Министерство образования Республики Беларусь
УО «Витебский государственный технологический университет»

Тема6. «Методы оптимизации.»

Кафедра теоретичской и прикладной математики.

разработана доц. Е.Б.Дуниной

6.1 Основные понятия.

Под оптимизацией понимают процесс выбора наилучшего варианта из всех возможных.

В процессе решения задачи оптимизации обычно необходимо найти оптимальные значения некоторых параметров определяющих данную задачу.

При решении инженерных задач их принято называть *проектными параметрами,* а в экономических задачах их обычно называют *параметрами плана.*

В качестве проектных параметров могут быть, в частности, значения линейных размеров объекта, массы, температуры и т.д.

Число *n* проектных параметров характеризует размерность задачи оптимизации.

Выбор оптимального решения или сравнение двух альтернативных решений проводится с помощью некоторой зависимой величины (функции), определяемой проектными параметрами.

Эта величина называется *целевой функцией* (или *критерием качества*).

В процессе решения задачи оптимизации должны быть найдены такие значения проектных параметров, при которых целевая функция имеет минимум (или максимум).

Целевую функцию можно записать в виде

$$u = f(x_1, x_2, \ldots x_n). \quad (6.1)$$

Примерами целевой функции являются прочность или масса конструкции, мощность установки, объем выпуска продукции, стоимость перевозок грузов, прибыль и т.д.

В случае одного проектного параметра $(n = 1)$

целевая функция (6.1) является функцией одной переменной, и ее график – некоторая кривая на плоскости.

При $n = 2$ **целевая функция является функцией двух переменных, и ее графиком является поверхность.**

Можно выделить два типа задач оптимизации – безусловные и условные.

Безусловная задача **оптимизации состоит в отыскании максимума или минимума действительной функции (6.1) от *n* действительных переменных и определении соответствующих значений аргументов на некотором множестве.**

Условные задачи **оптимизации или задачи с *ограничениями*, - это такие, при формулировке которых задаются некоторые условия (ограничения). Эти ограничения задаются совокупностью некоторых функций, удовлетворяющих уравнениям или неравенствам.**

Число *m* ограничений-равенств может быть произвольным и их можно записать в виде

$$g_1(x_1, x_2, ..., x_n) = 0$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$g_m(x_1, x_2, ..., x_n) = 0 \qquad (6.2)$$

В ряде случаев из этих соотношений можно выразить одни проектные параметры через другие.

Это позволяет исключить некоторые параметры из процесса оптимизации, что приводит к уменьшению размерности задачи и облегчает ее решение.

Аналогично могут вводится также ограничения-неравенства, имеющие вид

$$a_1 \le \varphi_1(x_1, x_2, ..., x_n) \le b_1$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$a_k \le \varphi_k(x_1, x_2, ..., x_n) \le b_k \qquad (6.3)$$

Пример.

Пусть требуется спроектировать контейнер в форме прямоугольного параллелепипеда объемом

$V = 1 м^3,$ причем желательно израсходовать на его изготовление как можно меньше материала.

Решение.

При постоянной толщине стенок последнее условие означает, что площадь полной поверхности контейнера S должна быть минимальной.

Если обозначить через x_1, x_2, x_3 длины ребер контейнера, то задача сводится к минимизации функции

$$S = 2(x_1x_2 + x_2x_3 + x_1x_3) \quad (6.4)$$

Эта функция в данном случае является целевой, а условие

$$V = 1$$

- ограничением-равенством, которое позволяет исключить один параметр:

$$V = x_1x_2x_3 = 1$$

$$x_3 = \frac{1}{x_1x_2}$$

$$S = 2\left(x_1x_2 + \frac{1}{x_1} + \frac{1}{x_2}\right) \quad (6.5)$$

Задача сводится к минимизации функции двух переменных.

В результате решения задачи будут найдены значения проектных параметров

$x_1, x_2,$ **а затем и** x_3.

В приведенном примере фактически получилась задача безусловной оптимизации для целевой функции (6.5).

Вместе с тем можно рассматриваемую задачу усложнить и поставить дополнительные условия.

Например, потребуем, чтобы данный контейнер имел длину не менее 2м.

Это условие запишется в виде ограничения-неравенства на один из параметров, например

$$x_1 \geq 2 \quad (6.6)$$

Т.е. мы получили следующую условную задачу оптимизации: минимизируя функцию (6.5) и учитывая ограничение – неравенство (6.6), найти оптимальные значения параметров плана x_1, x_2

$$(x_1 \geq 0, x_2 \geq 0).$$

6.2 Одномерная оптимизация. Методы поиска.

Надо найти наименьшее (или наибольшее) значение целевой функции $y = f(x)$,

заданной на множестве σ,

и определить значение проектного параметра $x \in \sigma$,

при котором целевая функция принимает экстремальное значение.

Будем предполагать, что целевая функция *унимодальна,* т.е. на данном отрезке она имеет только один минимум.

Процесс решения задачи состоит в последовательном сужении интервала изменения проектного параметра, называемого *интервалом неопределенности.*

В начале процесса оптимизации его длина равна

$b - a$, **а к концу она должна стать менее заданного допустимого значения** ε.

Наиболее простым способом сужения интервала неопределенности является деление его на некоторое число равных частей с последующим вычислением целевой функции в точках разбиения.

Пусть *n* число элементарных отрезков,

$h = (b - a) / n$ **-шаг разбиения.**

Вычислим значения целевой функции $y_k = f_k(x)$

в узлах $x_k = a + kh$ $(k = 0, 1, \ldots, n).$

Сравнивая полученные значения $f_k(x),$

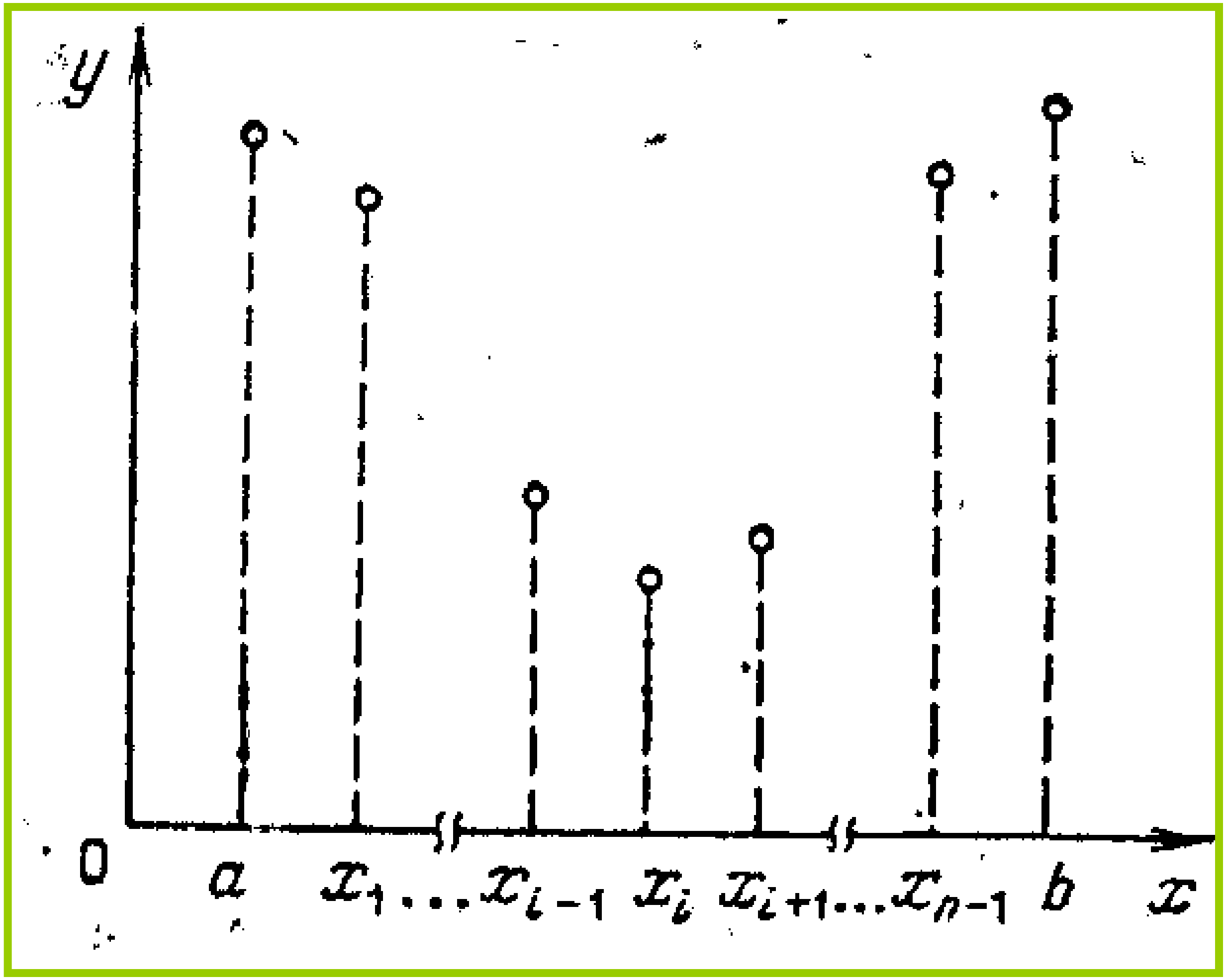
найдем среди них наименьшее $y_i = f_i(x).$

Число $m_n = y_i$ **можно приближенно принять за наименьшее значение целевой функции** $f(x)$ **на отрезке** $[a,b]$.

Очевидно, что близость m_n **к минимуму** m **зависит от числа точек, и для непрерывной функции** $f(x)$

$$\lim_{n\to\infty} m_n = m,$$

т.е. с увеличением числа точек разбиения погрешность в определении минимума стремится к нулю.



В данном методе, который можно назвать *методом перебора*, основная трудность состоит в выборе *n* и оценке погрешности.

Более экономичным способом уточнения оптимального параметра является использование свойства унимодальности целевой функции, которое позволяет построить процесс сужения интервала неопределенности.

Пусть, как и ранее, среди всех значений унимодальной функции $y = f(x)$,

вычисленной в узлах x_k, **наименьшим оказалось** y_i.

Это означает, что оптимальное значение проектного параметра находится на отрезке

$[x_{i-1}, x_{i+1}]$, **т.е. интервал неопределенности сузился до длины двух шагов.**

Если размер интервала недостаточен для удовлетворения заданной погрешности, то его снова можно уменьшить путем нового разбиения и т. д.

Процесс оптимизации продолжается до достижения заданного размера интервала неопределенности.

В описанном методе общего поиска можно с помощью выбора разумного шага разбиения добиться эффективного поиска.

Например, пусть начальная длина интервала неопределенности равна $b - a = 1.$

Нужно добиться его уменьшения в 100 раз.

Это легко достичь разбиением интервала на 200 частей.

Вычислив значение целевой функции $f(x_k)$

$(k = 0,1,...,200),$ **найдем ее минимальное значение**

$f(x_i).$

Тогда искомым интервалом неопределенности будет отрезок $[x_{i-1}, x_{i+1}].$

Однако можно поступить и иначе.

Сначала разобьем отрезок $[a, b]$ **на 20 частей**

и найдем интервал неопределенности длиной 0.1, при этом вычислим значения целевой функции в точках

$$x_k = a + 0.05k \qquad (k = 0,1,...,20).$$

Теперь отрезок $[x_{i-1}, x_{i+1}]$

снова разобьем на 20 частей; получим искомый интервал длиной 0.01, причем значения целевой функции вычисляем в точках

$$x_h = x_{i-1} + 0.005k \qquad (k = 1,2,...,19)$$

В точках x_{i-1} **и** x_{i+1} **значения** $f(x)$ **уже найдены.**

Таким образом,во втором случае в процессе оптимизации произведено 40 вычислений значений целевой функции против 201 в первом случае, т.е. способ разбиения позволяет получить существенную экономию вычислений.

6.3 Метод золотого сечения.

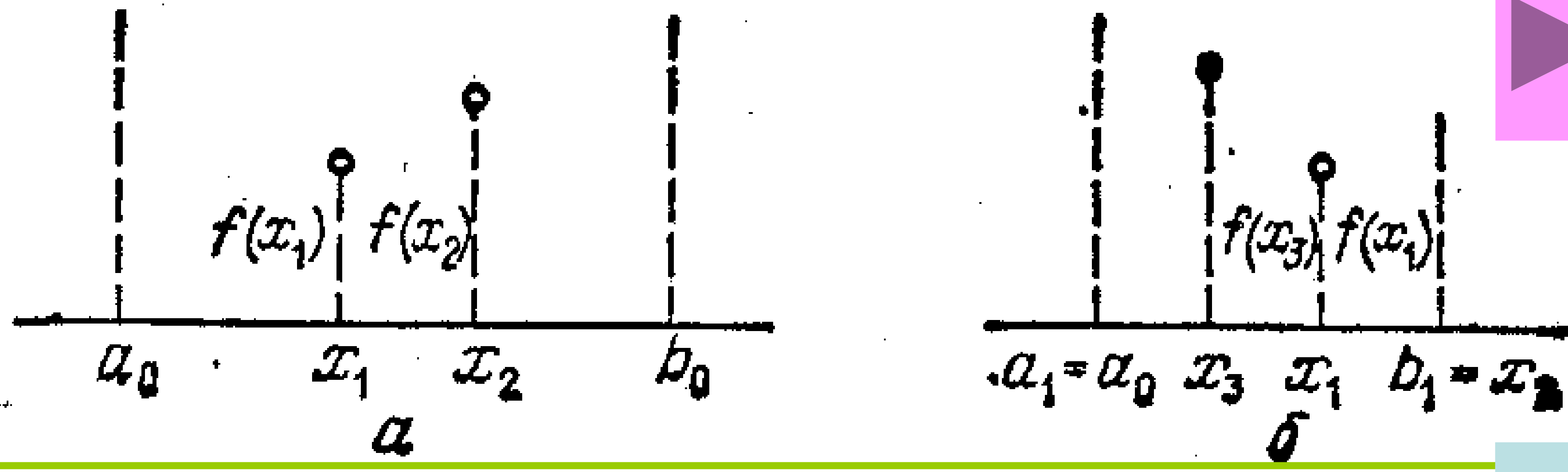
Метод состоит в построении последовательности отрезков $[a_0, b_0]$, $[a_1, b_1], \ldots,$

стягивающихся к точке минимума функции $f(x)$.

На каждом шаге, за исключением первого, вычисление значения функции $f(x)$

проводится лишь один раз.

Эта точка, называемая золотым сечением, выбирается специальным образом.



На первом шаге процесса оптимизации внутри отрезка $[a_0, b_0]$ **(рис.а)**

выбираем две внутренние точки x_1 **и** x_2

и вычисляем значения целевой функции $f(x_1)$ **и** $f(x_2)$. **Поскольку в данном случае**

$$f(x_1) < f(x_2),$$

очевидно, что минимум расположен на одном из прилегающих к x_1 отрезков $[a_0, x_1]$ или $[x_1, x_2]$.

Поэтому отрезок $[x_2, b_0]$ **можно отбросить, сузив тем самым первоначальный интервал неопределенности.**

Второй шаг проводим на отрезке $[a_1, b_1]$, где

$$a_1 = a_0, b_1 = x_2.$$

Нужно снова выбрать две внутренние точки, но одна из них x_1 осталась из предыдущего шага, поэтому достаточно выбрать лишь одну точку $x_3,$ вычислить значение $f(x_3)$ и провести сравнение.

Поскольку здесь $f(x_3) > f(x_1)$ (рис.б), ясно, что минимум находится на отрезке $[x_3, b_1].$ Обозначим этот отрезок $[a_2, b_2],$ снова выберем внутреннюю точку и повторим процедуру сужения интервала неопределенности.

Процесс оптимизации повторяется до тех пор, пока длина очередного отрезка $[a_n, b_n]$ не станет меньше заданной величины $\varepsilon.$

Рассмотрим способ размещения внутренних точек на каждом отрезке $[a_k, b_k]$.

Пусть длина интервала неопределенности равна l,

а точка деления делит его на части l_1, l_2 :

$$l_1 > l_2, \qquad l = l_1 + l_2.$$

Золотое сечение **интервала неопределенности выбирается так, чтобы отношение длины большего отрезка к длине всего интервала равнялось отношению длины меньшего отрезка к длине большего отрезка:**

$$\frac{l_1}{l} = \frac{l_2}{l_1} \quad (6.7)$$

Из этого соотношения можно найти точку деления, определив отношение $l_2 / l_1.$

Преобразуем выражение (6.7) и найдем это значение:

$$l_1^2 = l_2 l,$$

$$l_1^2 = l_2 (l_1 + l_2),$$

$$l_2^2 + l_2 l_1 - l_1^2 = 0,$$

Разделим на $l_1^{\,2}$

$$\left(\frac{l_2}{l_1}\right)^2 + \frac{l_2}{l_1} - 1 = 0,$$

Пусть $t = \dfrac{l_2}{l_1},$ **тогда**

$$t^2 + t - 1 = 0.$$

Находим корни

$$t_{1,2} = \frac{-1 \pm \sqrt{1 - 4 \cdot (-1)}}{2} = \frac{-1 \pm \sqrt{5}}{2}.$$

Или

$$\frac{l_2}{l_1} = \frac{-1 \pm \sqrt{5}}{2}$$

Поскольку нас интересует только положительное решение, то

$$\frac{l_2}{l_1} = \frac{l_1}{l} = \frac{-1 + \sqrt{5}}{2} \approx 0.618$$

Отсюда

$$l_1 \approx 0.618 l, \qquad l_2 \approx 0.382 l.$$

Поскольку заранее неизвестно, в какой последовательности

$$(l_1 \text{ и } l_2 \text{ или } l_2 \text{ и } l_1)$$

делить интервал неопределенности, то

рассматривают внутренние точки, соответствующие двум этим способам деления.

На рис. а) точки деления x_1 и x_2

выбираются с учетом полученных значений для частей отрезка. В данном случае имеем

$$x_1 - a_0 = b_0 - x_2 = 0.382 d_0,$$

$$b_0 - x_1 = x_2 - a_0 = 0.618 d_0,$$

$$d_0 = b_0 - a_0$$

После первого шага оптимизации получается новый интервал неопределенности – отрезок $[a_1, b_1]$ (см. рис б). Можно показать, что точка x_1

делит этот отрезок в требуемом отношении, при этом

$$b_1 - x_1 = 0.382 d_1,$$

$$d_1 = b_1 - a_1$$

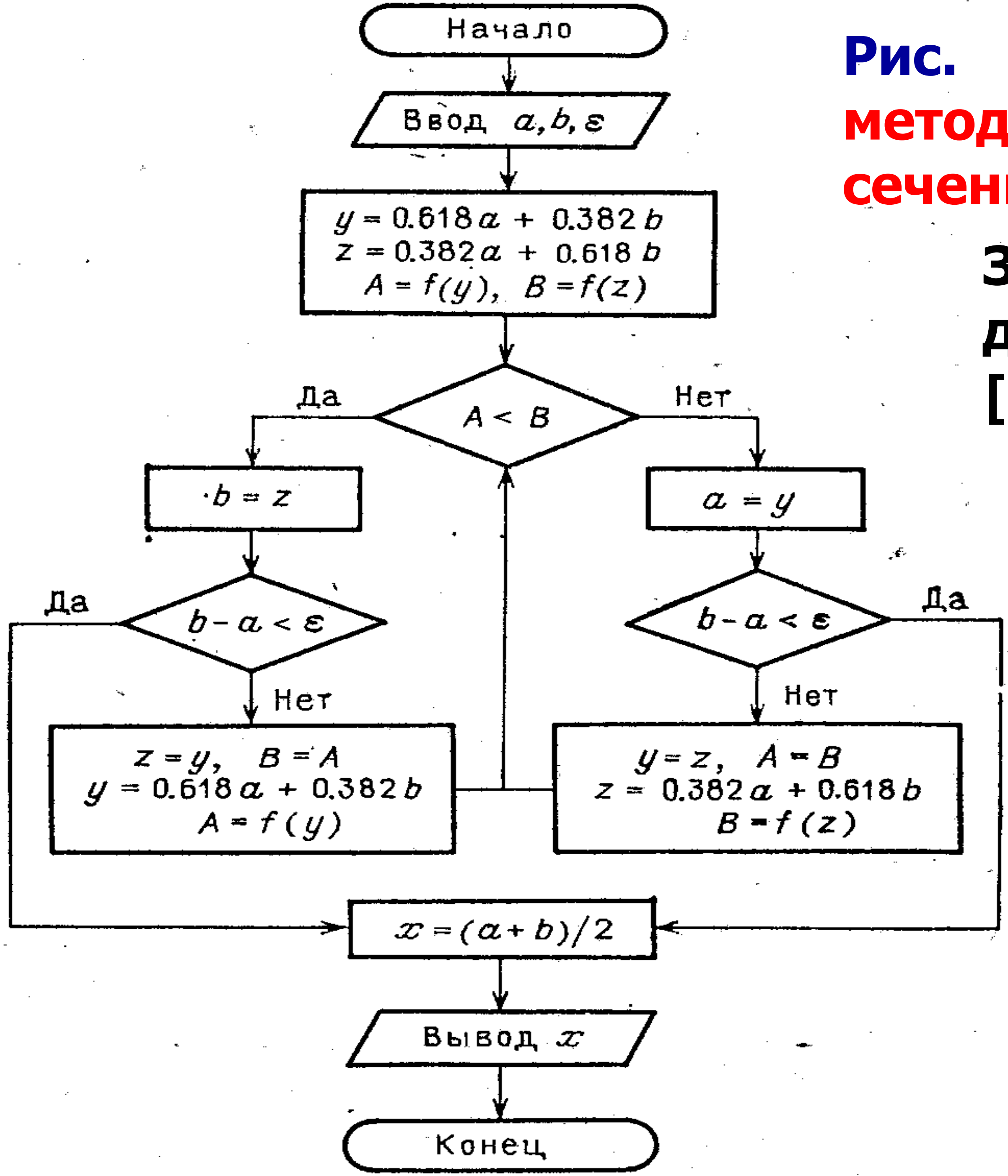


Рис. Блок-схема метода золотого сечения

Здесь *y*,*z* – точки деления отрезка [*a*,*b*], причем *y*<*z*.

Для этого проведем очевидные преобразования:

$$b_1 - x_1 = x_2 - x_1 = (b_0 - a_0) - (x_1 - a_0) - (b_0 - x_2) =$$
$$= \mathrm{d}_0 - 0.382d_0 - 0.382d_0 = 0.236d_0$$

$$d_1 = x_2 - a_0 = 0.618d_0$$

$$b_1 - x_1 = 0.236(d_1 / 0.618) = 0.382d_1$$

Вторая точка деления x_3

выбирается на таком же расстоянии от левой границы отрезка, т.е.

$$x_3 - a_1 = 0.382d_1.$$

И снова интервал неопределенности уменьшается до размера

$$d_2 = b_2 - a_2 = b_1 - x_3 = 0.618 d_1 = 0.618^2 d_0.$$

Используя полученные соотношения, можно записать координаты точек деления *y* и *z* отрезка

$[a_k, b_k]$ **на *k+1* шаге оптимизации (*y*<*z*):**

$$y = 0.618 a_k + 0.382 b_k$$

$$z = 0.382 a_k + 0.618 b_k. \quad (6.8)$$

При этом длина интервала неопределенности равна

$$d_k = b_k - a_k = 0.618^k d_0. \quad (6.9)$$

Процесс оптимизации заканчивается при выполнении условия $d_k < \varepsilon.$

При этом проектный параметр оптимизации составляет $a_k < x < b_k$

Можно в качестве оптимального значения принять $x = a_k$ (или $x = b_k,$ или $x = (a_k + b_k)/2$ и т. д.).

Пример.

Для оценки сопротивления дороги движению автомобиля при скорости υ км/ч можно использовать эмпирическую формулу

$$f(\upsilon) = 24 - \frac{2}{3}\upsilon + \frac{1}{30}\upsilon^2$$ (для шоссе).

Определить скорость, при которой сопротивление будет минимальным.

Решение.

Это простейшая задача одномерной оптимизации. Здесь сопротивление

$$f(\upsilon)$$ **- целевая функция,**

скорость υ **- проектный параметр.**

Данную задачу легко решить путем нахождения минимума с помощью вычисления производной.

Действительно,

$$f'(\upsilon) = -\frac{2}{3} + \frac{2\upsilon}{30} = 0,$$

$$\upsilon = 10 км/ч$$

Проиллюстрируем на этой простейшей задаче метод золотого сечения. Первоначально границы интервала неопределенности примем равными

$$a = 5, b = 20.$$

Результаты вычислений представим в виде таблицы. Обозначения аналогичны используемым в блок-схеме, погрешность

$$\varepsilon = 1 км / ч.$$

Проведем решение для первого этапа:

$$y = 0.618 \cdot 5 + 0.382 \cdot 20 \approx 10.7$$

$$z = 0.382 \cdot 5 + 0.618 \cdot 20 \approx 14.3$$

$$A = 24 - \frac{2}{3} \cdot 10.7 + \frac{1}{30} \cdot 10.7^2 \approx 20.7$$

$$B = 24 - \frac{2}{3} \cdot 14.3 + \frac{1}{30} \cdot 14.3^2 \approx 21.3$$

$$A < B$$

Шаг	*a*	y	z	*b*	*A*	*B*	*b-a*
1	5	10.7	14.3	20	20.7	21.3	15
2	5	8.6	10.7	14.3	20.73	20.68	9.3
3	8.6	10.7	12.1	14.3	20.68	20.81	5.7
4	8.6	9.9	10.7	12.1	20.66	20.68	3.5
5	8.6	9.4	9.9	10.7	20.68	20.66	2.1
6	9.4			10.7			1.3

При невысокой точности вычислений достаточно четырех шагов оптимизации.

В этом случае искомое значение скорости равно

$$\upsilon = (8.6 + 10.7)/2 = 9.65 км/ч.$$

После пяти шагов этот результат получается с меньшей погрешностью:

$$\upsilon = (9.4 + 10.7)/2 = 10.07 км/ч.$$

6.4 Многомерные задачи оптимизации. Метод покоординатного спуска.

В большинстве реальных задач оптимизации, целевая функция зависит от многих проектных параметров.

Минимум дифференцируемой функции многих переменных

$$u = f(x_1, x_2, \ldots, x_n)$$

можно найти, исследуя ее значения в критических точках, которые определяются из решения системы дифференциальных уравнений

$$\frac{\partial f}{\partial x_1} = 0, \frac{\partial f}{\partial x_2} = 0, \ldots, \frac{\partial f}{\partial x_n} = 0.$$

Во многих случаях никакой формулы для целевой функции нет, а имеется лишь возможность определения ее значений в произвольных точках рассматриваемой области с помощью некоторого вычислительного алгоритма или путем физических измерений.

Задача состоит в приближенном определении наименьшего значения функции во всей области при известных ее значениях в отдельных точках.

Для решения подобной задачи в области проектирования G,

в которой ищется минимум целевой функции

$$u = f(x_1, x_2, \dots, x_n),$$

можно ввести дискретное множество точек (узлов) путем разбиения интервалов изменения параметров

$x_1, \dots, x_n$ **на части с шагами** $h_1, \dots, h_n$.

В полученных узлах можно вычислить значение целевой функции и среди этих значений найти наименьшее.

Однако, в многомерных задачах оптимизации, где число проектных параметров достигает пяти и более, этот метод требует большого объема вычислений.

Рассмотрим метод покоординатного спуска.

Требуется найти наименьшее значение целевой функции

$$u = f(x_1, x_2, ..., x_n).$$

В качестве начального приближения выберем в *n-мерном* пространстве некоторую точку

$$M_0(x_1^{(0)}, x_2^{(0)}, x_n^{(0)}).$$

Зафиксируем все координаты функции *u*, кроме первой. Тогда

$$u = f(x_1, x_2^{(0)}, ..., x_n^{(0)})$$

-функция одной переменной x_1.

Решая одномерную задачу оптимизации для этой функции, мы от точки M_0

переходим к точке $M_1(x_1^{(1)}, x_2^{(0)}, ..., x_n^{(0)})$,

в которой функция *u* принимает наименьшее значение по координате x_1

при фиксированных остальных координатах.

В этом состоит первый шаг процесса оптимизации, состоящий в спуске по координате x_1.

Зафиксируем теперь все координаты, кроме x_2,

и рассмотрим функцию этой переменной $u = f(x_1^{(1)}, x_2, ..., x_n^{(0)})$.

Снова решая одномерную задачу оптимизации, находим ее наименьшее значение при $x_2 = x_2^{(1)}$,

т.е. в точке $M_2(x_1^{(1)}, x_2^{(1)}, ..., x_n^{(0)})$.

Аналогично проводится спуск по координатам

$x_3, x_4, ..., x_n$, **а затем прцедура снова повторяется**

от x_1 **до** x_n **и т.д.**

В результате этого процесса получается последовательность точек $M_0, M_1, ...,$

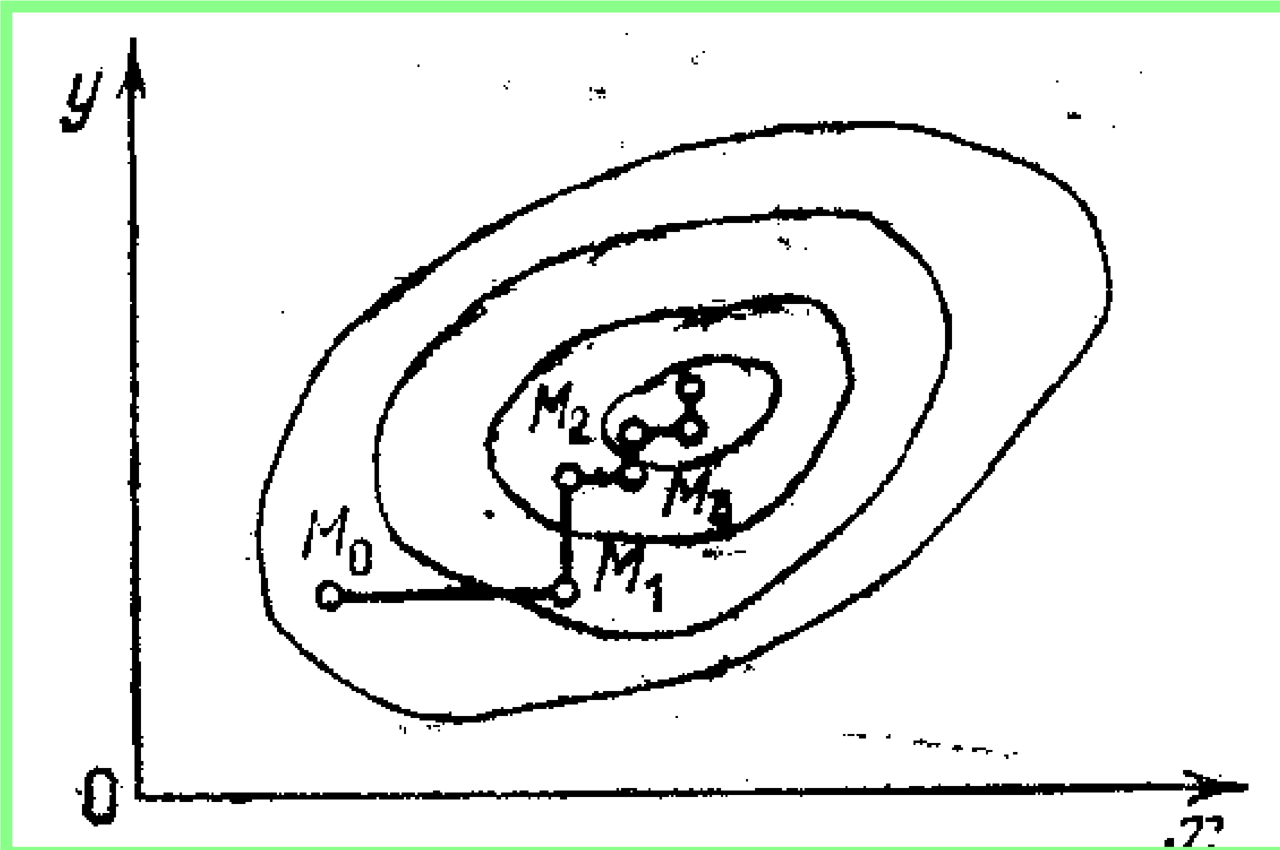
в которых значения целевой функции составляют монотонно убывающую последовательность

$$f(M_0) \geq f(M_1) \geq ....$$

На любом *k*-м шаге этот процесс можно прервать, и значение $f(M_k)$ **принимается в качестве наименьшего значения целевой функции в рассматриваемой области.**

Таким образом, метод покоординатного спуска сводит задачу о нахождении наименьшего значения функции многих переменных к многократному решению одномерных задач оптимизации по каждому проектному параметру.

Данный метод легко проиллюстрировать геометрически для случая функции двух переменных $z = f(x, y)$, **описывающей некоторую поверхность в трехмерном пространстве.**



Точка $M_0(x_0, y_0)$ **описывает начальное приближение.**

Проводя спуск по координате *x*, попадаем в точку

$$M_1(x_1, y_0).$$

Далее, двигаясь параллельно оси ординат, придем в точку $M_2(x_1, y_1)$ **и т.д.**

6.5 Метод градиентного спуска.

Направление наискорейшего спуска соответствует направлению наибольшего убывания функции.

Направление наибольшего возрастания функции двух переменных $u = f(x, y)$

характеризуется ее *градиентом*

$$grad\ u = \frac{\partial u}{\partial x}\vec{e}_1 + \frac{\partial u}{\partial y}\vec{e}_2,$$

где $\vec{e}_1, \vec{e}_2$ **- единичные векторы в направлении координатных осей.**

Следовательно, направление, противоположное градиентному, укажет путь, ведущий вниз вдоль наиболее крутой линии.

Методы, основанные на выборе пути оптимизации с помощью градиента, называются градиентными.

Идея метода градиентного спуска состоит в следующем.

Выбираем начальную точку и вычисляем в ней градиент рассматриваемой функции.

Делаем шаг в направлении, обратном градиентному.

В результате приходим в точку, значение функции в которой меньше первоначального.

Если это условие не выполнено, т.е. значение функции не изменилось либо даже возросло, то нужно уменьшить шаг.

В новой точке процедуру повторяем: вычисляем градиент и снова делаем шаг в обратном к нему направлении.

Процесс продолжается до получения наименьшего значения целевой функции.

Момент окончания поиска наступит тогда, когда движение из полученной точки с любым шагом приводит к возрастанию целевой функции.

В описанном методе требуется вычислять на каждом шаге оптимизации градиент целевой функции $f(x_1, x_2, \ldots, x_n)$

$$grad f = \left\{ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right\}.$$

Формулы для частных производных можно получить в явном виде лишь в том случае, когда целевая функция задана аналитически.

В противном случае эти производные вычисляются с помощью численного дифференцирования:

$$\frac{\partial f}{\partial x_i} \approx \frac{1}{\Delta x_i}[f(x_1, \ldots, x_i + \Delta x_i, \ldots, x_n) - f(x_1, \ldots, x_i, \ldots, x_n)], i = 1, 2, \ldots, n.$$

При использовании градиентного спуска основной объем вычислений приходится обычно на вычисление градиента целевой функции в каждой точке траектории спуска.

Иногда применяют модифицированный метод наискорейшего спуска.

Согласно этому методу, после определения в начальной точке направления, противоположного градиенту целевой функции, в этом направлении делают не один шаг, а двигаются до тех пор, пока целевая функция убывает, достигая таким образом минимума в некоторой точке.

В этой точке снова определяют направление спуска (с помощью градиента) и ищут новую точку минимума целевой функции и т.д.

7. Задачи с ограничениями

7.1 Метод штрафных функций

Пусть $f(x_1, x_2, \ldots, x_n)$ **-целевая функция,**

для которой нужно найти минимум m **в ограниченной области**

$$D(x_1, x_2, \ldots, x_n \in D).$$

Данную задачу заменяем задачей с безусловной минимизации однопараметрического семейства функций

$$F(x, \beta) = f(x) + \frac{1}{\beta}\varphi(x),$$

$$x = \{x_1, x_2, \ldots, x_n\} \quad (7.1)$$

При этом дополнительную (штрафную) функцию

$\varphi(x)$ **выберем таким образом, чтобы при** $\beta \to 0$

решение вспомогательной задачи стремилось к решению исходной или, по крайней мере, чтобы их минимумы совпадали:

$\min F(x, \beta) \to m$ **при** $\beta \to 0.$

Штрафная функция $\varphi(x)$ **должна учитывать ограничения, которые задаются при постановке задачи оптимизации.**

В частности, если имеются ограничения-неравенства вида $g_j(x_1, x_2, \ldots, x_n) \geq 0,$

то в качестве штрафной можно взять функцию, которая:

1) равна нулю во всех точках пространства проектирования,

2) стремиться к бесконечности в тех точках, в которых эти неравенства не выполняются.

Таким образом, при выполнении ограничений-неравенств функции $f(x)$ и $F(x, \beta)$

имеют один и тот же минимум.

Если хотя бы одно неравенство не выполнится, то вспомогательная целевая функция $F(x, \beta)$

получает бесконечно большие добавки, и ее значения далеки от минимума функции $f(x)$.

Другими словами, при несоблюдении ограничений-неравенств налагается «штраф». Отсюда и термин «метод штрафных функций».

Рассмотрим блок-схему метода штрафных функций.

В качестве исходных данных вводятся начальное приближение искомого вектора $x^* = \{x_1^*, x_2^*, \ldots, x_n^*\}$,

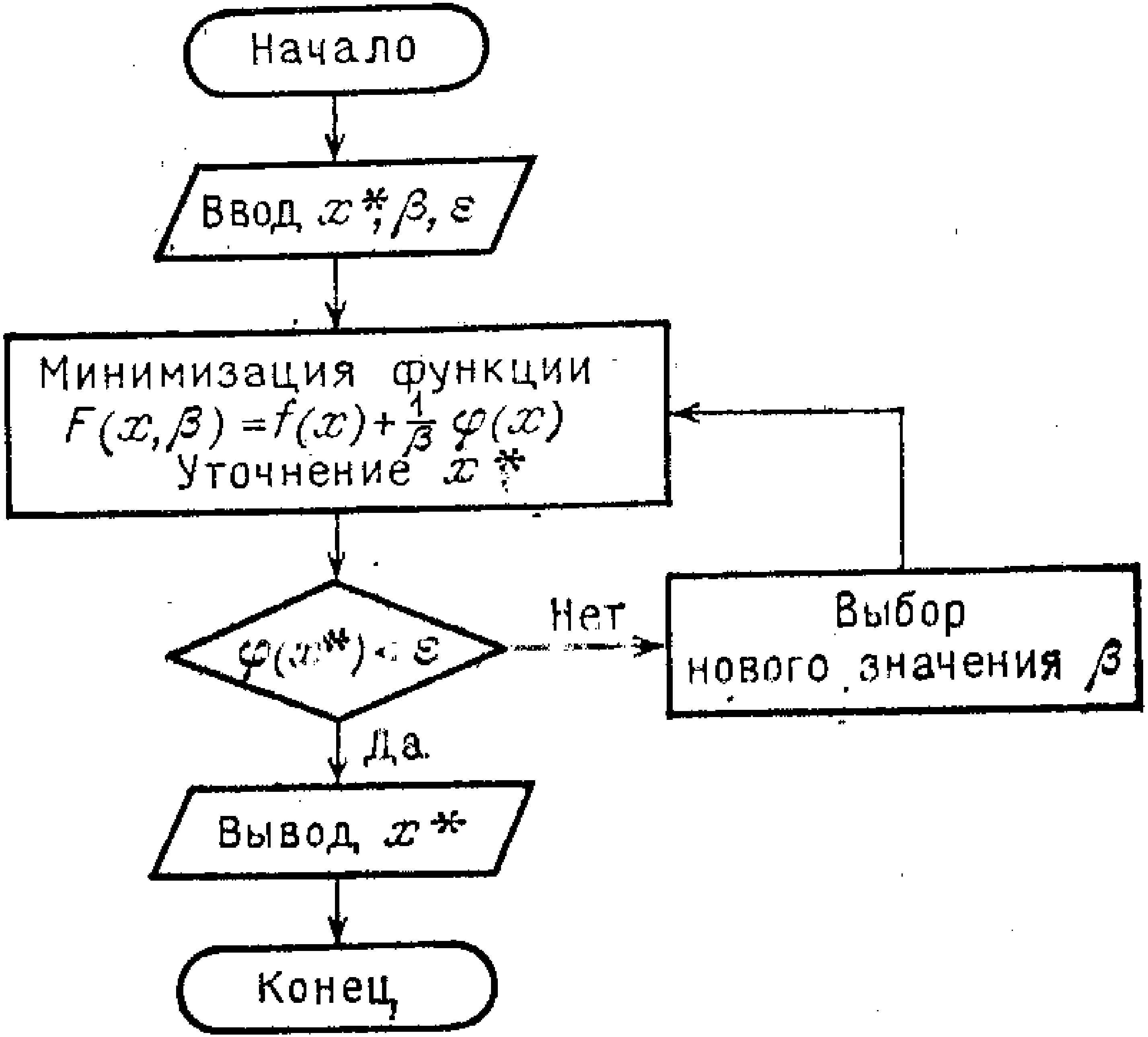
начальное значение параметра β

и некоторое малое число ε,

характеризующее точность расчета.

На каждом шаге итерационного процесса определяется оптимальное значение x^* **вектора** X,

при этом в качестве начального приближения принимается результат предыдущей итерации.


Начало
Ввод x^*, β, ε
Минимизация функции
$F(x,\beta) = f(x) + \frac{1}{\beta}\varphi(x)$
Уточнение x^*
$\varphi(x^*) < \varepsilon$
Нет
Выбор
нового значения β
Да.
Вывод x^*
Конец

7.2 Линейное программирование

Линейное программирование изучает задачи оптимизации, в которых целевая функция является линейной функцией проектных параметров, а ограничения задаются в виде линейных уравнений и неравенств.

Стандартная (каноническая) постановка задачи линейного программирования формулируется следующим образом:

найти значения переменных $x_1, x_2, \ldots, x_n,$ **которые:**

1) удовлетворяют системе линейных уравнений

$$
\begin{cases}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m
\end{cases}
\qquad (7.2)
$$

2) являются неотрицательными, т.е.

$$
x_1 \geq 0, x_2 \geq 0, \ldots, x_n \geq 0 \quad (7.3)
$$

3) обеспечивают наименьшее значение линейной целевой функции

$$
f(x_1, x_2, \ldots, x_n) = c_0 + c_1x_1 + c_2x_2 + \ldots + c_nx_n \quad (7.4)
$$

Всякое решение системы уравнений (7.2), удовлетворяющее системе неравенств (7.3), называется допустимым решением.

Допустимое решение, которое минимизирует целевую функцию (7.4), называется оптимальным решением.