^{(1)}(J_{оп}) = A_{оп}^{-1}(b - A(I, J_{н})\kappa^{(1)}(J_{н}))$.
Если

$$d_*(J_{оп}) \leq \kappa^{(1)}(J_{оп}) \leq d^*(J_{оп}), \quad (19)$$

то $\kappa^{(1)}$ — β -оптимальный план задачи (5), $\beta = \delta'(J_{оп+}) \times \times (\kappa^{(1)}(J_{оп+}) - d_*(J_{оп+})) + \delta'(J_{оп-}) (\kappa^{(1)}(J_{оп-}) - d^*(J_{оп-}))$, $J_{оп+} = \{j \in J_{оп} : \delta_j > 0\}$, $J_{оп-} = \{j \in J_{оп} : \delta_j < 0\}$.

При $\beta \leq \varepsilon$ решение задачи (5) прекращается на ε -оптимальном плане $\kappa^{(1)}$. Эта часть итерации совпадает с аналогичной из § 1.

Рассмотрим случай, когда на псевдоплане условие (19) не выполняется или выполняется, но $\beta > \varepsilon$. В этом случае найдется такой индекс*) $j_1 \in J_{\text{оп}}$, что $\kappa_{j_1}^{(1)} \notin [\bar{d}_{*j_1}, \bar{d}_{j_1}^*]$. Здесь $\bar{d}_{*j_1} = \bar{d}_{j_1}^* = d_{*j_1}$ при $\delta_{j_1} > 0$, $\bar{d}_{*j_1} = \bar{d}_{j_1}^* = d_{j_1}^*$ при $\delta_{j_1} < 0$; $\bar{d}_{*j_1} = d_{*j_1}$, $\bar{d}_{j_1}^* = d_{j_1}^*$ при $\delta_{j_1} = 0$.

Положим $\mu_1 = 1$, если $\kappa_{j_1}^{(1)} < \bar{d}_{*j_1}$; $\mu_1 = -1$, если $\kappa_{j_1}^{(1)} > \bar{d}_{j_1}^*$; $z_0(J_{\text{нб}}^{(0)}) = \delta(J_{\text{нб}}^{(0)})$. С этими данными корректируем псевдоплан $\kappa^{(4)}$ по правилам п. 5, начиная вычисления с $s=1$. Корректировка псевдоплана $\kappa^{(4)}$ заканчивается либо построением ε -оптимального плана x^ε исходной задачи (т. е. задача (5) решена), либо построением нового опорного коплана $\{\delta, \bar{A}_{\text{оп}}\}$, на котором значение двойственной целевой функции задачи (5) лучше, чем на $\{\delta, A_{\text{оп}}\}$, либо будет обнаружена несовместимость ограничений задачи (5). В первом и третьем случаях решение задачи (5) прекращается. Во втором случае продолжаем решение задачи (5), исходя из опорного коплана $\{\bar{\delta}, \bar{A}_{\text{оп}}\}$.

5. Процедура корректировок псевдоплана κ . Пусть для натурального числа s известны:

$$J_{\text{оп}}, J_{\text{н}} = J \setminus J_{\text{оп}}, J_{\text{нб}}^{(k)}, k = \overline{0, s-1}, J_{\text{нб}}^{(s-1)}; \kappa^{(s)}; j_k, \mu_k, k = \overline{1, s}; z_k(J_{\text{нб}}^{(k)}), k = \overline{0, s-1}. \quad (20)$$

Положим $z_s(J_{\text{н}}) = \mu_s A_{\text{оп}}^{-1}(j_s, I) A(I, J_{\text{н}})$, $z_s(j_s) = \mu_s$. Возможны случаи: 1) $J_{\text{нб}}^{(s-1)} = \emptyset$; 2) $J_{\text{нб}}^{(s-1)} \neq \emptyset$.

В случае 1) корректировку псевдоплана $\kappa^{(s)}$ начинаем с замены опоры $A_{\text{оп}}$. Для этого найдем числа

$$\sigma(j) = \begin{cases} -z_k(j)/z_s(j), & \text{если } z_k(j) z_s(j) < 0, \\ \infty & \text{в остальных случаях, } j \in J_{\text{нб}}^{(k)}, \\ & k = s-1, s-2, \dots, 0; \end{cases}$$

$$\sigma(j_s) = \begin{cases} |\delta_{j_s}|, & \text{если } \delta_{j_s} z_s(j_s) < 0, \\ \infty & \text{в остальных случаях.} \end{cases}$$

В каждой группе $\{\sigma(j), j \in J_{\text{нб}}^{(k)}\}$, $k = s-1, \dots, 1$; $\{\sigma(j), j \in J_{\text{нб}}^{(0)} \cup j_s\}$ числа $\sigma(j) < \infty$ упорядочим по возрастанию: $\sigma(j_{1,k}) \leq \delta(j_{2,k}) \leq \dots \leq \sigma(j_{p_k,k})$. Положим $\bar{d}_{*j_s} = \bar{d}_{j_s}^* =$

*) При конкретных вычислениях индекс j_1 можно искать из условия максимального удаления $\kappa_{j_1}^{(1)}$ от $[\bar{d}_{*j_1}, \bar{d}_{j_1}^*]$, $j \in J_{\text{оп}}$.

$= d_{*j_s}$ при $\delta_{j_s} > 0$; $\bar{d}_{*j_s} = \bar{d}_{j_s}^* = d_{j_s}^*$ при $\delta_{j_s} < 0$; $\bar{d}_{*j_s} = d_{*j_s}$, $\bar{d}_{j_s}^* = d_{j_s}^*$ при $\delta_{j_s} = 0$;

$$\alpha(0, s-1) = \begin{cases} \kappa_{j_s}^{(s)} - \bar{d}_{*j_s} & \text{при } \mu_s = 1, \\ \bar{d}_{j_s}^* - \kappa_{j_s}^{(s)} & \text{при } \mu_s = -1. \end{cases} \quad (21)$$

Из рекуррентных соотношений $\alpha(i, k) = \alpha(i-1, k) + |z_s(j_i, k)(d_{j_i, k}^* - d_{*j_i, k})|$, $i = \overline{1, p_k}$, $k = s-1, s-2, \dots, 0$, с начальными условиями $\alpha(0, k) = \alpha(p_{k+1}, k+1)$, $k = s-2, \dots, 0$, найдем числа $\alpha(i, k)$, $i = \overline{1, p_k}$, $k = \overline{0, s-1}$.

Если $\alpha(p_0, 0) < 0$, то решение задачи (5) прекращается, ибо задача (5) не имеет планов (двойственная целевая функция задачи (5) неограниченно убывает на множестве двойственных планов).

Предположим, что $\alpha(p_0, 0) \geq 0$. Тогда найдутся такие индексы v, s_* , что $\alpha(v, s_*) \geq 0$, $\alpha(v-1, s_*) < 0$ (при $v=1$ полагаем $\alpha(0, s_*) = \alpha(p_{s_*+1}, s_*+1)$). Вместо $A_{\text{оп}}$ берем

новую опору $\tilde{A}_{\text{оп}} = A(I, \tilde{J}_{\text{оп}})$, $\tilde{J}_{\text{оп}} = (J_{\text{оп}} \setminus j_s) \cup j_{v, s_*}$. Если $s_* = 0$, $v > 1$, то корректировка завершается построением опорного коплана $\{\bar{\delta}, \bar{A}_{\text{оп}}\}$, $\bar{\delta}_j = \delta_j + \sigma(j_{v, s_*}) z_s(j)$, $j \in J_{\text{н}} \cup j_s$; $\bar{\delta}_j = \delta_j$, $j \in J_{\text{оп}} \setminus j_s$; $\bar{A}_{\text{оп}} = A(I, \bar{J}_{\text{оп}})$, $\bar{J}_{\text{оп}} = \tilde{J}_{\text{оп}}$.

Нетрудно показать, что при замене коплана δ на $\bar{\delta}$ двойственная целевая функция задачи (5) уменьшается

на величину $\sum_{k=1}^v \alpha(k, 0) [\sigma(j_{k, 0}) - \sigma(j_{k-1, 0})] < 0$, где $\sigma(j_{0, 0}) = 0$.

З а м е ч а н и е 1. При $j_s = j_{v, s_*}$ фактически меняется только коплан: $\delta \rightarrow \bar{\delta}$. Опора остается прежней, поскольку $\tilde{J}_{\text{оп}} = (J_{\text{оп}} \setminus j_s) \cup j_s = J_{\text{оп}}$.

При $s_* > 0$ ($s_* = 0$, $v = 1$) всю информацию, связанную с номерами $k = s_* + 1, s$, забываем. Корректировку продолжаем с исходными данными: s_* , $\tilde{J}_{\text{оп}}$, $\tilde{J}_{\text{н}} = J \setminus \tilde{J}_{\text{оп}}$, $\tilde{J}_{\text{нб}}^{(k)} = J_{\text{нб}}^{(k)}$, $k = \overline{0, s_* - 1}$, $\tilde{y}_{\text{нб}}^{(s_* - 1)} = \tilde{J}_{\text{н}} \setminus \bigcup_{k=0}^{s_* - 1} \tilde{J}_{\text{нб}}^{(k)}$; $\tilde{\kappa}^{(s_*)} = \kappa^{(s)}$, $\tilde{j}_k = j_k$, $\tilde{\mu}_k = \mu_k$, $k = \overline{1, s_*}$; $\tilde{z}_k(\tilde{J}_{\text{нб}}^{(k)}) = z_k(J_{\text{нб}}^{(k)})$, $k = \overline{0, s_* - 1}$.

Рассмотрим случай 2): $J_{\text{нб}}^{(s-1)} \neq \emptyset$. Положим $J_{\text{нб}}^{(s)} = \{j \in J_{\text{нб}}^{(s-1)} : z_s(j) = 0\}$, $J_{\text{н-}}^{(s)} = \{j \in J_{\text{нб}}^{(s-1)} : z_s(j) < 0\}$, $J_{\text{н+}}^{(s)} =$

$= \{j \in J_{\text{нб}}^{(s-1)} : z_s(j) > 0\}$, $J_{\text{нб}}^{(s)} = J_{\text{нб}}^{(s)} \cup J_{\text{н+}}^{(s)}$. Построим новый псевдоплан $\kappa^{(s+1)} = \kappa^{(s+1)}(J)$:

$$\kappa^{(s+1)}(J_{\text{н}} \setminus J_{\text{нб}}^{(s)}) = \kappa^{(s)}(J_{\text{н}} \setminus J_{\text{нб}}^{(s)}); \quad \kappa^{(s+1)}(J_{\text{н-}}^{(s)}) = d^*(J_{\text{н-}}^{(s)});$$

$$\kappa^{(s+1)}(J_{\text{н+}}^{(s)}) = d_*(J_{\text{н+}}^{(s)}); \quad (22)$$

$$\kappa^{(s+1)}(J_{\text{оп}}) = A_{\text{оп}}^{-1}(b - A(I, J_{\text{н}}) \kappa^{(s+1)}(J_{\text{н}})).$$

Возможны подслучаи:

2а) $\kappa_{j_s}^{(s+1)} < \bar{d}_{*j_s}$ при $\mu_s = 1$ или $\kappa_{j_s}^{(s+1)} > \bar{d}_{j_s}^*$ при $\mu_s = -1$;

2б) $\kappa_{j_s}^{(s+1)} \in [\bar{d}_{*j_s}, \bar{d}_{j_s}^*]$;

2в) $\kappa_{j_s}^{(s+1)} > \bar{d}_{j_s}^*$ при $\mu_s = 1$ или $\kappa_{j_s}^{(s+1)} < \bar{d}_{*j_s}$ при $\mu_s = -1$.

В случае 2а) действуем согласно случаю 1), заменив в (21) $\kappa_{j_s}^{(s)}$ на $\kappa_{j_s}^{(s+1)}$. В случае 2б) проверяем условия

$$d_*(J_{\text{оп}}) \leq \kappa^{(s+1)}(J_{\text{оп}}) \leq d^*(J_{\text{оп}}),$$

$$\beta = \delta'(J_{\text{оп+}})(\kappa^{(s+1)}(J_{\text{оп+}}) - d_*(J_{\text{оп+}})) +$$

$$+ \delta'(J_{\text{оп-}})(-d^*(J_{\text{оп-}}) + \kappa^{(s+1)}(J_{\text{оп-}})) \leq \varepsilon.$$

Если они выполняются, то корректировка прекращается на ε -оптимальном плане $x^e = \kappa^{(s+1)}$ задачи (1). В противном случае существует такой индекс $j_{s+1} \in J_{\text{оп}} \setminus \{j_1, \dots, j_s\}$, что $\kappa_{j_{s+1}}^{(s+1)} \notin [\bar{d}_{*j_{s+1}}, \bar{d}_{j_{s+1}}^*]$, где $\bar{d}_{*j_{s+1}} = \bar{d}_{j_{s+1}}^* = d_{*j_{s+1}}$ при $\delta_{j_{s+1}} > 0$; $\bar{d}_{*j_{s+1}} = \bar{d}_{j_{s+1}}^* = d_{j_{s+1}}^*$ при $\delta_{j_{s+1}} < 0$; $\bar{d}_{*j_{s+1}} = d_{*j_{s+1}}, \bar{d}_{j_{s+1}}^* = d_{j_{s+1}}^*$ при $\delta_{j_{s+1}} = 0$. Полагаем $\mu_{s+1} = 1$, если $\kappa_{j_{s+1}}^{(s+1)} < \bar{d}_{*j_{s+1}}$; $\mu_{s+1} = -1$, если $\kappa_{j_{s+1}}^{(s+1)} > \bar{d}_{j_{s+1}}^*$, и повторяем описанный процесс корректировки, заменив s на $s+1$.

В случае 2в) множество $J_{\text{оп}}$ заменяем на множество $\tilde{J}_{\text{оп}} = (J_{\text{оп}} \setminus j_s) \cup j_*$, где j_* — любой элемент из $J_{\text{нб}}^{(s)} \neq \emptyset$. При $\delta_{j_s} = 0$ множества $J_{\text{н}}, J_{\text{нб}}^{(s)}, J_{\text{нб}}^{(s)}$ заменяем на множества $\tilde{J}_{\text{н}} = \tilde{J} \setminus \tilde{J}_{\text{оп}}, \tilde{J}_{\text{нб}}^{(s)} = j_s, \tilde{J}_{\text{нб}}^{(s)} = J_{\text{нб}}^{(s-1)} \setminus j_*$; вектор $z_s(J_{\text{нб}}^{(s)})$ — на вектор $\tilde{z}_s(\tilde{J}_{\text{нб}}^{(s)}) = -\mu_s$; остальные данные не меняются, т. е. $\tilde{s} = s, \tilde{J}_{\text{нб}}^{(k)} = J_{\text{нб}}^{(k)}, k = 0, s-1$, и т. д. При $\delta_{j_s} \neq 0$ в исходных данных (20) производим следующие изменения: номер s заменяем на $\tilde{s} = s-1$; множества $J_{\text{оп}}, J_{\text{н}}, J_{\text{нб}}^{(0)}$ — на $\tilde{J}_{\text{оп}}, \tilde{J}_{\text{н}} = J \setminus \tilde{J}_{\text{оп}}, \tilde{J}_{\text{нб}}^{(0)} = J_{\text{нб}}^{(0)} \cup j_s$; вектор $z_0(J_{\text{нб}}^{(0)})$ —

на вектор $\tilde{z}_0(\tilde{J}_{\text{нб}}^{(0)}) = \delta(\tilde{J}_{\text{нб}}^{(0)})$; в псевдоплане $\kappa^{(s-1)}$ j_s -ю компоненту полагаем равной d_{*j_s} при $\delta_{j_s} > 0$ и $d_{j_s}^*$ при $\delta_{j_s} < 0$; все остальные данные, связанные с номерами $k = \overline{0, s-1}$, не меняются (информация, связанная с номером s , во внимание не принимается). Используя новые данные, переходим к построению псевдоплана $\kappa^{(\tilde{s}+1)}$ по формулам (22), заменив s на \tilde{s} . При этом обязательно реализуется случай 2б) и дальнейшие операции проводим согласно этому случаю.

З а м е ч а н и е 2. В случае 2в) индекс j_* можно выбирать из $J_{\text{нб}}^{(s)}$ по следующему правилу. Пусть $J_{\text{нб}}^{(s)} = \{i_1, \dots, i_p\}$. Положим $\gamma_0 = \kappa_{j_s}^{(s+1)}$, $\gamma_k = \gamma_{k-1} - \mu_s |z_s(i_k)| (d_{i_k}^* - d_{*i_k})$, $k = \overline{1, p}$. По построению, при $\mu_s = 1$ имеем $\gamma_0 = \kappa_{j_s}^{(s+1)} > \bar{d}_{j_s}^*$, $\gamma_p = \kappa_{j_s}^{(s)} < \bar{d}_{*j_s}$, $\gamma_k \leq \gamma_{k-1}$; при $\mu_s = -1$ имеем $\gamma_0 = \kappa_{j_s}^{(s+1)} < \bar{d}_{*j_s}$, $\gamma_p = \kappa_{j_s}^{(s)} > \bar{d}_{j_s}^*$, $\gamma_k \geq \gamma_{k-1}$. Следовательно, существует такой номер \bar{k} , что $\gamma_{\bar{k}} \leq \bar{d}_{j_s}^*$, $\gamma_{\bar{k}-1} > \bar{d}_{j_s}^*$ при $\mu_s = 1$; $\gamma_{\bar{k}} \geq \bar{d}_{*j_s}$, $\gamma_{\bar{k}-1} < \bar{d}_{*j_s}$ при $\mu_s = -1$.

В качестве индекса j_* , вводимого в опору вместо j_s , выбираем индекс $j_* = i_{\bar{k}} \in J_{\text{нб}}^{(s)}$. Если в псевдоплане $\kappa^{(s)}$ изменить компоненты $\kappa_{i_{\bar{k}}}^{(s)}$, $k = \overline{1, \bar{k}-1}$, положив $\kappa_{i_{\bar{k}}}^{(s)} = d_{*i_{\bar{k}}}$, если $\kappa_{i_{\bar{k}}}^{(s)} = d_{i_{\bar{k}}}^*$, и $\kappa_{i_{\bar{k}}}^{(s)} = d_{i_{\bar{k}}}^*$, если $\kappa_{i_{\bar{k}}}^{(s)} = d_{*i_{\bar{k}}}$, $k = \overline{1, \bar{k}-1}$, то для j_* -й опорной компоненты псевдоплана $\kappa^{(\tilde{s}+1)}$, построенного по формулам (22), будет выполняться критерий оптимальности, т. е. $\kappa_{j_*}^{(\tilde{s}+1)} \in [\bar{d}_{*j_*}, \bar{d}_{j_*}^*]$.

З а м е ч а н и е 3. Можно упростить алгоритм, если в случае 2) при $\delta_{j_s} \neq 0$ индекс j_s сразу вывести из опорного множества $J_{\text{оп}}$: $J_{\text{оп}} \rightarrow \bar{J}_{\text{оп}} = (J_{\text{оп}} \setminus j_s) \cup j_*$, где j_* — любой элемент из $J_{\text{нб}}^{(s-1)} \neq \emptyset$, и закончить корректировку на опорном коплане $\{\bar{\delta}, \bar{A}_{\text{оп}}\}$, $\bar{\delta} = \delta$, $\bar{A}_{\text{оп}} = A(I, \bar{J}_{\text{оп}})$.

Лемма 7. Процесс корректировки состоит из конечного числа шагов. Доказательство леммы аналогично доказательству леммы п. 3.

Теорема 3. При любых $A, b, c, d_*, d^*, m < \infty, n < \infty, \varepsilon \geq 0$ для каждого начального опорного коплана $\{\delta, A_{\text{оп}}\}$ через конечное число итераций задача (5) будет решена, т. е. будет построен ее ε -оптимальный план либо обнаружена несовместимость ее ограничений.

Доказательство. Как отмечалось выше, каждая итерация метода (корректировка псевдоплана) заканчивается либо решением задачи (5), либо построением нового опорного коплана $\{\delta, \bar{A}_{\text{оп}}\}$ с лучшим значением двойственной целевой функции.

Пусть $\{\delta^{(k)}, A_{\text{оп}}^{(k)}\}$, $A_{\text{оп}}^{(k)} = A(I, J_{\text{оп}}^{(k)})$, — опорный коплан, $f(\delta^{(k)})$ — значение двойственной целевой функции на k -й итерации. Обозначим $J_{\text{оп}*}^{(k)} = \{j \in J_{\text{оп}}^{(k)} : \delta_j^{(k)} \neq 0\}$. Согласно процедуре корректировки имеем

$$J_{\text{оп}*}^{(s)} \subset J_{\text{оп}*}^{(k)}; \quad (23)$$

$$\delta_j^{(s)} = \delta_j^{(k)}, \quad j \in J_{\text{оп}*}^{(s)}; \quad (24)$$

$$f(\delta^{(s)}) < f(\delta^{(k)}) \quad (25)$$

для любого $s > k$.

Предположим, что возможно бесконечное число итераций. Тогда, в силу (23) и $|J| = n < \infty$, через конечное число итераций будут построены опорные копланы $\{\delta^{(k)}, A_{\text{оп}}^{(k)}\}$, $\{\delta^{(s)}, A_{\text{оп}}^{(s)}\}$, $s > k$, для которых $J_{\text{оп}}^{(k)} = J_{\text{оп}}^{(s)}$, $J_{\text{оп}*}^{(k)} = J_{\text{оп}*}^{(s)}$. Учитывая последние равенства, (24) и то, что $\delta_j^{(s)} = \delta_j^{(k)} = 0$, $j \in J_{\text{оп}} \setminus J_{\text{оп}*}^{(k)}$, заключаем, что $\delta^{(k)}(J_{\text{оп}}^{(k)}) = \delta^{(s)}(J_{\text{оп}}^{(k)})$. Откуда следует $\delta^{(k)} = \delta^{(s)}$, ибо коплан однозначно определяется своими опорными компонентами. Однако это противоречит (25). Полученное противоречие доказывает теорему.

6. Пример. Рассмотрим задачу линейного программирования

$$\begin{aligned} & -2x_{12} - 5x_{13} + 4x_{14} \rightarrow \max, \\ & x_1 + 4x_9 + 2x_{10} - x_{11} + 2x_{12} + x_{13} = 19, \\ & x_2 + 2x_7 + x_8 + 5x_9 - 5x_{11} + x_{14} = 52, \\ & x_3 - x_6 + 4x_7 + 2x_9 - 4x_{10} - 4x_{12} - x_{13} = 1, \\ & x_4 - 5x_5 + x_7 - 2x_9 + x_{10} + 3x_{11} + 8x_{13} + 2x_{14} = 36, \quad (26) \\ & 0 \leq x_1 \leq 5, \quad 0 \leq x_2 \leq 4, \quad 0 \leq x_3 \leq 8, \quad -1 \leq x_4 \leq 9, \\ & 0 \leq x_5 \leq 2, \quad 0 \leq x_6 \leq 10, \quad 4 \leq x_7 \leq 5, \quad -1 \leq x_8 \leq 4, \\ & 1 \leq x_9 \leq 4, \quad 2 \leq x_{10} \leq 4, \quad 4 \leq x_{11} \leq 6, \quad 0 \leq x_{12} \leq 1, \\ & 1 \leq x_{13} \leq 5, \quad 1 \leq x_{14} \leq 5, \end{aligned}$$

для которой известен начальный опорный план $\{x, A_{\text{оп}}\}$, $x = \{4, 2, 4, 0, 1, 0, 5, -1, 2, 4, 6, 1, 3, 1\}$, $A_{\text{оп}} = A(I, J_{\text{оп}}) = E$, $J_{\text{оп}} = \{1, 2, 3, 4\}$. Оценка субоптимальности начального опорного плана $\beta(x, A_{\text{оп}}) = 28$. Положим $\varepsilon = 9$, $\eta =$

$=0,001$ и осуществим одну итерацию алгоритма, описанного в пп. 2—4.

Всю информацию о задаче, необходимую в ходе вычислений, удобно, как и в табличной реализации симплекс-алгоритма, представлять в виде таблиц. Информация о задаче (26) помещена в табл. 17, где цифры над

Табл. 17

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
d_*	0	0	0	-1	0	0	4	-1	1	2	4	0	1	1
d^*	5	4	8	9	2	10	5	4	4	4	6	1	5	5
x	4	2	4	0	1	0	5	-1	2	4	6	1	3	1
1	1	0	0	0	0	0	0	0	4	2	-1	2	1	0
2	0	1	0	0	0	0	2	1	5	0	5	0	0	1
3	0	0	1	0	0	-1	4	0	0	-4	0	-4	-1	0
4	0	0	0	1	-1	0	1	0	-2	1	1	0	8	2
Δ	0	0	0	0	0	0	0	0	0	0	0	2	5	4
$l^{(1)}$	4	-4	-6	8	0	0	0	0	0	0	0	-1	-2	4/1
$x^{(1)}$	5	1	5/2	2	1	0	5	-1	2	4	6	3/4	5/2	2/2
$l^{(2)}$	-7	-3	-9/2	12	0	0	0	0	2	0	-2	-3/4	-3/2	3/3
$x^{(2)}$	8/3	0	1	6	1	0	5	-1	8/3	4	16/3	1/2	2	3/4
$l^{(3)}$	-14/3	0	1	9	0	0	-1	0	4/3	0	-4/3	-1/2	-1	2/5
$x^{(3)}$	10/3	0	4/3	9	1	0	14/3	-1	28/9	4	44/9	1/3	5/3	11/36
$l^{(4)}$	-28/9	0	2/3	5	-1	0	-2/3	0	8/9	0	-8/9	-1/3	-2/3	4/3

$$\Delta_1 \theta^{(1)} = 1/4; \Delta_2 \mu_1 = -1; \Delta_3 \theta^{(2)} = 1/3; \Delta_4 \mu_2 = 1; \Delta_5 \theta^{(3)} = 1/3; \Delta_6 \mu_3 = -1$$

таблицей указывают номера столбцов, т. е. соответствуют индексам множества J ; в первых трех строках записаны компоненты векторов d_* , d^* , x ; в последующих четырех строках записаны компоненты матрицы $\bar{A}(K, J) = B(K, I)A(I, J)$, $K=J_{\text{оп}}$, $B(K, I) = A_{\text{оп}}^{-1}$. Эту часть таблицы назовем основной (она обведена жирной линией). Слева от нее записаны индексы $j \in K$. В дальнейшем эле-

менты j_s строки, стоящие в столбцах с индексами $j \in J_{\text{нб}}^s$, будем заштриховывать; в седьмой строке таблицы запишем вектор оценок Δ ; в последующих строках — компоненты направлений $l^{(s)}$ и планов $x^{(s)}$, $s=1, 2, \dots$, получаемых в процессе корректировок направлений без замены множества K . Справа от $l^{(s)}$ строки удобно записывать шаг $\Theta^{(s)} = \Theta_{j_s}$.

Приступим к вычислениям. Следуя (6), (7), найдем вектор оценок Δ , направление $l^{(1)}$ и шаг $\Theta^{(1)} = \Theta_1 = 1/4$. Построим план $x^{(1)}$, для которого получим $\beta(x^{(1)}, A_{\text{оп}}) = (1 - 1/4) \cdot 28 = 21 > \varepsilon$. Далее переходим к процедуре корректировок направлений. Полагаем $s=1$, $K = \{1, 2, 3, 4\}$, $J_{\text{нб}}^{(0)} = \{12, 13, 14\}$, $J_{\text{нб}}^{(0)} = \{j = \overline{5, 11}\}$, $j_0 = \emptyset$, $\mu_1 = -1$, $z_0(j) = \Delta(j)$, $j = \overline{1, 14}$. (В табл. 17 компоненты Δ -строки с индексами из $J_{\text{нб}}^{(0)}$ заштрихованы.)

Поскольку $j_1 = 1 \in \{j_0\}$, то имеет место случай 1). Компоненты вектора $z_s(K_{\text{н}})$ (9) легко найти по таблице. Для этого необходимо элементы строки основной части таблицы, соответствующей индексу j_s , умножить на μ_s . В силу того, что $J_{\text{нб}}^{(0)} \neq \emptyset$, дальнейшие операции осуществляем по правилам случая 1б): полагаем $J_{\text{нб}}^{(1)} = \{5, 6, 7, 8\}$, $J_{\text{нб}}^{(1)} = \{9, 10, 11\}$ и по формулам (7) строим $l^{(2)}$. Поскольку $\alpha(0, s-1) = d_{j_1}^* - x_{j_1}^{(1)} - l_{j_1}^{(2)} > 0$ и $\eta > \min\{21; 0,001\}$, то имеет место случай 1б₂). Находим шаг $\Theta^{(2)} = \Theta_{j_2} = \Theta_2 = 1/3$ и новый план $x^{(2)}$, оценка субоптимальности которого равна $\beta(x^{(2)}, A_{\text{оп}}) = (1 - 1/3) \cdot 21 = 14$. Полагаем $\mu_2 = 1$, номер $s=1$ заменяем на $s=2$ и повторяем процесс корректировки.

Поскольку $j_2 \notin \{j_0, j_1\}$, $J_{\text{нб}}^{(1)} \neq \emptyset$, то согласно случаю 1б) строим $J_{\text{нб}}^{(2)} = \{5, 6\}$, $J_{\text{нб}}^{(2)} = \{7, 8\}$ и направление $l^{(3)}$. Для $l^{(3)}$ имеем: $\alpha(0, s-1) = 0$, $\gamma > \min\{14; 0,001\}$. Следовательно, дальнейшие вычисления проводим по правилам случая 1б₂): находим шаг $\Theta^{(3)} = \Theta_{j_3} = \Theta_4 = 1/3$ и строим план $x^{(3)}$, для которого получается $\beta(x^{(3)}, A_{\text{оп}}) = 2/3 \cdot 14 = 28/3$. Полагаем $\mu_3 = -1$, $j_3 = 4$, $s=2 \rightarrow s=3$ и продолжаем корректировку по правилам 1б).

Для направления $l^{(4)}$ имеем: $\alpha(0, s-1) = -5 < 0$. Следовательно, реализовался случай 1б₁). Перейдем к операциям по замене множества K (см. случай 1а)). Подсчитаем числа (10): $\sigma(7) = 2$, $\sigma(8) = \infty$, $\sigma(9) = 2$, $\sigma(10) = \infty$, $\sigma(11) = 1$, в каждой группе упорядочим их по воз-

растанию: $\sigma(7) = \sigma(j_{1,2}) = 2$, $\sigma(11) = \sigma(j_{1,1}) = 1 < \sigma(9) = \sigma(j_{2,1}) = 2$ и найдем $\alpha(0, 2) = -5 < 0$, $\alpha(1, 2) = -5 + 1 = -4 < 0$, $\alpha(1, 1) = -4 + 1 \cdot (6 - 4) = -2 < 0$, $\alpha(2, 1) = -2 + 2(4 - 1) = 4 > 0$. Следовательно, $j_{v,s_*} = j_{2,1} = 9$, $v = 2$, $s_* = 1$. Множество K заменяем на $\tilde{K} = (K \setminus j_4) \cup j_{2,1} = \{1, 2, 3, 9\}$. Поскольку $s_* = 1 > 0$, то продолжаем корректировку, используя данные (13) при $s_* = 1$, $s = 3$. Этим данным соответствует табл. 18. Чтобы получить табл. 18

Табл. 18

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	0	0	$\frac{2}{3}$	$-\frac{2}{3}$	0	$\frac{2}{3}$	0	0	$\frac{4}{3}$	$\frac{1}{3}$	2	17	4
2	0	1	0	$\frac{5}{2}$	$-\frac{5}{2}$	0	$\frac{9}{2}$	1	0	$\frac{5}{2}$	$\frac{15}{2}$	0	20	6
3	0	0	1	0	0	-1	4	0	0	-4	0	-4	-1	0
9	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$	0	1	$-\frac{1}{2}$	$-\frac{1}{2}$	0	-4	-1
$\tilde{l}^{(2)}$	$\frac{26}{9}$	-7	$-\frac{10}{3}$	0	-1	0	$\frac{1}{3}$	0	$-\frac{1}{9}$	0	$\frac{10}{9}$	$-\frac{1}{3}$	$-\frac{2}{3}$	$\frac{4}{3}$

из табл. 17, надо основную часть таблицы пересчитать по правилу прямоугольника симплекс-алгоритма с ведущим элементом $(j_s, j_{v,s_*}) = -2$. Легко проверить, что при этом строки с индексами $j \in \{j_1, \dots, j_{s-1}\}$ меняться не будут. Не изменится и Δ -строка.

Следуя процедуре корректировок и используя данные (13), по формулам (14) строим направление $\tilde{l}^{(s_*+1)} = \tilde{l}^{(2)}$.

Для $\tilde{l}^{(2)}$ имеем: $\tilde{\alpha}(0, 1-1) = 5 - 10/3 - 26/9 = -11/9 < 0$, т. е. реализовался случай 1б₁). Переходим к операции по

замене множества \tilde{K} . Подсчитаем числа $\sigma(12) = 1$, $\sigma(13) = 5/17$, $\sigma(14) = \infty$, упорядочим их по возрастанию: $\sigma(j_{1,0}) = \sigma(13) = 5/17 < \sigma(j_{2,0}) = \sigma(12) = 1$ и найдем $\tilde{\alpha}(0, 0) = -11/9 < 0$, $\tilde{\alpha}(1, 0) = -11/9 + 4 \cdot 17 > 0$. Следовательно, $j_{\tilde{v}, \tilde{s}_*} = j_{1,0} = 13$, $\tilde{v} = 1$, $\tilde{s}_* = 0$.

Множество \tilde{K} заменяем на $\bar{J}_{\text{оп}} = (\tilde{K} \setminus j_1) \cup j_{1,0} = \{13, 2, 3, 9\}$. Так как $\tilde{s}_* = 0$, то процедура корректировок заканчивается на опорном плане $\{\bar{x}, \bar{A}_{\text{оп}}\}$, $\bar{x} = x^{(3)}$, $\bar{A}_{\text{оп}} = A(I, \bar{J}_{\text{оп}})$, для которого $\beta(\bar{x}, \bar{A}_{\text{оп}}) = 28/3 - (11/9) \cdot (5/17) = 8,97 < \beta(x, A_{\text{оп}}) = 28$. Поскольку

$\beta(\bar{x}, \bar{A}_{\text{оп}}) = 8,97 < \varepsilon = 9$, то решение задачи (26) прекращаем на ε -оптимальном плане $x^* = \bar{x} = \{10/3, 0, 4/3, 9, 1, 0, 14/3, -1, 28/9, 4, 44/9, 1/3, 5/3, 11/3\}$.

§ 7. μ -адаптивный метод

Конечные варианты адаптивного метода, описанные в § 6, основаны на чисто вырожденных сопровождающих копланам и поэтому нацелены прежде всего на теоретические приложения. Для численных расчетов интерес представляют алгоритмы [21, 23], учитывающие конечную разрядность представления чисел на ЭВМ, т. е. вычисления с округлением. В данном параграфе разрабатываются алгоритмы (прямой и двойственный), обладающие упомянутым свойством. При этом построения связываются не с вырожденными, а с почти вырожденными сопровождающими копланами. В конечном счете получаются алгоритмы, работающие на произвольных опорных планах, но по ходу операций приспособляющиеся к ситуациям, близким к вырождению. Численные эксперименты [21], некоторые результаты которых приведены в конце параграфа, подтверждают эффективность алгоритмов. Определенный интерес для проблемы испытания алгоритмов представляют, на наш взгляд, и два класса задач, использованных в указанных экспериментах.

1. Прямой метод. Рассмотрим задачу

$$c'x \rightarrow \max, Ax = b, d_* \leq x \leq d^*. \quad (1)$$

Коплан δ назовем μ -сопровождающим опором $J_{\text{оп}} (\mu \geq 0)$, если $J_{\text{оп}} \subset J_\mu$, где $J_\mu = \{j: |\delta_j| \leq \mu\}$. Опорный коплан $\{\delta, J_{\text{оп}}\}$ называем μ -вырожденным, если $J_\mu \cap J_{\text{н}} \neq \emptyset$, где $J_{\text{н}} = J \setminus J_{\text{оп}}$.

При $\mu = 0$ понятия μ -сопровождающего и μ -вырожденного копланов переходят в понятия сопровождающего и вырожденного копланов.

Пусть перед началом решения задачи известен некоторый ее опорный план $\{x, J_{\text{оп}}\}$. Зададим $\varepsilon \geq 0$ — требуемую точность приближения к оптимуму по целевой функции; $\mu^1 \geq 0$, $\eta \geq 0$ — начальные значения параметров метода. Исходной задаче припишем первый уровень ($s = 1$) и положим $J^1 = J$.

Шаг 1. Строим 0-сопровождающий коплан δ^s : $\delta^s(J_{\text{оп}}) = 0$, $\delta^s(J_{\text{н}}^s) = A'(I, J_{\text{н}}^s)(A_{\text{оп}}^{-1})' c^s(J_{\text{оп}}) - c^s(J_{\text{н}}^s)$,

где $c^1=c$, $c^s = -\bar{\delta}^s$ при $s>1$. В качестве $\bar{\delta}^s$ можно взять предыдущее значение δ^s .

Шаг 2. Вычисляем оценку субоптимальности:

$$\beta(x, \delta^1) = \sum_{j: \delta_j^1 > 0} \delta_j^1 (x_j - d_{*j}) + \sum_{j: \delta_j^1 < 0} \delta_j^1 (x_j - d_j^*).$$

Если $\beta(x, \delta^1) \leq \epsilon$, то решение закончено, так как текущий план x является ϵ -оптимальным. В противном случае корректируем направление l улучшения плана x :

$$l_j = \begin{cases} d_{*j} - x_j, & \text{если } \delta_j^s > \mu^s, \\ d_j^* - x_j, & \text{если } \delta_j^s < -\mu^s, \quad j \in J_{\text{н}}, \\ 0, & \text{если } |\delta_j^s| \leq \mu^s, \end{cases}$$

$$l(J_{\text{оп}}) = -A_{\text{оп}}^{-1} A(I, J_{\text{н}}) l(J_{\text{н}}).$$

При $l(J_{\text{н}}^s) = 0$ уменьшаем μ^s и переходим к шагу 1. (Правила уменьшения μ^s будут описаны ниже.) Если $l(J_{\text{н}}^s) \neq 0$, то вычисляем шаг Θ^0 вдоль l :

$$\Theta^0 = \min \{1, \Theta_{j_0}\}, \quad \Theta_{j_0} = \min \{\Theta_j, j \in J_{\text{оп}}\},$$

$$\Theta_j = \begin{cases} (d_j^* - x_j)/l_j & \text{при } l_j > 0, \\ (d_{*j} - x_j)/l_j & \text{при } l_j < 0, \\ \infty & \text{при } l_j = 0. \end{cases}$$

Вычисляем $\bar{x} = x + \Theta^0 l$. Если $\beta(\bar{x}, \delta^1) \leq \epsilon$, то \bar{x} — ϵ -оптимальный план. При $\beta(\bar{x}, \delta^1) > \epsilon$ проверяем неравенство $c'\bar{x} - c'x \geq \eta$, где $c'(\bar{x} - x) = -\Theta^0 l' \delta^1$. Если это неравенство выполнено, то план x заменяем на \bar{x} , в противном случае x не меняем. В случае $\Theta^0 = 1$ уменьшаем μ^1 , полагаем $s=1$ и осуществляем переход к шагу 1.

Шаг 3. Выбираем $j_0 \in J_{\text{оп}}$, $\Theta_{j_0} < 1$. Строим направление $\Delta\delta(J^s)$ уменьшения коплана $\delta^s(J_s)$: $\Delta\delta_{j_0} = -\text{sign } l_{j_0}$,

$$\Delta\delta(J_{\text{оп}} \setminus j_0) = 0, \quad \Delta\delta'(J_{\text{н}}^s) = \Delta\delta'(J_{\text{оп}}) A_{\text{оп}}^{-1} A(I, J_{\text{н}}^s).$$

Шаг 4. Вычисляем шаг σ^* вдоль $\Delta\delta$. Для этого вначале находим шаги σ_j по каждой компоненте δ_j^s , $j \in J_{\text{н}}^s$:

$$\sigma_j = \begin{cases} 0, & \text{если } |\delta_j^s| \leq \mu^s \text{ и } \Delta\delta_j \neq 0, \\ -\delta_j^s / \Delta\delta_j, & \text{если } |\delta_j^s| > \mu^s \text{ и } \Delta\delta_j \delta_j^s < 0, \\ \infty & \text{в остальных случаях.} \end{cases}$$

Далее вычисляем $\alpha_0 = -|l_{j_0}|(1 - \Theta_{j_0})$ и $\Delta\alpha_j =$

$= \Delta \delta_j (x_j + l_j - \bar{d}_j)$ для $j \in J_{\text{н}}^s$, $\sigma_j \neq \infty$, где $\bar{d}_j = d_{*j}$ при $\Delta \delta_j > 0$, $\bar{d}_j = d_j^*$ при $\Delta \delta_j < 0$.

Пусть $\sigma_{j_1} \leq \sigma_{j_2} \leq \dots \leq \sigma_{j_p} \neq \infty$ и $\sigma_j = \infty, \forall j \in J_{\text{н}}^s \setminus \{j_1, j_2, \dots, j_p\}$. Тогда если $\alpha_0 + \sum_{k=1}^p \Delta \alpha_{j_k} \geq 0$, то $\sigma^* = \sigma_{j_q}$, где

$q = \min \{i : \alpha_0 + \sum_{k=1}^i \Delta \alpha_{j_k} \geq 0\}$. В случае $\alpha_0 + \sum_{k=1}^p \Delta \alpha_{j_k} < 0$ полагаем $\sigma^* = \infty$. Если $\sigma^* = 0$, то переход к шагу 6; если $\sigma^* = \infty$ — к шагу 7.

Шаг 5. Коплан δ^s заменяем на $\bar{\delta}^s = \delta^s + \sigma^* \Delta \delta$, опору $J_{\text{оп}}$ — на $\bar{J}_{\text{оп}} = (J_{\text{оп}} \setminus j_0) \cup j_q$. Переход к шагу 2.

Шаг 6. Строим $J^{s+1} = \{j \in J^s, |\delta_j^s| \leq \mu^s\}$. Полагаем $\delta^{s+1}(J_{\text{оп}}) = 0$, $\delta^{s+1}(J_{\text{н}}^{s+1}) = \Delta \delta(J_{\text{н}}^{s+1})$; выбираем $\mu^{s+1} \geq 0$, если этот параметр не определялся ранее; s увеличиваем на единицу. Переход к шагу 2.

Шаг 7. Вычисляем $\Delta \delta'(J^{s-1} \setminus J^s) = \Delta \delta'(J_{\text{оп}}) A_{\text{оп}}^{-1} A(I, J^{s-1} \setminus J^s)$; s уменьшаем на единицу. Переход к шагу 4.

З а м е ч а н и е. Правило уменьшения параметров μ^s должно обладать следующим свойством: $\mu^s \rightarrow 0$ при $\varepsilon = 0$; через конечное число уменьшений $\mu^0 = 0$.

Теорема 1. При любых $A, b, c, d_*, d^*, m < \infty, n < \infty, \varepsilon \geq 0, \eta \geq 0$ для любого начального опорного плана $\{x, J_{\text{оп}}\}$ через конечное число итераций будет построен ε -оптимальный план задачи (1).

Каждую итерацию задачи начинаем выполнением шагов 1, 2 или 4 и заканчиваем на шагах 2, 5, 6 или 7 переходом к шагу с меньшим номером. Любую итерацию можно отнести к одному из следующих типов:

- 1) начало итерации — шаг 1, 2 или 4, конец — шаг 5;
- 2) начало итерации — шаг 1 или 2, конец — шаг 6;
- 3) начало итерации — шаг 1, 2 или 4, конец — шаг 7;
- 4) начало итерации — шаг 1 или 2, конец — шаг 2 при $l(J_{\text{н}}^s) = 0$;
- 5) начало итерации — шаг 1 или 2, конец — шаг 2 при $\Theta^0 = 1, \beta > \varepsilon$;
- 6) начало итерации — шаг 1 или 2, конец — останов по $\beta \leq \varepsilon$.

Доказательству теоремы предположим несколько лемм.

Лемма 1. Величина s при работе метода ограничена.

Доказательство. Параметр s увеличивается в случае, когда $\exists j_1 \in J_{\text{н}}^s, j_2 \in J_{\text{н}}^s, |\delta_{j_1}^s| \leq \mu^s, |\delta_{j_2}^s| > \mu^s$.

Это возможно только при $|J_n^s| \geq 2$. С другой стороны, справедливо строгое вложение $J_n^{s+1} = \{j \in J_n^s, |\delta_j^s| \leq \mu^s\} \subset J_n^s$. Следовательно, s не может увеличиваться бесконечно, в частности, $s \leq n - m$. Лемма доказана.

Лемма 2. Число итераций шестого типа конечно.

Доказательство. Очевидно, что итерация шестого типа не может встретиться более одного раза. Лемма доказана.

Лемма 3. Число итераций пятого типа конечно.

Доказательство. Рассмотрим случай $\varepsilon = 0$. Тогда через конечное число итераций пятого типа $\mu^4 = 0$. При этом если $d_* \leq x + l \leq d^*$, то $\beta(x + l, \delta^1) = 0$ и итерации заканчиваются. В случае $\varepsilon > 0$ при бесконечном числе итераций пятого типа $\mu^1 \rightarrow 0$. Значит, через конечное число итераций пятого типа $\mu^1 \leq \varepsilon / (\sum_{j \in J} (d_j^* - d_{*j}))$, что влечет $\beta(x + l, \delta^1) \leq \varepsilon$ при $d_* \leq x + l \leq d^*$. Лемма доказана.

Лемма 4. Число итераций четвертого типа конечно.

Доказательство. Вначале покажем, что число итераций четвертого типа при $s > 1$ не может быть бесконечным. Действительно, каждый вектор $\delta^s(J^s)$ является копланом задачи

$$c^s x \rightarrow \max, \quad A(I, J^s) x^s(J^s) = b - A(I, J \setminus J^s)(x(J \setminus J^s) + l(J \setminus J^s)), \quad d_*(J^s) \leq x(J^s) \leq d^*(J^s), \quad (2)$$

где c^s определяется по правилам шага 1.

Начальные опорные копланы $\delta_{\text{нач}}^s$ этих задач являются 0-согласованными и обладают свойством $\delta_{\text{нач}}^s(J_n^s) \neq 0$. Следовательно, и у любого другого 0-согласованного коплана задачи (2) $\delta^s(J_n^s) \neq 0$. Для любых копланов $\delta^s(J^s)$ и $\bar{\delta}^s(J^s)$ справедливо равенство $\delta^s(J_n^s) - \bar{\delta}^s(J_n^s) = A'(I, J_n^s)(A_{\text{оп}}^{-1})' \times (\delta^s(J_{\text{оп}}) - \bar{\delta}^s(J_{\text{оп}}))$. Число различных опор задачи (1) конечно, поэтому $\exists R$ такое, что из неравенства $\|\delta^s(J_{\text{оп}}) - \bar{\delta}^s(J_{\text{оп}})\| \leq \mu$ следует неравенство $\|\delta^s(J_n^s) - \bar{\delta}^s(J_n^s)\| \leq R\mu$, $\forall J_{\text{оп}}$. Число различных 0-согласованных копланов в задаче (2) конечно, следовательно, $\exists Q > 0$ такое, что $\|\delta^s(J_n^s)\| \geq Q$ для любого 0-согласованного опорного коплана $\{\delta^s, J_{\text{оп}}\}$.

Положим $0 \leq \mu^s < Q/2R$. Тогда для любого μ^s -согласованного опорного коплана $\{\delta^s, J_{\text{оп}}\}$ выполняется неравенство $\|\delta^s(J_n^s)\| > \mu^s$. При бесконечном количестве итераций

четвертого типа $\mu^s \rightarrow 0$. Значит, через конечное число итераций четвертого типа будет выполняться неравенство $\mu^s < Q/2R$, после чего дальнейшие итерации четвертого типа на этом уровне невозможны до перехода на уровень $s-1$. Число s ограничено, и число задач вида (2) на каждом уровне конечно, следовательно, число итераций четвертого типа при $s > 1$ конечно.

Рассмотрим случай $s=1$. При этом c либо линейно не зависит от строк матрицы A (тогда можно повторить рассуждения для случая $s > 1$), либо линейно зависит от них. Но тогда после первой же итерации четвертого типа при $s=1$ будет выполняться равенство $\delta^1(J)=0$, а это влечет $\beta(x, \delta^1)=0 \quad \forall x$ и решение задачи на этом закончится. Лемма доказана.

Лемма 5. Двойственная целевая функция любой задачи линейного программирования вида (1) ограничена на множестве ее μ -согласованных копланов.

Доказательство. Для доказательства несуществен факт совместности ограничений задачи (1). Двойственная к (1) задача имеет вид

$$b'y + d^*w - d_*v \rightarrow \min, \quad A'y + w - v = c, \quad w \geq 0, \quad v \geq 0, \quad (3)$$

где $y \in R^m$; $w, v \in R^n$. Совокупность $\{y, w, v\}$ называется двойственным планом. Будем рассматривать только согласованные двойственные планы, т. е. такие, для которых выполняются условия: $w_j = 0, v_j = \delta_j$, если $\delta_j \geq 0$; $w_j = -\delta_j, v_j = 0$, если $\delta_j < 0$, где $\delta = A'y - c$. При этом $b'y + d^*w - d_*v = \min \{b'y + d^*\bar{w} - d_*\bar{v}, A'y + \bar{w} - \bar{v} = c, \bar{w} \geq 0, \bar{v} \geq 0\}$. Тогда целевую функцию задачи (3) можно считать функцией от δ и вычислить по формуле

$$\begin{aligned} \Phi(\delta) &= b'y + d^*w - d_*v = \\ &= c'x + \sum_{\delta_j > 0} \delta_j (\kappa_j - d_{*j}) + \sum_{\delta_j < 0} \delta_j (\kappa_j - d_j^*), \end{aligned}$$

где $x \in R^n$ удовлетворяет соотношению $Ax = b$ (x — псевдопланы задачи (1)).

Пусть δ и $\bar{\delta}$ — копланы задачи (1). Тогда

$$\begin{aligned} |\Phi(\bar{\delta}) - \Phi(\delta)| &= \left| \sum_{\substack{\delta_j > 0, \\ \bar{\delta}_j > 0}} (\bar{\delta}_j - \delta_j) (\kappa_j - d_{*j}) + \sum_{\substack{\delta_j < 0, \\ \bar{\delta}_j < 0}} (\bar{\delta}_j - \delta_j) + \right. \\ &\quad \left. + \sum_{\substack{\delta_j > 0, \\ \bar{\delta}_j < 0}} [\bar{\delta}_j (\kappa_j - d_j^*) - \delta_j (\kappa_j - d_{*j})] + \right. \end{aligned}$$

$$+ \sum_{\substack{\delta_j < 0, \\ \delta_j > 0}} [\bar{\delta}_j (\kappa_j - d_{*j}) - \delta_j (\kappa_j - d_j^*)] |.$$

Вектор κ не зависит от δ , $\bar{\delta}$. Следовательно, для любой задачи (1) существует $T > 0$ такое, что $|\Phi(\bar{\delta}) - \Phi(\delta)| \leq T \|\bar{\delta} - \delta\|$. Множество 0-согласованных копланов конечно. Поэтому $\Phi(\delta)$ ограничена на этом множестве. Для любого μ -согласованного коплана справедлива оценка $\|\bar{\delta} - \delta\| \leq R\mu$, где δ — некоторый 0-согласованный коплан. Значит, выполняется неравенство $|\Phi(\bar{\delta}) - \Phi(\delta)| \leq TR\mu$, которое означает ограниченность двойственной целевой функции на множестве μ -согласованных копланов. Лемма доказана.

Лемма 6. Число итераций первого типа конечно.

Доказательство. Воспользовавшись леммами 3 и 4, без ограничения общности можно допустить, что числа μ^s постоянны при работе метода и итерации четвертого и пятого типов не встречаются. Предположим, что число итераций первого типа бесконечно. Величина s ограничена. Следовательно, найдется s_0 такое, что число итераций первого типа при $s = s_0$ бесконечно. Будем считать, что s_0 — минимальное среди таких чисел. Тогда на всех дальнейших итерациях, начиная с некоторой, выполняется неравенство $s \geq s_0$. Действительно, в противном случае при $s = s_0$ выполнялось бы бесконечно много итераций третьего типа. Поскольку $|\delta_j^{s-1}| > \mu^{s-1} \forall j \in J^{s-1} \setminus J^s, \forall s > 1$, то за итерацией третьего типа может следовать итерация только третьего или первого типов. Так как случай $s = s_0$ встречается бесконечное число раз, то при $s < s_0$ должно происходить бесконечное число итераций первого типа, что противоречит предположению о минимальности числа s_0 . Таким образом, при $s = s_0$ найдутся задачи вида (2) и последовательность копланов $\delta_{(i)}^{s_0}, i = N, N+1, \dots$, такие, что $\delta_{(i)}^{s_0} \neq \delta_{(i+1)}^{s_0}$ и все копланы $\delta_{(i)}^{s_0}$ μ^{s_0} -согласованы с соответствующими им опорами. (Из описания метода следует, что переход от μ^s -согласованного опорного коплана происходит только к новому μ^s -согласованному опорному коплану.)

Вычислим

$$\Phi(\delta_{(i+1)}^{s_0}) - \Phi(\delta_{(i)}^{s_0}) = \sigma^* \left(\sum_{\substack{\delta_{j(i)}^{s_0} > 0, \delta_{j(i+1)}^{s_0} > 0, \\ j \in J^{s_0}}} \Delta \delta_j (\kappa_j - d_{*j}) + \right.$$

$$\begin{aligned}
& + \sum_{\substack{j \in J^{s_0}, \delta_{j(i)}^{s_0} < 0, \\ \delta_{j(i+1)}^{s_0} < 0}} \Delta \delta_j (\kappa_j - d_j^*) + \sum_{\substack{j \in J^{s_0}, \delta_{j(i)}^{s_0} \geq 0, \\ \delta_{j(i+1)}^{s_0} < 0}} (\delta_{j(i+1)}^{s_0} (\kappa_j - d_j^*) - \\
& - \delta_{j(i)}^{s_0} (\kappa_j - d_{*j})) + \sum_{\substack{j \in J^{s_0}, \delta_{j(i)}^{s_0} < 0, \\ \delta_{j(i+1)}^{s_0} \geq 0}} (\delta_{j(i+1)}^{s_0} (\kappa_j - d_{*j}) - \delta_{j(i)}^{s_0} (\kappa_j - d_j^*)).
\end{aligned}$$

Подставим вместо κ согласованный с $(\delta_{j(i)}^{s_0}, J_{\text{оп}(i)})$ и $\Delta \delta$ псевдоплан, т. е. псевдоплан, удовлетворяющий следующим соотношениям: для $j \in J_{\text{н}}^{s_0}$

$$\kappa_j = \begin{cases} d_j^*, & \text{если } \delta_{j(i)}^{s_0} < -\mu^{s_0} \text{ или } |\delta_{j(i)}^{s_0}| \leq \mu^{s_0}, \Delta \delta_j < 0, \\ d_{*j}, & \text{если } \delta_{j(i)}^{s_0} > \mu^{s_0} \text{ или } |\delta_{j(i)}^{s_0}| \leq \mu^{s_0}, \Delta \delta_j \geq 0, \end{cases}$$

для $j \in J \setminus J^{s_0}$

$$\kappa_j = x_j + l_j,$$

$$\kappa(J_{\text{оп}(i)}) = A(I, J_{\text{оп}(i)})^{-1} (b - A(I, J \setminus J_{\text{оп}(i)}) \kappa(J \setminus J_{\text{оп}(i)})).$$

При этом расстояние от κ_{j_0} до отрезка $[d_{*j_0}, d_{j_0}^*]$ равно величине $-(\alpha_0 + \sum_{\sigma_j=0} \Delta \alpha_j)$. Учитывая, что $\Delta \delta(J_{\text{оп}} \setminus j_0) = 0$,

$\Delta \delta_{j_0} = \pm 1$, получим

$$\begin{aligned}
& \Phi(\delta_{(i+1)}^{s_0}) - \Phi(\delta_{(i)}^{s_0}) \leq \sigma_{j_l} (\alpha_0 + \sum_{\sigma_j=0} \Delta \alpha_j) + \\
& + \sum_{\substack{j \in J_{\text{н}}^{s_0}(i), \\ 0 \leq \delta_{j(i)}^{s_0} \leq \mu^{s_0}, \\ \delta_{j(i+1)}^{s_0} \geq 0, \Delta \delta_j < 0}} \Delta \delta_j (d_j^* - d_{*j}) + \\
& + \sum_{\substack{j \in J_{\text{н}}^{s_0}(i), \\ -\mu^{s_0} \leq \delta_{j(i)}^{s_0} < 0, \\ \delta_{j(i+1)}^{s_0} < 0, \Delta \delta_j > 0}} \Delta \delta_j (d_{*j} - d_j^*) - \sum_{\substack{j \in J_{\text{н}}^{s_0}(i), \\ 0 \leq \delta_{j(i)}^{s_0} \leq \mu^{s_0}, \\ \delta_{j(i+1)}^{s_0} < 0}} \delta_{j(i)}^{s_0} (d_j^* - d_{*j}) - \\
& - \sum_{\substack{j \in J_{\text{н}}^{s_0}(i), \\ -\mu^{s_0} \leq \delta_{j(i)}^{s_0} < 0, \\ \delta_{j(i+1)}^{s_0} \geq 0}} \delta_{j(i)}^{s_0} (d_{*j} - d_j^*) \leq \sigma_{j_l} (\alpha_0 + \sum_{\sigma_j=0} \Delta \alpha_j),
\end{aligned}$$

где σ_{j_l} — минимальный ненулевой шаг, полученный при переходе от $\delta_{(i)}^{s_0}$ к $\delta_{(i+1)}^{s_0}$,

$$\alpha_0 + \sum_{\sigma_j=0} \Delta \alpha_j = \begin{cases} \kappa_{j_0} - d_{*j_0}, & \text{если } \Delta \delta_{j_0} = 1, \\ d_{j_0}^* - \kappa_{j_0}, & \text{если } \Delta \delta_{j_0} = -1. \end{cases}$$

Все неопорные компоненты псевдоплана κ — критические (лежат на границах). Следовательно, таких псевдопланов конечное число и существует $q < 0$ такое, что из

$$\alpha_0 + \sum_{\sigma_j=0} \Delta \alpha_j < 0 \text{ следует } \alpha_0 + \sum_{\sigma_j=0} \Delta \alpha_j < q < 0, \sigma_{j_l} \geqslant \geqslant |\mu^{s_0}| / \max_{\Delta \delta \neq 0} |\Delta \delta_j|. \text{ Но различных возможных векторов}$$

$\Delta \delta_j$ также конечное число, поэтому если $\mu^{s_0} > 0$, то $\exists p < 0$ такое, что $\Phi(\delta_{(i+1)}^{s_0}) - \Phi(\delta_{(i)}^{s_0}) < p$, и тогда $\Phi(\delta_{(i)}^{s_0}) \xrightarrow{i \rightarrow \infty} -\infty$, что противоречит лемме 5.

Если же $\mu^{s_0} = 0$, то выполняется неравенство $\Phi(\delta_{(i+1)}^{s_0}) - \Phi(\delta_{(i)}^{s_0}) \leqslant \sigma_{j_l} (\alpha_0 + \sum_{\sigma_j=0} \Delta \alpha_j) < 0$ и все копланы $\delta_{(i)}^{s_0}$ в рас-

сматриваемой последовательности 0-согласованны и, в силу $\Phi(\delta_{(i+1)}^{s_0}) - \Phi(\delta_{(i)}^{s_0}) < 0$, различны. У любой задачи линейного программирования найдется только конечное число 0-согласованных копланов. Следовательно, последовательность $\delta_{(i)}^{s_0}$ не может быть бесконечной. Лемма доказана.

Лемма 7. Число итераций третьего типа конечно.

Доказательство. В силу лемм 3, 4 считаем, что происходят итерации только первого, второго и третьего типов. За каждой итерацией третьего типа может следовать итерация только третьего или первого типов, так как после каждой итерации третьего типа шаг σ^* на следующей итерации строго положителен. Бесконечная последовательность итераций третьего типа подряд невозможна, поскольку выполнение итерации третьего типа означает несовместность ограничений задачи (2). Но ограничения задачи (2) при $s=1$ совместны (у этой задачи предполагается существование начального плана). Значит, за каждой итерацией третьего типа через конечное число итераций этого же типа должна следовать итерация первого типа, т. е. будет выполняться бесконечное число итераций первого типа, что невозможно. Лемма доказана.

Лемма 8. Число итераций второго типа конечно.

Доказательство. Ранее показано, что число итераций остальных типов конечно. Следовательно, если число итераций второго типа бесконечно, то, начиная с некоторой итерации, все остальные должны принадлежать ко второму типу. При этом s должно неограниченно увеличиваться, что противоречит лемме 1. Лемма доказана.

Доказательство теоремы 1. С одной стороны, итерации могут закончиться только построением ϵ -оптимального плана задачи (1), с другой — число итераций каждого типа, а значит, и их общее число конечно. Следовательно, через конечное число итераций будет получен ϵ -оптимальный план. Теорема доказана.

2. Двойственный метод. Рассмотрим задачу (1). Решение начинаем с некоторого начального опорного коплана $\{\delta, J_{\text{оп}}\}$. Задаем $\epsilon \geq 0$ и параметр метода $\mu^1 \geq 0$. Исходной задаче приписываем первый уровень ($s=1$) и полагаем $d_*^1 = d_*$, $d^{*1} = d^*$, $\delta^1 = \delta$.

Шаг 1. Вычисляем неопорные компоненты псевдоплана κ_j , $j \in J_{\text{н}}$:

$$\kappa_j = \begin{cases} d_j^{*s} & \text{при } \delta_j^s < -\mu^s, \\ d_{*j}^s & \text{при } \delta_j^s > \mu^s, \\ \text{произвольно в интервале } [d_{*j}^s, d_j^{*s}] & \text{при } |\delta_j^s| \leq \mu^s. \end{cases}$$

Шаг 2. Вычисляем опорные компоненты псевдоплана $\kappa(J_{\text{оп}}) = -A_{\text{оп}}^{-1}(b - A(I, J_{\text{н}}) \kappa(J_{\text{н}}))$. Если $d_*(J_{\text{оп}}) \leq \kappa(J_{\text{оп}}) \leq d^*(J_{\text{оп}})$, то переход к шагу 3, в противном случае — к шагу 4.

Шаг 3. Вычисляем $\beta(\kappa, \delta^1)$. Если $\beta(\kappa, \delta^1) \leq \epsilon$, то κ — ϵ -оптимальный план и решение заканчивается; если $\beta(\kappa, \delta^1) > \epsilon$, то переход к шагу 4.

Шаг 4. Выбираем $j_0 \in J_{\text{оп}}$, $\kappa_{j_0} \notin [d_{*j_0}^s, d_{j_0}^{*s}]$, где

$$\bar{d}_{*j}^s = \begin{cases} d_{*j}^s & \text{при } \delta_j^s \geq -\mu^s, \\ d_j^{*s} & \text{при } \delta_j^s < -\mu^s, \end{cases} \quad \bar{d}_j^{*s} = \begin{cases} d_j^{*s} & \text{при } \delta_j^s \leq \mu^s, \\ d_{*j}^s & \text{при } \delta_j^s > \mu^s. \end{cases}$$

Если такой индекс j_0 найдется, то переход к шагу 5. При $\bar{d}_{*j_0}^s(J_{\text{оп}}) \leq \kappa(J_{\text{оп}}) \leq \bar{d}_{j_0}^{*s}(J_{\text{оп}})$ уменьшаем μ^1 , полагаем $s = 1$ и переходим к шагу 1.

Шаг 5. Вычисляем подходящее направление $\Delta \delta$ изменения коплана δ^s :

$$\Delta\delta_{j_0} = \begin{cases} 1, & \text{если } \kappa_{j_0} < \bar{d}_{*j_0}^s, \\ -1, & \text{если } \kappa_{j_0} > \bar{d}_{j_0}^{*s}, \end{cases}$$

$$\Delta\delta(J_{\text{оп}} \setminus j_0) = 0, \quad \Delta\delta'(J_{\text{н}}) = \Delta\delta'(J_{\text{оп}}) A_{\text{оп}}^{-1} A(I, J_{\text{н}}).$$

Шаг 6. Находим шаг σ^* вдоль $\Delta\delta$. Для этого вначале вычисляем σ_j для $j \in J_{\text{н}} \cup j_0$:

$$\sigma_{j_0} = \begin{cases} |\delta_{j_0}^s|, & \text{если } |\delta_{j_0}^s| > \mu^s, \Delta\delta_{j_0} \delta_{j_0}^s < 0, \\ \infty & \text{в остальных случаях,} \end{cases}$$

$$\sigma_j = \begin{cases} 0, & \text{если } |\delta_j^s| \leq \mu^s, \Delta\delta_j \neq 0, \\ -\delta_j^s / \Delta\delta_j, & \text{если } |\delta_j^s| > \mu^s, \Delta\delta_j \delta_j^s < 0, \\ \infty & \text{в остальных случаях.} \end{cases}$$

Далее вычисляем

$$\alpha_0 = \begin{cases} \kappa_{j_0} - \bar{d}_{*j_0}^s, & \text{если } \kappa_{j_0} < \bar{d}_{*j_0}^s, \\ \bar{d}_{j_0}^{*s} - \kappa_{j_0}, & \text{если } \bar{d}_{j_0}^{*s} < \kappa_{j_0}, \end{cases}$$

и $\Delta\alpha_j$ для $j \in J_{\text{н}} \cup j_0, \sigma_j \neq \infty$:

$$\Delta\alpha_{j_0} = d_{j_0}^{*s} - d_{*j_0}^s, \text{ если } \sigma_{j_0} \neq \infty,$$

для $j \in J_{\text{н}}$

$$\Delta\alpha_j = \begin{cases} \Delta\delta_j (\kappa_j - d_{*j}^s), & \text{если } \Delta\delta_j > 0, \\ \Delta\delta_j (\kappa_j - d_j^{*s}), & \text{если } \Delta\delta_j < 0. \end{cases}$$

Пусть $\sigma_{k_1} \leq \sigma_{k_2} \leq \dots \leq \sigma_{k_p} \neq \infty$ и $\sigma_j = \infty \forall j \in (J_{\text{н}} \cup j_0) \setminus \{k_1, k_2, \dots, k_p\}$. Тогда если $\alpha_0 + \sum_{i=1}^p \Delta\alpha_{k_i} \geq 0$, то $\sigma^* = \sigma_{k_q}$, где $q = \min \{t : \alpha_0 + \sum_{i=1}^t \Delta\alpha_{k_i} \geq 0\}$. В случае

$\alpha_0 + \sum_{i=1}^p \Delta\alpha_{k_i} < 0$ полагаем $\sigma^* = \infty$.

Если $\sigma^* = 0$, переход к шагу 8; если $\sigma^* = \infty$ — к шагу 9; в случае $0 < \sigma^* < \infty$ переход к шагу 7.

Шаг 7. Коплан δ^s заменяем на $\bar{\delta}^s = \delta^s + \sigma^* \Delta\delta$, опоры $J_{\text{оп}}$ — на $\bar{J}_{\text{оп}} = (J_{\text{оп}} \setminus j_0) \cup k_q$. Если $k_q \neq j_0$, то производим пересчет матрицы $A_{\text{оп}}^{-1}$ по известным рекуррентным формулам. Полагаем

$$\kappa_{k_i} = \begin{cases} d_{k_i}^{*s} & \text{при } \Delta\delta_{k_i} < 0, \\ d_{*k_i}^s & \text{при } \Delta\delta_{k_i} > 0, \end{cases}$$

для $i < q$. При $k_q \neq j_0$ полагаем

$$x_{j_0} = \begin{cases} \bar{d}_{j_0}^{*s} & \text{при } \Delta\delta_{j_0} = -1 \text{ и } l > q, \text{ где } k_l = j_0, \\ d_{j_0}^{*s} & \text{при } \Delta\delta_{j_0} = -1 \text{ и } l < q, \\ \bar{d}_{*j_0}^s & \text{при } \Delta\delta_{j_0} = 1 \text{ и } l > q, \\ d_{*j_0}^s & \text{при } \Delta\delta_{j_0} = 1 \text{ и } l < q. \end{cases}$$

Затем переход к шагу 2.

Шаг 8. Если $s > 1$ и $|\delta_j^s| \leq \mu^s \forall j$, $d_{*j}^s < d_j^{*s}$, то уменьшаем μ^s и осуществляем переход к шагу 1. В противном случае полагаем $\delta^{s+1}(J_H) = \Delta\delta(J_H)$, $\delta^{s+1}(J_{оп}) = 0$, $d_*^{s+1} = \bar{d}_*^s$, $d^{s+1} = \bar{d}^{*s}$, увеличиваем s на единицу и переходим к шагу 1.

Шаг 9. Если $s=1$, то решение заканчивается: задача (1) не имеет допустимых планов. При $s > 1$ уменьшаем s на единицу. Переход к шагу 6.

Теорема 2. При любых $A, b, c, d_*, d^*, m < \infty, n < \infty, \varepsilon \geq 0$ для каждого начального опорного коплана $\{\delta, J_{оп}\}$ через конечное число итераций будет или построен ε -оптимальный план задачи (1), или установлена несовместность ее ограничений.

Любую итерацию метода можно отнести к одному из следующих типов:

- 1) начало итерации — шаг 1, 2 или 6, конец — шаг 7;
- 2) начало итерации — шаг 1 или 2, конец — 8 с увеличением s ;
- 3) начало итерации — шаг 1, 2 или 6, конец — шаг 9;
- 4) начало итерации — шаг 1 или 2, конец — шаг 8 с уменьшением μ^s ;
- 5) начало итерации — шаг 1 или 2, конец — шаг 4 с уменьшением μ^1 и переходом к шагу 1 при $s=1$;
- 6) начало итерации — шаг 1 или 2, конец — останов при $\beta \leq \varepsilon$ на шаге 3;
- 7) итерация седьмого типа начинается на шаге 1, 2 или 6, конец — останов на шаге 9.

Очевидно, что итерации шестого и седьмого типов могут выполняться только один раз. Поэтому из дальнейшего рассмотрения их исключаем.

Докажем ряд лемм, необходимых для доказательства теоремы.

Лемма 9. Величина s при работе метода ограничена.

Доказательство. Увеличение s происходит только в случае, когда $\exists j \in J$, $d_{*j}^s < d_j^{*s}$, $|\delta_j^s| > \mu^s$. Для таких j выполняется $d_{*j}^{s+1} = d_j^{*s+1}$. Следовательно, $s \leq |J| = n$. Лемма доказана.

Лемма 10. При работе метода число итераций четвертого типа конечно.

Доказательство. При работе метода на каждом уровне решается задача

$$c^s x \rightarrow \max, Ax = b, d_*^s \leq x \leq d^{*s}, \quad (4)$$

где $c^s(J \setminus j_0) = 0$, $c_{j_0}^s = \Delta \delta_{j_0}$, индекс j_0 и число $\Delta \delta_{j_0}$ получены непосредственно перед переходом на уровень s . Предположим, что при решении некоторой задачи (4) число итераций четвертого типа на этом уровне (на каждом уровне число различных задач конечно) бесконечно. Компонент j со свойствами $\delta_j^s \neq 0$, $d_{*j}^s = d_j^{*s}$ в опоре быть не может. Поэтому перед уменьшением μ^s текущий опорный коплан является μ^s -согласованным. Действительно, начальный опорный коплан в задаче (4) является 0-согласованным, а компоненты j , $d_{*j}^s = d_j^{*s}$ в опору входить не могут. Таким образом, в задаче (4) существует опора $\bar{J}_{оп}$ и последовательность копланов $\{\delta_{(i)}^s\}_{i=1}^\infty$ такая, что $\|\delta_{(i)}^s(\bar{J}_{оп} \cup \{j : d_{*j}^s < d_j^{*s}\})\| \xrightarrow{i \rightarrow \infty} 0$.

В силу замкнутости множества копланов, в задаче (4) существует такой коплан $\delta_{(0)}^s$, что $\|\delta_{(0)}^s(\bar{J}_{оп} \cup \{j : d_{*j}^s < d_j^{*s}\})\| = 0$. Каждый коплан полностью определяется своими опорными компонентами. Следовательно, $\delta_{(0)}^s$ совпадает с начальным копланом задачи (4). Но это означает, что при формировании этой задачи выполнялось равенство $\Delta \delta(J_n \cap \{j : d_{*j}^s < d_j^{*s}\}) = 0$. В этом случае обязательно $\sigma^* > 0$ и переход к уровню s невозможен. Полученное противоречие доказывает лемму.

Лемма 11. Число итераций пятого типа конечно.

Доказательство. Итерация пятого типа выполняется в случае $d_* \leq \kappa \leq d^*$, $\beta(\kappa, \delta^1) > \varepsilon$. На каждой итерации выполняется неравенство $\beta(\kappa, \delta^1) \leq \mu^1 \sum_{j \in J} (d_j^* - d_{*j})$.

Правило уменьшения μ^1 (см. прямой метод) через конечное число итераций обеспечивает $\mu^1 \rightarrow 0$ либо $\mu^1 = 0$. Следовательно, через конечное число итераций будет вы-

полняться неравенство $\beta(\kappa, \delta^1) \leq \mu^1 \sum_{j \in J} (d_j^* - d_{*j}) \leq \varepsilon$.

Лемма доказана.

Лемма 12. Число итераций первого типа конечно.

Доказательство. Предположим противное: пусть найдется задача, в процессе решения которой встречается бесконечное число итераций первого типа. Согласно лемме 1 величина s ограничена. Следовательно, найдется хотя бы один уровень s , на котором встречается бесконечное число итераций первого типа. Пусть s_0 — минимальный такой уровень. Покажем, что в этом случае, начиная с некоторой итерации, все время выполняется неравенство $s \geq s_0$. Действительно, в противном случае на уровне s выполнялось бы бесконечное число итераций третьего типа.

Воспользовавшись леммами 10 и 11, без ограничения общности предположим, что итераций четвертого и пятого типов не происходит. Тогда за итерацией третьего типа может следовать итерация только третьего или первого типа. Бесконечная последовательность итераций третьего типа невозможна. Значит, за каждой итерацией первого типа на уровне s_0 должна следовать итерация первого типа на уровне меньшем чем s_0 , что противоречит принятому предположению. Таким образом, приходим к выводу о существовании задачи вида (4) и бесконечной последовательности копланов в ней $\{\delta_{(i)}^{s_0}\}_{i=1}^\infty$, $\delta_{(i)}^{s_0} \neq \delta_{(i+1)}^{s_0} \forall i$. В силу лемм 2 и 3 без ограничения общности считаем, что μ^{s_0} при работе метода не изменяется. Из описания метода следует, что если $\delta_{(i)}^{s_0}$ — μ^{s_0} -согласованный коплан, то и $\delta_{(i+1)}^{s_0}$, а значит, и все последующие копланы будут μ^{s_0} -согласованными.

Рассмотрим два случая: 1) все $\delta_{(i)}^{s_0}$, $i = 1, 2, \dots$, μ^{s_0} -согласованы с соответствующими им опорами, 2) все $\delta_{(i)}^{s_0}$ не μ^{s_0} -согласованны. Случай, когда в последовательности $\{\delta_{(i)}^{s_0}\}_{i=1}^\infty$ имеется конечное число не μ^{s_0} -согласованных копланов, легко сводится к случаю 1).

Рассмотрим случай 1). Двойственная целевая функция задачи (4) может быть вычислена по формуле

$$\Phi(\delta^s) = c' \kappa + \sum_{\delta_j^s > 0} \delta_j (\kappa_j - d_{*j}) + \sum_{\delta_j^s < 0} \delta_j (\kappa_j - d_j^*),$$

где κ — любой псевдоплан задачи. Тогда

$$\begin{aligned}
& \Phi(\delta_{(i+1)}^{s_0}) - \Phi(\delta_{(i)}^{s_0}) = \sigma_* \left(\sum_{\substack{\delta_{j(i)}^{s_0} \geq 0, \\ \delta_{j(i+1)}^{s_0} \geq 0}} \Delta \delta_j (\kappa_j - d_{*j}^{s_0}) + \right. \\
& + \sum_{\substack{\delta_{j(i)}^{s_0} < 0, \\ \delta_{j(i+1)}^{s_0} < 0}} \Delta \delta_j (\kappa_j - d_j^{*s_0}) \left. \right) + \sum_{\substack{\delta_{j(i)}^{s_0} \geq 0, \\ \delta_{j(i+1)}^{s_0} < 0}} (\delta_{j(i+1)}^{s_0} (\kappa_j - d_j^{*s_0}) - \\
& - \delta_{j(i)}^{s_0} (\kappa_j - d_{*j}^{s_0})) + \sum_{\substack{\delta_{j(i)}^{s_0} < 0, \\ \delta_{j(i+1)}^{s_0} \geq 0}} (\delta_{j(i+1)}^{s_0} (\kappa_j - d_{*j}^{s_0}) - \\
& - \delta_{j(i)}^{s_0} (\kappa_j - d_j^{*s_0})).
\end{aligned}$$

Подставим в качестве κ μ^{s_0} -согласованный с $\delta_{(i)}^{s_0}$ и $\Delta \delta$ псевдоплан, т. е. псевдоплан, удовлетворяющий соотношениям

$$\begin{aligned}
\kappa_j &= \begin{cases} d_j^{*s_0}, & \text{если } \delta_{j(i)}^{s_0} < -\mu^{s_0} \text{ или } |\delta_{j(i)}^{s_0}| \leq \mu^{s_0}, \Delta \delta_j < 0, j \in J_{H(i)}, \\ d_{*j}^{s_0}, & \text{если } \delta_{j(i)}^{s_0} > \mu^{s_0} \text{ или } |\delta_{j(i)}^{s_0}| \leq \mu^{s_0}, \Delta \delta_j > 0, j \in J_{H(i)}, \end{cases} \\
\kappa(J_{\text{он}(i)}) &= A_{\text{он}}^{-1} (b - A(I, J_{H(i)}) \kappa(J_{H(i)})).
\end{aligned}$$

При этом получим

$$\begin{aligned}
& \Phi(\delta_{(i+1)}^{s_0}) - \Phi(\delta_{(i)}^{s_0}) \leq \sigma_{k_l} \left(\alpha_0 + \sum_{\sigma_j=0} \Delta \alpha_j + \right. \\
& + \sum_{\substack{j \in J_{H(i)}, \\ 0 \leq \delta_{j(i)}^{s_0} \leq \mu^{s_0}, \\ \delta_{j(i+1)}^{s_0} \geq 0, \Delta \delta_j < 0}} \Delta \delta_j (d_j^{*s_0} - d_{*j}^{s_0}) + \sum_{\substack{j \in J_{H(i)}, \\ -\mu^{s_0} \leq \delta_{j(i)}^{s_0} < 0, \\ \delta_{j(i+1)}^{s_0} < 0, \Delta \delta_j > 0}} \Delta \delta_j (d_{*j}^{s_0} - d_j^{*s_0}) \left. \right) - \\
& - \sum_{\substack{j \in J_{H(i)}, \\ 0 \leq \delta_{j(i)}^{s_0} \leq \mu^{s_0}, \\ \delta_{j(i+1)}^{s_0} < 0}} \delta_{j(i)}^{s_0} (d_j^{*s_0} - d_{*j}^{s_0}) - \sum_{\substack{j \in J_{H(i)}, \\ -\mu^{s_0} \leq \delta_{j(i)}^{s_0} < 0, \\ \delta_{j(i+1)}^{s_0} \geq 0}} \delta_{j(i)}^{s_0} (d_{*j}^{s_0} - d_j^{*s_0}) \leq \\
& \leq \sigma_{k_l} \left(\alpha_0 + \sum_{\sigma_j=0} \Delta \alpha_j \right),
\end{aligned}$$

где σ_{k_l} — минимальный ненулевой шаг, построенный при вычислении σ^* , $\delta_{(i+1)}^{s_0} = \delta_{(i)}^{s_0} + \sigma^* \Delta \delta$. Кроме того,

$$\alpha_0 + \sum_{\sigma_j=0} \Delta \alpha_j = \begin{cases} \kappa_{j_0} - d_{*j_0}^{s_0} & \text{при } \Delta \delta_{j_0} = 1, \\ d_{j_0}^{*s_0} - \kappa_{j_0} & \text{при } \Delta \delta_{j_0} = -1. \end{cases} \quad (5)$$

Из описания метода следует, что $\alpha < 0$. Покажем, что для любой задачи (1) существуют константы $P > 0$, $Q < 0$ такие, что если $\sigma_j > 0$, то $\sigma_j > P\mu^s$, и если $\alpha_0 + \sum_{\sigma_j=0} \Delta \alpha_j < 0$,

то $\alpha_0 + \sum_{\sigma_j=0} \Delta \alpha_j < Q$. В самом деле, если $0 < \sigma_j < \infty$, то

$\sigma_j = |\delta_j^s|/|\Delta \delta_j| > \mu^s/|\Delta \delta_j|$. Векторы $\Delta \delta$ составлены из коэффициентов разложения столбцов матрицы A по текущим опорным векторам. Поскольку число различных опор конечно, то $\exists P > 0$, $1/|\Delta \delta_j| > P$. Следовательно, $\sigma_j > \mu^s P$.

В формуле (5) фигурирует компонента псевдоплана, все неопорные компоненты которого принимают граничные значения. Количество таких псевдопланов ограничено. Значит, $\exists Q < 0$ такое, что из $\alpha_0 + \sum_{\sigma_j=0} \Delta \alpha_j < 0$ следует

$$\alpha_0 + \sum_{\sigma_j=0} \Delta \alpha_j < Q \quad \text{и оценка} \quad \Phi(\delta_{(i+1)}^{s_0}) - \Phi(\delta_{(i)}^{s_0}) < \mu^{s_0} PQ.$$

В случае $\mu^{s_0} = 0$ все копланы $\delta_{(i)}^{s_0}$ 0-согласованны. Таких копланов конечное число. В то же время $\Phi(\delta_{(i+1)}^{s_0}) - \Phi(\delta_{(i)}^{s_0}) < 0 \quad \forall i$, поэтому бесконечная последовательность $\{\delta_{(i)}^{s_0}\}_{i=1}^{\infty}$ не может существовать. Если же $\mu^{s_0} > 0$, то $\Phi(\delta_{(i)}^{s_0}) \xrightarrow{i \rightarrow \infty} -\infty$, что противоречит лемме 5. Для случая 1) лемма доказана.

Рассмотрим случай 2). Из описания метода следует, что если некоторый индекс j вошел в опору, то $|\delta_j^{s_0}| \leq \mu^{s_0}$. Если же $j \in J_{\text{оп}}$ и $|\delta_j^{s_0}| \leq \mu^{s_0}$, то $\delta_j^{s_0}$ не изменяется в течение

всего времени, пока $j \in J_{\text{оп}}$. Обозначим: $J_1 = \bigcup_{i=1}^{\infty} J_{\text{оп}(i)}$, $J_2 = \bigcap_{i=1}^{\infty} J_{\text{оп}(i)}$. Ясно, что $J_2 \neq \emptyset$, ибо в противном случае,

начиная с некоторого i_0 , все опорные копланы $\{\delta_{(i)}^{s_0}, J_{\text{оп}(i)}\}$, $i \geq i_0$, были бы μ^{s_0} -согласованными. Без ограничения общности можно считать, что $|\delta_{j(i)}^{s_0}| \leq \mu^{s_0}$ для $j \in J_{\text{оп}(i)} \setminus J_2$. Кроме того, если $j \in J_2$, то $\delta_{j(i)}^{s_0} = \delta_{j(k)}^{s_0} \quad \forall i, k$.

Пусть $J_3 = \{j \in J_2, |\delta_{j(i)}^{s_0}| \geq \mu^{s_0} \forall i\}$. Рассмотрим задачу (4) на s_0 уровне с измененным условием: $\delta_{j(1)}^{s_0} = 0 \quad \forall j \in J_3$, остальные $\delta_{j(1)}^{s_0}$ сохраняют прежние значения. В этой задаче по правилам метода можно построить ту же

самую последовательность итераций первого типа с единственным отличием: $\delta_{j(i)}^{s_0} = 0 \quad \forall i$. Все копланы в этой последовательности будут μ^{s_0} -согласованными. Эта ситуация соответствует случаю 1). Выше было показано, что она невозможна. Следовательно, невозможна и ситуация случая 2). Лемма доказана.

Лемма 13. Число итераций третьего типа конечно.

Доказательство. Без ограничения общности считаем, что встречаются итерации только первых трех типов. Предположим, что число итераций третьего типа бесконечно. За итерацией третьего типа не может следовать итерация второго типа, так как на следующей итерации обязательно $\sigma^* > 0$. Бесконечная последовательность итераций третьего типа подряд невозможна, так как итерации третьего типа происходят только при $s \geq 2$. Но тогда в бесконечной последовательности итераций задачи встречается бесконечная последовательность итераций первого типа, что противоречит лемме 12. Лемма доказана.

Лемма 14. Число итераций второго типа конечно.

Доказательство леммы совпадает с доказательством леммы 8.

Доказательство теоремы 2. С одной стороны решение задачи может закончиться только на итерации шестого или седьмого типа, с другой — число итераций остальных типов конечно. Следовательно, через конечное число итераций будет получен ε -оптимальный план задачи (7) или установлена несовместность ее ограничений. Теорема доказана.

3. Численный эксперимент. Для проверки эффективности μ -адаптивного метода был проведен численный эксперимент. Сравнивались прямой μ -адаптивный, прямой опорный [5] и адаптивный методы (см. § 1). В первой части эксперимента задачи ЛП генерировались следующим образом: векторы d_* , d^* , начальный план x , а также первые n_c столбцов матрицы A и компонент c заполнялись случайными числами, равномерно распределенными на отрезке $[-1, 1]$. Оставшаяся часть столбцов A и компонент c заполнялась случайными линейными комбинациями уже построенных столбцов A и компонент c . Начальная опора $J_{\text{оп}}$ полагалась равной $\{j=1 + [(n_c-1)/(m-1)](i-1), i=1, m\}$. Вычислялся вектор $b = Ax$. Результаты эксперимента представлены в

Табл. 19

m	n	n_c	Среднее время, с			Среднее число смен опоры			$-\lg \mu$
			адаптивный	прямой	μ -адаптивный	адаптивный	прямой	μ -адаптивный	
10	40	20	4.76	4.61	4.48	11.2	17.0	11.0	8
10	60	20	8.39	9.71	7.89	13.2	26.8	13.4	8
10	80	20	13.75	22.26	13.10	16.2	48.8	17.2	8
10	100	20	16.80	30.48	15.91	16.0	55.6	17.4	8
5	100	20	8.32	21.06	7.77	11.2	70.0	11.0	8
3	100	20	4.19	16.00	3.98	6.8	74.8	6.8	8
10	500	50	122.77	600	106.18	33.3	—	34.7	14
5	500	50	37.00	600	35.32	12.7	—	14.7	14
10	200	50	32.31	250	30.73	25.0	—	27.3	14

табл. 19. (Прочерки в таблице означают, что хотя бы одна задача серии данным методом не решена.)

Во второй части эксперимента решались задачи линейного программирования, полученные из линейной задачи оптимального управления дискретной системой: $x(t_1) \rightarrow \max$, $x(t+1) = Ax(t) + bu(t)$, $t \in T = \{0, 1, \dots, t_2-1\}$, $x(0) = 0$, $x(t_2) = 0$, $|u(t)| \leq 1$, где $x(t) \in R^n$, $A = E + H$, E — единичная матрица, H — матрица, у которой элементы над главной диагональю равны h , остальные элементы равны 0, $b = \{0, \dots, 0, h\}$. Результаты представлены в табл. 20.

Программы реализованы на языке FORTRAN-IV, трансляторы FORTRAN G и FORTRAN H ОС ЕС. Измерялось процессорное время работы подпрограммы решения задачи ЛП. В каждом сравнении участвовали программы, обработанные одним и тем же транслятором. Вычисления проводились на ЭВМ ЕС-1022.

§ 8. Блочный метод

В этом параграфе строится новый метод решения $m \times n$ -задачи линейного программирования

$$c'x \rightarrow \max, Ax = b, d_* \leq x \leq d^*. \quad (1)$$

n	t ₂	t ₁	h	Время решения, с			Число смен опоры			lg μ
				адаптивный	прямой	μ-адаптивный	адаптивный	прямой	μ-адаптивный	
3	500	475	0.25	59.24	43.14	38.16	17	69	11	18
5	500	475	0.25	271.28	161.43	59.10	113	198	22	12
8	500	475	0.25	77.54	—	70.98	17	—	16	4
8	400	397	0.25	243.20	—	175.47	53	—	37	8
9	400	380	0.25	163.65	—	156.77	38	—	35	4
8	200	199	0.30	—	—	84.92	—	—	44	7
8	200	196	0.30	177.35	—	82.16	74	—	61	7
9	200	196	0.30	67.24	—	66.38	32	—	33	8
8	100	11	0.30	—	43.10	18.44	—	94	21	15
8	100	12	0.30	10.30	44.86	10.30	12	101	12	8
8	100	20	0.30	14.72	46.84	13.98	15	104	15	14
8	100	95	0.30	22.36	16.90	14.02	45	69	33	5
8	100	96	0.30	10.30	46.30	10.30	12	107	12	8
8	100	97	0.30	10.30	53.94	10.30	12	126	12	8
8	100	98	0.30	12.12	109.67	11.34	14	259	13	8
8	100	99	0.30	51.72	—	30.06	51	—	40	8

Идея метода состоит в следующем. Пусть $\{x, A_{\text{оп}}\}$, $A_{\text{оп}} = A(I, J_{\text{оп}})$, $J_{\text{оп}} \subset J$, — опорный план задачи (1). По схеме адаптивного метода вычислим новый план $\tilde{x} = x + \Theta_0 l^0$. Пусть \tilde{x} не является ε -оптимальным планом задачи (1) и $i_0 \in J_{\text{оп}}$ — индекс, на котором достигался шаг Θ_0 . С помощью i_0 сформируем производную задачу, решение которой даст направление спуска l^1 для опорного плана $\{\tilde{x}, A_{\text{оп}}\}$. Построим новый план $\bar{x} = \tilde{x} + \Theta_1 l^1$. Пусть $i_1 \in J_{\text{оп}}$ — индекс, на котором реализовался шаг Θ_1 . В результате у опорного плана $\{\bar{x}, A_{\text{оп}}\}$ будут две критические опорные компоненты \bar{x}_{i_0} , \bar{x}_{i_1} . С учетом этих компонент построим для плана \bar{x} опору $\bar{A}_{\text{оп}}$. Новая итерация метода начинается с опорного плана $\{\bar{x}, \bar{A}_{\text{оп}}\}$.

1. Вспомогательный алгоритм. Предварительно по-

строим необходимый для дальнейшего вспомогательный алгоритм решения задачи

$$-\alpha'(K)y(K) \rightarrow \max, \quad b_* \leq -\gamma'(K)y(K) \leq b^*, \quad (2)$$

$$\beta_i \leq y_i \leq \beta^i, \quad i \in K, \quad K = \{1, 2, \dots, k\}.$$

Положим $y^1 = y^1(K)$,

$$y_i^1 = \begin{cases} \beta_i, & \text{если } \alpha_i > 0, \\ \beta^i, & \text{если } \alpha_i < 0, \\ \text{любое значение из } [\beta_i, \beta^i], & \text{если } \alpha_i = 0, \end{cases}$$

и вычислим $\kappa = -\gamma'(K)y^1(K)$. Если $\kappa \in [b_*, b^*]$, то оптимальный план y^0 задачи (2) совпадает с y^1 и на этом процесс решения задачи (2) прекращается.

В противном случае положим $\eta = \kappa - b_*$, $\mu = 1$, если $\kappa < b_*$; $\eta = b^* - \eta$, $\mu = -1$, если $\kappa > b^*$. Подсчитаем числа

$$\sigma(i) = \begin{cases} -\alpha_i/\mu\gamma_i, & \text{если } \operatorname{sign} \alpha_i \neq \operatorname{sign} \mu\gamma_i, \quad \mu\gamma_i \neq 0, \\ \infty & \text{в остальных случаях, } i \in K. \end{cases}$$

Если $\sigma(i) = \infty \quad \forall i \in K$, то ограничения задачи (2) несовместны. (Забегая вперед, заметим, что ограничения всех задач типа (2), встречающихся в предлагаемом методе, в силу существования опорного плана задачи (1), всегда совместны.) В случае, когда $\{i \in K : \sigma(i) < \infty\} \neq \emptyset$, числа $\sigma(i) < \infty$, $i \in K$, упорядочим по возрастанию: $\sigma(i_1) \leq \sigma(i_2) \leq \dots \leq \sigma(i_s)$ и вычислим η_t , $t = \overline{1, s}$: $\eta_{t-1} + |\gamma_{i_t}|(-\beta_{i_t} + y_{i_t}^1)$ при $\mu\gamma_{i_t} > 0$, $\eta_{t-1} + |\gamma_{i_t}|(\beta_{i_t} - y_{i_t}^1)$ при $\mu\gamma_{i_t} < 0$, считая $\eta_0 = \eta$.

Если $\eta_s < 0$, то ограничения задачи (2) несовместны. В противном случае существует такой индекс s_* , $1 \leq s_* \leq s$, что $\eta_{s_*-1} < 0$, $\eta_{s_*} \geq 0$. Компоненты оптимального плана задачи (2) $y_i^0 = y_i^1$, $i \in \{i_{s_*+1}, \dots, i_s\} \cup \{i \in K : \sigma(i) = \infty\}$,

$$y_i^0 = \begin{cases} \beta^i, & \text{если } \mu\gamma_i < 0, \\ \beta_i, & \text{если } \mu\gamma_i > 0, \quad i \in \{i_1, \dots, i_{s_*-1}\}, \end{cases}$$

$$y_{i_{s_*}}^0 = y_{i_{s_*}}^1 + \eta_{s_*-1}/\mu\gamma_{i_{s_*}}.$$

На этом заканчивается работа вспомогательного алгоритма.

2. Итерация. Пусть $\{x, J_{\text{оп}}\}$, $J_{\text{оп}} \subset J$, — начальный опорный план задачи (1). Вычислим вектор оценок $\Delta =$

$= \Delta(J): \Delta'(J) = u'(I)A(I, J) - c'(J), \quad u'(I) = c'(J_{\text{оп}})A_{\text{оп}}^{-1},$
 где $A_{\text{оп}}^{-1} = [A(I, J_{\text{оп}})]^{-1}$. Если $\beta(x, J_{\text{оп}}) = \sum_{\Delta_j < 0} \Delta_j (x_j - d_j^*) +$
 $+ \sum_{\Delta_j > 0} \Delta_j (x_j - d_j^*) \leq \varepsilon$, где ε — заданное неотрицательное
 число, то x — ε -оптимальный план задачи (1). Предполо-
 жим, что $\beta(x, J_{\text{оп}}) > \varepsilon$.

Шаг 1. Построим вектор $l^0 = l^0(J)$:

$$l_j^0 = \begin{cases} d_j^* - x_j & \text{при } \Delta_j < 0, \\ 0 & \text{при } \Delta_j = 0, \\ d_{*j} - x_j & \text{при } \Delta_j > 0, \quad j \in J_{\text{н}}, \end{cases}$$

$$l^0(J_{\text{оп}}) = -A_{\text{оп}}^{-1}(J_{\text{оп}}, I)A(I, J_{\text{н}})l^0(J_{\text{н}}).$$

Найдем максимально допустимый шаг вдоль направле-
 ния l^0 из следующих соотношений:

$$\Theta_0 = \min\{1, \Theta_{i_0}\}, \quad \Theta_{i_0} = \min\{\Theta_i, i \in J_{\text{оп}}\},$$

$$\Theta_i = \begin{cases} (d_i^* - x_i)/l_i^0 & \text{при } l_i^0 > 0, \\ \infty & \text{при } l_i^0 = 0, \\ (d_{*i} - x_i)/l_i^0 & \text{при } l_i^0 < 0, \quad i \in J_{\text{оп}}. \end{cases}$$

Если $(1 - \Theta_0)\beta(x, J_{\text{оп}}) \leq \varepsilon$, то вектор $\tilde{x} = x + \Theta_0 l^0$ является
 ε -оптимальным планом задачи (1). В противном случае
 перейдем к шагу 2.

Шаг 2. Описанным в п. 1 алгоритмом решим произ-
 водную задачу

$$\beta^*(y) = -\Delta'(J_{\text{н}})y(J_{\text{н}}) \rightarrow \max,$$

$$d_{*i_0} - \tilde{x}_{i_0} \leq -X(i_0, J_{\text{н}})y(J_{\text{н}}) \leq d_{i_0}^* - \tilde{x}_{i_0},$$

$$d_{*}(J_{\text{н}}) - \tilde{x}(J_{\text{н}}) \leq y(J_{\text{н}}) \leq d^*(J_{\text{н}}) - \tilde{x}(J_{\text{н}}),$$

где $X(i, J_{\text{н}}) = A^{-1}(i, I)A(I, J_{\text{н}})$, $i \in J_{\text{оп}}$. В результате полу-
 чим оптимальный план $y^0(J_{\text{н}})$, числа $i_{s*} \in J_{\text{н}}$, $\sigma(i_{s*})$, $\mu =$
 $= \pm 1$. Перейдем к шагу 3.

Шаг 3. Построив вектор $l^1 = l^1(J)$, $l^1(J_{\text{н}}) = y^0(J_{\text{н}})$,
 $(J_{\text{оп}}) = -A_{\text{оп}}^{-1}(J_{\text{оп}}, I)A(I, J_{\text{н}})l^1(J_{\text{н}})$, вычислим новый

план задачи (1): $\bar{x} = \tilde{x} + \Theta_1 l^1$, $\Theta_1 = \min\{1, \Theta_{i_1}\}$, $\Theta_{i_1} =$
 $= \min\{\Theta_i, i \in J_{\text{оп}}\}$,

$$\Theta_i = \begin{cases} (d_i^* - \tilde{x}_i)/l_i^1 & \text{при } l_i^1 > 0, \\ \infty & \text{при } l_i^1 = 0, \\ (d_{*i} - \tilde{x}_i)/l_i^1 & \text{при } l_i^1 < 0, \quad i \in J_{\text{оп}}. \end{cases}$$

Если $(1-\Theta_1)\beta^*(y^0) \leq \varepsilon$, то \bar{x} — ε -оптимальный план *). В противном случае положим $j_0=i_0$, $j_1=i_1$, $K_H=J_H$, $z_0(K_H)=X(i_0, K_H)$, $z_1(K_H)=X(i_1, K_H)$ и перейдем к шагу 4.

Шаг 4. С помощью вспомогательного алгоритма (см. п. 1) решим задачу

$$\begin{aligned} \beta^0(y) &= -\Delta'(K_H) y(K_H) \rightarrow \max, \\ d_{*i_1} - \bar{x}_{i_1} &\leq -z'_1(K_H) y(K_H) \leq d_{i_1}^* - \bar{x}_{i_1}, \\ d_*(K_H) - \bar{x}(K_H) &\leq y(K_H) \leq d^*(K_H) - \bar{x}(K_H), \end{aligned} \quad (3)$$

исходя из начальной информации $y^1(K_H) = x(K_H) + l^0(K_H) = \bar{x}(K_H) + (1-\Theta_0)l^0(K_H)$. В результате получим оптимальный план $y^0(K_H)$, числа i_{s*} , $\sigma(i_{s*})$, $\mu = \pm 1$. При $\beta^0(y^0) > \varepsilon$ перейдем к шагу 5.

Шаг 5. Положим $\bar{K}_H = (K_H \setminus i_{s*}) \cup i_1$ и построим новые векторы $\bar{l}^1(\bar{K}_H)$, $\bar{z}_0(\bar{K}_H)$, $\bar{z}_1(\bar{K}_H)$, $\bar{\Delta}(\bar{K}_H)$:

$$\begin{aligned} \bar{l}_i^1 &= \begin{cases} y_i^0 & \text{при } i \in (K_H \setminus i_1), \\ d_{i_1}^* - \bar{x}_{i_1} & \text{при } i = i_1, \mu = -1, \\ d_{*i_1} - \bar{x}_{i_1} & \text{при } i = i_1, \mu = 1; \end{cases} \\ \bar{z}_0(\bar{K}_H \setminus i_1) &= z_1(\bar{K}_H \setminus i_1) / z_1(i_{s*}), \\ \bar{z}_0(i_1) &= 1 / z_1(i_{s*}), \\ \bar{z}_1(\bar{K}_H \setminus i_1) &= z_0(\bar{K}_H \setminus i_1) - z_0(i_{s*}) \bar{z}_0(\bar{K}_H \setminus i_1), \\ \bar{z}_1(i_1) &= -z_0(i_{s*}) / z_1(i_{s*}); \\ \bar{\Delta}(\bar{K}_H \setminus i_1) &= \Delta(\bar{K}_H \setminus i_1) + \mu \sigma(i_{s*}) z_1(K_H \setminus i_1), \\ \bar{\Delta}(i_1) &= \mu \sigma(i_{s*}). \end{aligned}$$

Считая $i_1=i_0$, $i_0=i_{s*}$, перейдем к шагу 6.

Шаг 6. Вычислим $\kappa = -\bar{z}'_1(\bar{K}_H) \bar{l}^1(\bar{K}_H)$. Если $d_{*i_1} - \bar{x}_{i_1} \leq \kappa \leq d_{i_1}^* - \bar{x}_{i_1}$, то перейдем к шагу 7. В противном случае возвращаемся к шагу 4, заменив y^1 , Δ , z_1 , K_H на $y^1 = \bar{l}^1$, $\bar{\Delta}$, \bar{z}_1 , \bar{K}_H .

) Обоснование этого положения следует из того, что число $\beta^(y^0)$ есть оценка субоптимальности опорного плана $\{\tilde{x}, \tilde{A}_{оп}\}$, $\tilde{A}_{оп} = A(I, \tilde{J}_{оп})$, где $\tilde{J}_{оп} = (J_{оп}/i_0) \cup i_{s*}$. На переходе от плана \tilde{x} к плану \bar{x} целевая функция задачи (1) увеличивается на величину $c'\bar{x} - c'\tilde{x} = \Theta_1 \beta^*(y^0)$, поэтому число $(1-\Theta_1)\beta^*(y^0)$ — оценка субоптимальности опорного плана $\{\bar{x}, \tilde{J}_{оп}\}$.

Шаг 7. Строим новый опорный план $\{\bar{x}, \bar{A}_{\text{оп}}\}$, $\bar{A}_{\text{оп}} = A(I, \bar{J}_{\text{оп}})$, $\bar{J}_{\text{оп}} = J/\bar{K}_H$ и начинаем новую итерацию метода с $\{\bar{x}, \bar{A}_{\text{оп}}\}$. При этом оценка субоптимальности опорного плана $\{\bar{x}, \bar{J}_{\text{оп}}\}$ равна $\beta(\bar{x}, \bar{J}_{\text{оп}}) = \beta^0(y^0)$. В качестве направления l^0 для следующей итерации целесообразно взять вектор $l^0 = l^0(J)$: $l^0(\bar{J}_H) = l^1(\bar{J}_H)$, $l^0(\bar{J}_{\text{оп}}) = -\bar{A}_{\text{оп}}^{-1}(\bar{J}_{\text{оп}}, I) A(I, \bar{J}_H) l^0(\bar{J}_H)$. Вдоль этого направления максимально допустимый шаг положителен, если среди компонент \bar{x}_i , $i \in \bar{J}_{\text{оп}} \setminus \{i_0, i_1\}$, нет критических.

3. О конечности метода. Итерацию $\{x, A_{\text{оп}}\} \rightarrow \{\bar{x}, \bar{A}_{\text{оп}}\}$ назовем регулярной, если число $\sigma(i_{s*})$, полученное на шаге 2, больше нуля. Очевидно, что описанный метод за конечное число итераций даст ε -оптимальный план задачи (1) при условии, что число нерегулярных итераций и число повторений шага 4 в процессе решения задачи (1) конечны. Однако, следуя § 4, 5, нетрудно построить несложную модификацию метода, конечную для любой задачи (1).

В основном все шаги конечной модификации метода остаются теми же, что и выше. К ним лишь добавляется следующий шаг, который по мере необходимости выполняется в процессе решения задачи (1).

Шаг 2'. (Выполняется, когда число $\sigma(i_{s*})$, полученное на шаге 2, равно нулю.) С начальной информацией \bar{x} , i_0 , l^0 , Θ_0 приступим к процедуре корректировки, описанной в § 6. Она закончится через конечное число шагов. На последнем шаге корректировки операцию замены опоры не производим. В результате получим новый опорный план $\{x^1, A_{\text{оп}}^1\}$, $A_{\text{оп}}^1 = A(I, J_{\text{оп}}^1)$, $J_{\text{оп}}^1 \subset J$; индекс $i_* \in J_{\text{оп}}^1$, который соответствует ведущей строке; направление l^1 и соответствующий ему максимально допустимый шаг Θ на последнем шаге процедуры корректировки. Приняв x^1 , $A_{\text{оп}}^1$, i_* , l^1 , Θ за \bar{x} , $A_{\text{оп}}$, i_0 , l^0 , Θ_0 , повторим шаг 2. При этом обязательно окажется, что $\sigma(i_{s*}) > 0$.

Для того, чтобы предотвратить возможность бесконечного числа повторений шага 4, введем в рассмотрение число $h \geq 1$ — параметр метода. Когда число повторений шага 4 станет больше h , то переходим к шагу 7, пропустив шаги 5 и 6.

З а м е ч а н и е. Конечность числа повторений шага 4 можно обеспечить, включив в итерацию метода шаг 4', на котором также выполняется процедура корректировки по схеме § 6. Описание соответ-

ствующих процедур оставим читателям в качестве полезного упражнения.

4. Численный эксперимент. Цель эксперимента — сравнить предложенный метод с адаптивным методом § 1. С помощью генератора GER (см. § 9) сформулированы 18 серий задач по четыре задачи в каждой. Все параметры A, b, c, d_*, d^* , а также начальный опорный план $\{x, A_{оп}\}$ построены аналогично § 7. Между числом n переменных и числом m ограничений задачи (1) выбиралось следующее соотношение: $n = m \times j + [m/i]$, где $i = 1, 3; j = 1, 2; m = 10, 12, 14; [a]$ — целое число, не превосходящее число a .

Блочный метод (метод I) и адаптивный (метод II) сравнивались по среднему числу итераций и по среднему времени решения каждой серии задач. Результаты эксперимента приведены в табл. 21.

Табл. 21

$m \times n$	Среднее число итераций		Среднее время решения, с		$m \times n$	Среднее число итераций		Среднее время решения, с	
	I	II	I	II		I	II	I	II
10×20	3.75	6.25	0.885	1.025	12×36	6.75	15.00	3.580	4.785
10×15	3.75	6.25	0.595	0.810	12×30	5.00	9.25	2.095	2.575
10×13	1.50	2.25	0.220	0.290	12×28	5.00	9.75	2.005	2.545
10×30	6.00	11.25	2.400	2.560	14×28	5.75	10.75	2.520	3.270
10×25	5.25	9.25	1.600	1.820	14×21	4.00	6.75	1.220	1.665
10×23	4.75	9.25	1.365	1.690	14×18	2.25	3.25	0.600	0.760
12×24	5.00	9.50	1.640	2.165	14×42	8.75	19.50	6.405	8.720
12×18	3.25	5.50	0.750	1.035	14×35	7.55	17.00	4.325	6.140
12×16	2.25	3.50	0.425	0.620	14×32	7.75	15.75	3.760	5.295

§ 9. Многошаговые методы

В этом параграфе строится многошаговый метод решения общей задачи линейного программирования [11, 36—38]. Для того, чтобы лучше понять его сущность, сначала опишем двухшаговый прямой метод.

1. Прямой двухшаговый метод. Рассмотрим $m \times n$ -задачу линейного программирования

$$c'x \rightarrow \max, Ax=b, d_* \leq x \leq d^*. \quad (1)$$

Пусть $\{x, A_{\text{оп}}\}$, $A_{\text{оп}}=A(I, J_{\text{оп}})$, $J_{\text{оп}} \subset J$, — начальный опорный план задачи (1), оценка субоптимальности $\beta(x, A_{\text{оп}})$ которого превышает заданное число $\varepsilon \geq 0$.

Следуя методу § 1, по опорному плану $\{x, A_{\text{оп}}\}$ построим подходящее направление $l^1 = l^1(J)$ и вычислим шаг $\Theta_1 = \Theta(l^1) = \min\{1, \Theta_{j_0}\}$, $\Theta_{j_0} = \min\{\Theta_j, j \in J_{\text{оп}}\}$, где Θ_j — максимальный шаг вдоль l^1 , допускаемый j -м опорным прямым ограничением. Процесс решения задачи (1) прекращается на ε -оптимальном плане $\tilde{x} = x + \Theta_1 l^1$, если $(1 - \Theta_1)\beta(x, A_{\text{оп}}) \leq \varepsilon$.

При $(1 - \Theta_1)\beta(x, A_{\text{оп}}) > \varepsilon$ по исходной информации $x, A_{\text{оп}}$ и вектору l^1 построим (см. § 1) новую опору $\tilde{A}_{\text{оп}} = A(I, \tilde{J}_{\text{оп}})$, $\tilde{J}_{\text{оп}} = (J_{\text{оп}} \setminus j_0) \cup J_*$, и подходящее направление $l^2 = l^2(J)$ для опорного плана $\{x + \Theta_1 l^1, \tilde{A}_{\text{оп}}\}$. Новый план \tilde{x} задачи (1) вычислим по формуле $\tilde{x} = x(z^0)$, где $x(z) = x + Lz$, $L = \{l^1, l^2\}$, $z = (z_1, z_2)$ — план, z^0 — решение задачи

$$\gamma'z \rightarrow \max, d_* - x \leq Lz \leq d^* - x \quad (\gamma' = c'L). \quad (2)$$

Задачу (2) решаем методом § 1 с начальных плана $z_0 = \{\Theta_1, 0\}$ и опорной 1×1 -матрицы $L_{\text{оп}} = L(1, j_0) = L_{j_0}^1 \neq 0$. После первой итерации получается план с $z_1^1 = \Theta(l^1)$, $z_2^1 = \Theta(l^2)$, где $\Theta(l^1)$, $\Theta(l^2)$ суть величины шагов на первых двух итерациях метода § 1, примененного к опорному плану $\{x, A_{\text{оп}}\}$. Таким образом, за один шаг целевая функция возрастает не меньше, чем за два шага метода § 1.

Оценка субоптимальности опорного плана $\{x(z), \tilde{A}_{\text{оп}}\}$ равна $\beta(x(z), \tilde{A}_{\text{оп}}) = \beta(x, \tilde{A}_{\text{оп}}) - \gamma'z = \beta(x, A_{\text{оп}}) + \Delta\beta(\tilde{A}_{\text{оп}}) - \gamma_1 z_1 - \gamma_2 z_2 = \Theta_1 \gamma_1 + \Delta\beta(\tilde{A}_{\text{оп}}) + (1 - \Theta_1)\beta(x, A_{\text{оп}}) - \gamma_1 z_1 - \gamma_2 z_2 = (\Theta_1 - z_1)\gamma_1 + (1 - z_2)\gamma_2$. Процесс решения задачи (2) прекращается, как только будет построен план z задачи (2), для которого выполняется неравенство $\beta(x(z), \tilde{A}_{\text{оп}}) \leq \varepsilon$. Тогда $x(z)$ — ε -оптимальный план задачи (1).

Специальная структура задачи (2) позволяет упростить метод § 1. Пусть на k -й итерации известен опорный план $\{z^k, L(j_k, 1)\}$. Для z^k строим подходящее направ-

ление $p^k = \{p_1^k = -l_{j_k}^2/l_{j_k}^1, p_2^k = 1\}$ и вычисляем вдоль него максимально допустимый шаг $\omega = \omega_{j_{k+1}}$, где j_{k+1} — индекс ограничения, которое первым стало активным при движении вдоль p^k . Полагаем $z^{k+1} = z^k + \omega p^k$. Если выполняется одно из условий

$$\begin{aligned} l_{j_{k+1}}^1 &= 0; \quad l_{j_{k+1}}^2/l_{j_{k+1}}^1 \leq l_{j_k}^2/l_{j_k}^1; \\ \gamma_2 - \gamma_1 l_{j_{k+1}}^2/l_{j_{k+1}}^1 &\leq 0, \end{aligned} \quad (3)$$

то z^{k+1} — оптимальный план задачи (2). В противном случае плану z^{k+1} приписываем новую опору $L(j_{k+1}, 1)$ и продолжаем решение задачи (2), исходя из опорного плана $\{z^{k+1}, L(j_{k+1}, 1)\}$.

З а м е ч а н и е. Задача (2) имеет $|J|=n$ основных ограничений. Специфика матрицы L позволяет сократить число основных ограничений задачи (2) до $m+2$. Действительно, задача (2) эквивалентна задаче

$$\gamma'z \rightarrow \max,$$

$$\begin{aligned} d_{*j} - x_j &\leq l_j^1 z_1 + l_j^2 z_2 \leq d_j^* - x_j, \quad j \in \tilde{J}_{\text{оп}} \cup j_*; \\ -\xi_{j_*} &\leq z_1 \leq \Theta_1. \end{aligned} \quad (4)$$

Здесь $\xi_{j_*} = \min \xi_j$, $j \in \tilde{J}_n: l_j^1 \neq 0$, $\text{sign } l_j^1 \neq \text{sign } l_j^2$, $\tilde{J}_n = J \setminus \tilde{J}_{\text{оп}}$,

$$\xi_j = \begin{cases} (-d_j^* + x_j)/l_j^1, & \text{если } l_j^1 < 0, \\ \infty, & \text{если } l_j^1 = 0, l_j^2 \neq 0, \\ (x_j - d_{*j})/l_j^1, & \text{если } l_j^1 > 0. \end{cases}$$

Индекс j_* определяется следующим образом. Если $\tilde{J}_n^* = \{j \in \tilde{J}_n: l_j^2 \neq 0, \text{sign } l_j^1 \neq \text{sign } l_j^2\} \neq \emptyset$, то $\xi_{j_*} = \min \xi_j$, $j \in \tilde{J}_n^*$. При $\tilde{J}_n^* = \emptyset$ в качестве j_* выбирается любой элемент $j \in \tilde{J}_n$, для которого $l_j^2 \neq 0$.

Эквивалентность задач (2), (4) следует из того, что множества планов этих задач, на которых значения целевой функции $\gamma'z$ не меньше $\gamma'z_0$, совпадают. Последнее нетрудно показать геометрическим путем, замечая при этом, что $\gamma > 0$ и точка $z = \{\Theta_1, 1\}$ удовлетворяет всем ограничениям: $d_{*j} - x_j \leq z_1 l_j^1 + z_2 l_j^2 \leq d_j^* - x_j$, $j \in \tilde{J}_n$.

Лемма 1. Если для плана z^{k+1} , $k \geq 0$, задачи (2) ни одно из условий (3) не выполняется, то $j_{k+1} \in \tilde{J}_{\text{оп}}^*$, при-

чем при $x_{j_{k+1}}(z^{k+1}) = d_{j_{k+1}}^*$ выполняются неравенства $\kappa_{j_{k+1}} > d_{j_{k+1}}^*, l_{j_{k+1}}^2 > 0$, а при $x_{j_{k+1}}(z^{k+1}) = d_{*j_{k+1}}$ — неравенства $\kappa_{j_{k+1}} < d_{*j_{k+1}}, l_{j_{k+1}}^2 < 0$, где $\tilde{J}_{\text{оп}}^* = \{j \in \tilde{J}_{\text{оп}} : \kappa_j = x_j + \Theta_1 l_j^1 + l_j^2 \in [d_{*j}, d_j^*]\}$.

Доказательство проведем по индукции. Для $k=0$ справедливость леммы очевидна. Пусть лемма верна для $k=s$. Покажем, что она верна и для $k=s+1$. Предположим, что это не так. Для определенности пусть $x_{j_{s+1}}(z^{s+1}) = d_{j_{s+1}}^*$, но $\kappa_{j_{s+1}} \leq d_{j_{s+1}}^*$ и пусть $x_{j_s}(z^s) = d_{*j_s}$. Тогда:

$$\begin{aligned} z_1^{s+1} l_{j_s}^1 + z_2^{s+1} l_{j_s}^2 &= d_{*j_s} - x_{j_s}, \\ z_1^{s+1} l_{j_{s+1}}^1 + z_2^{s+1} l_{j_{s+1}}^2 &= d_{j_{s+1}}^* - x_{j_{s+1}}; \end{aligned} \quad (5)$$

$$\begin{aligned} \Theta_1 l_{j_s}^1 + l_{j_s}^2 &< d_{*j_s} - x_{j_s}, \\ \Theta_1 l_{j_{s+1}}^1 + l_{j_{s+1}}^2 &\leq d_{j_{s+1}}^* - x_{j_{s+1}}. \end{aligned} \quad (6)$$

Из (5), (6) следует, что

$$\begin{aligned} z_1^{s+1} l_{j_s}^1 + z_2^{s+1} l_{j_s}^2 &> \Theta_1 l_{j_s}^1 + l_{j_s}^2, \\ z_1^{s+1} l_{j_{s+1}}^1 + z_2^{s+1} l_{j_{s+1}}^2 &\geq \Theta_1 l_{j_{s+1}}^1 + l_{j_{s+1}}^2. \end{aligned} \quad (7)$$

В силу того, что z^{s+1} не является оптимальным планом задачи (2), решение $\{u_1, u_2\}$ уравнений

$$u_1 l_{j_s}^1 + u_2 l_{j_{s+1}}^1 = \gamma_1, \quad u_1 l_{j_s}^2 + u_2 l_{j_{s+1}}^2 = \gamma_2 \quad (8)$$

удовлетворяет неравенствам $u_1 > 0, u_2 \geq 0$. Умножив неравенства (7) соответственно на u_1, u_2 и сложив их, с учетом (8) получим $z_1 \gamma_1 + z_2 \gamma_2 > \Theta_1 \gamma_1 + \gamma_2$. Последнее противоречит тому, что $\gamma' z \leq \Theta_1 \gamma_1 + \gamma_2$ для любого плана z задачи (2). Лемма доказана.

Из доказанной леммы следует

Лемма 2. Решение задачи (2) получается не более чем за $|\tilde{J}_{\text{оп}}| + 1$ итераций.

Замечание. Лемму 1 в случае, когда $\exists j \in \tilde{J}_{\text{н}}, l_j^1 l_j^2 < 0$, можно усилить. А именно, если для плана $z^{k+1}, k \geq 0$, задачи (2) выполняются условия $j_{k+1} \in \tilde{J}_{\text{н}}, l_{j_{k+1}}^1 l_{j_{k+1}}^2 \neq 0$, то $x(z^{k+1})$ —

оптимальный план задачи (1). Доказательство этого утверждения нетрудно провести от противного, замечая, что в этом случае $z_1 \leq \Theta_1$, $z_2 \leq 1$.

Вспомогательную задачу (2) необязательно решать до конца. Если на k -й итерации решения задачи (2) все условия (3) нарушаются, то для опорного плана $\{z^{k+1}, L(j_{k+1}, 1)\}$ выполняется неравенство $\gamma' z^0 - \gamma' z^{k+1} \leq \leq \eta(z^{k+1}, j_{k+1})$, где $\eta(z^{k+1}, j_{k+1}) = u_{j_*} \bar{d}_{j_*}$; $u_{j_*} = (\gamma_1 l_{j_{k+1}}^2 - \gamma_2 l_{j_{k+1}}^1) / (l_{j_{k+1}}^1 l_{j_{k+1}}^2 - l_{j_*}^2 l_{j_{k+1}}^1)$, $\bar{d}_{j_*} = d_{j_*} - x_{j_*}(z^{k+1})$ при $u_{j_*} < 0$; $\bar{d}_{j_*} = d_{j_*}^* - x_{j_*}(z^{k+1})$ при $u_{j_*} > 0$. При этом $\eta(z^{k+1}, j_{k+1}) \leq \eta(z^k, j_k)$.

В случае $\beta(x(z^{k+1}), \hat{A}_{\text{оп}}) - \eta(z^{k+1}, j_{k+1}) > \varepsilon^*)$ эффект для задачи (1) от дальнейшего решения задачи (2) из-за плохой опоры $\hat{A}_{\text{оп}}$ незначителен, поэтому процесс решения задачи (2) можно остановить и перейти к построению новой опоры задачи (1).

Предположим, что процесс решения задачи (2) остановлен на опорном плане $\{z^s, L(j_s, 1)\}$, $s \geq 1$. Новый план задачи (1) строим в виде $\bar{x} = x(z^s)$. По построению, компоненты \bar{x}_{j_s} , $\bar{x}_{j_{s-1}}$ плана \bar{x} — критические. Если $s = 1$, то $j_0 \in \bar{\tilde{J}}_{\text{оп}}$, $j_1 \in \tilde{J}_{\text{оп}}^*$. Элемент j_1 выводим из опорного множества по правилам метода § 1. При $s > 1$ различаем два случая: 1) $j_s \in \bar{\tilde{J}}_{\text{оп}}$, 2) $j_s \in \tilde{J}_{\text{оп}}$. В случае 1) при выполнении условий замечания план \bar{x} оптимален в задаче (1). Если эти условия замечания не выполняются, то элемент j_{s-1} выводим из опорного множества по правилам § 1. В случае 2) элементы j_s , j_{s-1} выводим из опорного множества по правилам, описанным в п. 3.

2. Численный эксперимент. Цель эксперимента — сравнить двухшаговый (метод I) и адаптивный методы (метод II) решения задачи (1). Для этого было сформировано 9 серий задач (1), по 5 задач в каждой. Параметры задачи (1) генерировались датчиком случайных чисел, равномерно распределенных в интервале $[-100, 100]$. Датчик реализован в виде следующей FORTRAN-подпрограммы:

SUBROUTINE GER (Y, Z)

L = Y * Z

*) Вместо ε здесь можно взять $\mu(\varepsilon) \geq 0$,

```

Y = Y * Z - L
Y = 100 * SIN (3.1416 * Y + 0.5)
Z = Z + 0.4141
RETURN
END

```

Он генерирует последовательность чисел y , которая используется для заполнения вектора s и матрицы A . Затем этим же датчиком генерировались тройки чисел, меньшее из которых принималось за d_{*j} , среднее — за x_j (компонента начального плана), большее — за d_j^* . По этим данным вычислялся вектор $b = Ax$. Начальные опорные индексы задавались числами $\{1, 2, \dots, m\}$. Каждая сгенерированная задача решалась сначала методом I, затем методом II. Критериями сравнения служили число итераций и время решения задачи (1). Выбор этих критериев определяется тем, что сравнение по числу итераций важно с точки зрения накопления ошибок округления, вызываемых преобразованием опор. Из п. 1 ясно, что число итераций метода I меньше числа итераций метода II, по крайней мере, в 2 раза, однако каждая итерация в методе I сложнее, чем в методе II, поэтому сравнивалось и время решения задачи (1) обоими методами. Результаты эксперимента приведены в табл. 22.

Установлено, что среднее число итераций в методе I

Табл. 22

$m \times n$	Среднее число итераций		Среднее время решения, с	
	I	II	I	II
10×30	6.80	23.80	4.04	7.94
10×40	11.00	33.80	8.184	14.716
10×50	11.80	39.40	11.244	20.828
15×30	08.50	18.20	6.510	9.144
15×40	11.60	30.00	11.668	19.916
15×50	20.00	47.60	23.84	37.68
20×30	6.00	14.80	6.964	10.912
20×40	11.60	27.60	15.852	25.072
20×50	17.40	46.40	28.82	50.164

меньше, чем в методе II, в 2—3,5 раза, а среднее время — в 1,5—1,8 раза.

3. Многошаговый метод. В этом пункте обоснован s -шаговый прямой метод решения задачи (1), который при $s=2$ совпадает с двухшаговым методом п. 1.

Пусть $\{x, A_{\text{оп}}\}$, $A_{\text{оп}}=A(I, J_{\text{оп}})$, $J_{\text{оп}} \subset J$, — начальный опорный план, оценка субоптимальности $\beta(x, A_{\text{оп}})$ которого больше заданного числа $\varepsilon \geq 0$. Алгоритм s -шагового метода строим индуктивным образом.

Следуя § 1, строим для опорного плана $\{x, A_{\text{оп}}\}$ подходящее направление $l^1 = l^1(J)$ и найдем максимально допустимый шаг Θ_0 вдоль направления l^1 :

$$\Theta_0 = \min\{1, \Theta_j, j \in J_{\text{оп}}\},$$

$$\Theta_j = \begin{cases} (d_j^* - x_j)/l_j^1, & l_j^1 > 0, \\ \infty, & l_j^1 = 0, \\ (d_{*j} - x_j)/l_j^1, & l_j^1 < 0. \end{cases}$$

Процесс решения задачи (1) прекращаем на ε -оптимальном плане $\tilde{x} = x + \Theta_0 l^1$, если $\beta(\tilde{x}, A_{\text{оп}}) = (1 - \Theta_0) \times \beta(x, A_{\text{оп}}) \leq \varepsilon$. Пусть $\beta(\tilde{x}, A_{\text{оп}}) > \varepsilon$ и $\Theta_0 = \Theta_{j_0}$. Нетрудно видеть, что совокупность $\{\Theta_0, l_{j_0}^1\}$ является оптимальным опорным планом задачи

$$\gamma_1 z_1 \rightarrow \max, \quad d_* - x \leq l^1 z_1 \leq d^* - x \quad (\gamma_1 = c' l^1).$$

Положим $k = 1$, $J_{\text{оп}}^{(0)} = J_{\text{оп}}$, $\gamma^{(1)} = \{\gamma_1\}$, $L^{(1)} = \{l^1\}$, $z^{(1)} = \{z_1\}$, $I^{(1)} = \{1\}$, $A_{\text{оп}}^{(0)} = A(I, J_{\text{оп}}^{(0)})$, $\bar{z}^{(1)} = \{\Theta_0\}$, $\bar{L}_{\text{оп}}^{(1)} = \{l_{j_0}^1\}$.

Опишем переход от k -го шага алгоритма к $(k+1)$ -му шагу. Пусть для натурального числа k известны $J_{\text{оп}}^{(k-1)}$, $\gamma^{(k)} = \{\gamma_1, \gamma_2, \dots, \gamma_k\}$, $\gamma_i = c' l^i$, $I^{(k)} = \{1, 2, \dots, k\}$, $A_{\text{оп}}^{(k-1)}$, $L^{(k)} = \{l^1, l^2, \dots, l^k\}$ и $\{\bar{z}^{(k)}, \bar{L}_{\text{оп}}^{(k)}\}$ — оптимальный опорный план задачи

$$\gamma^{(k)'} z^{(k)} \rightarrow \max, \quad d_* - x \leq L^{(k)} z^{(k)} \leq d^* - x, \quad (9)$$

причем $\bar{L}_{\text{оп}}^{(k)}$ — невырожденная $k \times k$ -подматрица матрицы $L^{(k)}$, составленная из активных ограничений задачи (9) на плане $\bar{z}^{(k)}$. Для $k = 1$ эти данные приведены выше.

Направление $l^{(k+1)}$ для $(k+1)$ -го шага алгоритма построим следующим образом. Пусть J_a — набор индексов активных ограничений задачи (9) на плане $\bar{z}^{(k)}$ и

$\{x^*, A_{\text{оп}}^*\}$, $A_{\text{оп}}^* = A(I, J_{\text{оп}}^*)$, — оптимальный опорный план задачи

$$c'x \rightarrow \max, Ax = b, d_*(\tilde{J}) \leq x(\tilde{J}) \leq d^*(\tilde{J}),$$

$$\tilde{J} = J_{\text{н}}^{(k-1)} \cup J_a, J_{\text{н}}^{(k-1)} = J \setminus J_{\text{оп}}^{(k-1)}. \quad (10)$$

Положим $l^{(k+1)} = x^* - x - L^{(k)}z^{(k)}$, $A_{\text{оп}}^{(k)} = A_{\text{оп}}^*$, $J_{\text{оп}}^{(k)} = J_{\text{оп}}^*$, $\gamma^{(k+1)} = \{\gamma^{(k)}, \gamma_{k+1} = c'l^{(k+1)}\}$. На этом заканчивается переход от k -го к $(k+1)$ -му шагу алгоритма. Отметим, что оценка субоптимальности опорного плана $\{x^{(k)}, A_{\text{оп}}^{(k)}\}$, где $x^{(k)} = x + L^{(k)}\bar{z}^{(k)}$, задачи (1) $\beta(x^{(k)}, A_{\text{оп}}^{(k)}) = c'x^* - c'x^{(k)}$. Как только получено неравенство $\beta(x^{(k)}, A_{\text{оп}}^{(k)}) \leq \varepsilon$, процесс решения задачи (1) можно прервать, не доходя до требуемого s -го шага алгоритма (тогда $x^{(k)}$ является ε -оптимальным планом задачи (1)).

Остановимся подробнее на s -м шаге алгоритма. Известны: векторы l^1, l^2, \dots, l^s ; $\{z^{(s-1)}, L_{\text{оп}}^{(s-1)}\}$ — оптимальный опорный план задачи (2) при $k = s-1$, причем $\bar{L}^{(s-1)} = L^{(s-1)}(J_a^{(s-1)}, I^{(s-1)})$, где $J_a^{(s-1)}$ — множество индексов активных ограничений задачи (2) на плане $\bar{z}^{(s-1)}$, $|J_a^{(s-1)}| = s-1$; $A_{\text{оп}}^{(s-1)} = A(I, J_{\text{оп}}^{(s-1)})$. Для упрощения записей индекс s в $L^{(s)}, z^{(s)}, \gamma^{(s)}$ и т. д. будем опускать.

Новый план задачи (1) построим по формуле $\bar{x} = x(z^0) = x + Lz^0$, где z^0 — решение задачи

$$\gamma'z \rightarrow \max_z, d_* - x \leq Lz \leq d^* - x. \quad (11)$$

Оценка субоптимальности для опорного плана $\{x(z), A_{\text{оп}}^{(s-1)}\}$ задачи (1) удовлетворяет соотношению

$$\beta(x(z), A_{\text{оп}}^{(s-1)}) = \beta(x, A_{\text{оп}}^{(s-1)}) - \gamma'z = \beta(x^{(s-1)}, A_{\text{оп}}^{(s-1)}) + \sum_{i=1}^{s-1} \gamma_i \bar{z}_i^{(s-1)} - \gamma'z = \sum_{i=1}^{s-1} \gamma_i (\bar{z}_i^{(s-1)} - z_i) + (1 - z_s) \gamma_s.$$

Как только будет найден такой план z задачи (11), что $\beta(x(z), A_{\text{оп}}^{(s-1)}) \leq \varepsilon$, процесс решения задачи (11) прекращается. Полученный план $x(z)$ является ε -оптимальным планом задачи (1).

Процесс решения задачи (11) начинается с начальных плана $z = \{\bar{z}^{(s-1)}, 0\}$ и опорной $(s-1) \times (s-1)$ -матри-

цы $L_{\text{оп}} = \bar{L}_{\text{оп}}^{(s-1)}$ и сводится к следующим операциям. Вычисляется вектор потенциалов $v' = \gamma' (I^{(s-1)}) L_{\text{оп}}^{-1}$ и оценка Δ_s вектора l^s : $\Delta_s = \sum_{j \in J_a^{\text{оп}}} v_j l_j^s - \gamma_s$, где $J_a^{\text{оп}}$ — множество

опорных (активных) индексов задачи (11) на плане z . Для плана z строится подходящее направление $p = \{-L_{\text{оп}}^{-1} l^s (J_a^{\text{оп}}), 1\}$ и вдоль него находится максимально допустимый шаг $\omega_0 = \omega_{j_*}$, где j_* — индекс ограничения, первым ставшего активным при движении вдоль направления p . Далее вычисляется число $\sigma_0 = \min \{\sigma^s, \sigma_j, j \in J_a^{\text{оп}}\}$, где

$$\sigma_i = \begin{cases} -v_j / k u_j, & \text{если } k v_j u_j < 0, \\ 0, & \text{если } v_j = 0, k u_j > 0, j \in J_{a-}^{\text{оп}}, \\ 0, & \text{если } v_j = 0, k u_j < 0, j \in J_{a+}^{\text{оп}}, \\ \infty & \text{в остальных случаях, } j \in J_a^{\text{оп}}; \end{cases}$$

$u' = -L(j_*, I^{(s-1)}) L^{-1}(I^{(s-1)}, J_a^{\text{оп}})$; $k = 1$ при $L(j_*, I^{(s-1)})(z + \omega_0 p) = d_{j_*}^* - x_{j_*}$; $k = -1$ при $L(j_*, I^{(s-1)}) \times \times (z + \omega_0 p) = d_{*j_*} - x_{j_*}$;

$$\sigma^s = -\Delta_s / k u^s \geq 0^*,$$

где $u^s = l_{j_*}^s - L(j_*, I^{(s-1)}) L_{\text{оп}}^{-1} l^s (J_a^{\text{оп}})$; $J_{a-}^{\text{оп}} = \{j \in J_a^{\text{оп}} : x_j(z) = d_{*j} - x_j\}$; $J_{a+}^{\text{оп}} = J_a^{\text{оп}} \setminus J_{a-}^{\text{оп}}$.

Если $\sigma_0 = \sigma^s$, то план $\tilde{z} = z + \omega_0 p$ является оптимальным в задаче (11). В противном случае плану \tilde{z} приписывается опора $\tilde{L}_{\text{оп}} = L(\tilde{J}_a^{\text{оп}}, I^{(s-1)})$, где $\tilde{J}_a^{\text{оп}} = (J_a^{\text{оп}} \cup j_*) \setminus j_0$. Здесь j_0 — такой индекс, что $\sigma_{j_0} = \min \{\sigma_j, j \in J_a^{\text{оп}}\}$. Далее процесс решения задачи (11) продолжается, исходя из опорного плана $\{\tilde{z}, \tilde{L}_{\text{оп}}\}$.

З а м е ч а н и е. Процедура решения задачи (9) аналогична описанной для задачи (11). Для получения оптимальной опоры $\bar{L}_{\text{оп}}^{(k)}$ необходимо к опорному множеству $J_a^{\text{оп}}$ добавлять элемент j_* — номер ограничения, которое стало активным на последней итерации решения задачи (9).

Процесс решения задачи (11) можно прервать на любой итерации. Нетрудно показать, что

*) Это неравенство следует из того, что $\Delta_s \leq 0$, а $u^s > 0$ при $k = 1$; $u^s < 0$ при $k = -1$.

$$\gamma' z_0 - \gamma' \tilde{z} \leq \tilde{v}' (J_a^{\text{оп}}) \tilde{d} (J_a^{\text{оп}}) = \eta (\tilde{z}), \quad (12)$$

где $\tilde{v}' (J_a^{\text{оп}}) = v' (J_a^{\text{оп}}) - v' (J_a^{\text{оп}}) l^s (J_a^{\text{оп}}) L (j_*, I^{(s-1)}) L^{-1} (I^{(s-1)}, J_a^{\text{оп}}) / u^s = v' (J_a^{\text{оп}}) + v' (J_a^{\text{оп}}) l^s (J_a^{\text{оп}}) u' (J_a^{\text{оп}}) / u^s$; $\tilde{d}_j = d_j^* - x_j (\tilde{z})$ при $\tilde{v}_j > 0$; $\tilde{d}_j = d_{*j} - x_j (\tilde{z})$ при $\tilde{v}_j \leq 0$, $j \in J_a^{\text{оп}}$.

Справедливость неравенства (12) вытекает из того, что $\eta (\tilde{z})$ является не чем иным, как оценкой субоптимальности опорного плана $\{\tilde{z}, L (I^{(s)}, J_a^{\text{оп}}) \cup j_*\}$ задачи (11)

(см. § 1). Когда $\eta (\tilde{z})$ достаточно мало, процесс решения задачи (11) можно остановить и перейти к построению новой опоры для плана $x (\tilde{z})$.

Предположим, что процесс решения задачи (11) остановлен на плане \tilde{z} . Новый план задачи (1) строим в виде $\bar{x} = x (\tilde{z})$. Пусть J_a — набор индексов активных ограничений на плане \tilde{z} задачи (4). Новую опору $\tilde{A}_{\text{оп}}$ плана \bar{x} строим следующим образом. Рассмотрим задачу

$$c'x \rightarrow \max, Ax = b, d_*(\tilde{J}) \leq x(\tilde{J}) \leq d^*(\tilde{J}), \quad (13)$$

где $\tilde{J} = J_H^{(s-1)} \cup J_a$, $J_H^{(s-1)} = J \setminus J_{\text{оп}}^{(s-1)}$.

Задачу (13) решаем, исходя из опорного плана $\{\bar{x}, A_{\text{оп}}^{(s-1)}\}$. Пусть $\{x^*, A_{\text{оп}}^*\}$ — оптимальный опорный план задачи (13). Плану \bar{x} приписываем опору $\bar{A}_{\text{оп}} = A_{\text{оп}}^*$. На этом завершается одна итерация s -шагового алгоритма. В качестве подходящего направления l^1 для следующей итерации возьмем вектор $l^1 = x^* - \bar{x}$. Вместо задачи (13) можно решать задачу

$$- \sum_{j \in J_H^{(s-1)}} \tilde{\Delta}_j l_j \rightarrow \max,$$

$$d_{*i} - \bar{x}_i \leq - \sum_{j \in J_H^{(s-1)}} x_{ij} l_j \leq d_i^* - \bar{x}_i, i \in I^*, \quad (14)$$

$$d_{*j} - \bar{x}_j \leq l_j \leq d_j^* - \bar{x}_j, j \in J_H^{(s-1)}, I^* = J_a \cap J_{\text{оп}}^{(s-1)},$$

$$x(i, J_H^{(s-1)}) = \tilde{A}_{\text{оп}}^{-1}(i, I) A(I, J_H^{(s-1)}), \tilde{A}_{\text{оп}} = A_{\text{оп}}^{(s-1)}.$$

Здесь $\tilde{\Delta}$ — вектор оценок, вычисленный по опоре $A_{\text{оп}}^{(s-1)}$ задачи (1). Задачу (14) целесообразно решать двойственным методом, исходя из начального двойственного плана $y(I^*) = 0$; $w_j = -\tilde{\Delta}_j$, $v_j = 0$ при $\tilde{\Delta}_j < 0$; $w_j = 0$; $v_j = \tilde{\Delta}_j$ при $\tilde{\Delta}_j \geq 0$, $j \in J_{\text{н}}^{(s-1)}$, и начальных опорных множеств $I_{\text{оп}}^* = \emptyset$, $J_{\text{н}}^{\text{оп}} = \emptyset$.

Пусть $\bar{I}_{\text{оп}}^*$, $\bar{J}_{\text{н}}^{\text{оп}}$ — оптимальные опорные множества задачи (14). Плану \bar{x} припишем новую опору $\bar{A}_{\text{оп}} = A(I, \bar{J}_{\text{оп}})$, $\bar{J}_{\text{оп}} = (J_{\text{оп}}^{(s-1)} \cup \bar{J}_{\text{н}}^{\text{оп}}) \setminus I_{\text{оп}}^*$. В качестве направления l^1 для следующей итерации возьмем вектор $l^1 = l^1(J)$, $l^1(J_{\text{н}}^{(s-1)}) = \bar{l}(J_{\text{н}}^{(s-1)})$ ($\bar{l}(J_{\text{н}}^{(s-1)})$ — решение задачи (14)), $l^1(J_{\text{оп}}^{(s-1)}) = -A_{\text{оп}}^{-1} A(I, J_{\text{н}}^{(s-1)}) l^1(J_{\text{н}}^{(s-1)})$.

Оценка субоптимальности для опорного плана $\{\bar{x}, \bar{A}_{\text{оп}}\}$ равна $\beta(\bar{x}, \bar{A}_{\text{оп}}) = c'x^* - c'\bar{x} = \bar{\beta}$, где $\bar{\beta}$ — оптимальное значение целевой функции задачи (14). При этом имеют место соотношения $\beta(\bar{x}, \bar{A}_{\text{оп}}) = \beta(\bar{x}, A_{\text{оп}}^{(s-1)}) - \Delta\beta(\bar{A}_{\text{оп}}) \leq \leq \beta(x, A_{\text{оп}})$. Число $-\Delta\beta(\bar{A}_{\text{оп}}) \leq 0$ — приращение двойственной целевой функции при переходе от двойственного плана, сопровождающего опору $A_{\text{оп}}^{(s-1)}$ к двойственному плану, сопровождающему опору $\bar{A}_{\text{оп}}$.

Введем понятие регулярности. Итерация $\{x, A_{\text{оп}}\} \rightarrow \rightarrow \{\bar{x}, \bar{A}_{\text{оп}}\}$ регулярна, если $\beta(\bar{A}_{\text{оп}}) < \beta(A_{\text{оп}})$, где $\beta(A_{\text{оп}})$ — мера неоптимальности опоры $A_{\text{оп}}$. По стандартной схеме нетрудно доказать, что справедлива

Теорема 1. Для любого числа $\varepsilon \geq 0$ и любого начального опорного плана $\{x, A_{\text{оп}}\}$ описанный алгоритм за конечное число итераций строит ε -оптимальный план, если все его итерации, за исключением конечного числа, регулярны.

З а м е ч а н и е. Следуя § 6, 7, можно построить и конечные версии описанного алгоритма, в которых никакие свойства регулярности не используются.

4. Двухшаговый двойственный метод. В этом пункте строится двойственный аналог двухшагового прямого метода из п. 1 для решения задачи (1).

Двойственной к (1) задачей является

$$\begin{aligned} b'y - d_*'v + d^{*'}w &\rightarrow \min, \\ A'y + w - v &= c, v \geq 0, w \geq 0. \end{aligned} \quad (15)$$

Вектор $\delta = A'y - c$ называется копланом, пара $\{\delta, A_{\text{оп}}\}$ из коплана δ и опоры $A_{\text{оп}}$ — опорным копланом. Предположим, что условие согласования § 1 выполняется для каждого двойственного плана $\{y, v, w\}$. Пусть $\{\delta, A_{\text{оп}}\}$, $A_{\text{оп}} = A(I, J_{\text{оп}})$, $J_{\text{оп}} \subset J$, — начальный опорный коплан задачи (1). Построим соответствующий псевдоплан κ :

$$\kappa_j = \begin{cases} d_j^* & \text{при } \delta_j < 0, \\ d_{*j} & \text{при } \delta_j \geq 0, \quad j \in J_{\text{н}} = J \setminus J_{\text{оп}}; \end{cases}$$

$$\kappa(J_{\text{оп}}) = A_{\text{оп}}^{-1} (b - A(I, J_{\text{н}}) \kappa(J_{\text{н}})).$$

Известно, что при $d_*(J_{\text{оп}}) \leq \kappa(J_{\text{оп}}) \leq d^*(J_{\text{оп}})$ псевдоплан κ является β -оптимальным планом задачи (1), где $\beta = \beta(\kappa, A_{\text{оп}})$: $\beta(x, A_{\text{оп}}) = \sum_{\delta_j > 0} \delta_j (\kappa_j - d_{*j}) + \sum_{\delta_j < 0} \delta_j (\kappa_j - d_j^*)$.

Если при этом имеет место неравенство $\beta \leq \varepsilon$ (ε — заданное неотрицательное число), то решение задачи (1) прекращается на ε -оптимальном плане κ . Дальнейшее изложение связано со случаем, когда либо псевдоплан κ не является планом задачи (1), либо псевдоплан κ — план задачи (1), но $\beta(\kappa, A_{\text{оп}}) > \varepsilon$. Тогда найдется такой индекс $j_1 \in J_{\text{оп}}$, что $\kappa_{j_1} \notin [\bar{d}_{*j_1}, \bar{d}_{j_1}^*]$. Здесь $\bar{d}_{*j_1} = \bar{d}_{j_1}^* = d_{*j_1}$ при $\delta_{j_1} > 0$; $\bar{d}_{*j_1} = \bar{d}_{j_1}^* = d_{j_1}^*$ при $\delta_{j_1} < 0$; $\bar{d}_{*j_1} = d_{*j_1}$, $\bar{d}_{j_1}^* = d_{j_1}^*$ при $\delta_{j_1} = 0$. Положим $\mu_1 = 1$, если $\kappa_{j_1} < d_{*j_1}$; $\mu_1 = -1$, если $\kappa_{j_1} > \bar{d}_{j_1}^*$. Направлением изменения опорного коплана $\{\delta, A_{\text{оп}}\}$, при движении вдоль которого целевая двойственная функция уменьшается, является вектор $\Delta^1 \delta = \Delta^1 \delta(J)$: $\Delta^1 \delta_j = 0$, $j \in J_{\text{оп}} \setminus j_1$, $\Delta^1 \delta_{j_1} = \mu_1$, $\Delta^1 \delta'(J_{\text{н}}) = \Delta^1 \delta'(J_{\text{оп}}) A_{\text{оп}}^{-1}(J_{\text{оп}}, I) A(I, J_{\text{н}})$.

Вдоль этого направления сделаем длинный двойственный шаг (см. § 1) (т. е. движемся по направлению $\Delta^1 \delta$ до тех пор, пока скорость уменьшения целевой функции задачи (15) не станет больше нуля). При этом возможны случаи: 1) максимально допустимый шаг σ_1 вдоль направления $\Delta^1 \delta$ равен бесконечности (тогда ограничения задачи (1) несовместны); 2) $\sigma_1 < +\infty$.

Пусть j_0 — индекс, на котором реализовался этот шаг. Вычислим новый коплан $\tilde{\delta} = \delta + \sigma_1 \Delta^1 \delta$. Пусть для опорного коплана $\{\tilde{\delta}, \tilde{A}_{\text{оп}}\}$, где $\tilde{A}_{\text{оп}} = A(I, \tilde{J}_{\text{оп}})$, $\tilde{J}_{\text{оп}} = (J_{\text{оп}} \setminus j_1) \cup j_0$, соответствующий ему псевдоплан либо не является планом задачи (1), либо является планом задачи (1), но не удовлетворяет условию субоптималь-

ности. По описанной выше схеме найдем для $\{\tilde{\delta}, \tilde{A}_{\text{оп}}\}$ новые направления изменения $\Delta^2\delta$, числа μ_2, j_2 . Новый коплан задачи (1) строим в виде $\bar{\delta} = \delta + z_1^0\Delta^1\delta + z_2^0\Delta^2\delta$, где $z^0 = \{z_1^0, z_2^0\}$ — решение задачи

$$b'\bar{y} - d_*'\bar{v} + d_*'\bar{w} \rightarrow \min_{z_1, z_2}, \quad \delta + z_1\Delta^1\delta + z_2\Delta^2\delta - \bar{v} + \bar{w} = 0, \quad (16)$$

$$A'\bar{y} - \delta - z_1\Delta^1\delta - z_2\Delta^2\delta = c, \quad \bar{v} \geq 0, \quad \bar{w} \geq 0.$$

Займемся преобразованием задачи (16). Сначала заметим, что значения \bar{v}, \bar{w} в задаче (16) однозначно определены по значению $\delta(z) = \delta + z_1\Delta^1\delta + z_2\Delta^2\delta$ в силу условия согласования двойственного плана. Обозначим: $\bar{y} = y + \Delta y$, тогда с учетом $A'y - \delta = c$ имеем $A'\Delta y = z_1\Delta^1\delta + z_2\Delta^2\delta$. Отсюда получим $\Delta y' = z_1\mu_1 A_{\text{оп}}^{-1}(j_1, I) + z_2\mu_2 A_{\text{оп}}^{-1} \times \times (j_2, I)$.

Задача (16) эквивалентна следующей задаче:

$$\gamma_1 z_1 + \gamma_2 z_2 - d_*'\bar{v} - d_*'\bar{w} \rightarrow \min_{z_1, z_2},$$

$$z_1\Delta^1\delta + z_2\Delta^2\delta - \bar{v} + \bar{w} = -\delta, \quad \bar{v} \geq 0, \quad \bar{w} \geq 0, \quad (17)$$

где $\gamma_1 = \mu_1 A_{\text{оп}}^{-1}(j_1, I)b$, $\gamma_2 = \mu_2 \tilde{A}_{\text{оп}}^{-1}(j_2, I)$. Задача (17), в свою очередь, является двойственной к задаче

$$-\delta'x \rightarrow \max_x, \quad \Delta^1\delta'x = \gamma_1, \quad \Delta^2\delta'x = \gamma_2, \quad d_* \leq x \leq d^*. \quad (18)$$

Полученные результаты позволяют вычислить значения z_1^0, z_2^0 . Для этого необходимо решить задачу (18) каким-нибудь известным методом (например, см. § 1), затем, пользуясь теорией двойственности линейного программирования, из оптимальных планов и опоры задачи (18) извлечь значения z_1^0, z_2^0 как решение двойственной к (18) задачи. Однако нас интересует только коплан $\bar{\delta} = \delta + z_1^0\Delta^1\delta + z_2^0\Delta^2\delta$, а не значения z_1^0, z_2^0 . Поэтому используем полученные выше результаты для вычисления $\bar{\delta}$ и опишем алгоритм вычисления $\bar{\delta}$, который фактически является реализацией двойственного метода (см. § 1), примененного к задаче (18).

Положим $K_{\text{н}} = \tilde{J}_{\text{н}}$, $i_0 = j_0$, $i_1 = j_2$; $\delta^1 = \tilde{\delta}$; $z_0(K_{\text{н}}) = \mu_1\Delta^1\delta(K_{\text{н}})/x_{j_1j_0} = \tilde{A}_{\text{оп}}^{-1}(j_0, I)A(I, K_{\text{н}})$; $z_1(K_{\text{н}}) = \mu_2\Delta^2\delta(K_{\text{н}}) = \tilde{A}_{\text{оп}}^{-1}(j_2, I)A(I, K_{\text{н}})$; $g_0 = \mu_1\gamma_1/x_{j_1j_0} = \tilde{A}_{\text{оп}}^{-1}(j_0, I)b$, $g_1 =$

$= \mu_2 \gamma_2 = \tilde{A}_{\text{оп}}^{-1}(j_2, I) b$. Процедура построения $\bar{\delta}$ состоит из следующих шагов.

Шаг 1. Положив $\bar{d}_{*i_1} = \bar{d}_{i_1}^* = d_{*i_1}$, если $\delta_{i_1}^1 > 0$; $\bar{d}_{*i_1} = \bar{d}_{i_1}^* = d_{i_1}^*$, если $\delta_{i_1}^1 < 0$; $\bar{d}_{*i_1} = d_{*i_1}$, $\bar{d}_{i_1}^* = d_{i_1}^*$, если $\delta_{i_1}^1 = 0$, построим вектор $l(K_H)$:

$$l_j = \begin{cases} d_j^* & \text{при } \delta_j^1 < 0, \\ d_{*j} & \text{при } \delta_j^1 \geq 0, \quad j \in K_H. \end{cases}$$

Шаг 2. Вычислим число $\alpha = g_1 - z_1'(K_H)l(K_H)$. Если $\alpha \in [\bar{d}_{*i_1}, \bar{d}_{i_1}^*]$, то процесс построения закончим по формуле $\bar{\delta} = \delta_1$. Приписав коплану $\bar{\delta}$ опору $\bar{A}_{\text{оп}} = A(I, \bar{J}_{\text{оп}})$, $\bar{J}_{\text{оп}} = J \setminus K_H$, начнем новую итерацию предлагаемого метода. Если $\alpha \in [\bar{d}_{*i_1}, \bar{d}_{i_1}^*]$, то, положив $\mu = 1$, $\eta = \alpha - \bar{d}_{*i_1}$ при $\alpha < \bar{d}_{*i_1}$; $\mu = -1$, $\eta = -\alpha + \bar{d}_{i_1}^*$ при $\alpha > \bar{d}_{i_1}^*$, перейдем к шагу 3.

Шаг 3. Подсчитаем числа

$$\sigma(j) = \begin{cases} -\delta_j^1 / \mu z_{1j} & \text{при } \text{sign } \delta_j^1 \neq \text{sign } \mu z_{1j}, \\ \infty & \text{в остальных случаях, } j \in K_H, \end{cases}$$

$$\sigma(i_1) = \begin{cases} |\delta_{i_1}^1| & \text{при } \mu \delta_{i_1}^1 < 0, \\ \infty & \text{в остальных случаях.} \end{cases}$$

Если $\sigma(j) = \infty \quad \forall j \in K_H \cup i_1$, то исходная задача (1) не имеет решения, поскольку ее ограничения несовместны. В противном случае упорядочим числа $\sigma(j) < \infty$ по возрастанию: $\sigma(j_1) \leq \sigma(j_2) \leq \dots \leq \sigma(j_k)$ и вычислим η_t , $t = \overline{1, k}$: $\eta_{t-1} + |z_{1j_t}|(l_{j_t} - d_{*j_t})$, если $\mu z_{1j_t} > 0$, $\eta_{t-1} + |z_{1j_t}|(d_{j_t}^* - l_{j_t})$, если $\mu z_{1j_t} < 0$, считая $\eta_0 = \eta$, $z_{1i_1} = 1$. Перейдем к шагу 4.

Шаг 4. Если $\eta_k < 0$, то исходная задача (1) не имеет решения из-за несовместности ее ограничений. В случае $\eta_k \geq 0$ найдем такой индекс s , что $\eta_s \geq 0$, $\eta_{s-1} < 0$. Положив $\bar{K}_H = (K_H \cup i_1) \setminus j_s$, $\bar{i}_1 = i_0$, $\bar{i}_0 = j_s$, $\bar{d}_{*i_1} = d_{*i_1}$, $\bar{d}_{i_1}^* = d_{i_1}^*$, построим векторы $\bar{z}_0(\bar{K}_H)$, $\bar{z}_1(\bar{K}_H)$, $\bar{\delta}^1(J)$, $\bar{l}(\bar{K}_H)$ по формулам

$$\bar{z}_{0i_1} = 1/z_{1j_s}, \quad \bar{z}_0(\bar{K}_H \setminus i_1) = z_1(\bar{K}_H \setminus i_1)/z_{1j_s};$$

$$\bar{z}_{1i_1} = -z_{0j_s}/z_{1j_s}, \quad \bar{z}_1(\bar{K}_H \setminus i_1) = z_0(\bar{K}_H \setminus i_1) - z_{0j_s}\bar{z}_0(\bar{K}_H \setminus i_1);$$

$$\bar{\delta}^1(K_H \cup i_1) = \delta^1(K_H \cup i_1) + \mu\sigma(j_s)z_1(K_H \cup i_1),$$

$$\bar{\delta}^1(J \setminus (K_H \cup i_1)) = \delta^1(J \setminus (K_H \cup i_1));$$

$$\bar{l}(K_H \setminus \{j_1, j_2, \dots, j_{s-1}\}) = l(K_H \setminus \{j_1, j_2, \dots, j_{s-1}\}),$$

$$\bar{l}_j = \begin{cases} d_j^*, & \text{если } \mu z_{1j} < 0, \\ d_{*j}, & \text{если } \mu z_{1j} > 0, \quad j \in \{j_1, j_2, \dots, j_{s-1}\}; \end{cases}$$

$$\bar{l}_{i_1} = \begin{cases} d_{i_1}^*, & \text{если } \mu < 0, \\ d_{*i_1}, & \text{если } \mu > 0, \end{cases}$$

и подсчитаем числа $\bar{g}_0 = g_1/z_{1j_s}$, $\bar{g}_1 = g_0 - z_{0j_s}\bar{g}_0$. Перейдем к шагу 2, заменив при этом $z_0, z_1, i_0, i_1, \delta^1, K_H, l, g_0, g_1$ на $\bar{z}_0, \bar{z}_1, \bar{i}_0, \bar{i}_1, \bar{\delta}^1, \bar{K}_H, \bar{l}, \bar{g}_0, \bar{g}_1$.

Таким образом описание процедуры построения коплан $\bar{\delta}$ закончено.

З а м е ч а н и я. 1. Нетрудно видеть, что коплан $\bar{\delta}^1$, полученный в результате первой итерации процедуры построения $\bar{\delta}$, является не чем иным, как копланом, полученным за две итерации двойственного метода.

2. Итерацию $\{\delta, A_{\text{оп}}\} \rightarrow \{\bar{\delta}, \bar{A}_{\text{оп}}\}$ назовем регулярной, если значение $\sigma(j_s)$ в процедуре построения $\bar{\delta}$ больше нуля, по крайней мере, один раз. Предлагаемый метод за конечное число итераций дает решение (ε -оптимальное) задачи (1), если конечно число нерегулярных итераций. Однако нетрудно построить несложную модификацию предлагаемого метода, которая решает любую задачу (1) за конечное число итераций (см. замечания к блочному методу § 8).

3. Процедуру построения δ по желанию можно прервать до выполнения условия $\alpha \in [d_{*i_1}, \bar{d}_{i_1}^*]$ на шаге 2 (например, когда число $|\eta|$ достаточно мало) и начинать новую итерацию с опорного коплана $\{\bar{\delta} = \delta^1, \bar{A}_{\text{оп}} = A(I, \bar{J}_{\text{оп}}), \bar{J}_{\text{оп}} = J \setminus K_H\}$.

§ 10. Безопорные методы

Следуя [5], безопорными будем называть точные методы, в которых для точного соблюдения на итерациях всех ограничений исходной задачи опоры не привлекаются. В связи с этим существенно изменяется принцип построения итерации. Для определенности поясним его только на двойственном методе. В опорных методах наличие опоры позволяло легко обеспечить выполнение основного ограничения, а прямое ограничение служило или для останова процесса (в случае оптимальности), или для построения подходящего направления. В без-

опорных методах в основу итерации берется прямое ограничение. Основное ограничение используется для останова процесса решения (на оптимальном плане) или для построения подходящего направления.

Двойственный безопорный метод, изложенный в данном параграфе, отличается от своего аналога из [5] следующими принципиальными моментами. Вспомогательная задача по проверке совместности систем сводится не к задаче первой фазы симплекс-метода с искусственными переменными, а к задаче кусочно-линейного программирования, для которой в [14] разработан специальный конечный алгоритм. Далее, вспомогательная задача, по решению которой делается заключение об оптимальности текущего двойственного плана или строится подходящее направление, решается в общем случае не до конца, а лишь до некоторого момента, позволяющего сделать определенные выводы. Наконец, в новом алгоритме вдоль каждого подходящего направления делается не «короткий» шаг, а «длинный». Аналогичный прием существенно повышает эффективность методов.

Сочетание перечисленных приемов с приемами работ [22, 23] (в частности, с ε -задачами и с использованием алгоритмов квадратичного программирования) может привести к созданию универсальных эффективных алгоритмов линейных задач планирования и управления.

1. Постановка задачи. Критерий оптимальности. Рассмотрим задачу линейного программирования

$$f(x) = c'x \rightarrow \max, Ax = b, d_* \leq x \leq d^*, \quad (1)$$

где $A = A(I, J)$, $b = b(I)$, $d_* = d_*(J)$, $d^* = d^*(J)$, $x = x(J)$, $I = \{1, 2, \dots, m\}$, $J = \{1, 2, \dots, n\}$. Понятие плана, оптимального и субоптимального планов задачи (1) сохраним такими же, как в § 1.

Пусть известен план $\lambda = (y, v, w)$ двойственной к (1) задачи

$$\varphi(\lambda) = b'y - d_*v + d^*w \rightarrow \min, A'y - v + w = c, \\ v \geq 0, w \geq 0. \quad (2)$$

Копланом задачи (1) назовем вектор $\delta = A'y - c$, построенный по компоненте y двойственного плана λ задачи (2). Будем считать, что выполняются условия согласования двойственного плана:

$$v_j = \delta_j, w_j = 0, \text{ если } \delta_j \geq 0; \\ v_j = 0, w_j = -\delta_j, \text{ если } \delta_j < 0, j \in J. \quad (3)$$

(В противном случае путем изменения компонент v , w можно уменьшить значение двойственной целевой функции.) Исходя из известного двойственного плана $\lambda = (y, v, w)$, построим алгоритм преобразования копланов δ с целью получения оптимального плана x^0 задачи (1).

По начальному коплану δ построим множества $J_+ = \{j \in J : \delta_j > 0\}$, $J_- = \{j \in J : \delta_j < 0\}$, $J_0 = \{j \in J : \delta_j = 0\}$. Для этого разбиения множества J построим квазиплан $\chi = \chi(J)$ с компонентами $\chi_j = d_{*j}$, если $j \in J_+$; $\chi_j = d_j^*$, если $j \in J_-$. Переменным χ_j , $j \in J_0$, припишем произвольные значения $d_{*j} \leq \chi_j \leq d_j^*$, $j \in J_0$. На построенном квазиплане *) χ вычислим вектор невязок $r = A\chi - b$.

Теорема 1. При $r=0$ начальный двойственный план λ оптимален. Соответствующий ему квазиплан χ является оптимальным планом задачи (1).

Доказательство. Поскольку $r=0$, т. е. $A\chi = b$, и, по построению, $d_* \leq \chi \leq d^*$, то вектор χ является планом задачи (1). Используя условия согласования (3) и определение квазиплана, имеем $f(\chi) = c'\chi = (y'A - \delta')\chi = b'y - \delta'(J_+)\chi(J_+) - \delta'(J_-)\chi(J_-) = b'y - d'v + d^*w = \varphi(\lambda)$. Таким образом, значения прямой и двойственной функций совпадают. Согласно теории двойственности λ — оптимальный план задачи (2), χ — оптимальный план задачи (1). Теорема доказана.

2. Формула приращения. Пусть δ — коплан задачи (1), $\bar{\delta} = \delta + \Delta\delta$ — другой коплан, построенный по согласованному двойственному плану ($\bar{y} = y + \Delta y$, $\bar{v} = v + \Delta v$, $\bar{w} = w + \Delta w$), $r = A\chi - b \neq 0$. Вычислим приращение двойственной целевой функции $\varphi(\lambda)$ при переходе от коплана δ к коплану $\bar{\delta}$: $\varphi(\bar{\lambda}) - \varphi(\lambda) = b'\Delta y - d'\Delta v + d^*\Delta w = (\chi'A' - r')\Delta y - d'\Delta v + d^*\Delta w = -r'\Delta y + \kappa'\Delta\delta - d'\Delta v + d^*\Delta w$.

Используя условия согласования (3), получаем

$$\varphi(\bar{\lambda}) - \varphi(\lambda) = -r'\Delta y + \sum_{j \in J_0, \bar{\delta}_j > 0} (\chi_j - d_{*j}) \bar{\delta}_j + \sum_{j \in J_0, \bar{\delta}_j < 0} (\chi_j - d_j^*) \bar{\delta}_j + \sum_{j \in J_+, \bar{\delta}_j < 0} (\chi_j - d_j^*) \bar{\delta}_j + \sum_{j \in J_-, \bar{\delta}_j > 0} (\chi_j - d_{*j}) \bar{\delta}_j. \quad (4)$$

*) Вектор χ называется квазипланом задачи (1), если на нем выполняются прямые ограничения $d_{*j} \leq \chi_j \leq d_j^*$, $j \in J$.

3. Итерация. Пусть на построенном по коплану δ квазиплане χ вектор невязок не равен нулю: $r \neq 0$. Введем множество $I_0 = \{i \in I: r_i = 0\}$ (множество I_0 может быть и пустым множеством). Сформируем задачу

$$f^*(\chi(J_0)) = \max_{i \in K} |A(i, J_0)\chi(J_0) - \bar{b}_i| \rightarrow \min,$$

$$A(I_0, J_0)\chi(J_0) = \bar{b}(I_0), d_*(J_0) \leq \chi(J_0) \leq d^*(J_0), \quad (5)$$

где $K = I \setminus I_0$, $\bar{b} = b - A(I, J_+)\chi(J_+) - A(I, J_-)\chi(J_-)$.

Задача (5) является задачей условной минимизации максимального уклонения*), для решения которой используем метод, изложенный в [14]. Выбирая начальный план $\chi(J_0)$ и приписывая ему произвольную опору ограничений $A_{\text{оп}} = A(I_0, J_{\text{оп}})$, строим опору целевой функции $F_{\text{оз}} = \{A_{\text{оп}}, D_{\text{оп}}\}$, где $J_{\text{оп}} = \emptyset$, $K_{\text{оп}} = i_0$, $f^*(\chi(J_0)) = |A(i_0, J_0)\chi(J_0) - \bar{b}_{i_0}|$, $D_{\text{оп}} = \{1\}$, если $A(i, J_0)\chi(J_0) - \bar{b}_{i_0} > 0$; $D_{\text{оп}} = \{-1\}$, если $A(i, J_0)\chi(J_0) - \bar{b}_{i_0} < 0$. Исходя из начального согласованного опорного плана $\{\chi(J_0), F_{\text{оз}}\}$, решаем задачу (5) методом из [14].

Пусть получен некоторый план $\bar{\chi}(J_0)$ задачи (5), для которого выполняется условие

$$f^*(\bar{\chi}(J_0)) - \beta^* > 0. \quad (6)$$

Здесь β^* — оценка субоптимальности согласованного опорного плана $\{\chi(J_0), \bar{F}_{\text{оз}}\}$. Для проведения дальнейших выкладок потребуется подробная детализация индексных множеств I, J_0 задачи (5), которую произведем с учетом ситуации, сложившейся на рассматриваемой итерации метода [14]. Пусть $K = K_{\text{оп}} \cup K_{\text{н}}$, $K_{\text{оп}} = K_{\text{оп}}^+ \cup K_{\text{оп}}^-$, $K_{\text{оп}}^+ \cap K_{\text{оп}}^- = \emptyset$, $J_0 = J_{0\text{оп}} \cup J_{0\text{он}} \cup J_{0\text{н}}$, $F_{\text{оз}} = \{A_{\text{оп}}, D_{\text{оп}}\}$, $D_{\text{оп}} = D_{\text{оп}}(K_{\text{оп}}, j_e \cup J_{0\text{он}}) = \begin{cases} e(K_{\text{оп}}^+) : D(K_{\text{оп}}^+, J_{0\text{он}}) \\ -e(K_{\text{оп}}^-) : D(K_{\text{оп}}^-, J_{0\text{он}}) \end{cases}$, $D(K, J_0) = -A(K, J_0) + A(K, J_{0\text{он}})A_{\text{оп}}^{-1}A(I_0, J_0)$.

Из построения метода [14] следует, что на согласованном опорном плане $\{\bar{\chi}(J_0), F_{\text{оз}}\}$ выполняются соотношения

*) В классическом безопорном методе (методе одновременного решения прямой и двойственной задач) на этом этапе для проверки совместности системы $A\chi = b$ с помощью искусственных переменных вводится линейная задача, которая решается симплекс-методом.

$$A(i, J_0) \bar{\chi}(J_0) - \bar{b}_i = f^*(\bar{\chi}(J_0)), i \in K_{\text{оп}}^+;$$

$$A(i, J_0) \bar{\chi}(J_0) - \bar{b}_i = -f^*(\bar{\chi}(J_0)), i \in K_{\text{оп}}^-;$$

$$\sum_{i \in K_{\text{оп}}^+} u_i - \sum_{i \in K_{\text{оп}}^-} u_i = 1, u'(K_{\text{оп}}) = [1 : 0(J_{\text{оп}})]' D_{\text{оп}}^{-1}, \quad (7)$$

$$\Delta'(J_0) = u'(K_{\text{оп}}) D(K_{\text{оп}}, J_0),$$

$$y'(I_0) = u'(K_{\text{оп}}) A(K_{\text{оп}}, J_{\text{воп}}) A^{-1}(J_{\text{воп}}, I_0),$$

$$\Delta(J_{\text{воп}} \cup J_{\text{ооп}}) = 0, \beta^* = \sum_{\Delta_j > 0, j \in J_{\text{он}}} \Delta_j (\bar{\chi}_j - d_{*j}) + \sum_{\Delta_j < 0, j \in J_{\text{он}}} \Delta_j (\bar{\chi}_j - d_j^*).$$

Покажем, что вектор $\Delta y = y^*$, $y^* = y^*(I)$, $y^*(K_{\text{оп}}) = u(K_{\text{оп}})$, $y^*(K_{\text{н}}) = 0$, $y^*(I_0) = y(I_0)$, задает направление убывания двойственной целевой функции. Из формулы приращения (4) следует, что начальная скорость изменения двойственной целевой функции вдоль построенного направления

$$\gamma_0 = -r'y^* + \sum_{j \in J_0, \Delta_j^* > 0} (\bar{\chi}_j - d_{*j}) \Delta_j^* + \sum_{j \in J_0, \Delta_j^* < 0} (\bar{\chi}_j - d_j^*) \Delta_j^*,$$

где $\Delta^* = A'y^*$. Из соотношений (7) вытекает, что $\Delta^*(J_0) = \Delta(J_0)$. С учетом (6), (7) получаем

$$\gamma_0 = - \sum_{i \in K_{\text{оп}}^+} u_i f^*(\bar{\chi}(J_0)) + \sum_{i \in K_{\text{оп}}^-} u_i f^*(\bar{\chi}(J_0)) +$$

$$+ \sum_{i \in K_{\text{н}}} 0 \cdot r_i + \sum_{i \in I_0} 0 \cdot y_i + \sum_{j \in J_0, \Delta_j > 0} (\bar{\chi}_j - d_{*j}) \Delta_j +$$

$$+ \sum_{j \in J_0, \Delta_j < 0} (\bar{\chi}_j - d_j^*) \Delta_j = -f^*(\bar{\chi}(J_0)) + \beta^* < 0.$$

В отличие от классической схемы безопорного метода задачу (5) необязательно решать до конца. Улучшение двойственной целевой функции $\varphi(\lambda)$ можно начать на таком шаге адаптивного метода решения задачи (5), на котором выполняется неравенство (6).

Скорость убывания двойственной целевой функции при движении вдоль Δ^* не изменяется до тех пор, пока компоненты коплана $\bar{\delta}_j^{(0)} = \delta_j + \sigma \Delta_j^*$, $j \in J_+ \cup J_-$, не меняют знаки. Движение вдоль построенного подходящего направления ведет к убыванию двойственной целевой функции (до тех пор, пока скорость отрицательна). Поэтому максимально допустимый шаг σ_0 при скорости γ_0 находим по формуле

$$\sigma_0 = \min \sigma_j, \quad \sigma_j = \begin{cases} -\delta_j/\Delta_j^*, & \text{если } \delta_j/\Delta_j^* < 0, j \in J_+ \cup J_-, \\ \infty & \text{в остальных случаях.} \end{cases} \quad (8)$$

Как только хотя бы одна из компонент коплана $\delta(J_+ \cup J_-)$ изменит знак, скорость убывания двойственной целевой функции уменьшится и станет равной:

$$\gamma_1 = \gamma_0 + \sum_{\delta_j^{(0)}=0, \Delta_j^* < 0} (\bar{\chi}_j - d_j^*) \Delta_j^* + \sum_{\delta_j^{(0)}=0, \Delta_j^* > 0} (\bar{\chi}_j - d_{*j}) \Delta_j^*.$$

При $\gamma_1 \geq 0$ движение вдоль Δ^* прекращаем. Если $\sigma_0 = \infty$, то согласно теории двойственности прямая задача (1) не имеет планов.

Пусть $\gamma_1 < 0$. Тогда вычисляем максимально допустимый шаг σ_1 при скорости γ_1 , полагая в формулах (8) $\delta_j = \delta_j^{(0)}$, и вновь пересчитываем скорость γ . Пусть движение вдоль Δ^* возобновлялось несколько раз (при этом неограниченность двойственной целевой функции не обнаружилась) и на очередном этапе происходит со скоростью $\gamma_\mu < 0$. Полагая в формулах (8) $\delta_j = \delta_j^{(\mu)}$, вычисляем максимально возможный шаг σ_μ и пересчитываем скорость

$$\gamma_{\mu+1} = \gamma_\mu + \sum_{\delta_j^{(\mu)}=0, \Delta_j^* < 0} (\bar{\chi}_j - d_j^*) \Delta_j^* + \sum_{\delta_j^{(\mu)}=0, \Delta_j^* > 0} (\bar{\chi}_j - d_{*j}) \Delta_j^*,$$

если шаг σ_μ конечный. В противном случае двойственная целевая функция не ограничена на множестве допустимых планов.

Движение вдоль Δ^* прекращается, как только получена скорость $\gamma_v \geq 0$, либо обнаружится неограниченность двойственной целевой функции. Поскольку на каждом этапе число элементов множества $J_+ \cup J_-$, по которым может быть вычислен конечный шаг σ , уменьшается, то движение вдоль Δ^* прекратится через конечное число этапов.

Новый коплан пересчитывается по формуле $\bar{\delta} = \delta + \sum_{q=0}^{\mu-1} \sigma_q \Delta^*$. Ему соответствует новое разбиение множества J . Новая итерация начинается с коплана $\bar{\delta}$. За одну итерацию значение двойственной целевой функции уменьшается на величину $\sum_{q=1}^{\mu-1} |\gamma_q| \sigma_q$.

Поскольку $\Delta^*(J_{0\text{оп}} \cup J_{0\text{оц}}) = 0$, то $J_{0\text{оп}} \cup J_{0\text{оц}} \in J_0$, где

$\bar{J}_0 = \{j \in J : \bar{\delta}_j = 0\}$. Кроме того, в множество входит, по крайней мере, один новый индекс $j_* \in J_+ \cup J_-$, которого не было в множестве J_0 .

Теорема 2. Пусть для каждого коплана δ при решении соответствующей задачи (5) субоптимальный план $\chi^*(J_0)$, удовлетворяющий неравенству (6), строится за конечное число итераций. Тогда вышеописанный алгоритм строит оптимальный план задачи (1) либо обнаруживает отсутствие планов задачи (1) за конечное число шагов.

Доказательство. В условиях теоремы на каждой итерации происходит улучшение двойственной целевой функции

$$\varphi(\bar{\lambda}) < \varphi(\lambda). \quad (9)$$

Каждая задача (5) определяется индексным множеством J_0 , которое, в силу (9) и условий теоремы, не может встретиться дважды. Поскольку число различных подмножеств J_0 множества J конечно, то через конечное число итераций будет либо получен оптимальный коплан δ (соответствующий ему квазиплан χ будет оптимальным планом задачи (1)), либо обнаружится отсутствие планов задачи (1). Теорема доказана.

§ 11. Приближенные методы

В арсенале средств линейного программирования кроме точных должны обязательно присутствовать и приближенные методы. В них на итерациях ограничения могут не выполняться, т. е. итерации ведутся не на планах, а только на их оценках (приближениях) [10, 15]. Целесообразность использования приближенных методов основана на том, что здесь нет необходимости хранить в памяти ЭВМ специальные элементы опорной матрицы, которые должны отвечать за точное соблюдение всех ограничений задачи. Кроме экономии оперативной памяти в приближенных методах появляется возможность упрощения операций. Если еще учесть, что среди приближенных методов существуют и конечные (основное достоинство многих точных методов), то нетрудно указать ситуации, где они предпочтительнее точных. В частности, приближенные методы целесообразно использовать при построении программных оптимальных управлений в тех

случаях, когда имеется достаточно времени на решение задачи и не используется режим реального времени.

1. Постановка задачи. Рассмотрим задачу

$$\begin{aligned} c'x &\rightarrow \max, \quad Ax \leq b, \\ x \in Q &= \{x: x_j \geq 0, j \in J = \{1, 2, \dots, n\}\}, \end{aligned} \quad (1)$$

и задачу, двойственную к (1):

$$\begin{aligned} b'y &\rightarrow \min, \quad A'y \geq c, \\ y \in R &= \{y: y_i \geq 0, i \in I = \{1, 2, \dots, m\}\}. \end{aligned} \quad (2)$$

Вектор $x \in Q$ ($y \in R$) называется прямым (двойственным) планом, если $Ax \leq b$ ($A'y \geq c$); $r(x) = Ax - b$ ($q(y) = c - A'y$) — вектором прямых (двойственных) невязок.

Нетрудно проверить, что вектор $x \in Q$ ($y \in R$) является прямым (двойственным) планом тогда и только тогда, когда $P_R r(x) = 0$ ($P_Q q(y) = 0$). Здесь $P_T z$ — проекция вектора z на множество T .

Как известно, методы решения задачи (1) делятся на точные и приближенные. В точных методах итерации ведутся на 1) прямых планах (прямые методы); 2) двойственных (двойственные методы); 3) парах из прямых и двойственных планов (комбинированные методы). Приближенными [8] (прямыми, двойственными, комбинированными) называются методы, в которых объекты итераций $x \in Q$, $y \in R$ — не планы, а только оценки (приближения) планов задач (1), (2).

Таким образом, в точных методах вектор $P_R r(x)$, или вектор $P_Q q(y)$, или оба равны нулю и поэтому цель каждой итерации — оптимизировать целевую функцию соответствующей задачи. В приближенных методах наряду с оптимизацией целевой функции нужно уничтожить ненулевые компоненты вектора $P_R r(x)$ ($P_Q q(y)$). Эта задача близка к задаче векторной оптимизации. Назовем ее сопровождающей задачу (1). Существует ряд принципов выбора для конкретного решения задач векторной оптимизации. Использование того или иного принципа выбора зависит от дополнительных требований на поведение в ходе итераций функций $c'x$, $r(x)$, $b'y$, $q(y)$. В рассматриваемой постановке задачи (1) эта информация отсутствует, что в конечном счете затрудняет объективное сравнение эффективности различных приближенных методов.

Специфика сопровождающей задачи состоит в том, что это не векторная задача в традиционном понимании.

В ней целевые функции, связанные с невязками, принципиально отличаются от исходной целевой функции. Значение невязок минимизируется, если знак невязки недопустим, и безразлично, если знак невязки допустим. В связи с этим в известные принципы решения задач векторной оптимизации необходимо внести соответствующие коррективы.

2. Идея методов. Опишем приближенные методы решения задачи (1), в основу которых положен известный в теории векторной оптимизации принцип выбора с оптимизацией взвешенных целевых функций сопровождающей задачи. Естественность такого подхода объясняется тем, что, во-первых, используемый принцип выбора является полным (с помощью него можно найти любую эффективную точку), а во-вторых, необходимые для него весовые коэффициенты явно содержатся в теореме Куна — Таккера. Согласно этой теореме существуют такие веса $y_i^0 \geq 0, i \in I$ (компоненты оптимального двойственного плана y^0), что максимум взвешенных целевых функций $c'x - y^0 r(x)$ достигается на оптимальном плане x^0 задачи (1) и для векторов y^0 и $r(x^0)$ выполняются условия дополняющей нежесткости, т. е.

$$x^0 \in \operatorname{Arg} \max_{x \in Q} (c'x - y^0 r(x)), P_{R(y^0)} r(x^0) = 0. \quad (3)$$

Здесь $R(y) = \{\Delta y : \Delta y_i \geq 0, \text{ если } y_i = 0, i \in I\}$.

Справедливо и обратное утверждение: если для $x^0 \in Q$ при некотором $y^0 \in R$ выполняются соотношения (3), то x^0 — оптимальный план задачи (1).

Аналогично для оптимальности в задаче (2) вектора $y^0 \in R$ необходимо и достаточно существование таких весов $x_j^0 \geq 0, j \in J$ (компонент оптимального прямого плана x^0), что

$$y^0 \in \operatorname{Arg} \min_{y \in R} (b'y + x^0 q(y)), P_{Q(x^0)} q(y^0) = 0. \quad (4)$$

Здесь $Q(x) = \{\Delta x : \Delta x_j \geq 0, \text{ если } x_j = 0, j \in J\}$. В дальнейшем будем пользоваться следующими обозначениями:

$$\begin{aligned} I_+(x) &= \{i \in I : r_i(x) > 0\}, I_-(x) = \{i \in I : r_i(x) < 0\}, \\ I_0(x) &= \{i \in I : r_i(x) = 0\}; \\ J_0(x) &= \{j \in J, x_j = 0\}, J_+(x) = \{j \in J : x_j > 0\}; \\ I_+(y) &= \{i \in I : y_i > 0\}, I_0(y) = \{i \in I : y_i = 0\}; \\ J_+(y) &= \{j \in J : q_j(y) > 0\}, J_-(y) = \{j \in J : q_j(y) < 0\}, \end{aligned} \quad (5)$$

$J_0(y) = \{j \in J : q_j(y) = 0\}$; $[a]_j$ — j -я компонента вектора a .

3. Прямой метод. Пусть $x_0 \in Q$ — начальная оценка оптимального плана x^0 . Из сказанного выше следует, что оценка x — оптимальный план задачи (1) тогда и только тогда, когда существует такой вектор $y \in R$, что выполняются соотношения:

$$a) \ x \in \operatorname{Arg} \max_{x \in Q} (c'x - y'r(x)); \quad б) \ P_{R(y)}r(x) = 0. \quad (6)$$

Для оценки x_0 проверим условия а), б). Покажем, что вектор x_0 удовлетворяет условию а) только в том случае, если существует такой вектор $y_0 \in R$, что $P_{Q(x_0)}q(y_0) = 0$. Действительно, пусть x_0 удовлетворяет условию а), т. е. существует такой вектор y_0 , что

$$x_0 \in \operatorname{Arg} \max_{x \in Q} (c'x - y_0'r(x)) = \operatorname{Arg} \max_{x \in Q} q'(y_0)x. \quad (7)$$

Из определения множества $Q(x_0)$ следует, что для любого вектора y справедливы соотношения $[P_{Q(x_0)}q(y)]_j \geq 0$, если $x_{0j} = 0$, $j \in J$, $q'(y)P_{Q(x_0)}q(y) = \|P_{Q(x_0)}q(y)\|^2$. Следовательно, если допустить, что $P_{Q(x_0)}q(y_0) \neq 0$, то направление $l = P_{Q(x_0)}q(y_0)$ оказалось бы подходящим для плана x_0 в задаче $\max_{x \in Q} q'(y_0)x$, что противоречит (7). С другой

стороны, если существует такой вектор y_0 , что $P_{Q(x_0)}q(y_0) = 0$, то очевидно, что $x_0 \in \operatorname{Arg} \max_{x \in Q} q'(y_0)x$.

Утверждение доказано. Таким образом, для проверки условия а) следует найти вектор $y_0 \in R$, минимизирующий невязку

$$\|P_{Q(x_0)}q(y)\|. \quad (8)$$

Оптимальность плана x_0 связана еще и с условием б). Поэтому невязку (8) будем минимизировать не по всем $y \in R$, а только по множеству $Y(x_0)$ таких векторов \bar{y} , что

$$\|P_{R(\bar{y})}r(x_0)\| = \min_{y \in R} \|P_{R(y)}r(x_0)\|.$$

В результате проверка оценки x_0 на оптимальность свелась к построению вектора $y_0 = y(x_0)$, где

$$y(x) \in \operatorname{Arg} \min_{y \in Y(x)} \|P_{Q(x)}q(y)\|^2; \quad (9)$$

$$Y(x) = \{y : y_i = 0, \text{ если } i \in I_-(x); y_i \geq 0, \text{ если } i \in I \setminus I_-(x)\}. \quad (10)$$

Лемма 1. Для вектора $\Delta x_0 = P_{Q(x_0)} q(y_0)$ верны соотношения

$$c' \Delta x_0 = \|\Delta x_0\|^2, \quad A(i, J) \Delta x_0 \leq 0, \quad i \in I \setminus I_-(x_0), \\ \Delta x_{0j} \geq 0, \quad j \in J_0(x_0). \quad (11)$$

Возможны случаи: 1) $\Delta x_0 \neq 0$, 2) $\Delta x_0 = 0$. В случае 1) условие а) не выполняется и, следовательно, оценка x_0 не оптимальна. Из (11) видно, что вектор Δx_0 задает в точке x_0 направление, при движении вдоль которого целевая функция задачи (1) строго возрастает, а компоненты вектора невязок $P_R r(x)$ не возрастают. Поэтому в соответствии со смыслом задачи (1) новую оценку x_1 строим по формуле $x_1 = x(\Theta_0)$, где $x(\Theta) = x_0 + \Theta \Delta x_0$. Шаг Θ_0 положим равным максимальному среди чисел

$$\Theta \in \text{Arg} \min_{\Theta > 0, x(\Theta) \geq 0} \|P_R r(x(\Theta))\|^2. \quad (12)$$

Лемма 2. Шаг Θ_0 строго положителен.

Если $\Theta_0 = \infty$, то задача (1) не имеет решений (либо пусто множество планов, либо функция $c'x$ не ограничена сверху на множестве планов).

Лемма 3. $y_0 \in Y(x_1)$.

Лемма 4. $\|P_{Q(x_1)} q(y_1)\| < \|P_{Q(x_0)} q(y_0)\|$, $\|P_{R(y_1)} r(x_1)\| \leq \|P_{R(y_0)} r(x_0)\|$, где $y_1 = y(x_1) \in R$ — вектор, соответствующий в силу (8) вектору x_1 .

Рассмотрим случай 2): $P_{Q(x_0)} q(y_0) = 0$, т. е. предположим, что для вектора x_0 выполняется условие а). С целью проверки условия б) построим множество $X(y_0)$, соответствующее весам y_{0i} , $i \in I$, по правилу:

$$X(y) = \{x : x_j = 0, j \in J_-(y); x_j \geq 0, j \in J \setminus J_-(y)\}. \quad (13)$$

Очевидно, что $x_0 \in X(y_0)$. Будем считать, что $x_0 = x(y_0)$, где

$$x(y) \in \text{Arg} \min_{x \in X(y)} \|P_{R(y)} r(x)\|^2. \quad (14)$$

В противном случае оценку x_0 заменим на $x(y_0)$, для которой всегда $P_{Q(x(y_0))} q(y_0) = 0$, $\|P_{R(y_0)} r(x(y_0))\| < \|P_{R(y_0)} r(x_0)\|$.

Если $P_{R(y_0)} r(x_0) = 0$, то x_0, y_0 — решения задач (1), (2). При $P_{R(y_0)} r(x_0) \neq 0$ вектор y_0 заменим на $y_1 = y(\sigma_0)$, где $y(\sigma) = y_0 + \sigma \Delta y_0$, $\Delta y_0 = P_{R(y_0)} r(x_0)$, σ_0 — максимальное среди чисел $\{\sigma : P_{Q(x_0)} q(y(\sigma)) = 0\}$. Если $\sigma_0 = \infty$, то решение задачи (1) прекращаем, ибо ее ограничения несовместны.

В случае $\sigma_0 < \infty$ положим $x_1 = x(y_1)$ и продолжим процесс решения.

Из леммы 5 следует (см. ниже), что $P_{Q(x_1)}q(y_1) = 0$, $y_1 \in Y(x_1)$. Значит, на следующей итерации опять реализуется случай 2). Поэтому сразу переходим к операциям этого случая.

Лемма 5.

$$\|P_{Q(x_1)}q(y_1)\| = \|P_{Q(x_0)}q(y_0)\| = 0;$$

$$\|P_{R(y_1)}r(x_1)\| < \|P_{R(y_0)}r(x_0)\|.$$

Из леммы 4, 5 следует, что на итерации $x_0 \rightarrow x_1$ векторы $\xi(x) = \{\|P_{Q(x)}q(y(x))\|, \|P_{R(y(x))}r(x)\|\}$ (при $\Delta x_0 \neq 0$), $\xi(y) = \{\|P_{Q(x(y))}q(y)\|, \|P_{R(y)}r(x(y))\|\}$ (при $\Delta x_0 = 0$) лексикографически строго убывают.

В терминах задачи (1) оценки x_0, x_1 обладают следующими свойствами:

$$\begin{aligned} x_0 \in \text{Arg} \max_{Ax < b_0, x \in Q} c'x, \quad x_1 \in \text{Arg} \max_{Ax < b_1, x \in Q} c'x, \\ \|c - c_1\| \leq \|c - c_0\|, \quad \|b - b_1\| \leq \|b - b_0\|, \end{aligned} \quad (15)$$

причем хотя бы одно из этих неравенств строгое. Здесь $c_0 = c(x_0, y_0)$, $c_1 = c(x_1, y_1)$, $c(x, y) = c - P_{Q(x)}q(y)$; $b_0 = b(x_0, y_0)$, $b_1 = b(x_1, y_1)$, $b(x, y) = b + P_{R(y)}r(x)$.

Теорема 1. Прямой метод решения задачи (1) конечен.

Доказательство. Итерации прямого метода условно можно разбить на два типа: итерации, соответствующие случаю 1), на которых $\|P_{Q(x)}q(y(x))\| \neq 0$, и итерации, соответствующие случаю 2), на которых $\|P_{Q(x(y))}q(y)\| = 0$, $\|P_{R(y)}r(x(y))\| \neq 0$. Покажем, что итераций первого типа может быть только конечное число. Действительно, согласно (9), (10) величина $\|P_{Q(x)}q(y(x))\|$ однозначно определяется множествами $I_-(x)$ и $J_0(x)$, ибо $\|P_{Q(x)}q(y(x))\|^2 = \min \|c - A'y + w\|^2$, $y \in Y(x)$, $w \in W(x)$, $Y(x) = Y(I_-(x)) = \{y: y_i = 0, i \in I_-(x); y_i \geq 0, i \in I \setminus I_-(x)\}$, $W(x) = W(J_0(x)) = \{w: w_j = 0, j \in J_+(x); w_j \geq 0, j \in J_0(x)\}$. Количество всевозможных наборов $\{I_-, J_0\}$ конечно. Из леммы 4 следует, что один и тот же набор не может встретиться дважды. Следовательно, через конечное число итераций либо докажем, что задача (1) не имеет решений, либо придем к случаю 2): $\Delta x = \|P_{Q(x)}q(y(x))\| = 0$, т. е. к итерациям второго типа. После этого итерации первого типа появиться уже не могут.

Рассуждая аналогично, покажем, что итераций второ-

го типа может быть только конечное число. Действительно, величина $\|P_{R(y)}r(x(y))\|$ однозначно определяется множествами $I_0(y)$, $J_-(y)$. Таких множеств конечное число. В силу леммы 5 один набор дважды встретиться не может. Следовательно, через конечное число итераций либо придем к случаю $\|P_{R(y)}r(x(y))\|=0$ (т. е. $x(y)$ — оптимальный план задачи (1)), либо докажем, что исходная задача не имеет решений. Теорема доказана. Основываясь на критерии оптимальности (4) и на рассуждениях, аналогичных приведенным в п. 3, можно построить двойственный метод решения задачи (1). Более подробно с изложением двойственного метода, комбинированного метода и приближенных методов, базирующихся на иных принципах выбора сопровождающей задачи, можно ознакомиться в [10, 15].

§ 12. Особенности реализации алгоритмов линейного программирования на ЭВМ

Программная реализация алгоритма, кроме его вычислительной схемы, обычно требует много дополнительной информации, связанной с точностью представления чисел в памяти ЭВМ, с возможностями системы, в которой реализуется работа программы, и т. д. Даже для самой простой программной реализации, наиболее близкой к математическому описанию алгоритма, необходимо задавать параметры для учета возможных отклонений от точных значений приближенно вычисляемых на ЭВМ величин, что позволяет избегать нарушений вычислительной схемы алгоритма. Например, операция сравнения с нулем в алгоритме может выполняться неправильно, если вместо нуля не ввести параметр, значение которого должно быть достаточно малым, но в то же время служить как бы оценкой сверху (по модулю) для возможных ошибок округления. При задании этих параметров учитываются как сложность производимых операций, так и размеры решаемой задачи. Например, в программах, по которым были проведены эксперименты в § 1, 5, 7—9 и решены прикладные задачи в § 5 гл. 2, оценки Δ_j , $j \in J_n$, полагались равными нулю при $|\Delta_j| < 10^{-7}$.

Накопление ошибок округления в алгоритмах в значительной мере зависит от структуры данных задачи. Одной из мер, позволяющей избежать чрезмерно быстрого накопления ошибок округления и, следовательно, непра-

вильного выполнения алгоритма, является масштабирование матрицы. Наиболее распространенный способ масштабирования состоит в делении каждой строки матрицы на наибольший по абсолютному значению элемент этой строки. В программных комплексах, предназначенных для решения больших прикладных задач, в настоящее время применяются и более совершенные методы масштабирования.

Основной момент, по которому различаются программные реализации, — форма представления числовых данных. Рассмотрим два из часто используемых способов хранения разреженных матриц в памяти ЭВМ. Один — запись ненулевого элемента и номера его строки в две соседние ячейки. Такие пары, соответствующие всем ненулевым элементам столбца, располагаются последовательно в одномерном массиве. В следующую свободную ячейку этого массива записывается первый ненулевой элемент другого столбца, а затем — его индекс и т. д. Номер ячейки (адрес), в которую записывается новый столбец, запоминается в другом массиве (для целых чисел). При таком способе хранения информация о матрице занимает $2t + n$ ячеек, где t — число ненулевых элементов матрицы, n — число столбцов.

Второй способ применяется в программах для ЭВМ, ячейки которых состоят из большого числа двоичных разрядов. Например, ячейка БЭСМ-6 состоит из 48 разрядов и поэтому элементы матрицы, как правило, занимают лишь старшие разряды ячейки, а в младших разрядах можно хранить их индексы. В этом случае информация о матрице требует всего $t + n$ ячеек. В процессе счета номера строк выделяются из ячеек с помощью процедуры PLA0R1, написанной на автокоде. (На «распаковку» информации, конечно, тратится дополнительное машинное время.) В больших задачах уменьшение объема информации, хранящейся во внешней памяти, ускоряет обмен с оперативной памятью и тем самым компенсирует время, затраченное на распаковку информации.

Особенность адаптивного алгоритма заключается в том, что на его итерациях используются не только столбцы, но и строки матрицы условий. При умножении вектора на матрицу слева, например, при вычислении $w = Ax$, когда матрица записана только по столбцам, счет удобнее вести по формуле $w(I) = \sum_{j \in J} A(I, j)x(j)$, т. е.

массив w заполняется суммой столбцов матрицы A , умноженной на компоненты x . Но в ряде операций адаптивного алгоритма требуется только одна строка матрицы или только ее опорные строки. Тогда вызов и распаковка всех столбцов для выделения элементов нужных строк значительно усложняет операцию со строкой. Поэтому для ускорения работы адаптивного алгоритма целесообразно иметь матрицу, записанную как по столбцам, так и по строкам. Однако следует отметить, что в задачах с большим числом переменных, но со сравнительно малым числом ненулевых элементов в каждом столбце такой подход может привести к неоправданным затратам памяти.

Эффективность алгоритма существенно зависит от формы хранения обратной матрицы в памяти ЭВМ. Наиболее распространенной формой для представления обратной опорной матрицы в программных реализациях симплекс-метода является мультипликативная форма. В § 3 излагается вариант адаптивного алгоритма с мультипликативным представлением обратной матрицы. В других вариантах адаптивного алгоритма предполагается прямое использование квадратной матрицы $A_{\text{оп}}^{-1}$.

Рассмотрим некоторые способы хранения этой матрицы в памяти ЭВМ. Максимальный размер опоры в задаче с основными ограничениями-неравенствами до ее решения, как правило, неизвестен, поэтому в программе, где матрица $A_{\text{оп}}^{-1}$ описана двумерным массивом, границы массива указываются равными m . В памяти ЭВМ этот массив располагается построчно (по столбцам), причем каждая строка (столбец) занимает m ячеек. Когда значение m^2 превышает допустимый объем оперативной памяти, часть строк (столбцов) необходимо записывать во внешнюю память. Между тем при плотной записи матрица $A_{\text{оп}}^{-1}$ занимает k^2 ячеек и при $k^2 \leq K$, где K — допустимый объем памяти, она полностью может быть размещена в оперативной памяти. Поэтому на практике для хранения $A_{\text{оп}}^{-1}$ выделяется одномерный массив, который заполняется с ростом размеров матрицы. Этот массив называется буфером. Если не все строки во время пересчета $A_{\text{оп}}^{-1}$ вмещаются на буфер, то он переписывается, например, на диск, а оставшиеся строки вновь записываются на буфер. В этом случае даже при больших m итерации с небольшой опорой выполняются без обмена с

внешней памятью. Для сильно разреженных матриц $A_{\text{оп}}^{-1}$ запоминаются только ее ненулевые элементы ($|a_{ij}| > 10^{-8}$), что позволяет сократить необходимый объем памяти, а также избежать операций с нулевыми элементами.

Поскольку в постановке реальных задач обычно присутствуют только некоторые компоненты векторов c , d_* , d^* , b_* , b^* , а значения остальных подразумеваются по смыслу, то и в память ЭВМ целесообразно заносить только заданные значения компонент и их индексы. Значения остальных компонент при работе алгоритма для каждого вектора будут восстанавливаться «по умолчанию». Например, если в списке индексов вектора c не обнаружится индекс j_0 , то компонента c_{j_0} в алгоритме полагается равной 0 и операция с ней исключается. Отсутствующие компоненты векторов d_* , d^* , b_* , b^* полагаются равными 0, ∞ , $-\infty$ и ∞ соответственно.

Для хранения промежуточных результатов, неоднократно применяющихся на итерации алгоритма, описываются рабочие массивы. Однако при использовании большого количества массивов основная часть информации будет храниться во внешней памяти. Поэтому легко вычисляемые вновь векторы не следует описывать массивами, так как время их пересчета может быть меньше времени обмена с внешней памятью. Поскольку требование алгоритма к объему памяти и время решения задачи всегда противоречивы, то построить программу со всеми оптимальными характеристиками для задач произвольной структуры невозможно. На практике учитываются, конечно, оба эти требования к программе, но каждому из них отдается предпочтение в зависимости от размеров и структуры решаемых задач.

Перейдем к рассмотрению стартовых процедур, используемых для расширения исходной задачи в случае, когда неизвестен ее начальный план. Пусть задан произвольный вектор x . Если некоторые его компоненты выходят за свои прямые ограничения, то «подтягиваем» их к ближайшей границе. Полученный вектор подставляем в основные ограничения. При их нарушении строим расширенную задачу и план для нее так, чтобы на итерациях алгоритма был найден сначала план, а затем и оптимальный план исходной задачи.

Для задач с основными ограничениями-равенствами

($Ax=b$) удобен классический способ расширения задачи, предполагающий следующие операции.

1. Если $x_j < d_{*j}$, то положим $x_j = d_{*j}$; если $x_j > d^*$, то $-x_j = d_j^*$, $j \in J$.

2. Вычислим вектор невязок $w = b - Ax$.

3. Положим $x_{n+i} = |w_i|$, $d_{*n+i} = 0$, $d_{n+i}^* = \infty$, $c_{n+i} = -M$, где M — большое число, $i = \overline{1, m}$.

4. Положим $A(i, n+j) = A_{\text{оп}}^{-1}(i, j) = 0$, $i, j = \overline{1, m}$, $i \neq j$;

$$A(i, n+i) = A_{\text{оп}}^{-1}(i, i) = \begin{cases} -1, & \text{если } w_i < 0, \\ 1, & \text{если } w_i \geq 0, \end{cases} i = \overline{1, m}.$$

5. Положим $J_{\text{оп}} = \{n+1, n+2, \dots, n+m\}$.

Процедура на языке АЛГОЛ-60, реализующая эти операции для простейшего случая, когда матрицам A и $A_{\text{оп}}^{-1}$ соответствуют двумерные массивы, имеет вид

```
'PROCEDURE' START1;
'BEGIN' 'INTEGER' I, J; 'REAL' R;
'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
'BEGIN' 'IF' X[J]<D1[J] 'THEN' X[J]:=D1[J];
'IF' X[J]>D2[J] 'THEN' X[J]:=D2[J] 'END';
'FOR' I:=1 'STEP' 1 'UNTIL' M 'DO'
'BEGIN' R:=B[I]; JOP[I]:=N+I; C[N+I]:=-107;
'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO' R:=R-A[1, J] *X[J];
X[N+I]:=ABS(R); D1[N+I]:=0; D2[N+I]:=106;
'FOR' J:=1 'STEP' 1 'UNTIL' M 'DO' A[I, N+J]:=A[I, J]:=0;
A[I, N+I]:=AOP[I, I]:='IF' R<0 'THEN'-1 'ELSE' 1
'END' I; N:=N+M
'END' START1;
```

В программе, где будет применяться эта процедура, параметры задачи $A, c, x, d_*, d^*, b_*, b^*$ описываются следующими массивами: $A[1:M, 1:N+M]$; $C, X, D1, D2[1:N+M]$; $B[1:M]$. Перед выполнением алгоритма, ориентированного на известный план, необходимо обратиться к процедуре START1. Если в результате решения расширенной задачи все дополнительные переменные x_{n+i} , $i = \overline{1, m}$, будут равны нулю, то x_j , $j = \overline{1, n}$, — компоненты оптимального плана; в противном случае основные ограничения задачи несовместны.

В задачах с интервальными ограничениями ($b_* \leq Ax \leq b^*$) применяется аналогичная стартовая процедура с той лишь разницей, что опора строится только по

нарушенным ограничениям и, следовательно, ее размер $k \leq m$. Пусть в качестве начального приближения задан вектор x , относительно которого неизвестно, удовлетворяет ли он ограничениям задачи. Тогда перед обращением к процедуре решения задачи выполним следующие операции.

1. Если $x_j < d_{*j}$, то положим $x_j = d_{*j}$; если $x_j > d_j^*$, то — $x_j = d_j^*$, $j \in J$.

2. Вычислим вектор $w = Ax$. Положим $k = 0$, $i = 1$.

3. Вычислим

$$R = \begin{cases} b_i^* - w_i, & \text{если } w_i > b_i^*, \\ b_{*i} - w_i, & \text{если } w_i < b_{*i}, \\ 0 & , \text{если } b_{*i} \leq w_i \leq b_i^*. \end{cases}$$

4. Если $|R| < \varepsilon$, то перейдем к п. 8.

5. Положим $k = k + 1$, $x_{n+k} = |R|$, $c_{n+k} = -M$ (большое число), $d_{*n+k} = 0$, $d_{n+k}^* = \infty$.

6. Положим $A_{\text{оп}}^{-1}(k, k) = A(i, n+k) = \text{sign } R$; $A(j, n+k) = 0$, $j = \overline{1, m}$, $j \neq i$; $A_{\text{оп}}^{-1}(j, k) = A_{\text{оп}}^{-1}(k, j) = 0$, $j = \overline{1, k-1}$.

7. Положим $J_{\text{оп}}(k) = n+k$, $I_{\text{оп}}(k) = i$.

8. Положим $i = i + 1$. Если $i \leq m$, то перейдем к п. 3.

9. Положим $n = n + k$.

Рассмотрим АЛГОЛ-процедуру, реализующую эти операции в случае, когда ненулевые элементы матрицы A запакованы в ячейки вместе со своими индексами. Адреса массива A , с которых начинаются столбцы матрицы, хранятся в массиве целых чисел $\text{NCA}[1:N+M+1]$.

```
'PROCEDURE' START2;
'BEGIN' 'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
'BEGIN' 'IF' X[J]<D1[J] 'THEN' X[J]:=D1[J];
'IF' X[J]>D2[J] 'THEN' X[J]:=D2[J] 'END';
'FOR' I:=1 'STEP' 1 'UNTIL' M 'DO' W[I]:=0; K:=0;
'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
'FOR' I:=NCA[J]+1 'STEP' 1 'UNTIL' NCA[J+1] 'DO'
'BEGIN' PLA0R1 (A [I], I1); W [I1]:=W [I1]+A [I] * X [J] 'END';
'FOR' I:=1 'STEP' 1 'UNTIL' M 'DO'
'BEGIN' R:=0; 'IF' W[I]>B2[I] 'THEN' R:=B2[I]-W[I];
'IF' W[I]<B1[I] 'THEN' R:=B1[I]-W[I];
'IF' ABS(R)>EPS 'THEN'
'BEGIN' K:=K+1; X[N+K]:=ABS(R); AK:=AK+1;
A[AK]:=AOP[K, K]:=SIGN(R); PLA031 (A[AK], I);
```



```

NCA[N+1+K]:=NCA[N+K]+1;
'FOR'J:=1 'STEP' 1 'UNTIL'K-1 'DO'
AOP[J, K]:=AOP[K, J]:=0;
C[N+K]:=-107; D2[N+K]:=-106; D1[N+K]:=0;
JOP[K]:=N+K; IOP[K]:=I
'END'
'END'I; N:=N+K
'END'START2;

```

Здесь PLA031 — процедура запаковки элемента массива и его индекса в одну ячейку; PLA0R1 — процедура выделения индекса из младших разрядов элемента массива; AOP [1: M, 1: M] — массив для матрицы $A_{оп}^{-1}$; A [1: AK + + M] — массив для матрицы A; AK — число ее ненулевых элементов.

В массивах X, C, D1, D2[1: N+M]; B1, B2[1: M] располагаются векторы x, c, d_*, d^*, b_*, b^* и компоненты расширенной задачи. Если при решении расширенной задачи переменная x_{n+j} станет равной нулю, то $n+j$ -й столбец исключается (вместе с c_{n+j}) и размер n задачи понижается на единицу.

Рассмотрим более простую стартовую процедуру, расширяющую исходную задачу только на один столбец при введении дополнительной переменной x_{n+1} . Она состоит из следующих операций.

1. При $x_j < d_{*j}$ положим $x_j = d_{*j}$; при $x_j > d_j^*$ положим $x_j = d_j^*$.
2. Вычислим вектор $w = Ax$. Положим $k=0, i=1$.
3. Если $w_i > b_i^*$ или $w_i < b_{*i}$, то положим $R = (b_i^* + + b_{*i})/2 - w_i$, иначе $R = 0$.
4. Если $|R| > \epsilon$, то положим $k=k+1, A(i, N+1) = R$.
5. При $i < m$ положим $i=i+1$ и перейдем к п. 3.
6. Если $k > 0$, то положим $c_{n+1} = -M, x_{n+1} = 1, d_{*n+1} = 0, d_{n+1}^* = \infty, n = n + 1$.

Приведем реализацию этой процедуры на языке АЛГОЛ-60 при тех же данных, которые применялись в START2.

```

'PROCEDURE' START3;
'BEGIN' 'FOR'J:=1 'STEP'1 'UNTIL' N 'DO'
'BEGIN' 'IF' X[J]<D1[J] 'THEN' X[J]:=D1[J];
'IF' X[J]>D2[J] 'THEN' X[J]:=D2[J] 'END';
'FOR'I:=1 'STEP' 1 'UNTIL' M 'DO' W[I]:=0; K:=0;
'FOR'J:=1 'STEP'1 'UNTIL' N 'DO'

```

```

'FOR'I:=NCA[J]+1 'STEP'1 'UNTIL' NCA[J+1] 'DO'
'BEGIN' PLA0R1 (A [I], I1); W [I1] :=W [I1]+A [I] * X [J] 'END';
'FOR'I:=1 'STEP'1 'UNTIL' M 'DO'
'BEGIN' R:=0; 'IF' W[I]>B2[I] 'OR' W[I]<B1[I]'THEN'
R:=(B2[I]+B1[I])/2-W[I];
'IF' ABS(R)>EPS 'THEN' 'BEGIN' K:=K+1;
I1:=NCA[N+1]+K; A[I1]:=R; PLA031 (A[I1], I) 'END'
'END'I;
'IF' K>0 'THEN' 'BEGIN' X[N+1]:=1; C[N+1]:=-107;
D1[N+1]:=0; D2[N+1]:=106; NCA[N+2]:=NCA[N+1]+K;
N:=N+1 'END'; K:=0
'END' START3;

```

Массивы для этой процедуры следующие: C , X , $D1$, $D2[1:N+1]$; $A[1:AK+M]$, $NCA[1:N+2]$. Размер опоры после работы $START3$ остается равным 0; опора будет «накапливаться» на итерациях адаптивного алгоритма. Компоненты расширенного плана в этой процедуре строятся так, чтобы основные ограничения задачи были пассивными, что удобно при выборе подходящего направления на первых итерациях адаптивного алгоритма. Как показала практика, применение процедуры $START3$ существенно сокращает время решения задачи в сравнении с процедурой $START2$.

Рассмотренные стартовые процедуры обычно вставляются в описание основных процедур, ориентированных на решение задачи с известным начальным планом. Если не известно, является ли заданный вектор x планом задачи, то следует выполнить стартовую процедуру (если x — план, расширения задачи не произойдет). Следовательно, алгоритм решения задачи можно всегда начинать с обращения к стартовой процедуре. В прикладных задачах основная проблема нередко в том и состоит, чтобы найти допустимое решение. Поэтому построение эффективных стартовых процедур, учитывающих специфику задачи и основной алгоритм для ее решения, играет немаловажную роль в создании программных комплексов для решения больших задач линейного программирования.

Глава 2. БОЛЬШИЕ ЗАДАЧИ

Размеры прикладных задач, возникающих в различных сферах человеческой деятельности, постоянно опе-

режают развитие вычислительной техники, темпы которого по общему признанию поразительны. В связи с этим стало распространенным понятие «большие задачи». Нет особой нужды искать для него строгие определения. В каждом конкретном случае ясно, о чем идет речь. Чаще всего говорят о больших задачах, если на имеющейся ЭВМ невозможно реализовать стандартный алгоритм или алгоритм допускает реализацию, но работает ненадежно. Тогда опять говорят, что задача большая. Задачу относят к большой и в том случае, если она решается с большими затратами ресурсов.

В больших задачах можно выделить как общие черты, так и специфические. К общим относят наличие большого числа или переменных, или ограничений. Специфическими чертами являются структура ограничений и структура множества параметров. Вообще говоря, специальных задач со специфическими чертами столько же, сколько вообще задач. В современной теории методов оптимизации идет непрерывный процесс выделения из общих задач достаточно содержательных специальных классов, относящихся к актуальным приложениям. Параллельно идет развитие методов, учитывающих особенности новых классов. В этом один из наиболее перспективных путей развития методов решения больших задач. Но на любом этапе будут играть известную роль и методы, работающие на нескольких классах задач. В хорошем матобеспечении ЭВМ должна быть представлена многоуровневая иерархия методов — от узкоспециальных до самых общих, учитывающих только общие сведения об экстремальной задаче.

Здесь излагаются только общие принципы решения больших задач, детальная структура множества параметров при этом не учитывается. В § 1 приводится метод учета большого числа переменных. Задачи с большим числом переменных типичны в теории управления, а также встречаются и в других приложениях [30]. Далее (§ 2) описывается метод решения задач с большим числом ограничений-неравенств. Среди задач управления к ним относятся задачи с фазовыми ограничениями. Специальный метод, в котором в ходе итераций используются опоры с размерами, не превосходящими заданных ограничений, строится в § 3. В § 4 обоснован метод решения больших интервальных задач. Параграф 5 посвящен алгоритму решения динамической транспортной задачи.

Здесь поясняются принципы перехода от адаптивного алгоритма решения статической задачи к его динамическому аналогу. В § 6 некоторые из разработанных алгоритмов применяются к решению прикладных задач. Глава завершается обсуждением вопросов реализации алгоритмов на ЭВМ.

§ 1. Задачи с большим числом переменных

В современных машинных реализациях симплекс-метода не накладывается ограничений на число переменных в решаемой задаче. В связи с этим сложность задачи часто определяется только количеством основных ограничений. В известной оценке $(m \div 2m)$ числа итераций симплекс-метода также не фигурирует второй размер (число переменных n) задачи. Может сложиться впечатление, что для симплекс-метода безразлично, сколько переменных в задаче. В действительности же это не так. С увеличением числа переменных растет число итераций. Кроме того, при слишком большой разнице между n и m («узкие», «длинные» задачи) появляется большое число почти коллинеарных векторов условий, что затрудняет процедуру идентификации знака оценок, играющих решающую роль в симплекс-методе. Не случайно, что в методе декомпозиции Данцига — Вулфа, где появляются задачи с большим числом переменных, характеристики симплекс-метода заметно хуже, чем в «нормальных» задачах.

Нетрудно понять, что и в адаптивном методе (см. § 1 гл. 1) число переменных оказывает влияние на ход итераций. В данном параграфе излагаются результаты по развитию адаптивного метода для решения задач с большим числом переменных. Численные эксперименты на ЭВМ показали эффективность предложенных модификаций.

1. Постановка задачи. Рассмотрим каноническую задачу линейного программирования

$$c'x \rightarrow \max, Ax = b, d_* \leq x \leq d^*, \quad (1)$$

где $A = A(I, J)$, $c = c(J)$, $b = b(I)$, $I = \{1, 2, \dots, m\}$, $J = \{1, 2, \dots, n\}$, $\text{rank } A = m$, $n \gg m$.

Специфика задачи с большим числом переменных (1), важная с точки зрения практической реализации описываемых ниже методов, состоит в следующем. По

построению, опорные оценки $\Delta_j=0$, $j \in J_{\text{оп}}$. При $n \gg m$ среди неопорных $(m+1)$ -мерных векторов $\{c_j, a_j\}$, $j \in J_{\text{н}}$, составленных из компонент вектора стоимости c и столбцов матрицы A , найдутся такие, для которых $\Delta_j=0$ или $\Delta_j \approx 0$. Между тем, как и в симплекс-методе, в методах, описанных в гл. 1, знаки оценок Δ_j играют большую роль. Если учесть, что почти каждая реализация метода на ЭВМ сопровождается ошибками округления и что все современные ЭВМ идентифицируют числа лишь с определенной точностью, то станут понятны те трудности, которые проявляются, например, в адаптивном методе в неправильном выборе подходящего направления из-за ошибок в установлении знака Δ_j . Серьезные трудности возникают даже в том случае, если вычисления ведутся абсолютно точно. Компоненты l_j , $j \in J_{\text{н}}$, подходящего направления вычисляются по x_j , d_{*j} , d_{j*}^* , но не зависят от $|\Delta_j|$. Поэтому увеличивается роль компонент x_j с малыми $|\Delta_j|$, хотя по смыслу такие x_j должны меняться незначительно. Может оказаться, что скорость увеличения целевой функции вдоль подходящего направления велика, но допустимый шаг очень мал и мало приращение целевой функции. Для преодоления этих трудностей были разработаны специальные методы решения задачи (1).

2. Использование части неопорных переменных для увеличения итерационного шага. Основная идея этого метода состоит в том, чтобы неопорные переменные с малыми $|\Delta_j|$ использовать не для увеличения скорости изменения целевой функции вдоль подходящего направления, а направить их в помощь опорным переменным с целью получения максимально допустимого шага. В силу малости $|\Delta_j|$ вариации переменной x_j мало скажутся на скорости изменения целевой функции, но не исключено, что они сильно повлияют на скорость изменения переменных x_j , $j \in J_{\text{оп}}$, определяющих величину допустимого шага Θ . Компоненты плана с большими оценками $|\Delta_j|$ целесообразно оставить для формирования подходящего направления по схеме адаптивного метода (см. гл. 1). Из описания идеи метода видно, что в нем удастся уменьшить влияние ошибок округления на процесс решения.

Выберем число $\xi \geq 0$ — параметр метода и разобьем множества $J = \{1, 2, \dots, n\}$, $J_{\text{н}}$ на подмножества $J_{\text{м}} = \{j \in J: |\Delta_j| \leq \xi\}$, $J_{\text{нб}} = \{j \in J_{\text{н}}: |\Delta_j| > \xi\}$, $J_{\text{нм}} = J_{\text{н}} \cap J_{\text{м}}$. По-

строим компоненты допустимого направления, соответствующие большим оценкам:

$$\begin{aligned} l_j &= d_{*j} - x_j \text{ при } \Delta_j > \xi; \\ l_j &= d_j^* - x_j \text{ при } \Delta_j < -\xi, \quad j \in J_{\text{нб}}. \end{aligned} \quad (2)$$

Компоненты l_j , $j \in J_M$, допустимого направления l построим так, чтобы шаг Θ вдоль l был максимальным, т. е.

$$\Theta \rightarrow \max_{l_j, j \in J_M}, \quad (3)$$

где, как известно (см. гл. 1),

$$\begin{aligned} \Theta &= \min \{ \Theta_j, j \in J_M; 1 \}, \\ \Theta_j &= \begin{cases} (d_{*j} - x_j)/l_j & \text{при } l_j < 0, \\ (d_j^* - x_j)/l_j & \text{при } l_j > 0, \\ \infty & \text{при } l_j = 0 \end{cases} \end{aligned} \quad (4)$$

и переменные l_j , $j \in J_M$, удовлетворяют условию допустимости направления:

$$A_M l_M + b_6 = 0, \quad b_6 = A_{\text{нб}} l_{\text{нб}}. \quad (5)$$

При $\Theta = 1$ план $\bar{x} = x + l$ β -оптимален,

$$\beta = \sum_{\Delta_j > 0, j \in J_{\text{нм}}} \Delta_j (\bar{x}_j - d_{*j}) + \sum_{\Delta_j < 0, j \in J_{\text{нм}}} \Delta_j (\bar{x}_j - d_j^*).$$

Если $\beta > \varepsilon$, то число ξ следует уменьшить и вновь повторить итерацию. При $\Theta < 1$ строим новый план $\bar{x} = x + \Theta l$. Ограничения (4) — (5) на шаг Θ и вектор l_M получены из соотношений: $d_{*M} - x_M \leq \Theta l_M \leq d_M^* - x_M$, $A_M l_M + b_6 = 0$, $0 \leq \Theta \leq 1$. Если последнее равенство умножить на Θ и перейти к новым переменным $s_M = \Theta l_M$, то получим следующую, эквивалентную задачу для вычисления максимального плана:

$$\begin{aligned} \Theta \rightarrow \max, \quad A_M s_M + b_6 \Theta = 0, \quad d_{*M} - x_M \leq s_M \leq d_M^* - x_M, \\ 0 \leq \Theta \leq 1. \end{aligned} \quad (6)$$

В качестве начального опорного плана этой задачи можно взять $\{\Theta^*, s_M^*, A_{\text{оп}}\}$, где $A_{\text{оп}}$ — опора плана x , $s_M^* = \{s_{\text{оп}}^*, s_{\text{нм}}^*\}$, $s_{\text{нм}}^* = 0$, $s_{\text{оп}}^* = -\Theta^* A_{\text{оп}}^{-1} b_6$, $\Theta^* = \min_{j \in J_{\text{оп}}} \Theta_j^*$, $\Theta_j^* = (d_{*j} - x_j)/l_j^*$, $l_j^* < 0$, $\Theta_j^* = (d_j^* - x_j)/l_j^*$, $l_j^* > 0$, $\Theta_j^* = \infty$, $l_j = 0$, $l_{\text{оп}}^* = -A_{\text{оп}}^{-1} b_6$. Задачу (6) решаем с помощью

алгоритма с адаптивной нормировкой (см. § 4 гл. 1). На первой итерации оценки всех неопорных векторов условий a^j , $j \in I_{\text{нм}}$, равны нулю, за исключением оценки $\Delta_\Theta = -1$ вектора b_6 . Следовательно, при $\Theta < 1$ на итерации среди неопорных переменных увеличивается только Θ .

Пусть i_0 — индекс опорной переменной s_{i_0} , которая при увеличении Θ на $[0, 1]$ первой вышла на одну из границ прямых ограничений (i_0 ищется по стандартным формулам типа (4)).

Введем множество $J^* = \{j \in J_{\text{м}} \setminus J_{\text{оп}}: d_{*j} - x_j < s_j < d_j^* - x_j \text{ при } r_{i_0j} \neq 0; s_j \neq d_{*j} - x_j \text{ при } kr_{i_0j} > 0; s_j \neq d_j^* - x_j \text{ при } kr_{i_0j} < 0\}$, где

$$k = \begin{cases} 1, & \text{если } s_{i_0} \text{ достигает нижней границы,} \\ -1, & \text{если } s_{i_0} \text{ достигает верхней границы,} \end{cases}$$

$$r_{i_0j} = \sum_{g=1}^m u_{i_0g} a_{gj} = e'_{i_0} A_{\text{оп}}^{-1} A(J, j).$$

Если $J^* = \emptyset$, то значение $\tilde{\Theta}$, на котором остановился процесс увеличения неопорной переменной Θ , равно максимальному шагу вдоль l (так как после замены в опоре вектора a_{i_0} на b_6 будет выполнен критерий оптимальности задачи (6)).

Опору $\tilde{A}_{\text{оп}}$, на которой $J^* = \emptyset$, приписываем в качестве промежуточной опоры плану $\bar{x} = x + \tilde{\Theta}l$, $l = (l_{\text{м}}, l_{\text{нб}})$, $l_{\text{м}} = \tilde{s}_{\text{м}}/\tilde{\Theta}$, $l_{\text{нб}}$ вычислено согласно (2), если $\tilde{\Theta} \neq 0$; $l = 0$, если $\tilde{\Theta} = 0$; $\{\tilde{s}_{\text{м}}, \tilde{\Theta}\}$ — решение задачи (6).

В задаче (1) по опоре $\tilde{A}_{\text{оп}}$ вычисляем оценки $\tilde{\Delta}_j$ неопорных векторов условий и псевдоплан $\tilde{\kappa}$. Если $\tilde{\kappa}_{i_0} < d_{*i_0}$ при $k=1$; $\tilde{\kappa}_{i_0} > d_{i_0}^*$ при $k=-1$, то вывод вектора a_{i_0} из $A_{\text{оп}}$ по правилам адаптивного метода ведет к улучшению $\tilde{A}_{\text{оп}}$ (уменьшению оценки субоптимальности опорного плана $\{\bar{x}, \tilde{A}_{\text{оп}}\}$). Построением опорного плана $\{\bar{x}, \bar{A}_{\text{оп}}\}$, где $\bar{A}_{\text{оп}} = (\tilde{A}_{\text{оп}} \setminus a_{i_0}) \cup a_{j_0}$, завершается итерация в задаче (1).

Если псевдоплан \tilde{x} , построенный по опоре $\tilde{A}_{\text{оп}}$, удовлетворяет соотношениям $\tilde{x}_{i_0} \geq d_{*i_0}$ при $k=1$ или $\tilde{x}_{i_0} \leq d_{*i_0}^*$ при $k=-1$, то промежуточную опору $\tilde{A}_{\text{оп}}$ берем в качестве опоры $\bar{A}_{\text{оп}}$ плана \bar{x} .

Пусть теперь $J^* \neq \emptyset$. В этом случае $\tilde{\Theta}$ не равно максимальному шагу вдоль l . Поэтому итерации в задаче (6) продолжаем. Вектор a_{i_0} удаляем из опоры. Вектор a_{j_0} , вводимый вместо него, находим по правилам адаптивного метода (см. § 1 гл. 1). Двойственный шаг равен нулю ($\sigma=0$), и достигается он на векторах условий a_j , $j \in J^*$, поэтому $j_0 \in J^*$. Среди индексов J^* выбираем прежде всего такой j_0 , для которого $d_{*j_0} - x_{j_0} < \tilde{s}_{j_0} < d_{j_0}^* - x_{j_0}$. В противном случае \tilde{j}_0 — любой элемент из J^* . Ясно, что после замены опоры $A_{\text{оп}} \rightarrow \bar{A}_{\text{оп}}$, $\bar{A}_{\text{оп}} = (A_{\text{оп}} \setminus a_{i_0}) \cup a_{j_0}$, т. е. на следующей итерации в задаче (6), оценки всех векторов условий, кроме вектора b_6 , равны нулю. Таким образом, в случае $J^* \neq \emptyset$ после итерации возникает та же ситуация, что и была до итерации.

3. Использование части неопорных переменных для увеличения итерационного приращения целевой функции. Согласно адаптивному методу неопорные компоненты допустимого направления выбираются из условия максимального приращения целевой функции в пространстве не всех переменных, а только неопорных. При $n \gg t$ истинное приращение целевой функции на итерации может существенно отличаться от его значения, полученного при построении допустимого направления. Это происходит всякий раз, когда оказывается малым шаг Θ вдоль допустимого направления. В п. 2 для увеличения Θ была использована часть неопорных переменных. Однако максимизация шага не является основной целью итерации. Главная задача всех одношаговых методов оптимизации состоит в максимальном приращении целевой функции. В данном пункте для решения этой задачи используются компоненты допустимого направления, соответствующие малым оценкам.

Из физического смысла вектора оценок Δ следует, что приращение целевой функции при изменении компонент $x_{\text{н}}$ равно $-\Delta'_{\text{н}} l_{\text{н}} = -\Delta'_{\text{нм}} l_{\text{нм}} - \Delta'_{\text{нб}} l_{\text{нб}}$. Компоненты $l_{\text{нб}}$ построим по (2). Обозначим: $\beta_{\text{нб}} = -\Delta'_{\text{нб}} l_{\text{нб}}$. Максимум по $l_{\text{нм}}$ величины $-\Delta'_{\text{н}} l_{\text{н}}$ достигается одновременно с макси-

мумом по $\Theta \in [0, 1]$ и $s_{\text{нм}} = \Theta l_{\text{нм}}$ выражения $-\Delta'_{\text{нм}} s_{\text{нм}} + \Theta \beta_{\text{нб}}$. Множество допустимых значений для Θ , $s_{\text{м}}$ описано в п. 2.

Таким образом, максимизация итерационного приращения целевой функции при заданном $l_{\text{нб}}$ сводится к задаче

$$\Theta \beta_{\text{нб}} - \Delta'_{\text{нм}} s_{\text{нм}} \rightarrow \max, \quad (7)$$

$$A_{\text{м}} s_{\text{м}} + b_6 \Theta = 0, \quad d_{*\text{м}} - x_{\text{м}} \leq s_{\text{м}} \leq d_{\text{м}}^* - x_{\text{м}}, \quad 0 \leq \Theta \leq 1.$$

Эта задача отличается от задачи (6) дополнительным слагаемым в целевой функции, которое характеризует вклад в приращение переменных с малыми оценками. Если $\Delta_{\text{нм}} = 0$, то задача (7) сводится к задаче (6), т. е. в этом случае максимальный шаг обеспечивает максимальное итерационное приращение целевой функции. Задачи (6) и (7) можно решать одним и тем же методом.

Приведем алгоритм решения задачи (1), в котором на каждой итерации формируется и решается вспомогательная задача. Алгоритм решения вспомогательной задачи реализует схему адаптивного метода с простым двойственным шагом. Вычислительная схема основного алгоритма основана на адаптивном методе с оптимальной заменой опоры.

4. Алгоритм. 1. Находим оценки $\Delta_j = u' a^j - c_j$, $j \in J_{\text{н}}$, где $u' = c'_{\text{оп}} A_{\text{оп}}^{-1}$; $\Delta_j = 0$, $j \in J_{\text{оп}}$.

2. Проверяем критерий оптимальности и вычисляем

$$\beta = \sum_{\Delta_j > 0, j \in J_{\text{н}}} \Delta_j (x_j - d_{*j}) - \sum_{\Delta_j < 0, j \in J_{\text{н}}} \Delta_j (d_j^* - x_j).$$

Если $\beta < \varepsilon$, то x — ε -оптимальный план и переходим в п. 20.

3. Среди оценок $|\Delta_j|$, $j \in J_{\text{н}}$, выбираем максимальную и задаем параметр отсечения на данной итерации: $\xi = \tau |\Delta|_{\max}$, где τ — константа отсечения, $\tau \in (0, 1)$ (например, $\tau = 0,01$).

4. По параметру ξ строим множества $J_{\text{нм}} = \{j \in J_{\text{н}} : |\Delta_j| \leq \xi\}$, $J_{\text{нб}} = J_{\text{н}} \setminus J_{\text{нм}}$ и для $j \in J_{\text{нб}}$ строим компоненты вектора направления l по формулам (2).

5. По формуле (5) строим вектор b_6 .

6. Вычисляем опорные компоненты вектора направления $l_{\text{оп}} = -A_{\text{оп}}^{-1} b_6$.

7. Вычисляем максимально допустимый шаг вдоль направления l :

$$\Theta_j = \begin{cases} (d_{*j} - x_j)/l_j & \text{при } l_j < 0, \\ (d_j^* - x_j)/l_j & \text{при } l_j > 0, \\ \infty & \text{при } l_j = 0, j \in J_{\text{оп}}, \end{cases}$$

$$\Theta_{j_0} = \min_{j \in J_{\text{оп}}} \Theta_j, \quad \Theta = \min \{1; \Theta_{j_0}\}.$$

Запоминаем номер j_0 .

8. Если $\Theta = 1$, то вычисляем план $\bar{x} = x + l$ и число

$$\beta = \sum_{\Delta_j > 0, j \in J_{\text{нм}}} \Delta_j (\bar{x}_j - d_{*j}) + \sum_{\Delta_j < 0, j \in J_{\text{нм}}} \Delta_j (\bar{x}_j - d_j^*).$$

При $\beta \leq \varepsilon$ план \bar{x} — ε -оптимален; переходим на конец алгоритма — в п. 20. При $\beta > \varepsilon$ полагаем $\xi = \tau \xi$ и переходим в п. 4.

9. Если $\Theta < 1$ и $J_{\text{нм}} = \emptyset$, то переходим в п. 11. Если $\Theta < 1$ и $J_{\text{нм}} \neq \emptyset$, то вычисляем компоненты начального опорного плана $s: s_{\text{оп}} = \Theta l_{\text{оп}}, s_{\text{нм}} = 0$ и решаем вспомогательную задачу. Получим $l(J_{\text{н}}), \tilde{A}_{\text{оп}}^{-1}, \Theta$.

10. Вычисляем вектор оценок $\tilde{\Delta}(J_{\text{н}})$. В случае совпадения знаков оценок Δ_j и $\tilde{\Delta}_j, j \in J_{\text{нб}}$, где $\tilde{\Delta}_j = u' A(I, j) - c_j, u' = c'_{\text{оп}} \tilde{A}_{\text{оп}}^{-1}$, операции 14—19 двойственного шага можно выполнить только для $J_{\text{нб}}$.

11. Вычисляем новый план $\bar{x} = x + \Theta l$.

12. Вычисляем псевдоплан $\kappa_j = d_{*j}$, если $\Delta_j > 0$; $\kappa_j = d_j^*$, если $\Delta_j < 0, j \in J_{\text{н}}, \kappa_{\text{оп}} = A_{\text{оп}}^{-1}(b - A_{\text{н}} \kappa_{\text{н}})$.

13. Если $\kappa_{j_0} \geq d_{*j_0}$ при $k = 1$ или $\kappa_{j_0} \leq d_{j_0}^*$ при $k = -1$, то, полагая $\bar{A}_{\text{оп}}^{-1} = \tilde{A}_{\text{оп}}^{-1}$, переходим в п. 1.

14. Определяем подходящее направление t для коплан Δ :

$$t_j = k \sum_{g=1}^m u_{j_0 g} a_{gj} = k e_{j_0} A_{\text{оп}}^{-1} A(I, j), j \in J_{\text{н}}.$$

15. Находим начальную скорость убывания двойственной функции $\alpha = \kappa_{j_0} - d_{*j_0}$ при $k = 1$; $\alpha = d_{j_0}^* - \kappa_{j_0}$ при $k = -1$.

16. Выявляем среди $\Delta_j, j \in J_{\text{н}}$, нулевые компоненты и подсчитываем α по формуле (18) § 1 гл. 1. Если среди элементов, по которым вычислены положительные добавки в (18) § 1 гл. 1, есть такие, что $d_{*j} < x_j < d_j^*, j \in J_{\text{н}}$, то полагаем $j_* = j$. Иначе в качестве j_* берем любой

элемент, по которому вычислена положительная добавка.

17. Если $\alpha \geq 0$, то преобразуем $A^{-1} = \tilde{A}_{\text{оп}}^{-1}$ по формуле (22) § 1 гл. 1 и, полагая $\bar{J}_{\text{оп}} = (J_{\text{оп}} \setminus j_0) \cup j_*$, переходим в п. 1.

18. Вычисляем двойственный шаг $\sigma = \min \{-\Delta_j/t_j\}$, $t_j \Delta_j < 0$, $j \in J_{\text{н}}$.

19. Пересчитываем коплан $\bar{\Delta}_j = \Delta_j + \sigma t_j$, $j \in J_{\text{н}}$, и переходим в п. 16.

20. Выдача результатов: x , $c'x$, $b - Ax$.

Решение задачи (7) можно получить с помощью стандартных алгоритмов (см., например, § 4 или § 5 гл. 1). При решении задачи (6) удобно воспользоваться ее спецификой (оценки неопорных векторов условий равны нулю, за исключением оценки $\Delta_\Theta = -1$ вектора b_Θ). Кроме того, итерацию в задаче (6) лучше начинать с преобразования опоры, так как первый шаг Θ вдоль l выполняется в основном алгоритме, а начальным планом задачи (6) является вектор Θl с опорой $A_{\text{оп}}$, при которой l было вычислено. Рассмотрим алгоритм для решения задачи (6).

5. Алгоритм решения вспомогательной задачи. Исходной информацией для алгоритма является: начальный план ($s_{\text{оп}} = \Theta l_{\text{оп}}$, $s_{\text{нм}} = 0$, Θ), обратная матрица $A_{\text{оп}}^{-1}$, множество индексов $J_{\text{оп}}$, вектор b_Θ , индекс j_0 .

1. Проверим критерий оптимальности и найдем элемент для замены опоры. Если среди элементов $j \in J_{\text{нм}}$ найдется такой, для которого $d_{*j} - x_j < s_j < d_j^* - x_j$ при $r_{j_0j} \neq 0$, то полагаем $j_* = j$; в противном случае в качестве j_* берем любой элемент, для которого $s_j \neq d_{*j} - x_j$ при $kr_{j_0j} > 0$ или $s_j \neq d_j^* - x_j$ при $kr_{j_0j} < 0$, где $k = -1$, если $s_{j_0} = d_{j_0}^* - x_{j_0}$; $k = 1$, если $s_{j_0} = d_{*j_0} - x_{j_0}$; $r_{j_0j} = = e'_{j_0} A_{\text{оп}}^{-1} A(l, j)$. Если элемент j_* остался неопределенным (ни одно из условий не выполнилось), то максимум Θ достигнут и, следовательно, переходим в п. 7.

2. Преобразуем опору. По формуле (22) § 1 гл. 1 пересчитаем элементы новой обратной матрицы $\bar{A}_{\text{оп}}^{-1}$ и положим $\bar{J}_{\text{оп}} = (J_{\text{оп}} \setminus j_0) \cup j_*$.

3. Неопорные компоненты подходящего направления \bar{l} для плана s положим равными нулю (кроме $\bar{l}_\Theta = 1 - \Theta$). Опорные компоненты найдем по формуле $\bar{l}_{\text{оп}} = (1 - \Theta) A_{\text{оп}}^{-1} b_\Theta$.

4. Найдем максимально допустимый шаг $\tilde{\Theta}$ вдоль направления \tilde{l} : $\tilde{\Theta} = \min \{1, \tilde{\Theta}_{j_0}\}$, $\tilde{\Theta}_{j_0} = \min_{j \in J_{\text{оп}}} \tilde{\Theta}_j$, $\tilde{\Theta}_j = (d_{*j} - x_j - s_j)/\tilde{l}_j$ при $\tilde{l}_j < 0$, $\tilde{\Theta}_j = (d_j^* - x_j - s_j)/\tilde{l}_j$ при $\tilde{l}_j > 0$, $\tilde{\Theta}_j = \infty$ при $\tilde{l}_j = 0$, $j \in J_{\text{оп}}$. При $\tilde{\Theta} < 1$ запомним индекс j_0 .

5. Если $\tilde{\Theta} = 1$, то, положив $\Theta = 1$, $s_{\text{оп}} = s_{\text{оп}} + \tilde{l}_{\text{оп}}$, перейдем в п. 7.

6. Вычислим новый план $\bar{s}_j = s_j + \bar{\Theta} l_j$, $j \in J_{\text{оп}}$, $\bar{\Theta} = \Theta + \tilde{\Theta}(1 - \Theta)$, и перейдем в п. 1.

7. По найденному плану s вычислим компоненты допустимого направления l в основной задаче (1): $l_j = s_j/\Theta$, $j \in J_M = J_{\text{оп}} \cup J_{\text{нм}}$.

Табл. 23

τ	Вспомогательная задача			Основная задача	
	$ J_{\text{нм}} $	Число итераций на одном шаге основного алгоритма	Число итераций	Число итераций	Время решения, с
0.008	0—4	1—6	41	26	8
	0—3	0—4	14	13	4
	0—3	1—4	19	21	7
	0—4	1—8	24	14	4
	0—2	0—4	35	20	7
0.032	4—9	1—19	67	12	6
	4—12	6—25	65	10	5
	2—8	1—20	58	10	5
	0—9	1—13	86	18	9
	1—11	0—14	73	15	8
0.05	4—12	1—23	84	11	7
	4—19	2—29	196	25	19
	1—14	1—26	181	27	16
	4—12	1—22	76	11	6
	2—18	2—24	112	14	9
0.128	2—32	0—64	428	70	119
	13—40	12—76	999	146	326

6. Численный эксперимент. Описанный алгоритм реализован на языке АЛГОЛ-ГДР. На ЭВМ БЭСМ-6 проведен эксперимент. Цель эксперимента — при разных значениях константы отсечения τ сравнить число итераций вспомогательного и основного алгоритмов с числом итераций стандартных алгоритмов § 4, 5 гл. 1 (см. табл. 8). Результаты эксперимента при $m=10$, $n=100$ приведены в табл. 23.

Сравнивая эти результаты с приведенными в табл. 8 результатами решения тех же задач, можно заключить, что при малых значениях рассматриваемый алгоритм незначительно уступает адаптивному алгоритму § 1 и существенно превосходит алгоритм с симплексной нормировкой по числу итераций. При этом в алгоритме число итераций всегда меньше, чем в алгоритмах § 4, 5 гл. 1. Применительно к большим задачам изложенный алгоритм можно считать декомпозиционным, поскольку в процессе его работы происходит разбиение множества неопорных компонент на небольшие группы. Для решения задач с небольшим числом переменных можно применить более эффективные стандартные алгоритмы.

Как показывает эксперимент, большие значения параметра отсечения $\tau=0,128$ существенно снижают эффективность алгоритма и поэтому не могут быть рекомендованы для практических расчетов.

§ 2. Задачи с большим числом ограничений

Знакомясь с линейным программированием, никто не оставляет без внимания геометрический метод, с помощью которого можно легко решить любую $m \times 2$ -задачу с ограничениями-неравенствами. При $n \geq 3$ метод реализовать трудно. Стандартный симплекс-метод, примененный к задачам рассматриваемого типа, не эффективен, ибо он предварительно сводит исходную $m \times n$ -задачу к $m \times (n+m)$ -задаче, т. е. увеличивает размеры исходной задачи. Специальная структура ограничений давно привлекает внимание специалистов. Известно много подходов, учитывающих специфику ограничений. Предложены разнообразные стратегии перебора среди них активных, но пока не найдено лучшего. Исследования стимулируются тем фактом, что в общем случае для построения оптимального плана достаточно знать только n ограничений.

В данном параграфе описывается детализация адаптивного метода, учитывающая специфику ограничений. Из принципа уменьшения оценки субоптимальности получается своеобразная стратегия перебора активных ограничений.

1. Постановка задачи. Рассмотрим интервальную задачу линейного программирования

$$c'x \rightarrow \max, b_* \leq Ax \leq b^*, d_* \leq x \leq d^*,$$

где c, x, d_*, d^* — n -векторы, b_*, b^* — m -векторы, $A = A(I, J)$ — $m \times n$ -матрица. Специфика задачи в том, что $m \gg n$. Прямые ограничения $d_* \leq x \leq d^*$ включим в основные (в A добавляется n единичных строк) и будем рассматривать задачу

$$c'x \rightarrow \max, b_* \leq Ax \leq b^*. \quad (1)$$

Неособую $n \times n$ -матрицу $A_{\text{оп}} = A(I_{\text{оп}}, J)$, $I_{\text{оп}} \subset I$, назовем опорой задачи (1); пару $\{x, A_{\text{оп}}\}$ из плана и опоры — опорным планом, который считается невырожденным, если $b_*(I_{\text{н}}) < A(I_{\text{н}}, J)x < b^*(I_{\text{н}})$, $I_{\text{н}} = I \setminus I_{\text{оп}}$. Пусть $\{x, A_{\text{оп}}\}$ — начальный опорный план. Введем обозначения: $I_{*\text{оп}} = \{i: i \in I_{\text{оп}}, A'(i, J)x = b_{*i}\}$, $I_{*\text{оп}}^* = \{i: i \in I_{\text{оп}}, A'(i, J)x = b_{*i}^*\}$, $\tilde{I}_{\text{оп}} = I_{\text{оп}} \setminus (I_{*\text{оп}} \cup I_{*\text{оп}}^*)$, $\omega_*(I_{\text{оп}}) = b_*(I_{\text{оп}}) - A_{\text{оп}}x$, $\omega^*(I_{\text{оп}}) = b^*(I_{\text{оп}}) - A_{\text{оп}}x$.

2. Формула приращения. Построим n -вектор потенциалов $u' = c' A_{\text{оп}}^{-1}$. Формула приращения целевой функции при учете только основных ограничений $b_* \leq Ax \leq b^*$ примет вид (см. § 1 гл. 1):

$$c' \Delta x = u' z, \quad (2)$$

где $z = z(I_{\text{оп}}) = A_{\text{оп}} \Delta x$. Отсюда следует

Критерий оптимальности. Соотношения

$$u(I_{*\text{оп}}^*) \geq 0, u(I_{*\text{оп}}) \leq 0, u(\tilde{I}_{\text{оп}}) = 0 \quad (3)$$

достаточны, а в случае невырожденности и необходимы для оптимальности опорного плана $\{x, A_{\text{оп}}\}$.

Максимум приращения (2) при ограничениях $\omega_*(I_{\text{оп}}) \leq z \leq \omega^*(I_{\text{оп}})$ равен

$$\beta = \sum_{u_i > 0, i \in I_{\text{оп}}} u_i \omega_i^* + \sum_{u_i < 0, i \in I_{\text{оп}}} u_i' \omega_{*i}. \quad (4)$$

Число β — оценка субоптимальности. Если $\beta \leq \varepsilon$, то x — ε -оптимальный план ($c'x^0 - cx \leq \beta \leq \varepsilon$).

3. Разложение оценки субоптимальности. Задача, двойственная (1), имеет вид

$$b^*{}'\omega - b^*{}'v \rightarrow \min, \quad A'y = c, \quad y + v - \omega = 0, \quad \omega \geq 0, \quad v \geq 0. \quad (5)$$

По опоре $A_{\text{оп}}$ построим двойственный план (y, v, ω) : $y(I_{\text{оп}}) = u(I_{\text{оп}})$, $y(I_{\text{н}}) = 0$; $\omega_i = y_i$, $v_i = 0$, если $y_i \geq 0$, $\omega_i = 0$, $v_i = -y_i$, если $y_i < 0$, $i \in I$, который назовем сопровождающим двойственным планом. Оценку субоптимальности (4) можно представить так:

$$\beta = -u'A_{\text{оп}}x + \sum_{u_i > 0, i \in I_{\text{оп}}} u_i b_i^* + \sum_{u_i < 0, i \in I_{\text{оп}}} u_i b_{*i} = -c'x + b^*{}'\omega - b^*{}'v = c'x^0 - c'x + b^*{}'\omega - b^*{}'v - b^*{}'\omega^0 + b^*{}'v^0 = \beta_x + \beta_{\text{оп}}, \quad (6)$$

где β_x — мера неоптимальности плана x , $\beta_{\text{оп}}$ — мера неоптимальности опоры $A_{\text{оп}}$, v^0 , ω^0 — компоненты оптимального плана задачи (5).

4. Подходящее направление. Предположим, что критерий оптимальности (3) и критерий субоптимальности не выполнены: $\beta > \varepsilon$. Тогда для плана x построим подходящее направление по решению задачи

$$c'l \rightarrow \max, \quad b_{* \text{оп}} \leq A_{\text{оп}}(x + l) \leq b_{\text{оп}}^*. \quad (7)$$

Пусть $\omega = \omega(I_{\text{оп}}) = A_{\text{оп}}l$. Тогда задачу (7) можно записать в виде

$$u'\omega \rightarrow \max, \quad \omega_{* \text{оп}} \leq \omega \leq \omega_{\text{оп}}^*. \quad (8)$$

Решением задачи (8) будет вектор ω : $\omega_i = \omega_i^*$, если $u_i > 0$; $\omega_i = \omega_{*i}$, если $u_i < 0$; $\omega_i = 0$, если $u_i = 0$, $i \in I_{\text{оп}}$, а вектор $l = A_{\text{оп}}^{-1}\omega$ — решение задачи (7).

5. Вычисление шага вдоль l . Максимально допустимый шаг Θ вдоль l найдем по формулам

$$\Theta = \min\{1, \Theta_{i_0}\}; \quad \Theta_{i_0} = \min \Theta_i, \quad i \in I_{\text{н}};$$

$$\Theta_i = \omega_i^*/a_i' l, \quad \text{если } a_i' l > 0; \quad \Theta_i = \omega_{*i}/a_i' l, \quad \text{если } a_i' l < 0;$$

$$\Theta_i = \infty, \quad \text{если } a_i' l = 0. \quad (9)$$

Новый план $\bar{x} = x + \Theta l$ при $\Theta = 1$ будет оптимальным планом, а при $\beta(1 - \Theta) \leq \varepsilon$ — ε -оптимальным. Дальнейшее улучшение оценки субоптимальности β возможно за счет уменьшения компоненты $\beta_{\text{оп}}$ в разложении (6).

6. Формула приращения для двойственной задачи. Из (5) получаем

$$\begin{aligned}
b' \Delta \omega - {}_b \Delta v = & (b_{*H} - \kappa_H)' \Delta y_H + (b_H^* - b_{*H})' \Delta \omega_H + \\
& + \sum_{\substack{y_i=0, \bar{y}_i > 0, \\ u_i=0, i \in I_{\text{оп}}}} (b_i^* - \kappa_i) \bar{y}_i + \sum_{\substack{y_i=0, \bar{y}_i < 0 \\ u_i=0, i \in I_{\text{оп}}}} (b_{*i} - \kappa_i) \bar{y}_i + \\
& + \sum_{\substack{y_i < 0, \bar{y}_i > 0, \\ i \in I_{\text{оп}}}} (b_i^* - b_{*i}) \bar{y}_i + \sum_{\substack{y_i > 0, \bar{y}_i < 0, \\ i \in I_{\text{оп}}}} (b_{*i} - b_i^*) \bar{y}_i, \quad (10)
\end{aligned}$$

где $\kappa = A(x+l)$, $\bar{y} = y + \Delta y$.

Пусть $k = -1$, если $A'(i_0, J) \bar{x} = b_{i_0}^*$; $k = 1$, если $A'(i_0, J) \bar{x} = b_{*i_0}$. Тогда в силу (10) начальная скорость изменения двойственной функции (на невырожденном плане)

$$\alpha_0 = b_{i_0}^* - \kappa_{i_0}, \text{ если } k = -1; \alpha_0 = \kappa_{i_0} - b_{*i_0}, \text{ если } k = 1. \quad (11)$$

7. Замена опорной матрицы. В качестве подходящего направления для двойственного плана возьмем вектор

$$\mu(I_{\text{оп}} \cup I_H) = k \{e'_{i_0} A(I, J) A_{\text{оп}}^{-1} = a'_{i_0} A_{\text{оп}}^{-1}; -e_{i_0}\}. \quad (12)$$

Если $\alpha < 0$, то при движении вдоль μ двойственная целевая функция уменьшается. Как только изменяется знак у y_i , то уменьшается (по модулю) и скорость α . Пусть движению вдоль направления μ соответствует скорость α_s изменения двойственной функции. После появления у вектора y новых нулевых компонент скорость изменения двойственной целевой функции станет равной

$$\alpha_{s+1} = \alpha_s - \sum_{\mu_i < 0, y_i^{(s+1)} = 0} (b_i^* - b_{*i}) \mu_i + \sum_{\mu_i > 0, y_i^{(s+1)} = 0} (b_{*i} - b_i^*) \mu_i. \quad (13)$$

Если $\alpha_{s+1} > 0$, то движение вдоль μ следует, очевидно, прекратить. При этом $\bar{\beta}_{\text{оп}} = \beta_{\text{оп}} - \sum_{i=0}^s |\alpha_i| \sigma_i$ и, если

$(1-\theta)\beta - \sum_{i=0}^s |\alpha_i| \sigma_i \leq \varepsilon$, то \bar{x} — приближенное решение задачи (1). В противном случае из множества $I^0 = \{i : i \in I_{\text{оп}}, y_i^{(s)} = 0, \mu_i > 0, \kappa_i \neq b_i^* \text{ или } y_i^{(s)} = 0, \mu_i < 0, \kappa_i \neq b_{*i}\}$ возьмем элемент i_* , для которого $b_{*i_*} < A'(i_*, J) \bar{x} < b_{i_*}^*$, а если такого нет, то любой элемент $i_* \in I_0$. Из опоры выводим строку $A(i_*, J)$, а вместо нее вводим $A(i_0, J) : \bar{I}_{\text{оп}} = (I_{\text{оп}} \setminus i_*) \cup i_0$, $\bar{A}_{\text{оп}} = (A_{\text{оп}} \setminus A(i_*, J)) \cup A(i_0, J)$. Обратную матрицу при замене строки вычисляем по формуле

$$\begin{aligned}\bar{A}_{\text{оп}}^{-1} &= A_{\text{оп}}^{-1} [E + e_{i_*} (e_{i_*} - A(i_0, J) A_{\text{оп}}^{-1}) / \alpha], \\ \alpha &= A(i_0, J) A_{\text{оп}}^{-1} e_{i_*} = k \mu_{i_*}.\end{aligned}\quad (14)$$

Вектор потенциалов был вычислен в процессе поиска i_* :

$$\bar{u}(I_{\text{оп}} \setminus i_0) = u^{(s)}(I_{\text{оп}} \setminus i_0) = u(I_{\text{оп}} \setminus i_0) + \sum_{i=0}^s |\alpha_i| \sigma_i \mu(I_{\text{оп}} \setminus i_0);$$

$$\bar{u}_{i_*} = -k \sum_{i=0}^s \sigma_i.$$

Поэтому новую итерацию можно начать без его вычисления.

Прежде чем изложить вычислительную схему описанного метода, заметим, что выделение прямых ограничений незначительно уменьшает трудоемкость итерации, если матрица условий A хранится в компактной форме. Алгоритм метода рассмотрим применительно к задаче (1), предполагая, что опорный план $\{x, A_{\text{оп}}\}$ задан.

8. Алгоритм. 1. Вычислим вектор потенциалов $u' = c'_{\text{оп}} A_{\text{оп}}^{-1}$.

2. Найдем вектор $w = Ax$.

3. Вычислим оценку субоптимальности

$$\beta = \sum_{u_i > 0, i \in I_{\text{оп}}} u_i (b_i^* - w_i) + \sum_{u_i < 0, i \in I_{\text{оп}}} u_i (b_{*i} - w_i).$$

Если $\beta \leq \varepsilon$, то переходим в п. 14.

4. Построим вектор $\omega(I_{\text{оп}})$: $\omega_i = b_i^* - w_i$, если $u_i > 0$, $\omega_i = b_{*i} - w_i$, если $u_i < 0$; $\omega_i = 0$, если $u_i = 0$, $i \in I_{\text{оп}}$, и вычислим подходящее направление $l = A_{\text{оп}}^{-1} \omega(I_{\text{оп}})$.

5. Найдем согласно (9) максимально допустимый шаг Θ вдоль l : $\Theta = \min \{1, \Theta_{i_0}\}$, $\Theta_{i_0} = \min \Theta_i$, $i \in I_{\text{н}}$; $\Theta_i = (b_i^* - w_i) / A'(i, J) l$, если $A'(i, J) l > 0$; $\Theta_i = (b_{*i} - w_i) / A'(i, J) l$, если $A'(i, J) l < 0$; $\Theta_i = \infty$, если $A'(i, J) l = 0$. Заметим, что в этом пункте есть смысл различать основные и прямые ограничения, так как вместо компонент векторов b_* , b^* , w , Al будут компоненты векторов d_* , d^* , x , l , когда номер строки i соответствует прямым ограничениям.

6. Вычислим новый план $\bar{x} = x + \Theta l$. При $(1 - \Theta)\beta \leq \varepsilon$ перейдем в п. 14.

7. Найдем, следуя (12), подходящее направление $\mu(I_{\text{оп}})$ для двойственного плана: положив $k = -1$, если $A'(i_0, J) \bar{x} = b_{i_0}^*$; $k = 1$, если $A'(i_0, J) \bar{x} = b_{*i_0}$, вычислим $\mu(I_{\text{оп}}) = k A'(i_0, J) A_{\text{оп}}^{-1}$.

8. По формуле (11) вычислим начальную скорость убывания двойственной функции α ; положим $\sigma=0$, $\beta = (1-\Theta)\beta$, $\kappa(I_{\text{оп}}) = A(I_{\text{оп}}, J)(x+l)$.

9. Среди индексов множества $I^0 = \{i : i \in I_{\text{оп}}, u_i = 0, \mu_i > 0, \kappa_i \neq b_i^* \text{ или } u_i = 0, \mu_i < 0, \kappa_i \neq b_{*i}\}$ фиксируем индекс i_* , для которого $b_{*i_*} < A'(i_*, J)\bar{x} < b_{i_*}^*$. При отсутствии такого в качестве i_* берем любой элемент I^0 .

10. Изменим согласно (13) скорость убывания двойственной функции

$$\alpha = \alpha - \sum_{u_i=0, \mu_i < 0} (b_i^* - b_{*i}) \mu_i + \sum_{u_i=0, \mu_i > 0} (b_i^* - b_{*i}) \mu_i.$$

Если $\alpha \geq 0$, то перейдем в п. 13.

11. Вычислим двойственный шаг $\sigma_s = \min \{-u_i/\mu_i, u_i\mu_i < 0, i \in I_{\text{оп}}\}$.

12. Пересчитаем вектор потенциалов $u(I_{\text{оп}}) = u(I_{\text{оп}}) + \sigma_s \mu(I_{\text{оп}})$, $\beta = \beta + \alpha \sigma_s$, $\sigma = \sigma + \sigma_s$. При $\beta \leq \varepsilon$ перейдем в п. 14, иначе — в п. 9.

13. Положим $u_{i_*} = -k\sigma$, $\bar{I}_{\text{оп}} = (I_{\text{оп}} \setminus i_*) \cup i_0$ и по формуле (14) пересчитаем обратную матрицу $A_{\text{оп}}^{-1}$. Перейдем в п. 2.

14. Выдача результатов.

З а м е ч а н и е. Данный алгоритм можно рассматривать как частный случай алгоритма § 1 гл. 1, когда $J_{\text{оп}} = J$ и размерность опоры фиксирована: $|I_{\text{оп}}| = |J_{\text{оп}}| = n$.

9. Численные эксперименты. Алгоритм реализован на языке АЛГОЛ-ГДР. На БЭСМ-6 проведен ряд экспериментальных расчетов. Задачи генерировались так же, как в эксперименте 1 § 1 гл. 1 (с помощью процедуры GENER1). Каждая из задач, начиная с плана $x=0$, решалась тремя алгоритмами: симплекс-методом, алгоритмом § 1 гл. 1 и данным алгоритмом. Симплекс-метод применялся к задаче $c'x \rightarrow \max$, $Ax = z$, $d_* \leq x \leq d^*$, $b_* \leq z \leq b^*$, и поэтому размер его базисной матрицы был $m \times m$ (на первой итерации она состояла из единичной матрицы при z). Для алгоритма данного параграфа начальная опора была также единичной, но имела размер $n \times n$. Она состояла из дополнительных строк матрицы, соответствующих прямым ограничениям. Векторы b_* и b^* дополнялись векторами d_* и $d^* : b_* = (b_*, d_*)$, $b^* = (b^*, d^*)$. Поскольку $d_* \leq 0$, $b_* \leq 0$, $d^* \geq 0$, $b^* \geq 0$, то $x=0$ — план задачи. Алгоритм § 1 гл. 1 начинал работу с нулевой опоры. Максимальные значения целевой функции, полу-

$n \times m$	Номер задачи	Число итераций			Время решения, с		
		симп-лекс	§ 1 гл. 1	§ 2 гл. 2	симп-лекс	§ 1 гл. 1	§ 2 гл. 2
10×100	1	27	25	20	49	4	4
	2	29	18	19	48	2	4
	3	45	22	23	88	3	4
	4	22	20	16	48	4	3
	5	36	18	17	63	2	3
	6	32	27	21	60	4	4
10×140	1	19	16	12	21	4	2
	2	21	19	17	39	4	4
	3	20	18	15	31	2	3
	4	22	16	10	19	2	2
	5	25	24	21	41	4	4
	6	27	26	24	49	2	5
10×160	1	17	17	17	51	4	5
	2	21	16	15	49	4	4
	3	30	27	26	68	7	7
	4	21	20	20	57	5	6
	5	19	12	10	40	3	3
	6	36	23	22	71	6	7
10×250	1	—	18	18	—	6	9
	2	—	25	19	—	9	10
	3	—	24	24	—	8	6
	4	—	26	19	—	8	9
	5	—	27	25	—	8	11
	6	—	26	21	—	9	10

ченные всеми алгоритмами, практически совпадали во всех задачах. Результаты экспериментов приведены в табл. 24 и 25. Прочерк в таблицах означает, что задача не решалась данным методом из-за большого расхода времени. Табл. 24 иллюстрирует работу алгоритмов с ростом m при фиксированном n . В табл. 25 приведены результаты экспериментов при разных значениях n . За-

$n \times m$	Номер задачи	Число итераций			Время решения, с		
		симп-лекс	§ 1 гл. 1	§ 2 гл. 2	симп-лекс	§ 1 гл. 1	§ 2 гл. 2
20×100	1	—	38	34	—	23	27
	2	—	41	37	—	25	31
	3	—	27	21	—	15	19
	4	—	34	29	—	17	23
	5	—	53	43	—	32	41
	6	—	44	39	—	28	35
30×100	1	95	53	51	312	30	32
	2	74	49	48	291	27	30
	3	91	59	56	344	31	35
	4	63	50	48	321	27	32
	5	83	57	54	363	31	38
	6	60	38	32	211	22	30
40×100	1	—	71	70	—	41	43
	2	—	83	78	—	49	54
	3	—	73	63	—	42	47
	4	—	80	75	—	40	49
	5	—	79	71	—	45	51
	6	—	90	83	—	71	80

метим, что алгоритм § 1 гл. 1 для больших значений n имеет преимущество перед алгоритмом § 2 гл. 2, так как в нем размерность опоры меняется от 0 до n и, следовательно, требует меньше времени для пересчета $A_{\text{оп}}^{-1}$.

§ 3. Декомпозиция опоры канонической задачи

Одним из возможных общих методов учета большого числа основных ограничений-равенств является расчленение соответствующей матрицы условий на блоки с малым количеством строк. Вводя для каждого блока свою опорную матрицу, можно операции адаптивного метода с опорной матрицей всей задачи заменить на операции с

малыми опорными матрицами блоков. При этом экономится оперативная память ЭВМ для хранения опорных матриц.

1. Мультиопорный план. Рассмотрим линейную задачу с основными ограничениями-равенствами

$$c'x \rightarrow \max, Ax=b, d_* \leq x \leq d^*, \quad (1)$$

где $A=A(I, J)$, $I=\{1, 2, \dots, m\}$, $J=\{1, 2, \dots, n\}$.

Пусть число m строк матрицы A таково, что применение алгоритмов с полным хранением обратной матрицы невозможно из-за недостатка оперативной памяти. Для таких задач одним из возможных методов решения является модификация адаптивного алгоритма, связанная с декомпозицией опоры. Из m_1 строк матрицы $A=A(I, J)$ составим матрицу $A_1=A(I_1, J)$, из остальных m_2 строк — матрицу $A_2=A(I_2, J)$. Предположим, что $\text{rank } A_1=m_1$, $\text{rank } A_2=m_2$, $m_1+m_2=m$. Множество опорных индексов $J_{\text{оп}}$ всей матрицы A разобьем на два подмножества: $J_{\text{оп}}=J_{1\text{оп}} \cup J_{2\text{оп}}$, $J_{1\text{оп}} \cap J_{2\text{оп}}=\emptyset$, $|J_{1\text{оп}}|=m_1$, $|J_{2\text{оп}}|=m_2$. Совокупность $Q_{\text{оп}}=\{Q_{1\text{оп}}, Q_{2\text{оп}}\}$, $Q_{1\text{оп}}=\{I_1, J_{1\text{оп}}\}$, $Q_{2\text{оп}}=\{I_2, J_{2\text{оп}}\}$, назовем мультиопорой задачи (1), если $\det A(I_1, J_{1\text{оп}}) \neq 0$, $\det \tilde{A}(I_2, J_{2\text{оп}}) \neq 0$, где

$$\tilde{A}(I_2, J) = A(I_2, J) - A(I_2, J_{1\text{оп}}) A^{-1}(J_{1\text{оп}}, I_1) A_1. \quad (2)$$

З а м е ч а н и е. Матрица $\tilde{A}(I_2, J)$ получается после подстановки в уравнение $A_2x=b(I_2)$ вектора $x(J_{1\text{оп}})$, найденного из уравнения $A_1x=b(I_1)$. На каждом шаге излагаемого ниже алгоритма при фиксированной опоре $Q_{1\text{оп}}$ и при $x(J_{1\text{оп}}) = A^{-1}(J_{1\text{оп}}, I_1)[b(I_1) - A(I_1, J_{1\text{н}})x(J_{1\text{н}})]$, $J_{1\text{н}}=J \setminus J_{1\text{оп}}$ решается задача (1). Полученное решение используется затем для построения подходящего направления в точке плана $x(J)$.

Пусть известны обратные матрицы $A_{11}^{-1}=A^{-1}(J_{1\text{оп}}, I_1)$, $\tilde{A}^{-1}(J_{2\text{оп}}, I_2)$. Тогда для опорной матрицы

$$A_{\text{оп}} = A(I, J_{\text{оп}}) = \begin{bmatrix} A(I_1, J_{1\text{оп}}) & A(I_1, J_{2\text{оп}}) \\ A(I_2, J_{1\text{оп}}) & A(I_2, J_{2\text{оп}}) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

обратная вычисляется по формуле

$$A_{\text{оп}}^{-1} = \begin{bmatrix} A_{11}^{-1} + A_{11}^{-1} A_{12} \tilde{A}_{22}^{-1} A_{21} A_{11}^{-1} & -A_{11}^{-1} A_{12} \tilde{A}_{22}^{-1} \\ -\tilde{A}_{22}^{-1} A_{21} A_{11}^{-1} & \tilde{A}_{22}^{-1} \end{bmatrix}, \quad (3)$$

где $\tilde{A}_{22} = \tilde{A}_{20п} = \tilde{A}(I_2, J_{20п})$. Пару $\{x, Q_{оп}\}$, в которой x — план задачи (1), будем называть мультиопорным планом.

Опишем операцию вычисления оценок $\Delta_j = c'(J_{оп}) \times \times A_{оп}^{-1} A(I, J) - c_j$ в случае, когда в памяти ЭВМ хранятся только матрицы $A_{1оп}^{-1}$ и $\tilde{A}_{2оп}^{-1}$. Сначала, учитывая формулу (3), вычислим вектор потенциалов $u' = (u'(I_1), u'(I_2)) = (c'(J_{1оп}), c'(J_{2оп})) A_{оп}^{-1} = (c'_1, c'_2) A_{оп}^{-1} = (c'_1 (A_{11}^{-1} + A_{11}^{-1} A_{12} \tilde{A}_{22}^{-1} A_{21} A_{11}^{-1}) - c'_1 \tilde{A}_{22}^{-1} A_{21} A_{11}^{-1}, -c'_1 A_{11}^{-1} A_{12} \tilde{A}_{22}^{-1} + c'_2 \tilde{A}_{22}^{-1})$. Далее заметим, что $\Delta' = u' A - c' = c'_1 (A_{11}^{-1} + A_{11}^{-1} A_{12} \tilde{A}_{22}^{-1} A_{21} A_{11}^{-1}) A_1 - c'_2 \tilde{A}_{22}^{-1} A_{21} A_{11}^{-1} A_1 - c'_1 A_{11}^{-1} A_{12} \times \times \tilde{A}_{22}^{-1} A_2 + c'_2 \tilde{A}_{22}^{-1} A_2 - c' = c'_1 A_{11}^{-1} A_1 - c'_1 A_{11}^{-1} A_{12} \tilde{A}_{22}^{-1} (A_2 - A_{21} A_{11}^{-1} A_1) + c'_2 \tilde{A}_{22}^{-1} (A_2 - A_{21} A_{11}^{-1} A_1) - c'$. Если обозначить $u'_1 = c'_1 A_{11}^{-1}$, $\Delta'_1 = u'_1 A_1 - c'_1$, $u'_2 = -\Delta'(J_{2оп}) \tilde{A}_{2оп}^{-1}$, то для вектора оценок получим формулу

$$\Delta' = \Delta'_1 + u'_2 \tilde{A}_2. \quad (4)$$

Зная вектор оценок Δ , вычислим оценку субоптимальности

$$\beta = - \sum_{\substack{\Delta_j > 0 \\ j \in J_H}} \Delta_j (d_{*j} - x_j) - \sum_{\substack{\Delta_j < 0, \\ j \in J_H}} \Delta_j (d_j^* - x_j). \quad (5)$$

Из § 1 гл. 1 следуют

Критерий оптимальности. Соотношения $\Delta_j \geq 0$ при $x_j = d_{*j}$; $\Delta_j \leq 0$ при $x_j = d_j^*$; $\Delta_j = 0$ при $d_{*j} < x_j < d_j^*$, $j \in J_H$, достаточны, а в случае невырожденности и необходимы для оптимальности мультиопорного плана $\{x, Q_{оп}\}$.

Критерий субоптимальности. Если

$$\beta \leq \varepsilon, \quad (6)$$

то мультиопорный план $\{x, Q_{оп}\}$ является ε -оптимальным: $c'x^0 - c'x \leq \varepsilon$, где x^0 — оптимальный план. Для каждого ε -оптимального плана x^ε найдется мультиопора $Q_{оп}$ такая, что для $\{x^\varepsilon, Q_{оп}\}$ выполняется неравенство (6).

2. Подходящее направление. Пусть мультиопорный план $\{x, Q_{оп}\}$ не удовлетворяет критерию (6). Подходящее направление для улучшения плана x найдем из решения следующей задачи:

$$\begin{aligned}
& -\Lambda'_1(J_{1H})x(J_{1H}) \rightarrow \max, \quad \tilde{A}(I_2, J_{1H})x(J_{1H}) = \\
& = \tilde{b}(I_2), \quad d_{*1H} \leq x_{1H} \leq d_{1H}^*,
\end{aligned} \tag{7}$$

где $\tilde{b}(I_2) = b(I_2) - A(I_2, J_{10H})A_{10H}^{-1}b(I_1)$, $d_{*1H} = d_*(J_{1H})$, $x_{1H} = x(J_{1H})$, $d_{1H}^* = d^*(J_{1H})$. Смысл этой задачи состоит в максимизации целевой функции $c'x$ по переменным $x(J_{1H})$ при фиксированной опоре Q_{10H} , т. е. при $x(J_{10H}) = A_{11}^{-1}(b(I_1) - A(I_1, J_{1H})x_{1H})$. В отличие от адаптивного метода, в котором подходящее направление вычисляется из условия максимума приращения целевой функции только по неопорным переменным и определяется однозначно, в данном случае для решения задачи (7) применяется алгоритм § 4 гл. 1.

Предположим, что для решения задачи (7) выполнено k итераций при $k \geq 1$ и найдены невырожденный β^k -оптимальный опорный план $\{x^k(J_{1H}), \tilde{A}_{20H}^{(k)}\}$, $\tilde{A}_{20H}^{(k)} = \tilde{A}(I_2, J_{20H}^{(k)})$, и $\Delta^{(k)}(J_{1H})$ — соответствующий ему вектор оценок. Заметим, что вектор оценок в задаче (7) вычисляется по формуле (4), поэтому имеем $\Delta(J_{1H}) = \Delta^{(0)}(J_{1H})$. Подходящим направлением для плана $x(J)$ будет вектор $l^*(J)$:

$$\begin{aligned}
l^*(J_{1H}) &= x^k(J_{1H}) - x(J_{1H}), \quad l^*(J_{10H}) = \\
&= -A_{10H}^{-1}A(I_1, J_{1H})l^*(J_{1H}).
\end{aligned} \tag{8}$$

3. Итерация. При решении задачи (7) были учтены все основные ограничения задачи (1) и прямые ограничения на переменные x_j , $j \in J_{1H}$. Поэтому максимально допустимый шаг вдоль $l^*(J_{1H})$ равен 1. Для переменных $x(J_{10H})$ этот шаг может оказаться недопустимым. Шаг вдоль $l(J_{10H})$ найдем по формулам $\Theta_{j_0} = \min \{\Theta_j, j \in J_{10H}\}$, $\Theta_j = (d_j^* - x_j)/l_j^*$, если $l_j^* > 0$; $\Theta_j = \infty$, если $l_j^* = 0$; $\Theta_j = (d_{*j} - x_j)/l_j^*$, если $l_j^* < 0$, $j \in J_{10H}$. Итак, максимально допустимый шаг вдоль $l^*(J)$

$$\Theta_0 = \min \{1, \Theta_{j_0}\}. \tag{9}$$

Если $\Theta_0 = 1$, то план $\bar{x} = x + l^*$ будет β^k -оптимальным планом задачи (1) и при $\beta^k \leq \varepsilon$ ее решение прекращается. В противном случае, начиная с плана $\{x^k(J_{1H}), \tilde{A}_{20H}^{(k)}\}$, продолжается решение задачи (7).

Если $\Theta_0 = \Theta_{j_0} < 1$, то строим новый план $\bar{x} = x + \Theta_0 l^*$. Для него оценка субоптимальности

$$\bar{\beta} = \sum_{\Delta_j^k > 0, j \in J_{1\text{н}}} \Delta_j^k (\bar{x}_j - d_{*j}) + \sum_{\Delta_j^k < 0, j \in J_{1\text{н}}} \Delta_j^k (\bar{x}_j - d_j^*).$$

При переходе от плана x к плану \bar{x} оценка субоптимальности уменьшается на величину $\Delta\beta = \Delta\beta_x + \Delta\beta_{\text{оп}}$, где $\Delta\beta_x = -\Theta_0 \Delta'(J_{\text{н}}) l^*(J_{\text{н}}) \geq 0$, $-\Delta\beta_{\text{оп}} \leq 0$ — приращение двойственной целевой функции. Если $\bar{\beta} \leq \varepsilon$, то $\bar{x} = x + \Theta_0 l^*$ — ε -оптимальный план задачи (1). В противном случае решение задачи (1) продолжается.

4. Преобразование мультипоры. Индекс j_0 критической переменной \bar{x}_{j_0} , препятствующей движению вдоль l^* , выводится из $J_{1\text{оп}}$. Вместо него вводится некоторый элемент из $J_{2\text{оп}}^{(k)}$ по следующим правилам.

Вычислим вектор $p' = p'(J_{2\text{оп}}^{(k)}) = e'_{j_0} A_{1\text{оп}}^{-1} A(I_1, A_{2\text{оп}}^{(k)})$. Рассмотрим два случая.

1) Существует элемент $p_{j_*} \neq 0$, $j_* \in J_{2\text{оп}}^{(k)}$. Тогда в множествах $J_{1\text{оп}}$ и $J_{2\text{оп}}^{(k)}$ поменяем местами элементы j_0 и j_* : $\bar{J}_{1\text{оп}} = (J_{1\text{оп}} \setminus j_0) \cup j_*$, $\bar{J}_{2\text{оп}} = (J_{2\text{оп}}^{(k)} \setminus j_*) \cup j_0$. В результате получим новую мультипору $\bar{Q}_{\text{оп}} = \{\bar{Q}_{1\text{оп}}, \bar{Q}_{2\text{оп}}\}$, $\bar{Q}_{1\text{оп}} = \{I_1, \bar{J}_{1\text{оп}}\}$, $\bar{Q}_{2\text{оп}} = \{I_2, \bar{J}_{2\text{оп}}\}$. Обратные матрицы $\bar{A}_{1\text{оп}}^{-1} = A^{-1}(\bar{J}_{1\text{оп}}, I_1)$ и $\bar{A}_{2\text{оп}}^{-1} = \tilde{A}^{-1}(\bar{J}_{2\text{оп}}, I_2)$ можно найти по формулам

$$\begin{aligned} \bar{A}_{1\text{оп}}^{-1} &= [E + (e_{j_0} - a_{j_*}) e'_{j_0} / a_{j_0 j_*}] A_{1\text{оп}}^{-1}, \\ \bar{A}_{2\text{оп}}^{-1} &= [E - e_{j_*} (p + e_{j_*})'] \tilde{A}_{2\text{оп}}^{-1}, \end{aligned} \quad (10)$$

где $a_{j_*} = A_{1\text{оп}}^{-1} A(I_1, j_*)$; $a_{j_0 j_*}$ — j_0 -й элемент вектора a_{j_*} .

2) $p(J_{2\text{оп}}) = 0$. Найдем отличный от нуля элемент $x_{j_0 i_0}$ в j_0 -й строке матрицы $A_{1\text{оп}}^{-1}$ и элемент $x_{i_* j_*} \neq 0$ в матрице $\tilde{A}(I_2, J_{2\text{оп}}^{(k)})$. Неособенность матриц $A_{1\text{оп}}$ и $\tilde{A}_{2\text{оп}}$ гарантирует существование таких элементов. Строим новую мультипору $\bar{Q}_{\text{оп}} = (\bar{Q}_{1\text{оп}}, \bar{Q}_{2\text{оп}})$, где $\bar{Q}_{1\text{оп}} = \{\bar{I}_1, \bar{J}_{1\text{оп}}\}$, $\bar{Q}_{2\text{оп}} = \{\bar{I}_2, \bar{J}_{2\text{оп}}\}$, $\bar{I}_1 = (I_1 \setminus i_0) \cup i_*$, $\bar{I}_2 = (I_2 \setminus i_*) \cup i_0$, $\bar{J}_{1\text{оп}} = (J_{1\text{оп}} \setminus j_0) \cup j_*$, $\bar{J}_{2\text{оп}} = (J_{2\text{оп}} \setminus j_*) \cup j_0$. Обратные матрицы $\bar{A}_{1\text{оп}}^{-1} = A^{-1}(\bar{J}_{1\text{оп}}, \bar{I}_1)$, $\bar{A}_{2\text{оп}}^{-1} = \tilde{A}^{-1}(\bar{J}_{2\text{оп}}, \bar{I}_2)$ можно найти с помощью старых обратных матриц, выполняя последовательно операции

исключения и добавления строки и столбца в каждой из них. Пересчет выполняется по формулам, приведенным в гл. 1.

Для новых опорных множеств вновь формулируем задачу (7). Начальный опорный план $\{\bar{x}(J_{1н}), \tilde{A}_{2оп}\}$ этой задачи является вырожденным, поскольку опорная переменная \bar{x}_{j_0} , $j_0 \in J_{2оп}$, критическая. Приведем алгоритм решения задачи (7), в котором на первой итерации начальный план строго улучшается.

1. Положим $\tilde{x}_j = x_j$, $j \in J_{1н}$; $it = 0$ — число итераций.

2. Положим $w_i = 0$, $i = \overline{1, m_2}$; $\beta = 0$. Вычислим $u' = \tilde{c}'(J_{2оп}) \tilde{A}_{2оп}^{-1}$. Пункты 3—6 выполним для $j \in J_{н}$.

3. Вычислим $p = \tilde{A}(I_2, j) = A(I_2, j) - A(I_2, J_{1оп}) \times \times A_{1оп}^{-1} A(I_1, j)$.

4. Найдем $\Delta_j = u'p - \tilde{c}_j$.

5. Построим $l_j = d_j^* - \tilde{x}_j$, если $\Delta_j < 0$; $l_j = d_{*j} - \tilde{x}_j$, если $\Delta_j > 0$; $l_j = 0$, если $\Delta_j = 0$; $\beta = \beta - \Delta_j l_j$.

6. В вектор w добавим слагаемое: $w = w + pl_j$.

7. Проверим критерий субоптимальности: $\beta \leq \varepsilon$. При выполнении этого неравенства переходим в п. 25.

8. Вычислим вектор $l(J_{2оп}) - \tilde{A}_{2оп}^{-1} w$.

9. Если $\tilde{x}_{j_0} = d_{*j_0}$ и $l_{j_0} < 0$ или $\tilde{x}_{j_0} = d_{j_0}^*$ и $l_{j_0} > 0$, то перейдем в п. 12.

10. Вычислим $\Theta_0 = \min \{1, \Theta_{j_0}\}$, $\Theta_{j_0} = \min \Theta_j$, $j \in J_{2оп}$, $\Theta_j = \infty$ при $l_j = 0$; $\Theta_j = (d_j^* - \tilde{x}_j)/l_j$ при $l_j > 0$; $\Theta_j = (d_{*j} - \tilde{x}_j)/l_j$ при $l_j < 0$.

11. Построим новый план $\tilde{x}(J_{1н}) = \tilde{x}(J_{1н}) + \Theta_0 l(J_{1н})$. Положим $it = it + 1$. При условии $(1 - \Theta_0)\beta \leq \varepsilon$ перейдем в п. 25.

12. Положим $k = -1$, если $\tilde{x}_{j_0} = d_{j_0}^*$; $k = 1$, если $\tilde{x}_{j_0} = d_{*j_0}$. Двойственный шаг σ и число $\gamma = -ke'_{j_0} \tilde{A}_{2оп}^{-1} \times \times \tilde{A}(I_2, J_{н}) l(J_{н})$ найдем, выполнив пп. 13—18 для $j \in J_{н}$ (полагая $\sigma = \infty$, $\gamma = 0$).

13. Как и в п. 3, вычислим вектор $p = \tilde{A}(I_2, j)$ по формуле (2).

14. Найдем $z_j = ke'_{j_0} \tilde{A}_{20п}^{-1} p$.

15. Вычислим $l_j = d_j^* - \tilde{x}_j$ при $\Delta_j < 0$; $l_j = d_{*j} - \tilde{x}_j$ при $\Delta_j > 0$; $l_j = 0$ при $\Delta_j = 0$.

16. Положим $\gamma = \gamma - z_j l_j$.

17. Если $\Delta_j = 0$ и $z_j (\tilde{x}_j - d_j^*) > 0$ или $\Delta_j = 0$ и $z_j (\tilde{x}_j - d_{*j}) > 0$, то положим $\sigma = 0$.

18. При $\Delta_j z_j < 0$ вычислим $\eta = -\Delta_j / z_j$. Если $\eta < \sigma$, то положим $\sigma = \eta$. Для $j \in J_n$ выполним операции 19—22 (чтобы найти индекс j_*).

19. Вычислим $\Delta_j = \Delta_j + \sigma z_j$.

20. Если $\Delta_j = 0$ и $d_{*j} < \tilde{x}_j < d_j^*$, то, полагая $j_* = j$, перейдем в п. 24.

21. Если $\Delta_j = 0$ и $\gamma = 0$, то, полагая $j_* = j$, перейдем в п. 24.

22. Если $\Delta_j = 0$ и $\gamma < 0$, то при $z_j > 0$ и $\tilde{x}_j = d_j^*$ либо при $z_j < 0$ и $\tilde{x}_j = d_{*j}$ полагаем $j_* = j$ и переходим в п. 24.

23. Если $\Delta_j = 0$ и $\gamma > 0$, то $j_* = j$.

24. Изменив опору $J_{20п} = (J_{20п} \setminus j_0) \cup j_*$, пересчитаем обратную матрицу $\tilde{A}_{20п}^{-1} = [E - (a_{j_*} - e_{j_0}) e'_{j_0}] \tilde{A}_{20п}^{-1}$, где $a_{j_*} = \tilde{A}_{20п}^{-1} \tilde{A}(I_2, j_*) / e'_{j_0} \tilde{A}_{20п}^{-1} \tilde{A}(I_2, j_*)$ — неединичный столбец левого мультипликатора. Переходим в п. 2.

25. Выход из алгоритма решения задачи (7).

З а м е ч а н и е. Вектор u и оценки Δ_j необходимо строить только на первой итерации, а в дальнейшем они будут пересчитаны в п. 13. Операции их вычисления в пп. 2, 4 следует опустить. В этом случае нужно положить $\Delta_{j_0} = \Delta_{j_0} + k\sigma$, так как на новой итерации $j_0 \in J_n$.

Рассмотрим теперь алгоритм решения задачи (1) для случая, когда задан мультиопорный план $\{x, Q_{оп}\}$, т. е. когда имеются план этой задачи x , матрицы $A_{1оп}^{-1}$, $\tilde{A}_{20п}^{-1}$ и множества I_1 , I_2 , $J_{10п}$, $J_{20п}$.

5. Алгоритм. 1. Сформируем задачу (7): $\tilde{c}' = u' A(I_1, J_{1н}) - c'(J_{1н})$, $u' = c'(J_{10п}) A_{1оп}^{-1}$, $\tilde{x}(J_{1н}) = x(J_{1н})$. Столбцы матрицы $\tilde{A}(I_2, J_{1н})$ по мере необходимости будут генерироваться по формуле (2).

2. Выполним заданное число итераций в задаче (7). Получим β -оптимальный план $\tilde{x}(J_{1н})$, вектор $\tilde{\Delta}(J_{1н})$ и матрицу $\tilde{A}_{2оп}^{-1}(I_2, \tilde{J}_{2оп})$.

3. По формулам (8) построим подходящее направление l .

4. Максимально допустимый шаг Θ_0 найдем по формулам (9).

5. Если $\Theta_0=1$ и $\beta > \varepsilon$, то перейдем в п. 2 для продолжения решения задачи (7).

6. Построим новый план $\bar{x} = x + \Theta_0 l$. Если $\Theta_0=1$ и $\beta \leq \varepsilon$, то план \bar{x} — ε -оптимальный план. Перейдем в п. 12.

7. Найдем оценку субоптимальности для плана \bar{x} (при $\Theta_0 < 1$):

$$\beta = \sum_{\substack{\tilde{\Delta}_j > 0, \\ j \in J_H}} \tilde{\Delta}_j (\bar{x}_j - d_{*j}) + \sum_{\substack{\tilde{\Delta}_j < 0, \\ j \in J_H}} \tilde{\Delta}_j (\bar{x}_j - d_j^+).$$

Если $\beta \leq \varepsilon$, то переходим в п. 12.

8. Вычислим вектор $p'(\tilde{J}_{2оп}) = e'_{j_0} A_{1оп}^{-1} A(I_1, \tilde{J}_{2оп})$ и найдем $p_{j*} = \max_{j \in J_{2оп}} |p_j|$. Если $p_{j*} = 0$, то перейдем в п. 10.

9. По формулам (10) пересчитаем матрицы $\tilde{A}_{2оп}^{-1}$ и $A_{1оп}^{-1}$ и, полагая $\bar{J}_{1оп} = (J_{1оп} \setminus j_0) \cup j_*$, $\bar{J}_{2оп} = (J_{2оп} \setminus j_*) \cup j_0$, перейдем в п. 1.

10. В j_0 -й строке матрицы $A_{1оп}^{-1}$ найдем максимальный по модулю элемент $x_{j_0 i_0}$. В матрице $\tilde{A}_{2оп}^{-1}$ найдем ненулевой элемент $x_{i_* j_*}$.

11. Положим $\bar{I}_1 = (I_1 \setminus i_0) \cup i_*$, $\bar{I}_2 = (I_2 \setminus i_*) \cup i_0$, $\bar{J}_{1оп} = (J_{1оп} \setminus j_0) \cup j_*$, $\bar{J}_{2оп} = (J_{2оп} \setminus j_*) \cup j_0$. Пересчитаем каждую из матриц $A_{1оп}^{-1}$ и $\tilde{A}_{2оп}^{-1}$ по формулам гл. 1 и перейдем в п. 1.

12. Выдача результатов.

6. Численные эксперименты. Алгоритм написан на языке АЛГОЛ-ГДР. Для него проведено несколько экспериментальных расчетов. Задачи генерировались процедурой с равномерным размещением ненулевых элементов по столбцу матрицы (GENER2 из § 1 гл. 1).

Эксперимент 1. Цель — проверить работу алгоритма для различных начальных приближений. Допустимое ре-

шение задавалось так: выбирался $x^1: x_j^1 \in [d_{*j}, d_j^*]$, $j = \overline{1, n}$, и вычислялся вектор правых частей $b = Ax^1$. При работе алгоритма m_1 итераций выполнялись с нулевым шагом — пока не закончится замена искусственных столбцов в опоре $A_{1\text{оп}}$. Затем шел процесс улучшения плана. В случае недопустимого начального приближения ($Ax^1 = b$) применялась стартовая процедура для расширения задачи и первые итерации алгоритма выполнялись с положительным шагом, чтобы сделать невязки $(b - Ax^1)$ равными нулю. В эксперименте 20×50 -матрица A имела по три ненулевых элемента в столбце, размещенных равномерно, но со сдвигом на одну строку (чтобы не было ненулевых строк в матрице). Векторы $c \in [-100, 100]$, $d_* \in [-100, 0]$ и $d^* \in [0, 100]$ генерировались полностью (все n элементов) с помощью процедуры ВЕК (см. § 1 гл. 1). Результаты эксперимента приведены в табл. 26.

Эксперимент 2. Цель — найти оптимальное с точки зрения скорости решения задачи значение параметра декомпозиции при заданном объеме памяти. Генерирование задач выполнялось, как и в первом эксперименте. Вектор b и начальное приближение x^1 задавались так, чтобы $Ax^1 = b$. Результаты эксперимента для 20×40 -задачи, решенной с разным числом внутренних итераций и для различных значений параметра декомпозиции m_1 , приведены в табл. 27. Из таблицы видно, что время решения задачи существенно увеличивается с ростом m_1 . Пусть объем допустимой памяти (для $A_{\text{оп}}^{-1}$) равен K . Тогда оптимальное значение параметра m_1 найдется из равенства

$$m_1^2 + (m - m_1)^2 = K, \quad (12)$$

где m_1 и $(m - m_1) = m_2$ — размеры $A_{1\text{оп}}$ и $\tilde{A}_{2\text{оп}}$. Наименьший объем памяти, необходимый для запоминания $A_{1\text{оп}}^{-1}$ и $\tilde{A}_{2\text{оп}}^{-1}$, очевидно, равен $m^2/2$. Следовательно, при наличии свободной памяти, когда $m^2/2 < K < m^2$, имеет смысл вычислять оптимальное значение параметра m_1 , при котором для решения задачи потребуется наименьшее время. При $K \geq m^2$ следует применять адаптивный алгоритм без декомпозиции (см. § 4 гл. 1).

7. Двухшаговый метод с декомпозицией опоры. Построим метод решения задачи (1), в котором сочетается идея двухшагового метода с методом декомпозиции опо-

Начальное приближение	$x^1=d_*,$ $b=0,$ $Ax^1 \neq b$	$x^1=(d_*+d^*)/2,$ $b=0,$ $Ax^1 \neq b$	$x^1=(d_*+d^*)/2,$ $b=A(d_*+d^*)/2,$ $Ax^1=b$	
m_1	5	5	5	10
Число внутренних итераций	1 2	1 2	1 2 3 4	1 2 3 4
Число внешних итераций	24 22	10 15	10 10 15 13	18 18 15 15
Время решения задачи, с	141 147	125 141	100 95 127 116	168 125 125 128

Примечание: В таблице указано только задаваемое число внутренних итераций, но при $\Theta = 1$ и $\beta > \varepsilon$ внутренние итерации повторялись вновь т. е. их было больше.

Табл. 27,

m_1	5	10	15
Число внутренних итераций	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
Число внешних итераций	11 9 12 9 11 8 8	16 14 22 17 16 14 14	29 23 23 25 28 24 24
Время решения задачи, с	78 48 70 59 76 51 66	109 93 136 124 139 127 158	184 95 138 169 255 234 225

ры. Разобьем матрицу A на блоки $A_i, i=\overline{1, k}$. Для упрощения дальнейшего изложения будем, как и в п. 1, считать, что $k=2$, т. е. пусть матрица имеет вид

$$A = A(I, J) = \begin{Bmatrix} A(I_1, J) \\ A(I_2, J) \end{Bmatrix}, \quad I = I_1 \cup I_2, \quad I_1 \cap I_2 = \emptyset.$$

Понятие мультипоры определяется, как и в п. 1. Однако в процессе решения задачи (1) описанным в указанном пункте методом множества I_1 и I_2 не меняются (см. также [17] *)), поэтому для обозначения мультипоры бу-

*) В [17] это достигается тем, что в случае $p(J_{20п})=0$ элемент $j_0 \in J_{10п}$ сразу выводится из множества J_n по правилам адаптивного метода. Множества I_1, I_2 при этом не меняются.

дем вместо $Q_{\text{оп}} = \{Q_{1\text{оп}}, Q_{2\text{оп}}\}$ использовать символ $J_{\text{оп}} = \{J_{1\text{оп}}, J_{2\text{оп}}\}$.

Пусть задан начальный мультиопорный план $\{x, J_{\text{оп}}\}$. Вычислим векторы $u'_1 = c' (J_{1\text{оп}}) A_{1\text{оп}}^{-1}$, $\Delta'_1 = u'_1 A (I_1, J) - c'$, $u'_2 = -\Delta'_1 (J_{2\text{оп}}) \tilde{A}_{2\text{оп}}^{-1}$, $\Delta' = u'_2 \tilde{A} (I_2, J) + \Delta'_1$. В п. 1 показано, что вычисленный таким образом вектор Δ совпадает с вектором оценок, найденным по адаптивному методу для опорного плана $\{x, A(I, J_{\text{оп}})\}$. Это позволяет легко вывести критерий оптимальности (субоптимальности) для мультиопорного плана $\{x, J_{\text{оп}}\}$.

Рассмотрим случай, когда мультиопорный план $\{x, J_{\text{оп}}\}$ нужно улучшить. Чтобы найти подходящее направление для $\{x, J_{\text{оп}}\}$, прямым методом при начальном опорном плане $\{0, \tilde{A}_{2\text{оп}}\}$ решим задачу

$$-\Delta'_1 (J_{1\text{н}}) l (J_{1\text{н}}) \rightarrow \max, \tilde{A} (I_2, J_{1\text{н}}) l_{1\text{н}} = 0, \quad (10)$$

$$d_{*1\text{н}} - x_{1\text{н}} \leq l_{1\text{н}} \leq d_{1\text{н}}^* - x_{1\text{н}},$$

где $J_{1\text{н}} = J \setminus J_{1\text{оп}}$, $l_{1\text{н}} = l (J_{1\text{н}})$ и т. д.

Эту задачу необязательно решать до конца. Пусть на k -й итерации получены опорный план $\{l^* (J_{1\text{н}}), J_{2\text{оп}}^*\}$ и соответствующее ему подходящее направление $l^k (J_{1\text{н}})$. Тогда подходящим направлением для $\{x, J_{1\text{оп}} \cup J_{2\text{оп}}^*\}$ будет вектор $l_1 = l_1 (J) : l_1 (J_{1\text{н}}) = l^* (J_{1\text{н}}) + l^k (J_{1\text{н}})$, $l_1 (J_{1\text{оп}}) = -A_{1\text{оп}}^{-1} A (I_1, J_{1\text{н}}) l_1 (J_{1\text{н}})$.

Процесс решения задачи (1) прекращается на ε -оптимальном плане $\tilde{x} = x + \Theta_1 l_1$, если для него выполняется критерий субоптимальности, где $\Theta_1 = \min \{1, \Theta_{j_0}\}$, $\Theta_{j_0} = \min \{\Theta_j, j \in J_{1\text{оп}} \cup J_{2\text{оп}}^*\}$, Θ_j — максимальный шаг, допускаемый j -м прямым ограничением задачи (1). В противном случае для плана $x + \Theta_1 l_1$ строим [17] новую мультиопору $\tilde{J}_{\text{оп}} = \{\tilde{J}_{1\text{оп}}, \tilde{J}_{2\text{оп}}\}$. По аналогии с вышеизложенным найдем новое подходящее направление l_2 , соответствующее $\{\tilde{x}, \tilde{J}_{\text{оп}}\}$. В результате получим и новое множество $\tilde{J}_{\text{оп}}^* = \tilde{J}_{1\text{оп}}^* \cup \tilde{J}_{2\text{оп}}^* = \tilde{J}_{1\text{оп}} \cup \tilde{J}_{2\text{оп}}^*$. Новый план задачи (1) вычислим по формуле $\bar{x} = x(z^0)$, где $x(z) = x + Lz$, $L = \{l_1, l_2\}$, $z = \{z_1, z_2\}$ — план, z^0 — решение задачи

$$\gamma' z \rightarrow \max, d_* - x \leq Lz \leq d^* - x, L = \{l_1, l_2\}. \quad (11)$$

Займемся решением задачи (11). Если $l_{2j_0} = 0$, то процесс решения задачи (11) аналогичен соответствующему из § 2. Пусть $l_{2j_0} \neq 0$. Найдем для l_2 максимально допустимый шаг $\Theta_2 = \min \{1, \Theta_{j_1}\}$, $\Theta_{j_1} = \min \{\Theta_j, j \in \tilde{J}_{\text{оп}}^*\}$. Здесь Θ_j вычисляется традиционным образом. При $\Theta_2 = 1$ план $\tilde{x} + l_2$ является оптимальным в задаче (1). Пусть $\Theta_2 < 1$ и для плана $\tilde{x} + \Theta_2 l_2$ критерий субоптимальности не выполняется. Задачу (11) решаем, начиная с опорного плана $\{z, L(z, j_1)\} = \{\Theta_1, \Theta_2, l_{2j_1}\}$.

Различаются два случая:

а) $\gamma_1 - \gamma_2 l_{1j_1} / l_{2j_1} = 0$, причем $\{\Theta_1, \Theta_2\}$ — оптимальный план задачи (11);

б) $\gamma_1 - \gamma_2 l_{1j_1} / l_{2j_1} \neq 0$. Пусть $\mu = \text{sign}(\gamma_1 - \gamma_2 l_{1j_1} / l_{2j_1})$. Положим $h=1$, $z^1 = \{\Theta_1, \Theta_2\}$. Опишем h -ю итерацию решения задачи (11). Известен опорный план $\{z^h, L(2, j_h)\}$. Построим для z^h подходящее направление $p^h = \{p_1^h = \mu, p_2^h = -\mu l_{1j_h} / l_{2j_h}\}$ и найдем новый план $z^{h+1} = z^h + \omega p^h$, где $\omega = \omega_{j_{h+1}}$, j_{h+1} — индекс ограничения задачи (11), первым ставшего активным при движении вдоль p^h . План z^{h+1} является оптимальным в задаче (11), если выполняется одно из условий:

$$l_{2j_{h+1}} = 0; \mu l_{1j_{h+1}} / l_{2j_{h+1}} \leq \mu l_{1j_h} / l_{2j_h}; \quad (12)$$

$$\mu (\gamma_1 - \gamma_2 l_{1j_{h+1}} / l_{2j_{h+1}}) \leq 0.$$

Если ни одно из условий не выполняется, то, приписав плану z^{h+1} опору $L(2, j_{h+1})$, продолжим решение задачи (11).

Пусть $J^* = \{j \in \tilde{J}_{\text{оп}}^* : \kappa_j = x_j + \Theta_1 l_{1j} + l_{2j} \in [d_{*j}, d_j^*]\}$. Имеет место утверждение: если z^{h+1} не является оптимальным планом задачи (11) (что означает нарушение всех условий (12)), то $j_{h+1} \in J^*$. Более того, при $x_{j_{h+1}}(z^{h+1}) = d_{j_{h+1}}^*$ выполняются неравенства $\kappa_{j_{h+1}} > d_{j_{h+1}}^*$, $l_{2j_{h+1}} > 0$, а при $\kappa_{j_{h+1}}(z^{h+1}) = d_{*j_{h+1}}$ — неравенства $\kappa_{j_{h+1}} < d_{*j_{h+1}}$, $l_{2j_{h+1}} < 0$.

С л е д с т в и е. Оптимальный план задачи (11) найдется не более чем за $|J^*| + 1$ итерацию.

Доказательство утверждения аналогично соответствующему из § 9 гл. 1. Отметим также, что все замечания,

относящиеся к п. 1 § 9 гл. 1, имеют место и в этом случае.

Пусть процесс решения задачи (11) остановлен на плане z^{h+1} . Компоненты $\bar{x}_{j_h}, \bar{x}_{j_{h+1}}$ нового плана $\bar{x} = x(z^{h+1})$

критические. Согласно утверждению имеем $j_h \in \tilde{J}_{оп}^*$. Для j_{h+1} возможны случаи: $j_{h+1} \in \tilde{J}_{оп}^*$, $j_{h+1} \in \tilde{J}_{оп}^*$, причем последний возможен лишь тогда, когда z^{h+1} является оптимальным планом задачи (11). Но каждый раз $I^* = \tilde{J}_{оп}^* \cap \{j_h, j_{h+1}\} \neq \emptyset$. Новая мультиопора строится следующим образом^{*)}. При $|I^*| = 1$ новая мультиопора получается аналогично [17]. Пусть $|I^*| = 2$.

1) $I^* \cap \tilde{J}_{2оп}^* = I^*$. Рассмотрим задачу

$$-\Delta^*(\tilde{J}_н^*) p(\tilde{J}_н^*) \rightarrow \max,$$

$$d_{*i} - \bar{x}_i \leq - \sum_{j \in \tilde{J}_н^*} \tilde{X}_{ij} p_j \leq d_i^* - \bar{x}_i, \quad i \in I^*, \quad (13)$$

$$d_{*j} - \bar{x}_j \leq p_j \leq d_j^* - \bar{x}_j, \quad j \in \tilde{J}_н^* = J \setminus \tilde{J}_{оп}^*,$$

где $\tilde{X}(\tilde{J}_{2оп}^*, \tilde{J}_н^*) = \tilde{A}_{2оп}^{-1} \tilde{A}(I_2, \tilde{J}_н^*)$, Δ^* — вектор оценок, пересчитанный по формулам (4) для мультиопоры $\tilde{J}_{оп}^*$ (он получается в процессе нахождения направления l_2). Пусть $\bar{I}_{оп}^* \subset I^*$, $\bar{J}_{ноп}^* \subset \tilde{J}_н^*$ — оптимальные опорные множества задачи (13). Положим $\bar{J}_{1оп} = \bar{J}_{1оп}^*$, $\bar{J}_{2оп} = (\bar{J}_{2оп}^* \setminus \bar{I}_{оп}^*) \cup \bar{J}_{ноп}^*$. Новая итерация метода начинается с мультиопорного плана $\{\bar{x}, \bar{J}_{оп}\} = \{\bar{x}, \bar{J}_{1оп} \cup \bar{J}_{2оп}\}$. При этом справедливы соотношения

$$\bar{A}_{1оп}^{-1} = A_{1оп}^{-1},$$

$$\bar{A}_{2оп}^{-1}(\bar{J}_{ноп}^*, I_2) = \tilde{X}^{-1}(\bar{J}_{ноп}^*, \bar{I}_{оп}^*) \tilde{A}_{2оп}^{-1}(\bar{I}_{оп}^*, I_2),$$

$$\begin{aligned} \bar{A}_{2оп}^{-1}(\bar{J}_{2оп} \setminus \bar{J}_{ноп}^*, I_2) &= \tilde{A}_{2оп}^{-1}(\bar{J}_{2оп} \setminus \bar{J}_{ноп}^*, I_2) - \\ &- \tilde{X}(\bar{J}_{2оп} \setminus \bar{J}_{ноп}^*, \bar{J}_{ноп}^*) \bar{A}_{2оп}^{-1}(\bar{J}_{ноп}^*, I_2). \end{aligned} \quad (14)$$

^{*)} В дальнейшем с целью упрощения записей для обозначения матриц $[A(I_1, \tilde{J}_{1оп}^*)]^{-1}$, $\{A(I_2, \tilde{J}_{2оп}^*) - A(I_2, \tilde{J}_{1оп}^*)[A(I_1, \tilde{J}_{1оп}^*)]^{-1}A(I_1, \tilde{J}_{2оп}^*)\}^{-1}$ будем использовать символы $A_{1оп}^{-1}$ и $\tilde{A}_{2оп}^{-1}$.

2) $I^* \cap \tilde{J}_{1\text{оп}}^* = j_0 \in \{j_h, j_{h+1}\}$. Тогда $I^* \setminus j_0 \in \tilde{J}_{2\text{оп}}^*$. Элемент j_0 выводим из $\tilde{J}_{1\text{оп}}^*$ по правилам [17]. Далее поступаем так, как в случае 1), при этом вместо I^* в задаче (13) рассматриваем множество $I^* \cap \tilde{J}_{2\text{оп}}^*$. (Пересечение этих множеств берется после того, как индекс j_0 уже выбыл из $\tilde{J}_{2\text{оп}}^*$.)

3) $I^* \cap \tilde{J}_{1\text{оп}}^* = I^*$. Обозначим: $X(\tilde{J}_{1\text{оп}}^*, J) = A_{1\text{оп}}^{-1} A(I_1, J)$. Различаются подслучаи:

3а) Матрица $X(I^*, \tilde{J}_{2\text{оп}}^*)$ содержит неособую подматрицу $X(I^*, J^*)$, $J^* \subset \tilde{J}_{2\text{оп}}^*$, $|J^*| = |I^*|$. Пусть $B = X^{-1}(J^*, I^*)$. В множествах $\tilde{J}_{1\text{оп}}^*$ и $\tilde{J}_{2\text{оп}}^*$ поменяем местами I^* и J^* , т. е. $\tilde{J}_{1\text{оп}}^n = (\tilde{J}_{1\text{оп}}^* \setminus I^*) \cup J^*$, $\tilde{J}_{2\text{оп}}^n = (\tilde{J}_{2\text{оп}}^* \setminus J^*) \cup I^*$. Возвращаемся к случаю 1). Имеют место соотношения

$$A_{1\text{оп}}^{-1}(J^*, I_1) = B A_{1\text{оп}}^{-1}(I^*, I_1),$$

$$\begin{aligned} A_{1\text{оп}}^{-1}(\tilde{J}_{1\text{оп}}^n \setminus J^*, I_1) &= A_{1\text{оп}}^{-1}(\tilde{J}_{1\text{оп}}^n \setminus J^*, I_1) - \\ &- X(\tilde{J}_{1\text{оп}}^n \setminus J^*, J^*) B A_{1\text{оп}}^{-1}(I^*, I_1); \end{aligned} \quad (14')$$

$$\tilde{A}_{2\text{оп}}^{-1}(\tilde{J}_{2\text{оп}}^n \setminus I^*, I_2) = \tilde{A}_{2\text{оп}}^{-1}(\tilde{J}_{2\text{оп}}^n \setminus I^*, I_2),$$

$$\begin{aligned} \tilde{A}_{2\text{оп}}^{-1}(I^*, I_2) &= -X(I^*, J^*) \tilde{A}_{2\text{оп}}^{-1}(J^*, I_2) - \\ &- X(I^*, \tilde{J}_{2\text{оп}}^* \setminus I^*) \tilde{A}_{2\text{оп}}^{-1}(\tilde{J}_{2\text{оп}}^* \setminus I^*, I_2), \end{aligned} \quad (15)$$

где $\tilde{A}_{2\text{оп}}^{-1} = [A(I_1, \tilde{J}_{1\text{оп}}^n)]^{-1}$, $\tilde{A}_{2\text{оп}}^{-1} = [A(I_2, \tilde{J}_{2\text{оп}}^n) - A(I_2, \tilde{J}_{1\text{оп}}^n) A_{1\text{оп}}^{-1} A(I_1, \tilde{J}_{2\text{оп}}^n)]^{-1}$.

3б) Матрица $X(I^*, \tilde{J}_{2\text{оп}}^*)$ нулевая. Рассмотрим задачу

$$-\Delta^*(\tilde{J}_{\text{н}}^*)' l(\tilde{J}_{\text{н}}^*) \rightarrow \max,$$

$$d_*(I^*) - \bar{x}(I^*) \leq -X(I^*, \tilde{J}_n^*) l(\tilde{J}_n^*) \leq d^*(I^*) - \bar{x}(I^*),$$

$$d_*(\tilde{J}_n^*) - \bar{x}(\tilde{J}_n^*) \leq l(\tilde{J}_n^*) \leq d^*(\tilde{J}_n^*) - \bar{x}(\tilde{J}_n^*). \quad (16)$$

Пусть $\bar{I}_{\text{оп}}^* \subset I^*$, $\bar{J}_{\text{ноп}}^* \subset \tilde{J}_n^*$ — оптимальные опорные множества задачи (16). Положим $\bar{J}_{1\text{оп}} = (\tilde{J}_{1\text{оп}}^* \setminus \bar{I}_{\text{оп}}^*) \cup \bar{J}_{\text{ноп}}^*$, $\bar{J}_{2\text{оп}} = \tilde{J}_{2\text{оп}}^*$. Матрицу $\tilde{A}(I_2, J)$ в этом случае не меняем. Новую обратную матрицу $\bar{A}_{1\text{оп}}^{-1}$ пересчитаем по формулам, аналогичным (14). Новую итерацию метода начинаем с мультиопорного плана $\{\bar{x}, \bar{J}_{1\text{оп}} \cup \bar{J}_{2\text{оп}}\}$.

Зв) Неособая подматрица $X(I^*, J^*)$, $J^* \subset \tilde{J}_{2\text{оп}}^*$, не существует, но матрица $X(I^*, \tilde{J}_{2\text{оп}}^*)$ не является нулевой. Пусть $X(i_0, j_0) \neq 0$, $i_0 \in I^*$, $j_0 \in \tilde{J}_{2\text{оп}}^*$. Поменяем в множествах $\tilde{J}_{1\text{оп}}^*$ и $\tilde{J}_{2\text{оп}}^*$ местами элементы i_0 и j_0 , т. е. положим $\tilde{J}_{1\text{оп}}^n = (\tilde{J}_{1\text{оп}}^* \setminus i_0) \cup j_0$, $\tilde{J}_{2\text{оп}}^n = (\tilde{J}_{2\text{оп}}^* \setminus j_0) \cup i_0$. Соответствующие новые обратные матрицы $\bar{A}_{1\text{оп}}^{-1}$, $\bar{A}_{2\text{оп}}^{-1}$ (смысл обозначений объяснен выше) пересчитаем по формулам, аналогичным (14'), (15). Вернемся к случаю 2).

З а м е ч а н и е. Все операции, связанные с построением новой мультиопоры $\bar{J}_{\text{оп}}$, описаны выше при умолчании, что среди компонент \bar{x}_j , $j \in \tilde{J}_{\text{оп}}^* \setminus I^*$, нет критических. Однако, следуя п. 1, нетрудно показать, что и в противном случае можно добиться строгого улучшения плана \bar{x} на следующей итерации.

§ 4. Адаптивный метод с декомпозицией опоры

Для интервальных задач линейного программирования, в которых основные ограничения записываются с помощью двухсторонних неравенств, метод декомпозиции опоры (см. § 3) в сочетании с адаптивным методом (см. § 1 гл. 1) приводит к новому методу с «плавающими» опорами, размеры которых в процессе итераций изменяются, но не превосходят заданных пределов. Это позволяет экономить оперативную память, предназначенную для хранения информации об обратной опорной матрице.

1. Постановка задачи. Рассмотрим линейную задачу с основными ограничениями-неравенствами

$$c'x \rightarrow \max, \quad b_* \leq Ax \leq b^*, \quad d_* \leq x \leq d^*, \quad (1)$$

где c, x, d_*, d^* — n -векторы; b_*, b^* — m -векторы; A — $m \times n$ -матрица.

В гл. 1 изложены различные варианты адаптивного алгоритма решения задачи (1). Здесь, как и в § 3, будем рассматривать случай, когда размеры опорной матрицы $A_{\text{оп}}$ превышают заданные пределы, обусловленные объемом оперативной памяти ЭВМ.

Пусть m_1 — заданный предел для размера одной опоры. С момента достижения опорой размера m_1 дальнейшее накопление опоры на итерациях адаптивного алгоритма будем вести с помощью двух рабочих матриц

$A_{1\text{оп}}$ и $\tilde{A}_{2\text{оп}}$, связанных соотношением (2) § 3. Размеры матрицы $\tilde{A}_{2\text{оп}}$ при работе алгоритма могут увеличиваться или уменьшаться до нуля, а размеры матрицы $A_{1\text{оп}}$ при этом могут только уменьшаться.

Приведем вычислительную схему адаптивного алгоритма для задачи (1), в котором обратная опорная матрица представлена через рабочие матрицы по формуле (3) § 3. Стандартные операции адаптивного алгоритма без декомпозиции (см. § 1 гл. 1) являются фрагментами данного алгоритма. Поэтому в вычислительной схеме предусматривается выполнение первых итераций (при $|J_{\text{оп}}| \leq m_1$) с отключением некоторых блоков. После достижения опорой размеров m_1 подключаются все блоки алгоритма. Столбцы и строки матрицы $\tilde{A}(I_2, J)$ генерируются по мере необходимости с помощью формулы (2) § 3 и матриц $A(I, J)$, $A_{1\text{оп}}^{-1}$.

2. Алгоритм. Пусть задан начальный опорный план $\{x, A_{1\text{оп}}\}$, $A_{1\text{оп}} = A(I_{1\text{оп}}, J_{1\text{оп}})$, $I_{1\text{оп}} \subset I$, $J_{1\text{оп}} \subset J$, $|I_{1\text{оп}}| = |J_{1\text{оп}}| = r \leq m_1$. Если задан только план x , а опора $A_{\text{оп}}$ неизвестна, то полагаем $I_{1\text{оп}} = \emptyset$, $J_{1\text{оп}} = \emptyset$, $it = r = 0$. В случае отсутствия плана предварительно применяется стартовая процедура, изложенная в § 6 гл. 1.

1. Вычислим $u'(I_{1\text{оп}}) = c'(J_{1\text{оп}}) A_{1\text{оп}}^{-1}$. Если $r \leq m_1$, то перейдем к п. 6, иначе — к п. 2.

2. Построим вектор $p'(J_{2\text{оп}}) = c'(J_{2\text{оп}}) - u'(J_{1\text{оп}}) A(I_{1\text{оп}}, J_{2\text{оп}})$.

3. Вычислим $u'(I_{2\text{оп}}) = p'(J_{2\text{оп}}) \tilde{A}_{2\text{оп}}^{-1}$.

4. Найдем $p'(J_{1\text{оп}}) = u'(I_{2\text{оп}})A(I_{2\text{оп}}, J_{1\text{оп}})$.

5. Вычислим $u'(I_{1\text{оп}}) = u'(I_{1\text{оп}}) - p'(J_{1\text{оп}})A_{1\text{оп}}^{-1}$.

6. Положим $\beta = 0$; $v(I) = w(I) = 0$. Следующие пп. 7, 8 выполним для $j \in J$.

7. Положим $w(I) = w(I) + A(I, j)x_j$.

8. Если $j \in J_{\text{н}}$, то вычислим $\Delta_j = u'(I_{\text{оп}})A(I_{\text{оп}}, j) - c_j$, $l_j = (d_j^* - x_j)$ при $\Delta_j < 0$, $l_j = (d_{j*} - x_j)$ при $\Delta_j > 0$, $l_j = 0$ при $\Delta_j = 0$, положим $\beta = \beta - \Delta_j l_j$ и $v(I) = v(I) + A(I, j)l_j$.

9. Вычислим $\beta = \beta + u_i R_1$, $R_1 = b_i^* - w_i$ при $u_i > 0$, $R_1 = b_{i*} - w_i$ при $u_i < 0$, $R_1 = 0$ при $u_i = 0$, $i \in I_{\text{оп}}$.

10. Проверим неравенство $\beta \leq \varepsilon$. Да \rightarrow п. 66, нет \rightarrow п. 11.

11. Вычислим $l(J_{1\text{оп}}) = A_{1\text{оп}}^{-1}[\omega(I_{1\text{оп}}) - v(I_{1\text{оп}})]$, причем компоненты ω_i вычисляются, как и число R_1 в п. 9 без запоминания. Если $r \leq m_1$, то перейдем к п. 16, иначе — к п. 12.

12. Вычислим $v(I_{2\text{оп}}) = v(I_{2\text{оп}}) + A(I_{2\text{оп}}, J_{1\text{оп}})l(J_{1\text{оп}})$.

13. Вычислим $l(J_{2\text{оп}}) = -\tilde{A}_{2\text{оп}}^{-1}[v(I_{2\text{оп}}) - \omega(I_{2\text{оп}})]$, где вместо вектора ω в вычислении участвуют лишь его компоненты, определяемые аналогично числу R_1 в п. 9.

14. Вычислим $v(I) = v(I) + A(I, J_{2\text{оп}})l(J_{2\text{оп}})$.

15. Вектор $l(J_{1\text{оп}}) = A_{1\text{оп}}^{-1}[\omega(I_{1\text{оп}}) - v(I_{1\text{оп}})]$ строится так же, как и в п. 11.

16. Вычислим $v(I_{\text{н}}) = v(I_{\text{н}}) + A(I_{\text{н}}, J_{1\text{оп}})l(J_{1\text{оп}})$.

17. Найдем

$$\Theta_{j_0} = \min_{j \in J_{\text{оп}}} \Theta_j, \quad \Theta_j = \begin{cases} (d_j^* - x_j)/l_j & \text{при } l_j > 0, \\ (d_{*j} - x_j)/l_j & \text{при } l_j < 0, \\ +\infty & \text{при } l_j = 0. \end{cases}$$

18. Найдем

$$\Theta_{i_0} = \min_{i \in I_{\text{н}}} \Theta_i, \quad \Theta_i = \begin{cases} (b_i^* - w_i)/v_i & \text{при } v_i > 0, \\ (b_{*i} - w_i)/v_i & \text{при } v_i < 0, \\ +\infty & \text{при } v_i = 0. \end{cases}$$

19. Найдем $\Theta = \min \{1, \Theta_{j_0}, \Theta_{i_0}\}$.

20. Если $\Theta = \Theta_{j_0}$, то при $l_{j_0} > 0$ полагаем $k = -1$, иначе — $k = 1$. Если $\Theta = \Theta_{i_0}$, то при $v_{i_0} < 0$ полагаем $k = -1$, иначе — $k = 1$.

21. Если $\Theta = \Theta_{i_0}$, то перейдем к п. 46. Если $r \leq m_1$, то перейдем к п. 40. Если $j_0 \in J_{2\text{оп}}$, то перейдем к п. 25.

22. При $j_0 \in J_{10\text{П}}$ вычислим $p(J_{20\text{П}}) = A_{10\text{П}}^{-1}(j_0, I_{10\text{П}}) A(I_{0\text{П}}, J_{20\text{П}})$ и найдем $|p_{j_*}| = \max_{j \in J_{20\text{П}}} |p_j|$. Если $|p_{j_*}| = 0$, то перейдем к п. 40.

23. Вычислим $v'(I_{20\text{П}}) = -p'(J_{20\text{П}}) \tilde{A}_{20\text{П}}^{-1}$.

24. Положим $\tilde{A}_{20\text{П}}^{-1}(j_*, I_{20\text{П}}) = v(I_{20\text{П}})$, $J_{10\text{П}} = (J_{10\text{П}} \setminus j_0) \cup j_*$, $J_{20\text{П}} = (J_{20\text{П}} \setminus j_*) \cup j_0$. Матрицу $A_{10\text{П}}^{-1}$ пересчитаем по формуле (10) § 3. Меняем местами компоненты l_{j_0} и l_{j_*} в векторе $l(J_{0\text{П}})$. Выполним операции пп. 25 — 30 для $j \in I_{20\text{П}} \cup I_{10\text{П}} \cup J_{\text{Н}}$, полагая $\sigma = +\infty$.

25. Если $j \in I_{10\text{П}}$, то полагаем $p(I_{20\text{П}}) = A(I_{20\text{П}}, J_{10\text{П}}) A_{10\text{П}}^{-1}(J_{10\text{П}}, j)$. Если $j \in J_{\text{Н}}$, то $-p(I_{20\text{П}}) = A(I_{20\text{П}}, j) - A(I_{20\text{П}}, J_{10\text{П}}) A_{10\text{П}}^{-1} A(I_{10\text{П}}, j)$.

26. Если $j \in I_{20\text{П}}$, то полагаем $R_1 = k \tilde{A}_{20\text{П}}^{-1}(j_0, j)$, иначе полагаем $R_1 = k \tilde{A}_{20\text{П}}^{-1}(j_0, I_{20\text{П}}) p(I_{20\text{П}})$. Если $j \in I_{10\text{П}}$, то полагаем $R_1 = -R_1$.

27. Положим $t = u_j$, если $j \in I_{0\text{П}}$; $t = u'(I_{0\text{П}}) A(I_{0\text{П}}, j) - c_j$, если $j \in J_{\text{Н}}$.

28. Пусть $t = 0$. Если $j \in I_{0\text{П}}$, то при $R_1 > 0$ и $\omega_j \neq b_j^*$ или при $R_1 < 0$ и $\omega_j \neq b_{*j}$ полагаем $\alpha = kR_1$, $\sigma = 0$, $i_* = j$. Если $j \in J_{\text{Н}}$, то при $R_1 > 0$ и $x_j \neq d_{*j}$ или при $R_1 < 0$ и $x_j \neq d_j^*$ полагаем $\alpha = kR_1$, $\sigma = 0$, $j_* = j$.

29. Пусть $tR_1 < 0$. Если $-t/R_1 < \sigma$, то полагаем $\sigma = -t/R_1$, $j_* = j$ при $j \in J_{\text{Н}}$ или $i_* = j$ при $j \in I_{0\text{П}}$, $\alpha = kR_1$.

30. Если $j \in J_{\text{Н}}$, то полагаем $l = d_j^* - x_j$ при $t < 0$, $l = d_{*j} - x_j$ при $t > 0$, $l = 0$ при $t = 0$. Вычислим $x_j = x_j + \Theta l$.

31. Вычислим опорные компоненты плана $x(J_{0\text{П}}) = x(J_{0\text{П}}) + \Theta l(J_{0\text{П}})$.

32. Проверим неравенство $(1 - \Theta)\beta \leq \varepsilon$. Да \rightarrow к п. 66, нет \rightarrow к п. 33.

33. Пусть $\sigma = \sigma_{j_*}$. Вычислим $p(I_{20\text{П}}) = A(I_{0\text{П}}, j_*) - A(I_{20\text{П}}, J_{10\text{П}}) A_{10\text{П}}^{-1} A(I_{10\text{П}}, j_*)$; $v(J_{20\text{П}}) = \tilde{A}_{20\text{П}}^{-1} p(I_{20\text{П}})$.

Пересчитаем матрицу $\tilde{A}_{20\text{П}}^{-1} = [E + (e_{j_0} - v(J_{20\text{П}})) e_{j_0}' / v(j_0)] \tilde{A}_{20\text{П}}^{-1}$ и, полагая $J_{20\text{П}} = (J_{20\text{П}} \setminus j_0) \cup j_*$, $it = it + 1$, перейдем к п. 1.

34. Если $\sigma = \sigma_{i_*}$, $i_* \in I_{20\text{П}}$, перейдем к п. 38.

35. Если $\sigma = \sigma_{i_*}$, $i_* \in I_{10\text{п}}$, вычислим $p(I_{20\text{п}}) = A(I_{20\text{п}}, J_{10\text{п}}) A_{10\text{п}}^{-1}(J_{10\text{п}}, i_*)$. Найдем $R_1 = |p_{i \times}| = \max_{i \in I_{20\text{п}}} |p_i|$.

36. Вычислим $v(J_{20\text{п}}) = -\tilde{A}_{20\text{п}}^{-1} p(I_{20\text{п}})$ и заменим столбец в $\tilde{A}_{20\text{п}}^{-1} : \tilde{A}_{20\text{п}}^{-1}(J_{20\text{п}}, i_*) = v(J_{20\text{п}})$.

37. Вычислим $p(I_{10\text{п}}) = A(i_*, J_{10\text{п}}) A_{10\text{п}}^{-1}$, пересчитаем матрицу $A_{10\text{п}}^{-1} : A_{10\text{п}}^{-1} = [E - e_{i_*}(p(I_{10\text{п}}) - e_{i_*})' / p(i_*)]$. Положим $I_{10\text{п}} = (I_{10\text{п}} \setminus i_*) \cup i_{\times}$, $I_{20\text{п}} = (I_{20\text{п}} \setminus i_{\times}) \cup i_*$.

38. Пересчитаем матрицу $A_{20\text{п}}^{-1} = A_{20\text{п}}^{-1}(J_{20\text{п}} \setminus j_0, I_{20\text{п}} \setminus i_*) - \tilde{A}_{20\text{п}}^{-1}(J_{20\text{п}} \setminus j_0, i_*) \tilde{A}_{20\text{п}}^{-1}(j_0, I_{20\text{п}} \setminus i_*) / \tilde{A}_{20\text{п}}^{-1}(j_0, i_*)$.

39. Положим $I_{20\text{п}} = I_{20\text{п}} \setminus i_*$, $J_{20\text{п}} = J_{20\text{п}} \setminus j_0$, $r = r - 1$, $it = it + 1$ и перейдем к п. 1. Операции пп. 41, 42 выполним для $j \in J_{\text{н}} \cup I_{10\text{п}}$; вектор $z(J_{\text{н}} \cup I_{10\text{п}})$ вычислим по формуле (15) § 1 гл. 1.

40. Вычислим $u'(I_{10\text{п}}) = c'(J_{10\text{п}}) A_{10\text{п}}^{-1}$. Положим $\sigma = +\infty$.

41. Если $j \in I_{10\text{п}}$, то полагаем $t = u_j$, $R_1 = k A_{10\text{п}}^{-1}(j_0, j)$; если $j \in J_{\text{н}}$, то полагаем $t = u'(I_{10\text{п}}) A(I_{10\text{п}}, j) - c_j$, $R_1 = k A_{10\text{п}}^{-1}(j_0, I_{10\text{п}}) A(I_{10\text{п}}, j)$.

42. Выполним пп. 28—30 данного алгоритма.

43. Выполним пп. 31, 32.

44. Если $\sigma = \sigma_{i_*}$, то пересчитаем матрицу $A_{10\text{п}}^{-1} = A_{10\text{п}}^{-1}(J_{10\text{п}} \setminus j_0, I_{10\text{п}} \setminus i_*) - A_{10\text{п}}^{-1}(J_{10\text{п}} \setminus j_0, i_*) A_{10\text{п}}^{-1}(j_0, I_{10\text{п}} \setminus i_*) / A_{10\text{п}}^{-1}(j_0, i_*)$ и, положив $J_{10\text{п}} = J_{10\text{п}} \setminus j_0$, $I_{10\text{п}} = I_{10\text{п}} \setminus i_*$, $it = it + 1$, $r = r - 1$, перейдем к п. 1.

45. Если $\sigma = \sigma_{j_*}$, то пересчитаем матрицу $A_{10\text{п}}^{-1} = [E + (e_{j_0} - p(J_{10\text{п}})) e_{j_0}' / p(j_0)] A_{10\text{п}}^{-1}$, где $p(J_{10\text{п}}) = A_{10\text{п}}^{-1} A(I_{10\text{п}}, j_*)$, и, полагая $J_{10\text{п}} = (J_{10\text{п}} \setminus j_0) \cup j_*$, $it = it + 1$, перейдем к п. 1.

46. Пусть $\Theta = \Theta_{i_0}$. Следуя пп. 53—55, вычислим $z'(J_{\text{н}} \cup I_{10\text{п}} \cup I_{20\text{п}}) = k \{ \tilde{A}(i_0, J_{\text{н}}) - \tilde{A}(i_0, J_{20\text{п}}) \tilde{A}_{20\text{п}}^{-1} \tilde{A}(I_{20\text{п}}, J_{\text{н}}), -\tilde{A}(i_0, I_{10\text{п}}) + \tilde{A}(i_0, J_{20\text{п}}) \tilde{A}_{20\text{п}}^{-1} \tilde{A}(I_{20\text{п}}, I_{10\text{п}}), -\tilde{A}(i_0, J_{20\text{п}}) \tilde{A}_{20\text{п}}^{-1} \}$. При $r \leq m_1$ вектор $z'(J_{\text{н}} \cup I_{\text{оп}})$ строится по формуле (14) § 1 гл. 1. Полагаем $\sigma = +\infty$.

47. Пусть $r \leq m_1$. Вычислим $p(I_{10п}) = A(i_0, J_{10п})A_{10п}^{-1}$. Для $j \in I_{10п} \cup J_{н}$ выполним пп. 48, 49.

48. Если $j \in I_{10п}$, то полагаем $t = u_j$, $R_1 = -kp_j$; если $j \in J_{н}$, то полагаем $t = u'(I_{10п})A(I_{10п}, j) - c_j$, $R_1 = A(i_0, j) - p(I_{10п})A(I_{10п}, j)$.

49. Выполним пп. 28—30 (с каждым $j \in I_{10п} \cup J_{н}$).

50. Выполним пп. 31, 32.

51. Если $\sigma = \sigma_{j_*}$, то вычислим $v(J_{10п}) = A_{10п}^{-1}A(I_{10п}, j_*)$ и пересчитаем матрицу

$$A_{10п}^{-1} = \begin{Bmatrix} A_{10п}^{-1} + v(J_{10п})p'(I_{10п})/\alpha, & -v(J_{10п})/\alpha \\ -p'(I_{10п})/\alpha, & 1/\alpha \end{Bmatrix}$$

(α запоминается при вычислении σ в пп. 28, 29).

52. Положим $r = r + 1$, $it = it + 1$, $J_{10п} = J_{10п} \cup j_*$, $I_{10п} = I_{10п} \cup i_0$ и перейдем к п. 1.

53. Пусть $r > m_1$. Вычислим $v(I_{20п}) = [A(i_0, J_{20п}) - A(i_0, J_{10п})A_{10п}^{-1}A(I_{10п}, J_{20п})]\tilde{A}_{20п}^{-1}$. Пункты 54—56 выполним для $j \in I_{20п} \cup I_{10п} \cup J_{н}$.

54. Если $j \in I_{20п}$, то полагаем $t = u_j$, $R_1 = -kv_j$; если $j \in I_{10п}$, то полагаем $t = u_j$, $R_1 = k[v'(I_{20п})A(I_{20п}, J_{10п}) - A(i_0, J_{10п})]A_{10п}^{-1}(J_{10п}, j)$.

55. Если $j \in J_{н}$, то $t = u'(I_{10п})A(I_{10п}, j) - c_j$, $R_1 = k[A(i_0, j) - A(i_0, J_{10п})A_{10п}^{-1}A(I_{10п}, j) - v'(I_{20п})(A(I_{20п}, j) - A(I_{20п}, J_{10п})A_{10п}^{-1}A(I_{10п}, j))]$.

56. Выполним пп. 28—30.

57. Выполним пп. 31, 32.

58. Пусть $\sigma = \sigma_{j_*}$. Вычислим $p(J_{20п}) = A_{20п}^{-1}[A(I_{20п}, j_*) - A(I_{20п}, J_{10п})]A_{10п}^{-1}A(I_{10п}, j_*)$ и пересчитаем матрицу

$$\tilde{A}_{20п}^{-1} = \begin{Bmatrix} \tilde{A}_{20п}^{-1} + p(J_{20п})v'(I_{20п})/\alpha, & -p(J_{20п})/\alpha \\ -v'(I_{20п})/\alpha, & 1/\alpha \end{Bmatrix}.$$

59. Положим $J_{20п} = J_{20п} \cup j_*$, $I_{20п} = I_{20п} \cup i_0$, $r = r + 1$, $it = it + 1$ и перейдем к п. 1.

60. Если $\sigma = \sigma_{i_*}$ и $r \leq m_1$, то перейдем к п. 65.

61. Пусть $\sigma = \sigma_{i_*}$, $r > m_1$. Если $i_* \in I_{20п}$, то пересчитаем матрицу $\tilde{A}_{20п}^{-1} = \tilde{A}_{20п}^{-1}[E - e_{i_*}(v(I_{20п}) - e_{i_*})'/v(i_*)]$ и, полагая $I_{20п} = (I_{20п} \setminus i_*) \cup i_0$, $it = it + 1$, перейдем к п. 1.

62. Выполним п. 35. Если $R_1 = 0$, то перейдем к п. 65, иначе — к п. 63.

63. Выполним пп. 36, 37.

64. Вычислим $v(I_{20п}) = [A(i_0, J_{20п}) - A(i_0, J_{10п}) A_{10п}^{-1} A(J_{10п}, J_{20п})] \tilde{A}_{20п}^{-1}$ и пересчитаем $\tilde{A}_{20п}^{-1} = \tilde{A}_{20п}^{-1} [E - e_{i_*} (v(I_{20п}) - e_{i_*})' / v(i_*)]$. Положим $I_{20п} = (I_{20п} \setminus i_*) \cup i_0$, $it = it + 1$ и перейдем к п. 1.

65. Вычислим $p(I_{10п}) = A(i_0, J_{10п}) A_{10п}^{-1}$ и пересчитаем матрицу $A_{10п}^{-1} = A_{10п}^{-1} [E - e_{i_*} (p(I_{10п}) - e_{i_*})' / p(i_*)]$. Положим $I_{10п} = (I_{10п} \setminus i_*) \cup i_0$, $it = it + 1$ и перейдем к п. 1.

66. Выдача результатов.

Настоящий алгоритм реализован на языке АЛГОЛ-ГДР (МС «ДУБНА»). Стандартные операции алгоритма, например, формулы пересчета матриц $A_{10п}^{-1}$ и $\tilde{A}_{20п}^{-1}$, реализованы в виде процедур с параметрами, значения которых настраивают процедуру на пересчет заданной матрицы. Аналогично оформлены и другие мало различающиеся фрагменты алгоритма (всего 23 процедуры).

Данные задачи (1) хранятся в компактной форме, без нулевых элементов. Программа не использует дополнительных n -мерных векторов (их компоненты при необходимости вычисляются вновь). Вводятся только пять m -мерных массивов, участвующих и в описании алгоритма.

В сравнении с вариантом адаптивного алгоритма без декомпозиции опоры машинная программа, реализующая данный алгоритм, занимает дополнительно около 1200 слов. Эти затраты окупаются экономией памяти для обратной опорной матрицы. Например, если размер опоры при решении задачи достигает 200, то для полной матрицы потребуется 40 000 слов, а для матрицы с декомпозицией — 20 000. Поэтому в изложенном алгоритме исключается обмен с внешней памятью при решении задач средних размеров.

3. Численные эксперименты. Для сравнения скорости работы двух алгоритмов — без декомпозиции и с декомпозицией — были проведены расчеты серии задач разных размеров. Задачи, как и в эксперименте 3 § 1 гл. 1, генерировались с помощью процедур GENER и VEK. Каждая из сгенерированных задач вначале была решена при $m_1 = m$, и, следовательно, блоки программы, выполняющие счет с помощью двух рабочих матриц $A_{10п}^{-1}$ и $\tilde{A}_{20п}^{-1}$, были отключены, а работал вариант адаптивного алгоритма, описанный в § 2 гл. 1. Затем вновь генерирова-

лись эти же задачи и каждая решалась изложенным здесь алгоритмом при m_1 , значение которого задавалось равным примерно половине максимального размера опоры. При отсутствии обмена с внешней памятью время решения задачи по алгоритму с декомпозицией опоры было несколько больше. При наличии обмена, который для задач небольших размеров был организован условно, алгоритм с декомпозицией опоры работал значительно быстрее, причем этот эффект растет с увеличением размеров задачи. Результаты эксперимента приведены в табл. 28, где номера задач 1, 2, 3, 4, 5 совпадают с числом ненулевых элементов в каждом столбце матрицы.

Табл. 28

$m \times n$	m_1	Число итераций	Максимальный размер опоры	Время решения, с	
				без декомпозиции	с декомпозицией
20×30	6	13	12	4.0	3.6
		11	11	3.5	3.2
		16	12	5.8	5.4
		19	12	6.7	6.2
		18	13	7.7	7.1
30×45	10	17	17	9.4	8.2
		21	19	14.6	13.1
		34	21	29.8	24.9
		36	21	30.5	26.2
		34	17	27.1	23.9
70×100	23	43	43	144	124
		58	42	226	185
		65	46	282	245
		67	45	284	247
		84	45	415	365
100×150	33	62	59	491	470
		86	71	825	750
		97	65	1282	942
		116	67	1626	1216
		132	63	1821	1371

§ 5. Прикладные задачи

Для иллюстрации адаптивного метода приведем решение двух небольших реальных задач прикладного характера. В процессе решения специфика задач не учитывалась, т. е. матрицы условий их математических моделей рассматривались как общие.

1. Задача о выборе способа добычи угля. В Кузнецком угольном бассейне работает несколько угледобывающих предприятий, каждое из которых может вести добычу угля различными способами. Множество способов добычи i го предприятия обозначим через K_i . Введем переменные λ_{ik} , принимающие значения либо 1, если на i -м предприятии используется k -й способ, либо 0, если этот способ не используется. Количество угля марки s , добываемого i -м предприятием по k -му способу, обозначим через a_{ik}^s . Пусть z_s — количество угля марки s , которое добывается во всем бассейне.

Будем считать, что модель добычи угля описывается следующими соотношениями баланса: $\sum_{k \in K_i} \lambda_{ik} = 1, i = 1, 2, \dots,$

33; $\bigcup_1^{33} K_i = K, |K| = 56$ — число переменных λ_{ik} ; $\sum_{i,k} a_{ik}^1 \lambda_{ik} - z_1 = 0$ (рядовой уголь); $\sum_{i,k} a_{ik}^2 \lambda_{ik} - z_2 = 0$ (сортовой уголь); $\sum_{i,k} a_{ik}^3 \lambda_{ik} - z_3 = 0$ (отсев); $z_3 - z_4 = 0$ (пылевое сжигание); $z_2 - z_5 = 0$ (слоевое сжигание); $\sum_{i,k} a_{ik}^4 \lambda_{ik} - z_6 = 0$ (общая добыча).

Кроме того, учитываются потребности угля: $z_4 \geq 85$ (для пылевого сжигания); $z_5 \geq 33$ (для слоевого сжигания).

Задача выбора каждым i -м предприятием k -го способа добычи угля, при котором минимальны суммарные приведенные затраты, сводится к минимизации функции $\sum_{i,k} c_{ik} \lambda_{ik} \rightarrow \min$ при выполнении перечисленных условий.

Балансовые соотношения объединим в матрицу A , в которой 41 строка, но лишь в строках с номерами 34—36, 39 находятся элементы a_{ij} , отличные от 1. Переменные объединим в вектор $x = \{x_j = \lambda_{ik}, j = 1, 56, x_j =$

$=z_{j-56}, j=\overline{57,62}\}$. Полученная матрица условий имеет специальную структуру, но рассматривалась как общая редкозаполненная матрица.

Для применения процедуры, реализующей адаптивный алгоритм в интервальной задаче, векторы b_*, b^*, d_*, d^* задавались следующими: $b_{*i} = 1 - 10^{-4}$, $b_i^* = 1 + 10^{-4}$, $i = \overline{1,33}$, $b_{*i} = -10^{-4}$, $b_i^* = 10^{-4}$, $i = \overline{34,39}$, $b_{*40} = 85$, $b_{*41} = 33$, $b_{40}^* = b_{41}^* = 10^8$; $d_{*j} = 0$, $d_j^* = 10^8$, $j = \overline{1,62}$.

Первое приближение: $x=0$. С помощью стартовой процедуры строилась расширенная задача, в которой дополнительные коэффициенты целевой функции полагались равными -10^7 (у остальных коэффициентов менялись знаки, так как процедура выполняет максимизацию). Критерий оптимальности в расширенной задаче был выполнен за 46 итераций, на что потребовалось 35 с. Решение этой задачи симплекс-методом было получено за 50 итераций (36 с). На полученном решении не все балансовые соотношения выполняются, что объясняется дефицитом угля некоторых марок. Первые 33 условия на λ_{i_k} выполнены.

Исходные данные и результаты вычислений приведены в табл. 29.

2. Задача оптимизации генерирующей структуры объединенной электроэнергетической системы (ОЭЭС). В математической модели ОЭЭС учитывались следующие ограничения: баланс мощности и энергии по зоне —

$$\sum_{q \in \Theta_n} \sum_{k \in K_q} \omega_{gk} x_k \geq y_{gn}, \quad g = 1, 2, \dots, 6, \quad n = 1, 2, \dots, N;$$

балансы мощности и энергии по энергоузлу —

$$\sum_{q \in \Theta_n} \sum_{k \in K_q} \omega_{lk} x_k + \sum_{p \in L_n} (\alpha_{pl} x_{pl} - x_{lp}) \geq y_l, \\ p \in L_n, l \in L_n, n = 1, 2, \dots, N;$$

$$\sum_{q \in \Theta_n} \sum_{k \in K_q} \tilde{\omega}_{lk} x_k + \sum_{p \in L_n} (\tilde{\alpha}_{pl} x_{pl} - x_{lp}) \geq \tilde{y}_l, \\ p \in L_n, l \in L_n, n = 1, 2, \dots, N;$$

связь потоков мощности и энергии по внутрисистемным ЛЭП с пропускной способностью этих ЛЭП —

$j: 1$	2	3	4	5	6
1 ; 1 2.97; 35 2.44; 36 6 ; 39 $c_j: 0$ $x_j^0: 1$	1 ; 2 1.96; 34 5.6 ; 35 4.64; 36 14.2 ; 39 0 1	1 ; 3 3.76; 35 2.19; 36 8 ; 39 0 1	1 ; 4 0.77; 34 2.13; 35 1.83; 36 5.5 ; 39 0 1	1 ; 5 0.57; 34 0.7 ; 39 0 1	1 ; 6 0.35; 34 0.57; 35 0.87; 36 2 ; 39 0 1
$j: 7$	8	9	10	11	12
1 ; 7 0.21; 35 0.51; 36 0.8 ; 39 $c_j: 0$ $x_j^0: 1$	1 ; 8 2.72; 34 3.1 ; 39 0 1	1 ; 9 1.21; 34 0.47; 35 0.83; 36 2.9 ; 39 0 1	1 ; 10 0.99; 34 1.3 ; 39 0 1	1 ; 11 1.67; 34 1.73; 35 4.06; 36 8.85; 39 0 1	1 ; 12 1.3 ; 35 2.62; 36 4.75; 39 0 1
$j: 13$	14	15	16	17	18
1 ; 13 0.71; 34 0.4 ; 35 0.67; 36 0.21; 39 $c_j: 0$ $x_j^0: 1$	1 ; 14 1 ; 35 5.4; 36 2 ; 39 23.5 1	1 ; 14 1 ; 35 0.38; 36 2 ; 39 18.6 0	1 ; 15 0.8 ; 35 0.88; 36 2 ; 39 15.4 1	1 ; 15 0.64; 35 0.7 ; 36 1.6 ; 39 11.7 0	1 ; 16 3.65; 35 2.33; 36 7 ; 39 61.5 1
$j: 19$	20	21	22	23	24
1 ; 16 4.1; 34 4.6; 39 $c_j: 25.9$ $x_j^0: 0$	1 ; 17 0.43; 35 0.72; 36 1.4 ; 39 20.6 1	1 ; 18 1.68; 35 1.7 ; 36 3.64; 39 49.4 0	1 ; 18 2.8 ; 35 2.78; 36 6 ; 39 69.4 1	1 ; 18 2.22; 34 1.14; 35 1.22; 36 4.7 ; 39 37.7 0	1 ; 19 1.54; 35 1.01; 36 2.8 ; 39 29.7 1

j : 25	26	27	28	29	30
1 ; 19 0.29; 34 1.14; 35 0.98; 36 2.55; 39 c_j : 24.5 x_j^0 : 0	1 ; 20 2.38; 35 0.34; 36 0.8 ; 39 18.7 0	1 ; 20 0.62; 35 0.55; 36 1.3 ; 39 25.4 1	1 ; 21 1.09; 35 1.42; 36 2.95; 39 69.2 1	1 ; 21 0.54; 35 0.55; 36 1.3 ; 39 25.9 0	1 ; 21 1.09; 35 1.42; 36 2.95; 39 54.3 0
j : 31	32	33	34	35	36
1 ; 22 0.22; 35 2.21; 36 3.6 ; 39 c_j : 46.8 x_j^0 : 1	1 ; 22 0.67; 35 2.04; 36 3.6 ; 39 46.6 0	1 ; 23 0.18; 35 0.33; 36 0.51; 39 10.5 1	1 ; 24 1.36; 35 3.65; 36 6 ; 39 38.1 1	1 ; 25 0.26; 35 0.52; 36 0.9 ; 39 6.5 0	1 ; 25 0.46; 35 0.93; 36 1.6 ; 39 14.1 1
j : 37	38	39	40	41	42
1; 26 0.38; 34 0.4 ; 39 c_j : 2.8 x_j^0 : 0	1 ; 26 1.41; 34 1.5 ; 39 12.9 1	1 ; 27 1.95; 34 2.2 ; 39 15.1 1	1 ; 28 1.75; 34 2.5 ; 39 30.5 0	1 ; 28 0.82; 35 0.98; 36 0.25; 39 33.7 1	1 ; 29 2.62; 34 4 ; 39 49.2 0
j : 43	44	45	46	47	48
1 ; 29 1.3; 35 1.4; 36 4 ; 39 c_j : 54.4 x_j^0 : 1	1 ; 30 1.37; 34 2 ; 39 26.9 0	1 ; 30 0.65; 35 0.75; 36 2 ; 39 29.3 1	1 ; 31 5.09; 34 6 ; 39 97.3 0	1 ; 31 2.54; 35 2.47; 36 6 ; 39 108.4 1	1 ; 32 3.35; 34 4 ; 39 67.7 0

j : 49	50	51	52	53	54		
1 ; 32 1.86; 35 1.43; 36 4 ; 39 c_j : 74.3 x_j^0 : 1	1 ; 33 0.16; 35 0.37; 36 0.6 ; 39 12.2 1	1; 28 0 0	1; 29 0 0	1; 30 0 0	1; 31 0 0		
j : 55	56	57	58	59	60	61	62
1; 32 c_j : 0 x_j^0 : 0	1; 33 0 0	-1; 34 1; 37 0 19.89	-1; 35 1; 38 0 28.89	-1; 36 1; 37 0 32.09	-1; 37 1; 40 0 51.98	-1; 38 1; 41 0 28.88	-1; 39 0 92.76

Примечание. В каждой клетке без имени помещены два числа: первое — значение элемента a_{ij} , второе — номер строки i . Остальные элементы матрицы A равны нулю, j — номер столбца, c_j — коэффициенты целевой функции, x_j^0 — компоненты оптимального плана.

$$x_{pl} + x_{lp} - z_{pl} \geq R_{pl}, p \in L_n, l \in L_n,$$

$$n = 1, 2, \dots, N;$$

$$x_{pl} + \tilde{x}_{lp} - h_{pl} z_{pl} \geq h_{pl} R_{pl}, p \in L_n, l \in L_n,$$

$$n = 1, 2, \dots, N;$$

балансы топлива и дефицитного оборудования —

$$\sum_{q \in \Theta_n} \sum_{k \in K_q} b_k^t x_k = y_n^t; t = 1, 2, \dots, T, n = 1, 2, \dots, N;$$

$$y_n^t = \sum_m y_{mn}^t, t = 1, 2, \dots, T, n = 1, 2, \dots, N;$$

баланс топлива по месторождениям и ограничения на их производительность, пропускную способность транспорта —

$$\sum_n y_{mn}^t = \sum_{f \in F_m^t} x_f^t, x_f^t \leq \varphi_f^t; f = 1, 2, \dots, F, t = 1, 2, \dots, T;$$

$$y_{mn}^t \leq D_{mn}^t, (m, n, t) \in K;$$

ограничения на мощность и режимы работы электростанций —

$$\sum_{k \in K_q} x_k \leq R_q, k_q \in K_q, q \in \Theta_n, n = 1, 2, \dots, N.$$

Целевая функция имеет вид

$$\begin{aligned} \sum_k c_k x_k + \sum_n \sum_{p \in L_n} \sum_{l \in L_n} c_{pl} z_{pl} + \sum_t \sum_m \sum_n c_{mn}^t y_{mn}^t + \\ + \sum_t \sum_f c_f^t x_f^t \rightarrow \min, \end{aligned}$$

$$\text{где } c_k = \bar{c}_k + \sum_t c_{nt}^t b_k^t.$$

Исходные данные и результаты вычислений приведены в табл. 30. Порядок расположения элементов такой же, как и в табл. 29. Кроме того, учитывались следующие ограничения. Нижние границы плана: $d_{*8}=0,31$; $d_{*56}=0,06$; $d_{*82}=17,8$; $d_{*83}=10,3$; $d_{*85}=11,1$; $d_{*86}=3,2$; $d_{*87}=7,8$; $d_{*88}=2,4$; остальные компоненты d_* равны 0. Верхние границы плана: $d_7^*=1,58$; $d_8^*=1,58$; $d_{55}^*=0,29$; $d_{56}^*=0,29$; $d_{82}^*=19,8$; $d_{83}^*=12,3$; $d_{84}^*=0,8$; $d_{85}^*=12,7$; $d_{86}^*=4,8$; $d_{87}^*=9,4$; $d_{88}^*=4$; $d_{89}^*=0,8$; $d_{90}^*=0,8$; остальные компоненты d^* равны ∞ (10^6). Векторы ограничений $b_* = \{5,89; 2,9; 22,45; 88,27; 9,31; 17,36; 68,56; 10,4; 44,49; 9,88; 43,29; 1,8 \cdot 10^{-4}; 2 \cdot 10^{-4}; 1,2 \cdot 10^{-4}; 0,6 \cdot 10^{-4}; 1,2 \cdot 10^{-4}; 0,92 \cdot 10^{-4}; -10^5; \dots; -10^5\}$; $b^* = \{10^5; 10^5; \dots; 10^5; b_{12}^* = 1,8 + 10^{-4}; 2 + 10^{-4}; 1,2 + 10^{-4}; 0,6 + 10^{-4}; 1,2 + 10^{-4}; 0,92 + 10^{-4}; b_{18}^* = 0, \dots, b_{30}^* = 0\}$.

Первое приближение: $x=d_*$. С помощью стартовой процедуры сформулирована расширенная задача, в процессе решения которой вначале был найден план задачи, а затем — оптимальный план. Задача решалась двумя методами: опорным с симплексной нормировкой и методом § 1. Оптимальный план в первом случае получен за 86 итераций (52 с), во втором — за 66 итераций (40 с).

§ 6. Адаптивный метод решения динамической транспортной задачи

Одним из современных направлений связанных с обобщением классических задач линейного программи-

j : 1	2	3	4	5	6
0.9 ; 3	0.9 ; 3	0.7; 3	0.3 ; 3	0.85; 3	0.7 ; 3
2.3 ; 4	3.3 ; 4	2.9; 4	1.1 ; 4	1.95; 4	2.9 ; 4
0.1 ; 5	0.1 ; 5	0.3; 5	0.7 ; 5	0.15; 5	0.3 ; 5
1 ; 6	1 ; 6	1 ; 6	1 ; 6	1 ; 6	1 ; 6
3 ; 7	4 ; 7	5 ; 7	6 ; 7	3 ; 7	5 ; 7
1 ; 12	1 ; 12	1 ; 12	1 ; 12	1 ; 13	1 ; 13
1.02; 22	1.33; 22	1.6; 22	1.69; 22	1.1 ; 23	1.72; 23
c_j : 0	0	0	0	0	0
x_j^0 : 0	1.8	0	0	0	0
j : 7	8	9	10	11	12
1; 1	1 ; 2	1 ; 1	1 ; 1	1 ; 1	1 ; 1
1; 6	1 ; 7	0.2; 2	1 ; 2	0.2 ; 2	1 ; 2
	0.45; 23	1 ; 6	1 ; 6	1 ; 3	1 ; 3
		0.2; 7	1 ; 7	6.71; 4	5.6; 4
		0.1; 22	0.5; 22	—1 ; 5	—1 ; 5
		1 ; 24	1 ; 24	1 ; 6	1 ; 6
				—0.09; 7	—0.4; 7
				1 ; 26	1 ; 26
c_j : 0	0	16.15	16.15	28.56	28.56
x_j^0 : 1.58	1.58	0.702	0	3.035	0.282
j : 13	14	15	16	17	18
0.9 ; 3	0.9 ; 3	0.9 ; 3	0.9 ; 3	0.6 ; 3	0.3 ; 3
2.3 ; 4	3.3 ; 4	2.3 ; 4	3.3 ; 4	2.7 ; 4	1.7 ; 4
0.1 ; 5	0.1 ; 5	0.1 ; 5	0.1 ; 5	0.4 ; 5	0.7 ; 5
1 ; 6	1 ; 6	1 ; 6	1 ; 6	1 ; 6	1 ; 6
3 ; 7	4 ; 7	3 ; 7	4 ; 7	5.5 ; 7	6.6 ; 7
1.42; 22	1.87; 22	1.46; 23	1.93; 23	1.72; 22	2.04; 22
1 ; 27	1 ; 27	1 ; 28	1 ; 28	1 ; 29	1 ; 29
c_j : 19	19	22.9	22.9	27.15	27.15
x_j^0 : 0	0	1.256	0	0	0

j : 19	20	21	22	23	24
1 ; 5	0.6 ; 3	0.2 ; 3	1 ; 6	0.3; 3	0.2; 3
1 ; 6	2.7 ; 4	1 ; 4	7 ; 7	0.6; 4	1.3; 4
7 ; 7	0.4 ; 5	0.8 ; 5	2.24; 23	0.7; 5	0.8; 5
2.16; 22	1 ; 6	1 ; 6	1 ; 30	1 ; 6	1 ; 6
1 ; 29	5.5 ; 7	6.6 ; 7		5.5; 7	6.9; 7
	1.76; 23	2.11; 23		1 ; 25	1 ; 25
	1 ; 30	1 ; 30			
c_j : 27.15	32.19	32.19	32.19	53.2	55.37
x_j^0 : 0	0	0	0	0	6.704
j : 25	26	27	28	29	30
1; 5	0.9 ; 3	0.9 ; 3	0.7; 3	0.3 ; 3	0.9 ; 3
1; 6	2.3 ; 4	3.3 ; 4	2.9; 4	1.1 ; 4	2.3 ; 4
7; 7	0.1 ; 5	0.1 ; 5	0.3; 5	0.7 ; 5	0.1 ; 5
1; 25	1 ; 8	1 ; 8	1 ; 8	1 ; 8	1 ; 8
	3 ; 9	4 ; 9	5 ; 9	6 ; 9	3 ; 9
	1 ; 14	1 ; 14	1 ; 14	1 ; 14	1 ; 15
	1.02; 22	1.33; 22	1.6; 22	1.89; 22	1.07; 22
c_j : 55.6	1.53	2	2.4		1.61
x_j^0 : 0	0	0	1.2	0	0.6
j : 31	32	33	34	35	36
0.7; 3	1 ; 1	1 ; 1	1 ; 1	1 ; 1	0.9 ; 3
2.9; 4	0.2; 2	1 ; 2	0.2 ; 2	1 ; 2	2.3 ; 4
0.3; 5	1 ; 8	1 ; 8	1 ; 3	1 ; 3	0.1 ; 5
1 ; 8	0.2; 9	1 ; 9	6.71; 4	5.6; 4	1 ; 8
5 ; 9	0.1; 22	0.5; 22	-1 ; 5	-1 ; 5	3 ; 9
1 ; 15	1 ; 24	1 ; 24	1 ; 8	1 ; 8	1.42; 22
1.7; 22			-0.09; 9	-0.4; 9	1 ; 27
			1 ; 26	1 ; 26	
c_j : 2.55	16.3	16.9	28.56	28.56	21.13
x_j^0 : 0	0	0	0	0	0.94

j : 37	38	39	40	41	42
0.9 ; 3	0.9 ; 3	0.9 ; 3	0.6 ; 3	0.3 ; 3	1 ; 5
3.3 ; 4	2.3 ; 4	3.3 ; 4	2.6 ; 4	1.7 ; 4	1 ; 8
0.1 ; 5	0.1 ; 5	0.1 ; 5	0.4 ; 5	0.7 ; 5	7 ; 9
1 ; 8	1 ; 8	1 ; 8	1 ; 8	1 ; 8	2.16; 22
4 ; 9	3 ; 9	4 ; 9	5.5 ; 9	6.6 ; 9	1 ; 29
1.87; 22	1.46; 23	1.93; 23	1.72; 22	2.04; 22	
1 ; 27	1 ; 28	1 ; 28	1 ; 29	1 ; 29	
c_f : 21.81	24.36	24.83	29.73	30.21	30.39
x_f^0 : 0	0	0	0	0	0
j : 43	44	45	46	47	48
0.6 ; 3	0.2 ; 3	1 ; 5	0.3; 3	0.2; 3	1; 5
2.7 ; 4	1 ; 4	1 ; 8	0.6; 4	1.3; 4	1; 8
0.4 ; 5	0.8 ; 5	7 ; 9	0.7; 5	0.8; 5	7; 9
1 ; 8	1 ; 8	2.24; 23	1 ; 8	1 ; 8	1; 25
5.5 ; 9	6.6 ; 9	1 ; 30	5.5; 9	6.9; 9	
1.76; 23	2.11; 23		1 ; 25	1 ; 25	
1 ; 30	1 ; 30				
c_f : 33.95	34.3	34.43	53.2	55.37	55.6
x_f^0 : 0	0	0	0	0.171	0
j : 49	50	51	52	53	54
0.9 ; 3	0.9 ; 3	0.7; 3	0.3 ; 3	0.85; 3	0.7 ; 3
2.3 ; 4	3.3 ; 4	2.9; 4	1.1 ; 4	1.95; 4	2.9 ; 4
0.1 ; 5	0.1 ; 5	0.3; 5	0.7 ; 5	0.15; 5	0.3 ; 5
1 ; 10	1 ; 10	1 ; 10	1 ; 10	1 ; 10	1 ; 10
3 ; 11	4 ; 11	5 ; 11	6 ; 11	3 ; 11	5 ; 11
1 ; 16	1 ; 16	1 ; 16	1 ; 16	1 ; 17	1 ; 17
1.02; 22	1.33; 22	1.6; 22	1.89; 22	1.1 ; 23	1.72; 23
c_f : 1.02	1.33	1.6	1.89	0	0
x_f^0 : 0.852	0	0.348	0	0	0.92

j : 55	56	57	58	59	60
1; 1 1; 10	1 ; 2 1 ; ; 11 0.45; 23	1 ; 1 0.2; 2 1 ; 10 0.2; 11 0.1; 22 1 ; 24	1 ; 1 1 ; 2 1 ; 10 1 ; 11 0.5; 22 1 ; 24	0.9 ; 3 2.3 ; 4 0.1 ; 5 1 ; 10 3 ; 11 1.42; 22 1 ; 27	0.9 ; 3 3.3 ; 4 0.1 ; 5 1 ; 10 4 ; 11 1.87; 22 1 ; 27
c_j : 0 x_j^0 : 0.29	0 0.29	16.25 0	16.65 0	20.42 0	20.87 0
j : 61	62	63	64	65	66
0.9 ; 3 2.3 ; 4 0.1 ; 5 1 ; 10 3 ; 11 1.46; 23 1 ; 28 c_j : 22.9 x_j^0 : 0.067	0.9 ; 3 3.3 ; 4 0.1 ; 5 1 ; 10 4 ; 11 1.93; 23 1 ; 28 22.9 1.077	0.6 ; 3 2.7 ; 4 0.4 ; 5 1 ; 10 5.5 ; 11 1.72; 22 1. ; 29 28.87 0	0.3 ; 3 1.7 ; 4 0.7 ; 5 1 ; 10 6.6 ; 11 2.04; 22 1 ; 29 29.19 0	1 ; 5 1 ; 10 7 ; 11 2.16; 22 1 ; 29 1 ; 30 29.31 0	0.6 ; 3 2.7 ; 4 0.4 ; 5 1 ; 10 5.5 ; 11 1.76; 23 1 ; 30 32.19 0
j : 67	68	69	70	71	72
0.2 ; 3 1 ; 4 0.8 ; 5 1 ; 10 6.6 ; 11 2.11; 23 1 ; 30 c_j : 32.19 x_j^0 : 0.202	1 ; 5 1 ; 10 7 ; 11 2.24; 23 1 ; 30 32.19 0	0.3; 3 0.6; 4 0.7; 5 1 ; 10 5.5; 11 1 ; 25 53.2 0	0.2; 3 1.3; 4 0.8; 5 1 ; 10 6.9; 11 1 ; 25 55.37 5.211	1; 5 1; 10 7; 11 1; 25 55.6 0	-1 ; 6 0.935; 8 1 ; 18 0.01 0

j : 73	74	75	76	77	78
0.935; 6 —1 ; 8 1 ; 18 c_j : 0.01 x_j^0 : 0	—1 ; 7 0.935; 9 1 ; 19 0.01 0	0.935; 7 —1 ; 9 1 ; 19 0.01 0	—1; 18 —8; 19 3.9 0	—1 ; 8 0.95; 10 1 ; 20 0.01 0.961	0.95; 8 —1 ; 10 1 ; 20 0.01 0
j : 79	80	81	82	83	84
—1 ; 9 0.95; 11 1 ; 21 c_j : 0.01 x_j^0 : 0	0.95; 9 —1 ; 11 1 ; 21 0.01 7.692	—1; 20 —8; 21 2.5 0.961	—1; 22 22.72 19.8	—1; 23 19.2 10.3	—1; 24 6.29 0.702
j : 85	86	87	88	89	90
—1; 25 c_j : 11.76 x_j^0 : 12.086	—1; 26 0 3.318	—1; 27 0 9.4	—1; 28 0 2.4	—1; 29 0 0	—1; 30 0 0.202

рования, является переход к динамическим постановкам. Многоэтапность, свойственная реальным ситуациям выбора решений, существенно усложняет математические модели задач. «Наивное» перенесение на новый круг задач методов решения статических задач бесперспективно. В связи с этим возникает проблема реализации классических принципов с максимальным учетом динамичности ограничений. Эта задача в данном параграфе решается для оптимизации транспортных потоков.

1. Постановка задачи. На нестационарной сети $S(t) = \{I, U(t)\}$, изменяющей свои параметры в дискретные моменты времени $t \in T = \{0, 1, 2, \dots, t_1 - 1\}$, рассмотрим следующую динамическую транспортную задачу:

$$\sum_{t \in T} \left(\sum_{(i, j) \in U(t)} c_{ij}(t) x_{ij}(t) + \sum_{i \in I} c_i(t+1) x_i(t+1) \right) \rightarrow \min,$$

$$x_i(t+1) = x_i(t) - \sum_{j \in I_i^+(U(t))} x_{ij}(t) + \sum_{j \in I_i^-(U(t))} x_{ji}(t) + a_i(t), \quad (1)$$

$$0 \leq x_i(t+1) \leq d_i(t+1), \quad x_i(0) = 0, \quad i \in I;$$

$$0 \leq x_{ij}(t) \leq d_{ij}(t), \quad (i, j) \in U(t), \quad t \in T.$$

Здесь $x_i(t)$ — запас некоторого продукта в узле i в момент t ; $a_i(t)$ — интенсивность узла i в момент t ($a_i(t)$ — объем производства, если $a_i(t) > 0$, $a_i(t)$ — объем потребления, если $a_i(t) < 0$); $w_i(t) = - \sum_{j \in I_i^+(U(t))} x_{ij}(t) + \sum_{j \in I_i^-(U(t))} x_{ji}(t)$ —

объем перемещенного продукта в узел i ($w_i(t)$ — объем завезенного продукта, если $w_i(t) > 0$, $w_i(t)$ — объем вывезенного продукта, если $w_i(t) < 0$); $x_{ij}(t)$ — объем перевозки из узла i в узел j ; $c_{ij}(t)$ — стоимость перевозки единицы продукта из i в j ; $c_i(t)$ — стоимость хранения единицы продукта в узле i ; $d_{ij}(t)$ — пропускная способность коммуникации $i \rightarrow j$; $d_i(t)$ — объем склада в узле i ; $I_i^+(U) = \{j \in I : (i, j) \in U\}$, $I_i^-(U) = \{j \in I : (j, i) \in U\}$. Динамическую транспортную задачу (1) представим в виде задачи оптимального управления

$$J(u) = \sum_{t \in T} c'_u(t) u(t) + c'_x(t+1) x(t+1) \rightarrow \min, \quad (2)$$

$$x(t+1) = x(t) - B(t) u(t) + a(t),$$

$$x(0) = 0, \quad u(t) \in V(t) = \{u : 0 \leq u \leq d_u(t)\},$$

$$0 \leq x(t+1) \leq d_x(t+1), \quad t \in T,$$

где $B(t) = \{b^{ij}(t), (i, j) \in U(t)\}$ есть $|I| \times |U(t)|$ -матрица, задаваемая сетью $S(t) = \{I, U(t)\}$; $b^{ij}(t)$ — $|I|$ -вектор с компонентами $b_k^{ij}(t) = 0$, если $k \neq i$, $k \neq j$, $b_i^{ij}(t) = 1$, $b_j^{ij}(t) = -1$; $x(t) = \{x_i(t), i \in I\}$ — вектор состояния системы, $u(t) = \{u_{ij}(t), (i, j) \in U(t)\}$ — вектор управления $d_x(t) = \{d_i(t), i \in I\}$, $c_u(t) = \{c_{ij}(t), (i, j) \in U(t)\}$, $c_x(t) = \{c_i(t), i \in I\}$, $d_u(t) = \{d_{ij}(t), (i, j) \in U(t)\}$.

Управление $u(\cdot) = \{u(t), t \in T\}$ называется допустимым, если вдоль него выполняются все ограничения задачи (2). Управление $u^\varepsilon(\cdot)$ назовем ε -оптимальным, если $J(u) - J(u^0) \leq \varepsilon$, где $u^0(\cdot)$ — оптимальное управление.

2. Опора. Критерий опорности. Совокупность множеств $Q_{\text{оп}} = \{I_{\text{оп}}(t+1), U_{\text{оп}}(t)\}$, $t \in T$ назовем опорой задачи (2), если система

$$\begin{aligned}x(t+1) &= x(t) - B(t)u(t), \quad x(0) = 0, \\x_i(t+1) &= 0, \quad i \in I \setminus I_*(t+1), \\u_{ij}(t) &= 0, \quad (i, j) \in U(t) \setminus U_*(t), \quad t \in T,\end{aligned}\tag{3}$$

при

$$I_*(t+1) = I_{\text{оп}}(t+1), \quad U_*(t) = U_{\text{оп}}(t), \quad t \in T,\tag{4}$$

имеет только тривиальное решение $u_{ij}(t) \equiv 0$, $(i, j) \in U_{\text{оп}}(t)$, $t \in T$, но имеет нетривиальное решение $u_{ij}(t) \not\equiv 0$, $(i, j) \in U_*(t)$, $t \in T$, при

$$\begin{aligned}I_*(t) &= I_{\text{оп}}(t), \quad t = \overline{1, t_1}, \quad t \neq t_0, \quad I_*(t_0) = I_{\text{оп}}(t_0) \cup i_0, \\U_*(t) &= U_{\text{оп}}(t), \quad t \in T,\end{aligned}\tag{5}$$

где $i_0 \in I_{\text{оп}}(t_0)$, t_0 — любой момент из $\{1, \dots, t_1\}$.

Приведем алгоритмический критерий опорности. Пусть $Q_{\text{оп}} = \{I_{\text{оп}}(t+1), U_{\text{оп}}(t)\}$, $t \in T$ — опора задачи (2). Для $t=0$ положим $\bar{U}_{\text{оп}}(0) = U_{\text{оп}}(0)$. Из определения опоры следует, что сеть $S_{\text{оп}}(0) = \{I, \bar{U}_{\text{оп}}(0)\}$ не содержит циклов, так как в противном случае система (3), (4) имела бы нетривиальное решение.

Пусть $S_{\text{оп}}^k(0) = \{I^k(0), \bar{U}_{\text{оп}}^k(0)\}$, $k = \overline{1, p(0)}$, — компоненты связности сети $S_{\text{оп}}(0)$. Покажем, что $I^k(0) \cap I_{\text{оп}}(1) \neq \emptyset$, $k = \overline{1, p(0)}$. Действительно, предположение об обратном ведет к существованию такого узла $i_0 \in I^{k_0}(0)$, $I^{k_0}(0) \cap I_{\text{оп}}(1) = \emptyset$, что система (3), (5) при $t_0 = 1$ имеет только тривиальное решение, что противоречит определению опоры. Следовательно, при каждом k , $k = \overline{1, p(0)}$, в множестве $I^k(0)$ найдется хотя бы один узел из $I_{\text{оп}}(1)$.

В множестве $I^k(0)$ выделим любой узел $i_k(1)$, принадлежащий $I_{\text{оп}}(1)$, и обозначим $I_{\text{оп}}^+(1) = \{i_k(1), k = \overline{1, p(0)}\}$. По построению, пара $Q_{\text{оп}}^R(0) = \{I_{\text{оп}}^+(1), \bar{U}_{\text{оп}}(0)\}$ обладает свойством:

$$\left\{ \begin{array}{l} \text{сеть } S_{\text{оп}}(t) = \{I, \bar{U}_{\text{оп}}(t)\} \text{ не содержит циклов;} \\ \text{в каждой компоненте связности этой сети имеется} \\ \text{ровно по одному узлу из множества } I_{\text{оп}}^+(t+1) \\ \text{при } t = 0. \end{array} \right. \tag{6}$$

Назовем такую пару $Q_{\text{оп}}^R(0)$ рабочей опорой в момент $t = 0$.

Пусть построены рабочие опоры $Q_{\text{оп}}^R(t) = \{I_{\text{оп}}^+(t+1), \bar{U}_{\text{оп}}(t)\}$ в моменты $t = 0, 1, \dots, s$ ($s < t_1 - 1$), для кото-

рых выполняются свойства (6) в указанные моменты t . Построим рабочую опору для момента $t = s + 1$. По построению, $I_{\text{оп}}^+(s+1) = \{i_k(s+1), k = \overline{1, p(s)}\}$, где $i_k(s+1)$ — любой узел из множества $I^k(s) \cap I_{\text{оп}}(s+1)$, $S_{\text{оп}}^k(s) = \{I^k(s), \bar{U}_{\text{оп}}^k(s)\}$ — k -я компонента связности сети $S_{\text{оп}}(s)$, $k = \overline{1, p(s)}$. Каждому узлу $j \in I^k(s) \cap I_{\text{оп}}(s+1)$, где $I_{\text{оп}}(s+1) = I_{\text{оп}}(s+1) \setminus I_{\text{оп}}^+(s+1)$, поставим в соответствие дугу $(i_k(s+1), j)$, $k = \overline{1, p(s)}$. Объединение множества дуг $U_{\text{оп}}(s+1)$ с множеством $U^*(s+1) = \{(i_k(s+1), j), j \in I^k(s) \cap I_{\text{оп}}(s+1), k = \overline{1, p(s)}\}$ обозначим через $\bar{U}_{\text{оп}}(s+1)$. Из определения опоры $Q_{\text{оп}}$ следует, что в сети $S_{\text{оп}}(s+1) = \{I, \bar{U}_{\text{оп}}(s+1)\}$ нет циклов и в каждой ее компоненте связности $S_{\text{оп}}^k(s+1) = \{I^k(s+1), \bar{U}_{\text{оп}}^k(s+1)\}$, $k = \overline{1, p(s+1)}$, имеются узлы из $I_{\text{оп}}(s+2)$. В множестве $I^k(s+1)$ выделим узел $i_k(s+2) \in I_{\text{оп}}(s+2)$ и обозначим $I_{\text{оп}}^+(s+2) = \{i_k(s+2), k = \overline{1, p(s+1)}\}$. Построенная опора $Q_{\text{оп}}^R(s+1) = \{I_{\text{оп}}^+(s+2), \bar{U}_{\text{оп}}(s+1)\}$ обладает свойством (6) при $t = s + 1$, т. е. является рабочей опорой в момент $t = s + 1$.

Для рабочей опоры $Q_{\text{оп}}^R(t_1 - 1) = \{I_{\text{оп}}^+(t_1), \bar{U}_{\text{оп}}(t_1 - 1)\}$ в момент $t_1 - 1$ выполняется равенство $I_{\text{оп}}^+(t_1) = I_{\text{оп}}(t_1)$, ибо в противном случае система (3), (4) имела бы нетривиальное решение.

Таким образом, по опоре $Q_{\text{оп}}$ задачи (2) алгоритмически построены рабочие опоры $Q_{\text{оп}}^R(t)$, $t \in T$, $\bar{U}_{\text{оп}}(0) = U_{\text{оп}}(0)$, $I_{\text{оп}}^+(t_1) = I_{\text{оп}}(t_1)$. Верно и обратное утверждение: если для совокупности множеств $\{I_{\text{оп}}(t+1), U_{\text{оп}}(t)\}$, $t \in T$, $(I_{\text{оп}}(t+1) \subset I, U_{\text{оп}}(t) \subset U(t))$ описанный алгоритм строит рабочие опоры $Q_{\text{оп}}^R(t) = \{I_{\text{оп}}^+(t+1), \bar{U}_{\text{оп}}(t)\}$, $t \in T$, $\bar{U}_{\text{оп}}(0) = U_{\text{оп}}(0)$, $I_{\text{оп}}^+(t_1) = I_{\text{оп}}(t_1)$ со свойствами (6), то рассматриваемая совокупность является опорой задачи (2). Следовательно, справедлива

Теорема 1 (первый критерий опорности). Совокупность множеств $Q_{\text{оп}} = \{\{I_{\text{оп}}(t+1), U_{\text{оп}}(t)\}, t \in T\}$ ($I_{\text{оп}}(t+1) \subset I, U_{\text{оп}}(t) \subset U(t)$) является опорой задачи (2) тогда и только тогда, когда описанный выше алгоритм выделяет из нее рабочие опоры $Q_{\text{оп}}^R(t)$, $t \in T$. При этом $\bar{U}_{\text{оп}}(0) = U_{\text{оп}}(0)$, $I_{\text{оп}}^+(t_1) = I_{\text{оп}}(t_1)$.

З а м е ч а н и я. 1. Пусть имеются такие множества дуг $U^*(t)$ и узлов $I^*(t+1)$, что $U^*(t)$ не содержит циклов и в каждой компоненте сети $\{I, U^*(t)\}$ есть хотя бы один узел из $I^*(t+1)$. Пару $\{\tilde{U}(t), \tilde{I}(t+1)\}$, $\tilde{U}(t) \subset U^*(t)$, $\tilde{I}(t+1) \subset I^*(t+1)$, $|\tilde{U}(t)| + |\tilde{I}(t+1)| = |I|$, удовлетворяющую свойствам (6), можно построить за счет удаления, во-первых, «лишних» узлов из $I^*(t+1)$: $\tilde{U}(t) = U^*(t)$, $\tilde{I}(t+1) \subset I^*(t+1)$, во-вторых, «лишних» дуг из $U^*(t)$: $\tilde{U}(t) \subset U^*(t)$, $\tilde{I}(t+1) = I^*(t+1)$, и, в-третьих, как «лишних» узлов из $I^*(t+1)$, так и «лишних» дуг из $U^*(t)$: $\tilde{U}(t) \subset U^*(t)$, $\tilde{I}(t+1) \subset I^*(t+1)$. В основу описанного алгоритма построения рабочих опор положен первый способ. Аналогичные алгоритмы можно построить, используя два других способа.

2. Критерий опорности в задаче (1) ((2)) можно сформулировать в другом виде. Нетрудно проверить, что динамическая транспортная задача (1) на нестационарной сети $S(t) = \{I, U(t)\}$, $t \in T$, эквивалентна классической транспортной задаче на стационарной сети $\overset{\vee}{S} = \{\overset{\vee}{I}, \overset{\vee}{U}\}$, где $\overset{\vee}{I} = \{i(t), i \in I, t \in T\} \cup s$, $\overset{\vee}{U} = \{(i(t), j(t)), (i, j) \in U(t), t \in T\} \cup \{(i(t), i(t+1)), i \in I, t = \overline{1, t_1-1}\} \cup \{(i(t_1), s), i \in I\}$. Тогда из критерия опорности в классической транспортной задаче следует, что (второй критерий опорности) $Q_{\text{оп}}\{\{I_{\text{оп}}(t+1), U_{\text{оп}}(t)\}, t \in T\}$ является опорой нестационарной сети $S(t)$, $t \in T$, тогда и только тогда, когда множество $\overset{\vee}{U}_{\text{оп}} = \{(i(t), j(t)), (i, j) \in U_{\text{оп}}(t), t \in T\} \cup \{(i(t), i(t+1)), i \in I_{\text{оп}}(t), t = \overline{1, t_1-1}\} \cup \{(i(t_1), s), i \in I_{\text{оп}}(t_1)\}$ является деревом сети $\overset{\vee}{S}$.

Сравним первый и второй критерии опорности. Очевидно, что формулировка у второго критерия проще. Однако при достаточно большом t_1 его проверка с помощью ЭВМ может оказаться неосуществимой, в то время как проверка первого критерия легко реализуема. Действительно, для проверки второго критерия в оперативную память ЭВМ надо записать всю сеть $\overset{\vee}{S}(t_1|I|+1$ узлов, $\sum_{t \in T} |U(t)| + t_1|I|$ дуг) и все множество $\overset{\vee}{U}_{\text{оп}}(t_1|I|$ дуг). При проверке же первого критерия в каждый момент времени t в оперативной памяти достаточно иметь сеть $S(t)$ ($|I|$ узлов, $|U(t)|$ дуг) и пару множеств $\{I_{\text{оп}}(t+1), U_{\text{оп}}(t)\}$ ($|I_{\text{оп}}(t+1)| + |U_{\text{оп}}(t)| \leq 2|I| - 1$).

Множество рабочих опор $Q_{\text{оп}}^R = \{Q_{\text{оп}}^R(t), t \in T\}$ назовем динамической опорой задачи (2).

Пару $\{u(\cdot), Q_{\text{оп}}\}$ из допустимого управления и опоры назовем опорным управлением. Опорное управление счи-

таётся невырожденным, если $0 < u_{ij}(t) < d_{ij}(t)$, $(i, j) \in U_{\text{оп}}(t)$; $0 < x_i(t+1) < d_i(t+1)$, $i \in I_{\text{оп}}(t+1)$, $t \in T$.

3. Принцип ε -максимума. Пусть в задаче оптимального управления (2) известно начальное опорное управление $\{u(\cdot), Q_{\text{оп}}\}$. Запишем сопряженную систему из теории оптимального управления

$$\psi(t-1) = \psi(t) + c_x(t) + \xi(t), \quad \psi(t_1) = 0, \quad t \in T, \quad (7)$$

где $\xi(t) = \{\xi_i(t), i \in I\}$, $t = \overline{1, t_1}$, — функция скачков. Ясно, что решение $\psi(t)$, $t \in T^1$, сопряженной системы однозначно определяется функцией скачков, которая играет роль управления в сопряженной системе. Построим функцию скачков $\xi(t)$, $t \in T$, соответствующую опоре $Q_{\text{оп}}$, и назовем ее опорной функцией скачков. Положим

$$\xi_i(t) = 0, \quad i \in I_{\text{оп}}(t), \quad t = \overline{1, t_1}. \quad (8)$$

Остальные $\xi_i(t)$, $i \in I \setminus I_{\text{оп}}(t)$, $t = \overline{1, t_1}$, однозначно найдутся из условий

$$\psi'(t) b^{ij}(t) = c_{ij}(t), \quad (i, j) \in U_{\text{оп}}(t), \quad t \in T. \quad (9)$$

(Методы построения опорной функции скачков (8), (9) и соответствующего ей решения сопряженной системы (7) описаны ниже.)

Опору $Q_{\text{оп}}$ назовем согласованной с управлением $u(\cdot)$, если для траектории $x(t)$, $t = \overline{1, t_1}$, соответствующей управлению $u(\cdot)$, и опорных скачков $\xi(t)$, $t = \overline{1, t_1}$, выполняются соотношения

$$\begin{aligned} \xi_i(t) &\geq 0 \text{ при } x_i(t) = d_i(t), \quad \xi_i(t) \leq 0 \text{ при } x_i(t) = 0, \\ \xi_i(t) &= 0 \text{ при } 0 < x_i(t) < d_i(t), \quad i \in I, \quad t = \overline{1, t_1}. \end{aligned} \quad (10)$$

Опорное управление $\{u(\cdot), Q_{\text{оп}}\}$ считается согласованным, если опора $Q_{\text{оп}}$ согласована с управлением $u(\cdot)$.

Можно показать, что для опорного управления $\{u(\cdot), Q_{\text{оп}}\}$ справедливо неравенство

$$\begin{aligned} \sum_{t \in T} (c'_x(t+1) x(t+1) + c'_u(t) u(t)) - \sum_{t \in T} (c'_x(t+1) x^0(t+1) + \\ + c'_u(t) u^0(t)) \leq \sum_{t \in T} (\varepsilon_u(t) + \varepsilon_x(t+1)), \end{aligned} \quad (11)$$

где $\varepsilon_u(t) = \max_{u \in V(t)} (\psi'(t) B(t) - c'_u(t)) u - (\psi'(t) B(t) - c'_u(t)) u(t)$;
 $\varepsilon_x(t) = \max_{0 \leq x \leq d_x(t)} \xi'(t) x - \xi'(t) x(t)$; $x(t)$, $t = \overline{1, t_1}$, — траекто-

рия системы (2), соответствующая управлению $u(\cdot)$; $u^0(\cdot)$ — оптимальное управление; $\psi(t)$, $t \in T$, — решение сопряженной системы (7), соответствующее опорным скачкам.

Ясно, что $\varepsilon_x(t) = 0$, $t = \overline{1, t_1}$, в том и только том случае, когда опора $Q_{\text{оп}}$ согласована с управлением $u(\cdot)$.

Величину $\beta = \beta(u(\cdot), Q_{\text{оп}}) = \sum_{t \in T} (\varepsilon_u(t) + \varepsilon_x(t+1))$ назовем оценкой субоптимальности опорного управления $\{u(\cdot), Q_{\text{оп}}\}$. Из неравенства (11) следует

Теорема 2 (принцип ε -максимума). Для ε -оптимальности управления $u^\varepsilon(\cdot)$ и соответствующей ему траектории $x^\varepsilon(t)$, $t = \overline{1, t_1}$, в задаче (2) необходимо и достаточно существование такой опоры $Q_{\text{оп}}$, что выполняются условия квазимаксимума по управлению и траектории:

$$(\psi'(t) B(t) - c'_u(t)) u^\varepsilon(t) = \max_{u \in V(t)} (\psi'(t) B(t) - c'_u(t)) u, \quad t \in T,$$

$$\xi'(t) x^\varepsilon(t) = \max_{0 \leq x \leq d_x(t)} \xi'(t) x - \varepsilon_x(t), \quad t = \overline{1, t_1},$$

$$\varepsilon_u(t) \geq 0, \quad \varepsilon_x(t+1) \geq 0, \quad t \in T,$$

$$\sum_{t \in T} (\varepsilon_u(t) + \varepsilon_x(t+1)) \leq \varepsilon,$$

где $\xi(t)$, $t = \overline{1, t_1}$, — функция скачков, построенная по опоре $Q_{\text{оп}}$; $\psi(t)$, $t \in T$, — соответствующее ей решение сопряженной системы (7).

Теорема 3 (принцип максимума). Для оптимальности опорного управления $\{u(\cdot), Q_{\text{оп}}\}$ в задаче (2) достаточно, а в случае невырожденности и необходимо, чтобы оно было согласованным и выполнялось условие максимума:

$$\begin{aligned} & (\psi'(t) B(t) - c'_u(t)) u^0(t) = \\ & = \max_{u \in V(t)} (\psi'(t) B(t) - c'_u(t)) u, \quad t \in T, \end{aligned} \quad (12)$$

где $\psi(t)$, $t \in T$, — решение системы (7), в которой векторы $\xi(t)$, $t = \overline{1, t_1}$, построены по опоре $Q_{\text{оп}}$.

Из теоремы 3 следует, что условия согласования (10) являются необходимыми для оптимальности опорного управления. Поэтому в дальнейшем будем рассматривать только согласованные опорные управления $\{u(\cdot), Q_{\text{оп}}\}$.

Опишем алгоритм построения опорных скачков $\xi(t)$, $t \in \overline{1, t_1}$, и соответствующего им решения сопряженной си-

стемы (7), т. е. векторов $\psi(t)$, $\xi(t+1)$, $t \in T$, удовлетворяющих соотношениям (7) — (9).

Согласно алгоритму п. 2 построим динамическую опору $Q_{\text{оп}}^R$ и вычислим коэффициенты $\bar{c}_{ij}(t)$, $(i, j) \in \bar{U}_{\text{оп}}(t) = U_{\text{оп}}(t) \cup U^*(t)$, $t \in T$:

$$\begin{aligned}\bar{c}_{ij}(t) &= c_{ij}(t), \quad (i, j) \in U_{\text{оп}}(t); \\ \bar{c}_{ij}(t) &= -c_i(t) + c_j(t) + \pi(i, j, \bar{U}_{\text{оп}}(t-1)), \quad (i, j) \in U^*(t), \quad t = 0, 1, \dots, t_1 - 1.\end{aligned}\quad (13)$$

Здесь $\pi(i, j, \bar{U}_{\text{оп}}(t-1))$ — длина пути из узла i в узел j по дугам множества $\bar{U}_{\text{оп}}(t-1)$, длины которых равны $\bar{c}_{ij}(t-1)$. Положим $\psi_i(t_1 - 1) = c_i(t_1)$, $i \in I_{\text{оп}}^+(t_1) = I_{\text{оп}}(t_1)$. Поскольку множество $\bar{U}_{\text{оп}}(t_1 - 1)$ не содержит циклов и в каждой компоненте связности сети $S_{\text{оп}}(t_1 - 1)$ есть узел из $I_{\text{оп}}^+(t_1)$, то числа $\psi_i(t_1 - 1)$, $i \in I \setminus I_{\text{оп}}^+(t_1)$, легко найдутся из условий

$$\psi_i(t_1 - 1) - \psi_j(t_1 - 1) = \bar{c}_{ij}(t_1 - 1), \quad (i, j) \in \bar{U}_{\text{оп}}(t_1 - 1), \quad (14)$$

по правилам классического метода потенциалов.

Для $t = t_1 - 2, t_1 - 3, \dots, 0$ векторы $\psi(t)$ строятся аналогично из условий

$$\begin{aligned}\psi_i(t) &= \psi_i(t + 1) + c_i(t + 1), \quad i \in I_{\text{оп}}^+(t + 1); \\ \psi_i(t) - \psi_j(t) &= \bar{c}_{ij}(t), \quad (i, j) \in \bar{U}_{\text{оп}}(t).\end{aligned}\quad (15)$$

Зная $\psi(t)$, $t \in T$, легко построить $\xi(t) = \psi(t-1) - \psi(t) - c_x(t)$, $t = \bar{1}, t_1$. Нетрудно проверить, что построенные векторы $\psi(t)$, $\xi(t+1)$, $t \in T$, удовлетворяют условиям (7) — (9).

4. Итерация. Пусть для начального согласованного опорного управления *) $\{u(\cdot), Q_{\text{оп}}\}$ принципы максимума и ε -максимума не выполняются, т. е. $\beta(u(\cdot), Q_{\text{оп}}) > \varepsilon$, где ε — заданная точность приближения к оптимальному управлению по значению целевой функции. Построим новое согласованное опорное управление $\{\bar{u}(\cdot), \bar{Q}_{\text{оп}}\}$, для которого $\beta(\bar{u}(\cdot), \bar{Q}_{\text{оп}}) \leq \beta(u(\cdot), Q_{\text{оп}})$. В основу итерации положим принципы, аналогичные приведенным в § 1 гл. 1. Начнем с построения нового управления $\bar{u}(\cdot) = u(\cdot) + \Delta u(\cdot)$, где приращение имеет вид $\Delta u(\cdot) =$

*) Если начальная опора не известна, то в качестве начальной можно взять опору $Q_{\text{оп}}\{I_{\text{оп}}(t+1) = I, U_{\text{оп}}(t) = \emptyset\}$, $t \in T$, которая согласована с любым допустимым управлением.

$=\Theta v(\cdot)$, $\Theta \geq 0$. При построении $\Delta u(\cdot) = \Theta v(\cdot)$ будем исходить из условия максимума (12).

Введем обозначения: $\Delta'(t) = \psi'(t) B(t) - c'_u(t) = \{\Delta_{ij}(t) = \psi_i(t) - \psi_j(t) - c_{ij}(t), (i, j) \in U(t)\}$. Компоненты $v_{ij}(t)$ вектора $v(t)$, для которых $\Delta_{ij}(t) \neq 0$, $t \in T$, однозначно найдутся из условия (12):

$$\begin{aligned} v_{ij}(t) &= d_{ij}(t) - u_{ij}(t) \text{ при } \Delta_{ij}(t) > 0, \\ v_{ij}(t) &= -u_{ij}(t) \text{ при } \Delta_{ij}(t) < 0, \\ (i, j) &\in U(t), t \in T. \end{aligned} \quad (16)$$

Значения остальных компонент $v_{ij}(t)$, $(i, j) \in U^0(t) = \{(i, j) \in U(t) : \Delta_{ij}(t) = 0\}$, $t = 0, t_1 - 1$, не влияют на условия максимума (12). Заметим, что, по построению, $U_{\text{оп}}(t) \subset U^0(t)$, $t \in T$. Компоненты $v_{ij}(t)$, $(i, j) \in U^0(t)$, $t \in T$, найдем следующим образом.

Для определенности положим

$$v_{ij}(t) = 0, (i, j) \in U^0(t) \setminus U_{\text{оп}}(t), t \in T \quad (17)$$

(можно положить $v_{ij}(t) = \alpha_{ij}$, где $\alpha_{ij} \in [-u_{ij}(t), d_{ij}(t) - u_{ij}(t)]$). Опорные компоненты $v_{ij}(t)$, $t \in U_{\text{оп}}(t)$, $t \in T$, найдем из системы

$$\begin{aligned} l(t+1) &= l(t) - B(t) v(t), l_i(t+1) = 0, \\ i \in I_{\text{н}}(t+1) &= I \setminus I_{\text{оп}}(t+1), t \in T, l(0) = 0. \end{aligned} \quad (18)$$

Опишем один из способов решения системы (16) — (18). Построим векторы $b(t) = \{b_i(t), i \in I\}$, $t \in T$, по правилу:

$$b_i(0) = -B_{i\text{н}}(0) v_{\text{н}}(0), i \in I,$$

$$b_i(t) = \begin{cases} -B_{i\text{н}}(t) v_{\text{н}}(t), & \text{если } i \in \overline{I_{\text{оп}}^+}(t), \\ -B_{i\text{н}}(t) v_{\text{н}}(t) + \sum_{j \in I^k} b_j(t-1), & \text{если} \\ i = i_k(t) \in I_{\text{оп}}^+(t), & i \in I, t = 1, 2, \dots, t_1 - 1, \end{cases}$$

где $v_{\text{н}}(t) = \{v_{ij}(t), (i, j) \in U_{\text{н}}(t) = U(t) \setminus U_{\text{оп}}(t)\}$, $B_{i\text{н}}(t)$ — i -я строка подматрицы $B_{\text{н}}(t) = \{b^{ij}(t), (i, j) \in U_{\text{н}}(t)\}$.

При $t = t_1 - 1$ рассмотрим сеть

$$S_{\text{оп}}^*(t) = \{I \cup s, \overline{U_{\text{оп}}}(t) \cup (i, s), i \in I_{\text{оп}}^+(t+1)\} \quad (19)$$

с интенсивностями узлов *)

*) Поскольку $I_{\text{оп}}^-(t_1) = \emptyset$, то из (20) следует, что $\overline{b}_i(t_1 - 1) = b_i(t_1 - 1)$, $i \in I$, $\overline{b}_s(t_1 - 1) = 0$.

$$\bar{b}_i(t) = \begin{cases} b_i(t), & \text{если } i \in I \setminus I_{\text{оп}}^-(t+1), \\ b_i(t) - l_i(t+1), & \text{если } i \in I_{\text{оп}}^-(t+1), \\ \sum_{j \in I_{\text{оп}}^-(t+1)} l_j(t+1), & \text{если } i = s, \end{cases} \quad (20)$$

и найдем на ней псевдопоток

$$\kappa^*(t) = \{\kappa_{ij}^*(t), (i, j) \in \bar{U}_{\text{оп}}(t)\} \cup \{(i, s), i \in I_{\text{оп}}^+(t+1)\}. \quad (21)$$

В силу того, что сеть $S_{\text{оп}}^*(t)$ является деревом, псевдопоток $\kappa^*(t)$ найдется однозначно по правилам двойственного опорного метода решения классической транспортной задачи.

Положим

$$\begin{aligned} v_{ij}(t) &= \kappa_{ij}^*(t), \quad (i, j) \in U_{\text{оп}}(t), \\ l_i(t+1) &= \kappa_{is}^*(t), \quad i \in I_{\text{оп}}^+(t+1), \\ l_j(t) &= \kappa_{ij}^*(t), \quad j \in \{j : (i, j) \in U^*(t)\} = I_{\text{оп}}^-(t). \end{aligned} \quad (22)$$

Повторим операции (19) — (22) для $t = t_1 - 2, t_1 - 3, \dots, 0$. Можно проверить, что построенные при этом числа (22) вместе с (16), (17) удовлетворяют системе (18). Ясно, что приращение $\Delta x(\cdot)$ траектории $x(\cdot)$, соответствующее приращению $\Delta u(\cdot) = \Theta v(\cdot)$ управления $u(\cdot)$, равно $\Delta x(\cdot) = \Theta l(\cdot)$, где $l(\cdot)$ найдено согласно (18).

Найдем максимальный шаг $\Theta_0 \geq 0$, при котором управление $\bar{u}(\cdot) = u(\cdot) + \Delta u(\cdot)$ будет допустимым:

$$\Theta_0 = \min \{1, \Theta_{i_0 j_0}(t_0), \Theta_{i_*}(t_*)\}, \quad (23)$$

где $\Theta_{i_0 j_0}(t_0) = \min \Theta_{ij}(t)$, $(i, j) \in U_{\text{оп}}(t)$, $t \in T$, $\Theta_{ij}(t) = (d_{ij}(t) - u_{ij}(t)) / v_{ij}(t)$ при $v_{ij}(t) > 0$; $\Theta_{ij}(t) = -u_{ij}(t) / v_{ij}(t)$ при $v_{ij}(t) < 0$; $\Theta_{ij}(t) = \infty$ при $v_{ij}(t) = 0$; $\Theta_{i_*}(t_*) = \min \Theta_i(t)$, $i \in I_{\text{оп}}(t)$, $t = 1, t_1$; $\Theta_i(t) = (d_i(t) - x_i(t)) / l_i(t)$ при $l_i(t) > 0$; $\Theta_i(t) = -x_i(t) / l_i(t)$ при $l_i(t) < 0$; $\Theta_i(t) = \infty$ при $l_i(t) = 0$.

Оценка субоптимальности опорного управления $\{\bar{u}(\cdot), Q_{\text{оп}}\}$ с $\bar{u}(\cdot) = u(\cdot) + \Theta_0 v(\cdot)$ равна $\beta(\bar{u}(\cdot), Q_{\text{оп}}) = (1 - \Theta_0) \beta(u(\cdot), Q_{\text{оп}})$. Если $\beta(\bar{u}(\cdot), Q_{\text{оп}}) \leq \varepsilon$, то процесс решения задачи (2) прекращаем на ε -оптимальном управлении $\bar{u}(\cdot)$. В противном случае переходим к операциям по замене опоры $Q_{\text{оп}}$.

Пусть $\beta(\bar{u}(\cdot), Q_{\text{оп}}) > \varepsilon$. Возможны случаи: а) $\Theta_0 = \Theta_{i_0 j_0}(t_0) < 1$; б) $\Theta_0 = \Theta_{i_*}(t_*) < 1$.

Рассмотрим случай а). Положим $c_i(t)=0$, $i \in I_{\text{оп}}(t)$, $t=\overline{1, t_1}$; $c_{i_0 j_0}(t_0)=\text{sign } v_{i_0 j_0}(t_0)$, $c_{ij}(t)=0$, $(i, j) \neq (i_0, j_0)$, $t \neq t_0$, $(i, j) \in U_{\text{оп}}(t)$, $t \in T$, и согласно (13) найдем коэффициенты $\bar{c}_{ij}(t)$, $(i, j) \in \bar{U}_{\text{оп}}(t)$, $t \in T$.

Подсчитаем векторы $\mu(t) = \{\mu_i(t), i \in I\}$, $t \in T$, из следующих рекуррентных соотношений:

$$\begin{aligned} \mu_i(t) &= \mu_i(t+1) + c_i(t+1), \quad i \in I_{\text{оп}}^+(t+1); \\ \mu_i(t) - \mu_j(t) &= \bar{c}_{ij}(t), \quad (i, j) \in \bar{U}_{\text{оп}}(t), \\ t &= t_1 - 1, t_1 - 2, \dots, 0; \quad \mu(t_1) = 0. \end{aligned} \quad (24)$$

Для того, чтобы найти элемент, вводимый в опору, вычислим

$$\sigma_0 = \min \{ \sigma_{\bar{i}_*}(\bar{t}_*), \sigma_{\bar{i}_0 \bar{j}_0}(\bar{t}_0) \}, \quad (25)$$

где $\sigma_{\bar{i}_*}(\bar{t}_*) = \min \sigma_i(t)$, $i \in I_{\text{н}}(t)$, $t = \overline{1, t_1}$; $\sigma_{\bar{i}_0 \bar{j}_0}(\bar{t}_0) = \min \sigma_{ij}(t)$, $(i, j) \in U_{\text{н}}(t)$, $t \in T$,

$$\sigma_{ij}(t) = \begin{cases} -\Delta_{ij}(t)/\gamma_{ij}(t), & \text{если } \Delta_{ij}(t) \gamma_{ij}(t) < 0, \\ 0, & \text{если } \Delta_{ij}(t) = 0, \gamma_{ij}(t) > 0, \bar{u}_{ij}(t) \neq d_{ij}(t) \text{ или} \\ \Delta_{ij}(t) = 0, \gamma_{ij}(t) < 0, \bar{u}_{ij}(t) \neq 0, \\ \infty & \text{в остальных случаях;} \end{cases}$$

$$\sigma_i(t) = \begin{cases} -\xi_i(t) \setminus \gamma_i(t), & \text{если } \xi_i(t) \gamma_i(t) < 0, \\ 0, & \text{если } \xi_i(t) = 0, \gamma_i(t) > 0, \bar{x}_i(t) \neq d_i(t) \text{ или} \\ \xi_i(t) = 0, \gamma_i(t) < 0, \bar{x}_i(t) \neq 0, \\ \infty & \text{в остальных случаях;} \end{cases}$$

$\gamma_{ij}(t) = \mu_i(t) - \mu_j(t)$, $\gamma_i(t) = \mu_i(t-1) - \mu_i(t)$. Если $\sigma_0 = \sigma_{\bar{i}_*}(\bar{t}_*)$, то новую опору $\bar{Q}_{\text{оп}}$ строим следующим образом: из множества $U_{\text{оп}}(t_0)$ удаляем дугу (t_0, j_0) , к множеству $I_{\text{оп}}(\bar{t}_*)$ добавляем узел \bar{i}_* . Если $\sigma_0 = \sigma_{\bar{i}_0 \bar{j}_0}(\bar{t}_0)$, то при построении новой опоры $\bar{Q}_{\text{оп}}$ из множества $U_{\text{оп}}(t_0)$ удаляем дугу (i_0, j_0) , в множество $U_{\text{оп}}(\bar{t}_0)$ вводим дугу (\bar{i}_0, \bar{j}_0) . Легко проверить, что новое опорное управление $\{\bar{u}(\cdot), \bar{Q}_{\text{оп}}\}$ будет согласованным и $\beta(\bar{u}(\cdot), \bar{Q}_{\text{оп}}) = (1 - \Theta_0) \times \times \beta(u(\cdot), Q_{\text{оп}}) - \sigma_0 |v_{i_0 j_0}(t_0)| \leq \beta(u(\cdot), Q_{\text{оп}})$.

Рассмотрим случай б): $\Theta_0 = \Theta_{i_*}(t_*) < 1$. Положим $c_{i_*}(t_*) = \text{sign } l_{i_*}(t_*)$, $c_i(t) = 0$, $i \neq i_*$, $t \neq t_*$, $i \in I_{\text{оп}}(t)$, $t = \overline{1, t_1}$; $c_{ij}(t) = 0$, $(i, j) \in U_{\text{оп}}(t)$, $t \in T$, и по правилам (13)

найдем коэффициенты $\bar{c}_{ij}(t)$, $(i, j) \in \bar{U}_{\text{оп}}(t)$, $t \in T$. Используя найденные коэффициенты, построим векторы $\mu(t)$, $t \in T$, и число σ_0 .

При $\sigma_0 = \sigma_{\bar{i}_*}(\bar{t}_*)$ новая опора $\bar{Q}_{\text{оп}}$ получается из старой удалением узла \bar{i}_* из множества $I_{\text{оп}}(\bar{t}_*)$ и добавлением узла \bar{i}_* к множеству $I_{\text{оп}}(\bar{t}_*)$. Если $\sigma_0 = \sigma_{\bar{i}_0 \bar{j}_0}(\bar{t}_0)$, то для построения новой опоры $\bar{Q}_{\text{оп}}$ из $I_{\text{оп}}(\bar{t}_*)$ удаляем узел \bar{i}_* и в $U_{\text{оп}}(\bar{t}_0)$ добавляем дугу (\bar{i}_0, \bar{j}_0) . Как и в предыдущем случае, можно показать, что новое опорное управление $\{\bar{u}(\cdot), \bar{Q}_{\text{оп}}\}$ будет согласованным и $\beta(\bar{u}(\cdot), \bar{Q}_{\text{оп}}) = (1 - \Theta_0) \times \times \beta(u(\cdot), Q_{\text{оп}}) - \sigma_0 |l_{i_*}(\bar{t}_*)| \leq \beta(u(\cdot), Q_{\text{оп}})$.

Новую итерацию начинаем с согласованного опорного управления $\{\bar{u}(\cdot), \bar{Q}_{\text{оп}}\}$. Итерацию $\{u(\cdot), Q\} \rightarrow \{\bar{u}(\cdot), \bar{Q}_{\text{оп}}\}$ назовем регулярной, если $\Theta_0 + \sigma_0 \neq 0$. Описанный алгоритм будет конечным, если в процессе решения не встречаются нерегулярные итерации.

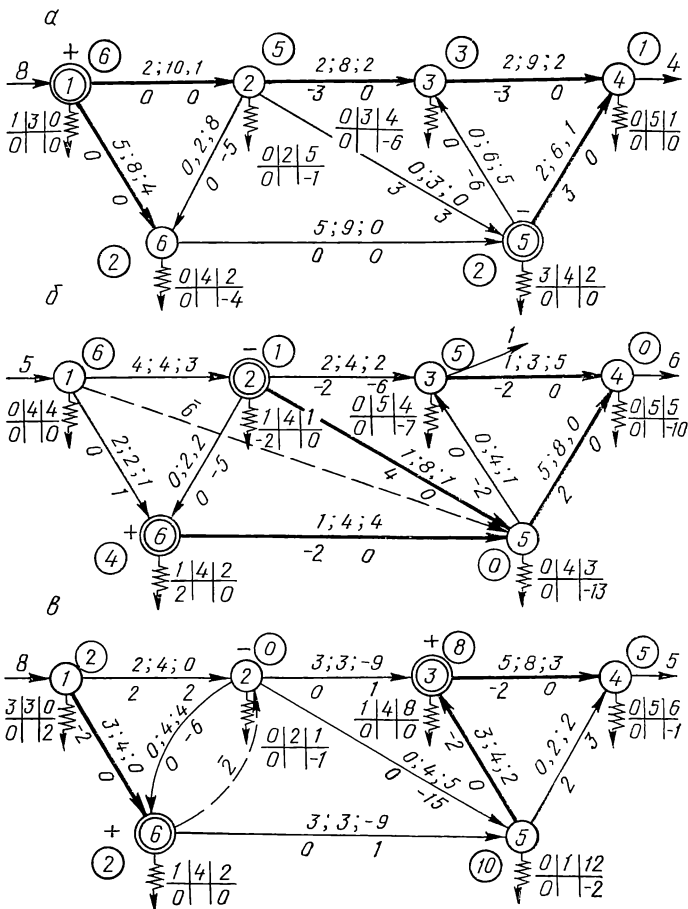
Можно, следуя § 6 гл. 1, построить конечную модификацию метода.

5. Пример. Рассмотрим динамическую сеть, изображенную на рис. 3, где каждому моменту $t \in T = \{0, 1, 2 = t_1 - 1\}$ соответствует своя сеть. На рисунке приняты следующие обозначения: прямые светлые стрелки в узлах означают интенсивности узлов (положительным интенсивностям соответствуют стрелки, входящие в узел, отрицательным — стрелки, выходящие из узла; если стрелка отсутствует, то интенсивность равна нулю); волнистые стрелки соответствуют компонентам состояния системы $x(t+1) = \{x_i(t+1), i \in I\}$. Вся информация, связанная с состоянием $x(t+1)$, записывается около стрелок, причем в верхней строке первое число — $x_i(t+1)$, второе — $d_i(t+1)$, третье — $c_i(t+1)$; информация, связанная с управлением $u(t) = \{u_{ij}(t), (i, j) \in U(t)\}$, записывается над дугами $(i, j) \in U(t)$ сети $S(t) = \{I, U(t)\}$: первое число — $u_{ij}(t)$, второе — $d_{ij}(t)$, третье — $c_{ij}(t)$.

Пусть известно начальное опорное управление $\{u(\cdot), Q_{\text{оп}}(\cdot)\}$, $Q_{\text{оп}}(\cdot) = \{Q_{\text{оп}}(t), t \in T\}$, $Q_{\text{оп}}(t) = \{I_{\text{оп}}(t+1), U_{\text{оп}}(t)\}$. На сети $S(t)$ дуги $(i, j) \in U_{\text{оп}}(t)$ обозначены сплошной жирной линией, узлы $i \in I_{\text{оп}}(t+1)$ — двойным кружком. Проверим, является ли данное управление оптимальным. Для того, чтобы найти решение $\psi(t)$, $t \in T$, сопряженной системы (7), построим согласно алгоритму из п. 3 и формулам (13) динамическую опору $Q_{\text{оп}}^R(\cdot) = \{Q_{\text{оп}}^R(t), t \in T\}$,

$Q_{\text{оп}}^R(t) = \{I_{\text{оп}}^+(t+1), \bar{U}_{\text{оп}}(t)\}, \quad \bar{U}_{\text{оп}}(t) = U_{\text{оп}}(t) \cup U^*(t),$
и найдем коэффициенты $c_{ij}(t), (i, j) \in U^*(t).$

Рассмотрим сеть $S(0)$ (рис. 3, а). Сеть $S_{\text{оп}}(0) = \{I, U_{\text{оп}}(0)\}$ состоит из одной компоненты связности и



Р и с. 3

$I_{\text{оп}}(1) = \{1, 5\}$. В качестве узла $i_1(1)$ возьмем узел $i_1(1) = 1$. Следовательно, $I_{\text{оп}}^+(1) = \{1\}$, $I_{\text{оп}}^-(1) = \{5\}$. Узел 1 помечаем знаком «+», узел 5—знаком «-». Тогда $U^*(1) = \{(1, 5)\}$, $\bar{c}_{15} = \bar{6}$. Переходим к рассмотрению сети $S(1)$ (рис. 3, б).

К опорным дугам $U_{\text{оп}}(1)$ сети $S(1)$ добавим дуги $U^*(1)$, которые на рисунке обозначены штриховой линией*). В результате получим множество $\bar{U}_{\text{оп}}(1) = U_{\text{оп}}(1) \cup U^*(1)$. Сеть $S_{\text{оп}}(1) = \{I, \bar{U}_{\text{оп}}(1)\}$ состоит из одной компоненты связности; $I_{\text{оп}}(2) = \{2, 6\}$. В качестве узла $i_1(2)$ возьмем узел 6. Следовательно, $I_{\text{оп}}^+(2) = \{6\}$, $I_{\text{оп}}^-(2) = \{2\}$. Узел 6 помечаем знаком «+», узел 2—знаком «-». Строим множество дуг $U^*(2) = \{(6, 2)\}$ и находим для них $\bar{c}_{62}(2) = 2$. Переходим к рассмотрению сети $S(2)$ (рис. 3, в).

К дугам $U_{\text{оп}}(2)$ добавим дуги $U^*(2)$, обозначенные на рисунке штриховой линией. Сеть $S_{\text{оп}}(2) = \{I, \bar{U}_{\text{оп}}(2) = U_{\text{оп}}(2) \cup U^*(2)\}$ состоит из двух компонент связности; $I_{\text{оп}}(3) = \{6, 3\}$, причем узлы 6, 3 принадлежат разным компонентам связности. Следовательно, $i_1(3) = 6$, $i_2(3) = 3$, $I_{\text{оп}}^+(3) = I_{\text{оп}}(3)$. Построение динамических опор закончено.

Теперь, последовательно просматривая сети $S(t)$ в обратном порядке и используя построенные динамические опоры $Q_{\text{оп}}^R(t)$, легко найти числа $\psi_i(t)$, $i \in I$, $t \in T$, по формулам (14), (15). Числа $\psi_i(t)$, $i \in I$, записываем в узлах сети $S(t)$, $t \in T$, и обводим кружком. Для всех $t \in T$ подсчитаем числа $\Delta_{ij}(t) = \psi_i(t) - \psi_j(t) - c_{ij}(t)$, $(i, j) \in U_{\text{н}}(t)$, $\xi_i(t+1) = \psi_i(t) - \psi_i(t+1) - c_i(t+1)$, $i \in I_{\text{н}}(t+1)$, и запишем их под числами $c_{ij}(t)$, $(i, j) \in U_{\text{н}}(t)$, $c_i(t+1)$, $i \in I_{\text{н}}(t+1)$, соответственно.

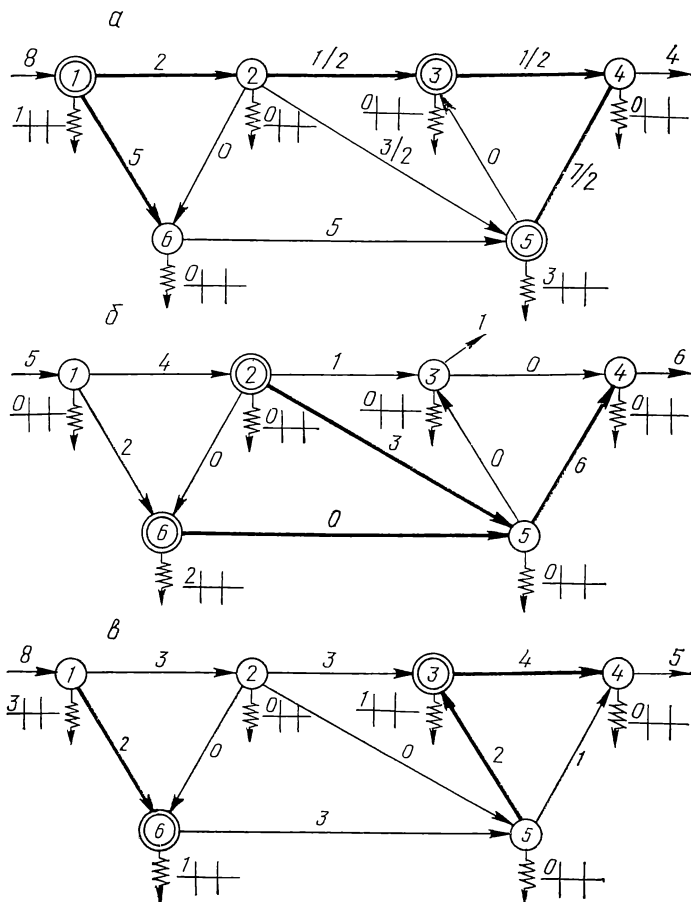
Нетрудно проверить, что начальное опорное управление является согласованным. Оценка субоптимальности равна $\beta(u(\cdot), Q_{\text{оп}}(\cdot)) = 31$. Поскольку $\beta(u(\cdot), Q_{\text{оп}}(\cdot)) > 0$, то ясно, что условия максимума (12) не выполняются. Переходим к операциям по улучшению управления $u(\cdot)$.

Найдем числа $v_{ij}(t)$, $(i, j) \in U_{\text{н}}(t)$, $t \in T$, согласно (16), (17) и запишем их под числами $u_{ij}(t)$. Далее построим векторы $b(t)$, $t \in T$, и по правилам (19)—(22) найдем решение $v_{ij}(t)$, $(i, j) \in U_{\text{оп}}(t)$, $l_i(t+1)$, $i \in I$, $t \in T$, системы (18). Числа $v_{ij}(t)$ запишем на дугах $(i, j) \in U_{\text{оп}}(t)$ под числами $u_{ij}(t)$, числа $l_i(t+1)$, $i \in I$, — около волнистых стрелок под числами $x_i(t+1)$.

По формулам (23) найдем шаг $\Theta_0 = \min \{1, \Theta_{34}(1) =$

) Числа $\bar{c}_{ij}(t)$ записываются над дугами $(i, j) \in U^(t)$.

$=1/2, \Theta_3(1)=1/2\}$. Пусть $\Theta_0=\Theta_{34}(1)$, т. е. $(i_0, j_0)=(3, 4)$, $t_0=1$. Построим новое управление $\bar{u}(\cdot)=u(\cdot)+\Theta_0 v(\cdot)$, которое приведено на рис. 4. Переходим к правилам замены опоры. Имеет место случай а) (см. п. 4): $\Theta_0=\Theta_{i_0 j_0}(t_0)<1$. Положим $c_{34}(1)=-1$, остальные $c_{ij}(t)$,



Р и с. 4

$(i, j) \in U_{\text{оп}}(t)$, $c_i(t+1)$, $i \in I_{\text{оп}}(t+1)$, $t \in T$, равны нулю. Найдем числа $\mu_i(t)=1$, $i \neq 3$, $t \neq 1$, $i=1$, $t \in T$, $\mu_3(1)=0$. Тогда шаг σ_0 равен $\sigma_0 = \sigma_{i_*}^-(t_*) = \sigma_3(1)$. Следовательно, новая опора $\bar{Q}_{\text{оп}}$ получается из старой удалением из

$U_{\text{оп}}(1)$ дуги (3, 4) и добавлением к $I_{\text{оп}}(1)$ узла 3. Для нового опорного управления $\beta(\bar{u}(\cdot), \bar{Q}_{\text{оп}}(\cdot)) = 19/2$. Если $19/2 \leq \varepsilon$, то заканчиваем вычисления. В противном случае переходим к следующей итерации, исходя из нового опорного управления.

§ 7. Вычислительная технология решения больших задач оптимизации

Изложенные варианты адаптивного алгоритма различаются своими основными характеристиками — требованиями к объему памяти и к расходу машинного времени. Ускорение работы программной реализации алгоритма, как правило, достигается ценой увеличения необходимого объема памяти. Несовместимость этих двух требований выдвигает проблему создания программного комплекса, объединяющего несколько различных вариантов реализации метода, каждый из которых применяется в наиболее подходящей для него ситуации.

Возможность выбора наиболее экономной программы часто обеспечивает успешное решение и сложных задач оптимизации, например, задач нелинейного программирования и оптимального управления. Многие итерационные методы решения этих задач используют алгоритмы линейного программирования при построении подходящего направления или при поиске оптимального управления в линеаризованной системе. Размеры задач при этом меняются на итерациях в зависимости от числа активных ограничений или от шага дискретизации непрерывной системы и, следовательно, их эффективное решение возможно при наличии нескольких алгоритмов с различными требованиями к объему памяти.

Итерационный процесс решения большой задачи линейного программирования можно разбить на несколько этапов (например, по размерам накапливаемой опоры) и на каждом этапе применить наиболее подходящий вариант реализации алгоритма. Так, в § 4 гл. 2 изложена вычислительная схема, объединяющая два алгоритма, один из которых работает на первом этапе, когда размер опоры не превышает заданный объем, а другой — с декомпозицией опоры — включается на втором этапе. Аналогично можно построить вычислительную схему с тремя алгоритмами: на первом этапе работает алгоритм с оптимальной заменой опоры (см. § 1 гл. 1), на втором —

алгоритм с пересчетом оценок (см. § 2 гл. 1), на третьем — алгоритм с декомпозицией опоры (см. § 4 гл. 2). Для задач больших размеров на следующих этапах можно предусмотреть применение алгоритмов, использующих внешнюю память, которые также могут различаться, например, по форме хранения информации. Следовательно, для эффективного решения большой задачи потребуется несколько различных вариантов алгоритма, каждый из которых в сравнении с другими обладает максимальной эффективностью на данном этапе.

Таким образом, успешное решение сложных задач возможно только на основе программного комплекса, включающего в себя несколько вариантов алгоритма с разными вычислительными характеристиками.

Программный комплекс самой простой структуры состоит из нескольких процедур, реализующих разные варианты алгоритма, и управляющей программы. В зависимости от структуры ограничений и размеров задачи управляющая программа вызывает наиболее подходящую процедуру, а в процессе решения на новом этапе меняет ее на другую. Одним из недостатков такого подхода является отсутствие гибкости при построении вычислительной схемы — невозможность использовать фрагменты разных процедур при построении комбинированной схемы. Другой недостаток заключается в том, что мало различающиеся программы хранятся в виде автономных процедур, в результате чего их библиотека занимает большой объем памяти.

Программный комплекс более сложной структуры предполагает наличие библиотеки модулей, монитора и сервисных программ, обеспечивающих хранение информации и выдачу результата в удобной форме. В этом случае все операции, входящие в состав разных алгоритмов, оформляются отдельными модулями, каждый из которых может зависеть от входных параметров, определяющих, например, форму хранения информации, режим счета и т. д.

Модульный принцип исключает повторение одинаковых частей программы в разных процедурах библиотеки и открывает возможности для построения вычислительных схем, более гибко учитывающих особенности решаемой задачи. Выбор и построение вычислительной схемы, а также изменение схемы в процессе решения задачи в программном комплексе осуществляет монитор. На на-

чальном этапе выбор алгоритма происходит по параметрам задачи и по заданному объему памяти. Затем в зависимости от объема накопившейся информации монитор может заменить некоторые модули или их входные параметры, а также изменить связи между отдельными модулями, перейдя к новой схеме счета.

Пусть, например, на начальном этапе для решения задачи был выбран алгоритм § 1 гл. 1 с оптимальной заменой опоры, в котором двойственный шаг вычисляется с помощью n -векторов Δ , κ , z . При решении задач малого размера эти векторы можно описать n -мерными массивами и выполнить схему, описанную в алгоритме § 1 гл. 1. В задачах несколько больших размеров вычисление двойственного шага можно реализовать без применения массива для κ , перейдя к схеме с восстановлением его компонент по формулам $\kappa_j = d_i^*$ при $\Delta_j < 0$, $\kappa_j = d_{*j}$ при $\Delta_j > 0$, $\kappa_j = x_j$ при $\Delta_j = 0$ (векторы Δ , x , d_* , d^* хранятся в памяти в виде массивов). Даже многократный пересчет компонент κ_j по этим формулам почти не изменит скорости счета программы и, следовательно, нет необходимости в описании этого вектора массивом. Аналогично можно перейти к покомпонентному вычислению вектора z . Но поскольку этот вектор вычисляется значительно сложнее, то его многократный пересчет при поиске оптимального шага по трудоемкости будет равен нескольким итерациям алгоритма. Поэтому при необходимости сократить объем временно запоминаемой информации лучше перейти к алгоритму с простым выполнением двойственного шага (как в двойственном симплекс-методе), в котором компоненты вектора z применяются один раз и, следовательно, могут не запоминаться.

Дальнейшего сокращения, требуемого для работы программы объема памяти, можно достигнуть за счет исключения массива для вектора оценок Δ . Вместо рекуррентной формулы $\Delta^{(k+1)} = \Delta^{(k)} + \sigma z$, по которой пересчитываются на итерациях элементы n -мерного массива, соответствующего этому вектору, следует применить формулу, восстанавливающую его компоненты по исходным данным задачи: $\Delta_j = u'A(I_{\text{оп}}, j) - c_j$, $j = 1, n$; $u' = c'(J_{\text{оп}})A_{\text{оп}}^{-1}$. Очевидно, трудоемкость вычисления Δ по этой формуле значительно выше, особенно при решении задач больших размеров, когда матрицы A и $A_{\text{оп}}^{-1}$ списываются с внешней памяти. Кроме того, поскольку вектор Δ

на итерации адаптивного алгоритма применяется как на прямом, так и на двойственном шаге, то необходим двойной пересчет его компонент. Как видно из экспериментов § 2 гл. 1, такой вариант алгоритма требует в 3 раза больше машинного времени для решения задачи, чем алгоритм с запоминанием этого вектора, и поэтому его применение может быть оправдано только при очень больших n .

Уменьшения используемого объема оперативной памяти можно добиться и переходом к алгоритму с декомпозицией опоры, требующему в 2 раза меньший объем памяти для матрицы $A_{\text{оп}}^{-1}$. Но при этом нужно учитывать, что в силу сложности его вычислительной схемы потребуются дополнительный объем памяти для самой программы. Пусть, например, допустимый для $A_{\text{оп}}^{-1}$ объем памяти равен $22 \cdot 10^3$ ячеек; опора достигла размера, равного 100; $m=150$, $n=10^3$. При дальнейшем решении задачи опора может достигнуть максимального размера, равного 150, что потребует $22.5 \cdot 10^3$ ячеек. Недостающий объем памяти можно получить посредством отказа от запоминания вектора Δ . Но в этом случае оставшиеся итерации будут выполняться в 3 раза медленнее. Переход к алгоритму с докомпозицией опоры потребует дополнительно 1200 ячеек для программы, но максимальный объем памяти, необходимый для $A_{\text{оп}}^{-1}$, будет равен $100 \times 100 + 50 \times 50 = 12.5 \times 10^3$. Время решения задачи при этом увеличится не более чем в 1,5 раза. В этой ситуации, конечно, предпочтительнее вычислительная схема с декомпозицией опоры. Этот пример показывает, что выбор и построение схемы решения задачи осуществляется на основе простых расчетов, которые могут выполняться управляющей программой или монитором. Принципы, заложенные в монитор для выбора вычислительной схемы, определяются в основном соотношением между скоростью выполнения операций в оперативной памяти ЭВМ и ее скоростью обмена с внешними носителями информации. Поэтому в программных комплексах, ориентированных на разные типы ЭВМ, принципы выбора вычислительной схемы могут быть существенно различными.

КОММЕНТАРИИ К ГЛАВЕ 1

§ 1. Академиком Л. В. Канторовичем [27] было замечено, что на современных предприятиях повышения эффективности производства

можно добиться не только улучшением качества сырья, совершенствованием способов его обработки, но и рациональной организацией производства. Он же показал, что при математическом моделировании подобные задачи во многих случаях сводятся к линейным задачам с ограничениями-неравенствами. Классическая техника исследования на экстремум для новых задач оказалась неэффективной. Л. В. Канторович разработал оригинальный подход к решению линейных задач, который впоследствии лег в основу теории линейного программирования. Интенсивное развитие линейного программирования началось после создания Дж. Данцигом симплекс-метода [26] — основного численного метода линейного программирования.

Задачи линейного программирования в аналитическом отношении являются простейшими среди задач математического программирования. Но аналитическая природа элементов экстремальной задачи является не единственной (и даже не самой важной) характеристикой сложности. В связи с необходимостью фактического решения современных прикладных задач, что, как правило, невозможно без использования ЭВМ, в последние годы на передний план выдвигается такая характеристика сложности, как размер задачи, под которым понимается совокупность из числа переменных и количества ограничений. Как ни велики поражающие воображение рядового человека возможности симплекс-метода и ЭВМ, в приложениях все чаще появляются задачи, решение которых не под силу самым совершенным версиям симплекс-метода, даже если их реализовать на самых мощных ЭВМ. В разных странах начались исследования по разработке новых методов решения больших задач. С одной стороны, старались найти методы, основанные на принципах, отличных от тех, на которых основан симплекс-метод. С другой стороны, шел поиск схем декомпозиции больших задач, в которых в качестве модуля использовался симплекс-метод.

Исследования, начатые в 70-е годы в Минске, были направлены прежде всего на совершенствование принципов, заложенных в сам симплекс-метод [4]. Этапы развития этих исследований отражены и в данной монографии. Основные идеи нового подхода реализованы в адаптивном методе [7, 9, 18, 20, 41—44], который подробно излагается в § 1. Остальная часть монографии так или иначе базируется на адаптивном методе.

§ 2. В основном варианте (§ 1) адаптивного метода для построения итерации используются и запоминаются векторы, число компонент которых равно числу переменных задачи. Такая реализация неудобна (требует большого объема оперативной памяти), если в задаче много переменных. С точки зрения экономии памяти в данной ситуации целесообразно упомянутые векторы генерировать в процессе итераций. При этом, конечно, увеличивается время решения, но часто это допустимо. Разработке версии адаптивного метода, обладающей указанным свойством, посвящен данный параграф.

§ 3. Излагается еще одна версия адаптивного метода, построенная по аналогии с мультипликативной версией симплекс-метода. Достоинства подобных версий подробно обсуждаются в [1, 30, 32].

§ 4. Каноническая задача является частным случаем интервальной задачи, но рассматривается здесь отдельно с целью упрощения элементов адаптивного метода. Именно для этой задачи в [16] впервые описан адаптивный метод.

§ 5. Метод данного параграфа был первым из разработанных в Минске на пути обобщения симплекс-метода. В приложении к транс-

портной задаче он опубликован [3] в 1975 г., хотя разработан был значительно раньше. Симплексная нормировка является одной из многих возможных. В [8] указан класс задач, для которых она естественна. Как показывают эксперименты в § 4, по эффективности метод данного параграфа близок к модификации симплекс-метода, предложенный Чарнсом [40].

§ 6. Проблема конечности алгоритма (или, другими словами, проблема избежания заикливания) долгое время многими исследователями относилась к теоретическим проблемам, не заслуживающим внимания практиков. Но постепенно (это особенно заметно в последние годы, см., например, [2, 34]) стали понимать, что ситуации, близкие к вырождению, могут явиться причиной замедления процесса решения линейных задач. Интерес к проблеме значительно вырос. Она была одной из центральных с самого начала наших исследований. Первые результаты, полученные в Минске, содержатся в [5]. В данном параграфе излагается новая версия конечного метода. Она впервые опубликована в [12, 13].

§ 7. Данный параграф посвящен обобщению метода предыдущего параграфа, направленному на учет квазивырожденных ситуаций. Результаты экспериментов говорят о работоспособности метода в самых жестких условиях, в которых другие версии адаптивного метода терпят неудачу.

§ 8. После создания и испытания основного варианта адаптивного метода (§ 1) возник вопрос о дальнейших путях повышения его эффективности. Здесь разработана версия адаптивного метода с блочной заменой опоры. Численные эксперименты показали перспективность метода. Заметим, что аналогичные вопросы исследуются и в теории симплекс-метода В. А. Булавским [1].

§ 9. Второй путь, по которому была сделана попытка увеличить эффективность адаптивного метода, был связан с идеей многошаговых методов. В линейной алгебре такие методы известны [35], но нам не удалось найти их аналогов в линейном программировании. Приведенные в параграфе результаты численных экспериментов указывают на эффективность и этого пути обобщения адаптивного метода.

§ 10. Адаптивный метод, по терминологии [5], относится к опорным методам линейного программирования. Другой класс составляют безопорные методы. Целью данного параграфа является изложение общего взгляда на построение безопорных методов. В новой схеме имеется ряд элементов, которые позволяют говорить о существенном развитии подхода [5]. Следует отметить, что в схему безопорных методов укладывается очень много известных методов (см. например, [31]). По-видимому, в классе безопорных методов в недалеком будущем будут созданы весьма эффективные методы решения задач оптимального планирования и управления.

§ 11. Материал данного параграфа основан на работах [10, 15]. Однако приближенные методы здесь излагаются более подробно. Несмотря на значительный интерес, который в последние годы проявляется к приближенным методам [25], обстоятельных численных экспериментов пока не опубликовано. К сожалению, это слабое место всех разделов математического программирования.

КОММЕНТАРИИ К ГЛАВЕ 2

§ 1. Как уже указывалось, размер линейной задачи определяется двумя параметрами: числом переменных n и числом основных огра-

ний m , т. е. размерами матрицы условий задачи. В симплекс-методе оперативная память расходуется в основном на хранение базисной матрицы, которая в канонических задачах имеет размер $m \times m$. Поскольку на итерации небазисные столбцы матрицы условий обрабатываются независимо друг от друга, то число переменных, на первый взгляд, не влияет на возможность решения линейной задачи симплекс-методом. В связи с этим иногда сложность (размер) задачи определяют только числом основных ограничений. В действительности же и для симплекс-метода не безразлично, сколько переменных в задаче. При постоянном m число симплексных итераций, как правило, растет с увеличением n . Кроме того, в задачах с $n \gg m$ возникают дополнительные сложности из-за большого количества почти коллинеарных векторов условий. Обсуждаемые задачи для нас представляют особый интерес, так как они типичны в оптимальном управлении, а решение задач оптимального управления — цель данной монографии.

Идея метода, изложенного в § 1, достаточно прозрачна и основана на физическом смысле оценок. Однако ее реализация использует особенности, свойственные именно адаптивному методу. Попытка реализовать эту идею в рамках симплекс-метода встречает серьезные трудности. Изложение материала следует в основном работам [8, 17].

§ 2. Ограничения-неравенства в некотором отношении существенно отличаются от ограничений-равенств. В связи с тем, что они типичны в задачах оптимального управления с фазовыми ограничениями, в данной главе ограничения-неравенства рассматриваются отдельно. Для простоты исследуется только случай $m \gg n$. Изложен метод, который впервые был предложен в [17]. (Там же обсуждается его связь с другими методами.) Здесь впервые приведены результаты численного эксперимента.

§ 3. Из всех известных методов декомпозиции в данной монографии рассматривается только один [19]. Проблема выбора конкретного метода до сих пор не разработана достаточно полно. В литературе часто упоминается о медленной сходимости метода декомпозиции Данцига — Вулфа [30]. Наши численные эксперименты по методу распределения ресурсов (известного еще под названием метода Карнаи — Липтака) также не дали обнадеживающих результатов. К сожалению, во многих работах ограничиваются словесным описанием методов, не указывая области их эффективной работы. На наш взгляд, на современном этапе развития теории оптимизации такой подход к делу уже недопустим.

Идея метода данного параграфа проста и известна давно. Однако ее реализация зависит от метода, которым решаются подзадачи. В этом отношении адаптивный метод представляет значительно больше возможностей, чем симплекс-метод.

§ 4. Цель параграфа — развить адаптивный метод на большие задачи, для которых опоры больших размеров невозможно реализовать. Конструкции с «плавающими» опорами, на наш взгляд, весьма перспективны. Есть надежда на этом пути построить метод решения весьма больших задач при скромных требованиях к объему оперативной памяти.

§ 6. В этом параграфе статическая транспортная задача осложнена динамическим эффектом, в результате чего резко возросли размеры задачи. Детальная разработка адекватных методов для нового класса задач требует отдельного изучения. Здесь изложена только одна модификация адаптивного метода. В связи с рассматриваемой задачей см. работу [28].

ПРИЛОЖЕНИЕ

§ 1. Программа адаптивного метода с оптимальной заменой элементов опоры

Алгоритм § 1 гл. 1 написан на языке АЛГОЛ-60 (МС «Дубна») в виде процедуры PLAV(A, C, D1, D2, B1, B2, X, M, N, EPS, EPS2, EMI, EMA), параметры которой имеют следующий смысл: A[1 : M, 1 : N+1] — массив, в котором задается полная матрица A и дополнительный (N+1)-й столбец с нулевыми элементами; C[1 : N+1] — коэффициенты целевой функции; D1[1 : N+1] — ограничения снизу для компонент плана; D2[1 : N+1] — ограничения сверху для компонент плана; B1[1 : M] — левый вектор основных ограничений; B2[1 : M] — правый вектор основных ограничений; X[1 : N+1] — массив, в котором при обращении к процедуре PLAV задается первое приближение, удовлетворяющее прямым ограничениям ($d_1 \leq x \leq d_2$), а в результате решения задачи в нем будет получен ε -оптимальный план (при $X[N+1] > EMI$ система основных условий несовместна); M — число основных ограничений; N — число переменных задачи; EPS — параметр, значением которого задается точность решения задачи; при $\beta \leq EPS$ решение задачи прекращается и выдается ε -оптимальный план; EPS2 — параметр, по которому прекращается поиск элемента для замены опоры ($\alpha \geq -EPS2$, $EPS2 \approx 10^{-5}$); EMI — параметр, применяемый в процедуре вместо 0 ($EMI \approx 10^{-9}$); EMA — большое число (вместо ∞), $EMA \approx 10^9$.

При обращении к процедуре PLAV в массивах, описанных до N+1, нужно задать только N компонент, а (N+1)-я компонента вычислится при расширении задачи (в процедуре START). Основным операциям алгоритма соответствуют блоки программы, оформленные в виде следующих процедур: OBR(P1) — для пересчета обратной опорной матрицы $A_{оп}^{-1}$. (По значению параметра P1 выбирается метка в списке переключателя и выполняется пересчет матрицы по схеме, реализованной после этой метки.); SHAG — для вычисления шага Θ . В ней же определяются индексы j_0 и i_0 (в программе I0, I00), вычисляются опорные компоненты коплана и новые компоненты плана; KROP — для проверки плана x на оптимальность и субоптимальность. При очень малых значениях EPS условие субоптимальности может не выполняться из-за накопившихся ошибок округления. Тогда отсутствие подходящего направления (условие оптимальности плана x) является признаком окончания решения задачи; SIG — для вычисления двойственного шага ($\sigma = \sum_{k=0}^{s-1} \sigma_k$) и одновременного определения

индексов i_* , j_* (в программе J0, JJ0) для замены элементов опоры; START — для проверки выполнения основных ограничений задачи и построения расширенной задачи. Если в качестве начального прибли-

жения берется план задачи, обращение к этой процедуре можно исключить, а фактические параметры процедуры следует описывать до N .

Для реализации алгоритма, кроме исходных данных задачи, в теле процедуры PLAV используются следующие массивы: $U[1:M, 1:M]$, в котором хранится матрица $A_{оп}^{-1}$ (при $M > N$ описание этого массива следует заменить на $U[1:N, 1:N]$); $DT[1:N+1]$ — коплан Δ ; $XP[1:N+1]$ — псевдоплан \mathcal{K} ; $TH[1:N+1]$, в котором при выполнении прямого шага хранится вектор $l(J)$, а при выполнении двойственного шага — вектор $z(J)$; $Q[1:M]$ — вектор потенциалов $u(I)$; $Y, Y1[1:M]$ — векторы ω^* и ω_* ; $P, Y2[1:M]$ — рабочие массивы; $BS[1:N]$ — индексные множества $J_{оп}$ (первые k компонент) и $J_{н}$ (остальные $n-k$ компонент); $BS1[1:M]$ — индексные множества $I_{оп}$ (первые k компонент) и $I_{н}$ (остальные $m-k$ компонент).

При работе процедуры на каждой итерации печатаются: номер итерации, значения прямой и двойственной целевых функций. По окончании решения задачи печатаются: число итераций, значение целевой функции, полученный план и размерность опоры.

§ 2. Программа опорного метода

Алгоритм опорного метода (§ 5 гл. 1), ориентированный на решение задач ЛП вида: $c'x \rightarrow \max, Ax=b, d_* \leq x \leq d^*$, написан на языке АЛГОЛ-60 (МС «ДУБНА») в виде процедуры ОПОРНМ ($M, N, A, B, C, D1, D2, X, S, E$). Здесь M — число уравнений; N — число переменных; $A[1:M, 1:N+M]$ — первые N столбцов матрицы условий, остальные потребуются при расширении задачи (в процедуре START); $B[1:M]$ — вектор правых частей; $C[1:N]$ — коэффициенты целевой функции; $D1[1:N]$ — прямые ограничения на план снизу (вектор d_*); $D2[1:N]$ — прямые ограничения на план сверху (вектор d^*); $X[1:N+M]$ — первые n компонент — начальное приближение ($d_1 \leq x \leq d_2$), остальные используются в процедуре START для дополнительных переменных; на выходе из процедуры X — решение задачи; S — большое число (например, 10^6), которому равны ограничения сверху на дополнительные переменные и соответствующие им коэффициенты целевой функции; E — параметр, по которому проверяется критерий оптимальности: при $\max \Delta_j < E$ решение задачи прекращается и выдаются результаты.

В процедуре ОПОРНМ в качестве рабочих массивов используются: $AB[1:M, 1:M]$ (для $A_{оп}^{-1}$), $L[1:M]$ (для $I_{оп}$), $U[1:M]$ (для потенциалов), $BS([1:M])$ (для $J_{оп}$).

При решении задачи на итерациях печатаются номер итерации и значение целевой функции, при выполнении критерия оптимальности — невязка ($b-Ax$), значение целевой функции ($c'x$) и оптимальный план (x).

Следует отметить, что здесь приводится простейший вариант программы с полным хранением матриц A и $A_{оп}^{-1}$. Для решения больших прикладных задач применяется программная реализация опорного метода с мультипликативным представлением обратной опорной матрицы $A_{оп}^{-1}$ и с компактным хранением матрицы условий A , вычислительная схема которого значительно сложнее.

§ 3. Программа адаптивного метода с декомпозицией опоры

Алгоритм, описанный в § 4 гл. 2, реализован в виде АЛГОЛ-процедуры DPLAV(A, C, D1, D2, B1, B2, X, NCA, M, N, AK, CK, D1K, D2K, B1K, B2K, M0, BUF, EPS). Параметры этой процедуры имеют следующий смысл: M — число основных ограничений; AK — число ненулевых элементов в матрице A; N — число переменных; CK — число ненулевых коэффициентов в целевой функции; B1K — число заданных основных ограничений слева; B2K — число заданных основных ограничений справа; D1K — число заданных (отличных от нуля) прямых ограничений слева; D2K — число заданных ($\neq \infty$) прямых ограничений справа; A[1:AK] — массив ненулевых элементов матрицы A (с каждым элементом упакован его индекс), выписанный по столбцам; C[1:CK], D1[1:D1K], D2[1:D2K] — массивы ненулевых элементов, упакованных вместе со своими индексами, векторов c , d_* , d^* ; B1[1:B1K], B2[1:B2K] — массивы заданных основных ограничений (слева и справа), вместе с элементами упакованы номера их строк; NCA[1:N+1] — адреса последних элементов столбцов матрицы в массиве A (NCA[1]=0, NCA[2]=K1, если в первом столбце K1 ненулевых элементов, ..., NCA[N+1]=AK); X[1:N] — вектор, в котором задается первое приближение ($D1 \leq X \leq D2$), а в результате решения задачи получается оптимальный план; M0 — размер опоры, при достижении которого в работу включается алгоритм с декомпозицией опоры (при M0=M, BUF=M+1 задача будет решена с помощью обычного алгоритма (§ 2 гл. 1) без декомпозиции опоры); BUF(M0 < BUF ≤ M) — максимальный размер опоры, который будет достигнут при решении задачи. Если оценка сверху на размер опоры неизвестна, то следует задать BUF=M (при M0=M BUF=M+1, иначе

верхняя граница массива для \tilde{A}_{20n}^{-1} будет меньше нижней); EPS — параметр, с которым сравнивается оценка субоптимальности. При $\beta \leq \epsilon$ решение задачи прекращается, выдаются результаты.

В случае, когда неизвестно, является ли первое приближение планом задачи, указанные массивы следует описать «с запасом» для расширения задачи (в процедуре START): A[1:AK+M], NCA[1:N±2], D2[1:D2K+1], C[1:CK+1], X[1:N+1]. Отсутствующие в заданных массивах компоненты векторов при решении задачи будут принимать следующие значения («по умолчанию»): $c_j = 0$, $d_{*j} = 0$, $d_j^* = 10^6$, $b_{*i} = -10^6$, $b_i^* = 10^6$. Для массивов, описанных в теле процедуры DPLAV, требуется $5 * BUF + 2 * M + M0 * 2 + (BUF - M0) * 2$ чеек (n -мерных массивов нет).

На каждой итерации печатаются: номер итерации, значение целевой функции, оценка субоптимальности и размер опоры. Эти же величины выдаются на печать вместе с ϵ -оптимальным планом задачи, когда решение задачи будет найдено.

В силу двойного пересчета на каждой итерации компонент ко-плана Δ данная процедура работает медленнее, чем процедура PLAV.

Применение процедуры DPLAV целесообразно в случае сильно разреженных векторов и матрицы задачи, а также в случае, когда число переменных столь велико, что описание n -мерных массивов недопустимо.

```

PROCEDURE PLAV(A, C, D1, D2, B1, B2, X, M, N, EPS, EPS2, EMI, EMA);
ARRAY A, C, D1, D2, B1, B2, X;
INTEGER M, N; REAL EPS, EPS2, EMI, EMA;
BEGIN
  INTEGER I, J, K, IT, IO, JO, JJ, K1, I00, JJ0, L, I1, J1;
  BOOLEAN BL1, BL; REAL R1, R2, BY, TT, XI0, TI0, AL, CX, GZ;
  ARRAY DT, XP, TH[1:N], P, Q, Y, Y1, Y2[1:M], U[1:N, 1:N];
  INTEGER ARRAY BS[1:N], BS1[1:M];
  PROCEDURE OBR(P1); INTEGER P1; BEGIN
    SWITCH PER:=M1, M2, M3, M4; GOTO PER[P1];
M1: FOR I:=1 STEP 1 UNTIL K'DO 'BEGIN' Y2[I]:=0;
    FOR J:=1 STEP 1 UNTIL K'DO Y2[I]:=Y2[I]+U[I, J]*A[BS1[J], J0]
    END; R1:=Y2[I0]; FOR J:=1 STEP 1 UNTIL K'DO
    BEGIN U[I0, J]:=U[I0, J]/R1;
    FOR I:=1 STEP 1 UNTIL IO-1, IO+1 STEP 1 UNTIL K'DO
    U[I, J]:=U[I, J]-U[I0, J]*Y2[I] END; GOTO M5;
M2: FOR I:=1 STEP 1 UNTIL K'DO
    BEGIN R1:=U[I0, I]; U[I0, I]:=U[K, I]; U[K, I]:=R1 END;
    FOR I:=1 STEP 1 UNTIL K'DO
    BEGIN R1:=U[I, J0]; U[I, J0]:=U[I, K]; U[I, K]:=R1 END;
    FOR I:=1 STEP 1 UNTIL K-1 DO 'BEGIN' R1:=U[I, K]/U[K, K];
    FOR J:=1 STEP 1 UNTIL K-1 DO
    U[I, J]:=U[I, J]-U[K, J]*R1 END;
    R1:=BS1[J0]; BS1[J0]:=BS1[K]; BS1[K]:=R1;
    R1:=BS[I0]; BS[I0]:=BS[K]; BS[K]:=R1; K:=K-1; GOTO M5;
M3: R1:=A[I00, J0];
    FOR I:=1 STEP 1 UNTIL K'DO 'BEGIN' Y2[I]:=0;
    FOR J:=1 STEP 1 UNTIL K'DO
    Y2[I]:=Y2[I]-U[I, J]*A[BS1[J], J0]; R1:=R1+Y2[I]*A[I00, BS[I]]
    END; FOR I:=1 STEP 1 UNTIL K'DO
    BEGIN 'FOR' J:=1 STEP 1 UNTIL K'DO U[I, J]:=U[I, J]+
    Y2[I]*P[J]/R1; U[I, K+1]:=Y2[I]; U[K+1, I]:=P[I]/R1
    END; K:=K+1; U[K, K]:=1/R1; BS1[I]:=BS1[K]; BS1[K]:=I00;
    BS[I1]:=BS[K]; BS[K]:=J0; GOTO M5;
M4: R1:=-P[IJ0]; FOR J:=1 STEP 1 UNTIL K'DO
    BEGIN AL:=U[J, J0]; U[J, J0]:=U[J, J0]/R1;
    FOR I:=1 STEP 1 UNTIL JJ0-1, JJ0+1 STEP 1 UNTIL K'DO
    U[J, I]:=U[J, I]+AL*P[I] END; GOTO M5;
M5: END OBR;
PROCEDURE SHAG;
  BEGIN TT:=1; FOR I:=1 STEP 1 UNTIL K'DO
    BEGIN L:=BS[I]; R1:=EMA; AL:=TH[L];
    IF ABS(AL)>EMI THEN 'BEGIN'
    IF AL>0 THEN R1:=(D2[L]-X[L])/AL;
    IF AL<0 THEN R1:=(D1[L]-X[L])/AL;
    IF R1<TT THEN 'BEGIN' TT:=R1; IO:=1 END 'END' END I; I00:=0;
    I00:=0;
    FOR K1:=K+1 STEP 1 UNTIL M'DO 'BEGIN' I:=BS1[K1];
    R1:=0; FOR J:=1 STEP 1 UNTIL N'DO R1:=R1+A[I, J]*TH[J];
    GZ:=EMA; IF R1>EMI THEN 'BEGIN' GZ:=Y[I1]/R1; L:=-1;
    CX:=Y[I1]-R1*END; IF R1<-EMI THEN
    BEGIN CX:=R1-Y[I1]; L:=1; GZ:=Y[I1]/R1 END;
    IF GZ<TT THEN 'BEGIN' TT:=GZ; I00:=I; AL:=CX; TI0:=L; I1:=K1
    END; EI: END I;
    FOR I:=1 STEP 1 UNTIL K'DO 'BEGIN' L:=BS[I]; XP[L]:=0;
    FOR J:=1 STEP 1 UNTIL N'DO
    XP[L]:=XP[L]+A[BS1[I], J]*X[J]+TH[J] END;
    IF I00=0 THEN XI0:=X[BS[I0]]+TH[BS[I0]];
    FOR J:=1 STEP 1 UNTIL N'DO X[J]:=X[J]+TT*TH[J] END SHAG;

PROCEDURE KROP; BEGIN BY:=0;
  FOR I:=1 STEP 1 UNTIL K'DO 'BEGIN' DT[BS[I]]:=-Q[I];
  P[I]:=(IF Q[I]>EMI THEN -Y[BS1[I]] ELSE 0;
  IF Q[I]<-EMI THEN P[I]:=-Y[BS1[I]] END;
  BL:=TRUE;
  FOR JJ:=K+1 STEP 1 UNTIL N'DO 'BEGIN' J:=BS1[J];
  R1:=0; XP[J]:=X[J];
  IF DT[J]>EMI THEN 'BEGIN' R1:=D1[J]-X[J]; XP[J]:=D1[J];
  IF ABS(R1)>GE'EMI THEN BL:=FALSE;
  BY:=BY-DT[J]*R1 END; IF DT[J]<-EMI THEN
  BEGIN R1:=D2[J]-X[J]; XP[J]:=D2[J];
  IF ABS(R1)>GE'EMI THEN BL:=FALSE;
  BY:=BY-DT[J]*R1 END; TH[J]:=R1;
  FOR I:=1 STEP 1 UNTIL K'DO P[I]:=P[I]+A[BS1[I], J]*R1/KJ;
  END J; FOR I:=1 STEP 1 UNTIL K'DO 'BEGIN' R1:=0;
  FOR L:=1 STEP 1 UNTIL K'DO R1:=R1+U[I, L]*P[L];

```

```

CX:=Y[BS1[I]]J;GZ:=Y1[BS1[I]]J;
'IF'Q[I]>0'THEN'BY:=BY+Q[I]*CX;
'IF'Q[I]<0'THEN'BY:=BY+Q[I]*GZ;
'IF'ABS(CX)<EMI'AND'Q[I]<-EMI'THEN'BL:='FALSE';
'IF'ABS(GZ)<EMI'AND'Q[I]>EMI'THEN'BL:='FALSE';
'IF'ABS(CX)>EMI'AND'ABS(GZ)>EMI'AND'ABS(Q[I])>EMI'THEN'
BL:='FALSE';TH[BS1[I]]:=-R1'END'KROP;
'PROCEDURE'CIG;'BEGIN''BOOLEAN'BL1;'REAL'RL,RP;
GZ:=EMA;R2:=0;TT:=0;'GOTO'MC;
MM:TT:=17;'FOR'J:=1'STEP'1'UNTIL'N'DO''BEGIN'
'IF'ABS(TH[J])<EMI'THEN'GOTO'JJK;
'IF'DT[J]*TH[J]<-EMI'THEN'BEGIN'R1:=-DT[J]/TH[J];
'IF'R1<TT'THEN'BEGIN'TT:=R1;J0:=J'END''END''JJK;'END'J;
MC:BL:=TRUE';J0:=0;R2:=R2+TT;
BL:=TRUE';J0:=0;R2:=R2+TT;
'FOR'JJ:=1'STEP'1'UNTIL'N'DO''BEGIN'J:=BS[JJ];
DT[J]:=DT[J]+TT*TH[J];'IF'ABS(DT[J])<EMI'THEN'
'BEGIN''IF'JJ>K'THEN'
'BEGIN'GZ:=D1[J];CX:=D2[J];RL:=D2[J]-X[J];
RP:=X[J]-D1[J];BL1:=TRUE'END''ELSE'
'BEGIN'BL1:=FALSE';GZ:=B1[BS1[JJ]];CX:=B2[BS1[JJ]];
RL:=Y[BS1[JJ]];RP:=-Y1[BS1[JJ]]'END';
R1:=0;'IF'TH[J]<-EMI'THEN'R1:=TH[J]*(XP[JJ]-CX);
'IF'TH[J]>EMI'THEN'R1:=TH[J]*(XP[JJ]-GZ);AL:=AL+R1;
'IF'R1>0'THEN'BEGIN''IF'RL>EMI'AND'RP>EMI'THEN''BEGIN'
'IF'BL1'THEN'BEGIN'J1:=JJ;J0:=J'END'
'ELSE'J0:=JJ;BL:=FALSE'END';
'IF'BL'THEN'BEGIN''IF'BL1'THEN'BEGIN'J1:=JJ;J0:=J'END'
'ELSE'J0:=JJ'END''END''END'J;
'IF'AL<-EPS2'THEN'GOTO'MM;
KC:'FOR'I:=1'STEP'1'UNTIL'K'DO'Q[I]:=-DT[BS1[I]]'END'CIG;
'FOR'I:=1'STEP'1'UNTIL'M'DO'BS1[I]:=I;
IT:=K:=0;'FOR'J:=1'STEP'1'UNTIL'N'DO'
'BEGIN'BS[J]:=J;DT[J]:=-C[J]'END';
'FOR'I:=1'STEP'1'UNTIL'M'DO''BEGIN'R1:=0;
'FOR'J:=1'STEP'1'UNTIL'N'DO'R1:=R1+A[I,J]*X[J];
Y1[I]:=B1[I]-R1;Y[I]:=B2[I]-R1'END';
NC:KROP;'IF'BL'THEN'GOTO'KON;SHAG;CX:=0;
'FOR'J:=1'STEP'1'UNTIL'N'DO'CX:=CX+C[J]*X[J];
PRINT(NEWLINE,'ИТЕРАЦИЯ',IT,'CX=',CX);
R1:=0;'FOR'I:=1'STEP'1'UNTIL'K'DO''IF'Q[I]<0
'THEN'R1:=R1+B1[BS1[I]]*Q[I]'ELSE'R1:=R1+B2[BS1[I]]*Q[I];
'FOR'J:=1'STEP'1'UNTIL'N'DO''BEGIN'
'FOR'I:=1'STEP'1'UNTIL'K'DO''IF'J=BS[I]'THEN''GOTO'KKK;
'IF'DT[I]<0'THEN'R1:=R1-DT[I]*D2[I]'ELSE'
R1:=R1-DT[I]*D1[I];KKK:'END'J;
PRINT('BY',R1,NEWLINE);
'IF'(1-TT)*BY/LE/EPSTHEN''GOTO'KON;
IT:=IT+1;
'FOR'I:=1'STEP'1'UNTIL'M'DO''BEGIN'R1:=0;
'FOR'J:=1'STEP'1'UNTIL'N'DO'R1:=R1+A[I,J]*X[J];
Y1[I]:=B1[I]-R1;Y[I]:=B2[I]-R1'END';
'IF'I0=0.5'THEN'GOTO'CB;
L:=BS[I0];
'IF'XIO<D1[L]+EMI'THEN'BEGIN'TIO:=1;
AL:=XIO-D1[L]'END''ELSE'BEGIN'TIO:=-1;
AL:=D2[L]-XIO'END';
'FOR'I:=1'STEP'1'UNTIL'K'DO''BEGIN'
P[I]:=UI0,I]*TIO;TH[BS[I]]:=-P[I]'END';
'FOR'JJ:=K+1'STEP'1'UNTIL'N'DO''BEGIN'J:=BS[JJ];
R1:=0;'FOR'I:=1'STEP'1'UNTIL'K'DO'
R1:=R1+P[I]*A[BS1[I],J];TH[J]:=R1'END'J;
CIG;'IF'JJ0<0.5'THEN'
'BEGIN'DT[BS[I0]]:=TIO*R2;OBR(1);
BS[J1]:=BS[I0];BS[I0]:=J0'END'
'ELSE'BEGIN'DT[BS[I0]]:=TIO*R2;Q[J0]:=Q[K];OBR(2)'END';
'GOTO'NC;
CB:'FOR'I:=1'STEP'1'UNTIL'K'DO''BEGIN'R1:=0;
'FOR'L:=1'STEP'1'UNTIL'K'DO'
R1:=R1-A[I0,BS[L]]*U[L,I];P[I]:=R1;TH[BS[I]]:=R1*TIO'END';
'FOR'JJ:=K+1'STEP'1'UNTIL'N'DO''BEGIN'J:=BS[JJ];
R1:=A[I0,J];'FOR'I:=1'STEP'1'UNTIL'K'DO'
R1:=R1+P[I]*A[BS1[I],J];TH[J]:=-R1*TIO'END'J;
CIG;'IF'JJ0=0'THEN'BEGIN'OBR(3);Q[K]:=-TIO*R2'END''ELSE'
'BEGIN'OBR(4);BS1[I1]:=BS1[IJ0];BS1[J0]:=I0;
Q[J0]:=-TIO*R2'END';
'GOTO'NC;

```

```
KON: CX:=0; 'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO' CX:=CX+C[I]*X[I];
PRINT(NEWLINE, 'ЧИСЛО ИТЕРАЦИЙ', IT, NEWLINE);
PRINT('ЗНАЧЕНИЕ ФУНКЦИИ', CX, NEWLINE);
PRINT('ОПТИМАЛЬНЫЙ ПЛАН', NEWLINE, X, NEWLINE);
PRINT(NEWLINE, 'РАЗМЕРНОСТЬ ОПОРЫ', K);
KK: 'END' PLAY;
```

§ 2

```
'PROCEDURE' ОПОРМ(M, N, A, B, C, D1, D2, X, S, E);
'INTEGER' M, N; 'ARRAY' A, B, C, D1, D2, X; 'REAL' S, E;
'BEGIN' 'INTEGER' I, J, IO, JO, IT; 'BOOLEAN' BI, BL;
'REAL' R, DJO; 'INTEGER' 'ARRAY' BS[I: M]; 'ARRAY'
AB[I: M, I: M], L, U[I: M]; 'PROCEDURE' START;
'BEGIN' 'INTEGER' I, J; 'REAL' R; IT:=0;
'FOR' I:=1 'STEP' 1 'UNTIL' M 'DO' 'BEGIN' R:=B[I];
'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO' R:=R-A[I, J]*X[J];
'FOR' J:=1 'STEP' 1 'UNTIL' M 'DO' 'BEGIN' AB[I, J]:=0;
A[I, N+J]:=0 'END'; AB[I, I]:= 'IF' R<0 'THEN' -1 'ELSE' 1;
A[I, N+I]:=AB[I, I];
X[N+I]:=ABS(R); BS[I]:=N+I; 'END';
'END' START; 'PROCEDURE' KROPT; 'BEGIN' 'REAL' R1, R2,
DT; 'FOR' I:=1 'STEP' 1 'UNTIL' M 'DO' U[I]:=0;
'FOR' I:=1 'STEP' 1 'UNTIL' M 'DO' 'BEGIN' J:=BS[I];
R:= 'IF' J>N 'THEN' -S 'ELSE' C[I]; 'FOR' J:=1
'STEP' 1 'UNTIL' M 'DO' U[J]:=U[J]+R*AB[I, J] 'END' I;
DJO:=0; 'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO' 'BEGIN'
'FOR' I:=1 'STEP' 1 'UNTIL' M 'DO' 'IF' J=BS[I]
'THEN' 'GOTO' KJ; 'IF' J>N 'THEN' 'BEGIN' R2:=S;
R1:=0; DT:=U[I]-N+I+S; 'END' 'ELSE' 'BEGIN' R1:=D1[I];
R2:=D2[I]; DT:=-C[I];
'FOR' I:=1 'STEP' 1 'UNTIL'
M 'DO' DT:=DT+U[I]*A[I, J] 'END'; 'IF' ABS(X[J]
-R1)<E 'THEN' 'BEGIN' 'IF' DT<-ABS(DJO) 'THEN'
'GOTO' MJ; 'GOTO' KJ; 'END'; 'IF' ABS(X[J]-R2)<E
'THEN' 'BEGIN' 'IF' DT>ABS(DJO) 'THEN' 'GOTO' MJ;
'GOTO' KJ; 'END'; 'IF' ABS(DT)<ABS(DJO) 'THEN'
'GOTO' KJ; MJ:=DJO:=DT; JO:=J; KJ:= 'END' J;
'IF' ABS(DJO)<E 'THEN' 'GOTO' KKK; 'END' KROPT;
'PROCEDURE' TETA; 'BEGIN' 'REAL' TT, LJO, R1;
LJO:=-SIGN(DJO); 'FOR' I:=1 'STEP' 1 'UNTIL' M
'DO' 'BEGIN' R:=0; 'FOR' J:=1 'STEP' 1 'UNTIL' M 'DO'
R:=R*AB[I, J]*A[J, JO]; U[I]:=R/L[I]; R:=LJO* 'END' I;
BL:= 'TRUE'; 'IF' JO>N 'THEN' 'BEGIN' TT:= 'IF' LJO>0
'THEN' S 'ELSE' X[JO] 'END' 'ELSE' TT:= 'IF' LJO>0
'THEN' DZ[JO]-X[JO] 'ELSE' X[JO]-D1[JO]; 'FOR' I:=1
'STEP' 1 'UNTIL' M 'DO' 'BEGIN' J:=BS[I]; R:=L[I];
R1:=S; 'IF' R>E 'AND' J<LE 'N 'THEN' R1:=(D2[I]-
X[J])/R; 'IF' R<-E 'THEN' 'BEGIN' R1:= 'IF' J<LE 'N
'THEN' (D1[I]-X[J])/R 'ELSE' -X[J]/R 'END'; 'IF'
R1<TT 'THEN' 'BEGIN' TT:=R1; IO:=1; BL:= 'FALSE';
'END'; 'END' I; 'FOR' I:=1 'STEP' 1 'UNTIL' M 'DO' 'BEGIN'
J:=BS[I]; X[J]:=X[J]+TT*L[I] 'END'; X[JO]:=X[JO]+
TT*LJO; IT:=IT+1; 'IF' BL 'THEN' 'GOTO' NN;
'END' TETA; 'PROCEDURE' OPORA; 'BEGIN' BS[IO]:=JO;
'FOR' I:=1 'STEP' 1 'UNTIL' M 'DO'
AB[IO, J]:=AB[IO, J]/U[IO]; 'FOR' I:=1 'STEP' 1 'UNTIL'
IO-1, IO+1
'STEP' 1 'UNTIL' M 'DO' 'FOR' J:=1 'STEP' 1
'UNTIL' M 'DO' AB[I, J]:=AB[I, J]-
AB[IO, J]*U[I]; 'END' OPORA; 'PROCEDURE' FIN;
'BEGIN' CX:=0; 'FOR' I:=1 'STEP' 1 'UNTIL' M 'DO' L[I]:=B[I];
'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO' 'BEGIN' CX:=CX+C[J]*X[J];
'FOR' I:=1 'STEP' 1 'UNTIL' M 'DO' L[I]:=L[I]-A[I, J]*X[J] 'END' J
PRINT(NEWLINE, 'НЕВЯЗКА', L);
PRINT(NEWLINE, 'CX=', CX);
PRINT(NEWLINE, 'ОПТИМАЛЬНЫЙ ПЛАН', X);
'END;
```

```
START; NN: KROPT; TETA;
CX:=0; 'FOR' I:=1 'STEP' 1 'UNTIL' N+M 'DO' CX:=
'IF' I<LE 'N 'THEN' CX+C[I]*X[I] 'ELSE' CX-S*X[I];
PRINT(NEWLINE, 'IT=', IT, 'CX=', CX);
OPORA; 'GOTO' NN; KKK: FIN; 'END' ОПОРМ;
```

```

( PROCEDURE 'DPLAV(A,C,D1,D2,B1,B2,X,NCA,M,N,AK,CK,D1K,D2K,B1K,
B2K,MO,BUF,EPS); 'ARRAY'A,C,D1,D2,B1,B2,X;'INTEGER'ARRAY'NCA;
'INTEGER'M,N,AK,CK,D1K,D2K,B1K,B2K,MO,BUF;'REAL'EPS;
'BEGIN' 'INTEGER'I,J,I1,J1,I2,J2,I,J,I,J,NC,ND1,ND2,IO,JO,
IOO,JOO,IX,JX,IT,ROP1,ROP2,K;'REAL'T,TT,R,PR,CR,AL,LH,E1,BR,
DT;'REAL'SIG,BT;'INTEGER'K1,NO,DN1,DN2;
'ARRAY'A1[1:MO,1:MO],A2[1:BUF-MO,1:BUF-MO],4,P,LOP[1:BUF],
V,W[1:M]; 'INTEGER'ARRAY'BS,BS1[1:BUF]; 'BOOLEAN'BL;
'REAL'PROCEDURE'KB(N,Y,PN,PK,B); 'VALUE'B,N;'INTEGER'N,PN,
PK;'REAL'B;'ARRAY'Y;'BEGIN' 'REAL'R;'INTEGER'I,J1;R:=B;
'FOR'I:=PN'STEP'1'UNTIL'PK'DO' 'BEGIN'PLAOR1(Y[I],J1);
'IF'J1>N'THEN'GOTO'EKB'; 'IF'J1=N'THEN' 'BEGIN'PN:=I+1;
R:=Y[I];GOTO'EKB'END'END'I;EKB:KB:=R'END'KB;
'REAL'PROCEDURE'RW(J); 'VALUE'J;'INTEGER'J;'BEGIN' 'REAL'R;
'INTEGER'I,I:=BS1[J];R:=0;ND1:=ND2:=1;'IF'U[I]>E1'THEN'
R:=KB(I,B2,ND2,B2K,BR)-W[I]; 'IF'U[J]<-E1'THEN'
R:=KB(I,B1,ND1,B1K,-BR)-W[I];RW:=R'END'RW;
'PROCEDURE'KROPT;'BEGIN' 'FOR'I:=1'STEP'1'UNTIL'ROP1'DO'
U[I]:=0;'FOR'I:=1'STEP'1'UNTIL'ROP1'DO' 'BEGIN'I1:=BS[I];
NC:=1;CR:=KB(I1,C,NC,CK,O); 'IF'CR=O'THEN'GOTO'KI;
'FOR'J:=1'STEP'1'UNTIL'ROP1'DO'U[J]:=U[I]+CR*A1[I,J];
KI:'END'I;'IF'ROP2=O'THEN'GOTO'KU;'FOR'I:=1'STEP'1'UNTIL'
ROP2'DO' 'BEGIN'I1:=BS[ROP1+I];NC:=1;
PC[I]:=KB(I1,C,NC,CK,O);J1:=NCACI1+I1-NCAI1;
JJ:=0;'FOR'J:=1'STEP'1'UNTIL'J1'DO' 'BEGIN'I2:=NCAI1+J;
PLAOR1(AI2,J2); 'FOR'I1:=1'STEP'1'UNTIL'ROP1'DO' 'IF'BS1[I1]
=J2'THEN' 'BEGIN'PC[I1]:=PC[I]-AI2*U[I1];JJ:=JJ+1;'IF'JJ=ROP1
'THEN'GOTO'ZJ'ELSE'GOTO'KJ'END';KJ:'END'J;ZJ:'END'I;
'FOR'J:=1'STEP'1'UNTIL'ROP2'DO' 'BEGIN'I:=ROP1+J;U[I]:=0;
'FOR'I1:=1'STEP'1'UNTIL'ROP2
'DO'U[I1]:=U[I]+PC[I1]*A2[I1,J]'END';
'FOR'I:=1'STEP'1'UNTIL'ROP1'DO'
'BEGIN'I1:=BS[I];JJ:=0;PC[I]:=0;
J1:=NCAI1+I1-NCAI1; 'FOR'J:=1'STEP'1'UNTIL'J1'DO'
'BEGIN'I2:=NCAI1+J;PLAOR1(AI2,J2);
'FOR'I1:=1'STEP'1'UNTIL'ROP2'DO' 'IF'BS1[ROP1+I1]=J2'THEN'
'BEGIN'PC[I1]:=PC[I]+U[ROP1+I1]*AI2;JJ:=JJ+1;
'IF'JJ=ROP2'THEN'GOTO'ZJ1'ELSE'GOTO'KJ1'END';KJ1:'END'J;
ZJ1:'END'I;'FOR'I:=1'STEP'1'UNTIL'ROP1'DO'
'FOR'J:=1'STEP'1'UNTIL'ROP1'DO'U[I]:=U[I]-PC[J]*A1[J,I];
KU:BT:=0;ND1:=ND2:=NC:=1;'FOR'I:=1'STEP'1
'UNTIL'M'DO'V[I]:=W[I]:=0;
'FOR'J:=1'STEP'1'UNTIL'N'DO' 'BEGIN'BL:=FALSE';
'FOR'I:=1'STEP'1'UNTIL'K'DO' 'IF'BS[I]=J'THEN' 'BEGIN'
BL:=TRUE;'GOTO'KD1'END';DT:=KB(J,C,NC,CK,O);KD1:
J1:=NCAI1+J1-NCAI1;JJ:=0;'FOR'I:=1'STEP'1'UNTIL'J1'DO'
'BEGIN'I2:=NCAI1+I;
PLAOR1(AI2,J2);W[J2]:=W[J2]+AI2*X[I];
'IF'BL'OR'JJ=K'THEN'GOTO'KII;'FOR'I:=1'STEP'1'UNTIL'K'DO'
'IF'BS1[I]=J2'THEN' 'BEGIN'DT:=DT+AI2*U[I];JJ:=JJ+1'END';
KII:'END'I;'IF'BL'THEN'GOTO'KBJ;LH:=0;'IF'DT<-E1'THEN'
LH:=KB(J,D2,ND2,D2K,BR)-X[I]; 'IF'DT>E1'THEN'
LH:=KB(J,D1,ND1,D1K,O)-X[I];BT:=BT+LH;
'FOR'I:=1'STEP'1'UNTIL'J1'DO' 'BEGIN'I2:=NCAI1+I;
PLAOR1(AI2,J2);V[J2]:=V[J2]+AI2*LH'END';KBJ:'END';
'FOR'I:=1'STEP'1'UNTIL'K'DO'BT:=BT+U[I]*RW(I);
'IF'BT<EPS'THEN'GOTO'KON'END'KROPT;
'PROCEDURE'BLOP;'BEGIN' 'FOR'I:=1'STEP'1'UNTIL'ROP1'DO'
'BEGIN'CR:=0;'FOR'J:=1'STEP'1'UNTIL'ROP1'DO'
CR:=CR+A1[I,J]*(RW(J)-V[BS1[J]]);LOP[I]:=CR'END';
'IF'ROP2=O'THEN'GOTO'MN;
'FOR'I:=1'STEP'1'UNTIL'ROP1'DO' 'BEGIN'I1:=BS[I];JJ:=0;
J1:=NCAI1+I1-NCAI1; 'FOR'J:=1'STEP'1'UNTIL'J1'DO' 'BEGIN'
I2:=NCAI1+J;PLAOR1(AI2,J2);
'FOR'I1:=ROP1+1'STEP'1'UNTIL'K'DO'
'IF'BS1[I1]=J2'THEN'
'BEGIN'V[J2]:=V[J2]+AI2*LOP[I];JJ:=JJ+1;
'IF'JJ=ROP2'THEN'GOTO'M2'ELSE'GOTO'M1'END';
M1:'END'J;M2:'END'I;
'FOR'I:=1'STEP'1'UNTIL'ROP2'DO' 'BEGIN'PR:=0;
'FOR'J:=1'STEP'1'UNTIL'ROP2'DO'
PR:=PR-A2[I,J]*(V[BS1[ROP1+J]]
-RW(ROP1+J));LOP[ROP1+I]:=PR'END';
'FOR'J:=ROP1+1'STEP'1'UNTIL'K'DO'
'BEGIN'I1:=BS[J];J1:=NCAI1+J1-NCAI1;

```



```

?FOR'I:=1'STEP'1'UNTIL'J1'DO'
#BEGIN'I2:=NCA[I1]+I;PLAOR1(A[I2],J2);
V[IJ2]:=V[IJ2]+A[I2]*LOP[IJ]
#END'I'END'J;'FOR'I:=1'STEP'1'UNTIL'ROP1'DO''BEGIN'PR:=0;
'FOR'J:=1'STEP'1'UNTIL'ROP1'DO'
PR:=PR-A1[I,J]*(V[BS1[IJ]]-RW(J));
LOP[IJ]:=PR'END'I;MN:='FOR'J:=1'STEP'1'UNTIL'ROP1'DO''BEGIN'
I1:=BS[IJ];J1:=NCA[I1+1]-NCA[I1];
'FOR'I:=1'STEP'1'UNTIL'J1'DO'
#BEGIN'I2:=NCA[I1]+I;
PLAOR1(A[I2],J2);V[IJ2]:=V[IJ2]+A[I2]*LOP[IJ]
#END'I'END'J'END'BLOP;
'PROCEDURE'SHAG;'BEGIN'TT:=1;
'FOR'I:=1'STEP'1'UNTIL'K'DO'
#BEGIN'J:=BS[I1];CR:=LOP[I1];
'IF'ABS(CR)<E1'THEN'GOTO'MT;
NC:=1;'IF'CR>E1'THEN'R:=KB(J,D2,NC,D2K,BR);'IF'CR<-E1'THEN'
R:=KB(J,D1,NC,D1K,);R:=(CR-X[IJ])/CR;'IF'R<TT'THEN'
#BEGIN'TT:=R;JO:=1'END';MT:'END'I;IO:=0;NC:=ND1:=1;
'FOR'I:=1'STEP'1'UNTIL'M'DO'
#BEGIN'FOR'J:=1'STEP'1'UNTIL'K'DO'
'IF'BS1[IJ]=I'THEN'GOTO'KS;
CR:=V[IJ];'IF'ABS(CR)<E1'THEN'GOTO'KS;
'IF'CR>E1'THEN'R:=KB(I,B2,NC,B2K,BR);'IF'CR<-E1'THEN'
R:=KB(I,B1,ND1,B1K,-BR);R:=(CR-W[I1])/CR;'IF'R<TT'THEN''BEGIN'
TT:=R;IO:=1'END';KS:'END'I;'IF'IO=0'THEN''BEGIN'
K1:='IF'LOP[IJ0]<0'THEN'1'ELSE'-1'END''ELSE''BEGIN'
K1:='IF'V[I0] THEN'1'ELSE'-1'END''END'SHAG;
'PROCEDURE'SIGJO;'BEGIN'SIG:=18;
BL:=FALSE;IOO:=0;JI:=JO-ROP1;
'FOR'I:=ROP1+1'STEP'1'UNTIL'K'DO''BEGIN'R:=A2[IJ,I-ROP1]*K1;
MINCIG;'IF'BL'THEN'GOTO'KC'END'I;
KC:'FOR'IJ:=1'STEP'1'UNTIL'ROP1'DO'
#BEGIN'J:=BS[IJ];R:=0;JJ:=0;J1:=NCA[J+1]-NCA[J];
'FOR'I:=1'STEP'1'UNTIL'J1'DO'
#BEGIN'I2:=NCA[J]+I;PLAOR1(A[I2],J2);
'FOR'II:=ROP1+1'STEP'1'UNTIL'K'DO''IF'BS1[I1J]=J2'THEN''BEGIN'
R:=R-A2[IJ,I-ROP1]*A[I2];JJ:=JJ+1;'IF'JJ=ROP2'THEN'GOTO'MJ2
'ELSE'GOTO'MJ1'END';MJ1:'END'I;MJ2:PC[IJ]:=R'END'IJ;
'FOR'J:=1'STEP'1'UNTIL'ROP1'DO''BEGIN'R:=0;
'FOR'IJ:=1'STEP'1'UNTIL'ROP1'DO'R:=R+P[IJ]*A1[J,I];V[IJ]:=R;
R:=RK1;'IF'NOT'BL'THEN'MINCIG'END'I;
ND1:=ND2:=NC:=1;J00:=0;'FOR'J:=1'STEP'1'UNTIL'N'DO'

'BEGIN'FOR'I:=1'STEP'1'UNTIL'K'DO''IF'J=BS[I1] THEN'GOTO'KEJ;
DT:=-KB(J,C,NC,CK,);R:=0;JJ:=0;J1:=NCA[J+1]-NCA[J];
'FOR'I:=1'STEP'1'UNTIL'J1'DO'
#BEGIN'I2:=NCA[J]+I;PLAOR1(A[I2],J2);
'FOR'II:=1'STEP'1'UNTIL'K'DO''IF'BS1[I1J]=J2'THEN''BEGIN'
R:=R-PR;DT:=DT+U[I1J]*A[I2];JJ:=JJ+1;'IF'JJ=K'THEN'GOTO'M6
'ELSE'GOTO'M5'END';M5:'END'I;M6:OPTDT;HXJ;KEJ:'END'J;
'FOR'I:=1'STEP'1'UNTIL'K'DO'X[BS[I1]]:=X[B2[I1]]+TT*LOP[I1];
'IF'(1-TT)*BT<EPS'THEN'GOTO'KON'END'SIGJO;
'PROCEDURE'MINCIG;'BEGIN'DT:=U[I1];'IF'ABS(DT)<E1'THEN''BEGIN'
ND1:=ND2:=1;
T:=0;I1:=BS1[I1];'IF'R>E1'THEN'T:=KB(I1,B2,ND2,B2K,BR)-W[I1];
'IF'R<-E1'THEN'T:=KB(I1,B1,ND1,B1K,-BR)-W[I1];
'IF'ABS(T)>E1'THEN'
#BEGIN'BL:='TRUE';AL:=K1*R;SIG:=0;IOO:=1'END''END';
'IF'DT<R<-E1'THEN''BEGIN'PR:=-DT1R;'IF'PR<SIG'THEN''BEGIN'
IOO:=1;SIG:=PR;AL:=K1*R'END''END''END'MINCIG;
'PROCEDURE'OPTDT;'BEGIN'R:=K1*R;'IF'ABS(DT)<E1'THEN''BEGIN'
DN1:=ND1;DN2:=ND2;
T:=0;'IF'R<-E1'THEN'T:=KB(J,D2,ND2,D2K,BR)-X[IJ];'IF'R>E1'THEN'
T:=KB(J,D1,ND1,D1K,)-X[IJ];'IF'ABS(T)=E1'THEN''BEGIN'
SIG:=0;AL:=K1*R;J00:=J;BL:='TRUE'END';ND1:=DN1;ND2:=DN2'END';
'IF'DT<R<-E1'THEN''BEGIN'PR:=-DT1R;'IF'PR<SIG'THEN''BEGIN'
J00:=J;SIG:=PR;AL:=K1*R'END''END';KJK:'END'OPTDT;
'PROCEDURE'HXJ;'BEGIN'XH:=0;'IF'DT=E1'THEN'
LH:=KB(J,D1,ND1,D1K,)-X[IJ];'IF'DT<-E1'THEN'
LH:=KB(J,D2,ND2,D2K,BR)-X[IJ];X[IJ]:=X[IJ]+TT*LH'END'HXJ;
'PROCEDURE'RCTOL(J00,P);'VALUE'J00;'INTEGER'J00;
'ARRAY'P;'BEGIN'JJ:=0;
'FOR'I:=1'STEP'1'UNTIL'ROP1'DO'P[I]:=0;
J1:=NCA[J00+1]-NCA[J00];
'FOR'I:=1'STEP'1'UNTIL'J1'DO''BEGIN'I2:=NCA[J00]+I;
PLAOR1(A[I2],J2);
'FOR'II:=1'STEP'1'UNTIL'ROP1'DO''IF'J2=BS1[I1]

```

```

'THEN' 'BEGIN' R:=A[I2]; 'FOR' I1:=1 'STEP' 1 'UNTIL' ROP1 'DO'
P[I1]:=P[I1]+R*A[I1,I1]; JJ:=JJ+1; 'IF' JJ=ROP1 'THEN'
'GOTO' M2 'ELSE' 'GOTO' M1 'END'; M1: 'END' I; M2: 'END' RCTOL;
'PROCEDURE' MCTOL(J0,J00,AOP,N,K);
'VALUE' J0,J00,N; 'INTEGER' J0,J00,N,K;
'ARRAY' AOP;
'BEGIN' 'IF' N=1 'THEN' RCTOL(J00,P) 'ELSE' RAZJO(J00,P);
R:=P[J0]; 'FOR' J:=1 'STEP' 1 'UNTIL' K 'DO'
'BEGIN' PR:=AOP[J,J1R;
AOP[J0,J1:=PR;
'FOR' I:=1 'STEP' 1 'UNTIL' J0-1,J0+1 'STEP' 1 'UNTIL' K 'DO'
AOP[I,J1:=AOP[I,J1-PR*P[I1] 'END' MSTOL;
'PROCEDURE' RCTP(I0,P); 'VALUE' IO;
'INTEGER' IO; 'ARRAY' P; 'BEGIN'
'FOR' I:=1 'STEP' 1 'UNTIL' ROP1 'DO' P[I]:=0;
'FOR' J:=1 'STEP' 1 'UNTIL' ROP1 'DO'
'BEGIN' I1:=BS[I]; J1:=NCA[I1+1]-NCA[I1];
'FOR' I:=1 'STEP' 1 'UNTIL' J1 'DO'
'BEGIN' I2:=NCA[I1]+I; PLAOR1(A[I2],J2);
'IF' J2=IO 'THEN' 'BEGIN'
R:=A[I2]; 'FOR' I1:=1 'STEP' 1 'UNTIL' ROP1
'DO' P[I1]:=P[I1]+R*A[I1,I1]
'END'; 'IF' J2=IO 'THEN' 'GOTO' CI 'END' I;
CI: 'END' J 'END' RCTP;

'PROCEDURE' MCTP(IOO,IO,AOP,N,K);
'VALUE' IOO,IO,N; 'INTEGER' IOO,IO,N,K;
'ARRAY' AOP; 'BEGIN' 'IF' N=1 'THEN' RCTP(IO,P) 'ELSE' RIOA2(IO,P);
R:=P[IOO]; 'FOR' J:=1 'STEP' 1 'UNTIL' K
'DO' 'BEGIN' PR:=AOP[J,IOO]/R;
AOP[J,IOO]:=PR;
'FOR' I:=1 'STEP' 1 'UNTIL' IOO-1,IOO+1 'STEP' 1 'UNTIL' K 'DO'
AOP[I,I1:=AOP[I,I1-PR*P[I1] 'END' J 'END' MCTP;
'PROCEDURE' DOB(IO,J00,AOP,N,K);
'VALUE' IO,J00,N; 'INTEGER' IO,J00,N,K;
'ARRAY' AOP; 'BEGIN' 'IF' N=1 'THEN' 'BEGIN' RCTP(IO,P);
RCTOL(J00,W) 'END'
'ELSE' 'BEGIN' RIOA2(IO,P); RAZJO(J00,6) 'END'; AL:=1/AL; I1:=K+1;
'FOR' I:=1 'STEP' 1 'UNTIL' K 'DO' 'BEGIN' R:=W[I]*AL; AOP[I,I1]:=-R;
AOP[I,I1]:=-P[I]*AL; 'FOR' J:=1 'STEP' 1 'UNTIL' K 'DO' AOP[I,J1:=
AOP[I,J1+R*P[J] 'END' I; AOP[I,I1]:=AL 'END' DOB;
'PROCEDURE' YDAL(IOO,J0,AOP,K); 'VALUE' IOO,J0; 'INTEGER' IOO,J0,K;
'ARRAY' AOP; 'BEGIN' 'IF' IOO=K 'THEN' 'GOTO' M1;
'FOR' I:=1 'STEP' 1 'UNTIL' K 'DO'
'BEGIN' R:=AOP[I,IOO]; AOP[I,IOO]:=AOP[I,K1]; AOP[I,K1]:=R 'END';
M1: 'IF' J0=K 'THEN' 'GOTO' M2; 'FOR' I:=1 'STEP' 1 'UNTIL' K 'DO' 'BEGIN'
R:=AOP[J0,I1]; AOP[J0,I1:=AOP[K,I1]; AOP[K,I1]:=R 'END';
M2: PR:=1/AOP[K,K1]; 'FOR' I:=1 'STEP' 1 'UNTIL' K-1 'DO' 'BEGIN'
R:=AOP[I,K1]*PR; 'FOR' J:=1 'STEP' 1 'UNTIL' K-1 'DO'
AOP[I,J1:=AOP[I,J1]-AOP[K,J1]*R 'END' 'END' YDAL;
'PROCEDURE' RIOA2(IO,P); 'VALUE' IO; 'INTEGER' IO;
'ARRAY' P; 'BEGIN' RCTP(IO,P);
'FOR' J:=1 'STEP' 1 'UNTIL' ROP2 'DO' 'BEGIN' I1:=BS[ROP1+J];
J1:=NCA[I1+1]-NCA[I1]; JJ:=0; R:=PR=0;
'FOR' I:=1 'STEP' 1 'UNTIL' J1 'DO'
'BEGIN' I2:=NCA[I1]+I; PLAOR1(A[I2],J2);
'IF' J2=IO 'THEN' PR:=A[I2];
'FOR' I1:=1 'STEP' 1 'UNTIL' ROP1 'DO' 'IF' J2=BS1[I1] 'THEN' 'BEGIN'
R:=R*P[I1]*A[I2]; JJ:=JJ+1; 'GOTO' M1 'END'; M1: 'IF' JJ=ROP1 'AND'
J2=GE 'IO' 'THEN' 'GOTO' M2 'END' I; M2: V[J1]:=PR 'END' J;
'FOR' I:=1 'STEP' 1 'UNTIL' ROP2 'DO' 'BEGIN' R:=0;
'FOR' J:=1 'STEP' 1 'UNTIL' ROP2 'DO' R:=R+V[J]*A2[J,I];
P[I1]:=R 'END' 'END' A2;
'PROCEDURE' RAZJO(J0,P); 'VALUE' J0; 'INTEGER' J0;
'ARRAY' P; 'BEGIN' RCTOL(J0,P);
'FOR' I:=1 'STEP' 1 'UNTIL' ROP2 'DO' V[I1]:=0; BL:= 'FALSE';
'FOR' J:=1 'STEP' 1 'UNTIL' ROP1 'DO' 'BEGIN' R:=P[J]; I1:=BS[J];
MON: J1:=NCA[I1+1]-NCA[I1]; JJ:=0;
'FOR' I:=1 'STEP' 1 'UNTIL' J1 'DO'
'BEGIN' I2:=NCA[I1]+I; PLAOR1(A[I2],J2);
'FOR' I1:=1 'STEP' 1 'UNTIL' ROP2 'DO' 'IF' J2=BS1[ROP1+I1] 'THEN'
'BEGIN' V[I1]:=V[I1]-R*A[I2]; JJ:=JJ+1; 'IF' JJ=ROP2 'THEN'
'GOTO' M2 'ELSE' 'GOTO' M1 'END' I; M1: 'END' I; M2: 'IF' BL 'THEN'
'GOTO' ZJ; 'IF' J=ROP1 'THEN' 'BEGIN' I1:=J0; R:=-1; BL:= 'TRUE';
'GOTO' MON 'END' 'END' J; ZJ: 'FOR' I:=1 'STEP' 1 'UNTIL' ROP2 'DO'
'BEGIN' R:=0; 'FOR' J:=1 'STEP' 1 'UNTIL' ROP2 'DO'
R:=R+A2[I,J]*V[J]; P[I1]:=R 'END' I 'END' RAZJO;

```

```

'PROCEDURE/MAXIX;'BEGIN/'FOR'I:=1'STEP'1
'UNTIL'ROP2'DO'VCIJ:=0;
'FOR'J:=1'STEP'1'UNTIL'ROP1'DO''BEGIN'I1:=BS[J];
J1:=NCAI1+1;NCAI1;JJ:=0;
'FOR'I:=1'STEP'1'UNTIL'J1'DO''BEGIN'
I2:=NCAI1+1;PLAOR1(A[I2],J2);
'FOR'II:=1'STEP'1'UNTIL'ROP2'DO'

'IF'J2=BS1[ROP1+I1]THEN'
'BEGIN/VCIJ:=VCIJ+ACI2]*A1[J,I00];
JJ:=JJ+1;'IF'JJ=ROP2'THEN'GOTO'KJ'ELSE''GOTO'IKI'END';
IKI:'END'I;KJ:'END'J;R:=0;'FOR'I:=1'STEP'1'UNTIL'ROP2'DO'
'IF'ABS(VCIJ)>R'THEN''BEGIN'IX:=I;
R:=ABS(VCIJ)'END''END'MAXIX;
'PROCEDURE/MENIX;'BEGIN/'FOR'I:=1'STEP'1'UNTIL'ROP2'DO'
'BEGIN'R:=0;'FOR'J:=1'STEP'1'UNTIL'ROP2
'DO'R:=R-A2[I,J]*VCIJ;
PCIJ:=R'END';'FOR'I:=1'STEP'1'UNTIL'ROP2'DO'A2[I,IX]:=P[I];
NC:=BS1[ROP1+IX];MCTP(I00,NC,A1,1,ROP1);
BS1[ROP1+IX]:=BS1[I00];BS1[I00]:=NC;I00:=ROP1+IX'END'MENIX;
'PROCEDURE/UJOAJ(I1,B);'INTEGER'I1;'BOOLEAN'B;'BEGIN'
IJ:=ROP1;
DT:=R:=0;JJ:=0;J1:=NCAI1+1-NCAI1;
'FOR'I:=1'STEP'1'UNTIL'J1'DO'
'BEGIN'I2:=NCAI1+1;PLAOR1(A[I2],J2);
'FOR'II:=1'STEP'1'UNTIL'IJ'DO'
'IF'J2=BS1[I1]THEN''BEGIN''IF''NOT'BL'AND'II'LE'ROP1'THEN'
R:=R+A1[IJ,I1]*ACI2;JJ:=JJ+1;
'IF'B'THEN'DT:=DT+U[I1]*ACI2;
'IF'JJ=IJ'THEN''GOTO'G2'ELSE''GOTO'G1'END';
G1:'END'I;G2:'END'UJOAJ;
'PROCEDURE/SIGIO;'BEGIN'I00:=0;
SIG:=18;BL:='FALSE';'IF'ROP2=0'THEN'
'BEGIN'RCTP(I0,P);
'FOR'I:=1'STEP'1'UNTIL'K'DO''BEGIN'R:=-K1+P[I];
VCIJ:=-P[I];
'IF''NOT'BL'THEN'MINCIG'END'I;
'GOTO'NI'END''ELSE'RIOA2(I0,P);
BL:='FALSE';'FOR'I:=ROP1+1'STEP'1'UNTIL'K
'DO''BEGIN'R:=-P[I]-ROP1;
VCIJ:=R;R:=-K1+R;'IF''NOT'BL'THEN'MINCIG'END'I;
'FOR'J:=1'STEP'1'UNTIL'ROP1'DO''BEGIN'I1:=BS[J];R:=0;JJ:=0;
BL:='FALSE';J1:=NCAI1+1-NCAI1;
'FOR'I:=1'STEP'1'UNTIL'J1'DO'
'BEGIN'I2:=NCAI1+1;PLAOR1(A[I2],J2);
'IF'J2=I0'THEN'R:=R-ACI2;
'FOR'II:=ROP1+1'STEP'1'UNTIL'K'DO'
'IF'BS1[II]=J2'THEN''BEGIN'
R:=R-VCIJ]*ACI2;JJ:=JJ+1;'GOTO'YA'END';
YA:'IF'JJ=ROP2'AND'J2'GE'I0
'THEN'GOTO'AY'END'I;AY:P[JJ]:=R'END'J;
'FOR'I:=1'STEP'1'UNTIL'ROP1'DO'
'BEGIN'R:=0;'FOR'J:=1'STEP'1'UNTIL'ROP1
'DO'R:=R+PCIJ]*A1[J,I1];VCIJ:=R;
R:=K1+R;'IF''NOT'BL'THEN'MINCIG'END'I;
NI:ND1:=ND2:=NC:=1;J00:=0;
'FOR'J:=1'STEP'1'UNTIL'N'DO'
'BEGIN/'FOR'I:=1'STEP'1'UNTIL'K'DO'
'IF'J=BS1[I]THEN''GOTO'KEJ;DT:=-KB(J,C,NC,CK,);R:=0;JJ:=0;
J1:=NCAI1+1-NCAI1;'FOR'I:=1'STEP'1'UNTIL'J1'DO''BEGIN'
I2:=NCAI1+1;PLAOR1(A[I2],J2);'IF'J2=I0'THEN'R:=R+ACI2;
'FOR'II:=1'STEP'1'UNTIL'K'DO''IF'BS1[II]=J2'THEN''BEGIN'
R:=R+VCIJ]*ACI2;DT:=DT+U[I1]*ACI2;JJ:=JJ+1;
'GOTO'M5'END'II;
M5:'IF'JJ=K'AND'J2'GE'I0'THEN'GOTO'M6;
'END'I;M6:OPTD;HXJ;KEJ:'END'J;
'FOR'I:=1'STEP'1'UNTIL'K'DO'X[BS[I]]:=X[BS[I]]+TT*LOP[I];

'IF'(1-TT)*BT<EPS'THEN'GOTO'KON'END'SIGIO;
'PROCEDURE/START;
'BEGIN/'FOR'I:=1'STEP'1'UNTIL'M'DO'WCIJ:=0;
'BEGIN/PLAOR1(A[I],I1);WCIJ:=WCIJ+A[I]*X[J]'END';I1:=0;
'FOR'J:=1'STEP'1'UNTIL'N'DO''FOR'I:=NCAI1+1
'STEP'1'UNTIL'NCAI1+1'DO'
ND1:=ND2:=1;'FOR'I:=1'STEP'1'UNTIL'M'DO''BEGIN'
CR:=KB(I,B1,ND1,B1K,-BR);PR:=KB(I,B2,ND2,B2K,BR);R:=0;
'IF'WCIJ>PR'OR'WCIJ<CR'THEN'R:=(CR+PR)/2-WCIJ;
'IF'ABS(R)>E1'THEN''BEGIN'I1:=I1+1;
J1:=NCAI1+1+I1;ACI1:=R;

```

```

PLA031(A[CJ1],I)'END'/'E'D'I'/'IF'I1>0'THEN'
/BEGIN'X[N+1]:=1;CK:=CK+1;C[CK]:=-BR;PLA031(C[CK],N+1);
D2K:=D2K+1;D2[D2K]:=1;PLA031(D2[D2K],N+1);
NCA[N+2]:=NCA[N+1]+I1;N:=N+1/'END'/'START';
IT:=K:=ROP1:=ROP2:=0;E1:=N-5;BR:=16;
/IF/EPS<0'THEN'/'BEGIN'/'START';EPS:=-EPS/'END';
HIT:=KROP;BLOP;SHAG;'IF'I0=0'THEN'/'GOTO'MI0;
/IF'ROP2=0'THEN'
/GOTO'ZK;'IF'J0>ROP1'THEN'/'GOTO'JOB2;PR:=0;BL:='FALSE';
/FOR'J:=1'STEP'1'UNTIL'ROP2'DO'/'BEGIN'I1:=BS[ROP1+J];
UJOAJ(I1,'FALSE');P[CJ]:=R;
/IF'ABS(R)>PR'THEN'/'BEGIN'PR:=ABS(R);
JX:=J/'END'/'END'J;'IF'ABS(PR)<E1'THEN'/'GOTO'ZK;
R:=LOP[J0];LOP[J0]:=LOP[ROP1+JX];LOP[ROP1+JX]:=R;
/FOR'I:=1'STEP'1'UNTIL'ROP2'DO'/'BEGIN'R:=0;
/FOR'J:=1'STEP'1'UNTIL'ROP2'DO'
R:=R-P[CJ]*A2[CJ,I];V[CJ]:=R/'END';
/FOR'I:=1'STEP'1'UNTIL'ROP2'DO'
A2[CJ,I]:=V[I];NC:=BS[ROP1+JX];
MCTOL(J0,NC,A1,1,ROP1);BS[ROP1+JX]:=BS[J0];
BS[J0]:=NC;J0:=ROP1+JX;
JOB2:SIGJ0;'IF'J00>0'THEN'
/BEGIN'MCTOL(J0-ROP1,J00,A2,0,ROP2);
BS[J0]:=J00;'GOTO'KZ/'END';
/ELSE'/'BEGIN'/'IF'I00=ROP1'THEN'
/BEGIN'YDAL(I00-ROP1,J0-ROP1,A2,ROP2);
ROP2:=ROP2-1;BS[J0]:=BS[K];
BS1[I00]:=BS1[K];K:=K-1;'GOTO'KZ/'END';
MAXIX;MENIX;YDAL(I00-ROP1,J0-ROP1,A2,ROP2);ROP2:=ROP2-1;
BS[J0]:=BS[K];BS1[I00]:=BS1[K];K:=K-1;'GOTO'KZ/'END';
ZK:SIG:=16;I00:=J00:=0;BL:='FALSE';NC:=ND1:=ND2:=1;
/FOR'I:=1'STEP'1'UNTIL'ROP1'DO'
U[CJ]:=0;'FOR'I:=1'STEP'1'UNTIL'ROP1'DO'/'BEGIN'I1:=BS[I];
NC:=1;CR:=KB(I,C,NC,CK,0);/IF'CR=0'THEN'/'GOTO'KI;
/FOR'J:=1'STEP'1'UNTIL'ROP1'DO'
U[CJ]:=U[I]+CR*A1[I,J];KI:'END'I;
NC:=1;'FOR'J:=1'STEP'1'UNTIL'N'DO'
/BEGIN'/'FOR'I:=1'STEP'1'UNTIL'K'DO'
/IF'J=BS[I] THEN'/'GOTO'UJ;UJOAJ(J,'TRUE');
DT:=DT-KB(J,C,NC,CK,0);
/IF'NOT'BL THEN'OPTDT;HXJ;UJ:'END'J;
/FOR'I:=1'STEP'1'UNTIL'K'DO'X[BS[I]]:=X[BS[I]]+TT*LOP[I];
/IF'(1-TT)*BT<EPS THEN'/'GOTO'KON;'IF'BL THEN'/'GOTO'UA;
/FOR'I:=1'STEP'1'UNTIL'ROP1'DO'/'BEGIN'R:=K1*A1[J0,I];
MINCIG;'IF'BL THEN'/'GOTO'UA/'END'I;
UA:'IF'I00>0 THEN'/'BEGIN'YDAL(I00,J0,A1,ROP1);
BS[J0]:=BS[ROP1];
BS1[I00]:=BS1[ROP1];

/FOR'I:=ROP1'STEP'1'UNTIL'K-1'DO'/'BEGIN'
BS[I]:=BS[I+1];BS1[I]:=BS1[I+1]'END';
ROP1:=ROP1-1;K:=K-1/'END'/'ELSE'
/BEGIN'MCTOL(J0,J00,A1,1,ROP1);BS[J0]:=J00/'END';/'GOTO'KZ;
MI0:SIGI0;'IF'J00>0 THEN'/'BEGIN'/'IF'K<M0 THEN'/'BEGIN'
DOB(I0,J00,A1,1,K);ROP1:=ROP1+1/'END'/'ELSE'/'BEGIN'
DOB(I0,J00,A2,0,ROP2);ROP2:=ROP2+1/'END';K:=K+1;BS[K]:=J00;
BS1[K]:=I0;'GOTO'KZ/'END';
/IF'ROP2=0 THEN'/'GOTO'MA;'IF'I00>ROP1 THEN'
/BEGIN'MCTP(I00-ROP1,I0,A2,0,ROP2);
BS1[I00]:=I0;'GOTO'KZ/'END';MAXIX;
/IF'ABS(R)>E1 THEN'/'BEGIN'MENIX;
MCTP(I00-ROP1,I0,A2,0,ROP2);BS1[I00]:=I0;'GOTO'KZ/'END';
MA:MCTP(I00,I0,A1,1,ROP1);
BS1[I00]:=I0;KZ:IT:=IT+1;R:=0;NC:=1;
/IF'ABS(C[CK]+BR)<E1 THEN'/'BEGIN'
/IF'ABS(X[N])<E1 THEN'N:=N-1/'END';
/FOR'I:=1'STEP'1'UNTIL'N
/DO'R:=R+KB(I,C,NC,CK,0)*X[I];;PRINT
(NEWLINE,'IT=',IT,'CX=',R,'BETA=',BT,'ОПОРА=',K);
/GOTO'HIT;
KON:R:=0;NC:=1;
/FOR'I:=1'STEP'1'UNTIL'N'DO'R:=R+KB(I,C,NC,CK,0)*X[I];
PRINT(NEWLINE,'IT=',IT,
'CX=',R,'BETA=',BT,'ОПОРА=',K);
PRINT(NEWLINE,'ОПТИМАЛЬНЫЙ ПЛАН');
/PRINT('Q(F11.6,X))';/FOR'I:=1'STEP'1'UNTIL'N'DO'X[I];
/END'DPLAY;

```

ЛИТЕРАТУРА

1. Булавский В. А., Звягина Р. А., Яковлева М. А. Численные методы линейного программирования.— М., 1977.—368 с.
2. Габасов Р., Кириллова Ф. М. Методы оптимизации.— 2-е изд., перераб. и доп.— Минск, 1981.—350 с.
3. Габасов Р., Кириллова Ф. М. Построение субоптимальных планов транспортной задачи.— Изв. АН СССР. Техн. кибернетика, 1975, № 6, с. 38—42.
4. Габасов Р., Кириллова Ф. М. Прямой и двойственный опорные методы решения общей задачи линейного программирования.— Автоматика и телемеханика, 1977, № 6, с. 85—96.
5. Габасов Р., Кириллова Ф. М. Методы линейного программирования. Минск, 1977. Ч. 1. Общие задачи.—176 с.
6. Габасов Р., Кириллова Ф. М. Методы линейного программирования. Минск, 1978. Ч. 2. Транспортные задачи.—239 с.
7. Габасов Р., Кириллова Ф. М., Костюкова О. И. Метод решения общей задачи линейного программирования.— ДАН БССР, 1979, т. 23, № 3, с. 197—200.
8. Габасов Р., Кириллова Ф. М. Методы линейного программирования: Ч. 3. Специальные задачи.— Минск, 1980.—368 с.
9. Габасов Р., Кириллова Ф. М., Костюкова О. И. К построению алгоритма первого порядка для решения задач оптимального управления.— ДАН СССР, 1980, т. 254, № 3, с. 562—566.
10. Габасов Р., Кириллова Ф. М., Костюкова О. И. Приближенные методы решения общей задачи линейного программирования.— ДАН БССР, 1980, т. 24, № 8, с. 680—683.
11. Габасов Р., Кириллова Ф. М., Костюкова О. И. Двухшаговый метод решения общей задачи ЛП.— ДАН БССР, 1980, т. 24, № 10, с. 873—876.
12. Габасов Р., Кириллова Ф. М., Костюкова О. И. Конечный прямой точный алгоритм линейного программирования.— ДАН БССР, 1981, т. 25, № 4, с. 301—304.
13. Габасов Р., Кириллова Ф. М., Костюкова О. И. Конечный двойственный точный алгоритм линейного программирования.— ДАН БССР, 1981, т. 25, № 7, с. 586—589.
14. Габасов Р., Шилкина Е. И. Прямой точный метод решения одного класса минимаксных задач.— ДАН БССР, 1981, т. 25, № 11, с. 971—973.
15. Габасов Р., Кириллова Ф. М., Костюкова О. И. К теории приближенных методов решения общей задачи линейного программирования.— Весці АН БССР. Сер. фіз.-мат. навук. 1981. № 3, с. 5—10.

16. Габасов Р., Кириллова Ф. М., Костюкова О. И. К методам решения общей задачи линейного программирования.— Минск, 1977. Препринт / Ин-т математики АН БССР; № 14(30), с. 22.

17. Габасов Р., Кириллова Ф. М., Костюкова О. И. Опорные методы решения больших задач линейного программирования.— Минск, 1978. Препринт / Ин-т математики АН БССР; № 11(43), с. 32.

18. Габасов Р., Кириллова Ф. М., Костюкова О. И. Метод решения общей задачи линейного программирования.— ДАН БССР, 1979, т. 23, № 3, с. 197—200.

19. Габасов Р., Кириллова Ф. М., Костюкова О. И. Некоторые модификации метода декомпозиции Данцига — Вулфа.— Минск, 1979. Препринт / Ин-т математики АН БССР; № 13(69), с. 20.

20. Габасов Р., Кириллова Ф. М., Костюкова О. И. Исследование задач линейного программирования с обобщенными ограничениями.— Минск, 1979. Препринт / Ин-т математики АН БССР; № 19(75), 24 с.

21. Габасов Р., Глушенков В. С. Метод решения общей задачи линейного программирования.— ДАН БССР, 1982, т. 26, № 10, с. 589—592.

22. Глушенков В. С. Численный эксперимент по решению бесконечномерных задач линейного программирования.— В кн.: Проблемы оптимального управления. Минск, 1981, с. 365—369.

23. Глушенков В. С. Построение алгоритмов линейного программирования с использованием двойственной квазивырожденности.— Деп. в ВИНТИ, 1 июля 1981 г., № 3225-81.

24. Глушенков В. С., Гневко С. В., Костюкова О. И. Численный эксперимент по методам решения общей задачи линейного программирования.— Вестн. БГУ им. В. И. Ленина. Сер. 1, 1982, № 2, с. 33—35.

25. Гольштейн Е. Г., Юдин Д. Б. Новые направления в линейном программировании.— М., 1966.—524 с.

26. Данциг Дж. Линейное программирование, его обобщения и применения.— М., 1966.— 600 с.

27. Канторович Л. В. Математические методы организации и планирования производства.— Л., 1939.—с. 67.

28. Костюкова О. И., Тонев П. Т. Прямой и двойственный методы решения динамической транспортной задачи.— В кн.: Проблемы оптимального управления. Минск, 1981, с. 285—305.

29. Костюкова О. И., Тятюшкин А. И. Численный эксперимент по одному методу решения общей задачи линейного программирования.— Вестн. БГУ им. В. И. Ленина. Сер. 1, 1982, № 2, с. 35—38.

30. Лэсдон Л. С. Оптимизация больших систем.— М., 1975.—432 с.

31. Поляк Б. Т., Третьяков Н. В. Об одном итерационном методе линейного программирования и его экономической интерпретации.— Экономика и мат. методы, 1972, т. 8, вып. 5, с. 740—751.

32. Романовский И. В. Алгоритмы решения экстремальных задач.— М., 1977.—352 с.

33. Сенько А. А. Алгоритмы решения интервальной задачи линейного программирования.— Весці АН БССР. Сер. фіз.-мат. навук, 1982. № 5, с. 21—26.

34. Федоренко Р. П. Приближенное решение задач оптимального управления.— М., 1978.—488 с.

35. Фаддеев Д. К., Фаддеева В. Н. Вычислительные методы линейной алгебры.— М., 1960.— 734 с.

36. Фам Тхе Лонг. Двухшаговый метод с декомпозицией опоры

для решения больших задач линейного программирования.— Весті АН БССР. Сер. фіз.-мат. наук, 1981, № 4, с. 45—49.

37. *Фам Тхе Лонг*. К многошаговым методам линейного программирования.— ДАН БССР, 1981, т. 25, № 9, с. 784—787.

38. *Фам Тхе Лонг*. Численный эксперимент по сравнению методов решения общей задачи линейного программирования.— Вестн. БГУ им. В. И. Ленина. Сер. 1, 1982, № 2, с. 67—68.

39. *Химмельблау Д.* Прикладное нелинейное программирование.— М., 1975.—536 с.

40. *Charnes A., Granot D., Granot G.* A primal algorithm for the interval linear programming problems.— Linear Algebra and its Applications, 1977, v. 17, N 1.

41. *Gabasov R., Kirillova F. M.* New methods of linear programming and their application to optimal control.— Proc. IFAC Workshop on Appl. Nonlinear Program., Pergamon Press, 1979.

42. *Gabasov R., Kirillova F. M.* Constructive Methods of Solving Extremal Problems.— Proc. III Polish-English Seminar on real time processes control., Poland, 1980.

43. *Gabasov R., Kirillova F. M.* Constructive Methods of Parametric and Functional Optimization.— Preprints of IFAC, 8th Triennial World Congress, 1981, August 24—28, Kyoto, Japan, v. 4, sessions 17—20, p. 111—116.

44. *Kirillova F. M., Kostjukova O. I. and Gabasov R.* Adaptive methods for solving large problems of linear programming: Symposium Optimization Methods (applied aspects).— Preprints of IFAC-IFOPS, 1979, X, p. 15—19, Varna, Bulgaria.

*Рафаил Габасов
Фаина Михайловна Кириллова
Александр Иванович Тятюшкин*

КОНСТРУКТИВНЫЕ МЕТОДЫ ОПТИМИЗАЦИИ

Ч. 1. Линейные задачи

*Зав редакцией Б. С. Мельников
Редактор Т. А. Акулович
Мл. редактор М. Г. Хаятина
Оформление Е. Ф. Ясногородского
Худож. редактор С. В. Баленок
Техн. редактор В. П. Безбородова
Корректор Л. В. Лебедева*

ИБ № 666

Сдано в набор 14.12.82. Подписано в печать 05.12.83.
АТ 08751. Формат 84×108^{1/32}. Бумага тип. № 1.
Гарнитура литературная. Высокая печать. Усл.
печ. л. 11,34. Усл. кр.-отт. 11,34. Уч.-изд. л. 11,92.
Тираж 1950 экз. Заказ 1330. Цена 2 р. 10 к.

Издательство «Университетское» Госкомиздата
БССР. 220048. Минск, проспект Машерова, 11. Ор-
дена Трудового Красного Знамени типография
издательства ЦК КПБ. 220041. Минск, Ленинский
пр. 79.

**В 1984 году
издательство
«Университетское»
выпустит:**

Г а б а с о в Р., К и р и л л о в а Ф. М.
Конструктивные методы оптимизации:
В 5-ти частях. Ч. 2. Задачи управления.—
12 л.— Рус. яз.— 2 р. (ориентировочно).

В книге рассматриваются линейные задачи оптимального управления в двух классах допустимых управлений: импульсных и кусочно-непрерывных. Для каждого случая развивается подход, обоснованный в ч. 1 (1983 г.). Последовательно изучаются задачи терминального управления, со смешанными и фазовыми ограничениями, алгоритмы оптимизации нестационарных и сингулярно возмущенных систем. Приведены результаты численного эксперимента.

Рассчитана на научных работников, инженеров, занятых прикладными задачами оптимизации.

