И. В. Семушин

Практикум по методам оптимизации

Компьютерный курс: учебное пособие для вузов

Е. Е. Курышова

Разработка компьютерной версии

Ульяновск 2005

Федеральное агентство по образованию РФ
Ульяновский государственный университет

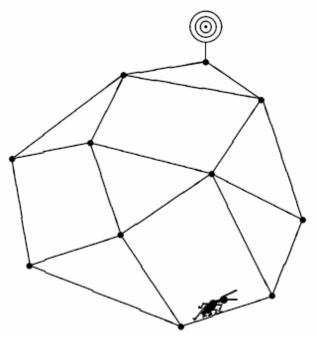
И. В. Семушин

Практикум по методам оптимизации

Компьютерный курс

Е. Е. Курышова

Разработка компьютерной версии



Ульяновск 2005

Оглавление

Предисловие

В целом, курс методов оптимизации охватывает задачи, существенно различающиеся по двум основным признакам: линейность / нелинейность и наличие / отсутствие ограничений. В соответствии с этим, базовый курс содержит линейное программирование и нелинейное (выпуклое) программирование, причем последний раздел распадается на две части: безусловная минимизация и задачи на условный экстремум. Некоторые методы нелинейного программирования используют линейное программирование как часть своих алгоритмов. В связи с этим, а также по методическим соображениям, в данном учебном пособии большее внимание уделено именно линейному программированию. Это оправдано также тем, что линейное программирование как инструмент решения реальных задач планирования с ограниченными ресурсами встречается, вероятно, чаще, чем другие методы.

К линейному программированию как инструменту решения оптимизационных задач обращаются многие специалисты. Основные его пользователи – те, кто решает проблемы управления, организации и планирования производства, или использования ресурсов. Реализацией на ЭВМ соответствующих математических пакетов занимаются математики, а адаптацией и сопровождением пакетов – специалисты по информационным системам и технологиям. Для всех этих категорий специалистов весьма важно понимание линейного программирования «изнутри». Оно не дается «слепым» использованием готовых программных продуктов, а приобретается только лишь через личный опыт разработки компьютерных программ, реализующих тот или иной алгоритм.

Таким образом, эта книга предназначена для студентов и специалистов, заинтересованных глубоко изучить сначала – линейное программирование и затем – методы нелинейной оптимизации, с акцентом на компьютерную реализацию методов и алгоритмов.

Линейное программирование – хорошо разработанный раздел прикладной математики. По нему имеется обширная научная и учебная литература – как отечественная, так и переводная. В своем большинстве, литература дает основные теоретические сведения и лишь иногда – практические рекомендации, как запрограммировать на ЭВМ тот или иной вариант симплекс-метода. Примером редких, очень эффективных учебных изданий с компьютерными программами является книга [4], вышедшая в издательстве «Радио и Связь» в 1989 году.

Новым, основанным на технологиях, методам обучения и преподавания наук повсюду в мире уделяется все возрастающее внимание [16]. Обучение, основанное на программных проектах, эффективно по многим причинам [17]. Кроме основного результата – понимания предмета «изнутри», оно совершенствует компьютерную культуру специалиста, особенно важную в настоящее время, поскольку требует изощренного программирования и развивает общие способности к логическому и численному анализу при поиске решений математически ориентированных проблем. Однако опыт преподавания показывает, что в настоящее время нет учебных пособий, рассчитанных на реализацию именно такого – проектно ориентированного – метода обучения. Максимальное приближение лабораторных работ студентов к задаче полноценной самостоятельной компьютерной реализации численного метода делает эти работы похожими на учебные программные проекты, отличающиеся строгой демонстрацией, как самих разновидностей изучаемого метода, так и всех деталей соответствующих алгоритмов.

При организации и проведении лабораторного практикума на ЭВМ всегда возникает проблема индивидуализации заданий. Многие пособия в этом вопросе идут по наиболее простому пути – размножают задания по одной и той же выполняемой работе за счет различий в исходных данных. То, что это малополезно, вполне очевидно. Делать лишь небольшие и количественные различия в заданиях – значит отказаться то всех указанных выше выгод проектно ориентированного обучения.

Цель данного пособия – снабдить специалистов и студентов полным, тщательно выверенным и внутренне завершенным (то есть не требующим обращения к другим источникам) набором возможных вариантов симплекс-метода для решения задач линейного программирования. Максимально размноженные на этой основе учебные задания отличаются не по вводимым для решения задачам, а по алгоритмам, подлежащим компьютерной программной реализации. Для этого после кратких вводных определений (разд. 1) собраны базовые теоретические факты, относящиеся к стандартной задаче линейного программирования (разд. 2). Далее основное внимание сосредоточено на детальном и строгом обосновании каждой разновидности симплекс-метода во всевозможных схемах реализации: базовой – (разд. 3), двойственной – (разд. 4) и модифицированной – (разд. 5). Особые случаи функционирования алгоритмов рассмотрены в разд. 6. Следующие разделы являются перечислением учебных заданий по линейному программированию (разд. 7), сборником тестовых задач (разд. 8) и перечнем учебных проектов по нелинейной оптимизации (разд. 9).

И. В. Семушин

Руководство по использованию электронного учебного пособия

Для ввода, анализа и обработки данных в электронном учебном пособии используются текстовые поля, радиокнопки и кнопки.

Специальные текстовые поля позволяют вводить данные для дальнейшей их обработки в программе:

- поле для ввода числовых значений.

Радиокнопки используются для выбора ведущего элемента на конкретном шаге примера:

- выбор одного значения из нескольких предложенных.

Для очистки и проверки полей, а также для передачи данных в программу существуют следующие кнопки:

Clear - очистка текстового поля на данном шаге;

Enter - передача данных в программу;

Next - переход на следующий этап решения примера;

Repeat - возврат на начало решения примера;

Start - подготовка примера к начальному этапу решения: очистка текстовых полей и радиокнопок, помещение кнопок в стартовое состояние.

1
Общие определения

> Материал этого раздела носит вводный характер. Его цель заключается в том, чтобы напомнить те основные концепции, которые студент может знать из курсов «Математический анализ» и «Линейная алгебра» и которые он должен полностью осознать, прежде чем двинуться дальше.

Пусть даны любые два объекта x и y, тогда **упорядоченная пара** есть (x, y), где x определяется как первый элемент и y как второй элемент этой пары. Множество упорядоченных пар называется **бинарным отношением** или, иначе, **соответствием**. Например, если множество упорядоченных пар вещественных чисел, чьи значения одинаковы, обозначено A, мы записываем это соответствие как $A = \{(x, y) \in \mathbb{R} \times \mathbb{R} : x = y\}$. Если A есть некоторое соответствие, то **область определения** соответствия A, сокращенно обозначаемая как *dmn* A, дается выражением

$$dmn\, A = \{ x : (x, y) \in A \text{ для некоторого } y\}.$$

Если A есть некоторое соответствие, то **область значений** соответствия A, сокращенно обозначаемая как *rng* A, дается выражением

$$rng\, A = \{ y : (x, y) \in A \text{ для некоторого } x\}.$$

Некоторая **функция** f есть соответствие, обладающее свойством, что всякий раз, когда $(x, y) \in f$ и $(x, y^*) \in f$, то $y = y^*$. Некоторая **последовательность** есть функция, чья область определения есть множество целых чисел. Некоторая **конечная последовательность** есть последовательность, чья область определения есть конечное множество целых чисел, обычно $\{1, 2,\ldots, n\}$ для некоторого положительного целого n. Она также называется n-**кой**. Совокупность вещественнозначных последовательностей с областью определения $\{1,2,\ldots, n\}$ называется n-мерным вещественным пространством и обозначается как $\mathbb{R}^n$. Каждый элемент из $\mathbb{R}^n$ есть некоторая **вещественная** n-**ка**, или некоторая **вещественная последовательность** из n **элементов**, или, иначе, некоторый n-**мерный вектор**. Любой n-вектор x записывается как $x = (x_1, x_2,\ldots, x_n)$.

Определение 1.0.1 Точка *или, иначе,* **вектор** *в* $\mathbb{R}^n$ *есть упорядоченная совокупность n вещественных чисел:* $x = (x_1, x_2,\ldots, x_n)$.

При использовании векторно-матричных операций принято вектор отождествлять с матрицей-столбцом. Мы будем также этим пользоваться, хотя в обычном тексте удобнее векторные компоненты писать в строку для экономии места. Если же в записи вектора потребуется акцент на то, что он трактуется как матрица-столбец, то будем пользоваться символом транспонирования: $x = (x_1, x_2,\ldots, x_n)^{\mathrm{T}}$ или $x^{\mathrm{T}} = (x_1, x_2,\ldots, x_n)$.

Определение 1.0.2 Отрезок $\overleftarrow{PQ}$, *где P и Q суть две точки, представленные векторами p, q* $\in \mathbb{R}^n$, *есть множество точек, определяемых соотношением*

$$\theta p + (1 - \theta)q, \quad 0 \le \theta \le 1.$$

Эквивалентная запись $q + \theta(p - q)$, $0 \le \theta \le 1$, имеет наглядную геометрическую интерпретацию: к концу вектора q добавляется доля θ ($0 \le \theta \le 1$) разностного вектора ($p - q$), который направлен из точки Q в точку P. Поскольку всегда можно заменить θ на γ, $\gamma = 1 - \theta$, то на самом деле направленность отрезка, обозначенная здесь верхней стрелкой, не имеет значения, и можно этот отрезок обозначать чертой: $\overline{PQ}$.

Определение 1.0.3 *Точечное множество S называется* **выпуклым множеством**, *если* $\forall p, q \in S : \theta p + (1 - \theta)q \in$ S, $0 \le \theta \le 1$, *т.е. если любой отрезок, ограниченный точками из S, лежит в S.*

Определение 1.0.4 *Экстремальная, или крайняя точка (или* **вершина**, *или, иначе, угловая точка) выпуклого множества S есть любая его точка, не лежащая внутри отрезка, соединяющего произвольную пару точек множества.*

Другими словами: точка $b \in S$ есть вершина выпуклого множества S, если не существует две такие точки p, $q \in S$, что $b = \theta p + (1 - \theta)q$ при некотором $0 < \theta < 1$.

Определение 1.0.5 Выпуклая оболочка *точек* $P_1, P_2, \ldots, P_k$, *представленных соответствующими векторами* $p_1, p_2, \ldots, p_k$, *есть множество точек вида*

$$y = \theta_1 p_1 + \theta_2 p_2 + \ldots + \theta_k p_k, \quad \sum_{i=1}^{k} \theta i = 1, \quad 0 \le \theta_i, \quad i = \overline{1,k}.$$

Выпуклую оболочку можно конструктивно вывести (построить), последовательно применяя определение отрезка. Одновременно это будет объяснением условия на весовые коэффициенты θ_i. Дадим такой вывод.

Пусть P_1, P_2 – произвольные точки. Тогда

$$y_2 = k_2 p_2 + (1 - k_2)p_1, \quad 0 \le k_2 \le 1,$$

есть любая точка отрезка $\overline{P_1P_2}$. Соединим ее отрезком с произвольной точкой P_3. Тогда

$$y_3 = k_3 p_3 + (1 - k_3)y_2 = k_3 p_3 + (1 - k_3)k_2 p_2 + (1 - k_3)(1 - k_2)p_1, \quad 0 \le k_3 \le 1,$$

есть любая точка трехгранника $P_1P_2P_3$. Соединим ее отрезком с произвольной точкой P_4. Тогда

$$y_4 = k_4 p_4 + (1 - k_4)y_3 = k_4 p_4 + (1 - k_4)k_3 p_3 +$$
$$+ (1 - k_4)(1 - k_3)k_2 p_2 + (1 - k_4)(1 - k_3)(1 - k_2)p_1, \quad 0 \le k_4 \le 1,$$

есть любая точка четырехгранника $P_1P_2P_3P_4$.

Такой процесс можно продолжать и далее или прекратить после исчерпания всех имеющихся точек. Например, если последняя из имеющихся точек есть P_5, то, соединяя ее отрезком с точкой y_5, получим, что

$$y_5 = k_5p_5 + (1 - k_5)y_4 = k_5p_5 +$$
$$+ (1 - k_5)(k_4p_4 + (1 - k_4)(k_3p_3 + (1 - k_3)(k_2p_2 + (1 - k_2)p_1))), \quad 0 \le k_4 \le 1,$$

есть любая точка пятигранника $P_1P_2P_3P_4P_5$.

По построению, имеем

$$y_5 = \theta_5p_5 + \theta_4p_4 + \theta_3p_3 + \theta_2p_2 + \theta_1p_1,$$

где

$\theta_5 = k_5$,
$\theta_4 = (1 - k_5)k_4$,
$\theta_3 = (1 - k_5)(1 - k_4)k_3$,
$\theta_2 = (1 - k_5)(1 - k_4)(1 - k_3)k_2$,
$\theta_1 = (1 - k_5)(1 - k_4)(1 - k_3)(1 - k_2)$.

Пусть теперь наложено условие на весовые коэффициенты θ_i, а именно:

$$\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5 = 1, \quad \theta_i \ge 0, \quad i = \overline{1{,}5}\,.$$

Тогда, выбрав любые θ_i, $0 \le \theta_i \le 1_i$, удовлетворяющие этому условию, последовательно найдем

$$k_5 = \theta_5, \quad 0 \le k_5 \le 1;$$
$$k_4 = \frac{\theta_4}{(1-\theta_5)} = \frac{\theta_4}{(\theta_1+\theta_2+\theta_3+\theta_4)}, \ 0 \le k_4 \le 1;$$
$$k_3 = \frac{\theta_3}{(1-\theta_5-\theta_4)} = \frac{\theta_3}{(\theta_1+\theta_2+\theta_3)}, \ 0 \le k_3 \le 1;$$
$$k_2 = \frac{\theta_2}{(1-\theta_5-\theta_4-\theta_3)} = \frac{\theta_2}{(\theta_1+\theta_2)}, \ 0 \le k_2 \le 1.$$

Это те же самые k_2, k_3, k_4 и k_5, которые выше участвуют в построении любых точек: двухгранника (k_2), трехгранника (k_3), четырехгранника (k_4) и пятигранника (k_5). Тем самым доказано, что выпуклая оболочка точек $p_1, p_2, \dots, p_k$ определена полностью условием $\theta_1 + \theta_2 + \dots + \theta_k = 1$ на неотрицательные коэффициенты θ_i в линейной комбинации

$$y = \theta_1p_1 + \theta_2p_2 + \dots + \theta_kp_k\,.$$

Определение 1.0.6 *Если f есть некоторая функция и A есть некоторое подмножество области определения dmn f, тогда f_A,* **сужение** *f к A, есть функция, чья область определения есть A, задаваемая условием*

$$f_A(t) = f(t) \quad \text{для} \quad t \in A.$$

Определение 1.0.7 *Некоторая* **кривая уровня функции** *f, где f есть некоторая функция (от) n вещественных переменных, есть множество $C \subseteq$ dmn f, если сужение функции f к C есть константа (постоянная величина).*

Определение 1.0.8 *Если $f(x)$ есть некоторая вещественная функция (от) n вещественных переменных $x = (x_1, x_2, \ldots, x_n)$ и если частные производные $f_1(x), \ldots, f_n(x)$ существуют (где $f_i(x) = \partial f(x)/\partial x_i$), то* **градиент функции** *f по x определяется как n-мерный вектор*

$$grad\, f(x) = (f_1(x), \ldots, f_n(x)).$$

В общем случае, градиент вещественнозначной функции f (от) n вещественных переменных есть вектор, имеющий направление наибольшего возрастания функции f в каждой точке в *dmn f*. Соответственно, вектор антиградиента – *grad f(x)*, показывает направление скорейшего спуска вдоль поверхности f в области определения этой функции. Для всех вещественнозначных функций f (от) n вещественных переменных, обладающих непрерывным градиентом, любая кривая уровня функции f ортогональна градиенту в каждой точке в *dmn f*.

2
Стандартная задача линейного программирования

В этом разделе мы используем стандартную матричную форму записи задачи линейного программирования, к которой может быть легко приведена любая ЛП-задача. Мы описываем несколько фактов, объясняющих, как сузить поиск оптимума над допустимой областью, и подготавливающих лучшее понимание симплекс-метода, который представляет собой алгебраический метод решения стандартной ЛП-задачи, непосредственно готовый для программирования на компьютере.

2.1 Постановка задачи

Стандартная задача ЛП имеет:
- ограничения $Ax = b$, $A = A(m, n)$, $rank\, A = m$, $(m < n)$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$;
- условие неотрицательности координат вектора x, $x \geq 0$ (т.е. $\forall i : x_i \geq 0$),
и при этом требует $\min\limits_x c^T x$, где $c \in \mathbb{R}^n$.

Замечание 2.1.1 *Вектор коэффициентов* $c = (c_1, c_2, \ldots, c_n)$ *определяет* **целевую функцию** $z = c^T x$ *и является ее градиентом:* $grad\ z = c$. *Вектор градиента, указывающий направление скорейшего возрастания функции в пространстве векторов* $x \in \mathbb{R}^n$, *ортогонален линиям (фиксированного) уровня функции, – см. разд. 1.*

Определение 2.1.1 *Говорят, что стандартная задача линейного программирования* **невырождена**, *если каждая* $m \times m$ *подматрица, выбранная из* $m \times (n + 1)$ *расширенной матрицы* $A_a = [A \mid b]$, *невырождена. В противном случае говорят, что это –* **вырожденная задача ЛП**.

Замечание 2.1.2 *Предложение о невырожденности справедливо для большинства задач, встречаемых на практике, и оно удобно для теории. Как справиться с решением вырожденных задач, будет рассмотрено в разд. 6.*

Данная задача обладает следующими основными свойствами: (1) выпуклость множества допустимых решений; (2) существование базисных допустимых решений; (3) тождественность базисных допустимых решений и вершин множества допустимых решений и (4) совпадение хотя бы одного оптимального решения задачи (если оно существует) с какой-либо вершиной допустимого множества решений. Докажем эти свойства последовательно.

2.2 Выпуклость множества допустимых решений

Определение 2.2.1 *Множество X точек в* $\mathbb{R}^n$, *удовлетворяющих ограничениям и условию задачи ЛП,* $Ax = b$, $x \geq 0$, *есть* **множество допустимых решений** *или, иначе, допустимое множество*.

Теорема 2.2.1 *Множество X допустимых решений*

$$X = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\} \tag{2.1}$$

задачи ЛП выпукло.

Доказательство. Пусть x, y – произвольная пара точек из X. Тогда

$$Ax = b, \quad x \geq 0; \quad Ay = b, \quad y \geq 0.$$

Отрезок, соединяющий эти точки, обозначим как множество точек w,

$$w = \alpha x + \beta y, \;\; \alpha + \beta = 1, \;\; \alpha \geq 0, \;\; \beta \geq 0, \;\; w \geq 0.$$

Имеем

$$Aw = A(\alpha x + \beta y) = \alpha Ax + \beta Ay = \alpha b + \beta b = (\alpha + \beta)b = b,$$

т.е. $Aw = b$, $w \geq 0$.

2.3 Существование базисных допустимых решений (БДР)

Определение 2.3.1 Базисным решением *задачи ЛП (в стандартной форме) называется вектор*

$$x = P\begin{pmatrix} B^{-1}b \\ O \end{pmatrix},$$

где P есть матрица перестановок столбцов матрицы A, такая что $AP = [B|R]$, *B – матрица полного ранга размера* $m\times m$; *O – (n – m) нулевых элементов вектора; R – (n – m) столбцов, линейно выражающиеся через столбцы матрицы B.*

Все базисные решения в количестве $\begin{pmatrix} n \\ m \end{pmatrix} = \dfrac{n!}{m!(n-m)!}$ определяются различными сочетаниями m линейно независимых столбцов матрицы A из n ее столбцов. Матрица перестановок P здесь введена, чтобы в результате перестановок все m линейно независимых столбцов матрицы A оказались сгруппированы влево в виде подматрицы B. Тогда ограничение $Ax = b$ равносильно уравнению $APy = b$, где $y = P^{-1}x$.

Разбивая вектор y на две части, $y = (y^B, y^F)$, y^B – первые m компонент, y^F – остальные $(n – m)$ компонент, имеем

$$y^B = B^{-1}b - B^{-1}Ry^F, \quad y = \begin{pmatrix} B^{-1}b - B^{-1}Ry^F \\ y^F \end{pmatrix}.$$

Таким образом, решение y уравнения $APy = b$ выражено через свободные переменные y^F. Обращая их в нуль, получаем вектор

$$y = \begin{pmatrix} B^{-1}b \\ O \end{pmatrix}.$$

От него вектор x отличается лишь перестановкой элементов: $x = Py$. Этот x и называется базисным решением.

Определение 2.3.2 *Нулевые элементы вектора $x = Py$, получившиеся равными нулю вследствие обращения в нуль свободных (последних n-m) переменных y^F в составе вектора y, удовлетворяющего ограничению $APy = b$, называются* **небазисными переменными**. *Остальные m элементов вектора $x = Py$ называются* **базисными переменными** *и образуют* **базис**. *(Относительно матрицы P см. определение 2.3.1).*

Определение 2.3.3 *Если базисное решение обладает свойством неотрицательности своих элементов, то оно называется* **базисным допустимым решением**.

Чтобы избежать формального понятия матрицы перестановок P в данных выше определениях, перефразируем их, использую следующие формулировки.

Определение 2.3.4 Векторами ограничений $a_i \in \mathbb{R}^m$, $i = \overline{1,n}$, *называются столбцы матрицы A:*

$$A = [a_1, a_2, \ldots, a_n].$$

Сами ограничения тогда имеют вид

$$x_1a_1 + x_2a_2 + \ldots + x_na_n = b. \tag{2.2}$$

Свойство матрицы ограничений A: так как $rank\ A = m$, то среди ее n столбцов существуют наборы по m линейно независимых столбцов-векторов ограничений. Число таких различных линейно независимых наборов достигает $C_n^m = \binom{n}{m} = \frac{n!}{m!(n-m)!}$ (сочетания из n по m).

Определение 2.3.5 (Определение 2.3.2 в иной формулировке.) *Переменные x_i, стоящие коэффициентами при векторах ограничений a_i в каждом линейно независимом наборе из m векторов, называются* **базисными переменными**, *а остальные –* **небазисными переменными** *(небазисные переменные иногда называют свободными переменными).*

Определение 2.3.6 *Любой вектор $x = (x_1, x_2, \ldots, x_n)$, удовлетворяющий ограничению (2.2) и условию неотрицательности всех своих координат ($x_i \geq 0$, $i = \overline{1,n}$) называется* **допустимым решением задачи ЛП**.

Определение 2.3.7 (Определение 2.3.1 в иной формулировке.) *Точка $x = (x_1, x_2,\ldots, x_n)$, которая удовлетворяет ограничениям (2.2) называется* **базисным решением***, если векторы a_i, не входящие в линейно независимый набор, имеют в ограничениях в качестве коэффициентов x_i нулевые значения.*

Базисное решение оказывается не всегда допустимым. Это случается тогда, когда в нем имеются отрицательные элементы.

Определение 2.3.8 (Определение 2.3.3 в иной формулировке.) *Точка $x = (x_1, x_2,\ldots, x_n)$ называется* **базисным допустимым решением** *(или опорным планом) задачи ЛП, если система векторов a_i, входящих в ограничение (2.2) с положительными коэффициентами x_i, $x_i > 0$, линейно независима.*

Из этого определения следует, что число r положительных компонент БДР не больше, чем m, $r \le m$, так как некоторые коэффициенты при векторах ограничений a_i в линейно независимом наборе из m векторов ограничений могут оказаться нулевыми.

Определение 2.3.9 *Базисное допустимое решение называется* **невырожденным***, если оно не содержит нулевых компонент. В противном случае оно называется* **вырожденным базисным допустимым решением**.

Теорема 2.3.1 *Если задача линейного программирования невырожденная, тогда переменные в любом базисном решении есть строго ненулевые переменные.*

Доказательство. Предположим, что существуют q базисных переменные в некотором базисном решении, $q = m - r$, $0 < q < m$, которые равны нулю. Тогда b должно бы быть некоторой линейной комбинацией из r столбцов матрицы B (и, таким образом, матрицы A), причем B составлена из тех столбцов a_i, которые ассоциированы с базисными переменными x_i в (2.2). Из этого вытекало бы, что задача вырождена. Это противоречит гипотезе данной теоремы, что доказывает желаемый результат.

Теорема 2.3.2 *Если ограничения имеют допустимое (неотрицательное) решение, то они имеют и базисное (допустимое) решение.*

Доказательство. (Непосредственным построением БДР). Пусть в допустимом решении $(n - r)$ переменных равны нулю, а остальные r положительны. Без потери общности (с учетом возможной перенумерации переменных, см. матрицу P в определении 2.3.1), это запишем так:

$$x_j = 0, \quad j = r+1, r+2,\ldots, n;$$
$$\sum_{j=1}^{r} x_j a_j = b, \quad x_j > 1, \quad j = 1, 2,\ldots, r.$$

Случай (1): $\{a_j, j = 1, 2,\ldots, r\}$ – линейно независимы. Тогда $r \le m$, где $m = rank\ A$, и данное допустимое решение является БДР, в котором $(m - r)$ переменных оказались равны нулю. Для этого случая теорема доказана.

Случай (2): $\{a_j, j = 1, 2,\ldots, r\}$ – линейно зависимы. Это означает, что равенство

$$\sum_{j=1}^{r} \alpha_j a_j = 0$$

выполняется не при всех $\alpha_i = 0$. Пусть $\alpha_k > 0$ для некоторого $k = \overline{1,r}$. (При необходимости последнее равенство может быть умножено на –1). Тогда

$$a_k = -\sum_{j=1, j\ne k}^{r} \left(\frac{\alpha_j}{\alpha_k}\right) a_j .$$

Имеем ограничение

$$x_1a_1 +\ldots+ x_ka_k +\ldots+ x_ra_r = b. \qquad (2.3)$$

Подставим сюда последнее выражение для a_k:

$$x_1a_1 +\ldots+ x_k(\ldots- \frac{\alpha_{k-1}}{\alpha_k}a_{k-1} - \frac{\alpha_{k+1}}{\alpha_k}a_{k+1} -\ldots) +\ldots+ x_ra_r = b.$$

Приведем подобные и получим

$$\sum_{j\ne k, j=1}^{r} (x_j - \frac{\alpha_j x_k}{\alpha_k}) a_j = b .$$

Всегда можно выбрать k так, что

$$\frac{x_k}{\alpha_k} = \min_j (\frac{x_j}{\alpha_j}; \alpha_j > 0) .$$

В силу этого значения

$$X_j = x_j - \alpha_j(\frac{x_k}{\alpha_k}), \quad j \ne k,\ j = \overline{1,r} \qquad (2.4)$$

оказываются положительными и

$$X_j = 0, \quad j = k, r+1, r+2,\ldots, n. \qquad (2.5)$$

В итоге решение $X = (X_1, X_2,\ldots, X_n)$ допустимо, так как ограничение

$$\sum_{i=1}^{n} X_i a_i = b \qquad (2.6)$$

Выполнено и $X_i \ge 0$, $i = \overline{1,n}$, причем в этом решении $(r - 1)$ переменных строго положительны и $(n - r + 1)$ переменных равны нулю.

Таким образом, количество строго положительных переменных в допустимом решении оказалось уменьшено на единицу. Продолжая, подобно этому, процесс исключения линейно зависимого вектора ограничений a_k из числа тех векторов a_j, $j = \overline{1,r}$, которые входят в ограничение $Ax = b$ с положительными коэффициентами, придем к случаю 1, в котором $r \le m$, то есть построим БДР.

2.4 Тождественность БДР и вершин множества допустимых решений

Теорема 2.4.1 *Базисные допустимые решения суть вершины (выпуклого) множества X допустимых решений x, $Ax = b$, $x \ge 0$.*

Замечание 2.4.1 *Иная формулировка этой теоремы: Для того, чтобы допустимое решение x^0 задачи ЛП было базисным допустимым решением, необходимо и достаточно, чтобы x^0 было вершиной выпуклого множества X допустимых решений x, $Ax = b$, $x \ge 0$.*

Доказательство. Обозначим по определению

$$x^0 = (x_1,\dots, x_m, 0,\dots, 0)^{\mathrm{T}}$$

какое-нибудь базисное допустимое решение. Это означает, что x^0 – единственное решение уравнения $Ax = b$ при условии $x \ge 0$, имеющее нули в последних $(n – m)$ координатах.

(1) **Условие необходимо.** (Дано x^0 – базисное допустимое решение. Доказать x^0 – вершина). Предположим противное, а именно, что x^0 – не вершина. Тогда в X найдутся две другие различные точки u и v такие, что $x^0 = \theta u + (1 - \theta)v$ для некоторого θ, $0 < \theta < 1$. Имеем: $Au = b$, $Av = b$, u, $v \ge 0$, что означает $u, v \in X$. Следовательно, последние $(n - m)$ координат векторов u и v удовлетворяют условиям равенства нулю:

$$0 = \theta u_{m+1} + (1 - \theta)v_{m+1};$$
$$0 = \theta u_{m+2} + (1 - \theta)v_{m+2};$$
$$\dots\dots\dots\dots\dots\dots$$
$$0 = \theta u_n + (1 - \theta)v_n.$$

В этих равенствах оба коэффициента θ и $(1 - \theta)$ положительны, так как по предположению x^0 – внутренняя точка отрезка между точками u и v. Кроме того, $u_{m+1},\dots, u_n \ge 0$ и $v_{m+1},\dots, v_n \ge 0$.

С учетом этого, указанные равенства нулю возможны только при $u_j = 0$ и $v_j = 0$, $j = m+1,\dots, n$. Но тогда u и v подпадают под определение БДР, а оно, по определению, есть единственное решение уравнения $Ax = b$, $x \ge 0$, с нулевыми последними $(n – m)$ координатами. Очевидное противоречие доказывает, что $x^0 = u = v$ есть вершина.

(2) **Условие достаточно.** (Дано x^0 – какая-нибудь вершина допустимой области X. Доказать: x^0 – есть БДР). Обозначим через r число координат вектора x^0, которые строго положительны. Докажем, что $r \le m$ и что соответствующие этим r координатам векторы ограничений a_j линейно

независимы, то есть, что x^0 есть БДР. Для этого будем действовать так же, как в доказательстве Теоремы 2.3.2. Как и там, имеем здесь **случай (1)** и **случай (2)**. В случае (1) доказательство готово. Случай же (2) означает, что не при всех $\alpha_j = 0, j = \overline{1,r}$ выполняется равенство

$$\sum_{j=1}^{r} \alpha_j a_j = 0\,. \tag{2.7}$$

Введем число ρ такое, что для этих $\alpha_j \neq 0$

$$0 < \rho < \min_j \frac{x_j^0}{|\alpha_j|}.$$

Тогда векторы

$$x^1 = x^0 + \rho\alpha, \quad x^2 = x^0 - \rho\alpha$$

окажутся неотрицательны: $x^1 \geq 0$ и $x^2 \geq 0$, где введено обозначение для вектора α, определяемого равенством $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_r, 0, \ldots, 0)$. Это дает возможность переписать исходное равенство в виде $A\alpha = 0$. На этом основании имеем:

$$Ax^1 = A(x^0 + \rho\alpha) = Ax^0 + \rho A\alpha = Ax^0 = b;$$
$$Ax^2 = A(x^0 - \rho\alpha) = Ax^0 - \rho A\alpha = Ax^0 = b.$$

Беря полусумму последних двух равенств и учитывая, что

$$x^0 = \frac{1}{2}(x^1 + x^2) = \theta x^1 + (1 - \theta)x^2, \quad \theta = \frac{1}{2},$$

убеждаемся, что построены две различные точки x^1, $x^2 \in X$, отличные от x^0 и такие, что x^0 есть внутренняя точка (середина) отрезка, их соединяющего, то есть x^0 – не вершина. Это противоречит данному условию, поэтому все α_j в исходном равенстве равны нулю, то есть векторы $a_j = 0$, $j = \overline{1,r}$ линейно независимы. Но, как известно, если $a_j \in \mathbb{R}^m$, то число r таких векторов не может превышать m, $r \leq m$. Мы возвратились к случаю (1), для которого достаточность условия теоремы доказана.

Замечание 2.4.2 Теорему 2.4.1 удобно сформулировать и наоборот: *Для того чтобы точка x^0 была вершиной выпуклого множества X допустимых решений задачи ЛП, необходимо и достаточно, чтобы векторы ограничений a_j, имеющие в качестве коэффициентов положительные координаты этой точки в ограничениях $Ax = b$, $x \geq 0$, были линейно независимы.*

В этой (эквивалентной) формулировке, очевидно, необходимость и достаточность просто меняются местами.

2.5 Совпадение решения задачи ЛП с вершиной допустимого множества

Следующее утверждение является главным для дальнейшего перехода к практическому вычислительному методу (симплекс-методу), позволяющему сравнительно быстро отыскивать решение задачи ЛП.

Теорема 2.5.1 *Если целевая функция $z = c^T x$ имеет конечный минимум, то по крайней мере одно оптимальное решение является базисным допустимым решением (то есть вершиной допустимого многогранника, обозначенного X).*

Доказательство. Пусть все БДР определяются векторами $p_1, p_2, \ldots, p_k$ и пусть целевая функция принимает в этих точках значения, соответственно, $z_1, z_2, \ldots, z_k$. Так как $z = c^T p_i$, то

$$z_i = c^T p_i, \quad i = 1, 2, \ldots, k.$$

Уже известно (из теоремы 2.4.1), что $p_1, p_2, \ldots, p_k$ – это все вершины множества X допустимых решений, и что это множество выпукло (из теоремы 2.2.1), и что любая выпуклая оболочка данных точек определяется как множество точек x вида

$$x = \theta_1 p_1 + \theta_2 p_2 + \ldots + \theta_k p_k, \quad \theta_i \geq 0, \quad \sum_{i=1}^{k} \theta_i = 1.$$

В любой такой точке x, $x \in X$, имеем

$$z = c^T x = \theta_1 c^T p_1 + \theta_2 c^T p_2 + \ldots + \theta_k c^T p_k = \theta_1 z_1 + \theta_2 z_2 + \ldots + \theta_k z_k.$$

Следовательно, отыскание решения $x \in X$, оптимального в смысле минимума целевой функции ($\min z = \min c^T x$) сводится к задаче нахождения таких чисел $\theta_i \geq 0$, которые в сумме дают 1 и при этом доставляют минимум величине

$$z = \theta_1 z_1 + \theta_2 z_2 + \ldots + \theta_k z_k,$$

где $z_i = c^T p_i$ – вещественные числа. Среди конечного набора чисел z_i всегда найдется одно или несколько, имеющих наименьшее значение z_{min}. Без ограничения общности можно считать, что эти первые l чисел, т.е. $z_j = z_{min}$, $j = 1, 2, \ldots, l$. Тогда величина $z = \theta_1 z_1 + \theta_2 z_2 + \ldots + \theta_k z_k$ будет, очевидно, наименьшей, равной z_{min}, если неотрицательные весовые коэффициенты выбрать так: $\theta_1 + \ldots + \theta_l = 1$ и $\theta_i = 0$ при $i = l + 1, \ldots, k$. Таким образом, доказано, что минимум функции $z = c^T x$ достигается в некоторой вершине p_j или же в выпуклой оболочке конечного числа вершин $p_1, p_2, \ldots, p_l$, то есть в точке

$$x = \sum_{i=1}^{l} \theta_i p_i, \quad \sum_{i=1}^{l} \theta_i = 1, \quad 0 \leq \theta_i$$

допустимого многогранника X.

Вывод к разделу 1: При поиске решения задачи ЛП(в стандартной форме) достаточно ограничиться нахождением базисных допустимых решений и только их (то есть вершин допустимого многогранника решений). Симплекс-метод, излагаемый ниже, представляет собой конечную целенаправленную процедуру поиска такого решения.

3
Симплекс-метод

Слово **simplex** в обычном смысле означает простой, несоставной, в противоположность **complex**. Как математическое понятие, симплекс – есть выпуклая оболочка m точек n-мерного метрического пространства; 0-мерный симплекс есть точка, 1-мерный симплекс – отрезок, 2-мерный – треугольник, 3-мерный – тетраэдр, и т.д. Поскольку уже установлено (см. главу 2), что в задаче ЛП решение ищется в вершинах множества допустимых решений X, являющегося, как видно, симплексом, сама процедура поиска получила название «симплекс метод». Он разработан американским математиком Г. Данцигом в 1947 году. Это остроумное изобретение позволяет находить оптимальное решение посредством количественной оценки только незначительной доли всех вершин при перемещении вдоль ребер на границе гипермногогранника (допустимого множества), отыскивая таким образом одну целевую вершину. Следовательно, симплекс-метод избегает посещения каждой вершины и продвигается от одной (стартовой) вершины к оптимальной вершине через множество промежуточных вершин. В каждой вершине все соседние (смежные) вершины оцениваются так, чтобы определить, какие из них способны уменьшить целевую функцию, а какие нет. Некоторая новая вершина выбирается среди способных вершин так, чтобы уменьшение целевой функции было наибольшим из возможных, и из этой новой вершины оценка соседних вершин повторяется, пока мы не достигнем целевой вершины. Данный метод получил несколько различных форм (алгоритмов). Для удобства их изложения рассмотрим предварительно представление стандартной задачи ЛП в иной, эквивалентной форме.

3.1 Приведение задачи ЛП к канонической форме для базиса

По определению БДР, для его нахождения нужно сначала найти m линейно-независимых столбцов a_j матрицы ограничений A. Пусть они уже найдены и (с учетом возможной перенумерации переменных) являются сомножителями при первых m переменных, $x_1, x_2,\dots, x_m$. Эти столбцы образуют невырожденную матрицу

$$B = [a_1, a_2,\dots, a_m].$$

Следовательно, $A = [B \mid R]$, где $R = [a_{m+1},\dots, a_n]$ – матрица из столбцов, являющихся линейными комбинациями столбцов матрицы B. Введем еще соответствующие обозначения: вектор $x^B = (x_1, x_2,\dots, x_m)^T$, составленный из базисных переменных, и вектор $x^F = (x_{m+1}, x_{m+2},\dots, x_n)^T$, составленный из небазисных (свободных) переменных. Тогда $x = (x^B, x^F)$. Аналогично $c^B = (c_1,\dots, c_m)^T$, $c^F = (c_{m+1},\dots, c_n)^T$ и $c = (c^B, c^F)$.

Имеем **стандартную задачу ЛП**: Минимизировать целевую функцию

$$z = c^Tx$$

при ограничениях $Ax = b$, $x \geq 0$. Теперь с помощью введенных обозначений эту задачу перепишем в другой форме, учитывая, что B^{-1} существует.

Сначала перепишем ограничения $Ax = b$:

$$Bx^B + Rx^F = b, \quad x^B + B^{-1}Rx^F = b', \quad b' = B^{-1}b.$$

Матрицу $B^{-1}R$, имеющую размеры $m \times (n - m)$, обозначим A' и запишем как совокупность столбцов:

$$A' = B^{-1}R = [a'_{m+1} : a'_{m+2} : \ldots : a'_n];$$
$$a'_j = B^{-1}a_j, \quad j = m+1,\ldots, n.$$

Далее, перепишем целевую функцию $z = c^T x$:

$$z = c_B^T x^B + c_F^T x^F = c_B^T (b' - A'x^F) + c_F^T x^F =$$
$$= c_B^T b' + (c_F^T - c_B^T A')x^F = z_0 + c'^T x^F,$$

где новые коэффициенты обозначены c', $c'^T = c_F^T - c_B^T A'$, то есть равны

$$c'_j = c_j - c_B^T a'_j, \quad j = m+1,\ldots, n,$$

и значение целевой функции z при $x^F = 0$ обозначено z_0, т.е. $z_0 = c_B^T b'$.

Следовательно, стандартная задача ЛП:

$$\min_x (z = c^T x), \quad Ax = b, \quad x \geq 0$$

имеет следующую эквивалентную формулировку, так называемую **каноническую форму задачи для базиса** $x^B = (x_1, x_2,\ldots, x_m)$:

минимизировать целевую функцию

$$z - z_0 = c'^T x^F$$

при ограничениях

$$x^B + A'x^F = b', \quad x^B \geq 0, \quad x^F \geq 0,$$

где

$$z_0 = c_B^T b'; \quad b' = B^{-1}b; \quad A' = \{a'_j\}, j = m + 1,\ldots, n,$$
$$c'_j = c_j - c_B^T a'_j, \quad a'_j = B^{-1}a_j, \quad j = m + 1,\ldots, n.$$

При этом в обозначениях учтено, что

$$B = \{a_j\}, \quad j = \overline{1,m}, \quad rank\, B = m; \quad A = \{a_j\}, \quad j = \overline{1,n}.$$

Особенности данной формы заключаются в следующем: (1) базисные переменные x^B входят в ограничения с единичной матрицей, но при этом в целевую функцию не входят; (2) целевая функция $(z - z_0)$ выражена только через свободные, небазисные переменные x^F. Именно такое представление задачи ЛП удобно для построения той вычислительной процедуры, которая успешно решает задачу и известна как симплекс-метод.

3.2 Симплекс-метод при известном базисном допустимом решении

Приведение задачи ЛП к канонической форме для базиса означает, что найдено какое-нибудь базисное решение и оно имеет вид:

небазисные переменные $x^F = 0$;

базисные переменные $x^B = b'$;
целевая функция $z = z_0 = c_B^T b'$.

Если при этом окажется, что $b' \geq 0$, то это базисное решение есть базисное допустимое решение (одна из вершин допустимого многогранника), в противном случае оно не является допустимым решением.

В данном пункте предполагаем, что в начале или на очередном шаге решения оказалось $b' \geq 0$. Спрашивается, нельзя ли сделать еще шаг и еще уменьшить значение целевой функции z, переходя к какой-либо другой вершине допустимого многогранника? Ответ на этот вопрос получаем, обратившись к виду z:

$$z - z_0 = c'^T x^F.$$

Теорема 3.2.1 (i) *Если все элементы вектора c' строго положительны, $c' > 0$ тогда невозможно уменьшить z, перемещаясь от $x^F = 0$ и оставаясь в допустимой области, $x^F \geq 0$. Минимум функции $z = z_0$ найден, и соответствующее базисное допустимое решение является единственным решением данной ЛП-задачи, т.е. это и есть оптимальное базисное допустимое решение (ОБДР).*

(ii) *Если среди элементов вектора c' существует, по крайней мере, один элемент, скажем c'_s, который меньше нуля, тогда есть только две возможности:*

(a) *Если $a'_{is} \leq 0$ для всех $i = 1,\ldots, m$, тогда существует допустимые решения такие, что функция z может быть сделана сколько угодно малой, и данная ЛП-задача не имеет конечного решения.*

(b) *Если $a'_{is} > 0$ хотя бы для одного i, тогда может быть найдено некоторое новое базисное решение такое, при котором значение функции z строго уменьшается.*

(iii) *Если все элементы вектора c' неотрицательны, $c' \geq 0$, в то время как по крайней мере один из них, скажем c'_s, равен нулю, тогда z достигает своего минимума по меньшей мере в двух различных точках x. Если два таких оптимальных решения обозначены как $x^{(1)}$ и $x^{(2)}$, то*

$$x = \theta x^{(1)} + (1-\theta) x^{(2)}, \quad 0 \leq \theta \leq 1$$

есть тоже оптимальное решение.

(iv) *Если процедура* (ii) (b) *повторяется, тогда после конечного числа шагов либо достигается случай* (ii) (a), *в котором функция z может быть сделана сколь угодно малой, либо достигается один из других случаев,* (i) *или* (iii), *в котором оказывается найденным некоторое решение данной ЛП-задачи.*

Доказательство этой теоремы становится очевидным после следующего анализа.

Если все коэффициенты c' неотрицательны, $c' \geq 0$, то в допустимой области при $x^F \geq 0$ нельзя уменьшить z, двигаясь от $x^F = 0$. Минимум $z = z_0$ найден.

Если же среди коэффициентов c' есть хотя бы один строго отрицательный, то z можно уменьшать и далее, если увеличивать от 0 те свободные переменные в составе x^F, которые входят в целевую функцию $(z - z_0) = c'^T x^F$ с отрицательными коэффициентами c', но следя за тем, чтобы при этом остаться в допустимой области

$$x^B + A'x^F = b', \quad x^B \geq 0, \quad x^F \geq 0. \tag{3.1}$$

Безразлично, сколько таких переменных и какие из них выбрать одновременно для увеличения, но симплекс-метод предлагает увеличивать только одну. Целесообразно выбрать ту переменную из x^F, которая входит в $(z - z_0) = c'^T x^F$ с наибольшим по модулю отрицательным коэффициентом. Это отвечает скорейшему спуску по z среди возможных вариантов покоординатного спуска.

Допустим, выбор сделан и x_s есть именно та свободная переменная, которую будем увеличивать от нуля, оставляя все другие небазисные переменные в x^F неизменными и равными нулю. Тогда ограничения, за которыми надо следить, приобретают простой вид

$$x_i^B + a'_{is} x_s = b'_i, \quad x_i^B \geq 0, \quad x_s > 0, \quad i = \overline{1,m}. \tag{3.2}$$

Из этого мы видим, что то изменение в величине x_s, которое мы вносим для уменьшения z, вынуждает нас уменьшить величину x_i^B, для того, чтобы сохранить в силе равенства (3.2). Естественно, x_i^B уменьшается, если только $a'_{is} > 0$. Если $a'_{is} < 0$, это изменение будет увеличивать x_i^B, и x_i^B останется неизменным, если $a'_{is} = 0$. В последних двух случаях ограничение $x_i^B \geq 0$ не может быть нарушено. Следовательно, единственный случай, который может вызвать беспокойство, есть $a'_{is} > 0$, так как в (3.2) мы обязаны сохранить неравенство $x_i^B \geq 0$.

Теперь нужно ответить на вопрос: до какого предела можно увеличивать x_s? С точки зрения отдельного взятого i-го ограничения: увеличение x_s ведет уменьшение x_i^B, и поскольку уменьшать x_i^B можно только до нулевого значения, $x_i^B = 0$, увеличивать x_s можно только до значения $x_s = b'_i/a'_{is}$. Однако, с точки зрения всех m ограничений, взятых одновременно, увеличение x_s допустимо лишь до наименьшего значения

$$x_s = \min_{\underline{i=1,m}} \left(\frac{b'_i}{a'_{is}} \right).$$

Допустим, что это наименьшее значение обнаружено в k-й строке ограничений (3.2), то есть

$$x_s = \min_{\underline{i=1,m}} \left(\frac{b'_i}{a'_{is}} \right) = \frac{b'_k}{a'_{ks}}.$$

Тогда имеем следующий результат: выбранная из числа небазисных переменная x_s увеличина с нулевого значения до значения b'_k/a'_{ks} и одновременно k-я переменная x_k из числа базисных уменьшена до нулевого значения. То есть x_s ведена в базис, а x_k выведена из базиса, – совершен переход из одного БДР к другому БДР, при котором уменьшено значение целевой функции.

Это и есть очередной шаг симплекс-метода. Поскольку число БДР (вершин допустимого многогранника) конечно (не превышает C_n^m), минимум целевой функции будет найден за конечное число шагов.

3.3 Алгоритм симплекс-метода при известном БДР

Из п. 3.1 видно, что стандартная задача ЛП имеет множество представлений, называемых канонической формой для базиса, – для каждого базиса своя каноническая форма. С другой стороны, в п. 3.2 показано, что очередной шаг симплекс-метода, на котором происходит переход к меньшему значению целевой функции, означает введение в базис некоторой переменной x_s взамен другой переменной x_k, которая из базиса выводится. Таким образом, чтобы выполнить шаг симплекс-метода, формально нужно переписать задачу ЛП в канонической форме для следующего, очередного базиса. Опишем эти действия, образующие **стандартный симплекс-алгоритм**.

Дана стартовая или текущая каноническая форма задачи ЛП для базиса (см. п. 3.1) в виде системы при $b' \geq 0$:

$$\begin{bmatrix} I & A' \\ O & c'^T \end{bmatrix} \begin{bmatrix} x^B \\ x^F \end{bmatrix} = \begin{bmatrix} b' \\ z - z_0 \end{bmatrix}, \quad x^B \geq 0, \quad x^F \geq 0,$$

где последнее уравнение дает выражение для целевой функции через небазисные (свободные) переменные.

Здесь I – единичная матрица размера ($m \times m$); $A' = \{a'_j\}$ при $j = m+1,\ldots, n$, – матрица из векторов-столбцов a'_j; c' – вектор (столбец) коэффициентов при небазисных переменных x^F; $z_0 = c_B^T b'$ – исходное значение целевой функции для (старого или текущего) БДР: $x = (x^B, x^F)$, $x^B = b'$, $x^F = 0$.

Действие 1°. Выясняют, какую из переменных x^F, скажем x_s, можно увеличить от нуля, чтобы уменьшить z от значения z_0. Свидетельством для любой такой переменной служит соответствующий отрицательный коэффициент в последней строке блочной матрицы указанной выше системы, то есть в строке c'^T. Если два (или более) коэффициента обладают этим свойством, любой из них может быть выбран. Обычно среди таких коэффициентов выбирают отрицательный элемент с наибольшим абсолютным значением и тем самым определяют искомую переменную x_s. Если все элементы в строке c'^T неотрицательны, система находится в финальной форме и решение получено, так как искомый x_s более не существует. Выбранную переменную x_s называют **ведущей переменной**, или **входящей переменной**, поскольку именно она подготовлена для введения в базис, и соответствующий (s-й) столбец называют **ведущим столбцом**.

Действие 2°. Выясняют, которая из m строк ограничений (первые m строк записанной выше системы) обеспечивает

$$\frac{b'_k}{a'_{ks}} = \min_{i:a'_{is}>0}\left(\frac{b'_i}{a'_{is}}\right).$$

Номер этой строки обозначают k и эту строку называют **ведущей** (ключевой) **строкой**. Если две или более строки обладают этим свойством, любая из них может быть выбрана как ведущая. Коэффициент a'_{ks} для x_s в ведущей строке называют **ведущим элементом**. Переменную, являющуюся базисной в ведущей строке, называют **выходящей переменной**, так как именно она подготовлена для исключения из базиса. Введение в базис входящей переменной и выведение из базиса выходящей переменной выполняют благодаря следующим действиям.

Действие 3°. Нормируют k-ю строку, деля ее на ведущий элемент a'_{ks}, в результате в этой строке ведущая переменная x_s получает коэффициент1.

Действие 4°. Проводят серию вычитаний пронормированной k-ой строки, беря ее с подходящими коэффициентами, из всех других строк системы, с тем чтобы исключить ведущую переменную x_s из всех уравнений системы, кроме k-го. (Подходящие коэффициенты к этому моменту уже находятся в матрице системы на пересечении всех строк, кроме k-ой, с **ведущим столбцом**, имеющим номер s).

В результате этих четырех действий s-й столбец станет столбцом единичной матрицы, а k-й столбец системной матрицы перестанет быть таковым. Это и означает, что переменная x_s оказалась введена в базис, а переменная x_k оказалась выведена из базиса. Совершен переход к канонической форме задачи ЛП для другого базиса. На следующем шаге алгоритма эти действия повторяют, пока действие 1° не даст останова из-за отсутствия ведущей переменной.

Очевидно, что действия 3° и 4° совпадают с процедурой исключения переменной x_s по методу Гаусса-Жордана из данной системы уравнений. Действия же 1° и 2° предваряют эту процедуру лишь специфическим выбором: ведущей переменной (действие 1°) и ведущей строки (действие 2°).

Пример 3.3.1 *Минимизировать*

$$-2x_1 - 4x_2 = z$$

при ограничениях $x_1, x_2 \geq 0$ *и*

$$\left.\begin{aligned} 2x_1 + 3x_2 \le 120 \\ 3x_1 + 9x_2 \le 270 \end{aligned}\right\}.$$

Сначала запишем эту задачу ЛП в стандартной форме, введя две дополнительные переменные x_3, $x_4 \ge 0$:

$$\begin{aligned} 2x_1 + 3x_2 + x_3 \quad &= 120; \\ 3x_1 + 9x_2 \quad + x_4 &= 270; \\ -2x_1 - 4x_2 \quad &= z. \end{aligned}$$

В матричном виде:

$$\begin{bmatrix} 2 & 3 & 1 & 0 \\ 3 & 9 & 0 & 1 \\ -2 & -4 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 120 \\ 270 \\ z \end{bmatrix}.$$

Одновременно это служит стартовой канонической формой задачи для базиса $x^B = (x_3, x_4)$. Свободные, небазисные переменные: $x^F = (x_1, x_2)$. Стартовое базисное допустимое решение: $x = (x_1, x_2, x_3, x_4)$= (0, 0, 0, 0). Значение целевой функции для него: $z = z_0 = 0$.

Для функционирования симплекс-метода, то есть для выполнения составляющих его действий 1° – 4°, достаточно оперировать только с расширенной матрицей. В этом примере она имеет следующий стартовый вид:

$$\left[\begin{array}{cccc|c} 2 & 3 & 1 & 0 & 120 \\ 3 & 9 & 0 & 1 & 270 \\ -2 & -4 & 0 & 0 & z \end{array}\right].$$

Такая матрица называется **симплекс-таблицей**.

Выполним в ней необходимые действия 1° – 4°.

Шаг 1.

1°. $x_s = x_2$ (ведущий столбец – второй).

2°. $min\left(\frac{120}{3}, \frac{270}{9}\right) = min(40, 30) = 30$ (ведущая строка – вторая),

$k = 2$.

3°. После нормировки:

$$\begin{array}{c} \\ \longrightarrow \\ \\ \\ \end{array} \left[\begin{array}{cccc|c} 2 & 3 & 1 & 0 & 120 \\ 1/3 & 1 & 0 & 1/9 & 30 \\ -2 & -4 & 0 & 0 & z \end{array}\right].$$
$$\qquad\qquad\uparrow$$

(ведущий столбец и ведущая строка помечены для наглядности стрелками).

4°. После вычитаний:

$$\begin{bmatrix} 1 & 0 & 1 & -1/3 & | & 30 \\ 1/3 & 1 & 0 & 1/9 & | & 30 \\ -2/3 & 0 & 0 & 4/9 & | & z+120 \end{bmatrix}.$$

Это следующая каноническая форма для базиса x^B = (x_2, x_3). Свободные переменные: x^F = (x_1, x_4). Базисное допустимое решение определено как x = (x_1, x_2, x_3, x_4) = (0, 30, 30, 0). Значение целевой функции для него: z + 120 = 0, то есть $z = z_0 = -120$.

Шаг 2.

1°. $x_s = x_1$ (ведущий столбец – первый).

2°. $min\left(\frac{40}{1}, \frac{30}{1/3}\right) = min(40, 90) = 40$ (ведущая строка – первая),

$k = 1$.

3°. После нормировки:

$$\begin{matrix} \rightarrow \\ \\ \\ \end{matrix}\begin{bmatrix} 1 & 0 & 1 & -1/3 & | & 30 \\ 1/3 & 1 & 0 & 1/9 & | & 30 \\ -2/3 & 0 & 0 & 4/9 & | & z+120 \end{bmatrix}.$$
$$\uparrow$$

4°. После вычитаний:

$$\begin{bmatrix} 1 & 0 & 1 & -1/3 & | & 30 \\ 0 & 1 & -1/3 & 2/9 & | & 20 \\ 0 & 0 & 2/3 & 2/9 & | & z+140 \end{bmatrix}.$$

Это следующая каноническая форма для базиса x^B = (x_1, x_2). Свободные переменные: x^F = (x_3, x_4). Базисное допустимое решение определено как x = (x_1, x_2, x_3, x_4) = (30, 20, 0, 0). Значение целевой функции для него: z + 140 = 0, то есть $z = z_0 = -140$.

Шаг 3.

1°. Нет свободной переменной, у которой был бы отрицательный коэффициент, среди n первых переменных, $n = 4$, в последней строке матрицы. Процесс решения закончен. Решением данной задачи являются: $x_1 = 30$, $x_2 = 20$ и $z = -140$.

Замечание 3.3.1 Геометрическая интерпретация *симплекс-метода. Если* *изобразить эту задачу на плоскости (Ox_1x_2), то решение стартовало из точки начала координат ($x_1 = x_2 = 0$). После шага 1 решение переместилось в точку с координатами $x_1 = 0$, $x_2 = 30$ (вдоль ребра четырехугольника ограничений, так как x_1 оставалось неизменным, $x_1 = 0$). После шага 2 решение переместилось вдоль граничной линии $3x_1 + 9x_2 = 270$ в точку с координатами $x_1 = 30$, $x_2 = 20$. Если бы на шаге 1 в действии 1° был выбран вариант $x_s = x_1$, что тоже допустимо, то после такого шага 1 решение переместилось бы в точку $x_1 = 60$, $x_2 = 0$, а после шага 2 оно переместилось бы вдоль граничной линии $2x_1 + 3x_2 = 120$ в ту же окончательную точку $x_1 = 30$, $x_2 = 20$ и $z = -140$. Для этих двух вариантов действий число шагов до окончательного решения здесь оказалось одинаковым (2 шага). В общем случае такого совпадения не случается, и заранее трудно, а то и невозможно, предсказать, какой из вариантов действий приводит быстрее к окончательному решению.*

Замечание 3.3.2 *В симплекс-таблице позиция в правом нижнем углу отведена для целевой функции. Символ z в ней был записан выше для пояснения, и он более не нужен, так как здесь вычисления – не символьные. Само же числовое значение целевой функции всегда равно значению этого элемента таблицы, взятому с противоположным знаком, так как в точке любого базисного допустимого решения имеем $z - z_0 = 0$, где символ z опущен. С учетом этого, в начале алгоритма элемент в этой позиции равен 0, а в конце он равен 140, т.е. $z = -140$.*

3.4 Организация вычислений симплекс-метода при известном БР

Имеем симплекс-таблицу в виде двумерного массива (матрицы A) размера $(m + 1) \times (n + 1)$. Первые m строк содержат коэффициенты и правые части ограничений, представленных в канонической форме для базиса. С точностью до нумерации переменных x (то есть перестановки столбцов) эти ограничения в указанной форме имеют вид уравнений

$$[\,I \quad A'\,]\begin{bmatrix} x^B \\ x^F \end{bmatrix} = b'.$$

В этом пункте мы снимаем ограничение $b' \geq 0$. Последняя, $(m + 1)$–я строка, строка матрицы содержит коэффициенты целевой функции, имеющей в канонической форме для базиса следующий вид:

$$[\;0 \;\; c'^T]\begin{bmatrix} x^B \\ x^F \end{bmatrix} = z - z_0.$$

Последний, $(n + 1)$-й, элемент этой строки отведен для числового значения целевой функции, взятого с противоположным знаком. Это значение достигается целевой функцией на текущем базисном решении, равном ($x^B = b'$, $x^F = 0$), и в начале процесса вычислений оно равно 0.

Кроме этого, удобно ввести два указателя: (1) указатель $NB(i)$, $i = \overline{1,m}$, базисной переменной в i-м ограничении, и (2) указатель $NF(j)$, $j = 1, 2,\ldots, n - m$, свободных переменных в канонической форме для целевой функции $z - z_0$. В этих указателях хранятся номера базисных и, соответственно, свободных переменных.

На каждом шаге алгоритма выполняют следующие действия 1° – 4°:

1°. Пробегают номера свободных переменных в $NF(j)$, с тем чтобы проверить значения элементов

$$a(m + 1, NF(j)),\quad j = 1, 2,\ldots, n - m,$$

есть ли среди них отрицательные. Если «нет», решение найдено и определено равенствами $x^B = b'$, $x^F = 0$, $z_{min} = z_0$, то есть для базисных переменных:

$$x(NB(i)) = a(i, n + 1),\quad i = 1, 2,\ldots, m;$$

для свободных переменных:

$$x(NF(j)) = 0,\quad j = 1, 2,\ldots, n - m$$

и для целевой функции

$$z_{min} = - a(m + 1, n + 1).$$

Если «да», то выбирают из указанных отрицательных элементов один, например, наибольший по абсолютной величине. Пусть он имеет номер s, $s = NF(l)$. Это номер свободной переменной, переводимой в состав базисных переменных. Тем самым столбец с номером s объявлен ведущим столбцом матрицы и соответствующая переменная $x(s)$ объявлена ведущей.

2°. Пробегают номера строк, $i = 1, 2,\ldots, m$, чтобы по определенному правилу выделить ключевую (ведущую) строку, ее номер k. Это правило заранее определяют так, чтобы базисная переменная, имеющая номер $NB(i)$, не вышла из допустимой области в результате увеличения ведущей переменной, имеющей номер s. Для этого записывают i-е ограничение:

$$x(NB(i)) + a'_{is}x(s) = b'_i,$$

где a'_{is} – элемент приведенной выше подматрицы A'; b'_i – элемент вектора b' правой части ограничений. В принятых обозначениях для симплекс-таблицы (матрица A, ее элементы $a(i, j)$) это же ограничение имеет вид

$$x(NB(i)) + a(i, s)x(s) = a(i, n + 1).$$

Теперь в зависимости от варианта сочетаний знаков и критических (нулевых) значений для величин a'_{is} и $b'_i = a(i, n + 1)$, определяют, до какого предельного значения $max(x_s)$ можно увеличивать ведущую переменную $x(s)$, чтобы в результате этого переменная $x(NB(i))$ не вышла в область отрицательных (недопустимых) значений. Полный перебор вариантов сведем в табл. 3.4.1, в которой третий столбец отвечает на данный вопрос.

Таблица 3.4.1 Предел допустимого увеличения входящей переменной x_s, когда базисная переменная в i-м ограничении имеет коэффициент «+1»

$a'_{is} = a(i, s)$	$b'_i = a(i, n + 1)$	$max(x_s)$
> 0 < 0 $= 0$	> 0 > 0 > 0	b'_i / a'_{is} ∞ ∞
> 0 < 0 $= 0$	< 0 < 0 < 0	$-\infty$ (запрет) ∞ ∞ (*)
> 0 < 0 $= 0$	$= 0$ $= 0$ $= 0$	$-\infty$ (запрет) ∞ ∞

(*) – решение остается в недопустимой области

Нужно пройти таблицу для каждой строки, $i = 1, 2,\dots, m$, при этом: если появится хотя бы один «запрет», то переменную $x(s)$ как ведущую следует отклонить и вернуться к действию 1°, чтобы найти другую переменную $x(s)$ и проделать с ней действие 2°; если же нет ни одного «запрета», то следует выбрать ключевое (наиболее сильное) ограничение, его номер k, по правилу:

$$k = arg \min_{i:a'_{i,s}>0,\, b'_i>0} \left[\frac{b'_i}{a'_{is}} \right] = arg \min_{i:a'_{i,s}>0,\, b'_i>0} \left[\frac{a(i, n+1)}{a(i, s)} \right].$$

В результате этого объявляют коэффициент $a'_{ks} = a(k, s)$ ведущим элементом и k-ю строку – ключевой, или ведущей строкой, для выполнения следующих двух действий, составляющих один цикл исключения по методу Гаусса-Жордана для систем линейных алгебраических уравнений. Факт такого исключения означает перевод базисной переменной с номером $s = NF(l)$ в число небазисных. Такая «рокировка» должна быть зафиксирована в итоге регистрацией взаимного обмена номеров:

$$NB(l) \leftrightarrow NB(k).$$

3°. Нормируют строку k по ведущему элементу $a(k, s)$ делением на него всех элементов строки.

4°. Вычитают строку k, умноженную на коэффициент $a(k, s)$, из всех других строк, $i = 1, 2,\dots, m + 1$, кроме $i = k$.

В целом, приведенный алгоритм в действиях 3° и 4° представляет собой обычное исключение Гаусса-Жордана, которое предваряется специальным выбором ведущего элемента $a(k, s)$. Выбор ведущего элемента подчинен идее последовательного анализа допустимых вершин, смежных с текущей вершиной, чтобы среди них найти ту, в которой целевая функция будет уменьшена (действие 1°), но решение останется допустимым (действие 2°).

Замечание 3.4.1 *Видно, что в данном алгоритме всегда пересчитываются только (n – m + 1) столбцов симплекс-таблицы (матрицы A), а именно:*

(1) столбец с номером n + 1, где хранятся элементы правой части b' ограничений;

(2) все столбцы для свободных переменных, кроме ведущего, который заменяется тривиальным столбцом (все нули и одна единица вместо ведущего элемента a(k, s)), всего (n – m - 1) столбцов;

(3) столбец базисной переменной x(NB(k)), переводимой в число свободных переменных.

Остальные m столбцов можно не пересчитывать.

Пример 3.4.1 *Найти решение* $x_1 \geq 0$, $x_2 \geq 0$ *следующей задачи ЛП:*

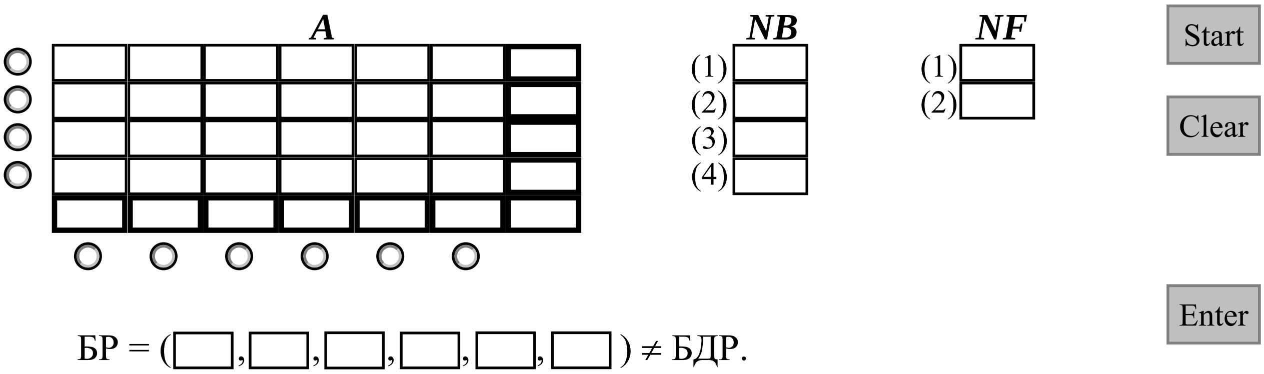
$$\begin{aligned} x_1 + x_2 &\leq 50; \\ -x_1 + 2x_2 &\leq 10; \\ x_1 &\geq 20; \\ x_2 &\geq 10; \\ -2x_1 - 3x_2 &= z \rightarrow min. \end{aligned}$$

Геометрическое построение показывает, что допустимая область есть четырехугольник на плоскости Ox_1x_2 с вершинами в точках: P = (20, 10); Q = (20, 15); R = (30, 20); S = (40, 10). Перемещая линию уровня z в направлении антиградиента – $grad(z)$ = (2, 3), определяем, что решение достигается в вершине R со значение z_{min} = – 120. Имея этот результат для проверки, выполним изложенный алгоритм формально.

Сначала запишем задачу в стандартной форме:

$$\begin{aligned} x_1 + x_2 + x_3 \qquad\qquad &= 50; \\ -x_1 + 2x_2 \qquad + x_4 \qquad &= 10; \\ x_1 \qquad\qquad - x_5 \quad &= 20; \\ x_2 \qquad\qquad - x_6 &= 10; \\ -2x_1 - 3x_2 \qquad\qquad &= z \rightarrow min. \end{aligned}$$

Заполняем массивы A(5, 7), NB(4) и NF(2) исходными данными:



Отмечено, что исходное базисное решение (БР) не является допустимым.

Шаг 1

1°. s $\boxed{\quad}$ (ведущий столбец);
l $\boxed{\quad}$ (номер ведущего столбца в NF) .
2°. min $\boxed{\quad}$ (минимум) ;
k $\boxed{\quad}$ (ведущая строка);
x_{ks} $\boxed{\quad}$ (ведущий элемент).

3°. После нормировки ведущей строки:

A

4°. После вычитаний и регистрации обмена $NF(l) \leftrightarrow NB(k)$:

A

NB

(1)	
(2)	
(3)	
(4)	

NF

(1)	
(2)	

БР = ($\boxed{\quad}$, $\boxed{\quad}$, $\boxed{\quad}$, $\boxed{\quad}$, $\boxed{\quad}$, $\boxed{\quad}$) ≠ БДР.

Шаг 2

1°. s $\boxed{\quad}$ (ведущий столбец);
l $\boxed{\quad}$ (номер ведущего столбца в NF) .
2°. min $\boxed{\quad}$ (минимум) ;
k $\boxed{\quad}$ (ведущая строка);
x_{ks} $\boxed{\quad}$ (ведущий элемент).

3°. После нормировки ведущей строки:

A

4°. После вычитаний и регистрации обмена $NF(l) \leftrightarrow NB(k)$:

A

NB

(1) $\square$
(2) $\square$
(3) $\square$
(4) $\square$

NF

(1) $\square$
(2) $\square$

Шаг 3

1°. Решение найдено:

$x_1 =$ $\square$
$x_2 =$ $\square$
$z_{min} =$ $\square$.

Repeat

В геометрической интерпретации решение было найдено путем перемещения из начала координат (0, 0) в точку (0, 5) и затем в точку R = (30, 20). Первые две точки не находились в допустимой области. Таким образом, увеличение ведущей переменной от нулевого значения позволяет не только уменьшать значение целевой функции, но и вводить решение в допустимую область.

Замечание 3.4.2 *По поводу ситуации – ∞ («запрет») в табл. 3.4.1 см. ниже пример 4.1.4, п. 4.1.*

3.5 Симплекс-метод без порождения начального БДР

Рассмотренный в пп. 3.3 и 3.4 алгоритм симплекс-метода пригоден для случая, когда какое-нибудь базисное допустимое решение известно с самого начала. Так происходит, если $b \geq 0$ и все m ограничений имеют вид «меньше или равно». Тогда для приведения каждого такого ограничения к виду «равно» в него вводят свою добавочную неотрицательную переменную (с коэффициентом «+ 1», см. табл. 3.4.1 для использования в алгоритме). Добавленные таким образом переменные как раз и являются базисными переменными x^B в начальном БДР ≥ 0.

Напомним принцип введения добавочных переменных в зависимости от вида исходного неравенства в ограничениях ЛП-задачи. Каждое ограничение, данное первоначально как неравенство типа «меньше или равно», должно быть преобразовано в равенство, чтобы осуществить переход к стандартной ЛП-задаче. Мы делаем это добавлением одной новой переменной слева в каждое такое неравенство и называем эту переменную «добавочной переменной». Она действует как «пружина», которая вставляется в «зазор» между левой частью и «правой частью» неравенства типа

«левая часть» ≤ «правая часть» (3.3)

и «растягивается» ровно настолько, чтобы превратить это неравенство в равенство

«левая часть» + «добавочная переменная» = «правая часть». (3.4)

Формально, определяя «добавочную переменную» выражением

«добавочная переменная» = «правая часть» – «левая часть» (3.5)

и применяя к ней общее требование неотрицательности любых переменных ЛП-задачи, получаем, что одновременно выполнены исходное неравенство (3.3) и равенство (3.4). Видно, что добавочная переменная должна входить в «левую часть» неравенства типа (3.3) с коэффициентом «+1».

Если же исходное ограничение имеет вид неравенства «больше или равно», то мы вычитаем слева новую «добавочную переменную». Эта переменная может быть уподоблена «пружине», которая вставляется в «зазор» между «левой частью» и «правой частью» неравенства типа

«левая часть» ≥ «правая часть» (3.6)

и «сжимается» ровно настолько, чтобы превратить это неравенство в равенство

«левая часть» – « добавочная переменная» = «правая часть». (3.7)

Формально, определяя «добавочную переменную» выражением

«добавочная переменная» = «левая часть» – «правая часть»

и применяя к ней общее требование неотрицательности любых переменных ЛП-задачи, получаем, что одновременно выполнены исходное неравенство (3.6) и равенство (3.7). Видно, что добавочная переменная должна входить в «левую часть» неравенства типа (3.6) с коэффициентом «-1».

Добавочные переменные, введенные таким образом в каждое неравенство, полностью независимы друг от друга, те есть они появляются в стандартной ЛП-задаче с коэффициентами, которые образуют все столбцы единичной матрицы, взятые либо со знаком «+», либо со знаком «-». Кроме того, добавочные переменные не входят в данную целевую функцию z. По этим признакам, такие добавочные переменные как раз и образуют набор базисных переменных x^B, и начальное базисное решение (БР), с которого стартует симплекс-метод, вполне очевидно.

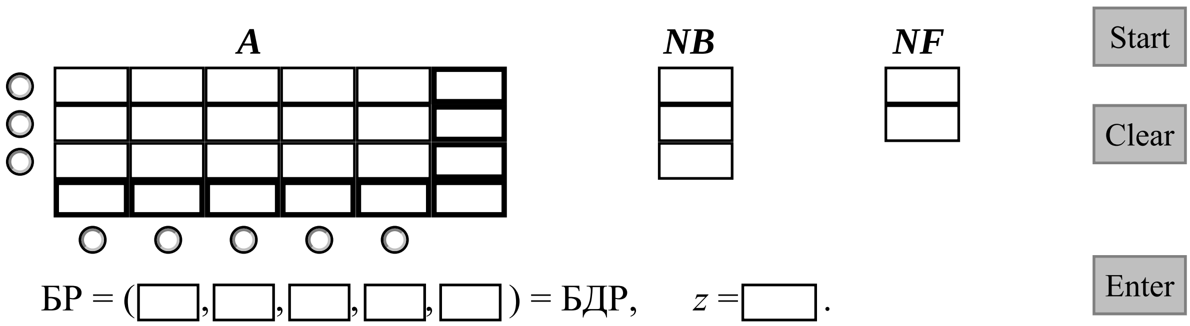
Замечание 3.5.1 *Можно предварительно умножить неравенства типа (3.6) на «-1», чтобы превратить их в неравенства типа (3.3) и тем самым сделать все коэффициенты для добавочных переменных строго равными столбцам единичной матрицы. Однако может появиться ситуация, что симплекс-метод стартует из недопустимой области, хотя, как видно из последнего примера 3.4.1, и при недопустимом начальном БР данный алгоритм успешно работает. Очевидно, если такая ситуация существует, то она существует в силу природы задачи (вне зависимости от указанного* *предварительного умножения), - см. предыдущий пример 3.4.1 и следующий пример 3.5.2.*

Если кроме ограничений вида «меньше или равно» или «больше или равно», имеется ограничение типа «равно», то, формально, нет необходимости вводить в него дополнительную переменную, однако, тогда начальное БР не очевидно. Поэтому в такое ограничение также вводят не добавочную, а *искусственную переменную*, роль которой совершенно иная, а именно: порождение начального БР, как изложено в следующем п. 3.6. Очевидно, что любая искусственная переменная после выполнения этой своей роли должна стать равной нулю. Для этого будет введена *искусственная целевая функция*, равная просто сумме всех искусственных переменных, и процесс решения начнется с минимизации этой функции. Соответственно, если невозможно сделать ее (а значит, и любую искусственную переменную) равной нулю в процессе решения (см. п. 3.6), задача ЛП решения не имеет.

Пример 3.5.1 *Для* $x_1, x_2 \geq 0$ *решить задачу:*

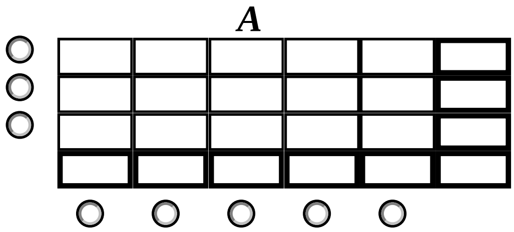
$$\begin{aligned} -2x_1 + 3x_2 &\leq 3; \\ 2x_1 + 3x_2 &\leq 21; \\ x_1 + x_2 &\leq 6; \\ -2x_1 - 5x_2 &= z \to min. \end{aligned}$$

Шаг 0. Ввод числовых коэффициентов из данных условий и автоматическое заполнение остальных числовых значений для получения следующего исходного вида массивов:

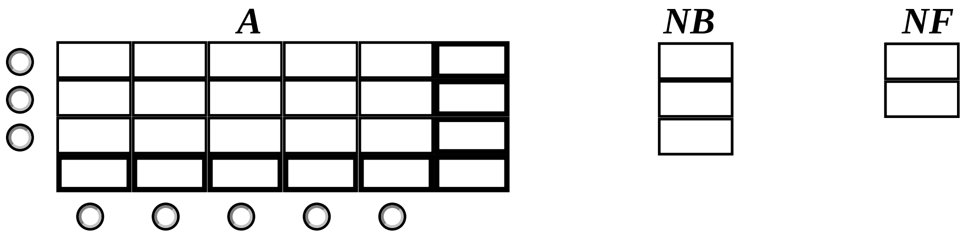


Шаг 1.

1°. s $\boxed{\quad}$,
l $\boxed{\quad}$.
2°. min $\boxed{\quad}$,
k $\boxed{\quad}$.
3°. После нормировки:



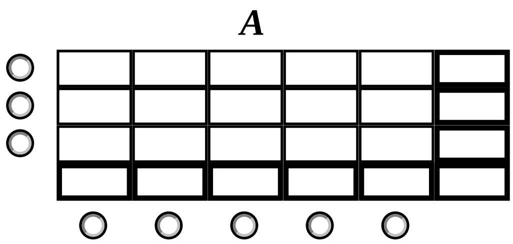
4°. После вычитаний:



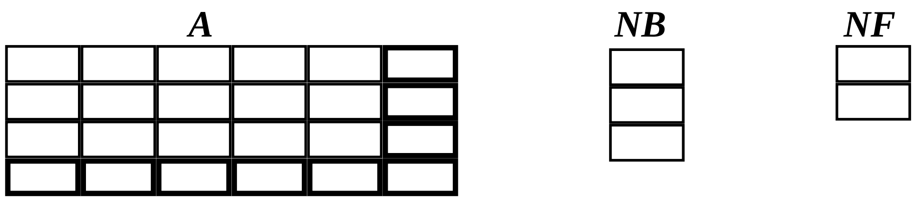
БР = ($\boxed{\quad}$, $\boxed{\quad}$, $\boxed{\quad}$, $\boxed{\quad}$, $\boxed{\quad}$) = БДР; z = $\boxed{\quad}$.

Шаг 2.

1°. s $\boxed{\quad}$,
l $\boxed{\quad}$.
2°. min $\boxed{\quad}$,
k $\boxed{\quad}$.
3°. После нормировки:



4°. После вычитаний:



БР = ($\boxed{\quad}$, $\boxed{\quad}$, $\boxed{\quad}$, $\boxed{\quad}$, $\boxed{\quad}$) = БДР; $min\ z$ = $\boxed{\quad}$.

Шаг 3.
1°. Нет отрицательных элементов среди $a(m+1, NF(l))$, $l = 1, 2$. Следовательно,

оптимальное базисное допустимое решение найдено:

ОБДР = ($\boxed{}$,$\boxed{}$,$\boxed{}$,$\boxed{}$); $min\ z$ = $\boxed{}$. Repeat

В геометрической интерпретации здесь на плоскости Ox_1x_2 происходило движение по точкам: (0, 0) → (0, 1) → (3, 3).

Рассмотрим теперь пример, где в ограничениях присутствуют неравенства любых типов. Мы здесь не будем, в отличие от примера 3.4.1, предварительно умножать неравенство типа (3.6) на «-1». В строках для неравенств этого типа, по причине указанного неумножения их на «-1», роль базисных переменных будут исполнять добавочные переменные с коэффициентами «-1». Следовательно, вместо табл. 3.4.1 понадобится другая таблица во время действия 2° симплекс-алгоритма, а именно, следующая табл. 3.5.1. В ней третий столбец указывает предел допустимого увеличения входящей переменной x_s для данного случая.

Таблица 3.5.1 Предел допустимого увеличения входящей переменной x_s, для случая, когда базисная переменная в i-м ограничении имеет коэффициент «-1»

$a'_{is} = a(i, s)$	$b'_i = a(i, n + 1)$	$max(x_s)$
> 0	> 0	∞
< 0	> 0	$-\infty$ (запрет)
$= 0$	> 0	∞ (*)
> 0	< 0	∞
< 0	< 0	b'_i / a'_{is}
$= 0$	< 0	∞ (*)
> 0	$= 0$	∞
< 0	$= 0$	$-\infty$ (запрет)
$= 0$	$= 0$	∞

(*) – решение остается в недопустимой области

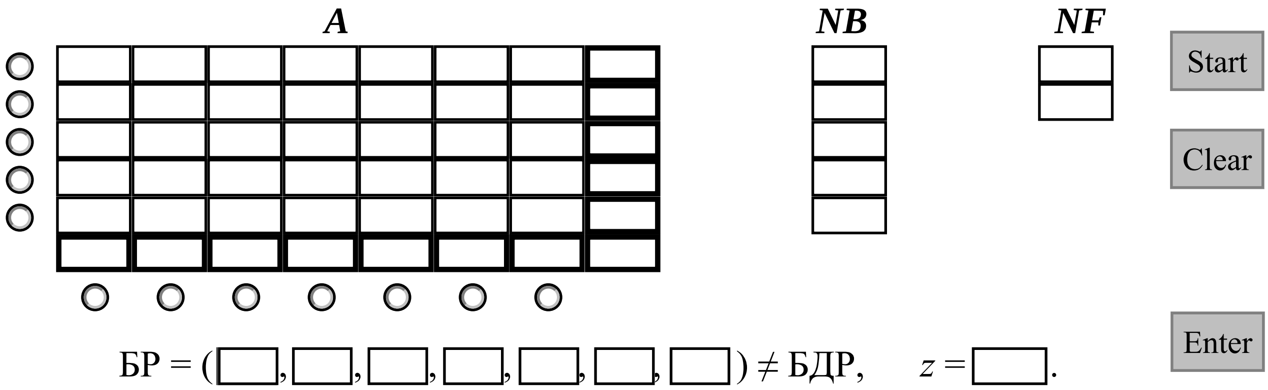
Кроме того, в ограничение типа «=» введем еще переменную с коэффициентом «+1», подразумевая под этим некоторую искусственную переменную, которая в итоге должна стать равной нулю, как выше отмечалось (непосредственно перед примером 3.5.1). Но заметим предварительно, что это введение мы делаем внешне неотличимо от случая ограничения типа (3.3). Что при этом получится, обсудим после следующего примера.

Пример 3.5.2 *Для* x_1, $x_2 \geq 0$ *решить задачу:*

$$\begin{aligned} 2x_1 \quad\quad &\geq 3; \\ x_2 &\geq 1; \end{aligned}$$

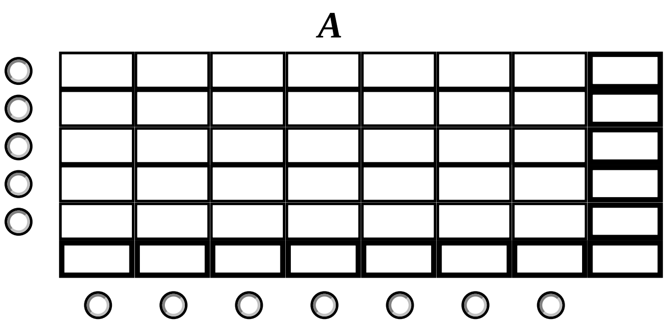
$$\begin{aligned} -\ 2x_1 + 3x_2 &\le 3; \\ 2x_1 + 3x_2 &\le 21; \\ x_1 + x_2 &= 6; \\ -2x_1 - 5x_2 &= z \to min. \end{aligned}$$

Шаг 0. Ввод числовых коэффициентов из данных условий и автоматическое заполнение остальных числовых значений для получения следующего исходного вида массивов:

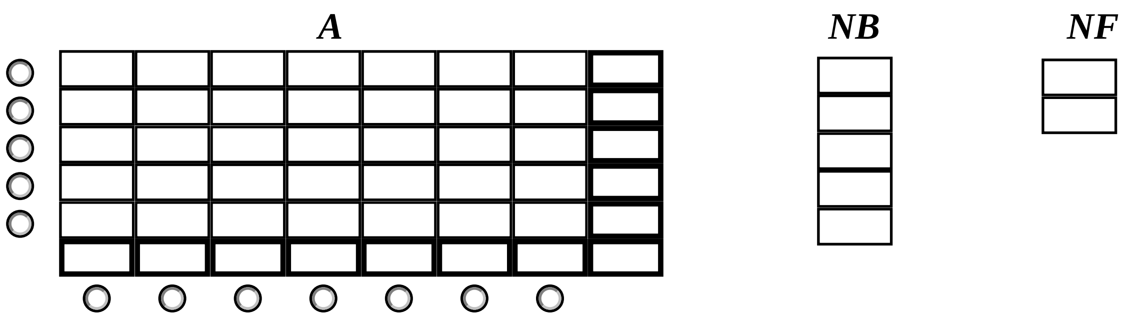


Шаг 1.

1°. s ,
l .
2°. min ,
k .
3°. После нормировки:



4°. После вычитаний:



БР = (□,□,□,□,□,□,□) ≠ БДР, z = □.

Шаг 2.

1°. s □,
l □.
2°. min □,
k □.
3°. После нормировки:

A

4°. После вычитаний:

A

NB

NF

БР = (□,□,□,□,□,□,□) ≠ БДР, z = □.

Шаг 3.

1°. Нет отрицательных элементов среди $a(m+1, NF(l))$, $l = 1, 2$, и NF содержит 7 (номер искусственной переменной, введенной на шаге 0 для третьего ограничения; имеем $x_7 = 0$). Следовательно, оптимальное базисное допустимое решение может быть выдано (опуская x_7) в виде:

ОБДР = (□,□,□,□,□,□); $min\ z$ = □.

Repeat

В геометрической интерпретации здесь на плоскости Ox_1x_2 происходило движение по точкам: (0, 0) → (0, 1) → (3, 3), причем первые две точки лежат вне допустимой области.

Определение 3.5.1 *Базис называется* **корректным***, если каждая базисная переменная стандартной ЛП-задачи появляется в (одном) ограничении с коэффициентом «+1» и не появляется в целевой функции. Если вместо этого, существует хотя бы одна базисная переменная, которая появляется с коэффициентом «-1», то такой базис называется* **некорректным**.

Таким образом, при корректном базисе мы оперируем только с табл. 3.4.1, тогда как в случае некорректного базиса есть потребность и в табл. 3.5.1.

Замечание 3.5.2 *В примере 3.5.2 третье ограничение имеет вид «равно», но данный пример показывает, как* **не нужно** *действовать в случае таких ограничений. Этот пример также демонстрирует, каким должно быть различие между добавочными и искусственными переменными. Каждая искусственная переменная должна остаться равной нулю в результате решения, в то время как добавочная не обязана. Хотя в данном примере искусственная переменная* x_7 *в конце решения равна нулю и ответ (результат) правильный, ход решения не является таковым, потому что он не отличим от случая, когда данное третье ограничение было бы заменено в условии задачи на «меньше равно». Ответ правильный, потому что ограничение* $x_1 + x_2 \le 6$ *«поглощено ограничением»* $2x_1 + 3x_2 \le 21$*, то есть это последнее ограничение могло бы быть без ущерба отброшено как более широкое. Как нужно действовать в случае ограничений типа «равно», показано ниже в* п. *3.6 и в примере 4.1.4. Этому посвящен также отдельный* п. *4.3 и, кроме того,* п. *5.7.*

3.6 Симплекс-метод с порождением БДР

Предыдущий алгоритм (п. 3.4 и п. 3.5) не исключает появления базисных решений, не являющихся допустимыми, в исходном состоянии или на промежуточных шагах. Чтобы исключить такую ситуацию, применяют следующий алгоритм с искусственными переменными. Его идея заключается в том, чтобы определить некоторую новую ЛП-задачу, для которой начальный базисный вектор легко обнаружить и решение которой приводит к искомому начальному БДР исходной ЛП-задачи.

Продемонстрируем эту идею на предыдущем примере 3.5.2:

минимизировать $z = -2x_1 - 5x_2$

с ограничениями:
$$\begin{cases} 2x_1 \ge 3; \\ x_2 \ge 1; \\ x_1 + x_2 = 6; \\ -2x_1 + 3x_2 \le 3; \\ 2x_1 + 3x_2 \le 21; \\ x_i \ge 0 \quad (i = 1, 2). \end{cases}$$

Из шага 0 примера 3.5.2 очевидно, что точка $x_1 = 0$, $x_2 = 0$ не удовлетворяет ограничениям из-за требований $2x_1 \geq 3$, $x_2 \geq 1$ и $x_1 + x_2 = 6$, поэтому мы должны найти некоторый другой начальный вектор. Если, как обычно, мы введем добавочные переменные x_3 и x_4 для первого и второго нарушенных неравенств, то получим взамен два неравенства:

$$\begin{aligned} 2x_1 \quad - x_3 \quad\quad &= 3; \\ x_2 \quad\quad - x_4 &= 1, \end{aligned}$$

с $x_3 \geq 0$, $x_4 \geq 0$. Невозможность положить $x_1 = x_2 = 0$ для этих неравенств показывает, в действительности, что $x_1 = x_2 = 0$ влечет $x_3 = -3$ и $x_4 = -1$ в нарушение требований $x_3 \geq 0$ и $x_4 \geq 0$.

Добавочные переменные x_5 и x_6, введенные в четвертое и пятое (ненарушенные) неравенства, не вызывают такой проблемы, поскольку

$$\begin{aligned} -2x_1 + 3x_2 + x_5 \quad\quad &= 3; \\ 2x_1 + 3x_2 \quad\quad + x_6 &= 21, \end{aligned}$$

и равенство $x_1 = x_2 = 0$ не нарушает условий $x_5 \geq 0$, $x_6 \geq 0$.

Маневр с неравенствами типа «≥» (и, как будет видно, с ограничениями типа «=») заключается во введении одной или более дополнительных переменных, - в этом примере x_7, x_8 и x_9, соответственно, для первого, второго и третьего нарушенных ограничений, - так называемых **искусственных переменных**, и затем:

1) в замене переменной x_3 в первом уравнении $2x_1 - x_3 = 3$ на $x_3 - x_7$ с тем, чтобы мы могли записать первое ограничение $2x_1 \geq 3$ в виде равенства $2x_1 - x_3 + x_7 = 3$ с $x_3 \geq 0$ и $x_7 \geq 0$;
2) в замене переменной x_4 во втором уравнении $x_2 - x_4 = 1$ на $x_4 - x_8$ с тем, чтобы мы могли записать второе ограничение $x_2 \geq 1$ в виде $x_2 - x_4 + x_8 = 1$ с $x_4 \geq 0$ и $x_8 \geq 0$ и
3) в записи третьего ограничения $x_1 + x_2 = 6$ как $x_1 + x_2 + x_9 = 6$ с $x_9 \geq 0$.

Теперь мы можем положить $x_1 = x_2 = 0$, так как можно легко обеспечить $x_7 - x_3 = 3$, $x_8 - x_4 = 1$ и $x_9 = 6$ при том, что как добавочные переменные (x_3, x_4, x_5 и x_6), так и искусственные переменные (x_7, x_8 и x_9) все неотрицательны. Мы, следовательно, рассматриваем расширенную систему ограничений:

$$\begin{aligned} 2x_1 \quad - x_3 \quad\quad\quad\quad + x_7 \quad\quad &= 3; \\ x_2 \quad - x_4 \quad\quad\quad\quad + x_8 \quad &= 1; \\ x_1 + x_2 \quad\quad\quad\quad\quad\quad\quad + x_9 &= 6; \\ -2x_1 + 3x_2 \quad\quad + x_5 \quad\quad\quad\quad &= 3; \\ 2x_1 + 3x_2 \quad\quad\quad + x_6 \quad\quad\quad &= 21, \end{aligned} \tag{3.8}$$

для которой базисный допустимый вектор очевиден как совокупность следующих значений: $x_1 = x_2 = x_3 = x_4 = 0$, $x_5 = 3$, $x_6 = 21$, $x_7 = 3$, $x_8 = 1$ и $x_9 = 6$. Если мы сможем каким-либо образом стартовать от этого вектора и, в конце концов, сможем найти еще одно БДР для расширенной системы ограничений (3.8), но такое, что все искусственные переменные (здесь x_7, x_8 и x_9) равны нулю, то первые шесть компонент этого решения образуют некоторый базисный допустимый вектор для исходной ЛП-задачи (только с добавочными переменными), а именно, - для исходных ограничений, записанных в виде уравнений

$$
\begin{aligned}
2x_1 \quad - x_3 \qquad\qquad &= 3;\\
x_2 \quad - x_4 \qquad &= 1;\\
x_1 + x_2 \qquad\qquad &= 6;\\
-2x_1 + 3x_2 \qquad + x_5 \quad &= 3;\\
2x_1 + 3x_2 \qquad\qquad + x_6 &= 21;\\
x_i \ge 0 \ \ (\text{для } i = 1,\ldots,6).
\end{aligned}
\tag{3.9}
$$

Обратно, любое БДР для последней системы ограничений (3.9), включающей только исходные и добавочные переменные, порождает некоторый базисный допустимый вектор для вышеуказанной расширенной системы ограничений (3.8) (расширенной с добавлением искусственных переменных) со всеми искусственными переменными равными нулю: $x_7 = 0$, $x_8 = 0$ и $x_9 = 0$. Следовательно, ограничения, о которых мы действительно заботимся, то есть исходные ограничения, имеют некоторое БДР, если и только если расширенная система ограничений имеет такой базисный допустимый вектор, для которого все искусственные переменные равны нулю, в данном примере, $x_7 = x_8 = x_9 = 0$. А так как все они, по общему требованию, должны быть неотрицательны, мы, в действительности, пытаемся *минимизировать* сумму искусственных переменных. В данном примере:

$$\text{минимизировать } w = x_7 + x_8 + x_9$$

$$\text{с ограничениями (3.8) и } x_i \ge 0 \ (\text{для } i = 1,\ldots,9).$$

Следовательно, мы вынуждены решить эту вспомогательную ЛП-задачу с искусственной целевой функцией w, равной простой сумме всех искусственных переменных, прежде чем решать исходную задачу. Так как искусственные переменные должны быть неотрицательны, $x_7 \ge 0$, $x_8 \ge 0$, $x_9 \ge 0$, имеются две возможности:

(1) Минимальное значение для w действительно есть 0. Это означает, что мы можем найти некоторое БДР для (3.8), в котором $x_7 = x_8 = x_9 = 0$, то есть мы можем найти БДР для (3.9).

(2) Минимальное значение для w больше, чем 0. Это означает, что никакого БДР для (3.9) не существует, так как предположив существование такого решения, мы могли бы найти некоторое БДР для (3.8) с $x_7 = x_8 = x_9 = 0$, то есть $w = 0$, и это противоречило бы нашему предположению.

Теперь можно продолжить наше рассмотрение метода на том же самом примере. Посмотрим, как этот метод работает от начала до конца.

Пример 3.6.1 *Решим задачу с условием из примера 3.5.2, одновременно иллюстрируя возможную организацию вычислений для общего случая, включающего ограничения любых типов.*

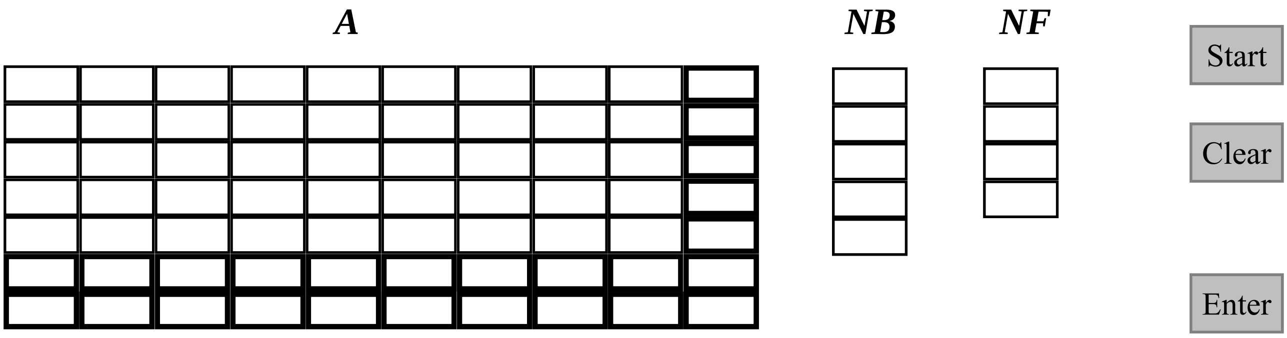
Шаг 0. Ввод значений для следующих исходных величин:
np – исходное количество переменных, здесь *np* = 2.
ng – количество ограничений вида «≥», здесь *ng* = 2.
ne – количество ограничений вида «=», здесь *ne* = 1.
nl – количество ограничений вида «≤», здесь *nl* = 2.

Вычисление вспомогательных величин:
ns = *ng* + *nl* – число искусственных переменных, здесь *ns* = 4.
na = *ng* + *ne* – число добавочных переменных, здесь *na* = 3.
n = *np* + *na* + *ns* – общее число добавочных переменных, здесь *n* = 9.
m = *na* + *nl* – общее число ограничений, здесь *m* = 5.

Выделение памяти для массивов:
(1) *A*(1..*m* + 2, 1..*n* + 1), здесь *A*(1..7, 1..10) – матрица (7× 10).
(2) *NB*(1..*m*), здесь *NB*(1..5) – вектор размерности 5.
(3) *NF*(1..*n* – *m*), здесь *NF*(1..4) – вектор размерности 4.

Ввод коэффициентов из условий задачи в части матрицы А:
(1) *A*(1..*m* + 1, 1..*np*), здесь *A*(1..6, 1..2),
(2) *A*(1..*m* + 1, *n*+1), здесь *A*(1..6, 10).

Заполнение остальных элементов массивов; в результате



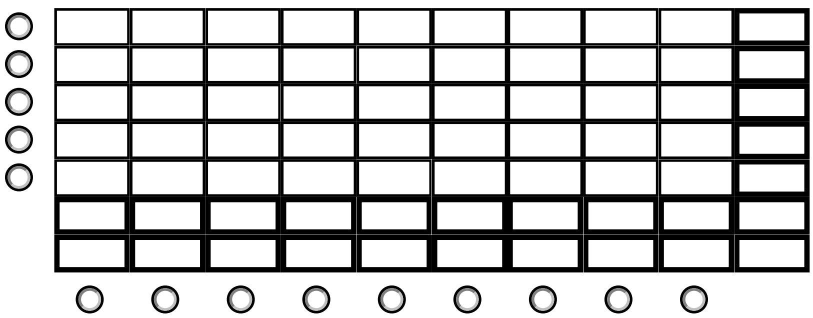
БР = ($\boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{}$) = БДР, z_{min} = $\boxed{}$, w = $\boxed{}$.

Замечание 3.6.1 *Из этого заполнения массивов видно, что переменные распределены следующим образом:*
x_1 *и* x_2 *– исходные переменные,*
x_3, x_4, x_5 *и* x_6 *– добавочные переменные,*
x_7, x_8 *и* x_9 *– искусственные переменные.*
Такое упорядочение переменных может быть закреплено в программной реализации. Искусственные переменные вводят только в неравенства вида «≥» и «=». В матрице A снизу добавлена еще одна строка, в которой хранят коэффициенты для искусственной целевой функции w. Для значения этой

функции с противоположным знаком (–w) отведен элемент на пересечении нижней строки и правого столбца. Вторая снизу строка, как и ранее, предназначена для коэффициентов основной целевой функции и ее значения с противоположным знаком (–z). Значение –z хранит крайний справа элемент этой строки.

Рассматриваемый симплекс-метод с искусственными переменными состоит из двух этапов. Этап I предназначен для решения вспомогательной ЛП-задачи, в то время как Этап II – для решения основной. Шаг 0, рассмотренный выше в примере, есть только установочный шаг. Чтобы подготовится к основным (рабочим) шагам Этапа I, нам нужно выразить искусственную целевую функцию w только через небазисные переменные. В этом примере, $w = x_7 + x_8 + x_9$ по определению, но все искусственные переменные (x_7, x_8, x_9) должны появится а начальном БДР в качестве базисных переменных. В силу этого требования, шаг 0 должен быть завершен исключением искусственных переменных из нижней строки. Это достигается путем вычитаний всех тех уравнений, куда были введения искусственные переменные, из нижней строки. При условии, что принято указанное выше упорядочение переменных (см. Замечание 3.6.1), первые *na* строк нужно вычесть из последней строки. Здесь *na* = 3, и после такого подготовительного вычитания мы имеем:

A



В результате этого вычитания получено правильное (исправленное) начальное значение искусственной целевой функции: $w = 10$.

Замечание 3.6.2 *При принятом порядке записи ограничений (сначала «≥», затем «=» и потом «≤») базисными переменными в начальном БДР будут всегда те переменные, которые имеют номера, начиная с (np + ng + 1) и кончая n = np + na +ns, в данном примере с 5 по 9.*

С этого момента начинается Этап I, где применяют обычный симплекс-алгоритм (см пп. 3.3 и 3.4) для решения вспомогательной задачи линейного программирования: $w \to \min$.

Шаг 1.

1°. s ☐ ,
l ☐ .
2°. min ☐ ,
k ☐ .
3°. После нормировки:

A

4°. После вычитаний:

A

NB

NF

БР = (☐ , ☐ , ☐ , ☐ , ☐ , ☐ , ☐ , ☐ , ☐) = БДР, z = ☐ , w = ☐ .

Шаг 2.

1°. s ☐ ,
l ☐ .
2°. min ☐ ,
k ☐ .
3°. После нормировки:

A

4°. После вычитаний:

A

NB

NF

БР = ($\square$, $\square$, $\square$, $\square$, $\square$, $\square$, $\square$, $\square$, $\square$) = БДР, $z =$ $\square$, $w =$ $\square$.

Шаг 3.

1°. s $\square$,
l $\square$.
2°. min $\square$,
k $\square$.
3°. После нормировки:

A

4°. После вычитаний:

A

NB

NF

БР = ($\square$, $\square$, $\square$, $\square$, $\square$, $\square$, $\square$, $\square$, $\square$) = БДР, $z =$ $\square$, $w =$ $\square$.

Шаг 4.

1°. *s* ____,
    l ____.
2°. *min* ____,
    k ____.
3°. После нормировки:

A

4°. После вычитаний:

A

NB

NF

БР=(__, __, __, __, __, __, __, __, __)=БДР, min(z) =__, min(w) =__.

Repeat

Этап I заключается в последовательном уменьшении w до нуля, и этим он завершился, так как нет отрицательных коэффициентов в нижней строке матрицы A. Здесь Этап II не потребовался, так как во второй снизу строке матрицы A нет отрицательных коэффициентов. Минимумы для z и w достигнуты одновременно. В геометрической интерпретации здесь на плоскости Ox_1x_2 были переходы по следующим точкам: (0, 0) → (3/2, 0) → (3/2, 1) → (3/2, 2) → (3, 3).

Чтобы показать, как выполняют Этап II, выберем после шага 2 другой возможный путь. Рассмотрим эти действия.

Шаг 3.

Next

1°. s ,

l .

2°. min ,

k .

3°. После нормировки:

A

4°. После вычитаний:

A

NB

NF

БР = (, , , , , , , ,) = БДР, $z =$, $w =$.

Признаком успешного окончания Этапа I является 0 для w и неотрицательность всех коэффициентов для w в нижней строке. Если бы все эти коэффициенты оказались неотрицательны, и w не обратилась в 0, то Этап I не мог быть начат: исходные ограничения не имели бы БДР. Это был бы случай *противоречивых ограничений* (см. также разд. 6).

Замечание 3.6.3 *Формальная причина для появления случая противоречивых ограничений заключается в том, что БДР, полученное после Этапа I, содержит, по крайней мере, одну искусственную переменную среди базисных переменных (следовательно, $w \neq 0$).*

Возвращаясь к нашему примеру, переходим к Этапу II.

В данном варианте продолжения примера Этап II возможен, и он необходим, так как z не достигла минимума: имеется отрицательный коэффициент –3 в предпоследней строке матрицы A. Для Этапа II уже не нужны ни искусственные переменные, ни искусственная целевая функция. Поэтому они в дальнейшем вычеркнуты из всех действий. Вычеркивают те столбцы матрицы A, для которых в последней строке получены единицы, а также всю последнюю строку. Это те столбцы, которые при данном упорядочении переменных имеют номера с ($np + na + 1$) по n, здесь с 7 по 9. Кроме того, из вектора NF вычеркивают элементы, содержащие эти же номера; при данном упорядочении это всегда первые ns элементов вектора, здесь $ns = 3$.

Этап II.
Шаг 4.

Next

1°. s ____ ,
l ____ .
2°. min ____ ,
k ____ .
3°. После нормировки:

A

4°. После вычитаний:

A

NB

NF

БР = (____, ____, ____, ____, ____, ____) = БДР, min(z) = ____.

Repeat

В геометрической интерпретации здесь на плоскости Ox_1x_2 здесь было движение по точкам: (0, 0) → (3/2, 0) → (3/2, 1) → (5, 1) → (3, 3).

4
Двойственный симплекс-метод

Линия (гиперплоскость) уровня $z = z_{min}$ делит все пространство переменных задачи ЛП на две части: в одной из них $z > z_{min}$ и в ней находится допустимая область, а в другой $z < z_{min}$ и в ней нет допустимой области значений переменных. Обычный симплекс-метод, изложенный выше, работает, если он стартует из области, где $z > z_{min}$. Если же стартовое базисное решение (БР) лежит в области, где $z < z_{min}$, то требуется не уменьшать, а увеличивать z до значения z_{min}, переходя от недопустимых БР до БДР. Для этого предназначен двойственный симплекс-метод. Другая ситуация, для которой он создан, может возникнуть в результате наложения дополнительных ограничений на уже решенную задачу. Она обязательно возникнет, если наложенное, дополнительное ограничение не содержит целиком исходное допустимое множество, а отделяет от него лишь некоторую часть. Тогда, естественно, новое z_{min} больше прежнего, из которого двойственный симплекс-метод и стартует, чтобы не решать всю задачу как новую.

Поскольку при движении из области $z < z_{min}$ требуется увеличивать z, пребывание в этой области обнаруживается тем условием, что все коэффициенты целевой функции, в канонической форме для базиса, неотрицательны. Движение к z_{min} обеспечивается последовательным исключением отрицательных переменных из базисных решений. Их отсутствие делает базисное решение допустимым и, при сохранении указанного условия, оптимальным.

4.1 Алгоритм с корректным видом базиса

Неважно, из-за чего сложилась ситуация готовности для работы двойственного симплекс-метода. Она определена двумя признаками: (1) нет отрицательных коэффициентов целевой функции в канонической форме для базиса и (2) соответствующее базисное решение не является допустимым (содержит отрицательные значения переменных). Чтобы легче обнаружить второй признак, вхождение базисных переменных (x^B) в ограничения делают всегда с коэффициентом «+1». Это **корректный вид базиса**. Чтобы его с самого начала иметь, возможное появление коэффициента «-1» устраняют предварительным умножением соответствующей строки ограничений на «-1». Следовательно, используют известную каноническую форму задачи для базиса,

$$\begin{bmatrix} I & A' \\ O^T & c'^T \end{bmatrix} \begin{bmatrix} x^B \\ x^F \end{bmatrix} = \begin{bmatrix} b' \\ z - z_0 \end{bmatrix}, \quad c' \geq 0,$$

с базисным решением БР = $\{x^B = b',\ x^F = 0\}$. Тем самым проверка второго признака сведена к ответу на вопрос, нет ли отрицательных элементов в векторе b'. С этого начинается любой шаг двойственного симплекс-метода. В целом же каждый его шаг распадается на следующие 4 действия:

Действие 1°. Проверяют элементы b'. Если среди них нет отрицательных, то z_{min} = x_0 найден со значением БДР = ($x^B = b'$, $x^F = 0$). Если есть несколько отрицательных элементов, то выбирают ту строку ограничений, ее номер k, куда входит наиболее отрицательный элемент b'_k вектора b'. Тем самым найдена одна базисная переменная x_r, $r = NB(k)$, которая должна быть далее выведена из состава x^B, из-за того, что $x_r = b'_k < 0$.

Далее возникает вопрос, какую из небазисных, свободных переменных в составе x^F ввести в базис путем ее увеличения от нулевого значения взамен выводимой переменной x_r? Допустим, этот вопрос решен, и выбрана свободная переменная x_s. Она предназначена для исключения из всех уравнений канонической формы, кроме k-го уравнения, которое должно быть предварительно нормировано делением на ведущий элемент $a'_{ks} < 0$. Благодаря такой нормировке, сохраняют корректность вида базиса и устраняют один отрицательный элемент в столбце b'. Исключение же в строке для целевой функции важно сделать таким, чтобы сохранить первый из указанных признаков для применимости и в последующем симплекс-метода. Чтобы прояснить это, запишем формулу пересчета j-го коэффициента целевой функции при исключении:

$$c_j^+ = c_j' - c_s' a_{kj}'' .$$

Здесь a''_{kj} есть j-ый коэффициент строки после ее нормировки:

$$a_{kj}'' = a_{kj}' / a_{ks}' .$$

Первый признак требует $c^+_j \geq 0$ при исходных $c_j' \geq 0$ и $c'_s \geq 0$. Ясно, что только для тех столбцов j существует опасность нарушить требование $c^+_j \geq 0$, для которых a''_{kj}. Так как $c^+_j \geq 0$ равносильно неравенству

$$\frac{c_j'}{\left|a_{kj}'\right|} \geq \frac{c_s'}{\left|a_{ks}'\right|},$$

То выбор s, номера ведущей переменной x_s, оказывается, подчинен условию

$$\frac{c_s'}{\left|a_{ks}'\right|} = \min_{j:a_{kj}'<0} \frac{c_j'}{\left|a_{kj}'\right|} . \tag{4.1}$$

Таким образом, выполняют следующие действия:

Действие 2°. В k-ой (ведущей) строке ищут только среди свободных переменных отрицательные коэффициенты a'_{kj}. Если их нет, то не существует базисного допустимого решения (выход из алгоритма с этим сообщением). Если таковые есть, то среди них выбирают один, a'_{ks}, его номер s, для которого выполнено только указанное условие (4.1). Элемент a'_{ks} объявляют ведущим.

Действие 3°. Нормируют k-ю строку делением на a'_{ks}.

Действие 4°. Вычитают k-ю строку, умноженную на элемент a'_{ks} матрицы задачи, из каждой i-ой строки (i ≠ k).

Отсюда видно, что двойственный симплекс-метод равносилен исключению переменных по схеме Гаусса-Жордана в действиях 3°, 4° при выборе на каждом шаге ведущего элемента по специальным правилам в действиях 1°, 2°.

Пример 4.1.1 *Найти неотрицательное решение* $x_1, x_2 \geq 0$ *задачи*

$$2x_1 + 3x_2 \leq 120;$$
$$3x_1 + 9x_2 \leq 270;$$
$$-2x_1 - 4x_2 = z \quad \rightarrow \ min$$

двойственным симплекс-методом.

Прежде всего запишем задачу в исходном стандартном виде с канонической формой для базиса:

2 3 1 0	120
3 9 0 1	270
-2 -4 0 0	0

$$A = \begin{pmatrix} 2 & 3 & 1 & 0 \\ 3 & 9 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 120 \\ 270 \end{pmatrix}, \quad c^T = \begin{pmatrix} -2 & -4 & 0 & 0 \end{pmatrix}, \quad Ax = b, \quad c^T x = z.$$

Эта форма отвечает базисному решению БР = (0, 0, 120, 270) = БДР и $z = 0$, которое изображается точкой (0,0) на плоскости Ox_1x_2. Всего же здесь имеется $C_4^2 = 6$ вариантов выбора двух линейно независимых векторов-столбцов ограничений из четырех: (1,2), (1,3), (1,4), (2,3), (2,4), (3,4).

Вариант (3,4) записан выше. Запишем и все остальные варианты, чтобы иметь полную картину базисных решений, с их изображением на плоскости.

Вариант (1,2)

$$B = \begin{pmatrix} 2 & 3 \\ 3 & 9 \end{pmatrix}, \quad B^{-1} = \begin{pmatrix} 1 & -1/3 \\ -1/3 & 2/9 \end{pmatrix}, \quad B^{-1}A = \begin{pmatrix} 1 & 0 & 1 & -1/3 \\ 0 & 1 & -1/3 & 2/9 \end{pmatrix}, \quad b' = B^{-1}b = \begin{pmatrix} 30 \\ 20 \end{pmatrix}$$

1 0 1 -1/3	30
0 1 -1/3 2/9	20
0 0 2/3 2/9	140

БР = (30, 20, 0, 0) = БДР; $-140 = z_{min}$; точка (30, 20)

Вариант (1,3)

$$B = \begin{pmatrix} 2 & 1 \\ 3 & 0 \end{pmatrix}, \quad B^{-1} = \begin{pmatrix} 0 & 1/3 \\ 1 & -2/3 \end{pmatrix}, \quad B^{-1}A = \begin{pmatrix} 1 & 3 & 0 & 1/3 \\ 0 & -3 & 1 & -2/3 \end{pmatrix}, \quad b' = B^{-1}b = \begin{pmatrix} 90 \\ -60 \end{pmatrix}$$

1 3 0 1/3	90
0 -3 1 -2/3	-60
0 2 0 2/3	180

БР = (90, 0, -60, 0) ≠ БДР; $-180 = z < z_{min}$; точка (90, 0)

Вариант (1,4)

$$B = \begin{pmatrix} 2 & 0 \\ 3 & 1 \end{pmatrix}, \quad B^{-1} = \begin{pmatrix} 1/2 & 0 \\ -3/2 & 1 \end{pmatrix}, \quad B^{-1}A = \begin{pmatrix} 1 & 3/2 & 1/2 & 0 \\ 0 & 9/2 & -3/2 & 1 \end{pmatrix}, \quad b' = B^{-1}b = \begin{pmatrix} 60 \\ 90 \end{pmatrix}$$

1 3/2 1/2 0	60
0 9/2 -3/2 1	90
0 -1 1 0	120

БР = (60, 0, 0, 90) = БДР; $-120 = z > z_{min}$; точка (60, 0)

Вариант (2,3)

$$B = \begin{pmatrix} 3 & 1 \\ 9 & 0 \end{pmatrix}, \quad B^{-1} = \begin{pmatrix} 0 & 1/9 \\ 1 & -1/3 \end{pmatrix}, \quad B^{-1}A = \begin{pmatrix} 1/3 & 1 & 0 & 1/9 \\ 1 & 0 & 1 & -1/3 \end{pmatrix}, \quad b' = B^{-1}b = \begin{pmatrix} 30 \\ 30 \end{pmatrix}$$

1/3 1 0 1/9	30
1 0 1 -1/3	30
-2/3 0 0 4/9	120

БР = (0, 30, 30, 0) = БДР; $-120 = z > z_{min}$; точка (0, 30)

Вариант (2,4)

$$B = \begin{pmatrix} 3 & 0 \\ 9 & 1 \end{pmatrix}, \quad B^{-1} = \begin{pmatrix} 1/3 & 0 \\ -3 & 1 \end{pmatrix}, \quad B^{-1}A = \begin{pmatrix} 2/3 & 1 & 1/3 & 0 \\ -3 & 0 & -3 & 1 \end{pmatrix}, \quad b' = B^{-1}b = \begin{pmatrix} 40 \\ -90 \end{pmatrix}$$

2/3 1 1/3 0	40
-3 0 -3 1	-90
2/3 0 4/3 0	160

БР = (0, 40, 0, -90) = БДР; $-160 = z < z_{min}$; точка (0, 40)

Чтобы применить двойственный симплекс-метод, нужно стартовать из области $z < z_{min}$, то есть из любого БР ≠ БДР. В данном примере для этого годятся две точки: (90, 0) или (0, 40). Возьмем, например, в качестве стартового вариант (1,3):

Start

Clear

Enter

A

NB

NF

Шаг 1.

1°. k _____.
2°. min _____,
s _____,
l _____.
3°. После нормировки:

A

4°. После вычитаний:

A

NB

NF

Шаг 2.

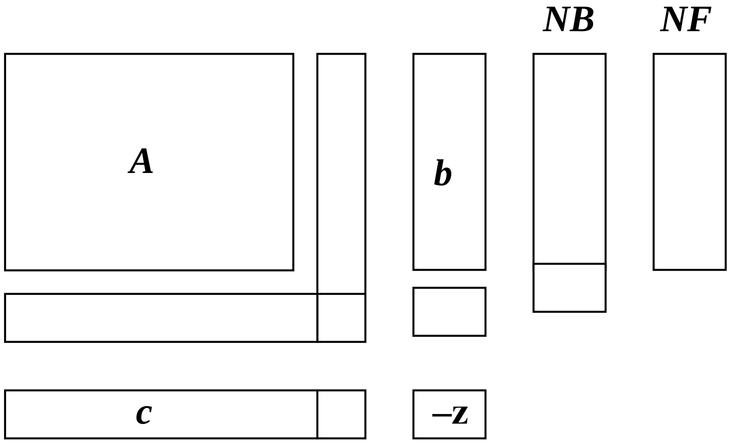
1°. Решение найдено: БР = (_____, _____, _____, _____) = БДР, z_{min} = _____ .

Repeat

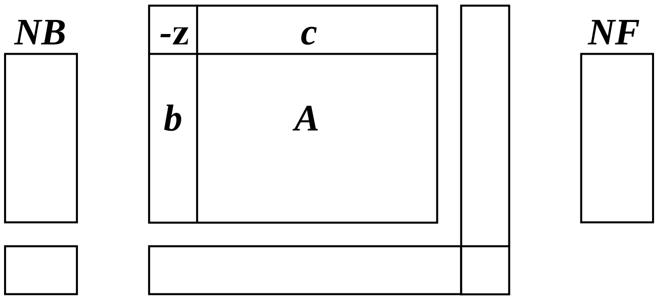
Пример 4.1.2 *Будем считать, что на задачу примера 4.1.1 наложено дополнительное ограничение*

$$5x_1 + 3x_2 \leq 150.$$

Можно решить всю задачу, применяя обычный симплекс-метод, но можно сэкономить время, если продолжить решение примера 4.1.1 с того места, где оно было найдено, добавив еще одно ограничение. Схема добавления может быть такой:



Однако технически это может оказаться неудобным, если (A, b, c, $–z$) хранятся в едином массиве. В этом случае более удобным представляется или раздельное наименование массивов, или другое расположение этих же и добавленных данных в едином массиве следующего вида:



Если язык программирования допускает для массива нумерацию с нуля строк и столбцов, то нулевую строку этого массива удобно отвести для хранения ($–z$, c), а нулевой столбец – для b. С учетом этого перепишем результат примера 4.1.1 и еще снизу и справа к таблице добавим строку и столбец для дополнительного ограничения и дополнительной переменной x_5. Рассмотрим эти действия.

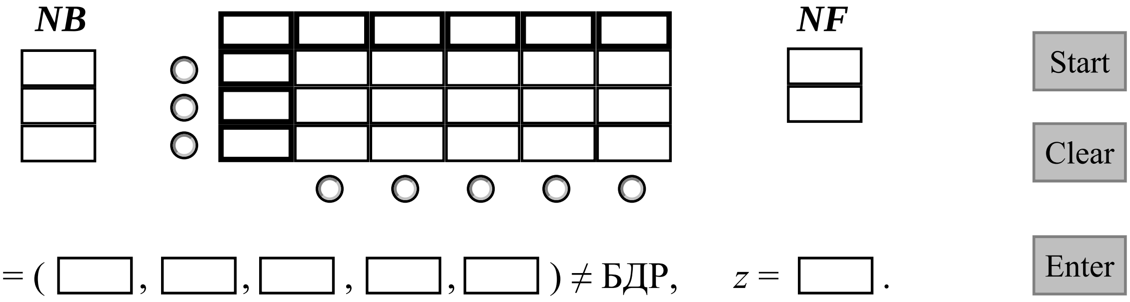
Шаг 0. Ввод дополнительной строки ограничений:

NB
1
2
(5)

140	0	0	2/3	2/9	0
30	1	0	1	-1/3	0
20	0	1	-1/3	2/9	0
150	5	3	0	0	1

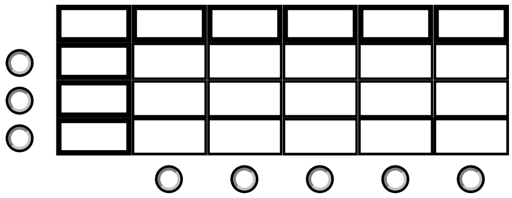
NF
3
4

Здесь эта нижняя строка еще не соответствует каноническому виду для базиса, поэтому в *NB* номер 5 пока проставлен в скобках. Чтобы этот вид «узаконить», нужно провести исключение прежних базисных переменных x_1 и x_2 по добавленной строке. Здесь для этого делают два предварительных вычитания из введенной строки: (1) вычитают первую строку, умноженную на 5, и (2) вычитают вторую строку, умноженную на 3. В результате получают:

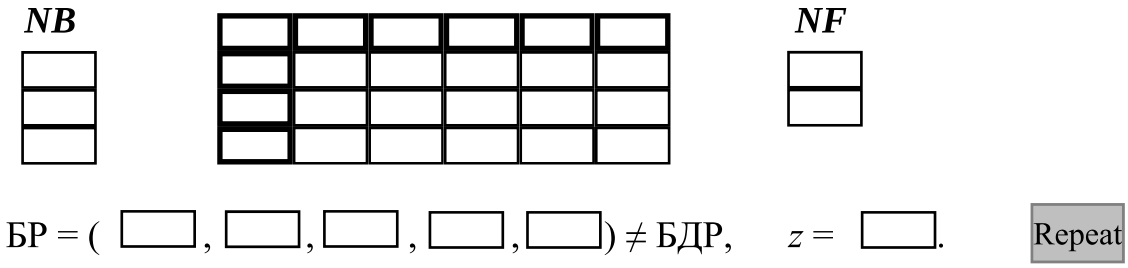


Шаг 1.

1°. *k* .
2°. *min* ,
l ,
s .
3°. После нормировки:



4°. После вычитаний и обмена *NB*(*k*) ↔ *NF*(*l*):



В геометрической интерпретации на плоскости Ox_1x_2 здесь было движение от точки (30, 20) в области $z < z_{min}$ к точке (15, 25). Что эта точка есть решение задачи, распознается в начале шага 2 по отсутствию отрицательных значений среди (15, 25, 15) в нулевом столбце матрицы. Решение найдено за один шаг, поскольку здесь дополнительное ограничение было наложено, когда запись задачи была получена в форме найденного решения примера 4.1.1. Это ограничение можно наложить на другую запись задачи, где коэффициенты целевой функции неотрицательны, и тогда решение двойственным симплекс-методом может большее число шагов.

Чтобы показать это, наложим дополнительные ограничения на вариант (2, 4) примера 4.1.1.

Шаг 0. Ввод дополнительного ограничения:

NB							NF
	160	2/3	0	4/3	0	0	
2	40	2/3	1	1/3	0	0	1
4	-90	-3	0	-3	1	0	3
(5)	150	5	3	0	0	1	

Обеспечивая готовность к двойственному симплекс-методу, исключаем из введенного ограничения переменные x_2 и x_4, являющиеся базисными перед началом шага 0. Для этого вычитаем из введенной строки k-ю строку, умноженную на подходящий коэффициент; сначала k = 1, то есть исключаем базисную переменную x_2, так как $NB(k) = NB(1) = 2$, и подходящий коэффициент, определяемый этим номеров в добавленной строке есть 3. Получаем после вычитания:

Start

Clear

NB							NF

Затем k = 2, и нужно исключить базисную переменную x_4, так как $NB(k) = NB(2) = 4$. Однако из-за того, что коэффициент в последней строке, определяемый этим номером, равен 0, вычитание не требуется. (Проверки на нуль такого рода желательны в программе, чтобы устранить ненужные затраты времени). Готовность к двойственному симплекс-методу обеспечена:

БР = (▭ , ▭ , ▭ , ▭ , ▭) ≠ БДР, ▭ $= z < z_{min}$.

Enter

Шаг 1.

1°. k ▭.

2°. l ▭,

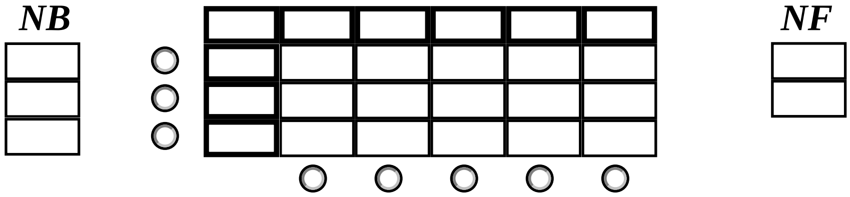
$j = NF(l)$ ▭,

min ▭,

s ▭.

3°. После нормировки:

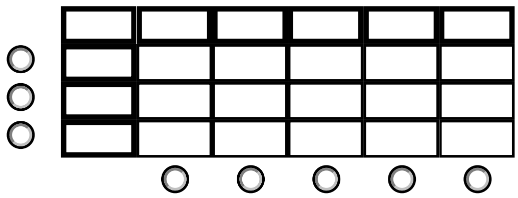
4°. После вычитаний и обмена $NB(k) \leftrightarrow NF(l)$:



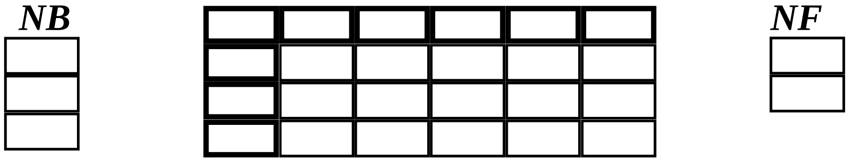
БР = (, , , ,) ≠ БДР, $= z < z_{min}$.

Шаг 2.

1°. k .
2°. l ,
$j = NF(l)$,
s .
3°. После нормировки:



4°. После вычитаний и обмена $NB(k) \leftrightarrow NF(l)$:



БР = (, , , ,) ≠ БДР, $= z_{min}$.

Шаг 3.
1°. Решение найдено:
БОР =(, , , ,), $z_{min} =$.

Repeat

В геометрической интерпретации здесь было движение за два шага по точкам: (0, 40) → (30, 20) → (15, 25).

Пример 4.1.3 *Для* $x_1, x_2 \geq 0$ *решить задачу:*

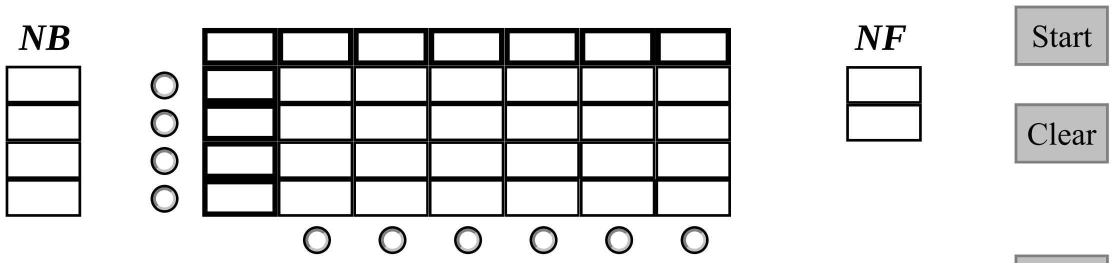
$$2x_1 \geq 3;$$
$$x_2 \geq 1;$$

$$
\begin{aligned}
-\ 2x_1 + 3x_2 &\le\ 3;\\
2x_1 + 3x_2 &\le 21;\\
-2x_1 - 5x_2 &=\ z \to min
\end{aligned}
$$

и затем наложить дополнительное ограничение

$$x_1 + \ x_2 \le\ 6;$$

Шаг 0.

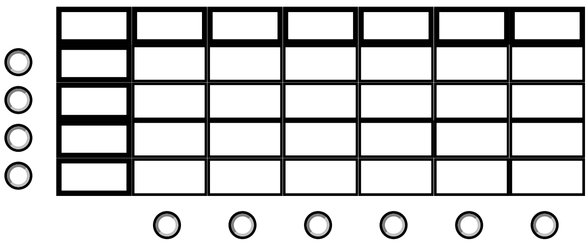


БР = (, , , , ,) ≠ БДР, z = .

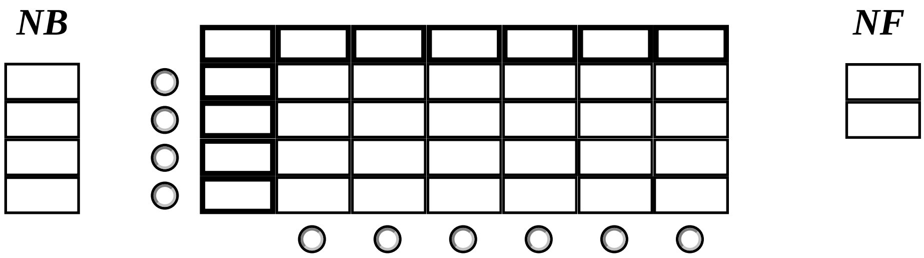
Сначала решаем задачу обычным симплекс-методом, как в п. 3.5.

Шаг 1.

1°. s ,
l .
2°. min ,
k .
3°. После нормировки:



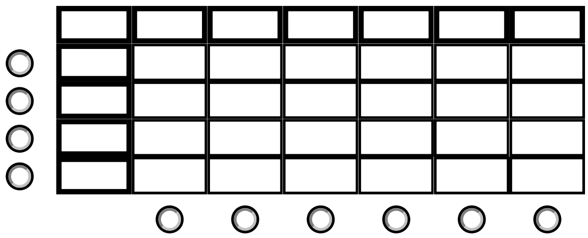
4°. После вычитаний и обмена $NB(k) \leftrightarrow NF(l)$:



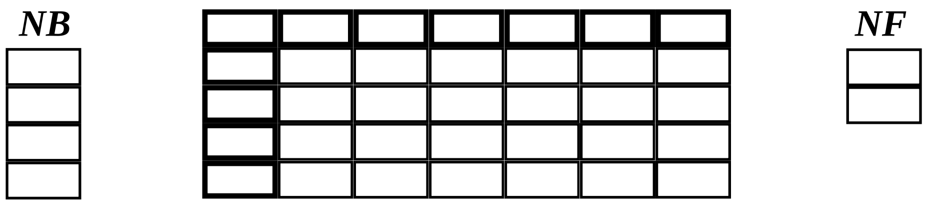
БР = (, , , , ,) ≠ БДР, z = .

Шаг 2.

1°. s $\boxed{}$,
l $\boxed{}$.
2°. min $\boxed{}$,
k $\boxed{}$.
3°. После нормировки:



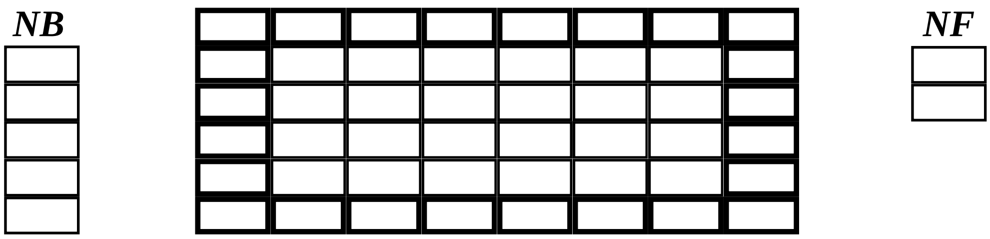
4°. После вычитаний и обмена $NB(k) \leftrightarrow NF(l)$:



БР = ($\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$) = БДР, z_{min} = $\boxed{}$.

Теперь накладываем дополнительное ограничение и одновременно в полученном только что решении обеспечиваем корректный вид базиса (умножаем на «-1» первые две строки). Нумерацию шагов начинаем заново, переходя к применению двойственного симплекс-метода.

Шаг 0.



Исключаем из добавленной строки ограничений те переменные, которые до этого были в списке базисных, (3, 4, 2, 1), и которые вошли в эту строку не с нулевыми коэффициентами (т.е. 1 и 2). Для этого применяем схему Гаусса исключения по добавленной строке, после чего имеем:

NB ***NF***

БР = (, , , , , ,) ≠ БДР, $= z$.

С этого момента стартует двойственный симплекс-метод.

Шаг 1.

1°. k .

2°. l ,

$j = NF(l)$,

min ,

s .

3°. После нормировки:

4°. После вычитаний и обмена $NB(k) \leftrightarrow NF(l)$:

NB ***NF***

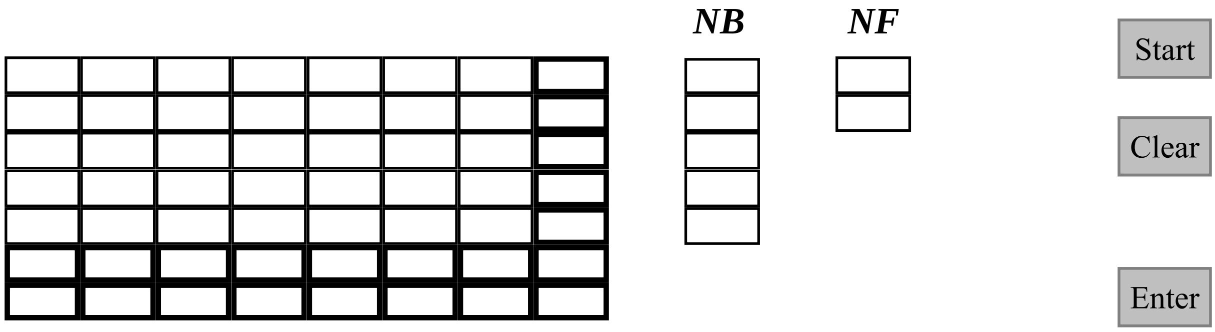
БР = (, , , , , ,) = БДР, $= z_{min}$. Repeat

Покажем особенности, когда дополнительно наложенное ограничение имеет вид «=».

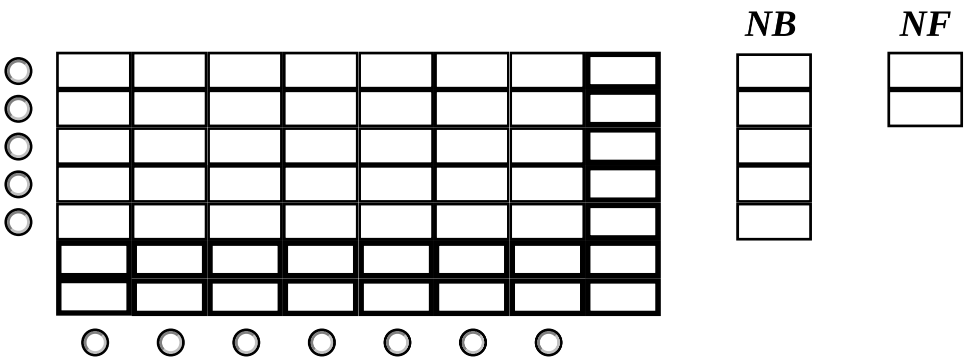
Пример 4.1.4 *Решить задачу предыдущего примера 4.1.3 и затем наложить дополнительное ограничение*

$$x_1 + x_2 = 6.$$

Продолжим решение после шага 2 и для этого ограничения введем искусственную переменную x_7 и искусственную целевую функцию $w = x_7$. С этого момента нумерацию шагов начнем заново. Запись симплекс-таблиц будем вести в форме, принятой здесь везде, кроме примеров 4.1.2 и 4.1.3. Кроме того, обеспечим корректный вид базиса.



Из-за введения третьей снизу строки (пятое ограничение) переменные x_2, x_1 перестали быть базисными. Также не является пока базисной и искусственная переменная x_7 по отношению к искусственной целевой функции (последняя строка). Легко восстановить для них вхождение в число базисных: нужно исключить переменные x_2, x_1 из пятого ограничения и переменную x_7 из последней строки путем очевидных вычитаний строк. Получим:



БР = (, , , , , ,) ≠ БДР, z = , w = .

Этап I. Обычный симплекс-метод для целевой функции w, поскольку в нижней строке имеется отрицательный коэффициент – 1/12.

Шаг 1.

1°. s ,

l $\boxed{\quad\quad}$.

2°. $min\ (\infty,\ \frac{3}{1/6},\ \frac{4}{1/6},\ \infty,\ -\infty)$.

Согласно табл. 3.4.1, «– ∞» означает «запрет». Если же ведущую строку выбрать по этому признаку, в данном примере $k=\boxed{\quad\quad}$, то это будет означать, что свободная переменная x_5 теперь станет базисной, но с отрицательным значением. Однако x_5, становясь отрицательной, увеличит искусственную целевую функцию от ее значения –5/2. Это и нужно, так как Этап I должен завершиться значением w = 0. Поэтому выбор k оправдан и фактически не должен рассматриваться как запретный.

3°. После нормировки:

4°. После вычитаний:

NB

NF

БР = ($\boxed{\quad}$,$\boxed{\quad}$,$\boxed{\quad}$,$\boxed{\quad}$,$\boxed{\quad}$,$\boxed{\quad}$,$\boxed{\quad}$) ≠ БДР, z = $\boxed{\quad}$, w = $\boxed{\quad}$.

Этап I успешно завершен. Нижняя строка, седьмой столбец и $NF(1) = 7$ могут быть вычеркнуты. После этого вид оставшейся нижней строки коэффициентов (0, 0, 0, 0, 0, 3) и вид столбца свободных членов (-9, 8, 9, -3, -30) говорят, что далее следует применить двойственный симплекс-метод по отношению к целевой функции z.

Этап II. Двойственный симплекс-метод с корректным видом базиса (см. п. 4.1).

Шаг 2.

1°. k $\boxed{\quad\quad}$.

2°. s $\boxed{\quad\quad}$.

3°. После нормировки:

4°. После вычитаний:

NB ***NF***

БР = (, , , , ,) = БДР, z = .

Шаг 3.

1°. Полученное выше БДР является ОБДР (оптимальным БДР), поскольку нет отрицательных элементов в правом столбце таблицы, и

z_{min} = .

Repeat

Геометрически, на плоскости Ox_1x_2, здесь после наложения ограничения типа равенства $x_1 + x_2 = 6$ произошло движение из точки (9/2, 4), которая была оптимальной до наложения ограничения, в точку (-3, 9) пересечения линий $2x_1 + 3x_2 = 21$ и $x_1 + x_2 = 6$ за пределы допустимой области (шаг 1). Следующее движение (шаг 2) произошло в точку (3, 3), доставляющую минимум функции z.

Замечание 4.1.1 *Из равенства примеров 4.1.3 и 4.1.4 видно, что ограничение типа «=» требует введения искусственной переменной и искусственной целевой функции.*

4.2 Алгоритм без корректного вида базиса

Выше двойственный симплекс-метод изложен для ситуации, когда все неравенства в ограничениях задачи заранее приведены к виду «меньше или равно». Тогда добавленные переменные входят с коэффициентами «+1», и следовательно, эти, а также и все последующие базисные переменные имеют в качестве векторов ограничений столбцы единичной матрицы. Такой базис (в виде столбцов единичной матрицы) условно называют корректным, чтобы отличить его от других ситуаций, когда одни базисные переменные входят в ограничения с коэффициентом «-1». Очевидно, эти ситуации соответствуют включению ограничений как вида «меньше или равно», так и «больше или равно» в исходную запись задачи.

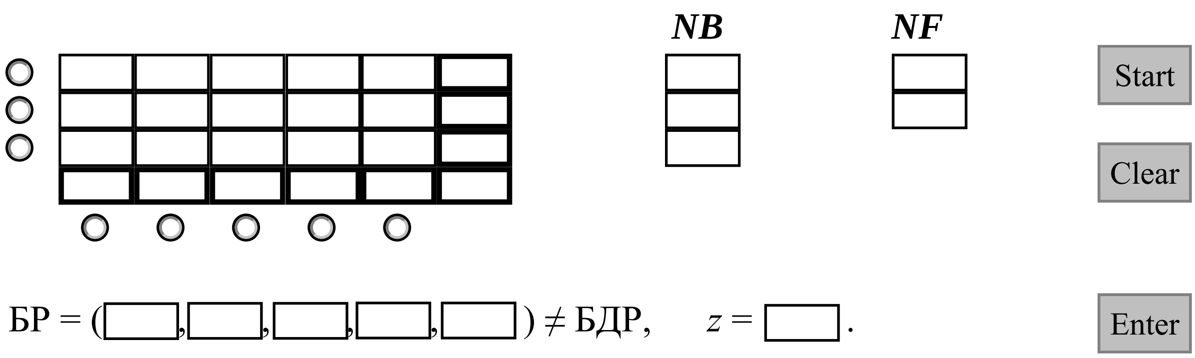
Изложение двойственного симплекс-метода для этого варианта включим в следующий пример, который иллюстрирует также тип условий, при которых решение удобно начинать искать именно двойственным симплекс-методом: начало координат, обычно служащее стартовой точкой, лежит в полупространстве, где $z < z_{min}$.

Замечание 4.2.1 *Пребывание текущего базисного решения в полупространстве $z < z_{min}$ обнаруживают по условию неотрицательности коэффициентов при небазисных (свободных) переменных в строке целевой функции и наличию среди них положительных. Лишь это условие разрешает применять двойственный симплекс-метод.*

Пример 4.2.1 *Для $x_1, x_2 \geq 0$ решить задачу:*

$$
\begin{aligned}
2x_1 + 3x_2 &\geq 6;\\
2x_1 + 6x_2 &\geq 9;\\
2x_1 + 2x_2 &\leq 7;\\
x_1 + 4x_2 &= z \to min.
\end{aligned}
$$

Шаг 0.



Разрешено применять двойственный симплекс-метод.

Шаг 1.

1°. Смотрят через $NB(i)$ на столбец b' и находят те строки, где произведение $a'[i, NB(i)] * b'(i) < 0$. Из них выбирают номер k-ой строки, в которой это отрицательное число наименьшее. Если таких строк не найдено, то задача решена, конец.

k $\boxed{\qquad}$.

2°. Смотрят через $NF(i)$ на те коэффициенты для свободных переменных в k-ой строке, которые положительны, если $a'[k, NB(k)] = -1$, и отрицательны, если $a'[k, NB(k)] = +1$. Среди них выбирают такой, который доставляет

$$\min_j \left(\frac{c'_j}{\left| a'_{kj} \right|} \right).$$

min $\boxed{\qquad}$,
l $\boxed{\qquad}$,
s $\boxed{\qquad}$.

3°. После нормировки k-ой строки делением ее на ведущий элемент a'_{ks}:

4°. После вычитаний k-ой строки, умножаемой на a'_{is} из всех строк ($i \neq k$) и после замены $NB(k) \leftrightarrow NF(l)$, находят:

NB

NF

БР = ($\boxed{\quad}$, $\boxed{\quad}$, $\boxed{\quad}$, $\boxed{\quad}$, $\boxed{\quad}$) ≠ БДР, $z = \boxed{\quad}$.

Шаг 2.

1°. $a'[\boxed{\quad}, NB(\boxed{\quad})] * b'(\boxed{\quad}) < 0 \Rightarrow$ k $\boxed{\qquad}$.
2°. $a'[k, NB(k)] = \boxed{\quad}$ $\Rightarrow$ l $\boxed{\qquad}$,
s $\boxed{\qquad}$.

3°. После нормировки:

4°. После вычитаний и обмена $NB(k) \leftrightarrow NF(l)$:

NB ***NF***

БР = (___, ___, ___, ___, ___) = БДР, ___ $= z = z_{min}$.

Шаг 3.

1°. Все $a'[i, NB(i)] * b'(i)$ >= 0.

Repeat

Замечание 4.2.2 *Искусственные переменные с искусственной целевой функцией, как в п. 3.6, и двойственный симплекс-метод, на первый взгляд, несовместимы. Действительно, если такие переменные ввести для всех ограничений типа «≥» и «=», то все базисные решения перейдут в категорию БДР (базисных допустимых решений). В результате исчезнут условия применимости двойственного симплекс-метода, который работает лишь в области, где БР ≠ БДР (см. подробнее п. 4.3).*

Это замечание иллюстрируем следующей задачей, где в первом примере ограничение типа «=» вводим в решение не с помощью искусственной переменной x_6 с искусственной целевой функцией $w = x_6$, а с заменой «=» на «≤» и одновременно «≥» (практически такой прием может оказаться чувствительным к ошибкам вычислений).

Пример 4.2.2 *Для* $x_1, x_2 \geq 0$ *решить задачу:*

$$
\begin{aligned}
2x_1 + 3x_2 &\geq 6;\\
2x_1 + 6x_2 &\geq 9;\\
x_1 + x_2 &\leq 4;\\
2x_1 + 2x_2 &= 7;\\
x_1 + 2x_2 &= z \to min.
\end{aligned}
$$

Шаг 0.

NB ***NF***

Start

Clear

БР = ($\square$,$\square$,$\square$,$\square$,$\square$,$\square$,$\square$) ≠ БДР, $z =$ $\square$. Enter

Разрешено применять двойственный симплекс-метод.

Шаг 1.

1°. $a'[\square, NB(\square)] * b'(\square) = \square < 0$
$a'[\square, NB(\square)] * b'(\square) = \square < 0$
$a'[\square, NB(\square)] * b'(\square) = \square < 0 \Rightarrow k = \square$.

2°. $a'[k, NB(k)] = \square$. Два положительных коэффициента:
$a'[k, NF(\square)] = \square$,
$a'[k, NF(\square)] = \square$, $\Rightarrow$
min $\square$, $\Rightarrow$ l $\square$,
s $\square$.

3°. После нормировки:

4°. После вычитаний:

NB

NF

БР = ($\square$,$\square$,$\square$,$\square$,$\square$,$\square$,$\square$) ≠ БДР, $z =$ $\square$.

Шаг 2.

1°. $a'[\square, NB(\square)] * b'(\square) = \square < 0$
$a'[\square, NB(\square)] * b'(\square) = \square < 0 \Rightarrow k = \square$.

2°. $a'[\,k, NB(k)] = \square$,
$a'[\,k, NF(\square)] = \square$,
$a'[\,k, NF(\square)] = \square$, $\Rightarrow$
$min\ \square$, $\Rightarrow$ $l\ \square$,
$s\ \square$.

3°. После нормировки:

4°. После вычитаний:

NB ***NF***

БР = ($\square,\square,\square,\square,\square,\square,\square$) = БДР, $\square = z_{min}$.

Шаг 3.

1°. Все $a'[i, NB(i)] * b'(i)$ >= 0.

Repeat

4.3 Алгоритм без корректного вида базиса с искусственными переменными

На примере предыдущей задачи (см. пример 4.2.2) покажем совместимость искусственных переменных и двойственного симплекс-метода.

Пример 4.3.1 *Для* $x_1, x_2 \geq 0$ *решить задачу:*

$$
\begin{aligned}
2x_1 + 3x_2 &\geq 6;\\
2x_1 + 6x_2 &\geq 9;\\
x_1 + x_2 &\leq 4;\\
2x_1 + 2x_2 &= 7;\\
x_1 + 2x_2 &= z \rightarrow min.
\end{aligned}
$$

Здесь в начальных условиях есть ограничение типа «=». Только для него введем искусственную переменную x_6 и функцию $w = x_6$.

Шаг 0.

Start

Clear

Enter

Нижняя строка здесь отведена для коэффициентов и значения функции w. Чтобы в начальное БР, применительно к w, вошли переменные (x_3, x_4, x_5, x_6), сделаем подготовительное вычитание, т.е. исключим x_6 из строки(выражения) для w. В результате получим стартовое состояние:

NB

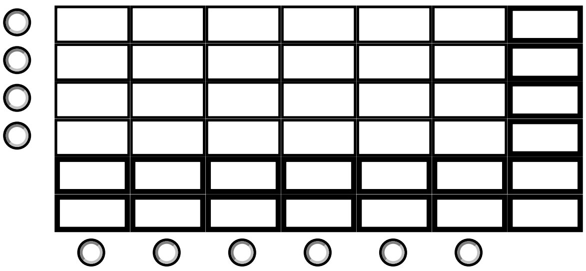
NF

БР = (___, ___, ___, ___, ___, ___) ≠ БДР, $z =$ ___, $w =$ ___ .

По последней строке таблицы распознаем, что для минимизации целевой функции w (этап I) здесь должен быть применен обычный (недвойственный) симплекс-метод без порождения начального БДР (см. п. 3.5).

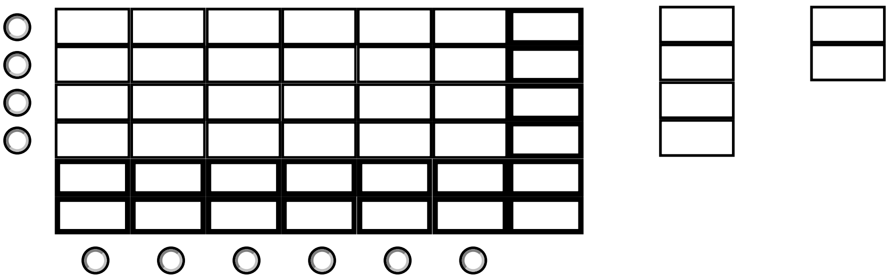
Этап I. Шаг 1.

1°. s ___ ,
l ___ .
2°. min ___ ,
k ___ .
3°. После нормировки:



4°. После вычитаний:

NB ***NF***



БР = (□, □, □, □, □, □) ≠ БДР, $z =$ □, $w =$ □.

Этап I успешно завершен, искусственная целевая функция (последняя строка) и искусственная переменная x_6 далее не нужны. Однако БР осталось недопустимым. Далее, по второй снизу строке распознаем, что должен быть применен двойственный симплекс-метод (см. замечание 4.2.1 в п. 4.2).

Шаг 2.

1°. a'[□, NB(□)] * b'(□) = □ < 0 ⇒ $k =$ □.

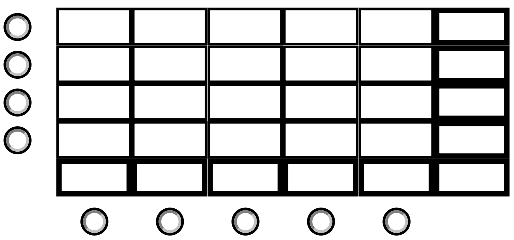
2°. a'[k, NB(k)] = □ ,

a'[k, NF(□)] = □, ⇒

min □, ⇒ l □,

s □.

3°. После нормировки:



4°. После вычитаний:

NB	***NF***
	4

БР = ($\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$) = БДР, $\boxed{}$ = z.

Шаг 3.

1°. Все $a'[i, NB(i)] * b'(i)$ $\boxed{>=}$ 0 $\Rightarrow$ найденное БДР является искомым ОБДР и $z_{min} =$ $\boxed{}$.

Repeat

5
Модифицированный симплекс-метод

Этот вариант называют также улучшенным симплекс-методом, поскольку он уменьшает объем вычислений на каждом шаге до необходимого минимума. Этот минимум почти очевиден: (1°) чтобы определить, какую небазисную переменную ввести в базис, нам нужно проверить только (n-m) коэффициентов для них в строке целевой функции; (2°) определив, какую переменную вводить, скажем, x_s, мы должны проанализировать только m чисел в столбце a'_s и m чисел в столбце b', чтобы решить, какую базисную переменную удалить из базиса. Таким образом, чтобы выбрать ведущий элемент в стандартном симплекс-методе, нам нужно иметь в распоряжении только (n + m) чисел.

Подобным свойством обладает и двойственный симплекс-метод: (1°) чтобы определить, какую базисную переменную удалить из базиса, нам нужно знать только m чисел в столбце b' при выборе номера k одного из этих чисел; (2°) определив, что удалять, скажем, переменную x_r, $r = NB(k)$, мы должны проанализировать только ($n – m$) чисел в k-й строке и ($n – m$) чисел в строке целевой функции, чтобы решить, какую из ($n – m$) небазисных переменных ввести в базис. Следовательно, чтобы выбрать ведущий элемент в двойственном симплекс-методе, нам нужно иметь в распоряжении только $2n – m$ чисел.

Ясно, что наиболее эффективным был бы такой порядок вычислений: сначала вычислить те числа, которые нужны для вышеуказанного действия 1°, и затем – числа для действия 2°, и выполнить эти действия, тем самым осуществляя требуемое «переключение» переменных. Что же касается действий 3° и 4°, составляющий метод Гаусса- Жордана, то их можно и вовсе исключить, если так осуществить обмен переменных, что действия 1° и 2° на следующей шаге могут быть повторены.

Идея заключается в том, что на каждом шаге каноническую форму задачи для текущего базиса можно получать независимо от других таких форм непосредственно из исходной записи стандартной задачи ЛП. Чтобы осуществить эту идею, нужно: (1) сохранять исходную запись задачи на протяжении всей работы метода, - эта та цена, которую приходится платить за большее быстродействие; (2) использовать так называемые симплекс-множители π – коэффициенты для непосредственного перехода от исходной записи задачи к ее текущей канонической форме для базиса; и (3) использовать обращенный базис B^{-1} – матрицу размера $m \times m$, позволяющую вычислять на каждом шаге ведущий столбец a'_s и обновлять симплекс-множители π, т.е. получать все числа, необходимые для действий 1° и 2°.

5.1 Симплекс-множители

Пусть дана исходная запись стандартной задачи ЛП:

$$Ax = b, \qquad x \in \mathbb{R}^n, \qquad b \in \mathbb{R}^m, \qquad m < n,$$
$$rank\, A = m, \qquad x \geq 0, \qquad c^T x = z \;\rightarrow\; min.$$

Текущий шаг симплекс-вычислений определяется каким-нибудь выбором m элементов вектора x для включения их в набор базисных переменных, x^B, при этом остальные $n - m$ элементов оказываются включены в набор небазисных (свободных) переменных, x^F. Допустим, такой выбор сделан, и оказалось $x = (x^B, x^F)$.

Замечание 5.1.1 *Допущение, что именно первые m элементов вектора x вошли в* x^B*, не ограничительно и сделано лишь для удобства изложения и записей.*

Выбор x^B означает, что из n столбцов матрицы ограничений A выбраны m линейно независимых столбцов, как известно, образующих базис в $\mathbb{R}^m$. Эту совокупность столбцов, упорядоченную согласно порядку вхождения элементов вектора x в x^B, называют «**текущим базисом**» и рассматривают в виде матрицы B. Естественно, что и элементы вектора c коэффициентов целевой функции оказываются разделены на две группы: c_B – для x^B и c_F – для x^F.

При сделанном допущении имеем для исходной записи задачи следующее представление:

$$\begin{bmatrix} B & R \\ c_B^T & c_F^T \end{bmatrix} \begin{bmatrix} x^B \\ x^F \end{bmatrix} = \begin{bmatrix} b \\ z \rightarrow \min \end{bmatrix}, \tag{5.1}$$

где через R обозначена матрица столбцов, с которыми входят в систему ограничений

$$Ax = Bx^B + Rx^F = b \tag{5.2}$$

небазисные переменные в порядке их возрастающей нумерации в x^F.

Текущий шаг симплекс-метода требует использования канонической формы задачи для выбранного базиса вместо данного представления исходной записи задачи. Как изложено в п. 3.1, этого достигают в два действия:

1) умножения векторного уравнения ограничений (5.2) слева на обращенный базис B^{-1}, чтобы иметь ограничения в виде уравнения
$$\begin{bmatrix} I & A' \end{bmatrix} \begin{bmatrix} \mathrm{x}^B \\ \mathrm{x}^F \end{bmatrix} = b', \quad A' = B^{-1}R, \quad b' = B^{-1}b,$$
2) умножение этого уравнения на c_B^T слева с получением записи
$$\begin{bmatrix} c_B^T & c_B^T A' \end{bmatrix} \begin{bmatrix} x^B \\ x^F \end{bmatrix} = c_F^T b',$$
которую затем вычитают из уравнения целевой функции, чтобы ее представить в виде
$$\begin{bmatrix} O^T & c'^T \end{bmatrix} \begin{bmatrix} x^B \\ x^F \end{bmatrix} = z - z_0 \rightarrow \min, \quad c'^T = c_F^T - c_B^T A' \tag{5.3}$$

и тем самым выяснить возможность ее дальнейшей минимизации (за счет увеличения какой-либо ее свободной переменной из набора x^F) относительно значения $z = z_0$, $z_0 = c_B^T b'$, уже полученного на данном шаге, то есть на текущем базисном решении БР = [$x^B = b'$, $x^F = 0$].

Однако того же самого можно достичь внешне иными, но вполне идентичными действиями: умножение векторного уравнения ограничений (5.2) слева на матрицу-строку коэффициентов $\pi^T = [\pi_1, \pi_2, \ldots, \pi_m]$ и сложение полученного скалярного произведения с уравнением целевой функции для представления ее в виде

$$\left[\left(\pi^T B + c_B^T\right) \quad \left(\pi^T R + c_F^T\right)\right]\begin{bmatrix} x^B \\ x^F \end{bmatrix} = \pi^T b + z \to \min,$$

где наложено условие $\pi^T B + c_B^T = 0^T$. Выбранные из этого условия множители π, $\pi^T = -\, c_B^T B^{-1}$, называют **симплекс-множителями**, так как они служат той же самой цели – представлению задачи в той же канонической форме базиса, что и выше. Поэтому $c'^T = \pi^T R + c_F^T$, $z_0 = -\,\pi^T b$.

Выражения $\pi^T B + c_B^T$ и $\pi^T R + c_F^T$, очевидно, определяют собой обновленные (пересчитанные) значения коэффициентов при x^B и x^F в целевой функции. Эти выражения совершенно одинаковы по правилу пересчета, а именно: если хотят пересчитать i-ый коэффициент (при переменной x_i), то берут i-й столбец из матрицы A в исходной системе ограничений $Ax = b$, умножают его скалярно на симплекс-множители π и складывают с коэффициентом, который был задан для переменной x_i в исходном выражении $c^T x = z$ для целевой функции. Для этого, естественно, нужны значения симплекс-множителей π.

Их определяют следующим образом. Берут очередной набор m линейно независимых столбцов из исходной матрицы A с номерами $j_1, j_2, \ldots, j_m$, организуя из них матрицу B. (Выбор этих столбцов означает выбор переменных с соответствующими номерами для их включения в базисный набор переменных x^B). Умножают B слева на матрицу-строку π^T, складывают результат с коэффициентами $c_B^T = (c_{j1}, c_{j2}, \ldots, c_{jm})$, которые были заданы для соответствующих переменных в исходном виде $c^T x = z$ целевой функции, и требуют равенства нулю этих сумм. Этому требованию, то есть линейной системе уравнений $\pi^T B = -\, c_B^T$, и подчиняются искомые симплекс-множители π. Каждый раз они единственны, так как $\operatorname{rank} B = m$, и каждый раз они определяются (как и обновленные коэффициенты $c'^T = \pi^T R + c_F^T$ при свободных переменных x^F), отталкиваясь от **исходных** столбцов матрицы ограничений A и **исходных** коэффициентов c_B^T, c_F^T при переменных x^B, x^F в целевой функции $c^T x = z$.

Один случай указанного пересчета коэффициентов отметим особо. Если исходный столбец a_j матрицы A имел вид q-го столбца, e_q, единичной матрицы и соответствующий исходный коэффициент c_j был равен нулю, то: (1) при включении такого столбца a_j в базис B имеем $\pi_q = 0$; (2) при включении такого столбца a_j в небазисную матрицу R имеем $c'_q = \pi_q$. Таким образом, любой элемент памяти компьютера, отведенный для пересчитываемого коэффициента целевой функции при той переменной задачи ЛП, которая входила в ее исходную стандартную формулировку с единичным вектором ограничений вида e_q и с нулевым значением этого коэффициента, в действительности сохраняет не что иное как симплекс-множитель, соответственно, вида π_q, причем $\pi_q = 0$, если соответствующая переменная на данном шаге симплекс-метода оказалась включена в базисный набор x^B.

5.2 Обращенный базис

Как уже отмечалось в пп. 3.3 и 4.1, симплекс-метод представляет собой исключение Гаусса-Жордана (как при решении систем линейных алгебраических уравнений), которому на каждом шаге предшествуют действия 1° и 2° по специальному выбору ведущего элемента. При этом в обычном симплекс-методе (п. 3.3) сначала находят ведущий столбец (действие 1°) и затем ведущую строку (действие 2°), а в двойственном симплекс-методе (п. 4.1) наоборот: сначала находят ведущую строку (действие 1°) и затем ведущий столбец (действие 2°). Для исключения Гаусса-Жордана не имеет значения, в каком порядке действий найдены ведущая строка и ведущий столбец. Существенно то, что ведущий элемент, расположенный в матрице на их пересечении, найден, чтобы выполнить один шаг исключения. С точки зрения решения систем, любой такой шаг означает исключение ведущей переменной (при ведущем столбце) из всех уравнений, кроме ведущего, где она останется с коэффициентом 1. Рассмотрим процедуру **исключения Гаусса-Жордана** с точки зрения обращения базиса в симплекс-методе.

Воспользуемся элементарными матрицами двух видов. Первая – диагональная размера $m \times m$ элементарная матрица D_k, у которой все диагональные элементы равны 1, кроме одного, k-го, который не равен 0. Если надо пронормировать ведущую k-ю строку матрицы A путем деления ее на ведущий элемент a_{ks}, то результат удобно записать в виде $D_{ks}^{-1}A$, где D_{ks} обозначает матрицу D_k, в которой k-й диагональный элемент равен a_{ks} (взят из s-го столбца матрицы A). Тем самым выполнена подготовка к исключению переменной x_s из всех уравнений и одновременно к включению этой переменной в базисный набор x^B взамен переменной x_r, $r = NB(k)$, вводимой из этого набора в набор x^F.

Вторая – полная столбцовая размера $m \times m$ элементарная матрица T_k^c. Ее ненулевые элементы расположены только на диагонали и в k-м столбце, причем все диагональные элементы равны 1. Если для ее k-го столбца, кроме единицы на диагонали, взяты соответствующие элементы s-го столбца матрицы A, то эту матрицу будем обозначать T_{ks}^{c}. Обратная к ней матрица, $(T_{ks}^{c})^{-1}$, отличается инвертированием знаков всех внедиагональных элементов *k*-го столбца. С помощью этой матрицы удобно записать результат второго действия в исключении Гаусса-Жордана, а именно: после вычитания пронормированной *k*-й строки из всех других строк матрицы *A* имеем результат $(T_{ks}^{c})^{-1} D_{ks}^{-1} A$.

При решении систем, когда матрица *A* имеет размер $m \times m$, номера *k* и *s* совпадают и пробегают все значения в естественном порядке: $k = s$ и $s = 1, 2, \ldots, m$. Тогда после *m* шагов исключения Гаусса-Жордана имеем

$$\left(T_{mm}^{c}\right)^{-1} D_{mm}^{-1} \ldots \left(T_{22}^{c}\right)^{-1} D_{22}^{-1} \left(T_{11}^{c}\right)^{-1} D_{11}^{-1} A = I,$$

поэтому

$$A^{-1} = \left(T_{mm}^{c}\right)^{-1} D_{mm}^{-1} \ldots \left(T_{22}^{c}\right)^{-1} D_{22}^{-1} \left(T_{11}^{c}\right)^{-1} D_{11}^{-1} I.$$

Это выражает тот известный факт, что применение стандартной процедуры исключения Гаусса-Жордана к единичной матрице *I* дает обратную матрицу A^{-1}. Рекуррентно данная процедура запишется так:

$$A_t^{-1} = \left(T_{tt}^{c}\right)^{-1} D_{tt}^{-1} A_{t-1}^{-1}; \qquad A_0^{-1} = I, \qquad t = 1, 2, \ldots, m, \tag{5.4}$$

где *t* – номер шага процедуры. По ее окончании имеем: $A_m^{-1} = A^{-1}$.

В симплекс-методе обычным является введение в задачу добавочных и иногда искусственных (см. п. 3.6) переменных, после чего их общее число равно *n*. Будем считать, что они введены в каждое из ограничений по порядку и с коэффициентами 1 (как, например, в пп. 3.3, 3.4, 3.6). В этом случае $m \times n$ матрица ограничений *A* содержит два блока. В исходном виде $A = [\ G \mid I\ ]$, где *G* – левый блок, а *I* – правый блок, образующий в исходном виде единичную $m \times m$ матрицу. Как и в стандартной процедуре Гаусса-Жордана для решения систем, присутствующая в симплекс-методе специальная процедура Гаусса – Жордана в той ее части, которая касается правого блока матрицы *A*, имеет вид:

$$B_t^{-1} = \left(T_{ks}^{c}\right)^{-1} D_{ks}^{-1} B_{t-1}^{-1}, \qquad B_0^{-1} = I, \quad t = 1, 2, \ldots, N, \tag{5.5}$$

где *t* – номер шага симплекс-метода; B_t – базис на шаге *t*; B_t^{-1} – обращенный базис на шаге *t*; B_0^{-1} – обращенный базис на шаге 0 (в исходном состоянии); *N* – полное число шагов этой процедуры. В данном выражении номера *k* и *s* (ведущей строки и ведущего столбца в пределах пересчитываемой на каждом шаге матрицы *A*) определяются по правилам обычного или двойственного симплекс-метода для каждого t, поэтому к ним следовало бы здесь приписать индекс t, но он для простоты опущен.

Таким образом, на месте правого блока матрицы A, имеющего в исходном виде значение I и собственную нумерацию столбцов k = 1, 2,…, m, результате применения указанной процедуры Гаусса-Жордана после t шагов оказывается вычислен **обращенный базис** B_t^{-1}. От обращенного базиса B_{t-1}^{-1} его отличие определяется следующим образом: базис B_t равен базису B_{t-1} после замены в нем k-го столбца, $k \in \{1, 2,\ldots, m\}$, столбцом номер s, $s \in \{1, 2,\ldots, n\}$, взятым из всей матрицы A. Именно такова роль сомножителей $\left(T_{ks}^c\right)^{-1} D_{ks}^{-1}$ в приведенной записи процедуры вычисления обращенного базиса.

Фактически, конечно, не производится никакого умножения матрицы B_{t-1}^{-1} на эти сомножители. Их действие в алгоритме вычисления обращенного базиса сводится, очевидно, к следующему.

Обозначим элементы матрицы обращенного базиса B^{-1} через β_{ij}, i, j = 1, 2,…, m, то есть

$$B^{-1} = \begin{bmatrix} \beta_{11} & \ldots & \beta_{1m} \\ \ldots & \ldots & \ldots \\ \beta_{m1} & \ldots & \beta_{mm} \end{bmatrix}.$$

При t = 0 сюда помещают правый блок исходной матрицы A, равный единичной матрице I. До шага t, t = 1, 2, …, N, здесь находится B_{t-1}^{-1}. После шага t, t = 1, 2, …, N, сюда помещают B_t^{-1}. Пересчет, то есть **обновление обращенного базиса** (от B_{t-1}^{-1} к B_t^{-1}), осуществляют следующим образом, реализующим действие сомножителей $\left(T_{ks}^c\right)^{-1}$ и D_{ks}^{-1} в вычислительной процедуре (5.5).

(1) Действие матрицы D_{ks}^{-1}:

Для j = 1, 2,…, m вычислить $\beta_{kj}^{+} = \beta_{kj} / a_{ks}^{'}$. (5.6)

(2) Действие матрицы $\left(T_{ks}^c\right)^{-1}$:

Для i = 1, 2,…, m, $i \neq k$ и j = 1, 2,…, m
вычислить $\beta_{ij}^{+} = \beta ij - a_{is}^{'} \beta_{kj}^{+}$. (5.7)

Здесь знаком «+» отмечены элементы матрицы B^{-1} после обновления; $a_{ks}^{'}$ – ведущий элемент (k-й элемент s-го столбца всей матрицы A); $a_{is}^{'}$ – любые другие элементы этого ведущего столбца $a_s^{'}$.

Участвующий здесь ведущий столбец определяется, очевидно, равенством

$$a'_s = B^{-1} a_s,$$

где $B^{-1} \ = B_t^{-1}$. Эта формула есть часть более общего выражения матрицы $A' = B^{-1}R$ в описании любого текущего шага симплекс-метода, п. 5.1.

5.3 Обновление симплекс-множителей

В п. 5.1 определено, что симплекс-множители на любом шаге симплекс-метода заданы выражением $\pi^{\mathrm{T}} = -\ c_B^T B^{-1}$, где используются исходные столбцы матрицы ограничений A, образующие на этом шаге базис B, и исходные коэффициенты c_B^T при переменных, образующих на этом шаге набор x^B. На следующем шаге, принадлежность к которому всюду обозначаем индексом «+», соответственно, имеем

$$(\pi^+)^{\mathrm{T}} = -\left(c_B^+\right)^T (B^+)^{-1}.$$

Для j-го элемента вектор-строки $(\pi^+)^{\mathrm{T}}$ это дает следующее выражение, которое преобразуем с учетом ранее установленных выражений для элементов $(B^+)^{-1}$ (п. 5.2):

$$\pi_j^+ \overset{(1)}{=} -\sum_{i=1}^{m} c_{NB^+(i)}\beta_{ij}^+ \overset{(2)}{=} -\sum_{i=1,(i\neq k)}^{m} c_{NB^+(i)}\beta_{ij}^+ - c_{NB^+(k)}\beta_{kj}^+ \overset{(3)}{=} -\sum_{i=1,(i\neq k)}^{m} c_{NB^+(i)}(\beta_{ij}^+ - a_{is}^{'}\beta_{kj}^+) - c_{NB^+(k)}\beta_{kj}^+ =$$

$$\overset{(4)}{=} -\sum_{i=1}^{m} c_{NB(i)}(\beta_{ij} - a_{is}^{'}\beta_{kj}^+) - c_{NB^+(k)}\beta_{kj}^+ \overset{(5)}{=} -\sum_{i=1}^{m} c_{NB(i)}\beta_{ij} + \sum_{i=1}^{m} c_{NB(i)}a_{is}^{'}\beta_{kj}^+ - c_{NB^+(k)}\beta_{kj}^+ =$$

$$\overset{(6)}{=} \pi_j - \beta_{kj}^+(c_s - \sum_{i=1}^{m} c_{NB(i)}a_{is}^{'}) \overset{(7)}{=} \pi_i - \beta_{kj}^+ c_s^{'}.$$

В этих преобразованиях первое, второе, третье и пятое равенства очевидны. Четвертое равенство освобождает в сумме от ограничения $i \neq j$, так как предыдущая разность в квадратных скобках при $i = k$ формально обращается в нуль. Кроме того, благодаря этому же обстоятельству, $c_{NB+(i)}$ заменено на $c_{NB(i)}$, так как именно k-й элемент вектора c^B (и только он один) обновляется при переходе к вектору c_B^T. Тое равенство очевидно. Шестое равенство получено в силу того, что $NB^+(k) = s$, s – номер вводимой переменной, и

$$c_s^{'} = c_s - \sum_{i=1}^{m} c_{NB(i)}a_{is}^{'} .$$

Это, по существу, s-й элемент формулы $c'^T = c_F^T - c_B^T A'$, установленной в п. 5.1 для пересчета тех коэффициентов, которые до следующего шага (отмеченного здесь индексом «+») были в наборе c_F. Действительно, в вектор-строке $\left(c_B^+\right)^T$ на k-м месте в качестве c_k^+ теперь расположен исходный коэффициент c_s,

$s = NB^{+}(k)$, при той переменной x_s, которая на шаге «+» вошла в набор x^B и которая до этого шага значилась в наборе x^F и была выбрана оттуда в качестве ведущей для включения в x^B взамен переменной x_r, $r = NB(k)$, состоявшей в x^B до этого шага «+». Следовательно,

$$\pi_j^{+} = \pi_j - \beta_{kj}^{+} c_s^{'}$$

есть последовательное **правило обновления симплекс-множителей**, $j = 1, 2, \ldots, m$.

5.4 Алгоритм модифицированного симплекс-метода

Суммируем материал пп. 5.1 – 5.3 применительно к обычному (недвойственному) симплекс-методу. При этом исходный вид задачи

$$\begin{bmatrix} G & I \\ c^T & O^T \end{bmatrix} x = \begin{bmatrix} b \\ z \end{bmatrix}, \quad b \geq 0,$$

предполагает, что I – единичная матрица размера $m \times m$ для базисных переменных x^B, не входящих в целевую функцию $c^T x^F = z \rightarrow min$, – обычно это добавленные переменные; G – обозначение остальных столбцов расширенной матрицы $A = [\ G \mid I\ ]$ ограничений $Ax = b$, – это столбцы для небазисных переменных x^F в качестве вектора $x = (x^F, x^B)$ переменных задачи, b - правая часть ограничений, для которой условие $b \geq 0$ здесь и далее (в п. 5.6) не является принципиальным ограничением, а всего лишь обозначением ситуации, когда начальное БР является БДР (см. ниже контр-ситуацию в примере 5.7.1). Из исходного вида задачи для дальнейшего сохраняют следующие исходные данные: A, b, c.

До начала каждого шага t ($t = 1, 2, \ldots, N$) алгоритма известны текущие значения π, B^{-1}, b', z, подлежащие последовательному обновлению. До начала первого шага они таковы: $\pi = 0$, $B^{-1} = I$, $b' = b$, $z = 0$. Из исходных данных используют: G и c – в расчете текущих коэффициентов $c'^T = \pi^T G + c^T$; A – в расчете текущего ведущего столбца $a'_s = B^{-1} a_s$, так как a_s – s-й столбец из A.

Шаг t. *Стандартный модифицированный симплекс-метод*

1°. Находят $c'^T = \pi^T G + c^T$. Смотрят через $NF(i)$ на составную строку (c'^T, π^T) (как на одно целое, соответствующее нижней строке исходного вида задачи), чтобы определить, есть ли в ней отрицательные элементы. Если нет, то решение найдено, конец. Если есть, то фиксируют номер l и, соответственно, номер $s = NF(l)$ той свободной переменной x_s, для которой найден наиболее отрицательный элемент c'_s и которая будет введена в базис.

2°. Находят $a'_s = B^{-1}a_s$. Сравнивают все соответствующие элементы столбцов b' и a'_s, чтобы найти номер k такой, что

$$\frac{b'}{a'_{ks}} = \min_{i, a'_{is}>0}\left(\frac{b'_i}{a'_{is}}\right).$$

Тем самым найден номер $r = NB(k)$ той переменной x_r, которая будет выведена из базиса. Кроме того, в составе ведущего столбца a'_s найден ведущий элемент a'_{ks}.

3°. Обновляют все величины к следующему шагу:

(a) Указатели: $NB(k) \leftrightarrow NF(l)$.

(b) Элементы столбца базисных значений:

$$b_k^+ = b'_k \,/\, a'_{ks}\ ; \qquad b_i^+ = b'_i - a'_{is}b_k^+ \quad (i \neq k).$$

(c) Целевую функцию: $z^+ = z - c'_s b_k^+$.

Примечание: *На самом деле здесь вычисляется целевая функция с противоположным знаком, т.е. –z.*

(d) Обращенный базис:

$$\beta_{kj}^+ = \beta_{kj} \,/\, a'_{ks}\ ; \qquad \beta_{ij}^+ = \beta_{ij} - a'_{is}\beta_{kj}^+ \quad (i \neq k).$$

(e) Симплекс-множители: $\pi^{+T} = \pi^T - \beta_k^+ c'_s$.

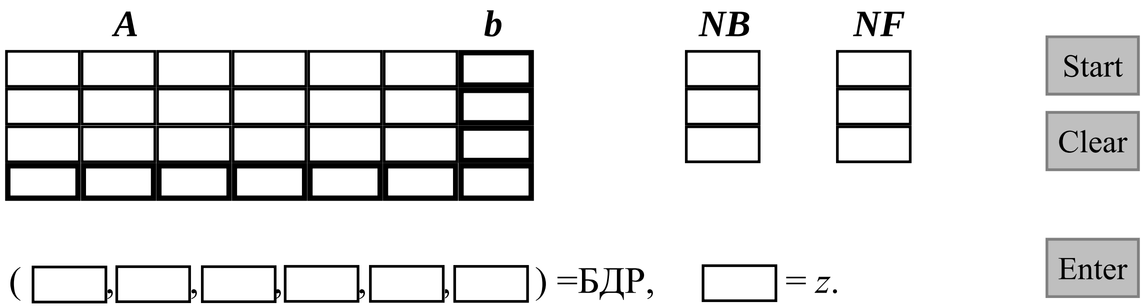
Замечание 5.4.1

1. *В последнем выражении β_k^+ обозначает k-ю строку обновленного обращенного базиса, составленную из элементов β_{kj}^+, j =1, 2,…, m.*
2. *Обновленные значения обозначены в* пп. *(b), (c), (d) и (e) индексом «+» только лишь для удобства. В программах им соответствуют операторы присваивания, например, вместо* п. *(c) имеем $z := z - c'_s b_k$.*

Пример 5.4.1 *Для $x_1, x_2, x_3 \geq 0$ решить задачу*

$$\begin{aligned} x_1 + 2x_2 + 5x_3 &\leq 45; \\ 2x_1 + 3x_2 + 3x_3 &\leq 60; \\ x_1 + x_2 + 2x_3 &\leq 27; \\ -9x_1 - 10x_2 - 15x_3 &= z \to min. \end{aligned}$$

Шаг 0. Исходный вид задачи.



Шаг 1.

1°. $c'^T = \pi^{\mathrm{T}} G + c^T$; $c'^T = c'^T = c^T = (\boxed{}, \boxed{}, \boxed{})$.

Так как в начале алгоритма $\pi^{\mathrm{T}} = (\boxed{}, \boxed{}, \boxed{})$, то на шаге 1 это действие можно опустить, положив $c'^T = c^T$.

$(c'^T, \pi^{\mathrm{T}}) = (\boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{}) \Rightarrow$

$l \ = \boxed{}$,

$s \ = \boxed{}$,

$c'_s = \boxed{}$.

2°. $a'_s = B^{-1} a_s$; $\quad a'_s = a_s = a_3 = \begin{array}{ccc} 1 & 2 & 3 \\ \boxed{} & \boxed{} & \boxed{} \end{array}$.

Так как в начале алгоритма $B^{-1} = I$, то на шаге 1 это действие можно опустить, положив $a'_s = a_s$.

$min = \boxed{} \qquad \Rightarrow$

$k \ = \boxed{}$,

$a'_{ks} = \boxed{}$.

3°. Обновляем все величины к следующему шагу:

(a) $NB = \begin{array}{ccc} 1 & 2 & 3 \\ \boxed{} & \boxed{} & \boxed{} \end{array}$, $\qquad NF = \begin{array}{ccc} 1 & 2 & 3 \\ \boxed{} & \boxed{} & \boxed{} \end{array}$.

(b) $b' = \begin{array}{ccc} 1 & 2 & 3 \\ \boxed{} & \boxed{} & \boxed{} \end{array}$.

(c) $-z = \boxed{}$.

(d) $B^{-1} = \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array}$.

(e) $\pi^T = \boxed{}\boxed{}\boxed{} = \boxed{}\boxed{}\boxed{} - \boxed{}\boxed{}\boxed{} * \boxed{}$.

Шаг 2.

1°. $\begin{array}{ccc} & c'^T & \\ \boxed{} & \boxed{} & \boxed{} \\ 1 & 2 & 3 \end{array} = \begin{array}{ccc} & \pi^{\mathrm{T}} & \\ \boxed{} & \boxed{} & \boxed{} \\ 4 & 5 & 6 \end{array} * \overset{G}{\begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array}} + \begin{array}{ccc} & c^T & \\ \boxed{} & \boxed{} & \boxed{} \end{array}$.

$l \ = \boxed{}$,

$s \ = \boxed{}$,

$c'_s = \boxed{}$.

2°. $a'_s = \begin{bmatrix} \square \\ \square \\ \square \end{bmatrix} = \begin{bmatrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{bmatrix} * \begin{bmatrix} \square \\ \square \\ \square \end{bmatrix}$.

$min = \square \quad \Rightarrow$
$k = \square$,
$a'_{ks} = \square$.

3°. (a) $NB = \begin{array}{ccc} 1 & 2 & 3 \\ \square & \square & \square \end{array}$, $NF = \begin{array}{ccc} 1 & 2 & 3 \\ \square & \square & \square \end{array}$.

(b) $b' = \begin{array}{ccc} 1 & 2 & 3 \\ \square & \square & \square \end{array}$.

(c) $-z = \square$.

(d) $B^{-1} = \begin{bmatrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{bmatrix}$.

(e) $\pi^T = \square\ \square\ \square = \square\ \square\ \square - \square\ \square\ \square * \square$.

Шаг 3.

1°. $\underset{1\quad 2\quad 3}{\overset{c'^T}{\square\ \square\ \square}} = \underset{4\quad 5\quad 6}{\overset{\pi^{\mathrm{T}}}{\square\ \square\ \square}} * \overset{G}{\begin{bmatrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{bmatrix}} + \overset{c^T}{\square\ \square\ \square}$.

$l = \square$,
$s = \square$,
$c'_s = \square$.

2°. $a'_s = \begin{bmatrix} \square \\ \square \\ \square \end{bmatrix} = \begin{bmatrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{bmatrix} * \begin{bmatrix} \square \\ \square \\ \square \end{bmatrix}$.

$min = \square \quad \Rightarrow$
$k = \square$,
$a'_{ks} = \square$.

3°. (a) $NB = \begin{array}{ccc} 1 & 2 & 3 \\ \square & \square & \square \end{array}$, $NF = \begin{array}{ccc} 1 & 2 & 3 \\ \square & \square & \square \end{array}$.

(b) $b' = \begin{array}{ccc} 1 & 2 & 3 \\ \boxed{} & \boxed{} & \boxed{} \end{array}$.

(c) $-z = \boxed{}$.

(d) $B^{-1} = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$.

(e) $\pi^T = \boxed{}\boxed{}\boxed{} = \boxed{}\boxed{}\boxed{} - \boxed{}\boxed{}\boxed{} * \boxed{}$.

Шаг 4.

1°. $\underset{1 \quad 2 \quad 3}{\overset{c'^T}{\boxed{}\boxed{}\boxed{}}} = \underset{4 \quad 5 \quad 6}{\overset{\pi^T}{\boxed{}\boxed{}\boxed{}}} * \overset{G}{\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}} + \overset{c^T}{\boxed{}\boxed{}\boxed{}}$.

$l = \boxed{}$,
$s = \boxed{}$,
$c'_s = \boxed{}$.

2°. $a'_s = \begin{bmatrix} \\ \\ \end{bmatrix} = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} * \begin{bmatrix} \\ \\ \end{bmatrix}$.

$min = \boxed{} \quad \Rightarrow$
$k = \boxed{}$,
$a'_{ks} = \boxed{}$.

3°. (a) $NB = \begin{array}{ccc} 1 & 2 & 3 \\ \boxed{} & \boxed{} & \boxed{} \end{array}$, $\quad NF = \begin{array}{ccc} 1 & 2 & 3 \\ \boxed{} & \boxed{} & \boxed{} \end{array}$

(b) $b' = \begin{array}{ccc} 1 & 2 & 3 \\ \boxed{} & \boxed{} & \boxed{} \end{array}$.

(c) $-z = \boxed{}$.

(d) $B^{-1} = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$.

(e) $\pi^T = \boxed{}\boxed{}\boxed{} = \boxed{}\boxed{}\boxed{} - \boxed{}\boxed{}\boxed{} * \boxed{}$.

Шаг 5.

$$1^\circ.\ \underset{1\quad 2\quad 3}{\overset{c'^T}{\boxed{}\boxed{}\boxed{}}} = \underset{4\quad 5\quad 6}{\overset{\pi^{\mathrm{T}}}{\boxed{}\boxed{}\boxed{}}} * \overset{G}{\begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array}} + \overset{c^T}{\boxed{}\boxed{}\boxed{}}\ .$$

Среди элементов строки $(c'^T,\ \pi^{\mathrm{T}})$, с номерами из NF, нет отрицательных; конец. Базисное, в данном случае, оптимальное решение прочитывают из b' через NB, при этом свободные переменные с номерами из NF равны нулю:

ОБДР = ($\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$); $z_{min} = \boxed{}$. Repeat

По этому же принципу базисные решения могут быть прочитаны по окончании любого шага алгоритма.

5.5 Модифицированный двойственный симплекс-метод

В п. 5.2 отмечалось, что процедура исключения Гаусса-Жордана, лежащая в основе любого варианта симплекс-метода, действует после нахождения ведущей строки и ведущего столбца вне зависимости от того, как эти элементы найдены. В соответствии с этим сформулируем модифицированный алгоритм применительно к двойственному симплекс-методу, опираясь на основные положения из пп. 5.1 – 5.3. Внешне задача имеет такой же исходный вид, как и в п. 5.4:

$$\begin{bmatrix} G & I \\ c^T & O^T \end{bmatrix} x = \begin{bmatrix} b \\ z \end{bmatrix},\quad b \ge 0,$$

Характерное отличие, разрешающее приступать к использованию двойственного симплекс-метода, заключается в том, что $c \ge 0$ и $b \ne 0$, в то время как для применения обыкновенного симплекс-метода (п. 5.4) обычные требования выглядят «наоборот»: $b \ge 0$ и $c \ne 0$.

До начала каждого шага t (t = 1, 2,…, N) известны b', π^T, B^{-1}, c' и z. Требования $\pi^T \ge 0$, $c' \ge 0$ и $b' \ne 0$ являются разрешающими для дальнейшего применения алгоритма. До начала первого шага имеем: $b' = b$, $\pi^T = 0^T$, $B^{-1} = I$, $c' = c$, $z = 0$. Из исходного вида задачи на каждом шаге используют: матрицу G для нахождения левой, соответствующей части k-й (ведущей) строки $a'^T_k = \beta^T_k G$, где β^T_k – k-я строка матрицы B^{-1}, и для определения обновленных коэффициентов $c'^T = \pi^{\mathrm{T}} G + c^T$, где используют их исходные значения c^T; всю матрицу A = [$G \mid I$] для нахождения s-го (ведущего) столбца $a'_s = B^{-1} a_s$, где a_s – исходный s-й столбец в этой матрице A.

Шаг t. *Двойственный модифицированный симплекс-метод.*

1°. Просматривают b', чтобы найти отрицательные элементы. Если таковых нет, то решение найдено; конец. Определяют номер k наиболее отрицательного элемента этого вектора.

2°. Находят $a_k^{'T} = \beta_k^T G$. Полную k-ю строку $(a_k^{'T}, \beta_k^T)$ просматривают как одно целое через $NF(i)$, чтобы из нее выбрать коэффициенты k-го ограничения только при свободных переменных. При этом среди них выбирают только отрицательные коэффициенты $a'_{kj} < 0$, чтобы затем зафиксировать номер l и найти номер $s = NF(l)$ ведущего столбца из условия

$$\min_j \left(\frac{(c'^T, \pi^T)_j}{|a'_{kj}|} \right) = \frac{(c'^T, \pi^T)_s}{|a'_{ks}|},$$

а также элемент $c'_s = (c'^T, \pi^T)_s$.

Примечание: $(c'^T, \pi^T)_j$ *обозначает j-й элемент составной строки* (c'^T, π^T).

3°. Находят ведущий столбец $a'_s = B^{-1} a_s$ и в нем ведущий элемент a'_{ks}.

4°. Обновляют элементы для подготовки к следующему шагу:

(a) Указатели: $NB(k) \leftrightarrow NF(k)$.

(b) Столбец значений базисных переменных:

$$b_k^+ = b'_k / a'_{ks}\ ; \qquad b_i^+ = b'_i - a'_{is} b_k^+ \quad (i \neq k).$$

(c) Целевую функцию: $z^+ = z - c'_s b_k^+$.

Примечание: *нужно помнить, что здесь* z^+ *и* z *обозначают целевую функцию до и после обновления с противоположным знаком.*

(d)Обращенный базис: $\beta_{kj}^+ = \beta_{kj} / a'_{ks}$; $\beta_{ij}^+ = \beta_{ij} - a'_{is} \beta_{kj}^+$ $(i \neq k)$.

(d) Симплекс-множители: $\pi^{+T} = \pi^T - \beta_k^+ c'_s$, $\pi := \pi^+$.

(e) Коэффициенты: $c'^T = \pi^T G + c^T$.

Замечание 5.5.1

1. *Верхний индекс «+» использован в* пп. *(b), (c), (d) и (e) для упрощения; в программе вместо этого применяют операторы присваивания, например,* $b'_k := b'_k / a'_{ks}$, *и т.п.*
2. *В* п. *(c) на самом деле вычисляется целевая функция с противоположным знаком, т.е. при отображении надо поменять знак и выводить* $-z$.

Пример 5.5.1 *Для* $x_1, x_2 \geq 0$ *решить задачу:*

$$
\begin{aligned}
2x_1 + 3x_2 &\geq 6;\\
2x_1 + 6x_2 &\geq 9;\\
2x_1 + 2x_2 &\leq 7;\\
x_1 + 4x_2 &= z \to min.
\end{aligned}
$$

Шаг 0. Исходный вид задачи.

A					***b***

NB

NF

Start

Clear

Enter

БР = ($\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$) ≠ БДР, $\boxed{} = z$.

Шаг 1.

1°. $b'^T = $

1	2	3

$\Rightarrow$

$k = \boxed{}$.

2°. $a'^T_k = \beta^T_k * G$:

a'^T_k	
1	2

=

	β^T_k	
3	4	5

*

G	

.

$min = \boxed{}$ $\Rightarrow$

$l = \boxed{}$,

$s = \boxed{}$,

$c'_s = \boxed{}$.

3°. $a'_s = B^{-1}a_s = a_s = (\boxed{}, \boxed{}, \boxed{})^T$;

$a'_{ks} = \boxed{}$.

4°. Обновление:

(a) $NB = $

1	2	3

, $NF = $

1	2

.

(b) $b'^T = $

1	2	3

.

(c) $-z = \boxed{}$.

(d) $B^{-1} = \begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \quad & \quad & \quad \\ \hline \quad & \quad & \quad \\ \hline \end{array}$.

(e) $\pi^T = \begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \end{array} - \begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \end{array} * \begin{array}{|c|}\hline \quad \\ \hline \end{array}$.

(f) $\underset{1 \qquad 2}{\overset{c'^T}{\begin{array}{|c|c|}\hline \quad & \quad \\ \hline \end{array}}} = \underset{3 \qquad 4 \qquad 5}{\overset{\pi^T}{\begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \end{array}}} * \overset{G}{\begin{array}{|c|c|}\hline \quad & \quad \\ \hline \quad & \quad \\ \hline \quad & \quad \\ \hline \end{array}} + \overset{c^T}{\begin{array}{|c|c|}\hline \quad & \quad \\ \hline \end{array}}$.

Шаг 2.

1°. $b'^T = \overset{1 \qquad 2 \qquad 3}{\begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \end{array}}$,

$k = \begin{array}{|c|}\hline \quad \\ \hline \end{array}$.

2°. $\underset{1 \qquad 2}{\overset{a'^T_k}{\begin{array}{|c|c|}\hline \quad & \quad \\ \hline \end{array}}} = \underset{3 \qquad 4 \qquad 5}{\overset{\beta^T_k}{\begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \end{array}}} * \overset{G}{\begin{array}{|c|c|}\hline \quad & \quad \\ \hline \quad & \quad \\ \hline \quad & \quad \\ \hline \end{array}}$.

$min = \begin{array}{|c|}\hline \quad \\ \hline \end{array} \qquad \Rightarrow$

$l = \begin{array}{|c|}\hline \quad \\ \hline \end{array}$,

$s = \begin{array}{|c|}\hline \quad \\ \hline \end{array}$,

$c'_s = \begin{array}{|c|}\hline \quad \\ \hline \end{array}$.

3°. $a'_s = \begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \quad & \quad & \quad \\ \hline \quad & \quad & \quad \\ \hline \end{array} * \begin{array}{|c|}\hline \quad \\ \hline \quad \\ \hline \quad \\ \hline \end{array} = \begin{array}{|c|}\hline \quad \\ \hline \quad \\ \hline \quad \\ \hline \end{array}$;

$a'_{ks} = \begin{array}{|c|}\hline \quad \\ \hline \end{array}$.

4°. Обновление:

(a) $NB = \overset{1 \qquad 2 \qquad 3}{\begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \end{array}}$, $\qquad NF = \overset{1 \qquad 2}{\begin{array}{|c|c|}\hline \quad & \quad \\ \hline \end{array}}$.

(b) $b'^T = \overset{1 \qquad 2 \qquad 3}{\begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \end{array}}$.

(c) $-z = \begin{array}{|c|}\hline \quad \\ \hline \end{array}$.

(d) $B^{-1} = \begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \quad & \quad & \quad \\ \hline \quad & \quad & \quad \\ \hline \end{array}$.

(e) $\pi^T = \boxed{}\boxed{}\boxed{} = \boxed{}\boxed{}\boxed{} - \boxed{}\boxed{}\boxed{} * \boxed{}$.

$$\text{(f)}\quad \underset{1\ \ 2}{\overset{c'^T}{\boxed{}\boxed{}}} = \underset{3\ \ 4\ \ 5}{\overset{\pi^T}{\boxed{}\boxed{}\boxed{}}} * \overset{G}{\begin{array}{|c|c|}\hline & \\ \hline & \\ \hline & \\ \hline \end{array}} + \overset{c^T}{\boxed{}\boxed{}}.$$

Шаг 3.

$$1°.\quad b'^T = \underset{}{\overset{1\ \ \ 2\ \ \ 3}{\boxed{}\boxed{}\boxed{}}}.$$

ОБДР = ($\boxed{}$,$\boxed{}$,$\boxed{}$,$\boxed{}$,$\boxed{}$); $z_{min} = \boxed{}$. Repeat

Замечание 5.5.2 *Для ускорения выхода на «конец» алгоритма целесообразно после* п. *4° (c) поставить проверку вектора b', аналогичную той, которая предусмотрена в начале шага в* п. *1°.*

Пример 5.5.2 *Для* $x_1, x_2 \geq 0$ *решить задачу:*

$$\begin{aligned} 2x_1 + 3x_2 &\geq 6; \\ 2x_1 + 6x_2 &\geq 9; \\ x_1 + x_2 &\leq 4; \\ 2x_1 + 2x_2 &= 7; \\ x_1 + 2x_2 &= z \rightarrow min. \end{aligned}$$

Шаг 0. Исходный вид задачи.

A							b

NB

NF

Start

Clear

Enter

БР = ($\boxed{}$,$\boxed{}$,$\boxed{}$,$\boxed{}$,$\boxed{}$,$\boxed{}$,$\boxed{}$) ≠ БДР, $\boxed{} = z$.

Шаг 1.

$$1°.\ b'^T = \overset{1\ \ \ 2\ \ \ 3\ \ \ 4\ \ \ 5}{\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}} \quad \Rightarrow$$

$k = \boxed{}$.

2°. $\underset{1\quad 2}{\overset{a_k^{\prime T}}{\boxed{}\boxed{}}} = \underset{3\quad 4\quad 5\quad 6\quad 7}{\overset{\beta_k^T}{\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}}} * \overset{G}{\begin{array}{|c|c|}\hline & \\ \hline & \\ \hline & \\ \hline & \\ \hline & \\ \hline \end{array}}$.

$min = \boxed{} \quad \Rightarrow$

$l = \boxed{}$,

$s = \boxed{}$,

$c'_s = \boxed{}$.

3°. $a'_s = B^{-1}a_s = a_s = (\boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{})^{\mathrm{T}}$;

$a'_{ks} = \boxed{}$.

4°. Обновление:

(a) $NB = \overset{1\quad 2\quad 3\quad 4\quad 5}{\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}}$, $\quad NF = \overset{1\quad 2}{\boxed{}\boxed{}}$.

(b) $b'^T = \overset{1\quad 2\quad 3\quad 4\quad 5}{\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}}$.

(c) $-z = \boxed{}$.

(d) $B^{-1} = \begin{array}{|c|c|c|c|c|}\hline & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline \end{array}$

(e) $\pi^T = \boxed{}\boxed{}\boxed{}\boxed{}\boxed{} - \boxed{}\boxed{}\boxed{}\boxed{}\boxed{} * \boxed{} =$

$= \boxed{}\boxed{}\boxed{}\boxed{}\boxed{}$.

(f) $\underset{1\quad 2}{\overset{c'^T}{\boxed{}\boxed{}}} = \underset{3\quad 4\quad 5\quad 6\quad 7}{\overset{\pi^T}{\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}}} * \overset{G}{\begin{array}{|c|c|}\hline & \\ \hline & \\ \hline & \\ \hline & \\ \hline & \\ \hline \end{array}} + \overset{c^T}{\boxed{}\boxed{}}$.

Шаг 2.

1°. $b'^T = \overset{1\quad 2\quad 3\quad 4\quad 5}{\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}}$,

$k = \boxed{}$.

$$2^\circ.\quad \underset{1\quad 2}{\overset{a_k'^T}{\boxed{}\boxed{}}} = \underset{3\quad 4\quad 5\quad 6\quad 7}{\overset{\beta_k^T}{\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}}} * \overset{G}{\begin{array}{|c|c|}\hline & \\ \hline & \\ \hline & \\ \hline & \\ \hline & \\ \hline \end{array}}.$$

$min = \boxed{} \quad \Rightarrow$

$l = \boxed{}$,

$s = \boxed{}$,

$c'_s = \boxed{}$.

3°. $a'_s = (\boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{})^T$;

$a'_{ks} = \boxed{}$.

4°. Обновление:

(a) $NB = \underset{}{\overset{1\quad 2\quad 3\quad 4\quad 5}{\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}}}$, $\quad NF = \overset{1\quad 2}{\boxed{}\boxed{}}$.

(b) $b'^T = \overset{1\quad 2\quad 3\quad 4\quad 5}{\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}}$.

(c) $-z = \boxed{}$.

(d), (e), (f) – опущены, см. замечание после предыдущего примера, ⇒ конец.

ОБДР = ($\boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{}$); $z_{min} = \boxed{}$.

Repeat

Замечание 5.5.3 *Очевидное преимущество модифицированного симплекс-метода заключается в уменьшении объема необходимых вычислений. Другое преимущество – возможность пошагового контроля вычислений на основе проверки соотношения $B^{-1}B = I$. Из* п. *5.2 следует и примерами подтверждается, что вычисляемая матрица обращенного базиса $B^{-1} = [\beta_{ij}]$, является обратной по отношению с матрице $B(i, j) = A\,[\,i, NB(j)]$. Например, после шага 3 примера 5.4.1 имеем проверочное соотношение*

$$\begin{pmatrix} 1/4 & -1/4 & 1/4 \\ 1/4 & 3/4 & -7/4 \\ -3/4 & -1/4 & 9/4 \end{pmatrix} * \begin{pmatrix} 5 & 2 & 1 \\ 3 & 3 & 2 \\ 2 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

в котором второй сомножитель, матрица B, получена из исходной матрицы A при извлечении из нее столбцов в порядке, определяемом указателем NB = (3, 2, 1).

5.6 Модифицированный метод с искусственными переменными

Алгоритм из п. 5.4 предполагает, что исходный вид задачи делает очевидным стартовое БДР. Это то же самое предположение, что было принято в пп. 3.2 и 3.3. Однако часто оно не выполняется, что возникает при ограничениях типа «больше или равно», когда они не заменены (умножением на -1) на «меньше или равно», или при ограничениях типа «равно». Чтобы в этих случаях стартовать все же из БДР, применяют искусственные переменные с искусственной целевой функцией, равной их сумме. В п. 3.6 этот прием изложен для обычного симплекс-метода. Сейчас его покажем для модифицированного симплекс-метода, проводя параллели с п. 5.4.

К началу алгоритма исходный вид задачи подобен следующему:

$$\begin{bmatrix} G & I \\ c^T & 0^T = \pi^T \\ d^T & 0^T = \sigma^T \end{bmatrix} x = \begin{bmatrix} b \\ z \\ w \end{bmatrix}, \qquad b \geq 0.$$

Это означает, что $x = (x^F, x^B)$ и стартовое БР = БДР =(0, b). (Если не выполнено $b \geq 0$, то это лишь означает, что стартовое БР = (0, b) $\neq$ БДР.) Для основной целевой функции z имеем коэффициенты c^T и симплекс-множители π^T (последние сначала равны нулю). Для искусственной целевой функции w имеем коэффициенты d^T и симплекс-множители σ^T (они сначала также нулевые).

Должны быть выполнены два этапа. Этап I – минимизация искусственной целевой функции w для порождения БДР к началу Этапа II. Этап II – минимизация основной целевой функции z; здесь из задачи уже удалены: искусственная целевая функция w и искусственные переменные. На этапе I все действия выполняют так же, как описано в п. 5.4, но одновременно с двумя парами коэффициентов: (c^T, π^T) и (d^T, σ^T). При этом для Этапа I определяющей является пара (d^T, σ^T), поскольку именно она связана с минимизируемой искусственной целевой функцией w. На этапе же II остаются в работе лишь первая пара (c^T, π^T) и минимизируемая целевая функция z.

Нет необходимости приводить описание алгоритма, поскольку оно, с учетом отмеченных особенностей Этапов I и II, полностью соответствует описанию алгоритма п. 5.4. Достаточно привести пример.

Пример 5.6.1 *Для* $x_1, x_2 \geq 0$ *решить задачу:*

$$\begin{aligned} &2x_1 + 2x_2 \leq 7; \\ &2x_1 + 3x_2 \geq 6; \\ &2x_1 + 6x_2 \geq 9; \\ &x_1 + 4x_2 = z \to min. \end{aligned}$$

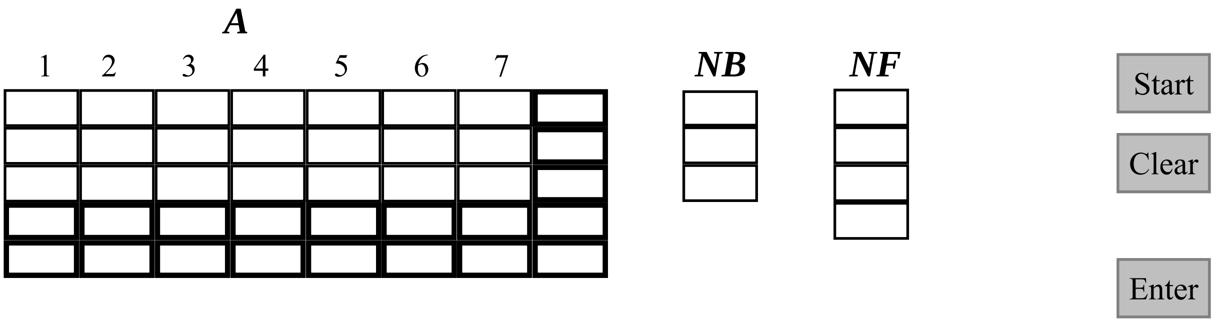
Сравним с примерами из п. 4.2 и п. 5.5, где решается эта же задача, но без искусственных переменных. Там это заставило, соответственно, применить не обычный, а двойственный симплекс-метод (п. 4.2), или же модифицированный двойственный симплекс-метод (п. 5.5).

Решение. Исходная запись задачи такова:

2 2 0 0	1 0 0	7
2 3 -1 0	0 1 0	6
2 6 0 -1	0 0 1	9
1 4 0 0	0 0 0	0
0 0 0 0	0 1 1	0

От нее нужно перейти к корректному исходному виду задачи, чтобы иметь $\sigma^T = 0^T$. Выше на месте σ^T имеем значения $\sigma^T = [0\ \ 1\ \ 1]$, что означает $w = x_6 + x_7$. Для получения $\sigma^T = [0\ 0\ 0]$ выполняют так называемое подготовительное вычитание. Оно, очевидно, означает исключение искусственных переменных x_6 и x_7 из представления искусственной целевой функции w и всегда сводится к вычитанию из первоначальной строки для w (нижняя строка таблицы) суммы тех строк ограничений, в которые искусственные переменные, здесь x_6 и x_7, входят как базисные переменные с коэффициентами 1 (по диагонали блока единичной матрицы I). Здесь вычтем вторую и третью строку из последней строки таблицы.

Шаг 0.



БР = ($\square$, $\square$, $\square$, $\square$, $\square$, $\square$, $\square$) ≠ БДР, $\square$ = z, $\square$ = - w.

Теперь задача приобрела корректный исходный вид, и при этом z и w имеют правильные значения: $z = 0$, $w = 15$.

Этап I. Следуем алгоритму из п. 5.4, используя в качестве определяющей пару (d^T, σ^T) и в качестве «попутной» пару (c^T, π^T).

Шаг 1.

1°. $d'^T = \sigma^T G + d^T$; $\quad c'^T = \pi^T G + c^T$.

Так как в начале алгоритма имеем $\sigma^T = 0$ и $\pi^T = 0$, то $d'^T = d^T$ и $c'^T = c^T$.
Здесь

$d'^T = \Box\,\Box\,\Box\,\Box$,

$c'^T = \Box\,\Box\,\Box\,\Box$,

$(d^T, \sigma^T) = (\Box, \Box, \Box, \Box, \Box, \Box, \Box) \Rightarrow$

$l = \Box$,

$s = \Box$,

$d'_s = \Box$,

$c'_s = \Box$.

2°. $a'_s = B^{-1}a_s = a_s = (\Box, \Box, \Box)^T$;

$b'^T = (\Box, \Box, \Box)^T$,

$min = \Box \qquad \Rightarrow$

$k = \Box$,

$a'_{ks} = \Box$.

3°.

NB	NF	b^+
$\Box$	$\Box$	$\Box$
$\Box$	$\Box$	$\Box$
$\Box$	$\Box$	$\Box$
	$\Box$	

$z^+ = z - c'_s b_k^+ = \Box$,

$w^+ = w - d'_s b_k^+ = \Box$,

$$B^{-1} = \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array}.$$

$$\underbrace{\Box\,\Box\,\Box}_{\pi^{+T}} = \underbrace{\Box\,\Box\,\Box}_{\pi^T} - \underbrace{\Box\,\Box\,\Box}_{\beta_k^T} * \underbrace{\Box}_{c'_s}.$$

$$\underbrace{\Box\,\Box\,\Box}_{\sigma^{+T}} = \underbrace{\Box\,\Box\,\Box}_{\sigma^T} - \underbrace{\Box\,\Box\,\Box}_{\beta_k^T} * \underbrace{\Box}_{d'_s}.$$

БР = $(\Box, \Box, \Box, \Box, \Box, \Box, \Box)$ = БДР, $z = \Box$, $w = \Box$.

Примечание: *Последние два равенства определяют истинные значения z и w. Их надо отличать от z, z^+, w, w^+, используемых в действии 3° на каждом шаге в качестве вычисляемых величин, знак которых всегда противоположен знаку истинных величин z и w.*

Шаг 2.

1°. $$\overset{d'^T}{\underset{1\qquad 2\qquad 3\qquad 4}{\boxed{}\boxed{}\boxed{}\boxed{}}} = \overset{\sigma^T}{\underset{5\qquad 6\qquad 7}{\boxed{}\boxed{}\boxed{}}} * \overset{G}{\begin{array}{|c|c|c|c|}\hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array}} + \overset{d^T}{\boxed{}\boxed{}\boxed{}\boxed{}}.$$

$$\overset{c'^T}{\underset{1\qquad 2\qquad 3\qquad 4}{\boxed{}\boxed{}\boxed{}\boxed{}}} = \overset{\pi^T}{\underset{5\qquad 6\qquad 7}{\boxed{}\boxed{}\boxed{}}} * \overset{G}{\begin{array}{|c|c|c|c|}\hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array}} + \overset{c^T}{\boxed{}\boxed{}\boxed{}\boxed{}}.$$

Просматриваем (d^T, σ^T) через NF, чтобы найти наиболее отрицательный элемент. Находим:

$l = \boxed{}$,
$s = \boxed{}$,
$d'_s = \boxed{}$.

Смотрим через NF на (c^T, π^T) и по s находим:

$c'_s = \boxed{}$.

2°. $$a'_s = B^{-1}a_s = a_s = \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} * \begin{array}{|c|}\hline \\ \hline \\ \hline \\ \hline \end{array} = \begin{array}{|c|}\hline \\ \hline \\ \hline \\ \hline \end{array}; \qquad b' = \begin{array}{|c|}\hline \\ \hline \\ \hline \\ \hline \end{array},$$

$min = \boxed{}$ $\Rightarrow$
$k = \boxed{}$,
$a'_{ks} = \boxed{}$.

3°. $$\overset{NB}{\begin{array}{|c|}\hline \\ \hline \\ \hline \\ \hline \end{array}} \qquad \overset{NF}{\begin{array}{|c|}\hline \\ \hline \\ \hline \\ \hline \\ \hline \end{array}} \qquad \overset{b^+}{\begin{array}{|c|}\hline \\ \hline \\ \hline \\ \hline \end{array}}$$

$z^+ = z - c'_s b_k^+ = \boxed{}$,
$w^+ = w - d'_s b_k^+ = \boxed{}$,

Появление $w := w^+ = 0$ означает предварительный признак успешного окончания Этапа I, при этом уже ясно, что

БР = ($\boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{}$) = БДР, $z = \boxed{}$.

Чтобы проверить окончательный признак и перейти далее к Этапу II, продолжаем вычисления.

$$B^{-1} = \begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \quad & \quad & \quad \\ \hline \quad & \quad & \quad \\ \hline \end{array}.$$

$$\underset{\pi^{+T}}{\begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \end{array}} = \underset{\pi^{T}}{\begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \end{array}} - \underset{\beta_k^T}{\begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \end{array}} * \underset{c'_s}{\boxed{\quad}}.$$

$$\underset{\sigma^{+T}}{\begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \end{array}} = \underset{\sigma^{T}}{\begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \end{array}} - \underset{\beta_k^T}{\begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \end{array}} * \underset{d'_s}{\boxed{\quad}}.$$

Примечание: *Как уже отмечалось, в модифицированных алгоритмах симплекс-метода всегда можно проконтролировать правильность вычислений по соотношению:* $B^{-1}*B = I$.

Шаг 3.

1°. $$\overset{d'^T}{\begin{array}{|c|c|c|c|}\hline \quad & \quad & \quad & \quad \\ \hline \end{array}}_{\;1\;\;2\;\;3\;\;4} = \overset{\sigma^T}{\begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \end{array}}_{\;5\;\;6\;\;7} * \overset{G}{\begin{array}{|c|c|c|c|}\hline \quad & \quad & \quad & \quad \\ \hline \quad & \quad & \quad & \quad \\ \hline \quad & \quad & \quad & \quad \\ \hline \end{array}} + \overset{d^T}{\begin{array}{|c|c|c|c|}\hline \quad & \quad & \quad & \quad \\ \hline \end{array}}.$$

$$\overset{c'^T}{\begin{array}{|c|c|c|c|}\hline \quad & \quad & \quad & \quad \\ \hline \end{array}}_{\;1\;\;2\;\;3\;\;4} = \overset{\pi^T}{\begin{array}{|c|c|c|}\hline \quad & \quad & \quad \\ \hline \end{array}}_{\;5\;\;6\;\;7} * \overset{G}{\begin{array}{|c|c|c|c|}\hline \quad & \quad & \quad & \quad \\ \hline \quad & \quad & \quad & \quad \\ \hline \quad & \quad & \quad & \quad \\ \hline \end{array}} + \overset{c^T}{\begin{array}{|c|c|c|c|}\hline \quad & \quad & \quad & \quad \\ \hline \end{array}}.$$

Смотрим на элементы (d'^T, σ^T) через NF = (6, 7, 3, 4). Среди них нет отрицательных и, кроме того, w = 0. Это окончательный признак завершения Этапа I. Далее (на Этапе II) искусственные переменные x_6, x_7 и искусственная целевая функция w уже не нужны. Также не нужны (d'^T, σ^T), и мы переходим к просмотру (c'^T, π^T) через NF = (6, 7, 3, 4), где номера 6 и 7 игнорируем.

Этап II. Завершаем шаг 3, действие 1°. В результате просмотра имеем:

l = $\boxed{\quad\quad}$,
s = $\boxed{\quad\quad}$,
c'_s = $\boxed{\quad\quad}$.

$$2^\circ.\ a'_s = B^{-1}a_s = a_s = \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} * \begin{array}{|c|}\hline \\ \hline \\ \hline \\ \hline \end{array} = \begin{array}{|c|}\hline \\ \hline \\ \hline \\ \hline \end{array}; \qquad b' = \begin{array}{|c|}\hline \\ \hline \\ \hline \\ \hline \end{array},$$

Примечание: *Нужна вся матрица B^{-1}, чтобы найти a'_s, несмотря на только что отмеченную дальнейшую ненужность искусственных переменных. То же самое относится и к π^T, что будет видно ниже на шаге 4, действие 1°.*

$$\begin{aligned} min &= \boxed{} \qquad \Rightarrow \\ k &= \boxed{}, \\ a'_{ks} &= \boxed{}. \end{aligned}$$

$$3^\circ. \quad NB \ \begin{array}{|c|}\hline \\ \hline \\ \hline \\ \hline \end{array} \quad NF \ \begin{array}{|c|}\hline \\ \hline \\ \hline \\ \hline \\ \hline \end{array} \quad b^+ \ \begin{array}{|c|}\hline \\ \hline \\ \hline \\ \hline \end{array} \qquad z^+ = z - c'_s b_k^+ = \boxed{}$$

$$B^{-1} = \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array}.$$

$$\underset{\pi^{+T}}{\begin{array}{|c|c|c|}\hline & & \\ \hline \end{array}} = \underset{\pi^{T}}{\begin{array}{|c|c|c|}\hline & & \\ \hline \end{array}} - \underset{\beta_k^T}{\begin{array}{|c|c|c|}\hline & & \\ \hline \end{array}} * \underset{c'_s}{\boxed{}}.$$

БР = ($\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$) = ОБДР, $z = \boxed{}$.

Однако факт, что решение задачи найдено и имеет этот вид, будет обнаружен лишь на следующем шаге 4.

Шаг 4.

$$1^\circ.\ \overset{c'^T}{\underset{1 \quad 2 \quad 3 \quad 4}{\begin{array}{|c|c|c|c|}\hline & & & \\ \hline \end{array}}} = \overset{\pi^T}{\underset{5 \quad 6 \quad 7}{\begin{array}{|c|c|c|}\hline & & \\ \hline \end{array}}} * \overset{G}{\begin{array}{|c|c|c|c|}\hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array}} + \overset{c^T}{\begin{array}{|c|c|c|c|}\hline & & & \\ \hline \end{array}}.$$

Через NF = (6, 7, 5, 4), в котором номера 6 и 7 уже игнорируем, смотрим на (c^T, π^T) и обнаруживаем, что среди чисел (3/4, 1/4) нет отрицательных. Это и указывает на конец работы и достижение оптимального базисного допустимого решения (ОБДР).

Repeat

Замечание 5.6.1 *Надо вычислять c'^T полностью, поэтому нужно находить π^T целиком, то есть вместе с элементами, которые имеют номера 6 и 7, хотя это номера игнорируемых искусственных переменных. Иными словами, эти номера игнорируют только при просмотре строки (c'^T, π^T).*

5.7 Модифицированный ДСМ с искусственными переменными

В п. 4.3 изложен двойственный симплекс-метод без корректного вида базиса в том случае, когда среди ограничений есть ограничения типа «=» и только ими обусловлено применение искусственных переменных и искуственной целевой функции на Этапе I в варианте обычного симплекс метода; при этом задача такова, что необходимость применения двойственного симплекс-метода обнаруживается на Этапе II. Данная ситуация рассмотрена в применении к немодифицированному алгоритму. Сейчас на том же примере, что в п. 4.3, покажем применение модифицированного алгоритма. При этом на Этапе I будет работать алгоритм из п. 5.4 (точнее, из п. 5.6), а на Этапе II будет работать алгоритм из п. 5.5.

Пример 5.7.1 *Для* $x_1, x_2 \geq 0$ *решить задачу:*

$$\begin{aligned} 2x_1 + 3x_2 &\geq 6; \\ 2x_1 + 6x_2 &\geq 9; \\ x_1 + x_2 &\leq 4; \\ 2x_1 + 2x_2 &= 7; \\ x_1 + 2x_2 &= z \to min. \end{aligned}$$

Во всех модифицированных алгоритмах базис должен быть корректным по виду, то есть должен быть образован в исходном виде столбцами единичной матрицы. Поэтому неравенства типа «≥» должны быть умножены предварительно на -1 для приведения их к виду «≤». С учетом этого исходный вид задачи характеризуется таблицей следующими данными.

Шаг 0.

Start

Clear

Enter

Переменная x_6, очевидно, искусственная, и нижняя строка отведена для коэффициентов и значения искусственной целевой функции w. Ее исходное значение пока неверное, так как x_6 пока еще не исключена из нижней строки, чтобы ввести x_6 в базис применительно к w. Сделав это подготовительное исключение, получим верное стартовое состояние:

NB

NF

БР = ($\square$, $\square$, $\square$, $\square$, $\square$, $\square$) ≠ БДР, $\square$ = z, $\square$ = w.

Теперь исходный вид задачи подобен указанному в п. 5.6 с тем отличием, что не выполнено $b \geq 0$, т.е. исходное БР ≠ БДР.

Этап I. Минимизация искусственной целевой функции w. По нижней строке, отведенной для w, распознаем, что должен быть применен обычный модифицированный симплекс-метод (п. 5.4 и п. 5.6).

Шаг 1.

1°. $d'^T = \sigma^{\mathrm{T}} G + d^T$; $c'^T = \pi^{\mathrm{T}} G + c^T$.

Так как в начале алгоритма имеем $\sigma^{\mathrm{T}} = 0$ и $\pi^{\mathrm{T}} = 0$, то $d'^T = d^T$ и $c'^T = c^T$.
Здесь
d'^T = $\square\square$,
c'^T = $\square\square$,
(d^T, σ^T) = ($\square$, $\square$, $\square$, $\square$, $\square$, $\square$) $\Rightarrow$
l = $\square$,
s = $\square$,
d'_s = $\square$,
c'_s = $\square$.

2°. $a'_s = B^{-1} a_s = a_s$ = ($\square$, $\square$, $\square$, $\square$)$^{\mathrm{T}}$;
b'^T = ($\square$, $\square$, $\square$, $\square$)$^{\mathrm{T}}$,
min = $\square$ $\Rightarrow$
k = $\square$,
a'_{ks} = $\square$.

3°.

NB	NF	b^+

$$z^+ = z - c'_s b_k^+ = \boxed{},$$
$$w^+ = w - d'_s b_k^+ = \boxed{},$$

$$B^{-1} = \begin{array}{|c|c|c|c|}\hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array}\,.$$

$$\overset{\pi^{+T}}{\boxed{}\boxed{}\boxed{}\boxed{}} = \overset{\pi^{T}}{\boxed{}\boxed{}\boxed{}\boxed{}} - \overset{\beta_k^T}{\boxed{}\boxed{}\boxed{}\boxed{}} * \overset{c'_s}{\boxed{}}.$$

$$\overset{\sigma^{+T}}{\boxed{}\boxed{}\boxed{}\boxed{}} = \overset{\sigma^{T}}{\boxed{}\boxed{}\boxed{}\boxed{}} - \overset{\beta_k^T}{\boxed{}\boxed{}\boxed{}\boxed{}} * \overset{d'_s}{\boxed{}}.$$

БР = ($\boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{}, \boxed{}$) ≠ БДР, $z = \boxed{}$, $w = \boxed{}$.

Шаг 2.

1°. $$\overset{d'^T}{\boxed{}\boxed{}} = \overset{\sigma^{\mathrm{T}}}{\boxed{}\boxed{}\boxed{}\boxed{}} * \overset{G}{\begin{array}{|c|c|}\hline & \\ \hline & \\ \hline & \\ \hline & \\ \hline \end{array}} + \overset{d^T}{\boxed{}\boxed{}}.$$

$$\overset{c'^T}{\boxed{}\boxed{}} = \overset{\pi^{\mathrm{T}}}{\boxed{}\boxed{}\boxed{}\boxed{}} * \overset{G}{\begin{array}{|c|c|}\hline & \\ \hline & \\ \hline & \\ \hline & \\ \hline \end{array}} + \overset{c^T}{\boxed{}\boxed{}}.$$

Смотрим на элементы (d^T, σ^T) через NF. Так как среди них нет отрицательных и w= 0, то Этап I успешно завершен. Далее при просмотрах строки (c^T, π^T) шестой элемент, соответствующий искусственной переменной x_6, игнорируем, то есть просмотр производим только через оставшийся элемент NF, т.е. $l = 2$, $NF(l) = 2$.

Этап II. Продолжаем шаг 2, действие 1°. Просмотр сообщает о переходе к двойственному симплекс-методу (см. замечание 4.2.1), причем здесь применяем модифицированный алгоритм (см. п. 5.5).

1°. $b' = \begin{array}{|c|c|c|c|} \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} \\ \hline & & & \\ \hline \end{array} \Rightarrow$

$k = \boxed{}$.

2°. $$\underset{1 \quad\quad 2}{\overset{a_k^{'\,T}}{\begin{array}{|c|c|} \hline & \\ \hline \end{array}}} = \underset{3 \quad 4 \quad 5 \quad 6}{\overset{\beta_k^T}{\begin{array}{|c|c|c|c|} \hline & & & \\ \hline \end{array}}} * \overset{G}{\begin{array}{|c|c|} \hline & \\ \hline & \\ \hline & \\ \hline & \\ \hline \end{array}}$$

$min = \boxed{} \Rightarrow$

$l = \boxed{}$,

$s = \boxed{}$,

$c'_s = \boxed{}$.

3°. $$a'_s = \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} * \begin{array}{|c|} \hline \\ \hline \\ \hline \\ \hline \\ \hline \end{array} = \begin{array}{|c|} \hline \\ \hline \\ \hline \\ \hline \\ \hline \end{array};$$

$a'_{ks} = \boxed{}$.

4°. Обновление:

(a) $NB = \begin{array}{|c|c|c|c|} \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} \\ \hline & & & \\ \hline \end{array}$, $NF = \begin{array}{|c|c|} \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} \\ \hline & \\ \hline \end{array}$.

(b) $b'^T = \begin{array}{|c|c|c|c|} \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} \\ \hline & & & \\ \hline \end{array}$.

(c) $z^+ = \boxed{}$.

(d) $B^{-1} = \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array}$.

(e) $$\overset{\pi^{+T}}{\begin{array}{|c|c|c|c|} \hline & & & \\ \hline \end{array}} = \overset{\pi^T}{\begin{array}{|c|c|c|c|} \hline & & & \\ \hline \end{array}} - \overset{\beta_k^T}{\begin{array}{|c|c|c|c|} \hline & & & \\ \hline \end{array}} * \overset{c'_s}{\boxed{}}.$$

(f) $$\underset{1 \quad\quad 2}{\overset{c'^T}{\begin{array}{|c|c|} \hline & \\ \hline \end{array}}} = \underset{3 \quad 4 \quad 5 \quad (6)}{\overset{\pi^{\mathrm{T}}}{\begin{array}{|c|c|c|c|} \hline & & & \\ \hline \end{array}}} * \overset{G}{\begin{array}{|c|c|} \hline & \\ \hline & \\ \hline & \\ \hline & \\ \hline \end{array}} + \overset{c^T}{\begin{array}{|c|c|} \hline & \\ \hline \end{array}}.$$

Шаг 3.

1°. b'^T = [ ] [ ] [ ] [ ],
БР = ([ ], [ ], [ ], [ ], [0.50]) = БДР = ОБДР;
z_{min} = [ ].

Repeat

5.8 Добавление ограничения в модифицированный метод

В разделе 4 мы рассматривали, как продолжить решение, когда на только что решенную задачу наложено какое-нибудь ограничение. Сейчас мы обратимся к этому же вопросу, но в применении к модифицированному симплекс-методу.

Рассмотрим пример 5.4.1 и продолжим его решение, спрашивая: «Что будет, если мы добавим еще одно ограничение?» Пусть при этом решено добавить следующее ограничение:

$$x_1 + x_2 + x_3 \leq 15.$$

В этот момент (см. конец разд. 5.4) мы имеем:

G

1.00	2.00	5.00
2.00	3.00	3.00
1.00	1.00	2.00
1	*2*	*3*

B^{-1}

1.00	-1.00	1.00
0.00	1.00	-2.00
0.00	-1.00	3.00

b'

12.00
6.00
21.00

NB

1	4
2	2
3	1

NF

1	6
2	5
3	3

c'^T

0.00	0.00	2.00
1	*2*	*3*

π^T

0.00	1.00	7.00
4	*5*	*6*

$-z$

249.00

A

1.00	2.00	5.00	1.00	0.00	0.00	45.00
2.00	3.00	3.00	0.00	1.00	0.00	60.00
1.00	1.00	2.00	0.00	0.00	1.00	27.00
-9.00	-10.00	-15.00	0.00	0.00	0.00	0.00

Добавление ограничения $x_1 + x_2 + x_3 \leq 15$ влечет следующие модификации этих данных:

G

1.00	2.00	5.00
2.00	3.00	3.00
1.00	1.00	2.00
1.00	1.00	1.00
1	*2*	*3*

B^{-1}

1.00	-1.00	1.00	0.00
0.00	1.00	-2.00	0.00
0.00	-1.00	3.00	0.00
0.00	0.00	0.00	1.00

b'

12.00
6.00
21.00
15.00

NB

1	4
2	2
3	1
4	(7)

NF

1	6
2	5
3	4

c'^T		
0.00	0.00	2.00
1	*2*	*<u>3</u>*

π^T			
0.00	1.00	7.00	0.00
4	*<u>5</u>*	*<u>6</u>*	*7*

$-z$
249.00

A

1.00	2.00	5.00	1.00	0.00	0.00	0.00	45.00
2.00	3.00	3.00	0.00	1.00	0.00	0.00	60.00
1.00	1.00	2.00	0.00	0.00	1.00	0.00	27.00
1.00	1.00	1.00	0.00	0.00	0.00	1.00	15.00
-9.00	-10.00	-15.00	-9.00	0.00	0.00	0.00	0.00

Искусственная переменная x_7 введена, чтобы представить добавленное (четвертое) ограничение в форме равенства:

$$x_1 + x_2 + x_3 + x_7 = 15, \quad x_7 \geq 0.$$

Однако, мы должны иметь x_7 как единственную базисную переменную в четвертом ограничении, чтобы представить задачу в канонической форме базиса. Это означает, что нам нужно исключить все другие базисные переменные, имеющиеся в этой (четвертой) строке. Через $NB = \left(\begin{array}{cccc} \scriptstyle 1 & \scriptstyle 2 & \scriptstyle 3 & \scriptstyle 4 \\ 4 & 2 & 1 & (7) \end{array}\right)$ мы видим, что надо исключить x_2 и x_1. Это исправляющее вычитание изменит только нижнюю (четвертую) строку в B^{-1} и нижний элемент в b'. (Пока мы этого не сделали, переменная x_7 помечена своим номером в *NB* условно в скобках.)

Start

Шаг 0. Исправляющее вычитание.

Смотрим через *NB* на добавленную (нижнюю) строку в *G*. Для этого мы пробегаем последовательно элементы вектора $NB(i)$ для всех $i = 1, 2, \ldots, m$ и далее просматриваем элементы нижней строки матрицы *G* в позициях, чьи номера суть $NB(i)$ (m означает начальное число ограничений, здесь m=3). Если хотя бы один такой элемент существует (присутствует) и он ненулевой, тогда нижние строки матрицы B^{-1} и столбца b' изменяются. Последовательно, это выглядит так:

1) $i = 1$, $NB(1) = \boxed{\quad}$ $\Rightarrow$ элемент отсутствует в нижней строке матрицы $G \Rightarrow$ продолжаем.

Enter

2) $i = 2$, $NB(2) = \boxed{\quad}$ $\Rightarrow G(m+1, NB(i))$ существует и он ненулевой $\Rightarrow$ мы вычитаем i-ю (вторую) строку матрицы B^{-1} и i-й (здесь второй) элемент вектора b', оба взятые с коэффициентом $G(m+1, NB(i))$, из нижней строки матрицы B^{-1} и, соответственно, нижнего элемента вектора b'. Выполняя это, получаем

B^{-1}

b'

3) i = 3, $NB(3)$ = $\square$ $\Rightarrow G(m+1, NB(i))$ существует и он ненулевой $\Rightarrow$ мы вычитаем i-е (третьи) строки матрицы B^{-1} и столбца b', взятые с коэффициентом $G(4, NB(i))$, из нижних строк матрицы B^{-1} и столбца b'. Действуя таким образом, получаем

B^{-1}

b'

NB

1	
2	
3	
4	

К этому моменту корректирующее исключение завершено; это условно отмечено тем, что скобки в четвертой строке вектора NB сняты: переменная x_7 с этого момента стала базисной в канонической форме задачи для базиса. Имеем

БР = ($\square$, $\square$, $\square$, $\square$, $\square$, $\square$, $\square$) ≠ БДР, z = $\square$.

Элементы вектора (c^T, π^T) не могут быть отрицательны, поэтому мы проверяем элементы вектора b' и, благодаря -12, видим, что должен стартовать двойственный симплекс-метод, в нашем примере, - его модифицированная версия.

Шаг 1. Работает модифицированный двойственный симплекс-метод, см. разд. 5.5.

1°. k = $\square$.

2°. a'^T_k = β^T_k * G

a'^T_k:

1	2	<u>3</u>

β^T_k:

4	<u>5</u>	<u>6</u>	7

G:

min = $\square$ $\Rightarrow$

l = $\square$,

s = $\square$,

c'_s = $\square$.

$$3^\circ \quad \underset{a'_s}{\begin{bmatrix} \ \\ \ \\ \ \\ \ \end{bmatrix}} = \underset{B^{-1}}{\begin{bmatrix} \ & \ & \ & \ \\ \ & \ & \ & \ \\ \ & \ & \ & \ \\ \ & \ & \ & \ \end{bmatrix}} * \underset{a_s}{\begin{bmatrix} \ \\ \ \\ \ \\ \ \end{bmatrix}}.$$

$a'_{ks} = \boxed{}$.

4°. (a) $NB = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ \boxed{} & \boxed{} & \boxed{} & \boxed{} \end{array}$, $NF = \begin{array}{ccc} 1 & 2 & 3 \\ \boxed{} & \boxed{} & \boxed{} \end{array}$.

(b) $b'^T = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ \boxed{} & \boxed{} & \boxed{} & \boxed{} \end{array}$.

(c) $-z = \boxed{}$.

БР = ($\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$) ≠ БДР.

(d) $B^{-1} = \begin{bmatrix} \ & \ & \ & \ \\ \ & \ & \ & \ \\ \ & \ & \ & \ \\ \ & \ & \ & \ \end{bmatrix}$.

(e)

$$\underset{\pi^{+T}}{\begin{bmatrix} \ & \ & \ & \ \end{bmatrix}} = \underset{\pi^{T}}{\begin{bmatrix} \ & \ & \ & \ \end{bmatrix}} - \underset{\beta_k^T}{\begin{bmatrix} \ & \ & \ & \ \end{bmatrix}} * \underset{c'_s}{\boxed{}}.$$

(f)

$$\underset{c'^T}{\begin{array}{ccc} \boxed{} & \boxed{} & \boxed{} \\ 1 & 2 & 3 \end{array}} = \underset{\pi^T}{\begin{array}{cccc} \boxed{} & \boxed{} & \boxed{} & \boxed{} \\ 4 & 5 & 6 & 7 \end{array}} * \overset{G}{\begin{bmatrix} \ & \ & \ \\ \ & \ & \ \\ \ & \ & \ \\ \ & \ & \ \end{bmatrix}} + \underset{c^T}{\begin{bmatrix} \ & \ & \ \end{bmatrix}}.$$

Шаг 2.

1°. $k = \boxed{}$.

$$2^\circ. \underset{a'^T_k}{\begin{array}{ccc} \boxed{} & \boxed{} & \boxed{} \\ 1 & 2 & 3 \end{array}} = \underset{\beta_k^T}{\begin{array}{cccc} \boxed{} & \boxed{} & \boxed{} & \boxed{} \\ 4 & \underline{5} & \underline{6} & \underline{7} \end{array}} * \overset{G}{\begin{bmatrix} \ & \ & \ \\ \ & \ & \ \\ \ & \ & \ \\ \ & \ & \ \end{bmatrix}}$$

$min = \boxed{} \quad \Rightarrow$

$l = \boxed{}$,

$s = \boxed{}$,

$c'_s = \boxed{}$.

3° $$\underset{a'_s}{\begin{bmatrix} \\ \\ \\ \end{bmatrix}} = \underset{B^{-1}}{\begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}} * \underset{a_s}{\begin{bmatrix} \\ \\ \\ \end{bmatrix}}$$

$a'_{ks} = \boxed{}$.

4°. (a) $NB = \overset{1\quad 2\quad 3\quad 4}{\boxed{}\boxed{}\boxed{}\boxed{}}$, $NF = \overset{1\quad 2\quad 3}{\boxed{}\boxed{}\boxed{}}$.

(b) $b'^T = \overset{1\quad 2\quad 3\quad 4}{\boxed{}\boxed{}\boxed{}\boxed{}}$.

(c) $-z = \boxed{}$.

БР = ($\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$) ≠ БДР.

(d) $$B^{-1} = \begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}.$$

(e) $$\underset{\pi^{+T}}{\begin{bmatrix} & & & \end{bmatrix}} = \underset{\pi^{T}}{\begin{bmatrix} & & & \end{bmatrix}} - \underset{\beta_k^T}{\begin{bmatrix} & & & \end{bmatrix}} * \underset{c'_s}{\boxed{}}.$$

(f) $$\underset{c'^T}{\overset{1\quad 2\quad 3}{\begin{bmatrix} & & \end{bmatrix}}} = \underset{\pi^{T}}{\overset{4\quad 5\quad 6\quad 7}{\begin{bmatrix} & & & \end{bmatrix}}} * \underset{G}{\begin{bmatrix} & & \\ & & \\ & & \\ & & \end{bmatrix}} + \underset{c^{T}}{\begin{bmatrix} & & \end{bmatrix}}.$$

Шаг 3.

1°. $k = \boxed{}$.

2°. $$\underset{a_k^{'T}}{\overset{1\quad 2\quad 3}{\begin{bmatrix} & & \end{bmatrix}}} = \underset{\beta_k^{T}}{\overset{4\quad 5\quad 6\quad 7}{\begin{bmatrix} & & & \end{bmatrix}}} * \underset{G}{\begin{bmatrix} & & \\ & & \\ & & \\ & & \end{bmatrix}}$$

$min = \boxed{}$ $\Rightarrow$

$l = \boxed{}$,

$s = \boxed{}$,

$c'_s = \boxed{}$.

$$3^\circ \quad \underset{4\times 1}{a'_s} = \underset{4\times 4}{B^{-1}} * \underset{4\times 1}{a_s}$$

$a'_{ks} = \square$.

4°. (a) $NB = (\square_1, \square_2, \square_3, \square_4)$, $NF = (\square_1, \square_2, \square_3)$.

(b) $b'^T = (\square_1, \square_2, \square_3, \square_4)$.

(c) $-z = \square$.

БР = ($\square$, $\square$, $\square$, $\square$, $\square$, $\square$, $\square$) ≠ БДР.

(d) $B^{-1} = \underset{4\times 4}{\square}$.

(e) $\pi^{+T} = \pi^T - \beta_k^T * c'_s$.

(f) $c'^T_{(1,2,3)} = \pi^T_{(4,5,6,7)} * G + c^T$.

Шаг 4.

1°. Нет отрицательных элементов в b' ⇒ конец.

ОБДР = ($\square$, $\square$, $\square$, $\square$, $\square$, $\square$, $\square$); $z_{min} = \square$.

Repeat

6
Особые случаи

Все рассмотренные выше примеры и теоремы относились к невырожденным задачам, когда имелось одно единственное решение и особых проблем не возникало. Единственное осложнение рассматривалось в разделе 3.6, где не было очевидного начального базисного допустимого решения. Чтобы справится с этим затруднением, мы вынуждены были решать вспомогательную ЛП-задачу с некоторыми искусственными переменными и искусственной целевой функцией. Тогда, в замечании 3.6.3, мы отмечали еще одну трудность: возможность противоречивых ограничений.

На практике могут возникать некоторые другие особые ситуации, которые также осложняют задачу: невырожденный базис, неограниченная допустимая область и неединственное оптимальное решение. Данный раздел иллюстрирует такие случаи, поскольку любая компьютерная программа должна быть настолько мощной, чтобы «уметь» справляться с любыми особыми ситуациями.

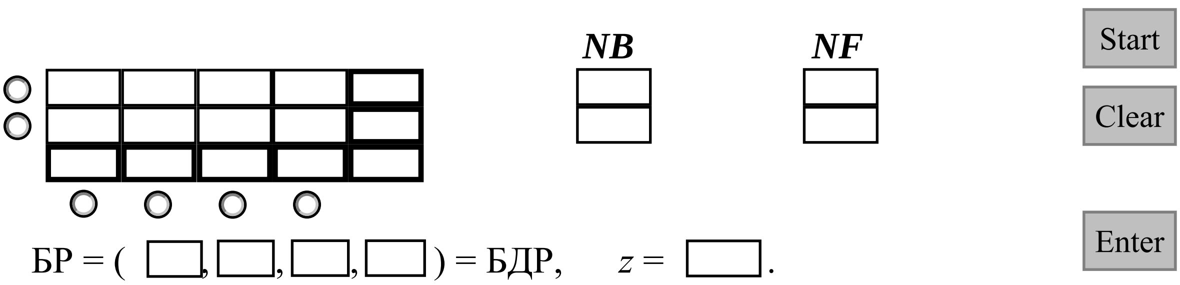
6.1 Допустимая область не существует

Начнем с ситуации противоречивых ограничений. Этот случай означает , что допустимая область не существует.

Пример 6.1.1 *Для* $x_1, x_2 \geq 0$ *решить задачу:*

$$x_1 + x_2 \geq 3;$$
$$x_1 + 2x_2 \leq 2;$$
$$-x_1 - x_2 = z \rightarrow min.$$

Шаг 0. Заполняем массивы для задачи исходными данными:



Шаг 1.

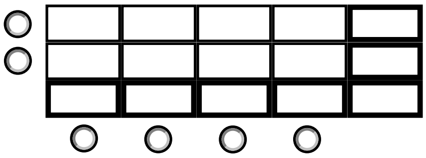
1°. s ,

l .

2°. min ,

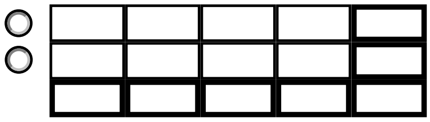
k .

3°. После нормировки:



4°. После вычитаний:

NB **NF**



БР = ($\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$) ≠ БДР, $z = \boxed{}$.

Выполнены условия применимости двойственного симплекс-метода (п. 4.1)

Шаг 2.

1°. k $\boxed{}$.

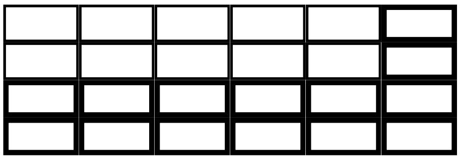
2°. В k-ой строке нет отрицательных элементов. Не существует базисного допустимого решения.

Repeat

Рассмотрим иной возможный путь решения в этом примере. Пытаясь найти некоторое БДР, мы введем искусственную переменную x_5 и попробуем решить (для начала) вспомогательную задачу: $w = x_5$, $w \to min$.

Шаг 0. Введя исходные данные, мы стартуем от таблицы, где третий и четвертый столбцы отведены для добавочных переменных x_3 и x_4, а пятый столбец – для искусственной переменной x_5:

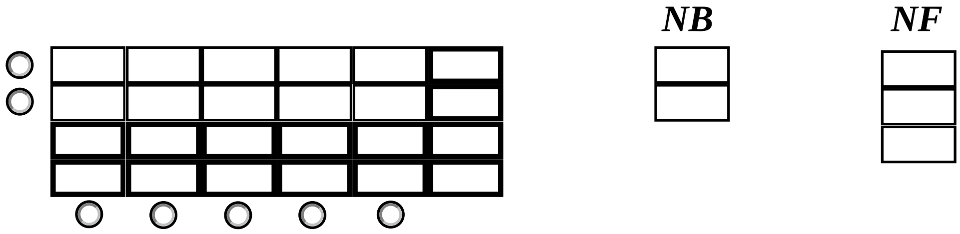
Start



Clear

Enter

Затем мы делаем подготовительное вычитание, чтобы представить стартовую таблицу в канонической форме в отношении базисных переменных $x^B = (x_4, x_5)$ и искусственной целевой функции $w = x_5$. Вычитание третьей строки из нижней строки дает



БР = (, , , ,) = БДР, z = , w = .

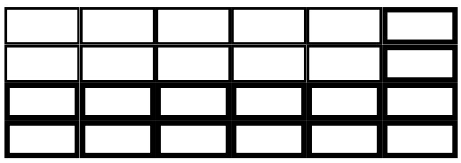
Это корректное начальное состояние, и мы готовы продолжить.

Этап I.
Шаг 1.

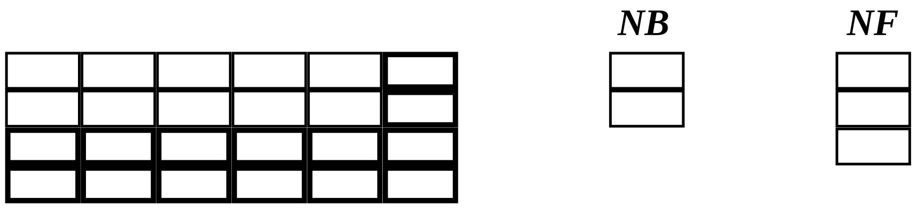
1°. s ,
l .

2°. min ,
k .

3°. После нормировки:



4°. После вычитаний:



БР = (, , , ,) = БДР, z = , w = .

Шаг 2.

1°. Нет отрицательных чисел в первых пяти элементах последней строки, поэтому минимум для w достигнут: w_{min} = 1. Мы пришли к возможности, отмечавшейся в разд. 3 (см. пример 3.6.1 и замечание 3.6.3): Этап II не может быть начат. Обнаружен случай противоречивых ограничений. С несколько иной, более общей, точки зрения (см. замечание 3.6.3) причина этого затруднения состоит в том, что БДР, полученное в п. 4° (выше), содержит (хотя бы) одну искусственную переменную, здесь x_5.

Repeat

6.2 Вырожденный базис

Вспомним формальное определение, что стандартная ЛП-задача ($Ax = b$, $c^T x = z \to min$) является вырожденной, если существует по крайней мере одна $m \times m$ подматрица, выбранная из расширенной матрицы [A, b] размера $m\times(n+1)$, которая сингулярна (вырождена). Легко видеть справедливость следующего факта.

Теорема 6.2.1 *Если ЛП-задача невырождена, то базисные переменные в любом базисном решении строго отличны от нуля.*

Поэтому, если встретится некоторое базисное решение – безразлично, допустимое или нет, - в котором хотя бы одна базисная переменная равна нулю, это служит признаком, что задача вырождена. Факт вырожденности опасен только при обнаружении именно такого базисного решения, т.е. если мы получаем нуль среди элементов столбца b' правой части ограничений $A'x = b'$. Если мы затем выберем в качестве ведущей строки ту строку, где встретился нуль среди элементов столбца b', то, очевидно, целевая функция останется неизменной, и в этом случае процедура решения может бесконечно зациклится. Неважно, уже встретился или нет нуль в правой части ограничений, – важно рассматривать это как реальную возможность для вычислительной процедуры решения. Эта процедура должна быть так разработана, чтобы оказаться способной выйти из любого потенциально бесконечного цикла, если он все же появился.

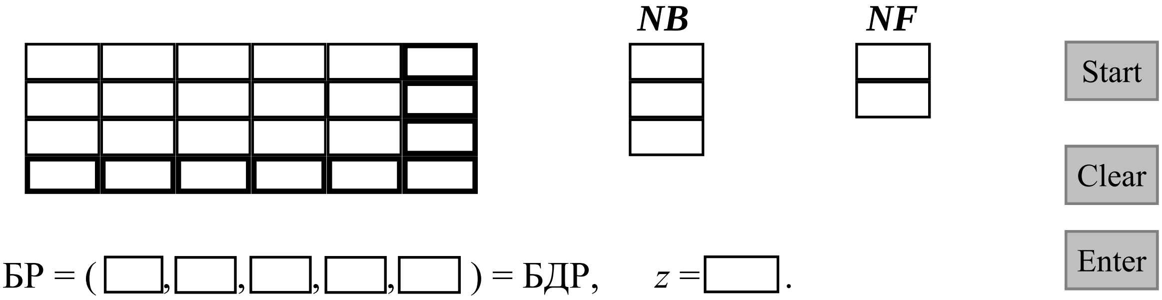
Рассмотрим несколько примеров.

Пример 6.2.1 *Для* $x_1, x_2 \geq 0$ *решить задачу:*

$$
\begin{aligned}
-2x_1 + x_2 &\leq 2;\\
-x_1 + 2x_2 &\leq 4;\\
x_1 + x_2 &\leq 8;\\
x_1 - 3x_2 &= z \to min.
\end{aligned}
$$

Используем обычный симлекс-метод.

Шаг 0.



Шаг 1.

1°. s $\boxed{\quad\quad}$,

l $\boxed{\quad\quad}$.

2°. min $\boxed{\quad\quad}$,

Выберем k $\boxed{\quad\quad}$.

3°. После нормировки:

4°. После вычитаний:

NB

NF

БР = ($\boxed{\quad}$,$\boxed{\quad}$,$\boxed{\quad}$,$\boxed{\quad}$,$\boxed{\quad}$) = БДР, z = $\boxed{\quad}$.

Базисная переменная x_4 = $\boxed{\quad}$.

Шаг 2.

1°. s $\boxed{\quad\quad}$,

l $\boxed{\quad\quad}$.

2°. min(∞, – ∞, 6/3).

Согласно п. 3.4 и табл. 3.4.1, ситуация «запрет» (– ∞) требует пересмотра значения s. Так как при этом на этом шаге другого выбора значения s нет, вернемся к предыдущему шагу 1. Там есть другой вариант выбора k, а этот шаг 2 пока отложим.

Шаг 1′.

1°. s $\boxed{\quad\quad}$,

l $\boxed{\quad\quad}$.

2°. min $\boxed{\quad\quad}$,

Выберем k $\boxed{\quad\quad}$.

3°. После нормировки:

○						
○						
○						
	○	○	○	○	○	

4°. После вычитаний:

○						
○						
○						
	○	○	○	○	○	

NB

NF

БР = (☐, ☐, ☐, ☐, ☐) = БДР, z = ☐.

Базисная переменная x_3 = ☐.

Шаг 2′.

1°. s ☐,
l ☐.
2°. min ☐,
k ☐.

3°. После нормировки:

○						
○						
○						
	○	○	○	○	○	

4°. После вычитаний:

NB

NF

БР = (☐, ☐, ☐, ☐, ☐) = ОБДР, z = ☐.

Как видно, данный ответ на ситуацию «запрет» ($-\infty$) позволил успешно завершить процесс решения. Попробуем ее наличие игнорировать, продолжая отложенный шаг 2 следующим образом.

Шаг 2.

Next

1°. s ,
l .
2°. $min(\infty, -\infty, 6/3)$.
Выберем k .
3°. После нормировки:

4°. После вычитаний:

NB ***NF***

БР = (, , , ,) = БДР, z = .

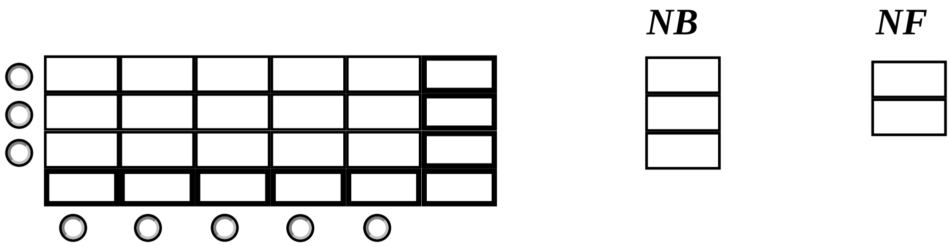
Базисная переменная x_1 = .

Шаг 3.

1°. s ,
l .
2°. $min(\infty, -\infty, 6/1)$.
Выберем, вопреки правилам из табл. 3.4.1, k .

3°. После нормировки:

4°. После вычитаний:



БР = ($\square$, $\square$, $\square$, $\square$, $\square$) = БДР, $z = \square$.

Базисная переменная $x_3 = \square$.

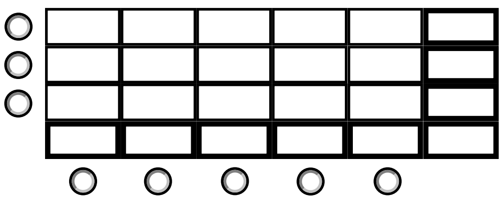
Шаг 4.

1°. s $\square$,
l $\square$.

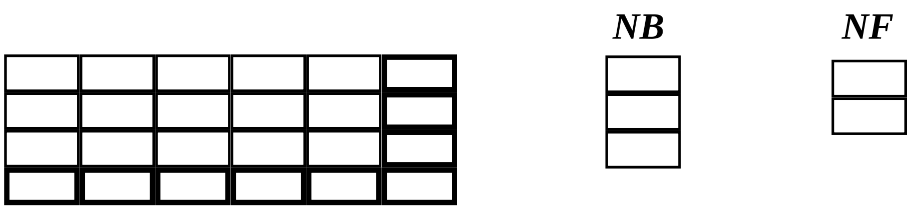
2°. $min(\ \infty, -\infty, \frac{6}{3/2})$.

Выберем, вопреки правилам из табл. 3.4.1, k $\square$.

3°. После нормировки:



4°. После вычитаний:



БР = ($\square$, $\square$, $\square$, $\square$, $\square$) = БДР, $z = \square$.

Базисная переменная $x_1 = \square$.

Объяснение происходящему при таком образе действий явлению **зацикливания** имеется простое. Объективная его причина – наложение вершин допустимого многогранника. В данном случае в точке (x_1, x_2) = (0, 2) пересекаются три ограничительных линии:

1) $x_1 = 0$;
2) $x_3 = 0$, то есть $-2x_1 + x_2 = 2$;
3) $x_4 = 0$, то есть $-x_1 + 2x_2 = 4$;

Это означает совпадение в этой точке трех вершин: (1) пересечение линий $x_1 = 0$ и $x_3 = 0$; (2) пересечение линий $x_1 = 0$ и $x_4 = 0$; (3) пересечение линий $x_3 = 0$ и $x_4 = 0$. Естественно, значения целевой функции в этих вершинах одинаковы, $z = -6$. Мы переходим из одной такой вершины в другую, фактически оставаясь в одной и той же точке, если пройдем шаг 1, шаг 2 и далее действуем так, как выбрано в шагах 3 и 4. Эти переходы соответствуют следующему графу (рис. 6.2.1).

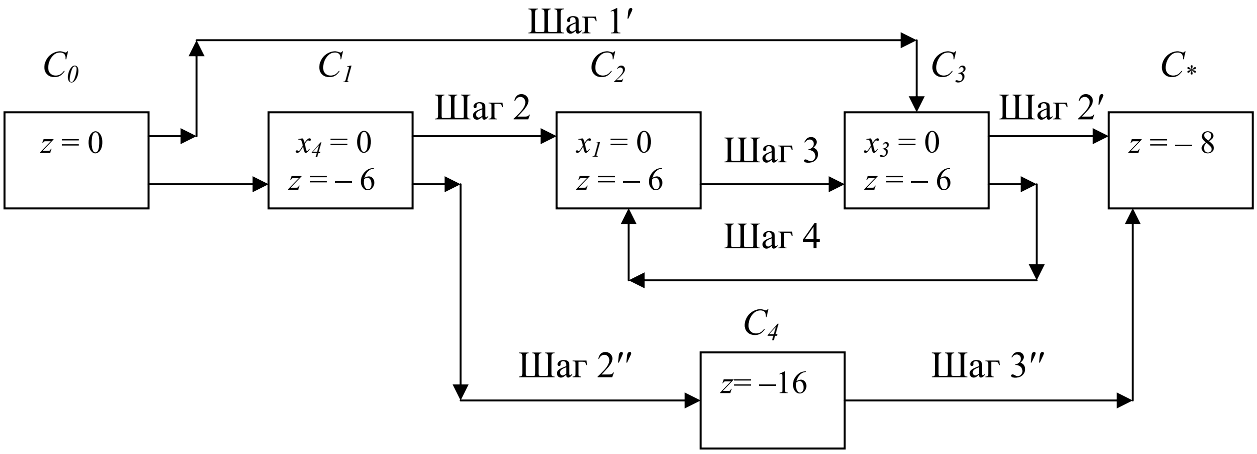


Рис. 6.2.1. Граф решения с возможным зацикливанием: C_0 – исходное состояние; C_* - оптимальное состояние; C_1, C_2, C_3 – состояния в одной из трех совпадающих вершин, в которых одна из базисных переменных (соответственно, x_4, x_1, x_3) равна нулю; C_4 – за пределами допустимой области.

Как видно, неудачны действия на шагах 2, 3 и 4. Общее в них то, что выбранная для нормировки (ведущая) строка, ее номер $k = 2$, имеет нулевое значение $b'_k = 0$ для базисной переменной от предыдущего шага: для шага 2 это $b'_k = 0$ есть значение $x_4 = 0$; для шага 3 это $b'_k = 0$ есть значение $x_1 = 0$; для шага 4 это $b'_k = 0$ есть значение $x_3 = 0$.

Как уже отмечалось, базис, в котором хотя бы одна базисная переменная равна нулю, называют **вырожденным базисом**. Он и является потенциальной причиной зацикливания, и эта причина становится актуальной, если действовать подобным образом.

Программа ЭВМ должна обладать способностью предотвращать зацикливания по указанной объективной причине. Создатель симплекс-метода Данциг предлагает изменять правые части ограничений так [4], чтобы к ним искусственно добавить малые величины ε^i, где $i = 1, 2, \ldots, m$; i - номер ограничения. Благодаря этому первоначально совпадающие вершины «разводятся», но затем в найденном приближенном решении нужно положить $\varepsilon = 0$.

Другой способ избежать зацикливания показан на этом примере: шаг 1′, шаг 2′ (рис. 6.2.1).

Шаг 1, ведущий в состояние C_1, сам по себе не содержит неудачных действий, но шаг 2 уже заводит в цикл $C_2 \leftrightarrow C_3$ (см. рис. 6.2.1). Поэтому можно иначе избежать попадания в цикл, если после шага 1 (и состояния C_1) действовать следующим образом.

Шаг 2″.

Next

1°. s ,
l .

2°. $min(\infty, -\infty, 6/3)$.
Выберем k .

3°. После нормировки:

4°. После вычитаний:

NB ***NF***

БР = (, , , ,) ≠ БДР, $z =$ $< z_{min}$.

Вышли в точку пересечения линий $-2x_1 + x_2 = 2$ (так как $x_3 = 0$) и $x_1 + x_2 = 8$ (так как $x_5 = 0$), но уже за пределы допустимой области. Вернутся оттуда и тем самым отыскать ОБДР можно теперь лишь двойственным симплекс-методом (условия его применимости распознаются как выполненные по нижней строке и по правому столбцу таблицы).

Шаг 3″.

1°. k ____.
2°. min ____, ⇒
l ____,
s ____.

3°. После нормировки:

4°. После вычитаний:

NB

NF

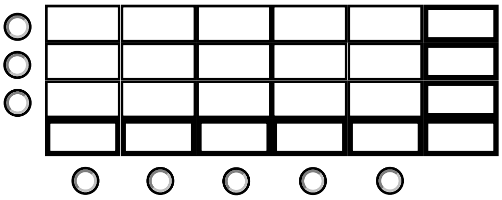
БР = (____, ____, ____, ____, ____) = ОБДР, z = ____.

В заключении отметим, что даже если и произошло попадание в цикл (здесь состояния $C_2 \leftrightarrow C_3$), из него можно выйти, если действовать правильно. Например, видно, что состояние C_3 после шага 3 идентично состоянию после шага 1′. Следовательно, если после шага 3 применить шаг 2′, то выход из цикла обеспечен. Точно так же можно выйти к финальному (оптимальному) состоянию C_* и после состояния C_2, если действовать удачно, то есть не допускать такого выбора номера k ведущей строки, при котором значении $b'_k = 0$. Рассмотрим пример.

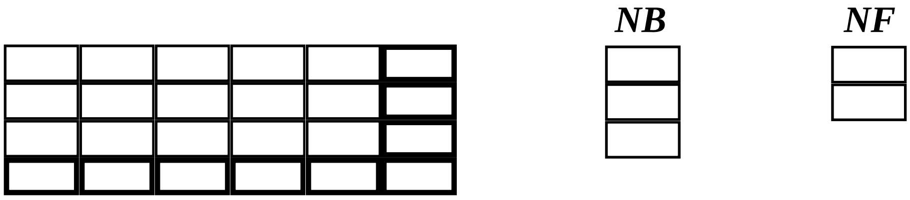
Шаг 3′. (после шага 2) Next

1°. s ____,
l ____.
2°. min ____,
k ____.

3°. После нормировки:



4°. После вычитаний:



БР = (, , , ,) = ОБДР, z = = z_{min}.

Repeat

Упражнение 6.1.1. Другой характерный пример с возможным зацикливанием известен как задача Била [4]:

$$\begin{array}{l} 1/4x_1 - \ 8x_2 - \qquad x_3 + 9x_4 \le \ 0; \\ 1/2x_1 - 12x_2 - \ 1/2x_3 + 3x_4 \le \ 0; \\ \qquad\qquad\qquad\qquad x_3 \qquad\ \le 1; \\ -3/4x_1 + 20x_2 - 1/2x_3 + 6x_4 = z \to min. \end{array}$$

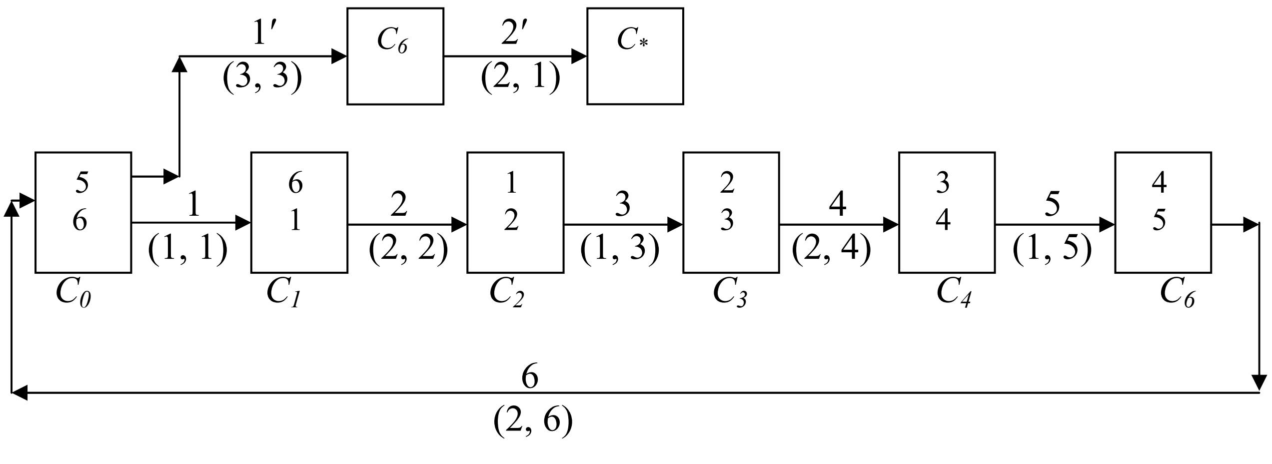


Рис. 6.2.2 Зацикливание и выход из цикла в задаче И.М.Л. Била

Пусть, как обычно, k – номер ведущей строки и s – номер ведущего столбца в матрице А ограничений задачи. Пронумеруйте ограничения в заданном порядке и выбирайте пары (k, s) шаг за шагом так: (1, 1), (2, 2), (1, 3), (2, 4), (1, 5), (2,6). Убедитесь, что после шага 6 задача вернется в исходное состояние C_0, бывшее перед шагом 1. При этом состояние C_0 имеет пару базисных переменных x_5 и x_6, равных нулю. Далее, по мере выполнения указанных шагов выбора, пары нулевых базисных переменных появляются в соответствии с рис. 6.2.2. Если же выбирать пары (k, s) так: на шаге 1′ выбрать (k, s) = (3, 3) и на шаге 2′ выбрать (k, s) = (2, 1), то появляющиеся после них состояния C_6 и C_* не имеют нулевых базисных переменных, поэтому зацикливания не возникает, причем C_* дает искомое решение: ОБДР = (1, 0, 1, 0, ¾, 0, 0), z_{min} = - 5/4.

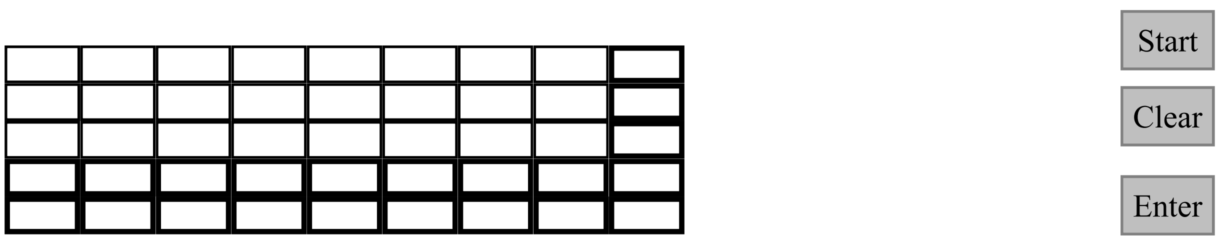
6.3 Допустимая область не ограничена

Ситуация, когда допустимая область не ограничена, указана в теореме 3.2.1 (ii) (a). Она показывает, что целевая функция может быть сделана сколь угодно малой и, следовательно, (конечного) решения не существует.

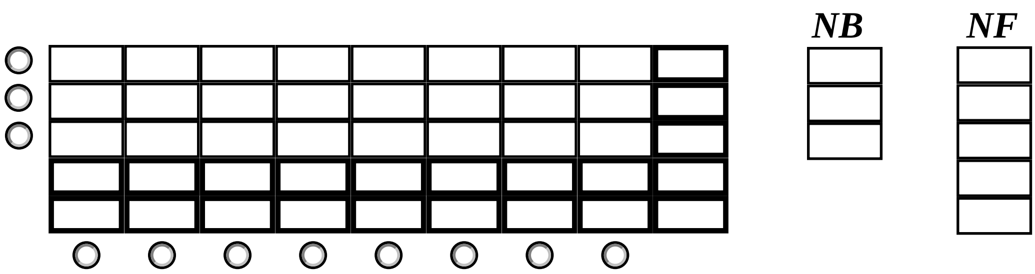
Пример 6.3.1 *Для* $x_1, x_2 \geq 0$ *решить задачу:*

$$\begin{aligned} 2x_1 + \ \ x_2 &\geq 7; \\ x_1 + \ \ x_2 &\geq 4; \\ x_1 + 3x_2 &\geq 9; \\ -2x_1 - \ 3x_2 &= z \to min. \end{aligned}$$

Шаг 0. Ввод исходных данных и генерация симплекс-таблицы:



В этой таблице последняя строка введена для вспомогательной ЛП задачи $w = x_6 + x_7 + x_8 \to min$ с искусственными переменными x_6, x_7, x_8 и искусственной целевой функцией w. На этом шаге мы делаем подготовительное вычитание, чтобы представить таблицу в канонической форме для базиса:



БР = (, , , , , , ,) = БДР, z = , w = .

Этап I. Решение вспомогательной ЛП-задачи.

Шаг 1.

1°. s ,

l .

2°. min ,

k .

3°. После нормировки:

4°. После вычитаний:

NB ***NF***

БР = (, , , , , , ,) = БДР, z = , w = .

Шаг 2.

1°. s ,

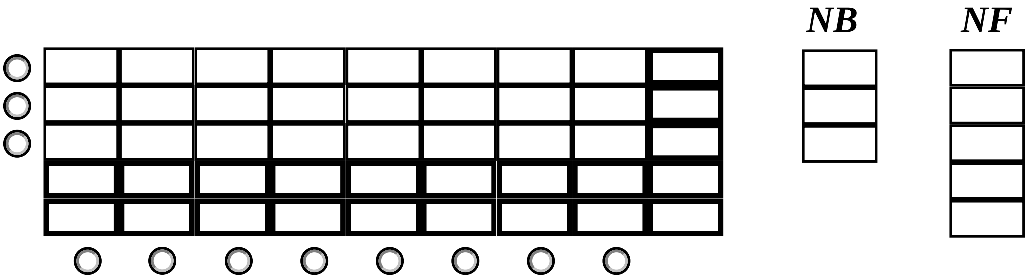
l .

2°. min ,

k .

3°. После нормировки:

4°. После вычитаний:



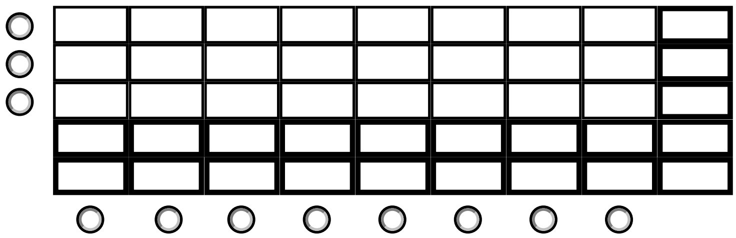
БР = (, , , , , , ,) = БДР, $z =$, $w =$.

Шаг 3.

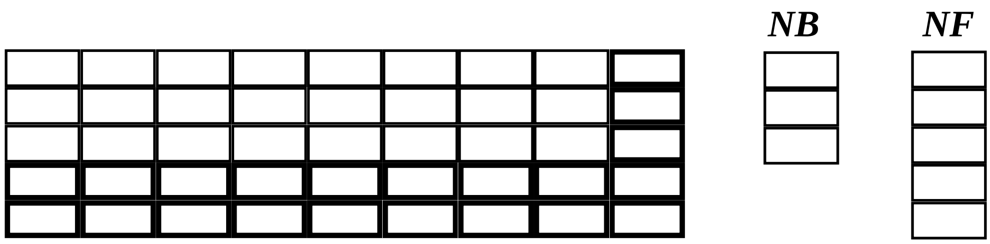
1°. s ,
l .

2°. min ,
k .

3°. После нормировки:

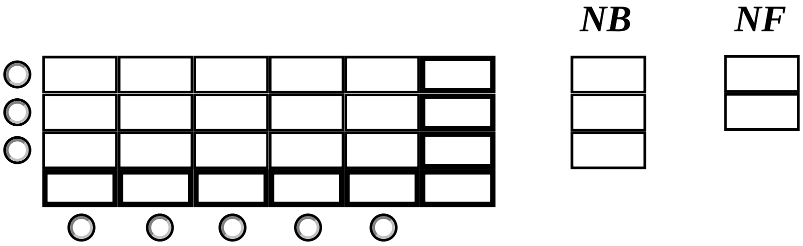


4°. После вычитаний:



БР = (, , , , , , ,) = БДР, $z =$, $w =$.

Этап I успешно завершен, и следует удалить искусственные переменные x_6, x_7, x_8, так же как искусственную целевую функцию w. В результате, мы начинаем основной этап, Этап II, со следующей таблицей:



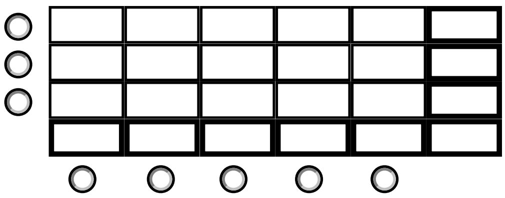
Этап II. Решение основной (заданной) ЛП-задачи.

Шаг 4.

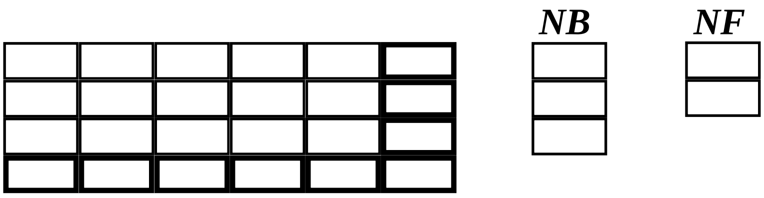
1°. *s* $\boxed{}$,
l $\boxed{}$.

2°. *min* $\boxed{}$,
k $\boxed{}$.

3°. После нормировки:



4°. После вычитаний:



БР = ($\boxed{}$,$\boxed{}$,$\boxed{}$,$\boxed{}$,$\boxed{}$) = БДР, $z = \boxed{}$.

Шаг 5.

1°. *s* $\boxed{}$,
l $\boxed{}$.

2°. $min = \infty$ $\Rightarrow$ конец.

Появление индикаторов «∞» для всех ограничений свидетельствует, что допустимая область неограничена.

Repeat

Если продолжить от этого состояния с произвольным выбором ведущего элемента в третьем столбце, то ситуация, когда все индикаторы равны «∞», будет повторяться, т.е. возможно бесконечное зацикливание. Чтобы показать это, рассмотрим еще один (более простой в вычислительном отношении) пример.

Пример 6.3.2 *Для* $x_1, x_2 \geq 0$ *решить задачу:*

$$
\begin{aligned}
x_1 - \ \ x_2 &\geq \ 1;\\
x_2 &\leq \ 2;\\
-x_1 - \ \ x_2 &= z \rightarrow min.
\end{aligned}
$$

Шаг 0.

NB	*NF*

Start

Clear

Enter

БР = (, , ,) = БДР, $z =$.

Шаг 1.

1°. s .

2°. $min(\infty, \infty) = \infty$ $\Rightarrow$

k .

3°. После нормировки:

4°. После вычитаний:

NB	*NF*

БР = (, , ,) = БДР, $z =$.

Шаг 2.

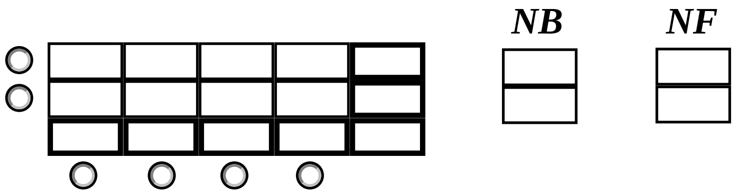
1°. s .

2°. min ,

k .

3°. После нормировки:

4°. После вычитаний:



БР = ($\boxed{\quad}$,$\boxed{\quad}$,$\boxed{\quad}$,$\boxed{\quad}$) = БДР, z = $\boxed{\quad}$.

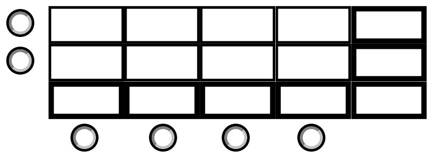
Шаг 3.

1°. s $\boxed{\quad}$.

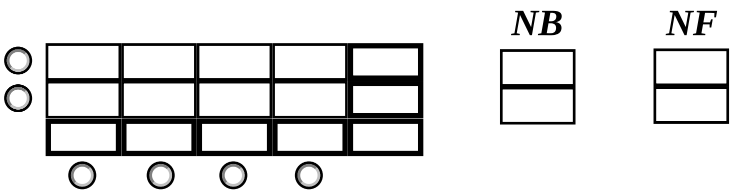
2°. $min(\infty, \infty) = \infty$ $\Rightarrow$

k $\boxed{\quad}$.

3°. После нормировки:



4°. После вычитаний:



БР = ($\boxed{\quad}$,$\boxed{\quad}$,$\boxed{\quad}$,$\boxed{\quad}$) = БДР, z = $\boxed{\quad}$.

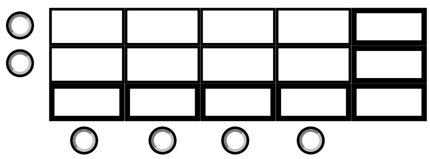
Шаг 4.

1°. s $\boxed{\quad}$.

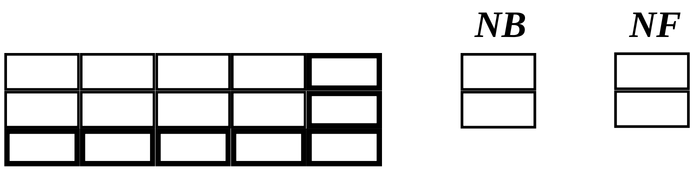
2°. $min(\infty, \infty) = \infty$ $\Rightarrow$

k $\boxed{\quad}$.

3°. После нормировки:



4°. После вычитаний:



БР = (, , ,) = БДР, $z =$.   Repeat

В этой задаче решение и минимальное значение функции z не ограничены. Это можно видеть по появлению всех признаков ∞ при пользовании табл. 3.4.1 или табл. 3.5.1 (см. п. 3.4 и п. 3.5). Однако формальное продолжение алгоритма приводит к явлению зацикливания, хотя здесь базис невырожденный.

Упражнение 6.1.2. Начертите блок-схему зацикливания в примере 6.3.1.

6.4 Неединственность оптимальных решений.

Пример 6.4.1 *Для* $x_1, x_2 \geq 0$ *решить задачу:*

$$\begin{aligned} 2x_1 + \ \ x_2 &\leq 7; \\ x_1 + \ \ x_2 &\leq 4; \\ x_1 + 3x_2 &\leq 9; \\ -x_1 - \ \ x_2 &= z \to min. \end{aligned}$$

Шаг 0.

NB   ***NF***

Start

Clear

Enter

БР = (, , , ,) = БДР, $z =$.

Шаг 1.

1°. s ,

l .

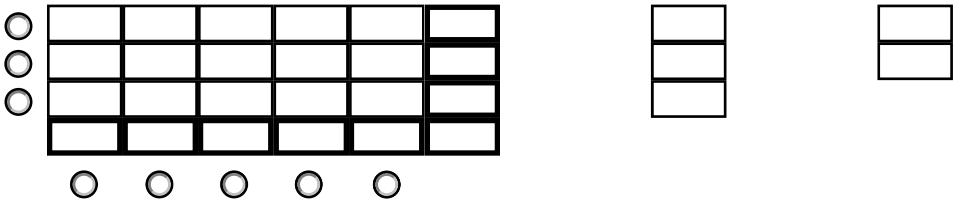
2°. min ,

k .

3°. После нормировки:

4°. После вычитаний:

NB **NF**



БР = ($\square$, $\square$, $\square$, $\square$, $\square$) = БДР, $z = \square$.

Шаг 2.

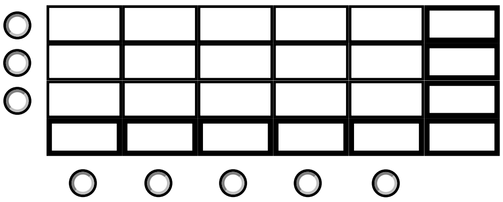
1°. s $\square$,

l $\square$.

2°. min $\square$,

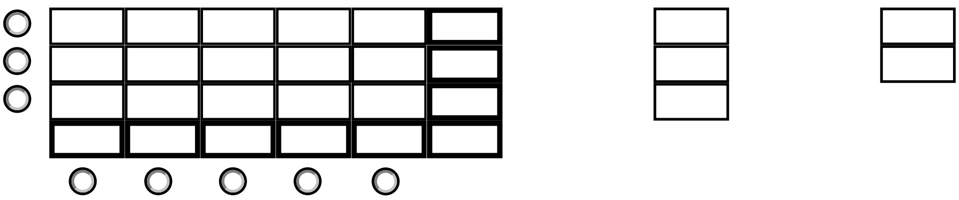
k $\square$.

3°. После нормировки:



4°. После вычитаний:

NB **NF**



БР = ($\square$, $\square$, $\square$, $\square$, $\square$) = БДР, $z = \square$.

Здесь мы имеем ситуацию, предусмотренную в теореме 3.2.1 (iii). Она обнаруживается появлением нуля в последней строке симплекс-таблицы (в строке целевой функции) для (хотя бы одной) небазисной переменной, в данном примере, - x_5, притом, что в этой строке присутствуют также и отрицательные числа (для небазисных переменных). Здесь, путем выбора ведущего элемента на позиции неотрицательного элемента пятого столбца, мы получаем второе допустимое решение из последней таблицы, причем без изменения значения целевой функции z, очевидно, благодаря упомянутому выше нулю.

Шаг 3.

1°. s ☐ ,
l ☐ .

2°. min ☐ ,
k ☐ .

3°. После нормировки:

4°. После вычитаний:

NB

NF

БР = (☐,☐,☐,☐,☐) = БДР, z = ☐.

Эта ситуация повторяется, теперь с x_3. Если продолжить, то мы получим:

Шаг 4.

1°. s ☐ ,
l ☐ .

2°. min ☐ ,
k ☐ .

3°. После нормировки:

4°. После вычитаний:

NB

NF

БР = ($\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$, $\boxed{}$) = БДР, $z = \boxed{}$. Repeat

Это состояние идентично состоянию после шага 2, так что можно видеть зацикливание между шагом 3 и шагом 4. Естественно, нет необходимости продолжать. Оптимальное базисное допустимое решение дается выпуклой оболочкой всех решений, полученных внутри цикла. Здесь мы имеем

$$\text{ОБДР} = (3 - \frac{3}{2}\theta,\ 1 + \frac{3}{2}\theta,\ \frac{3}{2}\theta,\ 0,\ 3 - 3\theta), \quad \text{где } 0 \le \theta \le 1.$$

Замечание 6.4.1 *Из вышеизложенного видно, что зацикливание возможно не только в вырожденных задачах.*

7 Учебные задания по линейному программированию

В этом разделе мы приводим набор заданий по лабораторному компьютерному практикуму по тематике линейного программирования. Каждая лабораторная работа выполняется как учебный проект, обладающий следующими особенностями:

1. Проект включает численный контрольный демонстрационный пример и формулировку алгоритма линейного программирования.
2. Проект базируется на знаниях, полученных в области информатики. Программа должна демонстрировать индивидуальный подход студента, иметь хороший стиль программирования, наиболее полно использовать возможности ЭВМ и алгоритмического языка. Рекомендуется PASCAL, но принимаются работы, написанные на C, C++ или на Delphi, Builder C++, Visual Basic и др.
3. Проект требует сбора и анализа экспериментальных данных, полученных в ходе экспериментов.
4. Все проекты ранжированы по относительной сложности, и их перечень включает 20 вариантов работ на оценку «удовлетворительно» - базовый уровень сложности, 30 вариантов на оценку «хорошо» - продвинутый уровень сложности и 20 вариантов на оценку «отлично» повышенный уровень сложности.
5. Сам студент выбирает уровень сложности из трех предложенных в соответствии со своим желанием получит ту или иную оценку за лабораторный практикум. Он может продвигаться с одного уровня на другой в соответствии с задачами, которые он ставит перед собой.

Все предлагаемые варианты заданий базируются на рассмотренных выше алгоритмах симплекс-метода. Таких алгоритмов десять; соответственно этому первая цифра номера варианта имеет возрастающие значения от 0 до 9. Выпишем их по порядку:

0. Обычный симплекс-метод при известном стартовом базисном решении, с корректным видом базиса. Ограничения «≥» и «≤», при этом ограничения типа «≥» при вводе задачи заменяют на «≤» умножением их на -1. Изложен в п. 3.4.
1. Обычный симплекс-метод при известном стартовом базисном решении, без корректного вида базиса. Ограничения «≥» и «≤». Изложен в п. 3.5.
2. Обычный симплекс-метод с порождением БДР, с искусственными переменными и искусственной целевой функцией, с корректным видом базиса. Изложен в п. 3.6. Этот алгоритм подразделен на следующие 5

вариантов заданий, отмеченных второй цифрой номера варианта и отличающихся типом ограничений, которые только и должны быть положены в основу построения программы:

2.1. Ограничения типа «≥». (Замена ограничений «≤» на «≥» должна производится вручную при подготовке к вводу задачи в ЭВМ).

2.2. Ограничения типа «≥» и «≤».

2.3. Ограничения типа «=» и «≤». (Замена ограничений «≥» на «≤» должна производится вручную при подготовке к вводу задачи в ЭВМ).

2.4. Ограничения типа «=» и «≥». (Замена ограничений «≤» на «≥» должна производится вручную при подготовке к вводу задачи в ЭВМ).

2.5. Ограничения типа «=», «≥» и «≤».

3. Двойственный симплекс-метод при известном стартовом базисном решении, с корректным видом базиса. Ограничения такого же типа, как в варианте 0. Изложен в п. 4.1.
4. Двойственный симплекс-метод при известном стартовом базисном решении, без корректного вида базиса. Ограничения такого же типа, как в варианте 1. Изложен в п. 4.2.
5. Двойственный симплекс-метод с искусственными переменными и искусственной целевой функцией (только для ограничений типа «=») для порождения базисного решения (БДР) без корректного вида базиса. Изложен в п. 4.3. Этот алгоритм подразделен на следующие 3 варианта заданий, отмеченные второй цифрой номера варианта и отличающиеся типом ограничений, которые только и должны быть положены в основу построения программы:

5.1. Ограничения типа «=» и «≤». (Замена ограничений «≥» на «≤» должна производится вручную при подготовке к вводу задачи в ЭВМ).

5.2. Ограничения типа «=» и «≥». (Замена ограничений «≤» на «≥» Должна производится вручную при подготовке к вводу задачи в ЭВМ).

5.3. Ограничения типа «=», «≥» и «≤».

6. Модифицированный симплекс-метод (обычный) при известном стартовом базисным решении. Ограничения такого же типа, как в варианте 0. Изложен в п. 5.4.
7. Модифицированный симплекс-метод (обычный) с порождением БДР, с искусственными переменными и искусственной целевой функцией. Изложен в п. 5.6. Этот алгоритм подразделен на 5 вариантов заданий в полном соответствии с предыдущими вариантами, номера которых начинаются с цифры 2:

7.1. Соответствует варианту 2.1.

7.2. Соответствует варианту 2.2.

7.3. Соответствует варианту 2.3.
7.4. Соответствует варианту 2.4.
7.5. Соответствует варианту 2.5.

8. Модифицированный симплекс-метод (двойственный) при известном стартовом базисным решении. Ограничения такого же типа, как в варианте 0. Изложен в п. 5.5.
9. Модифицированный симплекс-метод (двойственный) с искусственными переменными и искусственной целевой функцией (только для ограничений типа «=») для порождения базисного решения (≠ БДР). Изложен в п. 5.7. Ограничения соответствуют варианту 5.1.

Замечание 7.1.2 *Варианты 6-9 (модифицированный метод) требуют корректного вида базиса: среди коэффициентов базисных переменных есть только +1 и нет -1.*

Отсюда имеем следующие 20 номеров вариантов базового уровня сложности (уровень 1):

0	1	2.1	2.2	2.3	2.4	2.5	3	4	5.1
5.2	5.3	6	7.1	7.2	7.3	7.4	7.5	8	9

Следующие 50 вариантов заданий получим, вводя требование режима «Что если …?». Это означает, что после решения задачи в рамках любого из вариантов базового уровня сложности программа должна спросить ввод одного добавочного ограничения и с этого места продолжить решение. Тип добавочного ограничения указан дополнительной цифрой, добавляемой к любому из указанных 20 номеров, в соответствии с таблицей:

≤	=	≥
1	2	3

Из этих 50 вариантов следующие 30 вариантов отнесены к продвинутому уровню сложности (уровень 2):

0.1	2.1.1	2.3.1	3.1	5.1.1
0.2	2.1.2	2.3.2	3.2	5.1.2
0.3	2.1.3	2.4.1	3.3	5.2.1
1.1	2.2.1	2.4.2	4.1	5.2.2
1.2	2.2.2	2.5.1	4.2	5.3.1
1.3	2.2.3	2.5.2	4.3	5.3.2

Остальные 20 вариантов из 50 отнесены к повышенному уровню сложности (уровень 3):

6.1	7.1.2	7.2.3	7.4.2	8.2
6.2	7.1.3	7.3.1	7.5.1	8.3
6.3	7.2.1	7.3.2	7.5.2	9.1
7.1.1	7.2.2	7.4.1	8.1	9.2

Таким образом, имеем 70 вариантов: 20 вариантов базового уровня сложности, 30 вариантов продвинутого уровня сложности и 20 вариантов повышенного уровня сложности. Выбранный студентом уровень сложности может быть соотнесен с оценкой, на которую он претендует в результате лабораторного практикума: «удовлетворительно», «хорошо», «отлично». Поскольку варианты продвинутого и повышенного уровней сложности являются производными от вариантов базового уровня, студент может в процессе работы продвинутся на более высокий уровень, чтобы улучшить свой рейтинг.

Программа должна иметь удобный интерфейс (меню пользователя), демонстрировать индивидуальную работу студента и наиболее полно использовать возможности ЭВМ и языка программирования в целях экономии машинного времени и оперативной памяти ЭВМ. Она должна предоставлять возможности решения любых стандартных задач линейного программирования, поэтому для экономии памяти целесообразно использовать динамические объекты и переменные ссылочного типа (указатели). Рекомендуется заранее подготовить демонстрационные задачи. В качестве проверочных могут быть использованы примеры и задачи из данного практикума или из других источников, например, [4] или [1].

8
Тестовые задачи

Этот раздел составлен из задач, заимствованных из [1], и предназначен для использования в качестве контрольных, тестовых заданий при приеме лабораторных работ или на экзамене. Ответы к этим задачам известны, и поэтому контроль осуществляется достаточно просто. Это позволяет также применять данные задачи в качестве отладочного материала при создании программного проекта. Ответы намеренно не включены в данное пособие, чтобы стимулировать обращение студентов к другой литературе, в частности, к [1].

Студентам адресуется следующее замечание. Решение задач рассчитано на знание основных понятий симплекс-метода и понимание его работы. Это означает, что к предлагаемым текстам задач нужно относиться не слишком формально, т.е. не всегда как к исходным ЛП-задачам, а возможно, как к задачам, в которых уже проделана некоторая часть подготовительной работы. В соответствии с этим, в некоторых задачах может быть предложено восстановить исходную ЛП-задачу по исходному тексту условия. Дополнительно может быть предложено подкрепить полученное решение геометрической интерпретацией. Переход к двойственному симплекс-методу может быть достигнут путем рассмотрения одного из ограничений как добавочного после решения упрощенной задачи (без этого ограничения) или же путем еаложения дополнительного ограничения, которое предложит студент. В любых случаях нужны: обоснование наложенного ограничения, пошаговая демонстрация алгоритмов в действии и верификация (доказательство верности) полученного решения.

Задача 1. Продемонстрируйте симплекс-метод в действии на примере:

$$\begin{aligned}
30x_1 + 40x_2 &\to max;\\
12x_1 + 4x_2 &\le 300;\\
4x_1 + 4x_2 &\le 120;\\
3x_1 + 12x_2 &\le 252.
\end{aligned}$$

Задача 2. Продемонстрируйте симплекс-метод в действии на примере:

$$\begin{aligned}
x_1 + x_2 &\to max;\\
x_1 + x_2 &\to min;\\
2x_1 + 4x_2 &\le 16;\\
-4x_1 + 2x_2 &\le 8;\\
x_1 + 3x_2 &\ge 9.
\end{aligned}$$

Задача 3. Продемонстрируйте симплекс-метод в действии на примере:

$$\begin{aligned}
2x_1 + 3x_2 &\to max;\\
2x_1 + x_2 &\le 10;\\
-2x_1 + 3x_2 &\le 6;\\
2x_1 + 4x_2 &\ge 8.
\end{aligned}$$

Задача 4. Продемонстрируйте симплекс-метод в действии на примере:

$$\begin{aligned}
x_1 + 2x_2 &\to max;\\
x_1 + x_2 &\le 6;\\
3x_1 + 10x_2 &\le 26;\\
x_1 + 11x_2 &\le 20.
\end{aligned}$$

Задача 5. Продемонстрируйте симплекс-метод в действии на примере:

$$\begin{aligned}
x_1 + x_2 &\to max;\\
x_1 + 2x_2 &\le 14;\\
-5x_1 + 3x_2 &\le 15;\\
4x_1 + 6x_2 &\ge 24.
\end{aligned}$$

Задача 6. Продемонстрируйте симплекс-метод в действии на примере:

$$\begin{aligned}
x_1 + 2x_2 &\to max;\\
4x_1 - 2x_2 &\le 12;\\
-x_1 + 3x_2 &\le 6;\\
2x_1 + 4x_2 &\ge 16.
\end{aligned}$$

Задача 7. Продемонстрируйте симплекс-метод в действии на примере:

$$\begin{aligned}
2x_1 - x_2 &\to max;\\
3x_1 - 2x_2 &\le 12;\\
-x_1 + 2x_2 &\le 8;\\
2x_1 + 3x_2 &\ge 6.
\end{aligned}$$

Задача 8. Продемонстрируйте симплекс-метод в действии на примере:

$$\begin{aligned}
-x_1 + 4x_2 + 2x_4 - x_5 &\to max;\\
x_1 - 5x_2 + x_3 &= 5;\\
-x_1 + x_2 + x_4 &= 4;\\
x_1 + x_2 + x_5 &= 8.
\end{aligned}$$

Задача 9. Продемонстрируйте симплекс-метод в действии на примере:

$$\begin{aligned}
-5x_1 + x_2 - x_3 &\to max;\\
3x_1 - x_2 - x_3 &= 4;\\
x_1 - x_2 + x_3 - x_4 &= 1;\\
2x_1 + x_2 + 2x_3 + x_5 &= 7.
\end{aligned}$$

Задача 10. Продемонстрируйте симплекс-метод в действии на примере:

$$\begin{aligned}
9x_1 + 10x_2 + 16x_3 &\to max;\\
18x_1 + 15x_2 + 12x_3 &\le 360;\\
6x_1 + 4x_2 + 8x_3 &\le 192;\\
5x_1 + 3x_2 + 3x_3 &\le 180.
\end{aligned}$$

Задача 11. Продемонстрируйте симплекс-метод в действии на примере:

$$\begin{aligned}
2x_1 - 6x_2 + 5x_3 &\to max;\\
-2x_1 + x_2 + x_3 + x_5 &= 20;\\
-x_1 - 2x_2 + x_4 + 3x_5 &= 24;\\
3x_1 - 2x_2 - 12x_5 + x_6 &= 18.
\end{aligned}$$

Задача 12. Продемонстрируйте симплекс-метод в действии на примере:

$$\begin{aligned}
2x_1 + x_2 - x_3 + x_4 - x_5 &\to max;\\
2x_1 + x_2 + x_4 &= 9;\\
x_1 + 2x_2 + x_5 &= 7;\\
x_1 + x_2 + x_3 &= 5.
\end{aligned}$$

Задача 13. Продемонстрируйте симплекс-метод в действии на примере:

$$\begin{aligned}
2x_1 - 3x_2 + 6x_3 + x_4 &\to max;\\
2x_1 + x_2 - 2x_3 + x_4 &= 24;\\
x_1 + 2x_2 + 4x_3 &\le 22;\\
x_1 - x_2 + 2x_3 &\ge 10.
\end{aligned}$$

Задача 14. Продемонстрируйте симплекс-метод в действии на примере:

$$\begin{aligned}
2x_1 - x_2 - x_4 &\to min;\\
x_1 - 2x_2 + x_3 &= 10;\\
-2x_1 - x_2 - 2x_4 &\ge 18;\\
3x_1 + x_2 + x_4 &\ge 36.
\end{aligned}$$

Задача 15. Продемонстрируйте симплекс-метод в действии на примере:

$$
\begin{aligned}
2x_1 - 3x_2 + 6x_3 + x_4 &\to max;\\
x_1 + 2x_2 - 4x_3 &\le 20;\\
x_1 - x_2 + 2x_3 &\ge 10;\\
2x_1 + x_2 - 2x_3 + x_4 &= 24.
\end{aligned}
$$

Задача 16. Продемонстрируйте модифицированный симплекс-метод в действии на примере:

$$
\begin{aligned}
30x_1 + 40x_2 &\to max;\\
12x_1 + 4x_2 &\le 300;\\
4x_1 + 4x_2 &\le 120;\\
3x_1 + 12x_2 &\le 252.
\end{aligned}
$$

Задача 17. Продемонстрируйте модифицированный симплекс-метод в действии на примере:

$$
\begin{aligned}
x_1 + x_2 &\to max;\\
x_1 + x_2 &\to min;\\
2x_1 + 4x_2 &\le 16;\\
-4x_1 + 2x_2 &\le 8;\\
x_1 + 3x_2 &\ge 9.
\end{aligned}
$$

Задача 18. Продемонстрируйте модифицированный симплекс-метод в действии на примере:

$$
\begin{aligned}
2x_1 + 3x_2 &\to max;\\
2x_1 + x_2 &\le 10;\\
-2x_1 + 3x_2 &\le 6;\\
2x_1 + 4x_2 &\ge 8.
\end{aligned}
$$

Задача 19. Продемонстрируйте модифицированный симплекс-метод в действии на примере:

$$
\begin{aligned}
x_1 + 2x_2 &\to max;\\
x_1 + x_2 &\le 6;\\
3x_1 + 10x_2 &\le 26;\\
x_1 + 11x_2 &\le 20.
\end{aligned}
$$

Задача 20. Продемонстрируйте модифицированный симплекс-метод в действии на примере:

$$
\begin{aligned}
x_1 + x_2 &\to max;\\
x_1 + 2x_2 &\le 14;\\
-5x_1 + 3x_2 &\le 15;\\
4x_1 + 6x_2 &\ge 24.
\end{aligned}
$$

Задача 21. Продемонстрируйте модифицированный симплекс-метод в действии на примере:

$$\begin{aligned} x_1 + 2x_2 &\to max; \\ 4x_1 - 2x_2 &\le 12; \\ -x_1 + 3x_2 &\le 6; \\ 2x_1 + 4x_2 &\ge 16. \end{aligned}$$

Задача 22. Продемонстрируйте модифицированный симплекс-метод в действии на примере:

$$\begin{aligned} 2x_1 - x_2 &\to max; \\ 3x_1 - 2x_2 &\le 12; \\ -x_1 + 2x_2 &\le 8; \\ 2x_1 + 3x_2 &\ge 6. \end{aligned}$$

Задача 23. Продемонстрируйте модифицированный симплекс-метод в действии на примере:

$$\begin{aligned} -x_1 + 4x_2 + 2x_4 - x_5 &\to max; \\ x_1 - 5x_2 + x_3 &= 5; \\ -x_1 + x_2 + x_4 &= 4; \\ x_1 + x_2 + x_5 &= 8. \end{aligned}$$

Задача 24. Продемонстрируйте модифицированный симплекс-метод в действии на примере:

$$\begin{aligned} -5x_1 + x_2 - x_3 &\to max; \\ 3x_1 - x_2 - x_3 &= 4; \\ x_1 - x_2 + x_3 - x_4 &= 1; \\ 2x_1 + x_2 + 2x_3 + x_5 &= 7. \end{aligned}$$

Задача 25. Продемонстрируйте модифицированный симплекс-метод в действии на примере:

$$\begin{aligned} 9x_1 + 10x_2 + 16x_3 &\to max; \\ 18x_1 + 15x_2 + 12x_3 &\le 360; \\ 6x_1 + 4x_2 + 8x_3 &\le 192; \\ 5x_1 + 3x_2 + 3x_3 &\le 180. \end{aligned}$$

Задача 26. Продемонстрируйте модифицированный симплекс-метод в действии на примере:

$$\begin{aligned} 2x_1 - 6x_2 + 5x_3 &\to max; \\ -2x_1 + x_2 + x_3 + x_5 &= 20; \\ -x_1 - 2x_2 + x_4 + 3x_5 &= 24; \\ 3x_1 - 2x_2 - 12x_5 + x_6 &= 18. \end{aligned}$$

Задача 27. Продемонстрируйте модифицированный симплекс-метод в действии на примере:

$$
\begin{aligned}
&2x_1 + x_2 - x_3 + x_4 - x_5 \to max;\\
&2x_1 + x_2 + x_4 = 9;\\
&x_1 + 2x_2 + x_5 = 7;\\
&x_1 + x_2 + x_3 = 5.
\end{aligned}
$$

Задача 28. Продемонстрируйте модифицированный симплекс-метод в действии на примере:

$$
\begin{aligned}
&2x_1 - 3x_2 + 6x_3 + x_4 \to max;\\
&2x_1 + x_2 - 2x_3 + x_4 = 24;\\
&x_1 + 2x_2 + 4x_3 \le 22;\\
&x_1 - x_2 + 2x_3 \ge 10.
\end{aligned}
$$

Задача 29. Продемонстрируйте модифицированный симплекс-метод в действии на примере:

$$
\begin{aligned}
&2x_1 - x_2 - x_4 \to min;\\
&x_1 - 2x_2 + x_3 = 10;\\
&-2x_1 - x_2 - 2x_4 \ge 18;\\
&3x_1 + x_2 + x_4 \ge 36.
\end{aligned}
$$

Задача 30. Продемонстрируйте модифицированный симплекс-метод в действии на примере:

$$
\begin{aligned}
&2x_1 - 3x_2 + 6x_3 + x_4 \to max;\\
&x_1 + 2x_2 - 4x_3 \le 20;\\
&x_1 - x_2 + 2x_3 \ge 10;\\
&2x_1 + x_2 - 2x_3 + x_4 = 24.
\end{aligned}
$$

9 Программа учебных проектов по методам оптимизации

В этом разделе мы приводим перечень заданий на курсовое проектирование по нелинейным задачам оптимизации без ограничений или с ограничениями. Число существующих задач и методов оптимизации очень велико и не может быть охвачено полным изложением в рамках семестрового курса. Поэтому здесь отобраны самые популярные методы, и каждому студенту предлагается выполнить проект по разработке компьютерной программы для реализации и испытаний одного численного метода оптимизации. Однако это требует навыков работы с научной или учебной литературой, самостоятельного поиска и изучения. Мы считаем, что после первой части курса, т.е. после проекта по линейному программированию, у студента будет достаточно умения выполнить задание на курсовое проектирование по нелинейным задачам оптимизации полностью в самостоятельном режиме.

В содержании данного проекта входят: необходимые определения, описания и алгоритмическая формулировка метода, описание конструктивных особенностей предлагаемой программной реализации метода, краткая инструкция пользователя программы, план вычислительного эксперимента, а также экспериментальные результаты и выводы, вытекающие из них. Для тестирования и вычислительных экспериментов рекомендуется брать функции и ограничения из [15].

Для защиты проекта студент представляет на кафедру пояснительную записку вместе с исходными файлами программы. Проект защищается с демонстрацией программы в действии.

Мы предлагаем следующий набор из тридцати заданий на курсовое проектирование по нелинейным задачам оптимизации.

№	Вариант метода оптимизации	Рекомендуемая литература
1.	Одномерный поиск, дихотомия.	[6, 14]
2.	Одномерный поиск (Фибоначчи).	[6, 10, 3, 14]
3.	Одномерный поиск, золотое сечение.	[6, 10, 3, 14, 13]
4.	Одномерный поиск, квадратичная интерполяция (Пауэлла).	[3]
5.	Одномерный поиск, кубическая интерполяция (Давидона).	[3]
6.	Многомерный поиск, покоординатный спуск.	[6, 10, 3, 13]
7.	Многомерный поиск, метод Хука-Дживса.	[15, 3]
8.	Многомерный поиск, метод Нелдера-Мида.	[15, 3]

(продолжение)

№	Вариант метода оптимизации	Рекомендуемая литература
9.	Многомерный поиск, метод вращающихся координат (Розенброка).	[15]
10.	Многомерный поиск, метод параллельных касательных (Пауэлла).	[15]
11.	Классический метод Ньютона.	[3, 7, 10, 15]
12.	Метод Ньютона-Рафсона с выбором шага дроблением.	[7, 10, 15]
13.	Метод наискорейшего спуска.	[3, 10, 15]
14.	Метод положительно определенной формулы секущих (BFGS-формула, Бройден-Флетчер-Голдфарб-Шано), нефакторизованная форма.	[3, 7]
15.	Метод положительно определенной формулы секущих (BFGS-формула, Бройден-Флетчер-Голдфарб-Шано), факторизованная форма.	[3, 7]
16.	Метод Давида-Флетчера-Пауэлла (DFP-формула).	[3, 7]
17.	Метод сопряженных градиентов (Флетчера-Ривса).	[3, 6, 10]
18.	Поиск с ограничениями, модифицированные Хука-Дживса.	[3]
19.	Поиск с ограничениями, комплексный метод Бокса.	[3, 15]
20.	Спуск с ограничениями, метод Франка-Вулфа.	[1, 8, 12]
21.	Спуск с ограничениями, метод проекции градиента.	[6, 10, 15]
22.	Спуск с ограничениями, метод возможных направлений.	[6, 8, 10]
23.	Внутренние штрафные функции, метод Фиакко и Маккормика.	[3, 8, 6, 10, 15]
24.	Внутренние штрафные функции, метод Эрроу-Гурвица.	[1, 3, 6, 10, 15]
25.	Метод внешних штрафных функций.	[6, 10, 15]
26.	Методы случайного поиска.	[6, 11]
27.	Транспортная задача, алгоритм последовательного улучшения плана.	[4, 9]
28.	Задача о назначениях, метод Мака.	[4, 9]
29.	Задача о ранце, метод динамического программирования.	[2, 5, 9]
30.	Задача коммивояжера, методы ветвей и границ.	[9]

Заключение

В данном учебном пособии решена задача обеспечения студентов полным, тщательно выверенным набором возможных вариантов симплекс-метода для решения задач линейного прогармммирования. Варианты заданий отличаются не по вводимым для решения задачам, а по алгоритмам, подлежащим программированию.

Пособие содержит не только необходимые теоретические сведения, но все обоснования к каждому варианту симплекс-метода, указания к организации вычислений и множество иллюстрирующих примеров. Все это делает пособие замкнутым, то есть не требующим многих обращений к дополнительной литературе для выполнения индивидуальных заданий в части линейного программирования. В этой части студент разрабатывает свой программный продукт по одному из 70 различных вариантов.

Во второй части курса студент выполняет учебный проект, также с разработкой и испытанием соего программного продукта по одному из 30 вариантов, но уже на основе самостоятельной проработки специальной литературы, в основном, по методам нелинейной оптимизации при наличии или отсутствии ограничений.

Список литературы

1. Акулич И. Л. Математическое программирование в примерах и задачах: Учеб. пособие для студ. Вузов. – М.: Высш. шк., 1986. – 319 с.: ил. (2-е изд., испр. и доп. – М.: Высш. шк., 1993. -336 с.: ил.)
2. Арис А. Дискретное динамическое программирование / Пер. с англ.; Под ред. Б. Т. Поляка. – М.: Мир, 1969. -172 с.: ил.
3. Банди Б. Методы оптимизации. Вводный курс / Пер. с англ.; Под ред. В.А. Волынского. – М.: Радио и связь, 1988. – 128 с: ил.
4. Банди Б. Основы линейного программирования / Пер. с англ.; Под ред. В.А. Волынского. – М.: Радио и связь. 1989. – 176 с.: ил.
5. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования / Пер. с англ.; Под ред. А.А. Первозванского. – М.: Наука, 1965. – 460 с.: ил.
6. Васильев Ф. П. Численные методы решения экстремальных задач: Учеб. Пособие для студ. вузов спец. «Прикладная математика». – М.: Наука, 1980. – 520 с.: ил.
7. Дэннис Дж., Шнабель Р. Численные методы безусловной оптимизации и решения нелинейных уравнений / Пер. с англ.; Под ред. Ю. Г. Евтушенко. – М.: Мир, 1988. – 440 с.: ил.
8. Зайченко Ю. П. Исследование операций. – Киев: Вища школа, 1975. – 320 с.: ил.
9. Кротов В. Ф. и др. Основы теории оптимального управления. – М.: Высш. шк., 1990. – 432 с.: ил.
10. Моисеев Н. Н., Иванилов Ю. П., Столярова Е. Ж. Методы оптимизации: Учеб. пособие для студ. вузов спец. «Прикладная математика» - М.: Наука, 1978. – 352 с.: ил.
11. Растригин Л.А. Статистические методы поиска. – М.: Наука, 1968. – 376 с.: ил.
12. Сеа Ж. Оптимизация. Теория и алгоритмы / Пер. с фр. Л. Г. Гурина. – М.: Мир, 1973. – 244 с.: ил.
13. Турчак Л. И. Основы численных методов: Учеб. пособие для студ. вузов. – М.: Наука, 1987. – 320 с.
14. Уайлд Д. Дж. Методы поиска экстремума / Пер. с англ.; Под ред. А. А. Фельдбаума. – М.: Наука, 1967. – 268 с: ил.
15. Фурунжиев Р. И., Бабушкин Ф. М., Варавко В.В. Применение математических методов и ЭВМ. Практикум: Учеб. пособие для студ. вузов спец. «Прикладная математика». – Минск: Вышэйм. шк., 1988. – 191 с.: ил.

16. Galbraith P. ‘Life wasn’t meant to be easy’: Separating wheat from chaff in technology aided learning? // Vakalis I. (ed) CD ROM Proceedings of the 2nd International Conference on the Teaching of Mathematics (at the undergraduate level), University of Crete, 1-6 July 2002/ Hersonissos, Crete, Greece. (Abstract in the Book of Abstracts, P. 46.)
17. Semoushin I. V., Tsyganova J. V., Ugarov V. V. Computational and soft skills development through the project based learning // Lecture Notes in Computer Science. 2003. Vol. 2658. P. 1098-1106. / Eds. Sloot P. M. A., Abramson D., Bogdanov A. V., Dongarra J. J., Zomaya A. Y., Gorbachev Y. E. Computational Science – ICCS 2003. Berlin Heidelberg New York Barcelona Hong Kong London Milan Paris Tokyo: Springer-Verlag (2003).

Предметный указатель

П

Р

С

Т

Ф

Э