Методы решения задачи линейного программирования

1. Опорные решения задачи линейного программирования

Пусть дана задача линейного программирования в канонической форме записи

$$\max\langle c, x\rangle \tag{1}$$

при условиях

$$Ax = b, \tag{2}$$

$$x \geq 0. \tag{3}$$

Будем обозначать через D множество решений системы (2) – (3). Предположим, что $r(A) = m$, где $r(A)$ – ранг матрицы A, m – количество уравнений в системе (2). Из системы $\{A_j\}_{j=1,\dots,n}$ векторов-столбцов матрицы A выберем некоторую линейно независимую подсистему из m векторов $\{A_{i_1}, A_{i_2}, \dots, A_{i_m}\}$. Она существует, так как $r(A) = m$. Эта система образует базис в E^m. Обозначим через $B = \{i_1, i_2, \dots, i_m\}$, $A_B = \{A_i\}_{i\in B}$. Назовем B ***множеством базисных значений*** индекса j, A_B – ***базисной подматрицей*** матрицы A. Координаты x_i вектора x будем называть ***базисными***, если $i \in B$, и ***небазисными*** в противном случае.

Запишем систему (2) в виде $\sum_{j=1}^{n} x_j A_j = b$.

Разобьем слагаемые левой части на базисные и небазисные, то есть

$$\sum_{i\in B} x_i A_i + \sum_{j\notin B} x_j A_j = b. \tag{4}$$

Определим частное решение этой системы следующим образом. Положим в (4) все небазисные переменные равными нулю $x_j = 0,\ \ j \notin B$. Тогда система (4) примет вид

$$\sum_{i\in B} x_i A_i = b. \tag{5}$$

Назовем (5) ***базисной подсистемой*** системы уравнений (2). Обозначим через x_B вектор, составленный из базисных координат вектора x. Тогда систему (2) можно переписать в векторно-матричном виде

$$A_B x_B = b. \tag{6}$$

Так как подматрица A_B является базисной, она невырождена. Поэтому система (6) имеет единственное решение x_B. Полученное таким образом частное решение $x = (x_B, 0)$ системы (2) назовем ***опорным решением*** прямой задачи линейного программирования, соответствующим базису A_B. (Иногда опорное решение также называют ***базисным***). Как видим, базису соответствует единственное опорное решение.

Очевидно, что число опорных решений конечно.

Для данного базиса определим также и опорное решение двойственной задачи линейного программирования. Напомним, что задача двойственная к канонической имеет вид

$$\min\langle b,\ y\rangle \tag{7}$$

при условиях

$$yA \geq c\,. \tag{8}$$

Запишем систему (8) в виде

$$\langle A_j,\ y\rangle \geq c_j, \quad j = 1, \ldots, n\,. \tag{9}$$

Напомним, что множество решений системы (8) обозначается через D .

Определим вектор y двойственных переменных из условия выполнения базисных ограничений в системе (9) как равенств. Получим следующую систему линейных уравнений:

$$\langle A_i,\ y\rangle = c_i, \quad i \in B\,. \tag{10}$$

Обозначим через c_B вектор, составленный из базисных координат вектора c. Тогда систему (10) можно переписать в векторно-матричном виде

$$yA_B = c_B\,. \tag{11}$$

Система (11) также имеет единственное решение y. Назовем его ***опорным*** (***базисным***) ***решением*** двойственной задачи линейного программирования, соответствующим базису A_B. Это опорное решение также определяется однозначно.

Итак, любому базису соответствуют два вектора – два опорных решения x и y прямой и двойственной задач линейного программирования, соответственно.

Определим далее следующие разновидности базисов и опорных решений. Если все координаты опорного решения x неотрицательны, то базис, которому соответствует это опорное решение, называется ***допустимым***. В этом случае вектор x называется ***опорным планом*** прямой задачи линейного программирования, а соответствующее тому же базису опорное решение y называется ***псевдопланом*** двойственной задачи. Фактически для допустимости базиса достаточно неотрицательности базисных координат $x_i \geq 0,\ i \in B$. Заметим, что опорный план является допустимым вектором прямой задачи линейного программирования ($x \in D$).

Если опорное решение y удовлетворяет всем ограничениям (9) двойственной задачи, то базис, которому соответствует это опорное решение, называется ***двойственно допустимым***. В этом случае вектор y называется ***опорным планом*** двойственной задачи линейного программирования, а соответствующее тому же базису опорное решение x называется ***псевдопланом*** прямой задачи. Для двойственной допустимости базиса достаточно выполнения только небазисных неравенств

$\langle A_j, y\rangle \geq c_j,\quad j \notin B$. Заметим, что опорный план y является допустимым вектором двойственной задачи линейного программирования ($y \in D$).

Разности левых и правых частей неравенств (9) обозначим через $\Delta_j = \langle A_j, y\rangle - c_j,\ \ j = 1, \dots, n$. Тогда двойственную допустимость базиса можно устанавливать, проверяя неотрицательность всех Δ_j. Заметим, что, как следует непосредственно из определения, все базисные невязки Δ_i равны нулю ($\Delta_i = 0,\ i \in B$). Поэтому достаточно убедиться в выполнении неравенств $\Delta_j \geq 0$ для всех $j \notin B$.

Познакомимся далее с некоторыми свойствами опорных решений, которые понадобятся нам при отыскании решений задач линейного программирования.

Теорема 1. Пусть x и y – опорные решения задачи линейного программирования, соответствующие некоторому базису A_B, тогда имеет место равенство $\langle c, x\rangle = \langle b, y\rangle$.

Доказательство. Из определений опорных решений легко получить равенства

$$\langle c, x\rangle = \langle c_B, x_B\rangle = \langle yA_B, x_B\rangle = \langle A_B x_B, y\rangle = \langle b, y\rangle,$$

откуда и следует справедливость теоремы.

Теорема 2. ***(Критерий оптимальности опорных решений)*** *Если базис A_B одновременно допустим и двойственно допустим, то* *соответствующие ему опорные решения x и y являются решениями, соответственно, прямой и двойственной задач линейного программирования.*

Доказательство. Справедливость этого утверждения следует из теории двойственности в линейном программировании и теоремы 1.

Теорема 3. Допустимое решение x задачи (1) – (3) является опорным планом задачи тогда и только тогда, когда оно является вершиной выпуклого многогранного множества D.

Доказательство. Пусть x – опорный план задачи (1) – (3). Докажем, что x – вершина множества D. По определению опорный план x – допустимое опорное решение, соответствующее некоторому базису A_B, то есть x – решение системы n линейных уравнений относительно n переменных

$$\begin{cases} \sum\limits_{i \in B} x_i A_i = b, \\ x_j = 0, \; j \in \{1, ..., n\} \setminus B. \end{cases}$$

Легко увидеть, что эта система имеет единственное решение. Значит, несущая плоскость грани, содержащей точку x, имеет размерность 0. Следовательно, x – вершина множества D.

Обратно. Пусть x – вершина множества D. Докажем, что x – опорный план задачи (1) – (3). Так как x – вершина, то она является гранью множества D, размерность которой равна нулю. Следовательно, у вектора x найдется не менее $n-m$

нулевых компонент, множество номеров которых обозначим $I(x)$. Таким образом, x – единственное решение системы

$$\begin{cases} \sum\limits_{i \in B(x)} x_i A_i = b, \\ x_j = 0, \ j \in I(x), \end{cases} \tag{12}$$

где $B(x) = \{1, ..., n\} \setminus I(x)$. Поэтому осталось доказать, что система векторов $\{A_i\}_{i \in B(x)}$ линейно независима. Предположим противное. Тогда существуют числа $\alpha_i, \ i \in B(x)$, не все равные нулю, такие что $\sum\limits_{i \in B(x)} \alpha_i A_i = 0$. Поэтому

$$\begin{cases} \sum\limits_{i \in B(x)} (x_i + \alpha_i) A_i = b, \\ x_j = 0, \ j \in I(x). \end{cases}$$

Это означает, что система (12) имеет решение, отличное от x, что противоречит единственности ее решения. Таким образом, $\{A_i\}_{i \in B(x)}$ – базис, а вектор x – соответствующий ему опорный план задачи (1) – (3). Что и требовалось.

Заметим, что допустимое решение y задачи (7), (8) (двойственной задаче (1) – (3)) также является опорным планом тогда и только тогда, когда оно является вершиной допустимого множества D.

2. Метод последовательного улучшения плана решения задачи линейного программирования

Метод последовательного улучшения плана основан на последовательном целенаправленном переборе допустимых базисов и соответствующих им опорных планов прямой задачи линейного программирования, начиная с любого опорного плана. Каждый последующий опорный план в этой цепочке лучше (по крайней мере, не хуже) предыдущего в смысле возрастания значений целевой функции. Это можно интерпретировать как последовательный перебор смежных (принадлежащих одному ребру) вершин выпуклого многогранного множества D. Значения целевой функции при каждом переходе от текущей вершины к следующей не убывают. Завершается такой процесс либо нахождением оптимальной вершины, либо установлением неразрешимости задачи линейного программирования.

Прежде чем сформулировать метод, основанный на таком целенаправленном переборе вер-шин, получим некоторые предварительные результаты.

Пусть выполнены все предположения о задаче линейного программирования, сделанные в предыдущем параграфе. Пусть $A_B = \left\{A_i\right\}_{i \in B}$ – некоторый базис. Разложим все векторы-столбцы A_j матрицы A по этому базису

$$A_j = \sum_{i \in B} g_{i,j} A_i . \tag{1}$$

<u>*Теорема 1.*</u> *Для любого* $j = 1, ..., n$ *справедливо равенство*

$$\Delta_j = \sum_{i \in B} c_i \, g_{i,j} - c_j . \tag{2}$$

<u>*Доказательство.*</u> Легко увидеть, что справедливы следующие равенства:

$$\Delta_j = \langle A_j, y \rangle - c_j =$$

$$= \left\langle \sum_{i \in B} g_{i,j} \, A_i , y \right\rangle - c_j = \sum_{i \in B} g_{i,j} \langle A_i , y \rangle - c_j = \sum_{i \in B} c_i \, g_{i,j} - c_j .$$

Что и требовалось.

Пусть A_B – некоторый базис и x – соответствующее ему опорное решение, пусть $k \notin B$. Построим однопараметрическое семейство векторов $x(t) = (x_1(t), ..., x_n(t))$, $t \in R$. Координаты этих век-
торов определяются формулами:

$$\begin{aligned} x_i(t) &= x_i - t \, g_{i,k}, \quad i \in B, \\ x_k(t) &= t, \\ x_j(t) &= 0, \qquad j \notin B, \, j \neq k. \end{aligned} \tag{3}$$

<u>*Теорема 2.*</u> *Для любого* $t \in R$ *выполняются равенства*

$$A x(t) = b, \tag{4}$$

$$\langle c, x(t) \rangle = \langle c, x \rangle - t \, \Delta_k . \tag{5}$$

<u>*Доказательство.*</u> Равенство $\sum_{i\in B} g_{i,k} A_i = A_k$ умножим на t и вычтем из равенства $\sum_{i\in B} x_i A_i = b$. Получим $\sum_{i\in B}\left(x_i - g_{i,k}\right)A_i + t\,A_k = b$. Отсюда и из определения векторов $x(t)$ и следует (4).

Далее,

$$\langle c, x(t)\rangle = \sum_{j=1}^{n} c_j\, x_j(t) = \sum_{i\in B} c_i\left(x_i - t\, g_{i,k}\right) + c_k t =$$

$$= \langle c_B, x_B\rangle - t\left(\sum_{i\in B} c_i\, g_{i,k} - c_k\right) = \langle c, x\rangle - t\,\Delta_k .$$

(Последнее равенство вытекает из теоремы 1.) Что и требовалось.

Множество векторов $x(t)$ представляет собой прямую, проходящую через опорное решение x (поскольку $x(0)=x$) и целиком лежащую в многообразии частных решений системы линейных алгебраических уравнений $Ax = b$.

Обозначим $B_k^+ = \left\{i :\ i\in B,\ g_{i,k} > 0\right\}$.

<u>*Теорема 3.*</u> *Пусть* A_B *– допустимый базис,* $k\notin B$, $\Delta_k < 0$ *и* $B_k^+ = \varnothing$. *Тогда* $x(t)\in D$ *при всех* $t \ge 0$ *и* $\langle c,\ x(t)\rangle \to +\infty$ *при* $t\to+\infty$.

<u>*Доказательство.*</u> Пусть $t \ge 0$. Для проверки включения $x(t)\in D$, как следует из теоремы 2, нужно убедиться в неотрицательности всех координат вектора $x(t)$. Пусть $i\in B$. Тогда имеем

$x_i(t) = x_i - t\,g_{i,k} \geq 0$, так как $x_i \geq 0,\quad t \geq 0,\quad g_{i,k} \leq 0$. А для $j = k$ имеем $x_k(t) = t \geq 0$. И, наконец, для $j \notin B,\ j \neq k$ из (3) $x_j = 0$.

Справедливость второго утверждения теоремы следует из теоремы 2 (равенство (5)) и отрицательности Δ_k. Теорема доказана.

Поясним полученный результат. Полупрямая $x(t),\ t \geq 0$, представляет собой луч с вершиной в точке x. В условиях теоремы вектор x является также вершиной (опорным планом) многогранного множества D, а луч $x(t),\ t \geq 0$, – неограниченным ребром этого множества. Следовательно, множество D неограничено и, как видим из второго утверждения теоремы, целевая функция $\langle c,\, x\rangle$ также неограниченна сверху на D. Таким образом, теорема 3 дает признаки неразрешимости прямой задачи по причине неограниченности целевой функции на допустимой области. Двойственная задача при этом также не имеет решения, так как ее допустимая область пуста.

<u>*Теорема 4*</u>. *Пусть* A_B *– допустимый базис,*

$k \notin B$ *и* $B_k^+ \neq \varnothing$. *Тогда* $x(t) \in D,\quad t \in [0,\, t_k]$, *где*

$$t_k = \min_{i \in B_k^+} \left\{ \frac{x_i}{g_{i,k}} \right\}. \tag{6}$$

<u>*Доказательство*</u>. Как видим, здесь множество

B_k^+, в отличие от условий теоремы 3, не пусто. Поэтому включение $x(t) \in D$ имеет место только для тех значений параметра t, которые удовлетворяют системе неравенств

$$x_i(t) = x_i - t\, g_{i,k} \geq 0, \quad \forall i \in B_k^+ . \tag{7}$$

Разрешив неравенства (7) относительно t, получим $t \leq \dfrac{x_i}{g_{i,k}}, \quad \forall i \in B_k^+$. Откуда и вытекает утверждение теоремы.

Обсудим полученный результат. Поскольку согласно теореме 4 величина t_k – максимально возможное значение параметра t, множество точек прямой $x(t)$, принадлежащих допустимой области D, представляет собой отрезок. Этот отрезок является ребром многогранного множества D. Опорный план x – это одна из вершин этого ребра. Вычислим вторую вершину, смежную с x по отношению к данному ребру. Для этого вычислим вектор $\tilde{x} = x(t_k)$. Пусть в формуле (6) минимум достигается при $i = l$. (Заметим, что $l \in B$.) Тогда $t_k = \dfrac{x_l}{g_{l,k}}$. Подставим определенное таким образом значение t_k в формулы (3). Получим

$$\tilde{x}_i = x_i - \frac{x_l}{g_{l,k}} g_{i,k}, \quad \forall i \in B,$$

$$\tilde{x}_k = \frac{x_l}{g_{l,k}},$$

$$\tilde{x}_j = 0, \qquad \forall j \notin B,\ j \neq k\,.$$

Легко увидеть, что $\tilde{x}_l = x_l - \dfrac{x_l}{g_{l,k}} g_{l,k} = 0$. Поэтому полученные формулы удобнее переписать следующим образом:

$$\begin{aligned}
\tilde{x}_i &= x_i - \frac{x_l}{g_{l,k}} g_{i,k}, \quad \forall i \in B,\ i \neq l,\\
\tilde{x}_k &= \qquad \frac{x_l}{g_{l,k}}, \\
\tilde{x}_j &= 0, \qquad \forall j \notin B,\ j \neq k,\ j = l.
\end{aligned} \tag{8}$$

Исходя из полученного, можно предположить, что вектор $\tilde{x}$ – это вторая вершина построенного ребра, то есть опорный план. Чтобы убедиться в справедливости этого предположения, необходимо показать, что подсистема $A_{\tilde{B}}$, где $\tilde{B} = \left(B \setminus \{l\}\right) \bigcup \{k\}$, является базисом. Этот факт устанавливается в следующей теореме.

Теорема 5. *Пусть выполнены все требования теоремы 4. Тогда подсистема* $A_{\tilde{B}}$, *где* $\tilde{B} = \left(B \setminus \{l\}\right) \bigcup \{k\}$, *является базисом, а вектор* $\tilde{x}$, *определяемый формулами (8), является соответствующим этому базису опорным планом прямой задачи линейного программирования.*

Доказательство. Очевидно, что для доказательства теоремы достаточно убедиться в линейной независимости подсистемы $A_{\tilde{B}}$. Предположим противное. Тогда существуют числа

$\alpha_i,\ i \in \tilde{B}$, не все равные нулю, такие, что

$$\sum_{i \in \tilde{B}} \alpha_i A_i = 0. \qquad (9)$$

Нетрудно увидеть, что $\alpha_k \neq 0$. Иначе $\sum_{i \in B \setminus \{l\}} \alpha_i A_i = 0$, что противоречит линейной независимости подсистемы векторов $\{A_i\}_{i \in B \setminus \{l\}}$. Поэтому из (9) получаем $A_k = \sum_{i \in B \setminus \{l\}} \beta_i A_i$, где $\beta_i = -\alpha_i / \alpha_k$. С другой стороны, $A_k = \sum_{i \in B} g_{i,k} A_i$. Таким образом, имеем два различных разложения вектора A_k по базису A_B. Что невозможно. Полученное противоречие доказывает теорему.

Заметим, что на основании равенства (5)

$$\langle c,\ \tilde{x} \rangle = \langle c,\ x \rangle - t_k \Delta_k = \langle c,\ x \rangle - \frac{x_l}{g_{l,k}} \Delta_k. \qquad (10)$$

Отсюда при $\Delta_k < 0$ получаем $\langle c,\ \tilde{x} \rangle \geq \langle c,\ x \rangle$. Если же $x_l > 0$, то $\langle c,\ \tilde{x} \rangle > \langle c,\ x \rangle$, что обеспечит увеличение значения целевой функции при переходе из вершины x в вершину $\tilde{x}$.

Полученные результаты позволяют сформулировать общую (принципиальную) схему метода последовательного улучшения плана. Метод состоит из последовательного выполнения однотипных итера-ций. Каждая итерация осуществляет переход от данного базиса к новому.

Предположим, что имеется некоторый допустимый базис A_B и соответствующий ему опорный план x прямой задачи линейного программирования. Далее выполняются следующие действия.

1. Нахождение опорного решения (псевдоплана) y, соответствующего базису A_B. Для этого решается система линейных алгебраических уравнений $yA_B = c_B$.

2. Проверка признака оптимальности. На основании критерия оптимальности опорных решений (см. теорему 2.1) необходимо проверить двойственную допустимость базиса, то есть выполнение неравенств

$$\Delta_j = \langle A_j, y \rangle - c_j \geq 0, \quad \forall j \notin B. \qquad (11)$$

Возможны два случая:

a) Все неравенства (11) выполняются. Тогда векторы x и y являются решениями, соответственно, прямой и двойственной задач линейного программирования. На этом работа метода прекращается.

b) Существуют $j \notin B$, для которых $\Delta_j < 0$. Переходим к следующему этапу.

3. Выбор номера $k \notin B$ вектора, вводимого в базис. Зафиксируем k такой, что $\Delta_k < 0$. (Например, $k : \Delta_k = \min\limits_{j \notin B} \{\Delta_j\}$.)

4. Нахождение коэффициентов $g_{i,k}$ разложения вектора A_k по базису. Для этого решается система линейных уравнений $\sum_{i\in B} g_{i,k} A_i = A_k$.

5. Построение множества значений индекса i: $B_k^+ = \left\{ i : i \in B,\ g_{i,k} > 0 \right\}$.

6. Проверка пустоты множества B_k^+. Возможны два случая.

a) $B_k^+ = \varnothing$. Тогда, как следует из теоремы 3, прямая задача линейного программирования не имеет решения в силу неограниченности целевой функции на допустимом множестве, а двойственная задача не имеет решения в силу пустоты допустимого множества. На этом работа метода прекращается.

b) $B_k^+ \neq \varnothing$. Переходим к следующему этапу.

7. Выбор номера $l \in B$ вектора, выводимого из базиса. Зафиксируем l: $\dfrac{x_l}{g_{l,k}} = \min\limits_{i \in B_k^+} \left\{ \dfrac{x_i}{g_{i,k}} \right\}$.

8. Построение нового базиса $A_{\tilde{B}}$, где $\tilde{B} = \left(B \setminus \{l\} \right) \bigcup \{k\}$.

9. Вычисление нового опорного плана $\tilde{x}$, соответствующего базису $A_{\tilde{B}}$, по формулам (8).

10. Замена базиса A_B на базис $A_{\tilde{B}}$ и опорного плана x на опорный план $\tilde{x}$. Переход к следующей итерации, к первому ее этапу.

Заключительную часть этого параграфа посвятим обоснованию конечности этого метода.

Введем следующие понятия. Опорный план x прямой задачи линейного программирования называется ***невырожденным***, если все его базисные координаты положительны ($x_i > 0, \quad \forall i \in B$). В противном случае опорный план называется ***вырожденным***. Легко увидеть, что невырожденный опорный план может иметь только один базис. В отличие от этого, вырожденный опорный план может соответствовать нескольким различным базисам. Далее, прямая задача линейного программирования называется ***невырожденной***, если все ее опорные планы невырождены. В противном случае задача линейного программирования называется ***вырожденной***.

<u>Теорема 6.</u> ***(Теорема о конечности метода последовательного улучшения плана)*** *Пусть прямая задача линейного программирования является невырожденной, имеется некоторый исходный допустимый базис A_B и соответствующий ему опорный план x прямой задачи. Тогда за конечное число итераций метода последовательного улучшения плана либо будут найдены решения прямой и двойственной задач линейного программирования, либо будет установлена их неразрешимость.*

Справедливость этой теоремы следует из предположения о невырожденности прямой задачи линейного программирования. В этом случае на каждой итерации $x_l > 0$, что согласно (10) гарантирует положительное приращение целевой функции на каждой итерации. Тогда конечность количества итераций вытекает из конечности числа опорных планов.

Итак, теорема 6 обосновывает возможность решения взаимосопряженных задач линейного программирования методом последовательного улучшения плана за конечное число итераций. Однако, как видим, это возможно при предположении о невырожденности прямой задачи линейного программирования. Можно ли применять этот метод для вырожденных задач линейного программирования, которые часто встречаются в различных приложениях? Чтобы ответить на этот вопрос, напомним, что у вырожденной задачи имеются вырожденные опорные планы. У таких опорных планов, как мы отмечали выше, может быть несколько различных базисов. Если на какой-либо итерации опорный план вырожден, у него имеется несколько различных базисов, и значение $x_l = 0$, то появляется возможность перебора различных базисов этого опорного плана без изменения значения целевой функции. Более того, возможно, что в этом переборе будут цикли-
чески строиться по очереди одни и те же базисы. Это явление называется «зацикливанием». При такой ситуации говорить о конечности метода не приходится. В некоторых руководствах по

линейному программированию приводятся примеры такого зацикливания и приемы его преодоления. Однако опыт применения метода показывает, что зацикливание происходит редко.

3. Симплексный метод

Метод последовательного улучшения плана является принципиальной схемой решения задач линейного программирования, основанной на целенаправленном переборе допустимых базисов. В методе последовательного улучшения плана не конкретизированы способы решения систем линейных алгебраических уравнений в пунктах 1 и 4, а также остается открытым вопрос об отыскании исходного допустимого базиса и соответствующего ему опорного плана. На основе метода последовательного улучшения плана созданы различные численно реализуемые методы (алгоритмы). Одним из них является ***симплексный метод***.

Введем так называемую ***симплексную таблицу*** $G(B)$, представляющую собой матрицу размерности $(m+1)\times(n+1)$, которая однозначно определяется по базису A_B следующим образом:

$$G(B)=\left\{\begin{matrix} x_{i_1} & g_{i_1,1} & \cdots & g_{i_1,j} & \cdots & g_{i_1,n} \\ . & . & \cdots & . & \cdots & . \\ x_{i_s} & g_{i_s,1} & \cdots & g_{i_s,j} & \cdots & g_{i_s,n} \\ . & . & \cdots & . & \cdots & . \\ x_{i_m} & g_{i_m,1} & \cdots & g_{i_m,j} & \cdots & g_{i_m,n} \\ \langle c_B, x_B\rangle & \Delta_1 & \ldots & \Delta_j & \ldots & \Delta_n \end{matrix}\right\}.$$

Значения Δ_j в этой матрице вычисляются по формулам $\Delta_j = \sum_{i\in B} c_i\, g_{i,j} - c_j$, полученным в теореме 1 параграфа 2.

Симплексная таблица содержит всю необходимую информацию для осуществления итерации метода последовательного улучшения плана. Заметим, что вычисление базисных координат опорного решения x_i и коэффициентов $g_{i,j}$ по определению (то есть путем решения систем линейных уравнений) довольно трудоемко и осуществляется только один раз на начальном шаге метода. В дальнейшем при переходе от данного базиса к следующему (от одной итерации метода к следующей) симплексная таблица пересчитывается по сравнительно простым правилам, основанным на методе исключения. Формулы пересчета симплексной таблицы получены в следующей теореме.

Теорема 1. *Пусть* A_B *и* $A_{\tilde{B}}$ *– допустимые базисы, где* $\tilde{B} = \left(B\setminus\{l\}\right)\cup\{k\}$, *значения индексов* l

и

k получены по правилам метода последовательного улучшения плана, $G(B)$ и $G\left(\tilde{B}\right)$ – соответствующие этим базисам симплексные таблицы. Тогда для элементов симплексной таблицы $G\left(\tilde{B}\right)$ справедливы следующие равенства:

$$\begin{aligned} \tilde{x}_k &= \frac{x_l}{g_{l,k}}, \\ \tilde{x}_i &= x_i - \frac{x_l}{g_{l,k}} g_{i,k}, \quad \forall i \in B,\ i \neq l, \end{aligned} \tag{1}$$

$$\begin{aligned} \tilde{g}_{k,j} &= \frac{g_{l,j}}{g_{l,k}},\ \ j = 1, \dots, n, \\ \tilde{g}_{i,j} &= g_{i,j} - \frac{g_{l,j}}{g_{l,k}} g_{i,k}, \quad \forall i \in B \setminus \{l\},\ \ j = 1, \dots, n, \end{aligned} \tag{2}$$

$$\left\langle c_{\tilde{B}},\ \tilde{x}_{\tilde{B}} \right\rangle = \left\langle c_B,\ x_B \right\rangle - \frac{x_l}{g_{l,k}} \Delta_k, \tag{3}$$

$$\tilde{\Delta}_j = \Delta_j - \frac{g_{l,j}}{g_{l,k}} \Delta_k, \quad j = 1, \dots, n. \tag{4}$$

Доказательство. Формулы (1) и (3) получены ранее (см. формулы (8) и (10) параграфа 2).

Обоснуем равенства (2). Из разложения вектора $A_k = \sum_{i \in B} g_{i,k} A_i = \sum_{i \in B \setminus \{l\}} g_{i,k} A_i + g_{l,k} A_l$ выразим

$$A_l = \sum_{i \in B \setminus \{l\}} \left(-\frac{1}{g_{l,k}} g_{i,k} \right) A_i + \frac{1}{g_{l,k}} A_k . \qquad (5)$$

Выберем произвольно $j = 1, ..., n$ и в разложение

вектора A_j подставим (5). Получим

$$A_j = \sum_{i \in B} g_{i,j} A_i = \sum_{i \in B \setminus \{l\}} g_{i,j} A_i + g_{l,j} A_l =$$

$$= \sum_{i \in B \setminus \{l\}} \left(g_{i,j} - \frac{g_{l,j}}{g_{l,k}} g_{i,k} \right) A_i + \frac{g_{l,j}}{g_{l,k}} A_l = \sum_{i \in \tilde{B}} \tilde{g}_{i,j} A_i .$$

Откуда и следуют формулы (2).

Далее убедимся в справедливости равенств (4). Выберем произвольно $j = 1, ..., n$. Тогда

$$\tilde{\Delta}_j = \sum_{i \in \tilde{B}} c_i \tilde{g}_{i,j} - c_j = \sum_{i \in \tilde{B}} c_i \left(g_{i,j} - \frac{g_{l,j}}{g_{l,k}} g_{i,k} \right) - c_j =$$

$$= \sum_{i \in B \setminus \{l\}} c_i \left(g_{i,j} - \frac{g_{l,j}}{g_{l,k}} g_{i,k} \right) + c_k \frac{g_{l,j}}{g_{l,k}} - c_j =$$

$$= \Delta_j - c_l g_{l,j} - \frac{g_{l,j}}{g_{l,k}} \left(\sum_{i \in B \setminus \{l\}} c_i g_{i,k} - c_k \right) =$$

$$= \Delta_j - c_l g_{l,j} - \frac{g_{l,j}}{g_{l,k}} \left(\Delta_k - c_l g_{i,k} \right) = \Delta_j - \frac{g_{l,j}}{g_{l,k}} \Delta_k .$$

Что и требовалось.

Теперь мы можем сформулировать ***симплексный метод***. Выберем некоторый допустимый базис A_B. Далее выполняются следующие действия:

0. Построение исходной симплексной таблицы $G(B)$ (этот этап осуществляется только один раз до начала выполнения первой итерации).

1. Проверка признака оптимальности

$$\Delta_j \geq 0, \quad \forall j \notin B. \tag{6}$$

Если все неравенства (6) выполняются, то x – решение задачи, и метод прекращает работу.

2. Выбор $k \notin B$. Положим $k:\ \Delta_k = \min_{j \notin B} \{\Delta_j\}$.

3. Построение $B_k^+ = \{i:\ i \in B,\ g_{i,k} > 0\}$.

4. Если $B_k^+ = \varnothing$, то задача не имеет решения в силу неограниченности целевой функции на допустимом множестве. Работа метода прекращается.

5. Выбор $l \in B$. Найдем номер l такой, что

$$\frac{x_l}{g_{l,k}} = \min_{i \in B_k^+} \left\{ \frac{x_i}{g_{i,k}} \right\}.$$

6. Построение нового базиса $A_{\tilde{B}}$, где $\tilde{B} = (B \setminus \{l\}) \bigcup \{k\}$.

7. Пересчет симплексной таблицы по формулам (1) – (4).

8. Переход к следующей итерации, к

первому ее этапу.

4. Метод искусственного базиса

Этот параграф посвящен приемам нахождения исходного допустимого базиса необходимого

для начала работы метода последовательного улучшения плана.

Пусть необходимо решить задачу линейного программирования в канонической форме записи

$$\max\langle c, x\rangle$$

$$Ax = b, \qquad (1)$$

$$x \geq 0.$$

Предположим, не ограничивая общности, что все координаты вектора b неотрицательны. Метод искусственного базиса имеет две разновидности.

Рассмотрим так называемый ***двухфазный*** вариант метода искусственного базиса.

Первая фаза. Этот этап состоит в решении вспомогательной задачи. Для ее построения вводятся переменные $x_{n+1}, ..., x_{n+m}$, которые называются ***искусственными***. Вспомогательная задача – это задача линейного программирования вида:

$$\max\ -\sum_{i=1}^{m} x_{n+i}$$

$$\langle a_i, x\rangle + x_{n+i} = b_i, \quad i = 1,\ldots,m, \tag{2}$$

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

Вспомогательная задача имеет следующие особенности.

1) Задача (2) имеет решение при любой исходной задаче (1).

Обозначим $\tilde{x} = \left(0,\ldots,0,\ b_1,\ldots,b_m\right) \in E^{n+m}$. Очевидно, что вектор удовлетворяет всем ограничениям задачи (2), то есть ее допустимая область непуста.

Далее, целевая функция ограничена сверху на допустимой области, так как в силу неотрицательности всех координат допустимого вектора

$$-\sum_{i=1}^{m} x_{n+i} \le 0\,.$$

2) Легко увидеть, что вектор $\tilde{x}$ является опорным планом задачи (2), его базис A_B состоит из искусственных векторов A_{n+1}, ..., A_{n+m}.

Таким образом, выполняются все требования, позволяющие применить к задаче (2) симплексный метод. В результате получим оптимальный план $\bar{x} = \left(\bar{x}_1,\ldots,\bar{x}_n,\bar{x}_{n+1},\ldots,\bar{x}_{n+m}\right)$ и соответствующий ему базис $A_{\bar{B}}$ вспомогательной задачи.

Вторая фаза. Начнем с анализа результата, полученного на первой фазе. Возможны следующие случаи.

1) Среди искусственных координат вектора $\bar{x}$ имеются ненулевые (положительные), то есть найдется хотя бы один номер $i \in I = \{1, ..., m\}$ такой, что $\bar{x}_{n+i} > 0$. Тогда, очевидно, $-\sum_{i=1}^{m} \bar{x}_{n+i} < 0$ (максимальное значение целевой функции задачи (2) отрицательно). Убедимся, что в этом случае допустимая область исходной задачи пуста. Допустим, что $D \neq \varnothing$. Тогда для любого $z = (z_1, ..., z_n) \in D$ вектор $(z_1, ..., z_n, 0, ..., 0)$ является допустимым решением задачи (2), на котором значение целевой функции равно нулю. Полученное противоречие означает, что $D = \varnothing$, то есть исходная задача линейного программирования (1) не имеет решения. На этом вторая фаза заканчивается.

2) Все искусственные координаты вектора $\bar{x}$ небазисные ($n+i \notin B, \quad i \in I$). В этом случае вектор $(\bar{x}_1, ..., \bar{x}_n)$, составленный из первых n координат вектора $\bar{x}$, является опорным планом задачи (1) с базисом $A_{\overline{B}}$. В этом случае вторая фаза заключается в решении задачи (1) симплексным методом, где в качестве исходного допустимого базиса используется $A_{\overline{B}}$.

3) Все искусственные координаты вектора $\bar{x}$ нулевые ($\bar{x}_{n+i} = 0, \quad i = 1, ..., m$), но среди них имеются

базисные (то есть найдется хотя бы один номер $i \in \{1, ..., m\}$ такой, что $n+i \in \bar{B}$). Чтобы воспользоваться этим опорным планом для решения исходной задачи симплексным методом, надо вывести эти искусственные векторы из базиса. Наличие нулевых базисных компонент означает, что построенный опорный план вырожден, а значит, ему соответствует несколько различных базисов. Значит, можно перейти к другому базису этого опорного плана, не содержащему искусственных векторов.

Пусть множества $I_1 = \left\{ i : i \in I,\ n+i \notin \bar{B} \right\}$ и $I_2 = \left\{ i : i \in I,\ n+i \in \bar{B} \right\}$ составляют разбиение множества I. Запишем задачу линейного программирования, которая решается на второй фазе.

$$\begin{aligned} &\max\ \langle c, x\rangle \\ &\qquad \langle a_i, x\rangle \qquad\quad = b_i, \quad i \in I_1, \\ &\qquad \langle a_i, x\rangle + x_{n+i} = b_i, \quad i \in I_2, \\ &\qquad x_j \ge 0, \quad j \in \{1, ..., n\} \cup I_2 . \end{aligned} \tag{3}$$

Для применения к задаче (3) симплексного метода необходимо исключить из базиса $A_{\bar{B}}$ оставшиеся искусственные векторы, после чего фактически решается исходная задача.

В заключение изложим так называемый ***однофазный*** вариант метода искусственного базиса. В нем вместо основной задачи (1) и вспомогательной задачи (2) решается следующая задача:

$$\max \quad \langle c, x \rangle - M \sum_{i=1}^{m} x_{n+i}$$
$$\langle a_i, x \rangle + x_{n+i} = b_i, \quad i = 1, \ldots, m, \qquad (4)$$
$$x_j \ge 0, \quad j = 1, \ldots, n+m,$$

где M – неопределенный положительный параметр. Эта задача в некотором смысле эквивалентна исходной задаче (1), так как она имеет следующие свойства.

1. Очевидно, что задача (4) всегда имеет допустимое решение (что устанавливается так же, как и для задачи (2)).

2. Задача (1) имеет решение тогда и только тогда, когда существуют значения параметра M такие, что задача (4) имеет решение, в котором все искусственные координаты равны нулю. Тогда если $\left(x_1^*, \ldots, x_n^*, 0, \ldots, 0\right)$ – решение задачи (4), то вектор $x^* = \left(x_1^*, \ldots, x_n^*\right)$ – решение задачи (1).

3. Задача (1) не имеет решения в силу несовместности ограничений тогда и только тогда, когда существуют значения параметра M такие, что задача (4) имеет решение, в котором некоторые искусственные координаты положительны.

4. Задача (1) не имеет решения в силу неограниченности целевой функции на допустимой

области тогда и только тогда, когда для любых значений параметра M задача (4) не имеет решения по той же причине.

Все эти утверждения легко проверяются.

5. Решение неканонических задач линейного программирования

5.1 Решение симметричных задач линейного программирования

Пусть необходимо решить задачу линейного программирования в симметричной форме записи:

$$\begin{gathered} \max \ \langle c, x \rangle \\ Ax \le b, \\ x \ge 0. \end{gathered} \tag{1}$$

Предположим, что все координаты вектора b неотрицательны. Заметим, что в этом случае допустимая область D задачи (1) непуста (например, точка $0 \in D$).

Преобразуем задачу (1) к каноническому виду. Добавим в левые части неравенств системы $Ax \le b$, так называемые ***дополнительные*** переменные $x_{n+1}, \dots, x_{n+i}, \dots, x_{n+m}$. Получим, таким образом, следующую каноническую задачу линейного программирования:

$$\max \langle c,\ x\rangle$$
$$\langle a_i,\ x\rangle + x_{n+i} = b_i,\quad i = 1, ..., m, \qquad (2)$$
$$x_j \geq 0,\quad j = 1, ..., n+m.$$

Легко убедиться, что задачи (1) и (2) эквивалентны.

Заметим, что система векторов $\{A_{n+1}, ..., A_{n+m}\}$ является базисом, а в силу неотрицательности всех b_i он допустим, то есть опорное решение $(0, ..., 0, b_1, ..., b_m) \in E^{n+m}$ является опорным планом задачи (2).

Отметим еще одну особенность задачи (2). Пусть симплексным методом найдено ее решение. Тогда для оптимального базиса справедливы равенства $y_i^* = \Delta_{n+i}^*,\quad i = 1, ..., m$. (В справедливости

этих равенств убедитесь самостоятельно.) Следовательно, вектор $y^* = (y_1^*, ..., y_m^*)$ является решением задачи двойственной к задаче (1).

Прежде, чем перейти к другим неканоническим задачам, заметим следующее. Часто матрица A задачи линейного программирования имеет столбцы, которые удобно включить в исходный базис. В таких случаях нет надобности вводить полный искусственный базис, а только лишь достроить уже имеющуюся часть базиса до полного введением некоторого числа искусственных переменных. Иногда этого удается добиться элементарными преобразованиями системы ограничений задачи. Если такие преобразования выполнить легко, то можно добиться заметной экономии при дальнейших вычислениях. Учет специфики конкретной задачи, опыт решения разнообразных задач линейного программирования могут помочь найти удачный способ преобразования задачи вместо введения полного искусственного базиса. Далее приведем пример задачи, где целесообразны предварительные преобразования.

5.2 Решение задач линейного программирования с ограничениями вида $Ax \geq b,\ x \geq 0$.

Укажем один тип задач линейного программирования, для которых простые предварительные преобразования могут позволить обойтись не

более чем одной искусственной переменной. Пусть необходимо решить задачу линейного программирования следующего вида:

$$\max \langle c, x \rangle \qquad Ax \geq b, \qquad x \geq 0. \tag{3}$$

Предположим, что все координаты вектора b неотрицательны. Очевидно, что сведение задачи (3) к симметричной форме записи умножением на –1 каждого неравенства системы $Ax \geq b$ не позволит использовать прием, описанный в пункте 5.1. Поэтому, прежде всего, преобразуем задачу (3) к каноническому виду. Для этого введем дополнительные переменные $x_{n+1}, ..., x_{n+m}$, вычитая их из левых частей соответствующих неравенств системы $Ax \geq b$. Получим, таким образом, следующую каноническую задачу:

$$\begin{gathered} \max \langle c, x \rangle \\ \langle a_i, x \rangle - x_{n+i} = b_i, \quad i = 1, ..., m, \\ x_j \geq 0, \quad j = 1, ..., n+m. \end{gathered} \tag{4}$$

Далее, выберем номер s одной компоненты вектора b так, что $b_s = \max\limits_{i=1,...,m} \{ b_i \}$ и преобразуем систему уравнений задачи (4) следующим образом. Заменим каждое i-тое уравнение ($i \neq s$) в этой системе на разность s-ого и i-ого уравнения, уравнение с номером s оставим без изменения. Получим задачу

$$\begin{aligned} &\max \langle c, x \rangle \\ &\langle a_s - a_i, x \rangle - x_{n+s} + x_{n+i} = b_s - b_i, \ i = 1, ..., m, \ i \neq s, \\ &\langle a_s, x \rangle - \quad x_{n+s} \qquad\quad = b_s, \\ &x_j \geq 0, \quad j = 1, ..., n+m. \end{aligned} \tag{5}$$

Задача (5) выгодно отличается от задачи (4) наличием почти полного исходного базиса. Столбцы с номерами $n+1, ..., n+s-1, n+s+1, ..., n+m$ матрицы коэффициентов ограничений этой задачи пригодны для включения в базис. Соответствующие им базисные координаты равны $b_s - b_i$ и, в силу определения величины b_s, неотрицательны. Для построения исходного базиса может не хватить только лишь одного вектора, поэтому в случае необходимости использования метода искусственного базиса потребуется введение одной искусственной переменной x_{n+m+1} в s-тое уравнение.

5.3 Решение задач линейного программирования при наличии переменных, не ограниченных по знаку.

Иногда моделями некоторых проблем могут являться задачи линейного программирования, у которых все или некоторые переменные не ограничены по знаку, то есть могут принимать как отрицательные, так и неотрицательные значения.

Для простоты изложения предположим, что дана задача, у которой все переменные не ограничены по знаку. Опишем прием, позволяющий привести такую задачу к эквивалентной форме, в которой все переменные неотрицательны. Пусть, например, дана задача

$$\begin{gathered} \max \langle c,\ x\rangle \\ Ax \le b. \end{gathered} \tag{6}$$

Воспользуемся тем, что любую величину a можно представить в виде разности двух неотрицательных чисел $a = b - c$, где $b \ge 0,\ \ c \ge 0$. Заметим, что такое представление неоднозначно. Благодаря этому, в задаче (6) можно сделать следующую замену переменных:

$$x_j = y_j - y_{n+1}, \quad j = 1, \dots, n, \tag{7}$$

где $y_j \geq 0,\ j = 1, \dots, n+1$. После подстановки (7) в условия задачи (6) получим следующую задачу:

$$\begin{gathered} \max \langle c, y\rangle - \left(\sum_{j=1}^{n} c_j\right) y_{n+1} \\ \langle a_i, y\rangle - \left(\sum_{j=1}^{n} a_{i,j}\right) y_{n+1} \leq b_i, \quad i = 1, \dots, m, \\ y \geq 0, \quad y_{n+1} \geq 0, \end{gathered} \tag{8}$$

где $y = (y_1, \dots, y_n)$.

Задача (8) эквивалентна задаче (6). Необходимо, конечно, после получения ее решения сделать обратный переход к переменным x.

6. Метод обратной матрицы

Метод обратной матрицы (называемый также ***модифицированным симплексным методом***) является еще одним численно реализуемым алгоритмом решения задач линейного программирования, основанным на общей схеме метода последовательного улучшения плана.

Напомним, что на каждой итерации метода последовательного улучшения плана решаются следующие системы линейных алгебраических уравнений:

- система для вычисления базисных координат опорного плана x

$$A_B x_B = b, \qquad (1)$$

- система для вычисления псевдоплана y

$$y A_B = c_B, \qquad (2)$$

- система для вычисления коэффициентов $g_{i,k}$ разложения вектора A_k по базису

$$\sum_{i \in B} g_{i,k}\ A_i = A_k. \qquad (3)$$

Решение этих систем составляет наиболее трудоемкий этап реализации метода последовательного улучшения плана.

Запишем систему (3) в векторно-матричном виде. Обозначив $G_k = \left(g_{i_1,k}, g_{i_2,k}, ..., g_{i_m,k}\right)$, имеем

$$A_B G_k = A_k. \qquad (4)$$

Так как A_B – базисная подматрица, то она

невырождена. Поэтому для нее существует обратная матрица A_B^{-1}. Воспользуемся обратной матрицей для отыскания решений систем уравнений (1), (2), (4). Получим следующие равенства:

$$x_B = A_B^{-1} b, \qquad (5)$$

$$y = c_B A_B^{-1}, \qquad (6)$$

$$G_k = A_B^{-1} A_k. \qquad (7)$$

Использование формул (5) – (7) и лежит в основе алгоритма метода обратной матрицы.

Следующая теорема дает формулы для пересчета обратной матрицы при переходе к новому базису.

<u>*Теорема 1*</u>. *Пусть* $A_B^{-1} = \{\alpha_{i,j}\}$ *и* $A_{\tilde{B}}^{-1} = \{\tilde{\alpha}_{i,j}\}$ *– две обратные матрицы, где* $\tilde{B} = (B \setminus \{l\}) \bigcup \{k\}$; *значения индексов* l *и* k *получены по правилам метода последовательного улучшения плана. Тогда справедливы следующие равенства:*

$$\begin{aligned} \tilde{\alpha}_{k,j} &= \frac{\alpha_{l,j}}{g_{l,k}}, && \forall j \in B, \\ \tilde{\alpha}_{i,j} &= \alpha_{i,j} - \frac{\alpha_{l,j}}{g_{l,k}} g_{i,k}, && \forall i \in B,\ i \neq l;\ j \in B. \end{aligned} \qquad (7)$$

Формулы (7) выводятся аналогично формулам для пересчета симплексной таблицы (см. теорему 3.1).

Из формул (7) следует, что обратная матрица на каждой итерации вычисляется не по

определению, а из предыдущей обратной матрицы по тем же правилам, которые использовались и для пересчета симплексных таблиц.

Теперь мы можем сформулировать общую итерацию метода обратной матрицы. Предположим, что имеется некоторый допустимый базис A_B и соответствующий ему опорный план x. Далее выполняются следующие действия:

0. Вычисление обратной матрицы A_B^{-1} для исходного базиса.

1. Нахождение опорного решения (псевдоплана) y, соответствующего базису A_B по формуле (6) $y = c_B A_B^{-1}$.

2. Проверка признака оптимальности. Для этого проверяется двойственная допустимость базиса, то есть выполнение неравенств

$$\Delta_j = \langle A_j, y \rangle - c_{j_j} \ge 0, \quad \forall j \notin B. \tag{8}$$

Если все неравенства (8) выполняются, то векторы x и y являются решениями, соответственно, прямой и двойственных задач линейного программирования. На этом работа метода прекращается.

3. Выбор $k \notin B$. Положим $k: \Delta_k = \min\limits_{j \notin B} \{\Delta_j\}$.

4. Нахождение коэффициентов разложения вектора A_k по базису: $G_k = A_B^{-1} A_k$.

5. Построение $B_k^+ = \{i : i \in B,\ g_{i,k} > 0\}$.

6. Если $B_k^+ = \varnothing$, то прямая задача линейного программирования не имеет решения в силу неограниченности целевой функции на допустимом множестве, а двойственная задача не имеет решения в силу пустоты допустимого множества. Работа метода прекращается.

7. Выбор номера $l \in B$ такого, что

$$\frac{x_l}{g_{l,k}} = \min_{i \in B_k^+} \left\{ \frac{x_i}{g_{i,k}} \right\}.$$

8. Построение нового базиса $A_{\tilde{B}}$, где $\tilde{B} = \left(B \setminus \{l\}\right) \cup \{k\}$.

9. Пересчет обратной матрицы по формулам (7) и вычисление нового опорного плана $\tilde{x}$, соответствующего базису $A_{\tilde{B}}$, по формулам (2.6).

10. Замена базиса A_B на $A_{\tilde{B}}$ и матрицы A_B^{-1} на $A_{\tilde{B}}^{-1}$. Переход к следующей итерации, к первому ее этапу.

Метод обратной матрицы имеет ряд существенных преимуществ по сравнению с симплексным методом. Обратная матрица имеет меньшие размеры, чем симплексная таблица. Особенно это существенно, когда m значительно меньше n. На каждой итерации метода обратной матрицы в большей мере, чем в симплексном методе, используются данные исходной задачи линейного программирования. Поэтому в ходе большого количества итераций погрешности вычислений меньше

накапливаются, что приводит к большей устойчивости этого алгоритма по сравнению с симплексным методом.

Наконец, отметим, что для отыскания исходного допустимого базиса и соответствующего ему
опорного плана также возможно применение метода искусственного базиса в сочетании с методом обратной матрицы.

7. Метод последовательного уточнения оценок

Метод последовательного уточнения оценок основан, так же как и метод последовательного улучшения плана, на последовательном целенаправленном переборе базисов. Но в методе последовательного уточнения оценок перебираются двойственно допустимые базисы (и соответствующие им псевдопланы прямой задачи линейного программирования). Каждый последующий псевдоплан в этой цепочке «не хуже» предыдущего в смысле убывания значений целевой функции (которые, как известно из теории двойственности (см. [1], теорема 13.1), являются оценками сверху для оптимального значения целевой функции). Благодаря таким различиям в этих методах, метод последовательного улучшения плана часто называют ***прямым методом***, а метод последовательного уточнения оценок – ***двойственным***.

Начнем с получения некоторых предварительных результатов. Пусть выполняются все предположения о задаче линейного программирования, сделанные в начале этого раздела.

Пусть A_B – некоторый базис, $l \in B$, вектор $v = (v_1, ..., v_m) \in E^m$ является решением следующей системы линейных алгебраических уравнений:

$$\begin{aligned} \langle A_i, v \rangle &= 0, \quad \forall i \in B,\ i \neq l, \\ \langle A_l, v \rangle &= 1. \end{aligned} \tag{1}$$

Система (1), легко увидеть, имеет единственное решение v. Чтобы записать эту систему в векторно-матричном виде, положим $l = i_q$ (вектор A_l занимает в базисе позицию с номером q) и обозначим через e_q единичный орт пространства E^m. Тогда системy (1) запишем в виде:

$$vA_B = e_q. \tag{2}$$

Далее, пусть векторы x, y – опорные решения, соответствующие базису A_B. Построим следующее однопараметрическое семейство векторов $y(t) \in E^m$:

$$y(t) = y + tv, \quad \forall t \in R. \tag{3}$$

Установим некоторые свойства семейства векторов $y(t)$. Очевидно, что оно представляет из себя прямую в пространстве E^m, проходящую через опорное решение y в направлении вектора v.

<u>*Теорема 1.*</u> *Для любого базиса A_B справедливо*

равенство

$$\langle b, y(t)\rangle = \langle b, y\rangle + t\,x_l, \quad \forall t \in R, \qquad (4)$$

где x_l – l*-тая координата опорного решения* x.

Доказательство. Подставляя (3) и (2) в (4) и пользуясь определением опорного решения x, для любого $t \in R$ получаем

$$\langle b, y(t)\rangle = \langle b, y\rangle + t\langle b, v\rangle = \langle b, y\rangle + t\langle A_B x_B, v\rangle =$$
$$= \langle b, y\rangle + t\langle vA_B, x_B\rangle = \langle b, y\rangle + t\langle e_q, x_B\rangle = \langle b, y\rangle + t\,x_l.$$

Что и требовалось.

Далее нам потребуется множество

$$J_l^- = \left\{ j : j \notin B,\ \langle A_j, v\rangle < 0 \right\}. \qquad (5)$$

Теорема 2. *Пусть* A_B *– двойственно допустимый базис,* $l \in B$, $x_l < 0$ *и* $J_l^- = \varnothing$. *Тогда* $y(t) \in D$, $\forall t \geq 0$ *и* $\langle b, y(t)\rangle \to -\infty$ *при* $t \to +\infty$.

Доказательство. Для проверки включения $y(t) \in D$ нужно убедиться в выполнении всех ограничений двойственной задачи $y(t)A \geq c$. Действительно, из включения $y \in D$, определения вектора v и условия теоремы $J_l^- = \varnothing$ следует $y(t)A = yA + t(vA) \geq c + t \cdot 0 = c$.

Справедливость второго утверждения теоремы следует из теоремы 1 (равенство (4)) и отрицательности x_l. Теорема доказана.

Результат, полученный в этой теореме, можно пояснить следующим образом. Полупрямая $y(t), \ \forall t \geq 0$, представляет собой луч с вершиной в точке y. В условиях теоремы вектор y является также и вершиной (опорным планом) многогранного множества D, а луч $y(t), \ \forall t \geq 0$, является неограниченным ребром этого множества. Следовательно, само множество допустимых решений двойственной задачи D неограниченно и, как видим из второго утверждения теоремы, целевая функция двойственной задачи линейного программирования $\langle b, y \rangle$ неограниченна снизу на D. Таким образом, теорема 2 дает набор признаков неразрешимости задач линейного программирования по причине неограниченности снизу целевой функции двойственной задачи на допустимой области. Напомним, что при этом прямая задача также не имеет решения, так как ее допустимая область пуста.

<u>*Теорема 3.*</u> *Пусть A_B – двойственно допустимый базис $l \in B$ и $J_l^- \neq \varnothing$. Тогда $y(t) \in D$, $\forall t \in \left[0, \, t_l\right]$, где*

$$t_l = \min_{j \in J_l^-} \left\{ \frac{-\Delta_j}{\langle A_j, v \rangle} \right\}. \tag{6}$$

<u>*Доказательство.*</u> Как видим, здесь множество J_l^-, в отличие от условий теоремы 2, не пусто.

Поэтому включение $y(t) \in D$ имеет место только для тех значений параметра t, которые удовлетворяют системе

$$\langle A_j,\ y(t)\rangle = \langle A_j,\ y\rangle + t\langle A_j,\ v\rangle \ \geq\ c_j,\ \forall j \in J_l^-. \quad (7)$$

Разрешив неравенства (7) относительно параметра t, получим $t \leq \dfrac{c_j - \langle A_j, y\rangle}{\langle A_j, v\rangle}$, или, что то же самое, $t \leq \dfrac{\Delta_j}{\langle A_j, v\rangle}$, для всех $j \in J_l^-$. Откуда и вытекает справедливость утверждения теоремы.

Обсудим результат, полученный в этой теореме. Множество точек $y(t)$ прямой (3), принадлежащих области D, здесь представляет из себя отрезок, который является ребром многогранного множества D. Опорный план y – одна из вершин этого ребра. Вычислим вторую вершину $\tilde{y}$, смежную с y по отношению к данному ребру. Для этого вычислим $\tilde{y} = y(t_l)$. Воспользуемся формулой (6) для максимально возможного значения параметра t в условиях теоремы 3. Пусть в (6) минимум достигается при $j = k$. Тогда $t_l = \dfrac{-\Delta_k}{\langle A_k, v\rangle}$. Подставим определенное таким образом значение t_l в формулу (3). Получим

$$\tilde{y} = y - \frac{\Delta_k}{\langle A_k, v\rangle} v. \qquad (8)$$

В силу построения, вектор $\tilde{y}$ допустим для двойственной задачи. Следовательно, он удовлетворяет всем ограничениям (неравенствам) двойственной задачи. Подставим (8) в k-тое неравенство

$$\langle A_k, \tilde{y}\rangle = \langle A_k, y\rangle - \frac{\Delta_k}{\langle A_k, v\rangle}\langle A_k, v\rangle = \langle A_k, y\rangle - \Delta_k = c_k.$$

Таким образом, вектор $\tilde{y}$ удовлетворяет k-му ограничению как равенству. Поэтому мы можем предположить, что вектор A_k будет базисным для $\tilde{y}$.

Подставим теперь вектор (8) в l-тое неравенство

$$\langle A_l, \tilde{y}\rangle = \langle A_l, y\rangle - \frac{\Delta_k}{\langle A_k, v\rangle}\langle A_l, v\rangle = \langle A_l, y\rangle - \frac{\Delta_k}{\langle A_k, v\rangle} \geq c_l.$$

Так как это ограничение может выполняться как строгое неравенство, то вектор A_l может не быть базисным для опорного решения $\tilde{y}$.

Исходя из полученного, можно предположить, что найденный вектор $\tilde{y}$ – вторая вершина построенного ребра, то есть опорный план двойственной задачи, соответствующий подсистеме $A_{\tilde{B}}$, где $\tilde{B} = \left(B \setminus \{l\}\right) \cup \{k\}$. Это предположение будет справедливо, если подсистема $A_{\tilde{B}}$ линейно независима, то есть является базисом. Этот факт устанавливается в следующей теореме.

Теорема 4. *Вектор* $\tilde{y}$*, определяемый формулой (8), является опорным планом двойственной задачи линейного программирования, соответствующим базису* $A_{\tilde{B}}$*, где* $\tilde{B} = \left(B \setminus \{l\}\right) \bigcup \{k\}$.

Заметим, что на основании равенства (8) имеем $\langle b, \tilde{y} \rangle = \langle b, y \rangle - \dfrac{\Delta_k}{\langle A_k, v \rangle} x_l$. Откуда можно сделать вывод, что при $x_l < 0$ выполняется неравенство $\langle b, \tilde{y} \rangle \le \langle b, y \rangle$. Если же вдобавок $\Delta_k > 0$, то $\langle b, \tilde{y} \rangle < \langle b, y \rangle$, что обеспечит уменьшение значения целевой функции при переходе из вершины y в вершину $\tilde{y}$.

Полученные результаты позволяют сформулировать общую (принципиальную) схему метода последовательного уточнения оценок. Метод состоит из последовательного выполнения однотипных итераций. Каждая итерация осуществляет переход от текущего двойственно допустимого базиса к новому. Опишем в общем виде одну итерацию. Предположим, что имеется некоторый двойственно допустимый базис A_B и соответствующий ему опорный план y

двойственной задачи. Далее выполняются следующие действия:

1. Нахождение опорного решения (псевдоплана) x, соответствующего базису A_B. Для этого решается система линейных алгебраических уравнений $A_B x_B = b$ и полагается $x = (x_B, 0)$.

2. Проверка признака оптимальности. На

основании критерия оптимальности опорных решений (см. теорему 1.2) необходимо проверить допустимость базиса, то есть проверить выполнение неравенств

$$x_i \geq 0, \quad \forall i \in B. \tag{9}$$

Возможны два случая.

a) Все неравенства (9) выполняются. Тогда векторы x и y являются решениями, соответственно, прямой и двойственной задач линейного программирования. На этом работа метода прекращается.

b) Существуют значения $i \in B$, для которых $x_i < 0$. Переходим к следующему этапу.

3. Выбор $l \in B$. Положим $l: x_l = \min_{i \in B} \{x_i\}$. (Очевидно, что $x_l < 0$.)

4. Вычисление вектора v. Для этого решается система линейных алгебраических уравнений $vA_B = e_q$.

5. Построение множества значений индекса j: $J_l^- = \left\{ j : j \notin B,\ \left\langle A_j, v \right\rangle < 0 \right\}$.

6. Проверка пустоты J_l^-. Возможны два случая.

a) $J_l^- = \varnothing$. Тогда, как следует из теоремы 2, прямая задача не имеет решения в силу пустоты допустимого множества, а двойственная задача не имеет решения в силу неограниченности снизу целевой функции на допустимом множестве. На этом работа метода прекращается.

b) $J_l^- \neq \varnothing$. Переходим к следующему этапу.

7. Выбор $k \notin B$. Положим

$$k : \frac{-\Delta_k}{\left\langle A_k, v \right\rangle} = \min_{j \in J_l^-} \left\{ \frac{-\Delta_j}{\left\langle A_j, v \right\rangle} \right\}.$$

8. Построение нового базиса $A_{\tilde{B}}$, где

$\tilde{B} = \left(B \setminus \{l\} \right) \bigcup \{k\}$.

9. Пересчет обратной матрицы по формулам (7) и вычисление нового опорного плана $\tilde{x}$, соответствующего базису $A_{\tilde{B}}$, по формулам (2.6).

10. Замена базиса A_B на $A_{\tilde{B}}$ и матрицы A_B^{-1} на $A_{\tilde{B}}^{-1}$. Переход к следующей итерации, к первому ее этапу.

Метод обратной матрицы имеет ряд существенных преимуществ по сравнению с симплексным методом. Обратная матрица имеет меньшие размеры, чем симплексная таблица. Особенно это существенно, когда m значительно меньше n. На каждой итерации метода обратной матрицы в большей мере, чем в симплексном методе, используются данные исходной задачи линейного программирования. Поэтому в ходе большого количества итераций погрешности вычислений меньше накапливаются, что приводит к большей устойчивости этого алгоритма по сравнению с симплексным методом.

Наконец, отметим, что для отыскания исходного допустимого базиса и соответствующего ему опорного плана также возможно применение метода искусственного базиса в сочетании с методом обратной матрицы.

7. Метод последовательного уточнения оценок

Метод последовательного уточнения оценок основан, так же как и метод последовательного улучшения плана, на последовательном целенаправленном переборе базисов. Но в методе последовательного уточнения оценок перебираются двойственно допустимые базисы (и соответствующие им псевдопланы прямой задачи линейного программирования). Каждый последующий псевдоплан в этой цепочке «не хуже» предыдущего в смысле убывания значений целевой функции (которые, как известно из теории двойственности, являются оценками сверху для оптимального значения целевой функции). Благодаря таким различиям в этих методах, метод последовательного улучшения плана часто называют ***прямым методом***, а метод последовательного уточнения оценок – ***двойственным***.

Начнем с получения некоторых предварительных результатов. Пусть выполняются все пред-положения о задаче линейного программирования, сделанные в начале этого раздела.

Пусть A_B – некоторый базис, $l \in B$, вектор $v = (v_1, ..., v_m) \in E^m$ является решением следующей системы линейных алгебраических уравнений:

$$\begin{aligned} &\langle A_i, v \rangle = 0, \quad \forall i \in B,\ i \neq l, \\ &\langle A_l, v \rangle = 1. \end{aligned} \tag{1}$$

Система (1), легко увидеть, имеет единственное решение v. Чтобы записать эту систему в векторно-матричном виде, положим $l = i_q$ (вектор A_l занимает в базисе позицию с номером q) и обозначим через e_q единичный орт пространства E^m. Тогда систему (1) запишем в виде:

$$vA_B = e_q. \tag{2}$$

Далее, пусть векторы x, y – опорные решения, соответствующие базису A_B. Построим следующее однопараметрическое семейство векторов $y(t) \in E^m$:

$$y(t) = y + tv, \quad \forall t \in R. \tag{3}$$

Установим некоторые свойства семейства векторов $y(t)$. Очевидно, что оно представляет из себя прямую в пространстве E^m, проходящую через опорное решение y в направлении вектора v.

<u>*Теорема 1.*</u> *Для любого базиса A_B справедливо равенство*

$$\langle b, y(t)\rangle = \langle b, y\rangle + t\, x_l, \quad \forall t \in R, \tag{4}$$

где x_l – l-тая координата опорного решения x.

<u>*Доказательство.*</u> Подставляя (3) и (2) в (4) и пользуясь определением опорного решения x, для любого $t \in R$ получаем

$$\langle b, y(t)\rangle = \langle b, y\rangle + t\langle b, v\rangle = \langle b, y\rangle + t\langle A_B x_B, v\rangle =$$
$$= \langle b, y\rangle + t\langle vA_B, x_B\rangle = \langle b, y\rangle + t\langle e_q, x_B\rangle = \langle b, y\rangle + t\, x_l .$$

Что и требовалось.

Далее нам потребуется множество

$$J_l^- = \left\{ j : j \notin B,\ \langle A_j, v\rangle < 0 \right\}. \tag{5}$$

<u>*Теорема 2.*</u> *Пусть A_B – двойственно допустимый базис, $l \in B$, $x_l < 0$ и $J_l^- = \varnothing$. Тогда $y(t) \in D$, $\forall t \geq 0$ и $\langle b, y(t)\rangle \to -\infty$ при $t \to +\infty$.*

<u>*Доказательство*</u>. Для проверки включения $y(t) \in D$ нужно убедиться в выполнении всех ограничений двойственной задачи $y(t)A \geq c$. Действительно, из включения $y \in D$, определения

вектора v и условия теоремы $J_l^- = \varnothing$ следует $y(t)A = yA + t(vA) \ge c + t \cdot 0 = c$.

Справедливость второго утверждения теоремы следует из теоремы 1 (равенство (4)) и отрицательности x_l. Теорема доказана.

Результат, полученный в этой теореме, можно пояснить следующим образом. Полупрямая $y(t),\ \forall t \ge 0$, представляет собой луч с вершиной в точке y. В условиях теоремы вектор y является также и вершиной (опорным планом) многогранного множества D, а луч $y(t),\ \forall t \ge 0$, является неограниченным ребром этого множества. Следовательно, само множество допустимых решений двойственной задачи D неограниченно и, как видим из второго утверждения теоремы, целевая функция двойственной задачи линейного программирования $\langle b, y \rangle$ неограниченна снизу на D. Таким образом, теорема 2 дает набор признаков неразрешимости задач линейного программирования по причине неограниченности снизу целевой функции двойственной задачи на допустимой области. Напомним, что при этом прямая задача также не имеет решения, так как ее допустимая область пуста.

<u>*Теорема 3.*</u> *Пусть A_B – двойственно допустимый базис $l \in B$ и $J_l^- \ne \varnothing$. Тогда $y(t) \in D$, $\forall t \in [0,\ t_l]$, где*

$$t_l = \min_{j \in J_l^-} \left\{ \frac{-\Delta_j}{\langle A_j, v \rangle} \right\}. \qquad (6)$$

<u>*Доказательство.*</u> Как видим, здесь множество J_l^-, в отличие от условий теоремы 2, не пусто. Поэтому включение $y(t) \in D$ имеет место только для тех значений параметра t, которые удовлетворяют системе

$$\langle A_j,\, y(t) \rangle = \langle A_j,\, y \rangle + t \langle A_j,\, v \rangle \geq c_j,\ \forall j \in J_l^-. \qquad (7)$$

Разрешив неравенства (7) относительно параметра t, получим $t \leq \dfrac{c_j - \langle A_j, y \rangle}{\langle A_j, v \rangle}$, или, что то же самое, $t \leq \dfrac{\Delta_j}{\langle A_j, v \rangle}$, для всех $j \in J_l^-$. Откуда и вытекает справедливость утверждения теоремы.

Обсудим результат, полученный в этой теореме. Множество точек $y(t)$ прямой (3), принадлежащих области D, здесь представляет из себя отрезок, который является ребром многогранного множества D. Опорный план y – одна из вершин этого ребра. Вычислим вторую вершину $\tilde{y}$, смежную с y по отношению к данному ребру. Для этого вычислим $\tilde{y} = y(t_l)$. Воспользуемся формулой (6) для максимально возможного значения параметра t в условиях теоремы 3. Пусть в (6) минимум достигается при

$j=k$. Тогда $t_l = \dfrac{-\Delta_k}{\langle A_k, v\rangle}$. Подставим определенное таким образом значение t_l в формулу (3). Получим

$$\tilde{y} = y - \frac{\Delta_k}{\langle A_k, v\rangle} v. \tag{8}$$

В силу построения, вектор $\tilde{y}$ допустим для двойственной задачи. Следовательно, он удовле-

творяет всем ограничениям (неравенствам) двойственной задачи. Подставим (8) в k-тое неравенство

$$\langle A_k, \tilde{y}\rangle = \langle A_k, y\rangle - \frac{\Delta_k}{\langle A_k, v\rangle}\langle A_k, v\rangle = \langle A_k, y\rangle - \Delta_k = c_k.$$

Таким образом, вектор $\tilde{y}$ удовлетворяет k-му ограничению как равенству. Поэтому мы можем предположить, что вектор A_k будет базисным для $\tilde{y}$.

Подставим теперь вектор (8) в l-тое неравенство

$$\langle A_l, \tilde{y}\rangle = \langle A_l, y\rangle - \frac{\Delta_k}{\langle A_k, v\rangle}\langle A_l, v\rangle = \langle A_l, y\rangle - \frac{\Delta_k}{\langle A_k, v\rangle} \ge c_l.$$

Так как это ограничение может выполняться как строгое неравенство, то вектор A_l может не быть базисным для опорного решения $\tilde{y}$.

Исходя из полученного, можно предположить, что найденный вектор $\tilde{y}$ – вторая вершина построенного ребра, то есть опорный план двойственной задачи, соответствующий подсистеме $A_{\tilde{B}}$, где $\tilde{B} = \left(B \setminus \{l\}\right) \bigcup \{k\}$. Это предположение будет справедливо, если подсистема $A_{\tilde{B}}$ линейно независима, то есть является базисом. Этот факт устанавливается в следующей теореме.

<u>*Теорема 4.*</u> *Вектор $\tilde{y}$, определяемый формулой (8), является опорным планом двойственной задачи линейного программирования, соответствующим базису $A_{\tilde{B}}$, где $\tilde{B} = \left(B \setminus \{l\}\right) \bigcup \{k\}$.*

Заметим, что на основании равенства (8) имеем $\langle b, \tilde{y} \rangle = \langle b, y \rangle - \dfrac{\Delta_k}{\langle A_k, v \rangle} x_l$. Откуда можно сделать вывод, что при $x_l < 0$ выполняется неравенство $\langle b, \tilde{y} \rangle \le \langle b, y \rangle$. Если же вдобавок $\Delta_k > 0$, то $\langle b, \tilde{y} \rangle < \langle b, y \rangle$, что обеспечит уменьшение значения целевой функции при переходе из вершины y в вершину $\tilde{y}$.

Полученные результаты позволяют сформулировать общую (принципиальную) схему метода последовательного уточнения оценок. Метод состоит из последовательного выполнения однотипных итераций. Каждая итерация осуществляет переход от текущего двойственно допустимого базиса к новому. Опишем в общем виде одну итерацию. Предположим, что имеется некоторый двойственно допустимый базис A_B и соответствующий ему опорный план y двойственной задачи. Далее выполняются следующие действия:

3. Нахождение опорного решения (псевдоплана) x, соответствующего базису A_B. Для этого решается система линейных алгебраических уравнений $A_B x_B = b$ и полагается $x = (x_B, 0)$.

4. Проверка признака оптимальности. На основании критерия оптимальности опорных решений (см. теорему 1.2) необходимо проверить допустимость базиса, то есть проверить выполнение неравенств

$$x_i \geq 0, \quad \forall i \in B. \tag{9}$$

Возможны два случая.

a) Все неравенства (9) выполняются. Тогда векторы x и y являются решениями, соответственно, прямой и двойственной задач линейного программирования. На этом работа метода прекращается.

b) Существуют значения $i \in B$, для которых $x_i < 0$. Переходим к следующему этапу.

8. Выбор $l \in B$. Положим $l: x_l = \min_{i \in B}\{x_i\}$. (Очевидно, что $x_l < 0$.)

9. Вычисление вектора v. Для этого решается система линейных алгебраических уравнений $vA_B = e_q$.

10. Построение множества значений индекса j: $J_l^- = \left\{ j : j \notin B, \left\langle A_j, v \right\rangle < 0 \right\}$.

11. Проверка пустоты J_l^-. Возможны два случая.

a) $J_l^- = \varnothing$. Тогда, как следует из теоремы 2, прямая задача не имеет решения в силу пустоты допустимого множества, а двойственная задача не имеет решения в силу неограниченности снизу целевой функции на допустимом множестве. На этом работа метода прекращается.

b) $J_l^- \neq \varnothing$. Переходим к следующему этапу.

12. Выбор $k \notin B$. Положим

$$k : \frac{-\Delta_k}{\left\langle A_k, v \right\rangle} = \min_{j \in J_l^-} \left\{ \frac{-\Delta_j}{\left\langle A_j, v \right\rangle} \right\}.$$

13. Построение нового базиса $A_{\tilde{B}}$, где $\tilde{B} = \left(B \setminus \{l\} \right) \bigcup \{k\}$.

14. Вычисление опорного плана $\tilde{y}$, соответствующего базису $A_{\tilde{B}}$, по формуле (8).

15. Замена базиса A_B на базис $A_{\tilde{B}}$ и опорного плана y на опорный план $\tilde{y}$. Переход к следующей итерации, к первому ее этапу.

Заключительную часть этого параграфа посвятим обоснованию конечности этого метода.

Введем следующие понятия. Опорный план y двойственной задачи линейного программирования называется ***невырожденным***, если $\Delta_j > 0, \quad \forall j \notin B$. В противном случае опорный план называется ***вырожденным***. Легко увидеть, что невырожденному опорному плану соответствует только один базис. В отличие от этого, вырожденному опорному плану соответствует несколько базисов. Далее, двойственная задача линейного программирования называется ***невырожденной***, если все ее опорные планы невырождены. В противном случае двойственная задача называется ***вырожденной***.

<u>*Теорема 5.*</u> ***(Теорема о конечности метода последовательного уточнения оценок)*** *Пусть двойственная задача линейного программирования является невырожденной, имеется некоторый двойственно допустимый базис A_B и соответствующий ему опорный план y двойственной задачи. Тогда за конечное число итераций метода последовательного уточнения оценок либо будут найдены решения прямой и двойственной задач линейного программирования, либо будет установлена их неразрешимость.*

Справедливость этой теоремы следует из предположения о невырожденности двойственной задачи линейного программирования. В этом случае на каждой итерации $\Delta_k > 0$, что как было отмечено выше, гарантирует отрицательность приращения целевой функции на каждой итерации.

Следовательно, учитывая конечность числа опорных планов, получаем, что число итераций конечно. В вырожденном же случае необходимо использование специальных приемов.

8. Двойственный симплексный метод

Метод последовательного уточнения оценок является принципиальной схемой решения задач линейного программирования, основанной на целенаправленном переборе двойственно допустимых базисов. На основе этой схемы созданы различные численно реализуемые методы (алгорит-мы). Так же, как для метода последовательного улучшения плана, численно реализуемым алгоритмом является симплексный метод (или, как еще его называют, прямой симплексный метод), так и для метода последовательного уточнения оценок разработан численно реализуемый алгоритм – двойственный симплексный метод. Знакомству с ним и посвящен данный параграф. На каждой итерации этого метода также используется симплексная таблица, которая определяется и пересчитывается по тем же правилам, что и в прямом симплексном методе. Симплексная таблица содержит всю необходимую информацию для осуществления итерации метода последовательного уточнения оценок. Об этом речь в следующей теореме:

<u>*Теорема 1.*</u> *Пусть* A_B *– некоторый базис,* $l \in B$*, вектор* $v \in E^m$ *удовлетворяет системе уравнений (7.2), тогда справедливы равенства*

$$\left\langle A_j, v \right\rangle = g_{l,\,j}, \quad \forall j = 1, \ldots, n. \tag{1}$$

<u>*Доказательство.*</u> Воспользуемся разложением $A_j = \sum\limits_{i \in B} g_{i,\,j} A_i$. Для любого $j = 1, \ldots, n$ имеем

$$\left\langle A_j, v \right\rangle = \left\langle \sum_{i \in B} g_{i,\,j} A_i, v \right\rangle = \sum_{i \in B} g_{i,\,j} \left\langle A_i, v \right\rangle = g_{l,\,j}.$$

Что и требовалось.

Заметим, что из теоремы 1 следует, что множество (5) совпадает с $\left\{ j:\ j \notin B,\ g_{l,j} < 0 \right\}$.

Теперь можно сформулировать общую итерацию двойственного симплексного метода. Предпо-ложим, что имеется некоторый двойственно допустимый базис A_B. Далее выполняются следующие действия:

0. Построение исходной симплексной таблицы $G(B)$.

1. Проверка признака оптимальности

$$x_i \ge 0, \quad \forall i \in B. \tag{2}$$

Если все неравенства (2) выполняются, то x – решение задачи, и метод прекращает работу.

2. Выбор $l \in B$. Положим $l:\ \ x_l = \min\limits_{i \in B} \{x_i\}$. (Очевидно, что $x_l < 0$.)

3. Построение $J_l^- = \left\{ j:\ j \notin B,\ g_{l,j} < 0 \right\}$ (см. теорему 1).

4. Если множество $J_l^- = \varnothing$, то прямая задача не имеет решения в силу пустоты допустимого множества, а двойственная задача не имеет решения в силу неограниченности снизу целевой функции на допустимом множестве. На этом работа метода прекращается.

5. Выбор номера $k \notin B$. Положим

$$k:\ \frac{-\Delta_k}{g_{l,k}} = \min_{j \in J_l^-} \left\{ \frac{-\Delta_j}{g_{l,j}} \right\}.$$

6. Построение нового базиса $A_{\tilde{B}}$, где $\tilde{B} = \left(B \setminus \{l\} \right) \bigcup \{k\}$.

7. Пересчет симплексной таблицы по формулам (1) – (4) из параграфа 3.

8. Замена базиса A_B на базис $A_{\tilde{B}}$ и симплексной таблицы $G(B)$ на таблицу $G\left(\tilde{B}\right)$. Переход к следующей итерации, к первому ее этапу.

В заключение этого параграфа обсудим вопрос об отыскании исходного двойственно допустимого базиса. Для прямого метода, как мы знаем, исходный базис можно находить при помощи метода искусственного базиса. Для двойственного метода его применять нельзя, так как он разработан специально для отыскания допустимых базисов и соответствующих им опорных планов. Существуют специальные приемы, которые позволяют, решая некоторые вспомогательные задачи линейного программирования, отыскивать двойственно допустимые базисы и соответствующие им псевдопланы. Мы не будем знакомиться с этими приемами, так как двойственный симплексный метод непосредственно к решению задач линейного программирования на практике не применяется. Однако весьма эффективно совместное применение прямого и двойственного симплексного метода для решения задач линейного программирования в рамках разнообразных схем «отсечений». Под «отсечением» мы понимаем такое линейное ограничение, вводимое в условия задачи после получения ее решения, которое делает его недопустимым. При этом оптимальный базис становится недопустимым, но остается двойственно допустимым (таким образом, вопрос об его отыскании становится неактуальным). Различные отсечения вводятся, например, в процессе моделирования тех или иных проблем, в методах декомпозиции, при-меняемых при решении задач линейного программирования с большим числом ограничений, при решении вспомогательных задач линейного программирования в методах целочисленного программирования, нелинейного программирования и в некоторых других случаях.

9. Транспортная задача

В предыдущих параграфах мы изучили универсальные методы решения задач, то есть позволяющие решить любую задачу линейного программирования. Однако в практике моделирования часто встречаются задачи, имеющие определенные особенности. Эти особенности позволяют построить численные реализации, более эффективные, чем общие схемы. Это удобно продемонстрировать на примере так называемой ***транспортной задачи,*** которая является одной из многих специальных задач линейного программирования.

Наиболее распространена следующая содержательная постановка транспортной задачи. Пусть имеется некоторый однородный продукт в количестве Q единиц. Запасы продукта находятся в

m ***пунктах отправления*** в количествах a_i единиц, $i=1,\ldots,m$. Продукт нужно доставить в n ***пунктов назначения*** в количествах b_j, $j=1,\ldots,n$. Известны ***тарифы*** c_{ij}, то есть стоимости перевозки единицы груза из пункта отправления i в пункт назначения j. Необходимо спланировать перевозки, то есть определить количества x_{ij} перевозимого груза из пункта i в пункт j, исходя из необходимости удовлетворения спроса и минимизации суммарных затрат на перевозки. Предполагая, что

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = Q, \tag{1}$$

получим следующую модель:

$$\begin{aligned} &\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{i,\,j} x_{i,\,j} \\ &\sum_{j=1}^{n} x_{i,\,j} = a_i, \quad \forall i = 1, \ldots, m, \\ &\sum_{i=1}^{m} x_{i,\,j} = b_j, \quad \forall j = 1, \ldots, n, \\ &x_{i,\,j} \geq 0, \quad \forall i, j. \end{aligned} \tag{2}$$

Задачу линейного программирования (2), полученную в предположении (1), называют ***замкнутой транспортной задачей***. Согласно общей терминологии, (2) является задачей линейного программирования, записанной в канонической форме. Характерно, что матрица A коэффициентов

ограничений задачи (2) имеет специфичную структуру – все её коэффициенты равны нулю или единице, причем каждый столбец матрицы A содержит всего по две единицы. Удобно нумеровать столбцы матрицы A двумя индексами (i, j). Так, в столбце $A_{i,\,j}$ единичными являются компоненты с номерами i и $m + j$.

В силу специфики транспортной задачи применение к ней алгоритмов, описанных ранее, неэффективно. Один из первых алгоритмов, специально разработанных для этой задачи, (***метод потенциалов***) является алгоритмом метода последовательного улучшения плана.

Легко увидеть, что транспортная задача (2) имеет решение. Действительно, план перевозок с компонентами $x_{i,\,j} = \dfrac{a_i\, b_j}{Q}, \quad \forall i, j$, удовлетворяет всем ограничениям задачи (2). Таким образом, допустимое множество D задачи не пусто. Так как $0 \le x_{i,\,j} \le \min\left\{a_i, b_j\right\}, \quad \forall i, j$, множество D ограничено. Следовательно, задача (2) имеет решение.

Далее заметим, что если сложить все уравнения системы ограничений задачи (2) и вычесть сумму любых $(n+m-1)$-го уравнений этой системы, то получим оставшееся уравнение. Это означает, что система содержит, по крайней мере, одно избыточное уравнение. Более того, можно показать, что число избыточных уравнений равно единице. Таким образом, ранг матрицы коэффициентов системы уравнений задачи (2) равен $n+m-1$, поэтому в системе ограничений задачи одно (любое) уравнение избыточно.

Задачу, двойственную к транспортной, удобно записать в виде

$$\max \sum_{i=1}^{m} a_i u_i + \sum_{j=1}^{n} b_j v_j$$

$$u_i + \quad v_j \le c_{i,j}, \qquad (3)$$

$$\forall i = 1, \ldots, m, \quad j = 1, \ldots, n.$$

где двойственные переменные $u_i, i = 1, \ldots, m$, принято называть ***потенциалами пунктов отправления***, а переменные v_j, $j = 1, \ldots, n$, – ***потенциалами пунктов назначения***. Заметим, что поскольку прямая задача содержит одно избыточное уравнение, в двойственной задаче имеется одна «лишняя» переменная.

Транспортную задачу, как любую задачу линейного программирования, можно решать универсальными методами, например, симплексным. Однако это неэффективно в силу специфики транспортной задачи – больших размеров симплекснойтаблицы и слабой ее заполненности. Заметим

также, что и для отыскания начального опорного плана перевозок и его базиса при использовании универсальных средств потребовалось бы введение искусственного базиса, а это еще заметнее увеличило бы размеры решаемой задачи. Итак, необходимы специальные приемы и методы решения транспортной задачи.

Среди множества алгоритмов, разработанных для этой цели, нам будет интересна реализация метода последовательного улучшения плана применительно к транспортной задаче, называемая ***методом потенциалов***.

Как известно, для работы метода необходимо наличие начального допустимого базиса. Наиболее простые и используемые способы построения начального опорного плана перевозок и его базиса – это правила ***«северо-западного угла»*** и ***«минимальной цены»***. Исходный допустимый базис находится однозначно, если задача (2) невырождена. В противном случае следует использовать так называемый ε***-прием***.

Далее согласно общей схеме метода требуется отыскать опорное решение двойственной задачи, соответствующее текущему базису. Для транс-портной задачи это означает отыскание потенциалов путем решения следующей системы ли-
нейных алгебраических уравнений

$$u_i + v_j = c_{i,j},\ \forall (i,j) \in B, \tag{4}$$

где B – множество номеров базисных векторов. Левые части уравнений (4) содержат только два слагаемых с коэффициентами, равными единице. Учитывая, что ранг системы (4) равен $n+m-1$, значение одного любого потенциала можно зафиксировать произвольно в любом из уравнений. После чего систему (4) можно преобразовать к треугольному виду и легко найти значения всех потенциалов.

Вычисленные значения потенциалов позволят, следуя общей схеме метода последовательного улучшения плана, проверить двойственную допустимость базиса (а значит, его оптимальность). Для этого найдем невязки небазисных неравенств системы ограничений двойственной задачи по формулам $\Delta_{i,j} = u_i + v_j - c_{i,j},\ \forall (i, j) \notin B$. Заметим, что признаком оптимальности для транспортной задачи (задачи минимизации) является неположительность всех $\Delta_{i,j}$. Если же существует пара (k, p) такая, что $\Delta_{k,p} > 0$, то вектор $A_{k,p}$ (небазисный) вводится в новый базис. При этом, как известно из общей теории метода, увеличение значения соответствующей переменной $x_{k,p}$ может привести к уменьшению значения целевой функции. Пусть $x_{k,p} = t$, где $t \geq 0$. Тогда, как известно из теории метода последовательного улучшения плана (см. теорему 2.2), приращение целевой функции задачи (2) равно $-t\Delta_{k,p}$. Найдем максимально возможное значение t, сохраняющее допустимость плана перевозок. Увеличение объема пере-возки $x_{k,p}$ приведет к изменению объемов некоторых других перевозок (бывших прежде базисными). Разобьем множество B на 3 подмножества

$$B^{+} = \left\{ (i, j) : (i, j) \in B,\ \tilde{x}_{i,j} = x_{i,j} + t \right\},$$

$$B^{-} = \left\{ (i, j) : (i, j) \in B,\ \tilde{x}_{i,j} = x_{i,j} - t \right\},$$

$$B^{0} = \left\{ (i, j) : (i, j) \in B,\ \tilde{x}_{i,j} = x_{i,j} \right\}.$$

Очевидно, что для обеспечения допустимости нового базиса нужно обеспечить выполнение условия $x_{i,j} - t \geq 0,\ \forall (i, j) \in B^-$, откуда $t = \min\limits_{(i,\, j) \in B^-} x_{i,j}$. Одну из пар $(i, j) \in B^-$, для которых здесь достигается минимум, обозначим (l, q). Вектор $A_{l,q}$ покинет базис.

Напомним, что метод потенциалов предназначен для решения замкнутых транспортных задач. Если же равенство (1) не выполняется (что, как правило, и имеет место в реальных ситуациях), то транспортная задача называется ***незамкнутой***. Возможны два варианта нарушения этого равенства.

Рассмотрим сначала случай $\sum\limits_{i=1}^{m} a_i < \sum\limits_{j=1}^{n} b_j$. Это означает, что общего количества груза не хватает для удовлетворения всех потребностей в нем и, следовательно, в некоторые пункты доставки будут завезены меньшие, чем требуются, количест-
ва груза. При этом, конечно, весь груз будет вывезен из всех пунктов отправки. Эти соображения приводят к следующей модели транспортной задачи:

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

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

$$\sum_{i=1}^{m} x_{i,\,j} \le b_j, \quad j = 1,\dots,n, \qquad (5)$$

$$x_{i,\,j} \ge 0, \quad \forall i, j.$$

Опишем прием, позволяющий привести задачу (5) к эквивалентной ей замкнутой задаче. Введем $(m+1)$-ый (***фиктивный***) пункт отправления груза, пусть $a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i$. Введем ***фиктивные переменные*** $x_{m+1,\,j}$, $j = 1,\dots,n$. Потребуем, чтобы $\sum_{j=1}^{n} x_{m+1,\,j} = a_{m+1}$. Положим $c_{m+1,\,j} = 0$, $j = 1,\dots,n$. Задача (5) примет следующий вид:

$$\min \sum_{i=1}^{m+1}\sum_{j=1}^{n} c_{i,\,j}x_{i,\,j}$$

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

$$\sum_{i=1}^{m+1} x_{i,\,j} = b_j, \quad j = 1,\dots,n, \qquad (6)$$

$$x_{i,\,j} \ge 0, \quad \forall i, j.$$

Задача (6) – замкнутая, и к ней применимы все приемы и методы этого параграфа. Заметим, что значения фиктивных переменных в решении этой задачи – это объемы груза недопоставленного в соответствующие пункты назначения.

Пусть теперь $\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$. Это означает, что общее количество груза превышает суммарную потребность в нем и, следовательно, во все пункты доставки будут завезены необходимые количества груза. При этом, конечно, часть груза останется в некоторых пунктах отправки. Получаем следующую модель:

$$
\begin{aligned}
&\min \sum_{i=1}^{m}\sum_{j=1}^{n} c_{i,\,j} x_{i,\,j} \\
&\sum_{j=1}^{n} x_{i,\,j} \le a_i, \quad i \le 1,\ldots,m, \\
&\sum_{i=1}^{m} x_{i,\,j} = b_j, \quad j = 1,\ldots,n, \\
&x_{i,\,j} \ge 0, \quad \forall i, j.
\end{aligned}
\tag{7}
$$

Ее сведение к замкнутой задаче осуществляется аналогично предыдущему случаю путем введения $(n+1)$-го фиктивного пункта доставки груза с потребностью $b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j$. Введем фиктивные переменные $x_{i,\,n+1}$, $i = 1,\ldots,m$. Потребуем, чтобы $\sum_{i=1}^{m} x_{i,\,n+1} = b_{n+1}$. Положим $c_{i,\,n+1} = 0$, $i = 1,\ldots,m$. Тогда задача (7) примет вид:

$$\min \sum_{i=1}^{m}\sum_{j=1}^{n+1} c_{i,\,j}x_{i,\,j}$$

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

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

$$x_{i,\,j} \ge 0, \quad \forall i, j.$$

Значения фиктивных переменных в решении этой задачи – это объемы груза, оставшегося в соответствующих пунктах отправления.

Методы решения нелинейных экстремальных задач

1. Основные определения

Как правило, известные методы нахождения экстремума нелинейной функции не позволяют решить задачу точно. В большинстве своем, они являются ***методами последовательных приближений***. Эти методы вырабатывают последовательности $\left\{x_k\right\}_{k=0,1,2,\dots} \subset E^n$, которые в каком-либо смысле приближаются к искомому решению.

<u>*Определение 1.*</u> *Последовательность приближений* $\left\{x_k\right\}_{k=0,1,2,\dots} \subset E^n$ *называется*

минимизирующей, *если* $\lim\limits_{k\to\infty} f\left(x_k\right) = f^*$.

Знание какой-либо минимизирующей последовательности позволяет найти с любой точностью значение f^*. Заметим, что минимизирующая последовательность не обязательно сходится к x^*.

<u>*Определение 2.*</u> *Последовательность приближений* $\left\{x_k\right\}_{k=0,1,2,\dots} \subset E^n$ *называется* ***сильно сходящейся (сходящейся по норме)***, *если* $\lim\limits_{k\to\infty} x_k = x^*$.

В последнем случае появляется возможность вычисления значения x^* с любой точностью.

Так как на практике невозможно бесконечно долго генерировать последовательности приближений, то весьма важным является вопрос о выборе критерия останова, который гарантировал бы до-стижение необходимых точностей в получении приближенных решений. Единых универсальных рецептов выбора таких критериев нет. Однако в каждом конкретном случае, учитывая специфику решаемой задачи, привлекая какую-либо дополни-
тельную информацию о ней, удается подобрать нужный критерий.

<u>*Определение 3.*</u> *Последовательность приближений* $\left\{x_k\right\}_{k=0}$, $\subset E_1^n$ *называется* ***релаксационной***, *если выполняются неравенства*

$$f\left(x_{k+1}\right) < f\left(x_k\right), \quad \forall k = 0, 1, 2, \ldots$$

Большинство методов последовательных приближений базируется на следующей общей схеме. Выбирается, до некоторой степени произвольно, начальное приближение $x_0 \in E^n$. Последующие приближения строятся по рекуррентному правилу

$$x_{k+1} = x_k + t_k s_k, \quad \forall k = 0, 1, 2, \ldots, \tag{1}$$

где векторы $s_k \in E^n$ – ***направления итерационного перехода***, $t_k \in R$ – ***шаговые множители***. В зависимости от применяемого способа выбора s_k и t_k получаются различные методы построения последовательностей $\{x_k\}$. Многие (хотя и не все) методы используют в качестве направлений s_k релаксационные направления для функции f в точках x_k, а для задач условной минимизации – условно релаксационные направления для f в точках $x_k \in D$ относительно допустимой области D.

В настоящее время известно большое количество разнообразных методов минимизации. Эти методы классифицируются по различным признакам. Одной из важнейших характеристик методов последовательных приближений является так называемый ***порядок метода***. Будем определять его по порядку частных производных функции f

, используемых для построения последовательности $\{x_k\}$. Таким образом, если используются первые частные производные функции f, то это – метод ***первого порядка***, в случае использования вторых частных производных – метод ***второго порядка***, и так далее. Если же производные не используются вообще, а вычисляются только лишь значения $f(x_k)$, то говорят о методах ***нулевого порядка*** (их еще называют ***методами прямого поиска***).

Наконец, обсудим способы выбора шагового множителя $t_k \in R$. Существуют различные приемы регулировки шага. Их выбор определяется многими факторами: свойствами минимизируемой функции и допустимого множества, способом нахождения направлений итерационных переходов, необходимой точностью решения и так далее. Часто в различных методах используется так называемый ***полный шаг***. Определяется он при помощи одномерной минимизации

$$f(x_{k+1}) = f(x_k + t_k s_k) = \min_{t \in [a_k, b_k]} f(x_k + t\, s_k).$$

Здесь множество $[a_k, b_k]$ в различных методах может быть отрезком, лучом, либо всей числовой прямой R. Методы отыскания минимума функции одной переменной мы обсудим позже.

2. Методы безусловной минимизации

2.1. Методы покоординатного спуска

Значительную часть методов прямого поиска можно уложить в следующую схему. Пусть в пространстве E^n выбран какой-либо базис $\{h_1, h_2, ..., h_n\}$. Последовательность приближений строится по формуле (1.1), где $x_0 \in E^n$ выбирается произвольно, а в качестве $s_k \in E^n$ выбирается вектор $h_i = h_{i(k)}$. Шаговый множитель $t_k \in R$ может регулироваться разнообразными способами.

Самыми простыми из методов этой группы являются ***методы покоординатного спуска***. В этих методах в качестве базиса выбирается система ортов $\{e_1, e_2, ..., e_n\}$ пространства E^n. Таким образом, на каждой итерации изменяется только одна координата вектора x_k, что отражено и в названии этих методов. Действительно, в покоординатной записи формулы для построения последовательности точек x_k при $s_k = e_i$ имеют вид

$$x_j^{k+1} = x_j^k \quad \forall j = 1, ..., n, \ \ j \neq i,$$
$$x_i^{k+1} = x_i^k + t_k.$$

Здесь x_j^k – j-тая координата вектора x_k.

Порядок чередования базисных векторов в ходе осуществления итераций может быть различным. В зависимости от способа чередования векторов e_i получаются различные реализации покоординатного метода. Укажем некоторые из них.

Метод циклического покоординатного спуска. В этом варианте метода на каждой итерации k номер $j=j(k)$ изменяемой координаты выбирается по правилу $j=k(\mathrm{mod}\,n)+1, \quad \forall k=0,1,2,\ldots$, то есть циклически перебираются все координаты по очереди с первой до последней.

Метод случайного покоординатного спуска. В этом варианте метода номер изменяемой координаты выбирается из целых чисел от 1 до n случайным образом с равной вероятностью. При реализации метода используют генератор псевдослучайных чисел.

Метод «быстрой переменной». В этом варианте метода покоординатного спуска на каждой итерации выбирается координатное направление с наибольшей скоростью изменения функции в окрестности точки x_k. Найти такое направление можно, вычисляя все частные производные функции f в точке x_k и выбирая из них наибольшую по модулю:

$$\left| \frac{\partial f\left(x_k\right)}{x_{j(k)}} \right| = \max_{j=1,\ldots,n} \left| \frac{\partial f\left(x_k\right)}{x_j} \right|.$$

Однако, как правило, затраты на выбор «быстрой переменной» не оправданы, так как заметного улучшения работы метода не наблюдается.

Как говорилось выше, существуют различные способы регулировки шага в методах покоординатного спуска. Чаще всего используется полный шаг, который, как мы определили в предыдущем параграфе, находится при помощи одномерной минимизации

$$f\left(x_{k+1}\right) = f\left(x_k + t_k e_j\right) = \min_{t\in R} f\left(x_k + t e_j\right).$$

В заключение параграфа отметим, что существует большое количество различных модификаций покоординатных методов. Здесь можно выделить три направления: первое из них связано с различными способами регулировки шагового множителя, второе – с порядком выбора координатного направления итерационного перехода, и, наконец, третье направление предполагает использование базисов $\left\{h_1, h_2, \ldots, h_n\right\}$ отличных от $\left\{e_1, e_2, \ldots, e_n\right\}$. Большинство таких модификаций но-сит характер эвристических приемов, которые для некоторых классов задач позволяют заметно повысить эффективность методов. Наилучших результатов можно добиться в рамках третьего направления модификаций. Здесь также можно выделить две группы методов. Одна из них предпо-лагает переход к новому базису после каждого цикла (состоящего в поочередном использовании n базисных координат). Другая группа методов основана на адаптации базиса $\{h_1, h_2, ..., h_n\}$ к свойствам минимизируемой функции. ***Методы сопряженных направлений***, с которыми мы познакомимся в следующем параграфе, относятся к этой группе методов.

2.2. Методы сопряженных направлений

В этом параграфе будем считать, что дана симметричная положительно определенная матрица A размерности $n \times n$. Введем следующие понятия.

*<u>Определение 1.</u> Ненулевые векторы h и g называются **сопряженными** относительно матрицы A (A-**сопряженными**), если выполняется равенство $\langle Ag, h \rangle = 0$.*

Например, ортогональные векторы сопряжены относительно единичной матрицы.

<u>Определение 2.</u> Систему векторов $\{h_i\}_{i=1}^{m}$ называют сопряженной относительно матрицы A, если любые два различных вектора в ней являются A-сопряженными.

<u>*Теорема 1.*</u> *Любая система векторов* $\{ h_i \}_{i=1}^{m}$ *сопряженная относительно матрицы* A *является линейно независимой.*

<u>*Доказательство.*</u> Пусть числа $\{ \alpha_i \}_{i=1}^{m}$ таковы что

$$\sum_{i=1}^{m} \alpha_i h_i = 0 . \tag{1}$$

Выберем произвольный номер $k \in \{1, ..., m\}$ и умножим равенство (1) на вектор Ah_k. Получим $\sum_{i=1}^{m} \alpha_i \langle Ah_k, h_i \rangle = 0$. Отсюда в силу попарной сопряженности векторов системы $\{h_i\}_{i=1}^{m}$ имеем $\alpha_k \langle Ah_k, h_k \rangle = 0,\ \forall k = 1, ..., m$. Следовательно, в силу положительной определенности матрицы A, $\alpha_k = 0,\ \forall k = 1, ..., m$. Откуда и следует линейная независимость системы векторов $\{h_i\}_{i=1}^{m}$.

Заметим, что из теоремы 1 следует, что система $\{h_i\}_{i=1}^{m}$ содержит не более n векторов. Поэтому если $m = n$, то система $\{h_i\}_{i=1}^{n}$ является базисом пространства E^n.

Пусть дана задача безусловной минимизации квадратичной функции $f(x) = \frac{1}{2}\langle Ax, x\rangle + \langle b, x\rangle + c$, пусть x^* – минимум функции f. В силу

положительной определенности матрицы A функция f
строго выпукла, а значит, минимум единственен. Согласно необходимым и достаточным условиям минимума выпуклой функции f имеем $f'\left(x^*\right)=0$, то есть

$$Ax^* + b = 0. \qquad (2)$$

Пусть система векторов $\left\{h_i\right\}_{i=1}^n$ сопряжена относительно матрицы A. Тогда согласно теореме 1 она образует базис в E^n. Значит, для любой фиксированной точки $x_0 \in E^n$ существуют числа $\alpha_i^*,\ i=1,\ldots,n,$ такие что

$$x^* = x_0 + \sum_{i=1}^n \alpha_i^* h_i\,. \qquad (3)$$

Отсюда и из (2)

$$\left(Ax_0 + b\right) + \sum_{i=1}^n \alpha_i^* Ah_i = 0. \qquad (4).$$

Выберем произвольный номер $k \in \left\{1,\ldots,n\right\}$, умножим (4) на вектор h_k и с учетом попарной сопряженности векторов системы $\left\{h_i\right\}_{i=1}^n$ получим $\left\langle Ax_0 + b, h_k\right\rangle + \alpha_k^*\left\langle Ah_k, h_k\right\rangle = 0$, откуда

$$\alpha_k^* = -\frac{\left\langle Ax_0 + b, h_k\right\rangle}{\left\langle Ah_k, h_k\right\rangle},\ \forall k = 1,\ldots,n\,. \qquad (5)$$

Таким образом, с помощью базиса $\{h_i\}_{i=1}^{n}$, состоящего из векторов, сопряженных относи-

тельно матрицы A, вектор x^* может быть найден по явным формулам (5).

Найдем формулу для вычисления полного шага для функции f из точки x по направлению h. Для этого потребуем, чтобы производная в точке $x + th$ по направлению h равнялась нулю: $f_t'(x + th) = \langle f'(x + th), h \rangle = 0$. Тогда имеем $\langle A(x + th) + b, h \rangle = 0$. Решая это уравнение относительно переменной t, получим

$$\tilde{t} = -\frac{\langle Ax + b, h \rangle}{\langle Ah, h \rangle}. \tag{6}$$

Сформулируем ***метод сопряжённых направлений*** для минимизации квадратичной функции $f(x) = \frac{1}{2}\langle Ax, x \rangle + \langle b, x \rangle + c$, где матрица A положительно определена.

Пусть известны система $\{h_i\}_{i=1}^{n}$ векторов взаимосопряжённых относительно матрицы A и произвольная точка x_0 из E^n.

Пусть найдена точка x_k, $k = 0, 1, \ldots$ Отыщем вектор $x_{k+1} = x_k + t_{k+1} h_{k+1}$, где t_{k+1} – полный шаг. Так как для квадратичной функции полный шаг вычисляется по формуле (6), в частности, имеем $x_1 = x_0 - \dfrac{\left\langle Ax_0 + b,\ h_1 \right\rangle}{\left\langle Ah_1,\ h_1 \right\rangle} h_1 = x_0 + \alpha_1^* h_1$. Далее,

$$x_2 = x_1 - \frac{\left\langle Ax_1 + b,\ h_2 \right\rangle}{\left\langle Ah_2,\ h_2 \right\rangle} h_2 = x_1 + \alpha_2^* h_2 .$$

Откуда, учитывая формулу для x_1, получаем

$$x_2 = x_0 + \alpha_1^* h_1 - \frac{\left\langle A\left(x_0 + \alpha_1^* h_1\right) + b,\ h_2 \right\rangle}{\left\langle Ah_2,\ h_2 \right\rangle} h_2 .$$

С учётом сопряженности векторов h_1 и h_2, получаем $x_2 = x_0 + \alpha_1^* h_1 + \alpha_2^* h_2$. Далее методом математической индукции нетрудно получить, что $x_{k+1} = x_0 + \sum\limits_{i=0}^{k} \alpha_{i+1}^* h_{i+1}$ для любого $k = 0, 1, ..., n-1$.

Отсюда при $k = n-1$ получаем $x_n = x_0 + \sum\limits_{i=0}^{n-1} \alpha_{i+1}^* h_{i+1}$.

Сравнивая полученную формулу с (3), делаем вывод, что $x_n = x^*$.

Итак, метод сопряженных направлений позволяет найти минимум квадратичной функции не более, чем за n итераций.

В рамках изложенной общей схемы метода сопряженных направлений существуют численно реализуемые алгоритмы, различающиеся способом построения системы векторов $\{h_i\}_{i=1}^{n}$ взаимосопряженных относительно матрицы A. Опишем некоторые способы построения системы векторов взаимосопряженных относительно заданной мат-рицы A.

Во-первых. Пусть $h_1 \in E^n$ – произвольный вектор. Допустим, что уже построены векторы $h_1, h_2, ..., h_k \in E^n$, $1 \le k \le n-1$, попарно сопряженные относительно матрицы A. Вектор h_{k+1} определяется как ненулевое частное решение однородной системы линейных алгебраических уравнений

$$\left\langle Ah_i, h_{k+1} \right\rangle = 0, \;\; i = 1, ..., k.$$

Этот способ, несмотря на кажущуюся простоту, на практике не используется в силу больших вычислительных затрат, так как для его реализации требуется решить $n-1$ систему с возрастающим числом уравнений.

Другой подход заключается в следующем. Пусть $\{p_1, p_2, ..., p_n\}$ – некоторая линейно независимая система векторов из E^n. Полагаем $h_1 = p_1$. Допустим, что уже построены векторы $h_1, h_2, ..., h_k \in E^n$, $1 \le k \le n-1$, попарно сопряженные относительно матрицы A. Вектор h_{k+1} находится как линейная комбинация

$h_{k+1} = p_{k+1} + \sum_{i=1}^{k} \alpha_{k+1,\,i} h_i$, коэффициенты которой $\alpha_{k+1,\,i}$ находятся из требования сопряженности вектора h_{k+1} и уже найденных векторов h_j:

$$\left\langle Ah_j, h_{k+1} \right\rangle = 0, \quad j = 1, ..., k.$$

Отсюда легко получаем, что $\alpha_{k+1,\,j} = -\dfrac{\left\langle Ah_j,\, p_{k+1} \right\rangle}{\left\langle Ah_j, h_j \right\rangle}$ $j = 1, ..., k$. Заметим, что этот процесс позволяет строить искомые векторы единственным образом с точностью до множителя ± 1.

Этот подход также не лишен недостатков, связанных с большим объёмом вычислений, предшествующих применению метода сопряженных направлений.

Основой еще одного подхода является следующее свойство. Допустим, что векторы $h_1, h_2, ..., h_k \in E^n$, $1 \le k \le n-1$, попарно сопряжены относительно матрицы A. Обозначим через L линейное подпространство, образованное системой векторов $\left\{ h_1, h_2, ..., h_k \right\}$. Пусть зафиксированы два вектора $z, w \in E^n$. Введем линейные многообразия $L(z) = z + L$, $L(w) = w + L$. Пусть точки u и v определяются следующим образом: $f\left(u\right) = \min\limits_{x \in L(z)} f\left(x\right)$, $f\left(v\right) = \min\limits_{x \in L(w)} f\left(x\right)$, где f – квадратичная функция.

<u>*Теорема 2.*</u> *Пусть* $h_{k+1} = u - v$. *Если вектор* $w \notin L(z)$, *то вектор* h_{k+1} *сопряжен относительно матрицы* A *с векторами* $h_1, h_2, ..., h_k$.

Теорема 2 позволяет находить взаимосопряженные векторы не до начала работы метода сопряженных направлений, а в процессе его реализации. На этом подходе базируется несколько численных алгоритмов метода.

Однако наиболее эффективной реализацией метода сопряжённых направлений является так называемый ***метод сопряженных градиентов***. В этом методе для нахождения сопряженных направлений последовательно используются элементы градиентного метода. .

В заключение отметим, что существуют обобщения метода сопряжённых направлений для минимизации функций, не являющихся квадратичными. В этом случае метод, вообще говоря, не является конечным.

2.3. Градиентные методы

Градиентные методы относятся к методам первого порядка. Предположим, что минимизируемая функция f определена и непрерывно дифференцируема на пространстве E^n. В градиентных методах для построения последовательности приближений в качестве

направлений итерационного перехода выбираются антиградиенты функции f, вычисленные в точках x_k, то есть $s_k = -f'(x_k)$.
Таким образом,

$$x_{k+1} = x_k - t_k f'(x_k),\ k = 0, 1, ... \qquad (1)$$

Здесь и далее в этом параграфе считаем, что $f'(x_k) \neq 0, \quad \forall k = 0, 1, 2, ...$, то есть ни одна точка последовательности $\{x_k\}$ не является стационарной точкой функции f. Легко убедиться, что если точка x – не стационарная, то $-f'(x)$ является релаксационным направлением для функции f в точке x. Более того, ненулевой антиградиент – это в некотором смысле даже направление наискорейшего убывания. Пояснить такой термин можно следующим образом. Решением задачи миними-зации производной функции f дифференцируемой в точке x по направлениям на единичном шаре, то есть задачи минимизации линейной функции $\langle f'(x), s \rangle$ на множестве $\{s : \| s \| \leq 1\}$, является вектор $-\dfrac{1}{\| f'(x) \|} f'(x)$.
Поэтому $-f'(x)$ и задает направление наискорейшего убывания функции в точке x. Приведенные доводы оправдывают выбор антиградиента как направления итерационного перехода в методах безусловной минимизации. Часто полношаговый градиентный метод называют

методом наискорейшего спуска.

Итак, общее правило генерации последовательности приближений $\{x_k\}$ в градиентных методах задается формулой (1). Разновидности градиентных методов отличаются между собой лишь способом выбора шагового множителя t_k.

Наиболее простым из них является метод с ***постоянным шагом***. Пусть задана некоторая константа $\alpha > 0$. Положим $t_k \equiv \alpha, \quad \forall k = 0,1,2,...$ Тогда

$$x_{k+1} = x_k - \alpha f'(x_k), \quad \forall k = 0,1,2,...$$

Конечно, при произвольно выбранном значении α последовательность приближений может оказаться нерелаксационной и даже неминимизирующей. Известны теоремы, в которых сформулированы требования к функции f, гарантирующие сходимость последовательности приближений при выборе α из некоторого промежутка $(0, \beta]$, где β – положительная константа. Однако на практике, как правило, не удается вычислить значение β до решения задачи.

Тем не менее, этот вариант градиентного метода все же можно использовать на практике, подбирая значение α в процессе минимизации. Начальное значение α выбирается произвольно. На каждой итерации метода проверяется неравенство $f(x_k - \alpha f'(x_k)) < f(x_k)$. Если оно выполняется, полагаем $x_{k+1} = x_k - \alpha f'(x_k)$, и значение α остается неизменным. Если же на какой-то итерации $f\left(x_k - \alpha f'\left(x_k\right)\right) \geq f\left(x_k\right)$ (то есть значение шагового множителя было выбрано слишком большим), заменим его, например, на $\frac{1}{2}\alpha$. Очевидно, что количество дроблений величины α конечно – дробления α прекратятся, как только при некотором значении i выполнится неравенство $\frac{1}{2^i}\alpha \leq \beta$. С этого момента метод становится методом с постоянным шагом.

Главное преимущество этого варианта метода – простота нахождения шагового множителя. Недостатком же его является то, что для получения заданной точности зачастую требуется очень большое число итераций. Поэтому градиентный метод с постоянным шагом обычно используют для решения сравнительно простых задач, где число переменных невелико, время, отпущенное на решение задачи, не лимитировано, и не требуется высокой точности вычисления решения. В противном случае используют более совершенные методы, позволяющие достичь той же точности за меньшее число итераций. Примером такого метода является ***полношаговый градиентный ме-тод***.

Как видно из названия, в этой разновидности метода шаговый множитель определяется как полный, то есть

$$f\left(x_{k+1}\right)=f\left(x_k-t_k f'\left(x_k\right)\right)=\min_{t\geq 0} f\left(x_k-t\, f'\left(x_k\right)\right).$$

Заметим, что одномерная минимизация осуществляется здесь не по всей числовой оси, а только на неотрицательной полуоси, так как направление итерационного перехода является релаксационным. Очевидно, что последовательность, генерируемая этим вариантом градиентного метода, является релаксационной.

Остановимся на изучении полношагового градиентного метода несколько подробнее.

<u>Определение 1.</u> Пусть функция f определена и дифференцируема на E^n. Говорят, что ее градиент f' удовлетворяет на E^n ***условию Липшица***, если существует константа $L>0$ (***константа Липшица***) такая, что

$$\left\|f'(x)-f'(y)\right\|\leq L\left\|x-y\right\|,\quad \forall x,y\in E^n.$$

Приведем следующую лемму.

<u>Лемма 1</u>. Пусть функция f определена и дифференцируема на E^n, ее градиент f' удовлетворяет на E^n условию Липшица. Тогда для любых x,y выполняется неравенство

$$f(y)\leq f(x)+\left\langle f'(x),y-x\right\rangle+\frac{L}{2}\left\|y-x\right\|^2. \quad (1)$$

На основе этой леммы получим оценку приращения функции на луче $x-t\,f'(x),\ \forall t\geq 0$.

Для этого в (1) положим $y = x - t\,f'(x)$, $\forall t \ge 0$. Получим

$$f\left(x - t\,f'(x)\right) \le f(x) + \left\langle f'(x), x - t\,f'(x) - x \right\rangle +$$
$$+\ \frac{L}{2}\left\| x - t\,f'(x) - x \right\|^2 .$$

После преобразований для всех $t \ge 0$ имеем

$$f(x) - f\left(x - t\,f'(x)\right) \ge t\left\| f'(x) \right\|^2 - \frac{L}{2}t^2\left\| f'(x) \right\|^2 .$$

Откуда

$$f(x) - f\left(x - t\,f'(x)\right) \ge t\left(1 - \frac{L}{2}t\right)\left\| f'(x) \right\|^2,\ t \ge 0\,. \quad (2)$$

<u>*Теорема 1*</u>. ***(Общая теорема сходимости полношагового градиентного метода)***

Пусть функция f определена, дифференцируема и ограничена снизу на E^n, ее градиент f' удовлетворяет на E^n условию Липшица, последовательность $\{x_k\}$, $k = 0,1,2,\dots$, построена по правилам полношагового градиентного метода, где x_0 – произвольный вектор из E^n. Тогда $\lim\limits_{k\to\infty} f'(x_k) = 0$.

<u>*Доказательство*</u>. В неравенстве (2) положим $x = x_k, \forall k = 0, 1, 2, \dots$ Получим для всех $t \ge 0$

$$f(x_k) - f\left(x_k - t\,f'(x_k)\right) \ge t\left(1 - \frac{L}{2}t\right)\left\| f'(x_k) \right\|^2 .$$

Отсюда в силу выбора шагового множителя t_k имеем

$f(x_k) - f(x_{k+1}) \geq t\left(1 - \frac{L}{2}t\right)\| f'(x_k)\|^2$, $t \geq 0$. В этом неравенстве положим $t = \frac{1}{L}$. Получим

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2L}\| f'(x_k)\|^2, \quad \forall k = 0,1,2,\ldots \quad (3)$$

Так как последовательность $\{x_k\}$, $k = 0,1,2,\ldots$, является релаксационной, и функция f ограничена снизу на E^n, то последовательность $\{f(x_k)\}$ сходится. Следовательно, $\lim\limits_{k\to\infty}\{f(x_k) - f(x_{k+1})\} = 0$. Отсюда и из (3) $\lim\limits_{k\to\infty}\| f'(x_k)\|^2 = 0$. Откуда и следует утверждение теоремы.

Теорема 2. Пусть выполняются все условия теоремы 1 и последовательность $\{x_k\}$, $k = 0,1,2,\ldots$, *имеет предельную точку* $\bar{x}$. *Тогда* $f'(\bar{x}) = 0$.

Доказательство. Так как $\bar{x}$ – предельная точка последовательности $\{x_k\}$, $k = 0,1,2,\ldots$, то существует подпоследовательность $\{x_{k_j}\} \subset \{x_k\}$ такая, что $\lim\limits_{j\to\infty}\{x_{k_j}\} = \bar{x}$. В силу теоремы 1 и непрерывности f' имеем

$$f'(\bar{x}) = f'\left(\lim_{j\to\infty} x_{k_j}\right) = \lim_{j\to\infty} f'\left(x_{k_j}\right) = \lim_{k\to\infty} f'(x_k) = 0.$$

Что и требовалось.

Таким образом, в условиях теоремы 2 любая предельная точка последовательности приближений является стационарной точкой функции f.

<u>*Теорема 3*</u>. (***Теорема сходимости полношагового градиентного метода для выпуклых функций***)

Пусть выполняются все условия теоремы 1. Если, кроме того, функция f выпукла и множество $E(x_0) = \{x : f(x) \le f(x_0)\}$ ограничено, то безусловный минимум функции f достигается, последовательность $\{x_k\}$, $k = 0,1,2,\ldots$, имеет предельные точки, любая ее предельная точка является точкой минимума функции f, и последова-тельность $\{x_k\}$ является минимизирующей.

<u>*Доказательство*</u>. Множество $E(x_0)$ замкнуто в силу непрерывности функции f и ограничено, поэтому безусловный минимум функции f достигается.

В силу релаксационности последовательности $\{x_k\}$ выполняется включение $\{x_k\} \subset E(x_0)$. Тогда, в силу замкнутости и ограниченности множества

$E(x_0)$, последовательность $\{x_k\}$ имеет предельные точки.

Пусть x^* – одна из этих предельных точек. Из теоремы 2 следует, что $f'(x^*)=0$. Это – достаточное условие безусловного минимума выпуклой функции f. Таким образом, x^* – точка безусловного минимума функции f.

Убедимся, наконец, что $\lim\limits_{k\to\infty} f(x_k) = f^*$. Пусть подпоследовательность $\{x_{k_j}\} \subset \{x_k\}$ такова, что $\lim\limits_{j\to\infty} x_{k_j} = x^*$. Тогда в силу непрерывности функции f и сходимости последовательности $\{f(x_k)\}$,

$$\lim_{k\to\infty} f(x_k) = \lim_{j\to\infty} f\left(x_{k_j}\right) = f\left(\lim_{j\to\infty} x_{k_j}\right) = f(x^*) = f^*.$$

Что и требовалось.

<u>*Теорема 4*</u>. *Пусть выполняются все условия теоремы 3, и функция f имеет единственную точку безусловного минимума x^*. Тогда $\lim\limits_{k\to\infty} x_k = x^*$.*

<u>*Доказательство*</u>. Пусть это не так. Тогда последовательность $\{x_k\}$ имеет еще одну предельную точку $y^* \neq x^*$. Но, как следует из теоремы 3, y^* – также точка безусловного минимума функции f, что противоречит условию теоремы. Полученное противоречие и доказывает теорему.

Заметим, что условие этой теоремы может

выполняться, например, в случае строгой выпуклости функции f.

Кроме изученных здесь вариантов градиентного метода известно большое количество и других модификаций этого метода, многие из которых часто используются на практике. Остановимся вкратце на некоторых из них.

Во-первых, вариант метода, в котором шаговый множитель t_k находится из условий $t_k \geq \delta$ и

$$f\left(x_k\right) - f\left(x_k - t_k\, f'\left(x_k\right)\right) \geq \gamma t_k \left\| f'\left(x_k\right)\right\|^2 \qquad (4)$$

при некотором достаточно малом $\delta > 0$ и любом $\gamma > 0$. Для отыскания значения t_k, удовлетворяющего неравенству (4), используют следующий прием. Выберем произвольное значение $t_{k,0} > 0$. Если для него неравенство (4) не выполняется (значение $t_{k,0}$ слишком велико), то положим $t_{k,1} = \dfrac{t_{k,0}}{2}$ и проверим, удовлетворяет ли $t_{k,1}$ неравенству (4). Если нет, то процесс дробления продолжим. Можно показать, что в условиях теоремы 1 количество таких дроблений конечно. Пусть i – наименьшее целое число, для которого величина $t_{k,\,i} = \dfrac{t_{k,\,0}}{2^i}$ удовлетворяет неравенству (4), тогда положим $t_k = t_{k,\,i}$. Если же начальное значение $t_{k,0}$

удовлетворяет неравенству (4), то желательно попытаться увеличить его значение, полагая $t_{k,\,1} = 2t_{k,\,0}$. Количество умножений также конечно. Пусть $t_{k,\,i} = 2^i t_{k,\,0}$ удовлетворяет неравенству (4), а $t_{k,\,i+1} = 2^{i+1} t_{k,\,0}$ не удовлетворяет. Тогда положим $t_k = t_{k,\,i}$. Для уменьшения количества дроблений либо умножений полезно выбирать $t_{k,\,0} = t_{k-1}$, а значение $t_{0,\,0}$ выбирается произвольно.

Во-вторых, вариант метода, в котором шаговый множитель $t_k = \dfrac{\lambda_k}{\| f'(x_k) \|}$, $k = 0, 1, \dots$, где $\lambda_k > 0$, $k = 0, 1, \dots$, $\lim\limits_{k \to \infty} \lambda_k = 0$, $\sum\limits_{k=0}^{\infty} \lambda_k = \infty$. Заметим, что в этом случае градиентный метод генерирует нерелаксационную, вообще говоря, последовательность приближений.

Наконец, приведем еще один вариант метода, который используется в случае, когда известно минимальное значение функции f^*. В этом случае шаговый множитель вычисляется по формуле $t_k = \dfrac{f(x_k) - f^*}{\| f'(x_k) \|^2}$, $k = 0, 1, \dots$ В этой точке касательная к графику функции $\varphi(t) = f\left(x_k - t\, f'(x_k)\right)$, построенная в точке 0, принимает значение f^*.

В заключение параграфа коротко опишем ***метод сопряженных градиентов*** (***метод***

Флетчера-Ривса). Его можно отнести к группе методов первого порядка и, в то же время, к методу сопряженных направлений. Пусть матрица A является симметричной и положительно определенной, функция $f(x) = \frac{1}{2}\langle Ax, x\rangle + \langle b, x\rangle + c$. В этом методе направление итерационного перехода s_k вычисляется по формулам

$$\begin{aligned} s_k &= -f'(x_k) + \alpha_{k-1}s_{k-1},\ k \ge 1, \\ s_0 &= -f'(x_0). \end{aligned} \tag{5}$$

Таким образом, s_k – это линейная комбинация градиента функции f в точке x_k и предыдущего направления s_{k-1}. Множитель α_{k-1} находится из требования сопряженности векторов s_k и s_{k-1} относительно матрицы A

$$\alpha_{k-1} = \frac{\langle As_{k-1}, Ax_k + b\rangle}{\langle As_{k-1}, s_{k-1}\rangle}.$$

Тогда $x_{k+1} = x_k + t_k s_k$, где шаг находится как полный, то есть

$$t_k = \frac{-\langle Ax_k + b, s_k\rangle}{\langle As_k, s_k\rangle}.$$

Векторы $\{s_0, ..., s_k\}$, полученные по формулам (5), попарно сопряжены относительно матрицы A. Поэтому данный метод позволяет за конечное число итераций отыскать безусловный минимум функции f.

Этот метод применяется также и для минимизации неквадратичных функций. Для этих целей формулам для нахождения коэффициента α_{k-1} придают следующий вид:

$$\alpha_{k-1} = \frac{\left\langle f'(x_k),\, f'(x_k) - f'(x_{k-1}) \right\rangle}{\left\| f'(x_{k-1}) \right\|^2},$$

или

$$\alpha_{k-1} = \frac{\left\| f'(x_k) \right\|^2}{\left\| f'(x_{k-1}) \right\|^2}.$$

Несмотря на то, что для неквадратичных функций метод сопряженных градиентов уже не является конечным, он – один из наиболее используемых методов безусловной минимизации.

2.4. Метод Ньютона

В этом параграфе мы познакомимся с ***методом Ньютона***, который был разработан одним из первых в группе методов второго порядка. Пусть функция f определена и дважды дифференцируема на E^n. Пусть найдены векторы $x_0, x_1, \dots, x_k$. Опишем правила вычисления следующего элемента последовательности приближений. Так как функция f дважды дифференцируема, то квадратичная часть ее приращения $f(x)-f(x_k)$ в окрестности точки x_k имеет вид

$$F_k(x)=\left\langle f'(x_k),x-x_k\right\rangle+\frac{1}{2}\left\langle f''(x_k)(x-x_k),x-x_k\right\rangle.$$

Здесь $f''(x_k)$ – матрица вторых частных производных функции f в точке x_k.

Пусть функция f такова, что квадратичная функция $F_k(x)$ имеет единственную точку безусловного минимума. Положим

$$F_k(x_{k+1})=\min_{x\in E^n}F_k(x)\cdot \qquad (1)$$

Равенство (1) и определяет правило построения последовательности приближений методом Ньютона.

Прежде чем обсуждать возможности численной реализации этого метода проведем некоторые построения. Градиент функции F_k имеет вид $F_k'(x)=f'(x_k)+f''(x_k)(x-x_k)$. Так как вектор x_{k+1} является безусловным минимумом функции F_k, то $F_k'(x_{k+1})=0$, то есть

$$f'(x_k)+f''(x_k)(x_{k+1}-x_k)=0. \qquad (2)$$

Таким образом, вектор x_{k+1} является решением системы линейных алгебраических уравнений

$$f'(x_k)+f''(x_k)(x-x_k)=0\,. \qquad (3)$$

Из единственности решения задачи (1) следует невырожденность матрицы $f''(x_k)$. Поэтому существует обратная ей матрица $\left[f''(x_k)\right]^{-1}$. Тогда из равенства (2) получим

$$x_{k+1}=x_k-\left[f''(x_k)\right]^{-1}f'(x_k). \qquad (4)$$

Итак, мы получили три соотношения (1), (2) и (4), определяющие вектор x_{k+1}. На этих равенствах основаны три группы численно реализуемых алгоритмов метода Ньютона.

Первая задается формулой (1). В зависимости от выбора того или иного метода безусловной минимизации функции F_k получаются различные алгоритмы этой группы. Для этих целей можно применять, например, методы прямого поиска либо методы первого порядка, изученные в предыдущих параграфах.

Вторая группа алгоритмов основана на решении системы уравнений (3). Существует боль-шое количество методов решения таких систем. В каждом конкретном случае имеется возможность подобрать подходящий метод. В результате будут получены различные численно реализуемые алгоритмы. Такой подход к отысканию приближения x_{k+1} – один из самых используемых на практике.

Наконец, формула (4) также может быть

положена в основу алгоритмов нахождения x_{k+1}. Для этого необходимо использовать те или иные методы обращения матрицы $f''(x_k)$. Однако, главная ценность формулы (4) в удобстве ее использования при анализе последовательности приближений, вырабатываемой методом Ньютона, а также для разработки различных модификаций этого метода. В частности, легко увидеть, что метод Ньютона укладывается в общую схему итерационных методов $x_{k+1} = x_k + t_k s_k$, где $t_k \equiv 1$, $s_k = -\left[f''(x_k) \right]^{-1} f'(x_k)$.

Метод Ньютона имеет ряд положительных особенностей. При помощи этого метода зачастую удается за небольшое количество итераций получать решение задачи на безусловной минимум с высокой точностью. Однако как теоретическое обоснование этого метода, так и опыт его практического применения показывают на ряд больших сложностей в реализации метода. Показано, что метод применим лишь при некоторых дополнительных требованиях к целевой функции и начальному приближению x_0. Вспомогательная задача, решаемая на каждой итерации, требует значительных вычислительных затрат независимо от варианта используемого алгоритма. Подробное обсуждение всех перечисленных проблем не входит в рамки первого знакомства с этим методом в нашем пособии. Следует иметь в виду, что эти проблемы труднопреодолимы, а их игнорирование может сделать применение метода бесполезным.

Большое количество работ посвящено развитию метода Ньютона. Известны различные модификации этого метода, поэтому первоначальный вариант метода, изложенный выше, часто называют ***классическим методом Ньютона***.

Модификации можно разделить на две группы. Во-первых, это методы, которые ценой еще большего усложнения позволяют устранить некоторые отрицательные особенности классического метода Ньютона. Примером такого метода является так называемый ***модифицированный метод Ньютона***. Как отмечалось выше, в классическом варианте метода шаговый множитель равен единице. В модифицированном методе Ньютона формула, задающая правило построения последовательности приближений, приобретает вид

$$x_{k+1} = x_k - t_k \left[f''(x_k) \right]^{-1} f'(x_k),$$

где шаговый множитель t_k определяется как полный, то есть путем одномерной минимизации

$$f(x_{k+1}) = \min_{t \geq 0} f\left(x_k - t\left[f''(x_k) \right]^{-1} f'(x_k) \right).$$

Конечно, такая модификация заметно увеличивает объем вычислений на каждой итерации метода, поскольку теперь кроме решения подзадачи отыскания направления итерационного перехода на каждой итерации решается еще одна вспомогательная задача отыскания шагового множителя. Однако у модифицированного метода имеются значительные преимущества. Главное – это то, что начальное приближение x_0 может быть произвольным, а также существенно ослабляются требования к свойствам функции f. Таким образом, диапазон применимости метода становится значительно шире. Заметим, что в модифицированном методе $\lim\limits_{k\to\infty} t_k = 1$, то есть «модифицированный метод в пределе превращается в классический». Поэтому можно на некоторой итерации переключить построение последовательности приближений с модифицированного метода на классический, что позволит уменьшить затраты на осуществление итерации без потери «качества приближений».

Другое направление модификаций основано на замене обратной матрицы $\left[f''(x_k)\right]^{-1}$ некоторой ее аппроксимацией A_k, что заметно упрощает выполнение каждой итерации и расширяет область применения таких методов. Тогда правило построения последовательности приближений принимает вид

$$x_{k+1} = x_k - A_k f'(x_k),$$

где A_k – симметричная квадратная матрица. Существует большое количество различных процедур вычисления этих матриц. Методы этой группы называются ***квазиньютоновскими***. Они заметно проще в реализации по сравнению с классическим методом Ньютона, но мало чем уступают ему по эффективности. Поэтому среди современных методов безусловной минимизации квазиньютоновские методы – одни из наиболее используемых.

3. Методы решения задачи нелинейного программирования

3.1. Метод проекции градиента

В этом разделе мы познакомимся с основными направлениями в построении методов условной минимизации. Начнем это знакомство с градиентных методов. По вполне понятным соображениям градиентные методы безусловной минимизации для решения условно экстремальных задач непригодны. Известные модификации градиентных методов используют разные механизмы учета ограничений на значения переменных. Это достигается различными способами. Поэтому в настоящее время известно довольно много разновидностей градиентных методов условной минимизации. Среди них метод проекции градиента, метод условного градиента, методы возможных направлений, метод проецируемых градиентов, метод линеаризации и некоторые другие методы.

Данный параграф посвящен изучению ***метода проекции градиента***.

Пусть требуется отыскать условный минимум определенной и непрерывно дифференцируемой на E^n функции f на выпуклом замкнутом множестве $D \subset E^n$. Предположим, что, начиная с произвольной точки $x_0 \in D$, уже найдены приближения $x_1, x_2, ..., x_k \in D$. Следующая формула задает

правило нахождения очередного приближения методом проекции градиента $x_{k+1} = P\left(x_k - t_k\, f'(x_k) \right)$, где $P(\cdot)$ – проекция точки на множество D (напомним, что в силу выпуклости и замкнутости множества D проекция существует и единственна [([1], параграф 1.7)]), t_k – шаговый множитель (возможные способы нахождения t_k мы обсудим ниже).

<u>*Теорема 1.*</u> *Пусть $D \subset E^n$ – выпуклое замкнутое множество, функция f определена, непрерывно дифференцируема и выпукла на E^n. Для того чтобы точка $x^* \in D$ была точкой условного минимума функции f на D, необходимо и достаточно, чтобы* $P\left(x^* - t\, f'(x^*) \right) = x^*, \ \forall t \geq 0$.

<u>*Доказательство.*</u> В известном неравенстве $\left\langle P(x)-x, y-P(x)\right\rangle \ge 0, \quad \forall y \in D$ положим $x = x^* - t\, f'(x^*)$, $P(x) = x^*$ при всех $t \ge 0$. Получим $\left\langle f'(x^*), y - x^*\right\rangle \ge 0$, $\forall y \in D$, что и доказывает утверждение теоремы.

Будем считать, что $P\left(x_k - t_k\, f'(x_k)\right) \ne x_k$, $\forall k = 0, 1, \ldots$ Это означает, что процесс построения последовательности приближений бесконечен, ни одна точка x_k не является точкой условного минимума.

Укажем теперь некоторые способы регулировки шагового множителя t_k используемые в методе проекции градиента. Один из наиболее простых – это нахождение t_k так же, как в полношаговом градиентном методе безусловной минимизации:

$$f\left(x_k - t_k\, f'(x_k)\right) = \min_{t \ge 0} f\left(x_k - t\, f'(x_k)\right).$$

Простота его объясняется тем, что при выборе шага t_k ограничения на область изменения переменных вообще не учитываются. Соблюдение ограниче-ний достигается только за счет проектирования. Область применения такого способа вычисления шага ограничена. Например, для многих, даже весьма простых задач на условный минимум, задача $\min\limits_{t\geq 0} f\left(x_k - t\, f'\left(x_k\right)\right)$ может не иметь решения (например, одномерная функция может быть неограничена снизу).

Другой метод определения шага заключается в следующем: $f\left(x_{k+1}\right) = \min\limits_{t\geq 0} f\left(P\left(x_k - t\, f'\left(x_k\right)\right)\right)$. Этот способ также имеет существенный недостаток – решение этой подзадачи требует слишком больших вычислительных затрат.

В некотором смысле «золотой серединой» между этими двумя способами является определение шагового множителя при помощи нахождения положительного решения неравенства

$$f\left(x_k\right) - f\left(P\left(x_k - t\, f'\left(x_k\right)\right)\right) \geq \gamma \left\| x_k - P\left(x_k - t\, f'\left(x_k\right)\right)\right\|^2,$$

где γ – действительная константа из $(0,1]$. Если какое-либо положительное значение t_k удовлетворяет этому неравенству, то, подставив его в неравенство, получим

$$f\left(x_k\right) - f\left(x_{k+1}\right) \geq \gamma \left\| x_k - x_{k+1}\right\|^2. \qquad (1)$$

Нахождение шагового множителя из неравенства (1) является наиболее употребимым способом регулировки шага в этом методе. Далее мы обоснуем возможность использования такого варианта метода проекции градиента, а также приведем метод решения неравенства (1).

Проведем некоторые вспомогательные построения. Для любого k в неравенстве $\left\langle P(x)-x, y-P(x)\right\rangle \geq 0, \quad \forall y \in D$ положим $x = x_k - t_k\, f'(x_k)$, $P(x) = x_{k+1}$. Получим

$$\left\langle x_{k+1} - x_k + t_k f'(x_k), y - x_{k+1}\right\rangle \geq 0,$$

откуда

$$\left\langle x_{k+1} - x_k, y - x_{k+1}\right\rangle + t_k \left\langle f'(x_k), y - x_{k+1}\right\rangle \geq 0.$$

Тогда $\forall y \in D$,

$$\left\langle f'(x_k), y - x_{k+1}\right\rangle \geq \frac{1}{t_k}\left\langle x_{k+1} - x_k, y - x_{k+1}\right\rangle, \; k = 0,1, \ldots \quad (2)$$

<u>*Теорема 2.*</u> ***(О выборе шагового множителя)*** *Пусть $D \subset E^n$ – выпуклое замкнутое множество,*

функция f определена, непрерывно дифференцируема и выпукла на E^n, ее градиент f' удовлетворяет на E^n условию Липшица с константой L, Тогда любое $t_k \in \left(0, \dfrac{2}{L+2\gamma}\right]$ удовлетворяет неравенству (1).

<u>*Доказательство.*</u> Воспользуемся неравенством из леммы 1 параграфа 2.3

$$f(y) \leq f(x) + \left\langle f'(x), y - x\right\rangle + \frac{L}{2}\left\| y - x\right\|^2, \; \forall x, y \in E^n.$$

Положим здесь $x = x_k$, $y = x_{k+1}$. Получим $\forall k = 0,1,\ldots$,

$$f(x_k) - f(x_{k+1}) \ge \left\langle f'(x_k), x_k - x_{k+1} \right\rangle - \frac{L}{2}\|x_k - x_{k+1}\|^2. \quad (3)$$

В неравенстве (2) подставим $y = x_k$, тогда

$$\left\langle f'(x_k), x_k - x_{k+1} \right\rangle \ge \frac{1}{t_k}\| x_k - x_{k+1} \|^2, \quad \forall k = 0,1,2,\ldots$$

Отсюда и из (3)

$$f(x_k) - f(x_{k+1}) \ge \frac{1}{t_k}\| x_k - x_{k+1} \|^2 - \frac{L}{2}\| x_k - x_{k+1} \|^2 =$$

$$= \left(\frac{1}{t_k} - \frac{L}{2}\right)\| x_k - x_{k+1} \|^2 \ge \left(\frac{L + 2\gamma}{2} - \frac{L}{2}\right)\| x_k - x_{k+1} \|^2 =$$

$$= \gamma \| x_k - x_{k+1} \|^2, \quad \forall k = 0,1,2,\ldots$$

Что и требовалось.

<u>*Теорема 3.*</u> *Пусть выполняются все условия*

теоремы 2. Если, кроме того, функция f *ограничена снизу на* E^n, *то*

$$\lim_{k\to\infty}\|x_k - x_{k+1}\| = 0.$$

<u>*Доказательство.*</u> Из (1) следует, что последовательность $\{x_k\}$, $k = 0,1,2,\ldots$ релаксационна, а так как функция f ограничена снизу, то последовательность $\{f(x_k)\}$ сходится. Следовательно, $\lim\limits_{k\to\infty}\{f(x_k) - f(x_{k+1})\} = 0$. Отсюда и из

(1) $\lim_{k\to\infty}\gamma\left\|x_k - x_{k+1}\right\|^2 = 0$. Откуда и следует утверждение теоремы.

Прежде, чем сформулировать следующую теорему, обозначим через $E(x_0)$ множество $\{x:\ x\in D,\ f(x)\ \le f(x_0)\}$.

<u>*Теорема 4.*</u> (***О сходимости метода проекции градиента***) *Пусть выполняются все условия теорем 2 и 3, константа* δ *такова, что* $0<\delta<\dfrac{2}{L+2\gamma}$, *функция* f *выпукла на множестве* D, *точка* $x_0\in D$, *множество* $E(x_0)$ *ограничено, последовательность* $\{x_k\}$, $k=0,1,2,\ldots$, *построена методом проекции градиента, где* $t_k\in\left(\delta, \dfrac{2}{L+2\gamma}\right]$. *Тогда существует минимум функции* f *на множестве* D, *последовательность* $\{x_k\}$, $k=0,1,2,\ldots$ *имеет предельные точки, является минимизирующей, и любая ее предельная точка является точкой условного минимума функции* f *на множестве* D.

<u>*Доказательство.*</u> Справедливость первого утверждения следует из непрерывности функции f и компактности множества $E(x_0)$.

Последовательность $\{x_k\}, \quad k = 0,1,2,...$ имеет предельные точки, так как (в силу релаксационности) $\{x_k\} \subset E(x_0), \quad \forall k = 0,1,2,...$

Убедимся в том, что последовательность $\{x_k\}, \quad k = 0,1,2,...$является минимизирующей. Пусть $x^* \in D^*$. Воспользуемся градиентным неравенством для выпуклых функций ([1], 1.4, теорема 1). Получим

$$f(x^*) - f(x_k) \geq \langle f'(x_k), x^* - x_k \rangle, \ \forall k = 0,1,...$$

Отсюда для всех $k = 0,1,...$

$$0 \leq f(x_k) - f(x^*) \leq \langle f'(x_k), \ x_k - x^* \rangle =$$

$$= \langle f'(x_k), \ x_k - x_{k+1} \rangle + \langle f'(x_k), \ x_{k+1} - x^* \rangle. \qquad (4)$$

Далее, в неравенстве (2) положим $y = x^*$, тогда

$$\langle f'(x_k), x^* - x_{k+1} \rangle \geq \frac{1}{t_k} \langle x_k - x_{k+1}, x^* - x_{k+1} \rangle.$$

Отсюда и из (4) получим $0 \leq f(x_k) - f(x^*) \leq$

$$\leq \langle f'(x_k), x_k - x_{k+1} \rangle + \frac{1}{t_k} \langle x_k - x_{k+1}, \ x^* - x_{k+1} \rangle \leq$$

$$\leq \| f'(x_k) \| \, \| x_k - x_{k+1} \| + \frac{1}{t_k} \| x^* - x_{k+1} \| \, \| x_k - x_{k+1} \| =$$

$$= \left(\| f'(x_k) \| + \frac{1}{t_k} \| x^* - x_{k+1} \| \right) \| x_k - x_{k+1} \| \le$$

$$\le \left(M + \frac{d}{\delta} \right) \| x_k - x_{k+1} \| = C \| x_k - x_{k+1} \|, \ k = 0, 1, \ldots$$

Здесь положительные константы M, d и C определены следующим образом: $M = \max_{x \in E(x_0)} \| f'(x) \|$,

$d = \max_{x,\, y \in E(x_0)} \| x - y \|$, $C = M + \dfrac{d}{\delta}$. Итак,

$$0 \le f(x_k) - f(x^*) \le C \| x_k - x_{k+1} \| \quad , \forall k = 0, 1, \ldots$$

Отсюда и из теоремы 3 $\lim_{k \to \infty} f(x_k) = f(x^*) = f^*$. Таким образом, последовательность $\{x_k\}$ является минимизирующей.

Докажем последнее утверждение теоремы. Пусть x^* – предельная точка последовательности $\{x_k\}$. Тогда

$$f(x^*) = f\left(\lim_{j \to \infty} x_{k_j} \right) = \lim_{j \to \infty} f(x_{k_j}) = \lim_{k \to \infty} f(x_k) = f^*.$$

Итак, x^* – минимум функции f на множестве D. Что и требовалось.

<u>*Теорема 5*</u>. *Пусть выполняются все условия теоремы 4, и функция* f *имеет единственную точку* x^* *условного минимума на множестве* D.

Тогда $\lim\limits_{k\to\infty} x_k = x^*$.

Доказательство аналогично доказательству теоремы 4 из параграфа 2.3.

Познакомимся с приемом отыскания значения шагового множителя t_k, удовлетворяющего неравенству (2). Заметим, что мы не можем не-посредственно использовать теорему 2 на практике, поскольку она предполагает знание константы Липшица градиента целевой функции. Следующий несложный прием позволяет вычислять значение t_k, удовлетворяющее неравенству (2), по возможности близкое к правому концу области решений этого неравенства. Зададимся, вообще говоря, произвольным начальным значением $t_{k,0} > 0$. Если для него неравенство (2) не выполняется, то, как следует из теоремы 2 значение $t_{k,0}$ выбрано слишком большим. Положим $t_{k,1} = \dfrac{t_{k,0}}{2}$ и проверим, удовлетворяет ли новое значение неравенству (2). Если нет, то процесс дробления продолжим. Количество таких дроблений, как следует из теоремы 2, конечно. Пусть i таково, что величина $t_{k,i} = 2^i t_{k,0}$ удовлетворяет неравенству (2), а $t_{k,i+1} = 2^{i+1} t_{k,0}$ не удовлетворяет. Тогда положим $t_k = t_{k,i}$. Для уменьшения количества дроблений либо домножений полезно начальное значение $t_{k,0}$

полагать равным t_{k-1}. Значение $t_{0,\,0}$ выбирается произвольно.

Отметим, наконец, что метод проекции градиента имеет смысл применять только к тем задачам условной минимизации, у которых допустимые множества являются «простыми» с точки зрения проектирования.

3.2. Метод условного градиента

В методе условного градиента для нахождения условно релаксационных направлений для минимизируемой функции f используется градиент f'. Это позволяет подбирать шаговый множитель, не выводящий очередное приближение за пределы допустимой области. Рассмотрим подробнее, какими средствами это достигается.

Пусть функция f определена и непрерывно дифференцируема на E^n, множество $D \subset E^n$ – выпуклый компакт. Предположим, что, начиная с произвольной точки $x_0 \in D$, найдены приближения $x_1, x_2, ..., x_k \in D$. Опишем действия, которые необходимо выполнить для отыскания следующего приближения $x_{k+1} \in D$.

Сформулируем вспомогательную задачу

$$\min_{y \in D} \left\langle f'(x_k),\, y \right\rangle. \qquad (1)$$

Легко увидеть, что при сделанных выше предположениях задача (1) имеет решение, обозначим его y_k. Далее, положим $s_k = y_k - x_k$.

Если $\langle f'(x_k), s_k \rangle \geq 0$, то согласно определению точки y_k имеем $\langle f'(x_k), y - x_k \rangle \geq 0, \quad \forall y \in D$. Таким образом, в точке $x_k \in D$ выполняется необходимое условие условного минимума функции f на множестве D. Если, кроме того, функция f – выпуклая на множестве D, то это условие также и достаточно ([1], 2.6, теорема 5).

Будем считать, что $\langle f'(x_k), s_k \rangle < 0 \quad \forall k = 0, 1, ...$ Это означает, что процесс построения последовательности приближений бесконечен, ни одна точка x_k не является точкой условного минимума. Тогда s_k является условно релаксационным направлением для функции f в точке x_k относительно множества D, значит, можно сделать положительный шаг в этом направлении, не нарушив ограничений и уменьшив значение функции. Например, такой шаговый множитель можно определить следующим образом: $f(x_k + t_k s_k) = \min\limits_{t \in [0,1]} f(x_k + t s_k)$ и найти следующее приближение $x_{k+1} = x_k + t_k s_k$.

В заключение – о сфере применения этого метода. Так как задача (1) – это задача минимизации линейной функции $\left\langle f'(x_k), y \right\rangle$ на выпуклом множестве D, то случай, когда D – выпуклое многогранное множество, является наиболее простым. Задача (1) тогда – задача линейного программирования, и для ее решения пригодны соответствующие методы.

3.3. Метод штрафных функций

Метод штрафных функций относится к группе методов последовательной безусловной минимизации для решения условно экстремальных задач. В этих методах решение условно экстремальной задачи сводится к решению последовательности задач на безусловный экстремум. На таком подходе, кроме метода штрафных функций, базируются и некоторые другие методы, например, метод барьеров, метод центров, методы модифицированных функций Лагранжа. Познакомимся с этим классом методов на примере метода штрафных функций.

<u>*Определение 1.*</u> *Пусть D – некоторое множество из E^n. Назовем функцию $P(x)$, определенную на всем пространстве E^n, **штрафной функцией** для множества D, если $P(x) = 0 \ \forall x \in D$ и $P(x) > 0 \ \forall x \notin D$.*

Легко увидеть, что можно построить различные функции, удовлетворяющие этому определению. Например, самой простой штрафной функцией может служить следующая функция:

$$I(x) = \begin{cases} 0, & x \in D, \\ c, & x \notin D, \end{cases}$$

где c – некоторая положительная константа. Эта функция имеет недостатки, затрудняющие использование ее на практике: она не является непрерывной, и величина штрафа не зависит от величины нарушения ограничений, задающих множество D. Учитывая информацию о системе ограничений, определяющих D, можно построить штрафные функции, лишенные этих недостатков.

При решении задач на условный экстремум нужны штрафные функции для множеств, задаваемых как системами уравнений, так и системами неравенств. Пусть $D = \{x : f_i(x) = 0,\ i = 1, \dots, m\}$, где f_i – некоторые функции, определенные на E^n. Для таких множеств можно использовать, например, следующие штрафные функции:

$$P_1(x) = \sum_{i=1}^{m} f_i^2(x) \quad \text{и} \quad P_2(x) = \sum_{i=1}^{m} \left| f_i(x) \right|.$$

Заметим, что свойства штрафных функций определяются свойствами функций f_i. Например, если все функции f_i непрерывны, то эти штрафные функции также непрерывны. Если все функции f_i дифференцируемы, то P_1 также дифференцируема, однако функция P_2 может быть недифференцируемой. Если все функции f_i линейны, то функции P_1 и P_2 выпуклы.

Прежде, чем указать примеры штрафных функций для множеств, определяемых системами неравенств, введем так называемую ***функцию срезки***.

<u>*Определение 2.*</u> *Пусть* f *– функция, определенная на* E^n*. Функцию*

$$f^+(x) = \begin{cases} f(x), & \text{если } f(x) \ge 0, \\ 0, & \text{если } f(x) < 0 \end{cases}$$

назовем ***функцией срезки*** *для функции* f.

Заметим, что при любом x справедливо неравенство $f(x) \le f^+(x)$.

Пусть теперь множество D задано неравенствами, то есть

$$D = \{x : f_i(x) \le 0,\ i = 1, \dots, m\},$$

где f_i – некоторые функции, определенные на E^n. Используя функции срезки, то же самое множество можно задать при помощи системы уравнений

$$D = \{x : f_i^+(x) = 0,\ i = 1, \dots, m\}.$$

Штрафные функции для такого множества можно задать тогда следующим образом:

$$P_1(x) = \sum_{i=1}^{m}\left(f_i^+(x)\right)^2 \quad \text{и} \quad P_2(x) = \sum_{i=1}^{m} f_i^+(x).$$

Заметим, что, несмотря на то, что функция срезки f^+, вообще говоря, недифференцируема (даже если дифференцируема функция f), функция P_1 является дифференцируемой. Если все функции f_i выпуклые, то (с учетом неотрицательности срезок) функции P_1 и P_2 выпуклы.

Легко увидеть, что линейная комбинация нескольких штрафных функций с положительными коэффициентами также является штрафной функцией.

Пусть имеется задача на отыскание условного минимума функции f на множестве $D \subset E^n$. Определим на E^n следующую функцию:

$$\Phi(x,\gamma) = f(x) + \gamma P(x),$$

где $\gamma > 0$. Легко увидеть, что из определения штрафной функции следует, что $\Phi(x,\gamma) = f(x)$ для $x \in D$ и $\Phi(x,\gamma) > f(x)$ для $x \notin D$.

Далее, при фиксированном положительном γ поставим задачу

$$\min_{x \in E^n} \Phi(x,\gamma). \tag{1}$$

Предположим, что задача (1) имеет решение. Обозначим через $x(\gamma)$ точку безусловного минимума функции $\Phi(x,\gamma)$.

Известно, что при достаточно больших значениях параметра γ значение $f\left(x(\gamma)\right) \to f^*$. Этот факт и является основой метода штрафных функций. В ходе практического применения метода приходится преодолевать некоторые трудности. Во-первых, невозможно заранее найти та-кое значение параметра γ, которое обеспечивало бы необходимую точность приближения. Во-вто-рых, решение задачи (1) при больших значениях параметра γ сопряжено со значительными вычислительными сложностями.

Для преодоления указанных трудностей при численной реализации метода задается некоторая

неограниченная сверху последовательность $\{\gamma_k\}$, такая, что $\gamma_k > 0$, $\gamma_{k+1} > \gamma_k$, $k = 0,1,\ldots$ Обозначим $\Phi_k(x) = \Phi(x,\gamma_k)$, $x_k = x(\gamma_k)$. Таким образом, для построения последовательности приближений $\{x_k\}$ решается последовательность задач на безусловный минимум

$$\min_{x \in E^n} \Phi_k(x). \tag{2}$$

Далее будем считать, что для любого значения $k = 0,1,\ldots$ Действительно, если найдется k такое что $x_k \in D$, то легко увидеть, что $x_k \in D^*$

(поскольку $P(x_k) = 0$). Приведем для одного частного случая условия, при которых последовательность приближений $\{x_k\}$ является минимизирующей.

По аналогии с ранее введенной векторнозначной функцией $F(x) = (f_1(x), ..., f_m(x))$, обозначим $F^+(x) = (f_1^+(x), ..., f_m^+(x))$. Тогда

$$D = \{x : F(x) \le 0\} = \{x : F^+(x) = 0\}.$$

<u>*Теорема 1.*</u> *Пусть дана задача выпуклого программирования* $\min\limits_{x \in D} f(x)$, *где множество* D *задано системой неравенств* $f_i(x) \le 0$, $i = 1, \ldots, m$, *удовлетворяющей условию Слейтера, множество* $D^* \ne \varnothing$, *последовательность* $\{x_k\}$ *построена методом штрафных функций с использованием функции штрафа* P_1. *Тогда является* $\{x_k\}$ *минимизирующей.*

<u>*Доказательство.*</u> Пусть точка $x^* \in D^*$. Тогда по теореме Куна-Таккера найдётся вектор такой, что (x^*, y^*) – седловая точка функции Лагранжа. Таким образом, при всех $x \in E^n$ имеем

$$f(x^*) + \langle y^*, F(x^*)\rangle \le f(x) + \langle y^*, F(x)\rangle.$$

Отсюда и из условий дополняющей нежесткости следует

$$f(x^*) \le f(x) + \langle y^*, F(x)\rangle, \ \forall x \in E^n.$$

Тогда всех $x \in E^n$

$$f\left(x^*\right) \le f(x) + \left\langle y^*, F^+(x)\right\rangle \le f(x) + \left\|y^*\right\| \left\|F^+(x)\right\|. \quad (3)$$

По построению точки x_k и по определению штрафной функции имеем $\Phi_k\left(x_k\right) \le \Phi_k\left(x^*\right) = f\left(x^*\right)$. Отсюда и из формулы (3) получаем оценки

$$f\left(x_k\right) + \gamma_k P_1\left(x_k\right) \le f\left(x^*\right) \le f\left(x_k\right) + \left\|y^*\right\| \left\|F^+\left(x_k\right)\right\| \quad (4)$$

для всех $k = 0, 1, \ldots$ Учитывая, что $P_1\left(x_k\right) = \left\|F^+\left(x_k\right)\right\|^2$, имеем

$$\left\|F^+\left(x_k\right)\right\| \le \frac{\left\|y^*\right\|}{\gamma_k}, \qquad \forall\, k = 0, 1, \ldots (5)$$

Переходя к пределу в неравенстве (5) при $k \to \infty$, получим $\lim\limits_{k\to\infty} F^+\left(x_k\right) = 0$.

Для приближенного решения задачи (2) можно использовать изученные ранее методы безусловной минимизации. В качестве начального приближения $x_{k,0}$ (при нахождении вектора x_k) используется вектор x_{k-1}, то есть $x_{k,0} = x_{k-1}$. Как правило, все точки x_k не принадлежат допустимому множеству. Таким образом, приближение к решению происходит извне множества D (а приближение к минимальному значению f^* осуществляется снизу). Поэтому метод штрафных функций иногда называют ***методом внешней точки***.

4. О недифференцируемой оптимизации

Недифференцируемая оптимизация – это обширная область современной теории экстремальных задач и вычислительной математики. Обсудим возможные подходы к исследованию и ре-

шению негладких экстремальных задач на примере минимизации выпуклых недифференцируемых функций.

Пусть выпуклая функция f определена на E^n и не является всюду дифференцируемой. Несмотря на то, что в любой точке пространства существует производная выпуклой функции f по любому направлениям, в точках недифференцируемости градиент не существует. Поэтому определены различные обобщения этого понятия, которые для некоторых целей могут быть «заменителем» градиента. В частности, этим целям служат понятия ***субградиента*** и ***субдифференциала***.

<u>*Определение 1.*</u> *Вектор* $g(x_0) \in E^n$ *называется* ***субградиентом*** *функции* f *в точке* $x_0 \in E^n$*, если выполняется неравенство*

$$f(x) - f(x_0) \geq \langle g(x_0), x - x_0 \rangle, \ \forall x \in E^n.$$

Множество $\partial f\left(x_0\right)$ *субградиентов называется* ***субдифференциалом*** *функции* f *в точке* x_0.

В любой точке x_0 субдифференциал выпуклой функции f – непустое выпуклое замкнутое и ограниченное множество. Если функция f дифференцируема в точке x_0, то субградиентом является единственный вектор $g\left(x_0\right) = f'\left(x_0\right)$. Субградиенты используются, в частности, для изучения экстремальных свойств выпуклых функ-

ций. В терминах субдифференциалов получены общие необходимые и достаточные условия экстремума. На основе субградиентов построены методы минимизации, обобщающие градиентные методы.

Важной и непростой задачей является вычисление субдифференциала или отдельно взятых субградиентов. Нет простых процедур, позволяющих вычислять субдифференциалы в общем случае. Однако для некоторых классов выпуклых функций построить $\partial f\left(x_0\right)$ и найти какой-либо субградиент $g\left(x_0\right)$ не представляет особой сложности.

В качестве примера рассмотрим ***функцию максимума*** $f(x) = \max\limits_{1 \le i \le m}\left\{ f_i(x)\right\}$. Предположим, что все функции f_i – выпуклые и дифференцируемые.

Обозначим через

$$I(x_0) = \{ i : i = 1,\dots,m;\ f_i(x) = f(x_0) \}.$$

Тогда $\partial f(x_0) = conv\{ f'(x_0) : \ i \in I(x_0) \}$.

В терминах субградиентов необходимое и достаточное условие безусловного минимума выпуклой функции f в точке x^* имеет вид следующего включения:

$$0 \in \partial f(x^*).$$

Познакомимся далее вкратце с ***субградиентным методом*** безусловной минимизации или,

как его еще называют, с ***методом обобщенного градиентного спуска***.

Пусть f – определенная на E^n выпуклая функция, x_0 – произвольный вектор. Начиная с вектора x_0, последовательность приближений строится по правилу

$$x_{k+1} = x_k - t_k\, g(x_k),\ \ k = 0,1,\dots \tag{1}$$

где $g(x_k) \in \partial f(x_k)$, $t_k = \dfrac{\lambda_k}{\| g(x_k) \|}$, числовая последовательность $\{\lambda_k\}$, $k = 0,1,\dots$, такова, что $\lambda_k > 0$, $\lim\limits_{k\to\infty} \lambda_k = 0$ и ряд $\sum\limits_{k=0}^{\infty} \lambda_k$ расходится.

Заметим, что последовательность $\{x_k\}$, генерируемая субградиентным методом, вообще говоря, не является релаксационной. При некоторых дополнительных предположениях о функции f последовательность $\{x_k\}$ – минимизирующая. Последовательность $\{\lambda_k\}$ может находиться по различным правилам. Например, $\lambda_k = \frac{c}{k+1}$, где c – произвольная положительная константа.

Метод прост в реализации (если, конечно, просто вычисляются субградиенты), однако скорость сходимости и достигаемая точность решения существенно зависят от многих факторов и поэтому успешное применение метода не всегда возможно.

Субградиентный метод легко обобщается и для решения условно экстремальных задач. Пусть требуется минимизировать выпуклую функцию f на множестве $D = \{ x : h(x) \le 0 \}$, где h – выпуклая на E^n функция.

Так же, как в только что приведенном методе, последовательность приближений строится, начиная с произвольного вектора x_0, по формуле (1) с той лишь разницей, что вектор $g(x_k)$ является субградиентом функции f, то есть $g(x_k) \in \partial f(x_k)$, если $x_k \in D$, и субградиентом функции h, то есть $g(x_k) \in \partial h(x_k)$, если $x_k \notin D$.

5. Методы одномерной минимизации

Заключительная глава пособия посвящена методам одномерной минимизации. Задача минимизации функций одной переменной актуальна не только как самостоятельная проблема, имеющая многочисленные приложения, но и как вспомогательная задача в рамках различных методов безусловной минимизации и методов решения задачи нелинейного программирования. Например, задача одномерной минимизации возникает при нахождении полного шага в различных вариантах вышеизложенных методов многомерной оптимизации.

Дадим следующие определения.

<u>Определение 1</u>. Функция φ одного действительного переменного определенная на отрезке $\left[a,\, b\right] \subset R$ называется ***унимодальной*** *на $\left[a,\, b\right]$, если существует точка $t^* \in \left[a,\, b\right]$ такая, что функция φ убывает на $\left[a,\, t^*\right)$ и возрастает на $\left(t^*,\, b\right]$.*

Примером унимодальной функции служит строго выпуклая на $\left[a,\, b\right]$ функция.

Если унимодальная функция φ достигает на $\left[a,\, b\right]$ точной нижней грани (например, в случае, когда функция φ непрерывна на ограниченном отрезке $[a, b]$), то только в точке t^*. В этом случае t^* – это единственный локальный минимум функции φ на отрезке $[a, b]$.

<u>*Определение 2.*</u> *Ограниченный отрезок* $[a, b]$, *содержащий единственный локальный минимум функции* φ, *называется* ***отрезком локализации***.

Легко увидеть, что справедлива следующая теорема.

<u>*Теорема 1.*</u> *Пусть функция* φ *унимодальна на отрезке* $[a, b]$, $t_1, t_2 \in [a, b]$ *и* $t_1 < t_2$. *Тогда, если* $\varphi(t_1) < \varphi(t_2)$, *то* $t^* \le t_2$. *Если же* $\varphi(t_1) > \varphi(t_2)$, *то* $t^* \ge t_1$.

Этот факт будет в дальнейшем использоваться в различных методах.

Знакомство с методами одномерного поиска начнем с методов решения задачи минимизации унимодальной функции φ на отрезке локализации $[a, b]$.

5.1. Метод пассивного поиска

Пусть N – натуральное число, точки $a = t_1 < t_2 < ... < t_{N-1} < t_N = b$. Обозначим $\varphi_i = \varphi(t_i)$,

$i = 1, ..., N$. Пусть $\varphi_s = \min\limits_{1 \le i \le N} \varphi(t_i)$. Тогда в силу унимодальности функции φ новым отрезком локализации будет $[t_{s-1}, t_{s+1}]$, если $1 < s < N$; $[t_1, t_2]$, если $s = 1$; и $[t_{N-1}, t_N]$, если $s = N$. В качестве приближённого значения t^* обычно выбирают либо t_s, либо середину полученного отрезка локализации.

Удобно, когда в этом методе узлы t_i – равноотстоящие, то есть $t_{i+1} = t_i + h$, где шаг табуляции $h > 0$. В этом случае легко вычислить значение N, обеспечивающее требуемую точность (длину полученного отрезка локализации).

Таким образом, при пассивном поиске число N точек табуляции определяется заранее, и значение функции вычисляется во всех N точках. Значение функции в каждой текущей точке не влияет на последующие вычисления.

Существуют методы, где, в отличие от методов пассивного поиска, в ходе вычислений используется ранее полученная информация. Таким методам можно дать общее название ***методы последовательного поиска***.

5.2. *Методы последовательного поиска*

Большая группа этих методов основана на

следующей схеме. Строится последовательность вложенных друг в друга отрезков локализации $\left[a_k,\, b_k\right]$, $k = 0,1,\ldots$, где $\left[a_0,\, b_0\right] = \left[a,\, b\right]$. Если $\lim\limits_{k\to\infty}\left(b_k - a_k\right) = 0$, то $\lim\limits_{k\to\infty} a_k = \lim\limits_{k\to\infty} b_k = t^*$. На практике

знание такой последовательности отрезков локализации позволяет локализовать минимум с необходимой точностью. Например, если задано некоторое $\varepsilon > 0$, то при $k = N$ таком, что $b_N - a_N \le \varepsilon$, любую точку $\tilde{t}$ отрезка $\left[a_N,\, b_N\right]$ можно считать приближенным решением задачи. При этом $\left|\tilde{t} - t^*\right| \le \varepsilon$.

Методы последовательного поиска различаются способами построения последовательности отрезков $\left[a_k,\, b_k\right]$, $k = 0,1,\ldots$ В частности, одним из вариантов последовательного поиска является многократное повторение пассивного поиска на получаемых отрезках локализации. Например, ес-

ли на каждом этапе k полагать $N = 5$, то есть $t_1 = a$, $t_5 = b$, $t_3 = \dfrac{t_1 + t_5}{2}$, $t_2 = \dfrac{t_1 + t_3}{2}$, и $t_4 = \dfrac{t_3 + t_5}{2}$, то точки t_i будут делить отрезок локализации на четыре равные части. Следовательно, длина нового отрезка локализации $\left[a_{k+1},\, b_{k+1}\right]$ будет в 2 (а иногда, возможно, и в 4) раза меньше длины отрезка $\left[a_k, b_k\right]$.

Положительным моментом этого метода является высокая скорость уменьшения длины отрезка локализации (на каждой итерации гарантируется ее уменьшение в два раза). Легко вычислить количество итераций N, необходимое для достижения заданной точности. Действительно, так как для любого k выполняется $b_k - a_k = \dfrac{b-a}{2^k}$, то N – это наименьшее значение k, для которого $\dfrac{b-a}{2^k} \le \varepsilon$. Таким образом, $N = \left]\log_2 \dfrac{b-a}{\varepsilon}\right[$. В то же время, отрицательным моментом этого метода является, то, что на каждой итерации необходимо к оставшимся с предыдущей итерации трем точкам t_{s-1}, t_s, t_{s+1}, которые теперь обозначим как t_1, t_3, t_5, добавить новые t_2, t_4 и вычислить значения функции φ_2, φ_4.

Далее мы обсудим возможность построения таких схем построения отрезков локализации, которые позволят на каждой итерации добавлять к

уже известным узлам только лишь один новый. Но прежде мы приведем еще одно свойство унимодальной на отрезке $[a, b]$ функции φ.

<u>*Теорема 1.*</u> *Пусть числа* t_1, t_2 *таковы, что* $a < t_1 < t_2 < b$. *Если* $\varphi_1 < \varphi_2$, *то отрезок* $[a, t_2]$

является отрезком локализации, если же $\varphi_1 > \varphi_2$, *то тогда отрезок* $\left[t_1, b\right]$ *является отрезком локализации.*

Это свойство унимодальных функций используется в различных методах одномерного поиска для построения последовательности отрезков локализации. Приведем основанную на теореме 2 ***принципиальную схему методов минимизации*** унимодальной функции φ, для которой известен отрезок локализации $\left[a_0, b_0\right]$.

Пусть построен отрезок локализации $\left[a_k, b_k\right]$, где $k = 0,1,...$ Опишем способ нахождения отрезка $\left[a_{k+1}, b_{k+1}\right]$. Тем или иным способом находятся числа t_1, t_2 такие, что $a_k < t_1 < t_2 < b_k$. Если $\varphi_1 \le \varphi_2$, то полагаем $a_{k+1} = a_k$, $b_{k+1} = t_2$. Иначе полагаем $a_{k+1} = t_1$, $b_{k+1} = b_k$.

Основное различие методов в рамках этой схемы состоит в разных правилах вычисления точек t_1, t_2. Во многих из этих методов точки t_1, t_2 расположены симметрично относительно середины текущего отрезка локализации. К таким методам относятся, например, метод дихотомии, метод золотого сечения, метод Фибоначчи и другие.

Остановимся коротко на двух из них.

5.2.1. Метод дихотомии

Пусть $\delta > 0$ – достаточно малое число. В этом методе на каждой итерации $t_1 = \frac{a_k + b_k}{2} - \delta$, $t_2 = \frac{a_k + b_k}{2} + \delta$. Как только длина очередного отрезка локализации станет меньше 2δ, построение последовательности прекращается. Таким образом, изложенный алгоритм конечен.

В этом методе длина отрезка локализации на каждой итерации сокращается почти в два раза. (Поэтому в некоторых источниках этот метод называется ***методом деления отрезка пополам***). Это, конечно, является положительным качеством метода. Следует также отметить тот факт, что, согласуя выбор параметра δ с необходимой точностью локализации минимума, легко заранее вычислить количество итераций нужное для нахождения приближенного решения задачи.

Вторым примером реализации изложенной выше схемы является ***метод золотого сечения***.

5.2.2. Метод золотого сечения

<u>Определение 1.</u> Пусть точка $\bar{t} \in (a, b)$ разбивает отрезок $[a, b]$ на две неравные части. Будем говорить, что она осуществляет ***золотое сечение*** *отрезка, если отношение длины меньшего из подотрезков к длине большего равно*

отношению длины большего подотрезка к длине всего отрезка.

Легко увидеть, что если $\bar{t} - a < b - \bar{t}$, то $\bar{t}$ является одним из корней (принадлежащим отрезку $[a, b]$) квадратного уравнения $\dfrac{\bar{t} - a}{b - \bar{t}} = \dfrac{b - \bar{t}}{b - a}$. Этот корень можно вычислить по формуле

$$\bar{t} = a + \frac{3 - \sqrt{5}}{2}(b - a).$$

Второй корень этого уравнения не принадлежит отрезку $[a, b]$ (превышает b).

Очевидно, что существуют две точки золотого сечения отрезка $[a, b]$ симметричные относительно середины отрезка. Обозначим вторую точку золотого сечения $\bar{\bar{t}}$ (очевидно, что $\bar{\bar{t}} \geq \bar{t}$). Точка $\bar{\bar{t}}$ также является одним из корней соответствующего квадратного уравнения и вычисля-ется по формуле $\bar{\bar{t}} = b - \dfrac{3 - \sqrt{5}}{2}(b - a)$. Заметим, что в силу симметрии имеет место равенство $\bar{t} + \bar{\bar{t}} = a + b$, что позволяет, зная одну из точек золотого сечения, вычислить другую (например,
$\bar{\bar{t}} = a + b - \bar{t}$).

Легко показать, что справедливо следующее утверждение.

<u>*Теорема 2.*</u> *Пусть* $\bar{t}$, $\bar{\bar{t}}$ *– точки золотого*

сечения отрезка $[a,b]$, $\overline{t} < \overline{\overline{t}}$. *Тогда* $\overline{\overline{t}}$ *является золотым сечением отрезка* $[\overline{t}, b]$, *а* $\overline{t}$ *– золотым сечением отрезка* $\left[a, \overline{\overline{t}}\right]$.

Итак, метод золотого сечения представляет собой такую реализацию общей схемы, в которой точки t_1, t_2 являются точками золотого сечения текущего отрезка локализации.

Отметим, что в силу теоремы 2 на каждой итерации, кроме начальной, необходимо вычислять только одну точку золотого сечения и значение функции в этой точке. В этом смысле данный метод выгодно отличается от предыдущих, где на каждой итерации вычисляются значения функции φ в двух новых точках t_1, t_2.

Скорости уменьшения длины отрезка локализации в методе золотого сечения несколько ниже, чем в методе дихотомии, а именно

$$b_{k+1} - a_{k+1} = \frac{\sqrt{5}-1}{2}\left(b_k - a_k\right) \approx 0,618\left(b_k - a_k\right).$$

Следовательно, $b_{k+1} - a_{k+1} = \left(\dfrac{\sqrt{5}-1}{2}\right)^{k+1}\left(b_0 - a_0\right)$.

Тем не менее, метод золотого сечения является более эффективным по сравнению с методом дихотомии, поскольку для достижения заданной точности (длины последнего отрезка локализации) здесь требуется меньшее количество вычислений функции.

Однако, заметим, что в формулах для вычисления точек золотого сечения присутствует иррациональность. Поскольку на практике используются приближенные значения $\sqrt{5}$, точки золотого сечения вычисляются с той или иной погрешностью. Накопление погрешности с ростом числа итераций приводит к невозможности достижения заданной точности решения. Существуют различные приемы компенсации накопления погрешности.

К методу золотого сечения близко примыкает так называемый ***метод Фибоначчи***, который, в отличие от метода золотого сечения, является конечным.

5.2.2. Метод парабол

На примере метода парабол познакомимся с методами, основанными на другой идее, а именно, на аппроксимации минимизируемой функции другой, более удобной для минимизации, функцией.

Пусть дифференцируемая функция φ унимодальна на отрезке $[a, b]$.

Определение 2. Назовём тройку чисел t_1, t_2, t_3 ***выпуклой тройкой*** *для функции* φ, *если* $t_1 < t_2 < t_3$ *и*

$$\varphi(t_1) \ge \varphi(t_2), \ \ \varphi(t_3) \ge \varphi(t_2). \qquad (1)$$

Метод парабол (метод квадратичной интерполяции) состоит в следующем. Пусть для функции φ известна выпуклая тройка $t_1, t_2, t_3 \in [a, b]$, в которой хотя бы одно из неравенств (1) является строгим. Найдем минимум интерполяционного полинома второго порядка для функции φ, построенного по узлам t_1, t_2, t_3, по следующей формуле:

$$\tilde{t} = t_2 + \frac{1}{2}\frac{(t_3 - t_2)^2(\varphi_1 - \varphi_2) - (t_2 - t_1)^2(\varphi_3 - \varphi_2)}{(t_3 - t_2)(\varphi_1 - \varphi_2) + (t_2 - t_1)(\varphi_3 - \varphi_2)}. \qquad (2)$$

Полученная точка $\tilde{t}$ принимается за приближенное значение минимума функции φ.

Удобно, когда в этом методе узлы t_i – равноотстоящие, то есть $t_{i+1} = t_i + h$, где $h > 0$. В этом случае из формулы (2) получаем

$$\tilde{t} = t_2 + \frac{h}{2}\frac{\varphi_1 - \varphi_3}{\varphi_1 - 2\varphi_2 + \varphi_3}.$$

Если полученное приближение $\tilde{t}$ – недостаточно точное, то процесс может быть продолжен для новой выпуклой тройки, которая находится с использованием точки $\tilde{t}$. Имеются различные ва-рианты построения новой выпуклой тройки. С одним из них мы познакомимся ниже.

Вышеизложенные методы являются методами нулевого порядка, так как используют только значения минимизируемой функции. Существует также большая группа методов, использующих производные (того или иного порядка) функции φ. Среди них наиболее известны метод касательных, метод Больцано (деления отрезка пополам), метод хорд, метод Ньютона. Однако в силу большой трудоемкости вычисления частных производных они редко используются в рамках методов многомерной минимизации.

5.3. Методы нахождения начального отрезка локализации

Далее будем считать, что минимизируемая функция φ унимодальна на R. Как видно из описания методов одномерной минимизации, для их применения необходимо знать начальный отрезок локализации либо выпуклую тройку точек. Напомним, что задачи одномерной минимизации, возникают, в частности, как задачи поиска полного шага в методах многомерной минимизации. Существуют три случая: полный шаг определяется на отрезке (например, в методе условного градиента), на полупрямой (например, в полношаговом градиентном методе), а также на всей число-вой оси (например, в методах покоординатного спуска).

Наиболее удобным для применения методов одномерного поиска является первый случай, когда известен отрезок $[a,\, b]$, на котором ищется минимум. В этом случае необходимо сначала выяснить, достигается ли минимум внутри отрезка либо на одном из его концов. Например, для дифференцируемой функции, это можно установить по знаку производной $\varphi'(t)$ на концах отрезка. Если окажется, что $t^* \in (a,\, b)$, то отрезок $[a,\, b]$ является отрезком локализации.

В случаях, если одномерный поиск осуществляется на полупрямой или всей числовой оси, для отыскания отрезка локализации можно использовать так называемый ***метод простого перебора***. Метод простого перебора предназначен для поиска отрезка локализации на заданной полупрямой. Если же такая полупрямая неизвестна, сначала необходимо ее определить. Это достигается с помощью следующего простого приема.

Пусть дана точка $t_0 \in R$. Задается $\delta > 0$. Вычисляется точка $v = t_0 + \delta$. Если $\varphi(v) < \varphi(t_0)$, то $t^* > t_0$. Если же $\varphi(v) > \varphi(t_0)$, то полагаем $t_0 = v$ и тогда $t^* < t_0$.

Теперь мы можем описать метод простого перебора.

5.3.1. Метод простого перебора

Предположим, что $t^* > t_0$. Строятся точки $t_{k+1} = t_k + h_k$, $k = 0, 1, \ldots$, где все $h_k > 0$. Положим $\varphi_k = \varphi(t_k)$, $k = 0, 1, \ldots$. Найдем минимальное значение k, для которого $\varphi_{k+1} > \varphi_k$. Обозначим его через s. Легко увидеть, что метод конечен. Предположим, что $s \geq 1$. Тогда в силу унимодальности функции φ отрезок $[t_{s-1}, t_{s+1}]$ является отрезком локализации. Заметим, что при этом точки t_{s-1}, t_s, t_{s+1} образуют выпуклую тройку. Если же $s = 0$, то отрезком локализации является отрезок $[t_0, t_1]$, а в качестве выпуклой тройки можно взять, например, точки $t_0, \dfrac{t_0 + t_1}{2}, t_1$.

В случае, когда $t^* < t_0$, точки t_k строятся по формуле $t_{k+1} = t_k - h_k$.

Варианты метода простого перебора отличаются друг от друга выбором шага табуляции h_k. В частности, используется формула $h_{k+1} = \alpha h_k$, где $\alpha \geq 1$. Если $\alpha = 1$, то получаем вариант метода с постоянным шагом h. Успех применения метода в этом случае полностью определяется удачным выбором значения h. Для ускорения поиска отрезка локализации, как пра-вило, выбирают $\alpha > 1$. Например, если

$\alpha = \dfrac{1+\sqrt{5}}{2}$, то t_s является точкой золотого сечения отрезка $\left[t_{s-1}, t_{s+1}\right]$. Такой вариант метода отыскания отрезка локализации удобно сочетать с последующим применением метода золотого сечения.

Если же предполагается использование метода парабол, то удобно полагать $\alpha = 2$. В этом случае после нахождения отрезка локализации $\left[t_{s-1}, t_{s+1}\right]$ вычисляют точку $w = \dfrac{t_s + t_{s+1}}{2}$. Из полученных четырёх равноотстоящих точек выбирают выпуклую тройку. Если $\varphi_s > \varphi(w)$, то точки t_s, w, t_{s+1} образуют выпуклую тройку, если же $\varphi_s < \varphi(w)$, то берутся точки t_{s-1}, t_s, w. Такой прием нахождения начальной выпуклой тройки используется в ***методе Дэвиса, Свенна и Кэмпи*** (***ДСК***).

5.3.2. Метод ДСК

В методе ДСК на основе выпуклой тройки t_1, t_2, t_3 (найденной по описанному выше правилу) с помощью метода парабол вычисляется точка $\tilde{t}$ (минимум квадратного трехчлена), после чего, если не достигнута заданная точность, отыскивается новая выпуклая тройка. Поскольку расположение искомого минимума t^* относительно $\tilde{t}$ неизвестно, поиск осуществляют на всей числовой оси в соответствии с описанным выше вариантом метода простого перебора. При этом в качестве точки $t_0 \in R$ выбирается $\tilde{t}$, а начальное значение шага h_0 заменяется на $\frac{h_0}{m+1}$, где m – некоторое натуральное число. Описанная процедура повторяется до достижения требуемой точности.

Изложенные в этом параграфе методы предполагают унимодальность функции. Если же минимизируемая функция не унимодальна, то для ее минимизации следует применять другие методы, например, метод ломаных.