Лекция **6**

6. ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

А. СИМПЛЕКС-МЕТОД ДАНЦИГА

А1. Решение канонической задачи

Постановка задачи

Найти максимум функции

$$f(x) = \sum_{j=1}^{n} c_j x_j$$

при ограничениях

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

$$x_j \ge 0, \; j = 1,\dots,n.$$

Задача называется *канонической*, а искомое решение $x^* = (x_1^*,\dots,x_n^*)^T$ – *оптимальным*.

З а м е ч а н и я.

1. Максимизируемая функция и ограничения линейны по x_j, $j = 1,\dots,n$.

2. Задача содержит ограничения на неотрицательность переменных, присутствие которых диктуется процедурой описанного ниже симплекс-метода.

3. Будем считать, что в ограничениях все числа $b_i \ge 0$, $i = 1,\dots,m$. Этого можно добиться, умножая ограничения, где $b_i < 0$, на “-1”.

Стратегия поиска

Стратегия метода Данцига (Dantzig) решения задачи основана на особенностях постановки этой задачи. Множество

$$X = \left\{ x \;\middle|\; \sum_{j=1}^{n} a_{ij} x_j = b_i; \quad i = 1,\dots,m; \quad x \in R^n; \quad x_j \ge 0; \; j = 1,\dots,n \right\}$$

допустимых решений задачи, определяемое ограничениями, есть выпуклое множество, которое геометрически представляет собой выпуклый политоп, имеющий конечное число крайних точек.

Крайней точкой выпуклого множества X называется точка $x \in X$, которая не может быть выражена в виде выпуклой комбинации других точек $y \in X$, $x \ne y$.

Классический метод Гаусса–Жордана решения систем линейных уравнений состоит в приведении их к виду

$$x_1 \qquad + \bar{a}_{1m+1}x_{m+1} + \ldots + \bar{a}_{1s}x_s + \ldots + \bar{a}_{1n}x_n = \bar{b}_1,$$

$$\ldots$$

$$x_k \quad + \bar{a}_{km+1}x_{m+1} + \ldots + \bar{a}_{ks}x_s + \ldots + \bar{a}_{kn}x_n = \bar{b}_k,$$

$$\ldots$$

$$x_m + \bar{a}_{mm+1}x_{m+1} + \ldots + \bar{a}_{ms}x_s + \ldots + \bar{a}_{mn}x_n = \bar{b}_m.$$

Переменные $x_1, \ldots, x_m$, входящие только в одно из уравнений системы с коэффициентами 1, а во все остальные уравнения с коэффициентами, равными нулю, называются *базисными*, в то время как остальные $n-m$ переменных называются *небазисными* (*свободными*). *Базисным решением* системы называется решение

$$x_i = \bar{b}_i,\ i = 1, \ldots, m; \quad x_{m+1} = \ldots = x_n = 0.$$

Базисное решение называется *допустимым*, если $x_i \geq 0$, $i = 1, \ldots, m$. Базисное решение называется *невырожденным*, если $x_i > 0$, $i = 1, \ldots, m$.

Множество крайних точек политопа *X*, определяемого ограничениями, соответствует множеству допустимых базисных решений системы, и при этом одному базисному решению соответствует одна крайняя точка.

Утверждение. *Если функция $f(x)$ в канонической задаче достигает максимума на политопе X, определяемом ограничениями, то она достигает его по крайней мере в одной крайней точке этого политопа. Если она достигает его в нескольких крайних точках, то она достигает его на любой выпуклой комбинации этих крайних точек.*

Теорема определяет стратегию решения задачи, реализованную с помощью симплекс-метода, – это направленный перебор базисных решений, определяющих крайние точки политопа. Направленность перебора предполагает следующую организацию вычислительного процесса.

1. Нахождение начального базисного решения.

2. Переход от одного базисного решения к другому таким образом, чтобы обеспечить возрастание $f(x)$.

А2. Решение основной задачи

Постановка задачи

Найти максимум функции

$$f(x) = \sum_{j=1}^{n} c_j x_j$$

при ограничениях

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1, \ldots, m;$$

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = m+1, \ldots, p;$$

$$x_j \geq 0, \quad j = 1, \ldots, n.$$

Данная задача линейного программирования называется *основной*. Предполагается, что $b_i \geq 0$, $i = 1, \ldots, p$.

Стратегия поиска

Для решения основной задачи симплекс-методом она должна быть приведена к канонической задаче путем введения в каждое ограничение по одной дополнительной переменной: в каждое ограничение-неравенство со знаком «$\le$» вводится дополнительная переменная со знаком «$+$» (она становится базисной), а в каждое ограничение-неравенство со знаком «$\ge$» вводится дополнительная переменная со знаком «–».

Каноническая задача записывается следующим образом:

$$f(x) = \sum_{j=1}^{n} c_j x_j \to \max,$$

$$\sum_{j=1}^{n} a_{ij} x_j - x_{n+i} = b_i, \qquad i = 1, \ldots, m;$$

$$\sum_{j=1}^{n} a_{ij} x_j + \underline{x_{n+i}} = b_i, \qquad i = m+1, \ldots, p;$$

$$x_1 \ge 0, \ldots, x_{n+p} \ge 0.$$

Так как в общем случае в уравнениях нет базисных переменных, то для того, чтобы можно было применить симплекс-метод, делается переход к M-задаче. В каждое из m уравнений вводится искусственная переменная со знаком «$+$» (она становится базисной), а к целевой функции добавляется сумма искусственных переменных, умноженная на «$-M$». В результате получаем задачу в расширенной форме:

$$f(x) = \sum_{j=1}^{n} c_j x_j - M \sum_{i=1}^{m} x_{n+p+i} \to \max,$$

$$\sum_{j=1}^{n} a_{ij} x_j - x_{n+i} + \underline{x_{n+p+i}} = b_i, \quad i = 1, \ldots, m;$$

$$\sum_{j=1}^{n} a_{ij} x_j + \underline{x_{n+i}} = b_i, \qquad i = m+1, \ldots, p;$$

$$x_1 \ge 0, \ldots, x_{n+p+m} \ge 0.$$

З а м е ч а н и я.

Если решается задача поиска минимума целевой функции, то при переходе к M-задаче перед числом M ставится знак «$+$».

Б. ГРАФИЧЕСКИЙ МЕТОД РЕШЕНИЯ

В случае двух переменных задача линейного программирования имеет простую геометрическую интерпретацию и может быть решена графически с помощью следующего алгоритма.

Алгоритм

1. Построить множество допустимых решений. В общем случае оно представляет собой выпуклый многоугольник. Если ограничения в задаче несовместны, множество допустимых решений является пустым множеством, а задача поиска экстремума не имеет смысла.
2. Найти градиент целевой функции. В силу ее линейности градиент постоянен и может быть построен в любой точке координатной плоскости (как правило, он строится в начале координат).
3. Провести линию уровня функции, перпендикулярную градиенту.
4. Передвигать линию уровня параллельно самой себе до касания с множеством допустимых решений. Точки касания являются точками экстремума.
5. Классифицировать точки касания с использованием свойств градиента.

В случае непустого множества допустимых решений возможны три типовые ситуации:

а) задача имеет *единственное решение* (линия уровня касается множества допустимых решений в одной точке);

б) задача имеет *бесконечное множество решений* (линия уровня касается множества допустимых решений вдоль стороны многоугольника);

в) задача *не имеет решения* (множество допустимых решений не ограничено).

Заметим, что графически можно решать и задачи с ограничениями типа равенств, если число ограничений на единицу или на два меньше числа переменных. Способ решения: сведение к задаче с одной или двумя переменными соответственно. Для этого следует выразить целевую функцию и базисные переменные через свободные и воспользоваться условием неотрицательности, типичным для задач линейного программирования. Далее необходимо пользоваться приведенным алгоритмом графического решения задач линейного программирования.

Пример. Решить графически следующие задачи линейного программирования:

а) $f(x) = -x_1 + x_2 \to \text{extr}$, $x_1 + x_2 \le 1$, $x_1 \ge 0,\ x_2 \ge 0$;

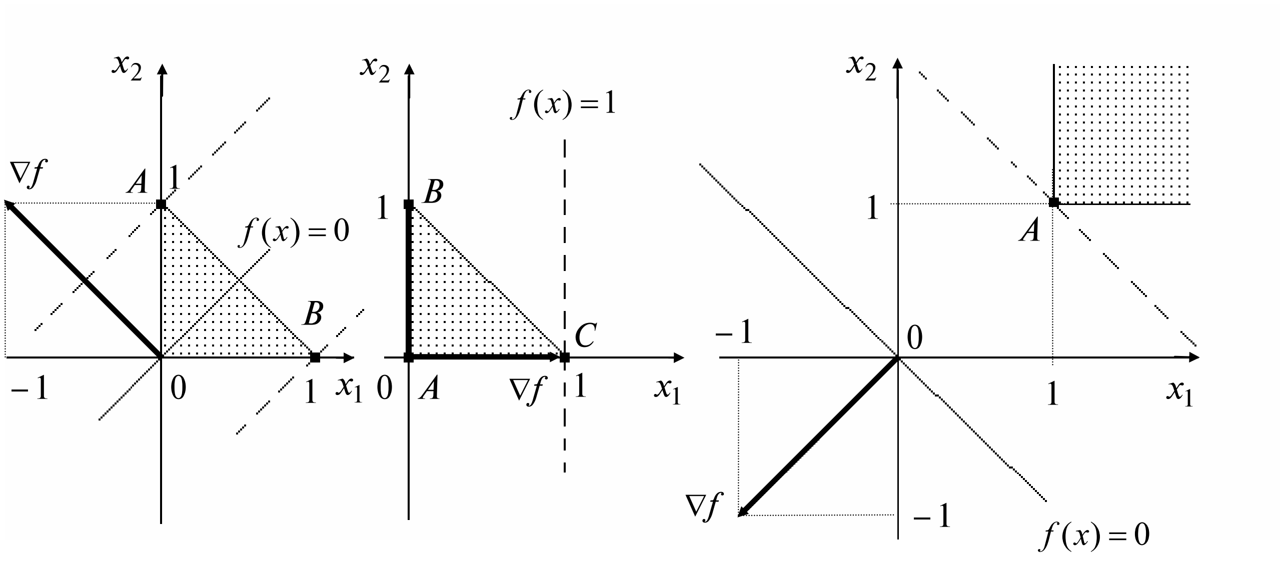
б) $f(x) = x_1 \to \text{extr}$, $x_1 + x_2 \le 1$, $x_1 \ge 0,\ x_2 \ge 0$;

в) $f(x) = -x_1 - x_2 \to \text{extr}$, $x_1 \ge 1$, $x_2 \ge 1$.

☐ Воспользуемся приведенным алгоритмом графического решения задач линейного программирования.

В задаче «а» в точке $A = (0;\,1)^T$ достигается максимум, а в точке $B = (1;\,0)^T$ – минимум. Очевидно, и минимум, и максимум единственные.

В задаче «б» в точке $C = (1;\,0)^T$ достигается максимум, а на отрезке AB – минимум, т.е. имеется бесконечное множество решений.



а *б* *в*

Рис. 1

В задаче «в» в точке $A = (1;\,1)^T$ достигается максимум, а минимума нет, так как множество допустимых решений в направлении антиградиента (наискорейшего убывания функции) неограниченное. ■

7. ЗАДАЧИ ЛИНЕЙНОГО ЦЕЛОЧИСЛЕННОГО ПРОГРАММИРОВАНИЯ. МЕТОД ВЕТВЕЙ И ГРАНИЦ

Постановка задачи

Найти максимум функции

$$f(x) = \sum_{j=1}^{n} c_j x_j$$

при ограничениях

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad i = 1, \ldots, m;$$

$$x_j \ge 0, \text{ целые}, \; j = 1, \ldots, n.$$

Задача называется *задачей линейного целочисленного программирования*.

Стратегия поиска

Задача решается симплекс-методом без учета ограничений на целочисленность переменных (эта задача обозначается ЗЛП-0). Считается, что она имеет решение. На оптимальном решении $x^{0*} = (x_1^{0*}, \ldots, x_n^{0*})^T$ вычисляется значение целевой функции $f\left(x^{0*}\right)$.

Если решение x^{0*} является целочисленным, то поставленная задача решена. Если решение x^{0*} оказывается нецелочисленным, то значение $f(x^{0*})$ является верхней границей возможных оптимальных значений $f(x)$ на целочисленных решениях. При нецелочисленном решении дальнейшая процедура решения задачи состоит в ее *ветвлении* на две: ЗЛП-1 и ЗЛП-2 (рис.2). Целью этого ветвления является разбиение множества допустимых решений, определяемого ограничениями, на два подмножества путем формирования дополнительных ограничений таким образом, чтобы исключить нецелочисленную точку x^{0*} и сделать решение, по крайней мере одной из задач, целочисленным по одной выбранной координате x_k.

Координатой x_k может быть:

- нецелочисленная координата с наименьшим или наибольшим индексом;
- нецелочисленная координата с наименьшей или наибольшей дробной частью;
- нецелочисленная координата, которой соответствует наибольший коэффициент в целевой функции;
- нецелочисленная координата, выбранная на основании приоритетов, определяемых физическим содержанием задачи.

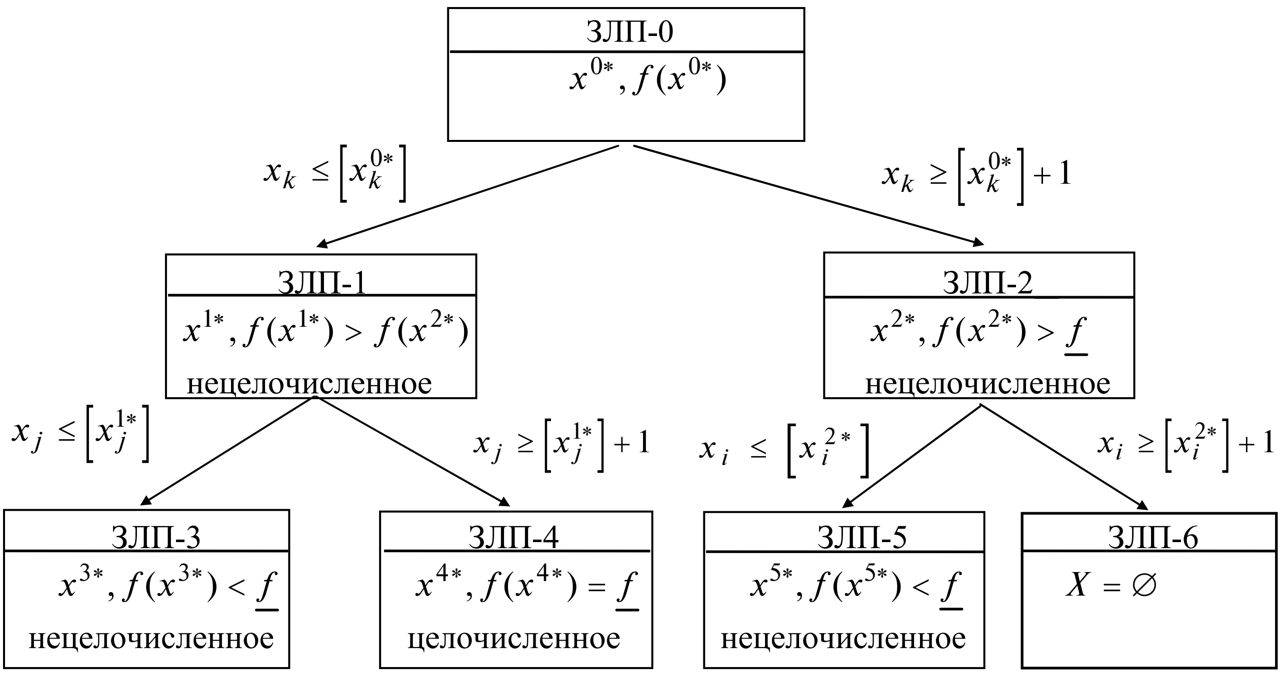


Рис. 2

Для формирования дополнительных ограничений выделяется целая часть $\left[x_k^{0*}\right]$ значения координаты x_k^{0*}. Дополнительные ограничения имеют вид $x_k \le \left[x_k^{0*}\right]$, $x_k \ge \left[x_k^{0*}\right] + 1$.

Задачи ЗЛП-1 и ЗЛП-2 записываются в следующем виде:

ЗЛП-1

$$f(x) = \sum_{j=1}^{n} c_j x_j \to \max,$$

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

$$x_k \le \left[x_k^{0*}\right],$$

$$x_j \ge 0, \ \ j = 1, \ldots, n;$$

ЗЛП-2

$$f(x) = \sum_{j=1}^{n} c_j x_j \to \max,$$

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

$$x_k \ge \left[x_k^{0*}\right] + 1,$$

$$x_j \ge 0, \ j = 1, \ldots, n.$$

Формирование дополнительных ограничений позволило исключить из рассмотрения оптимальное нецелочисленное решение x^{0*} и обеспечить целочисленность значений координаты x_k.

Задачи ЗЛП-1 и ЗЛП-2 решаются самостоятельно симплекс-методом без учета требований на целочисленность значений координат x_j, $j = 1, \ldots, n$. Вычисляются значения функции $f(x)$ на оптимальных решениях обеих задач. Если ни одна из них не имеет целочисленного решения, то выбирается задача для приоритетного дальнейшего ветвления по установленному правилу: например приоритетному ветвлению подлежит та задача, в которой значение $f(x)$ на оптимальном нецелочисленном решении максимально. Допустим, что $f\left(x^{1*}\right) > f\left(x^{2*}\right)$ и задача ЗЛП-1 первой ветвится на ЗЛП-3 и ЗЛП-4, которые решаются симплекс-методом без учета требований на целочисленность с последующим анализом решений. Если ни одна из задач ЗЛП-3 и ЗЛП-4 не имеет целочисленного решения, приступают к ветвлению задачи ЗЛП-2.

Процесс ветвления продолжается до тех пор, пока не будет получено в одной из ветвей целочисленное решение. Пусть задача ЗЛП-4 (рис. 2) имеет целочисленное решение. Обозначим $\underline{f}$ – значение функции на первом целочисленном решении: $\underline{f} = f\left(x^{4*}\right)$. Соответствующее целочисленное решение включается в множество X^* возможных оптимальных решений исходной задачи. После того как найдено первое целочисленное решение, вопрос о дальнейшем ветвлении других задач решается на основании сравнения значений $f(x^{k*})$ на оптимальных нецелочисленных решениях в оставшихся ветвях со значением $\underline{f}$.

Если $f\left(x^{k*}\right) \le \underline{f}$ для всех оставшихся k, то расчет закончен. Решениями исходной задачи являются те целочисленные решения x^{k*}, для которых $f(x^{k*}) = \underline{f}$. Если

$f\left(x^{k*}\right) > \underline{f}$, то соответствующая этому номеру k задача ветвится далее. Так, на рис. 2 имеем $f\left(x^{2*}\right) > \underline{f}$ и $f\left(x^{3*}\right) < \underline{f}$. Задача ЗЛП-2 подлежит ветвлению на ЗЛП-5 и ЗЛП-6, а ЗЛП-3 не подлежит ветвлению. Задача ЗЛП-6 не имеет решения, так как множество допустимых решений пустое, и поэтому далее она не рассматривается. Задача ЗЛП-5 имеет нецелочисленное решение $x^{5*}, f(x^{5*})$. Если $f\left(x^{5*}\right) < \underline{f}$, то решение задачи закончено и $x^* = x^{4*}$, $f(x^*) = \underline{f}$. В противном случае задача ЗЛП-5 ветвится дальше.

Если в одной из задач получено целочисленное решение, то ее ветвление далее не производится. Если соответствующее значение целевой функции не меньше $\underline{f}$, решение считается принадлежащим множеству X^* возможных оптимальных решений исходной задачи. Если значение целевой функции меньше $\underline{f}$, целочисленное решение не включается в множество X^*.

Таким образом, ветвление какой-либо задачи заканчивается, если выполняется одно из условий, а именно: решение целочисленное; значение целевой функции данной задачи не больше $\underline{f}$; множество допустимых решений пустое.

Если ветвление всех задач закончено, то в множестве X^* выбирается решение (решения), которому соответствует наибольшее значение целевой функции. Оно и является решением исходной задачи. Если множество X^* пустое, то исходная задача не имеет решения.

Алгоритм определяет правила ветвления задач и правила окончания ветвления (нахождения границ), что соответствует его названию.

З а м е ч а н и е. Распространенным методом решения задач линейного целочисленного программирования, опирающимся на сведение исходной задачи к решению последовательности задач линейного программирования без учета требования целочисленности, является метод Гомори [1,2].