

МНОГОМЕРНАЯ УСЛОВНАЯ ОПТИМИЗАЦИЯ

5 Метод линейной аппроксимации

Рассмотрим задачу оптимизации

$$\min_{\bar{X} \in D \subset \mathbb{R}^n} \Phi(\bar{X}) = \Phi(\bar{X}^*) = \Phi^*$$

Пусть множество допустимых значений D определяется *только ограничениями типа неравенств*

$$D = \{\bar{X} \mid g_j(\bar{X}) \geq 0, j \in [1, \dots, l]\}$$

и функции Φ и g_j являются непрерывными, дифференцируемыми и выпуклыми.

Суть метода линейной аппроксимации

На каждой итерации строится линейная аппроксимация целевой функции $\Phi(X)$ и ограничивающих функций g_j в окрестности текущей точки X^r

$$\tilde{\Phi}^r(\bar{X}) = \Phi(\bar{X}^r) + (\nabla \Phi(\bar{X}^r), (\bar{X} - \bar{X}^r))$$

$$\tilde{g}_j^r(\bar{X}) = g_j(\bar{X}^r) + (\nabla g_j(\bar{X}^r), (\bar{X} - \bar{X}^r)), j = 1, \dots, l$$

Вместо исходной задачи на каждой итерации решается вспомогательная задача линейного программирования:

$$\min_{\bar{X} \in D^r} \tilde{\Phi}^r(\bar{X}) = \tilde{\Phi}^r(\bar{X}^{r+1})$$

где

$$D^r = \{\bar{X} \mid \tilde{g}_j^r(\bar{X}) \geq 0, j \in [1, \dots, l]\}$$

В изложенном виде метод может привести к выходу точки X^{r+1} за пределы допустимой области.

Пример

Рассмотрим следующую двумерную задачу условной оптимизации с тремя ограничениями типа неравенств:

$$\Phi(\bar{X}) = x_1^2 + (x_2 - 6)^2 - 12$$

$$g_1(\bar{X}) = -x_1^2 - x_2^2 + 25 \geq 0$$

$$g_2(\bar{X}) = x_1 \geq 0$$

$$g_3(\bar{X}) = x_2 \geq 0$$

Пусть текущая точка есть $X^r = (2, 4)$.

Линеаризуем целевую функцию $\Phi(X)$ и ограничивающую функцию в окрестности точки X^r .

$$\tilde{\Phi}^r(\bar{X}) = \Phi(\bar{X}^r) + (\nabla \Phi(\bar{X}^r), (\bar{X} - \bar{X}^r)) = ?$$

$$\tilde{g}_i^r(\bar{X}) = g_i(\bar{X}^r) + (\nabla g_i(\bar{X}^r), (\bar{X} - \bar{X}^r)) = ?$$

$$\Phi(\bar{X}^r) = 2^2 + (4-6)^2 - 12 = -4$$

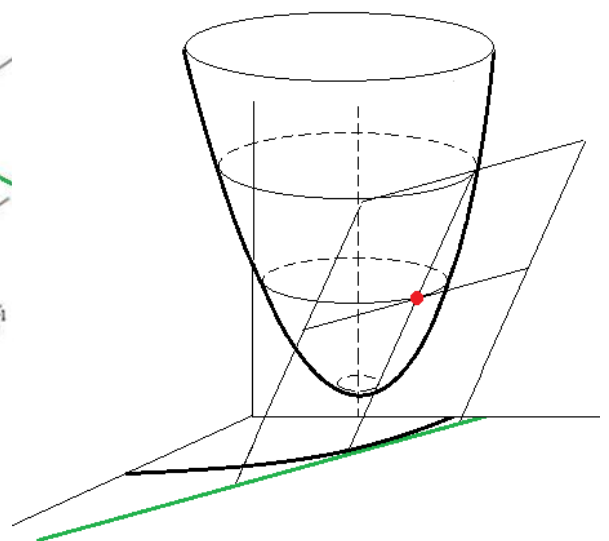
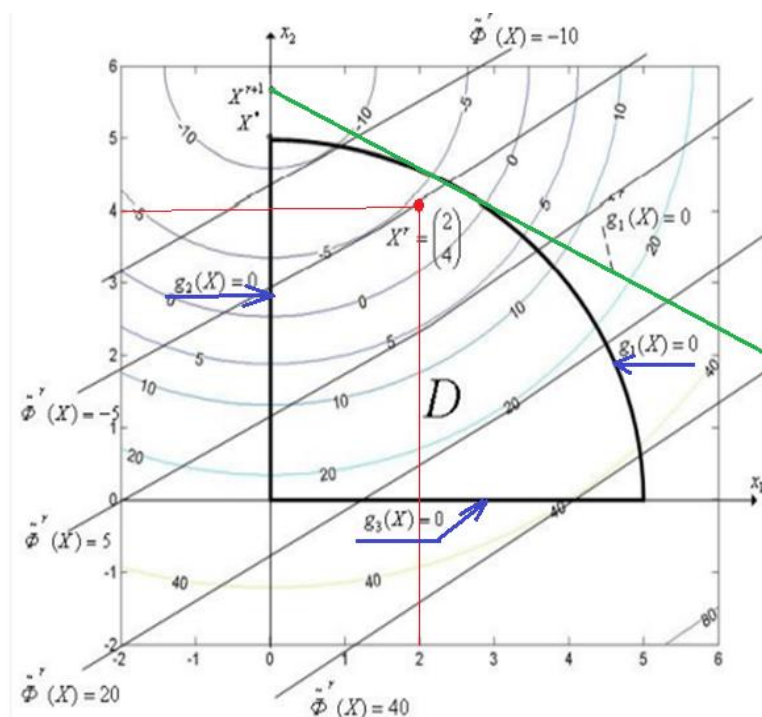
$$\nabla \Phi(\bar{X}) = \begin{pmatrix} 2x_1 \\ 2(x_2 - 6) \end{pmatrix}; \quad \nabla g_1(\bar{X}) = \begin{pmatrix} -2x_1 \\ -2x_2 \end{pmatrix}$$

$$\tilde{\Phi}^r(\bar{X}) = \Phi(\bar{X}^r) + (\nabla \Phi(\bar{X}^r), (\bar{X} - \bar{X}^r)) = -4 + \left(\begin{pmatrix} 4 \\ -4 \end{pmatrix}, \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} 2 \\ 4 \end{pmatrix} \right) = 4x_1 - 4x_2 + 4$$

$$\tilde{g}_1^r(\bar{X}) = g_1(\bar{X}^r) + (\nabla g_1(\bar{X}^r), (\bar{X} - \bar{X}^r)) = 5 + \left(\begin{pmatrix} -4 \\ -8 \end{pmatrix}, \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} 2 \\ 4 \end{pmatrix} \right) = -4x_1 - 8x_2 + 45$$

$$\tilde{g}_2^r(\bar{X}) = g_2(\bar{X}^r) = x_1$$

$$\tilde{g}_3^r(\bar{X}) = g_3(\bar{X}^r) = x_2$$



Прямая $\tilde{g}_1^r(\bar{X}) = 0$ представляет собой след от пересечения плоскости $0x_1x_2$ с плоскостью, которая является касательной к поверхности $z = \Phi(X)$ в точке X^r . Эта

прямая не обязательно является касательной к линии $g_1^r(\bar{X}) = 0$ – прямая $\tilde{g}_1^r(\bar{X}) = 0$ может пересекать кривую $g_1^r(\bar{X}) = 0$, быть касательной к ней или не иметь с ней общих точек.

Аналогично, линия уровня функции $\Phi^r(X)$ представляет собой след от пересечения плоскости, которая является касательной к поверхности $z = \Phi(X)$ в точке X^r , с плоскостью $z = c$.

Чтобы избежать выхода текущей точки за границы области допустимых значений, следующее приближение X^{r+1} к точке минимума X^* функции $\Phi(X)$ из множества D находится по формуле:

$$\bar{X}^{r+1} = \bar{X}^r + \lambda^r (\tilde{\bar{X}}^r - \bar{X}^r)$$

где $\tilde{\bar{X}}$ – решение вспомогательной задачи линейного программирования.

Величина шага $\lambda^r \in [0, 1]$ в формуле в разных вариантах метода линейной аппроксимации может определяться разными способами, например:

1. Величина λ^r находится как решение задачи одномерной оптимизации функции

$$\min_{\lambda} \Phi(\bar{X}^r + \lambda^r (\tilde{\bar{X}}^r - \bar{X}^r)) \text{ на отрезке } [0, 1].$$

2. Полагаем $\lambda^r = 1$ и находим вектор X_{r+1} . Вычисляем значение целевой функции в полученной точке $\Phi(\bar{X}^{r+1})$.

Если условие $\Phi(\bar{X}^{r+1}) < \Phi(\bar{X}^r)$ не выполнено, то уменьшаем величину шага λ^r (например, в два раза) и повторно проверяем выполнение условия, и т.д., пока не выполнится.

Схема метода линейной аппроксимации

Рассмотрим вариант метода, в котором используется 1-й способ выбора величины шага λ^r .

1. Задаем начальную точку $X_0 \in D$ и полагаем счетчик числа итераций $r = 0$.
2. Вычисляем градиенты функций $\Phi(X)$ и $g_j(X)$ в точке X^r .

3. Решаем задачу линейного программирования – находим точку $\tilde{\bar{X}}$.

4. Решаем одномерную задачу минимизации $\min_{\lambda} \Phi(\bar{X}^r + \lambda^r (\tilde{\bar{X}}^r - \bar{X}^r))$ – находим величину шага λ^r и вектор X_{r+1} .

5. Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то полагаем $X^*=X^{r+1}$, $\Phi^*=\Phi(X^{r+1})$ и завершаем итерации.

Иначе – полагаем $r=r+1$ и переходим к п.2.

В качестве критерия окончания поиска можно использовать стандартные условия окончания итераций:

$$\|\bar{X}^{r+1} - \bar{X}^r\| \leq \epsilon_{psx}$$

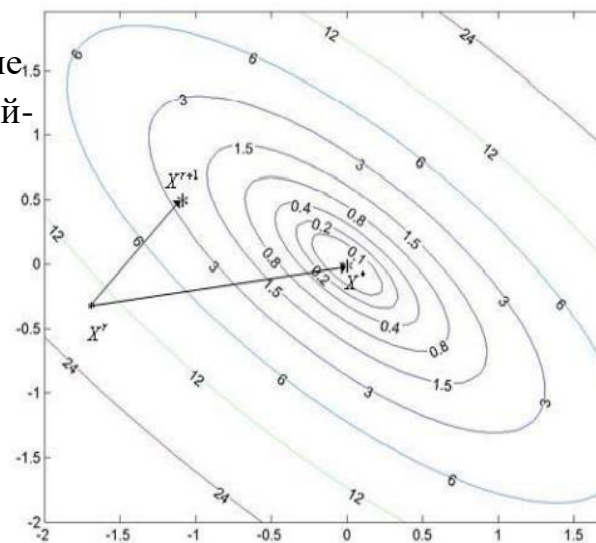
$$|\Phi(\bar{X}^{r+1}) - \Phi(\bar{X}^r)| \leq \epsilon_{ps\Phi}$$

$$\|\nabla \Phi(\bar{X}^{r+1})\| \leq \epsilon_{ps_grad}$$

Недостатки (трудности) метода линейной аппроксимации:

1. Метод требует, чтобы точка X_0 принадлежала множеству допустимых значений D . Если это требование не выполнено, то прежде придется использовать какой-либо метод поиска точки, принадлежащей множеству допустимых значений.

2. Если функция $\Phi(X)$ имеет высокую степень нелинейности, то на основе решения вспомогательной задачи минимизации направление поиска может быть выбрано слишком неточно, что приводит к медленной сходимости метода (см рис).



6 Метод проекции градиента

Рассмотрим многомерную задачу локальной условной оптимизации:

$$\min_{\bar{X} \in D \subset \mathbb{R}^n} \Phi(\bar{X}) = \Phi(\bar{X}^*) = \Phi^*$$

где множество допустимых значений D определяется только ограничениями типа неравенств

$$D = \{\bar{X} \mid g_j(\bar{X}) \geq 0, j \in [1, \dots, J]\},$$

целевая функция $\Phi(X)$ и ограничивающие функции g_j являются непрерывными и дифференцируемыми функциями, а ограничивающие функции выпуклы.

Проектирование точки на множество

Идея метода проекции градиента состоит в том, что если на некоторой итерации точка

$$\bar{X}^{r+1} = \bar{X}^r + \lambda^r (\bar{S}^r), \quad \bar{S}^r = -\frac{\nabla \Phi^r}{\|\nabla \Phi^r\|}$$

полученная с помощью градиентного *метода наискорейшего спуска*, оказывается ВНЕ множества D , то она возвращается на это множество.

Возврат производится с помощью процедуры "*проекция точки на множество*".

Проекцией точки $X \in \mathbb{R}^n$ на замкнутое множество $D \subset \mathbb{R}^n$ называется ближайшая к X точка \hat{X} множества D .

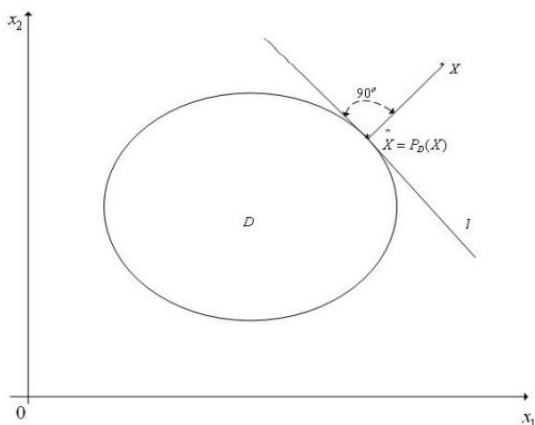
Т.е. точка $\hat{X} \in D$ называется проекцией точки $X \in \mathbb{R}^n$ на замкнутое множество $D \subset \mathbb{R}^n$, если

$$\rho(X, \hat{X}) = \min_{Y \in D} \rho(X, Y)$$

где $\rho(A, B)$ расстояние между точками $A \in \mathbb{R}^n$, $B \in \mathbb{R}^n$ в некоторой метрике.

Проекцию \hat{X} точки X на замкнутое множество D будем обозначать $P_D(X)$

Очевидно, что если $X \in D$, то $P_D(X) = X$.



Можно показать, что если D — замкнутое выпуклое множество пространства \mathbb{R}^n , то для любой точки $X \in \mathbb{R}^n$ существует единственная ее проекция на это множество.

Задача поиска проекции точки на множество также является многомерной задачей условной оптимизации и ее решение может вызвать в общем случае значительные затруднения.

Задача становится задачей квадратичного программирования, если множество D задается лишь линейными ограничениями типа неравенств и если функция ρ является квадратичной функцией Y .

Наибольший практический интерес представляет ситуация, когда множество D таково, что задача может быть решена в явном виде.

Схема комбинации метода проекции градиента с методом дробления шага

Метод проекции градиента может быть скомбинирован со многими градиентными методами. Рассмотрим комбинацию метода проекции градиента с градиентным методом дробления шага. Напомним, что в градиентном методе с дроблением шага величина шага λ^r находится из условия:

$$\Phi(\bar{X}^r) - \Phi(\bar{X}^{r+1}) < 0.5\lambda^r \|\nabla \Phi^r\|$$

Схема метода

1. Задаем начальную точку X^0 , начальную величину шага λ^0 и коэффициент дробления шага $\nu \in (0,1)$. Полагаем счетчик числа итераций $r=0$.
2. Вычисляем координаты точки
$$\bar{X}^{r+1} = \bar{X}^r + \lambda^r (\bar{S}^r)$$
и проекцию этой точки на множество D.
3. Вычисляем $\Phi(X^{r+1})$
4. Если условие дробления шага выполнено, то переходим к следующему пункту. Иначе – переходим к п.6.
5. Полагаем $\lambda^r = \nu \lambda^r$ и переходим к п.2.
6. Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то полагаем $X^* = X^{r+1}$, $\Phi^* = \Phi(X^{r+1})$ и завершаем итерации. Иначе – полагаем $r=r+1$ и переходим к п.2.

Критерии окончания поиска:

а) $\|\bar{X}^{r+1} - \bar{X}^r\| \leq \epsilon_{psx}$

б) $|\Phi(\bar{X}^{r+1}) - \Phi(\bar{X}^r)| \leq \epsilon_{ps\Phi}$

в) $\|\nabla \Phi^r\| \leq \epsilon_{ps_grad}$

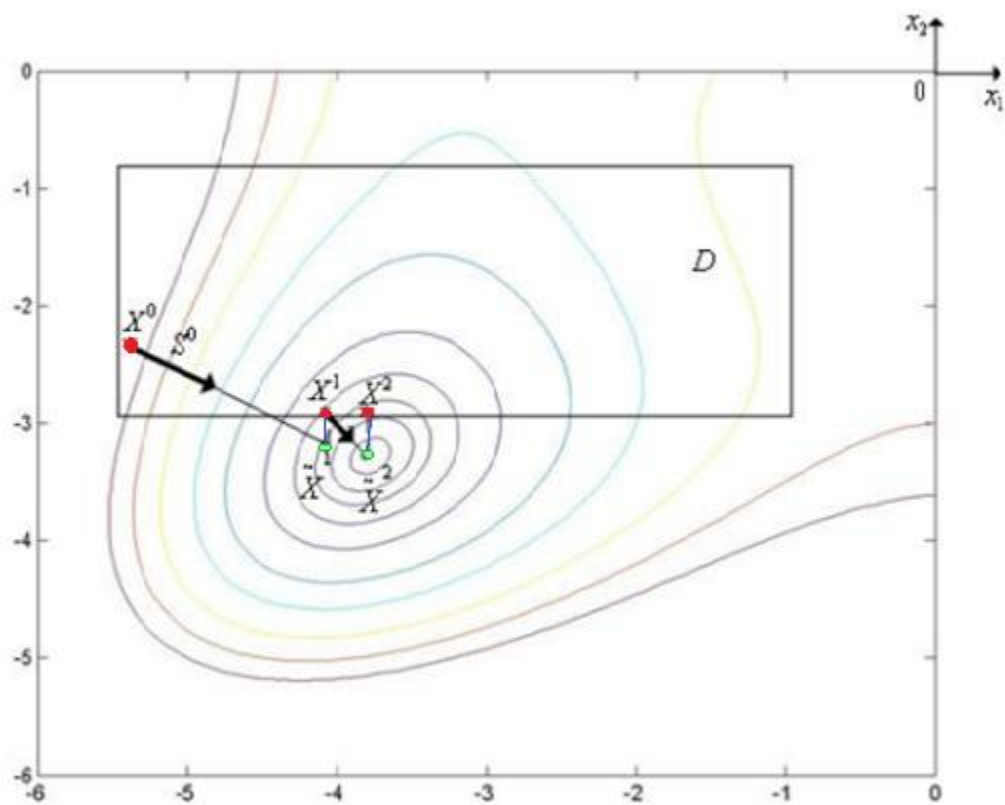


Рис. Траектория поиска минимума функции Химмельблау комбинацией метода проекции градиента и градиентного метода с дроблением шага

7 Метод сведения к совокупности вложенных задач глобальной одномерной минимизации

Рассмотрим многомерную задачу глобальной условной оптимизации: найти минимум критерия оптимальности $\Phi(X)$, определенного во множестве D пространства R^n

$$\min_{\bar{X} \in D \subset R^n} \Phi(\bar{X}) = \Phi(\bar{X}^*) = \Phi^*$$

где множество допустимых значений D задается только с помощью ограничений типа неравенств и представляет собой гиперпараллелепипед

$$D = \{\bar{X} \mid x_i^L \leq x_i \leq x_i^U, i \in [1, \dots, n]\}$$

Метод сведения к совокупности вложенных одномерных задач глобальной оптимизации состоит в решении вместо исходной задачи следующей совокупности вложенных одномерных задач условной оптимизации:

$$\min_{\bar{X} \in D} \Phi(\bar{X}) = \min_{\bar{X} \in [x_1^L; x_1^U]} \min_{\bar{X} \in D(x_1)} \dots \min_{\bar{X} \in D(x_1, \dots, x_{n-1})} \Phi(\bar{X})$$

где множества $D(x_1), D(x_1, x_2), \dots, D(x_1, x_2, \dots, x_{n-1})$ представляют собой соответствующие сечения множества D (см. пример ниже).

Пример

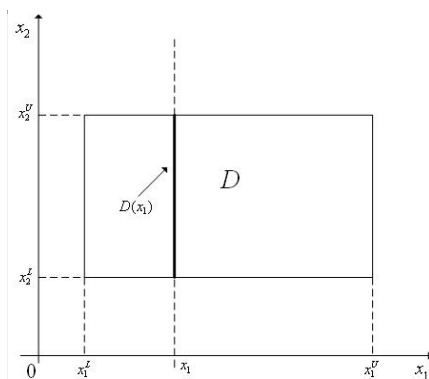
$$\Phi(\bar{X}) = \Phi(x_1, x_2) \text{ и } D = \{\bar{X} \mid x_1^L \leq x_1 \leq x_1^U, x_2^L \leq x_2 \leq x_2^U\}$$

Вложенные одномерные задачи глобальной оптимизации в этом случае можно представить в виде

$$\min_{\bar{X} \in D} \Phi(\bar{X}) = \min_{x_1 \in [x_1^L; x_1^U]} \Phi(x_1)$$

$$\Phi(x_1) = \min_{x_2 \in D(x_1)} \Phi(x_1, x_2)$$

где $D(x_1)$ — сечение области D прямой, параллельной оси Ox_2 .



Задача представляет собой одномерную задачу глобальной оптимизации критерия оптимальности $\Phi(x_1)$ по параметру $x_1 \in [x_1^L, x_1^U]$, для вычисления значения $\Phi(x_1)$ которого при заданном фиксированном x_1 необходимо решить одномерную задачу глобальной оптимизации критерия оптимальности $\Phi(x_1, x_2)$ по параметру $x_2 \in D(x_1)$.

Метод решения многомерной задачи глобальной условной оптимизации путем сведения к совокупности вложенных одномерных задач глобальной оптимизации может быть скомбинирован со всеми рассмотренными методами решения одномерных задач глобальной оптимизации. Рассмотрим вариант, когда для решения всех вложенных одномерных задач глобальной оптимизации используется метод случайного поиска.

Обозначим N_k число испытаний, необходимых для отыскания глобального минимума функции $\Phi(x_1, \dots, x_{k-1}, x_k, x_{k+1}, \dots, x_n)$ по параметру $x_k \in [x_k^L, x_k^U]$ **методом перебора** с заданной точностью (при этом параметры $x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n$ фиксированы).

Тогда общее количество испытаний для решения задачи равно $N = \prod_{k=1}^n N_k$.

Поэтому при $n > 3$ такой алгоритм становится неэффективным.

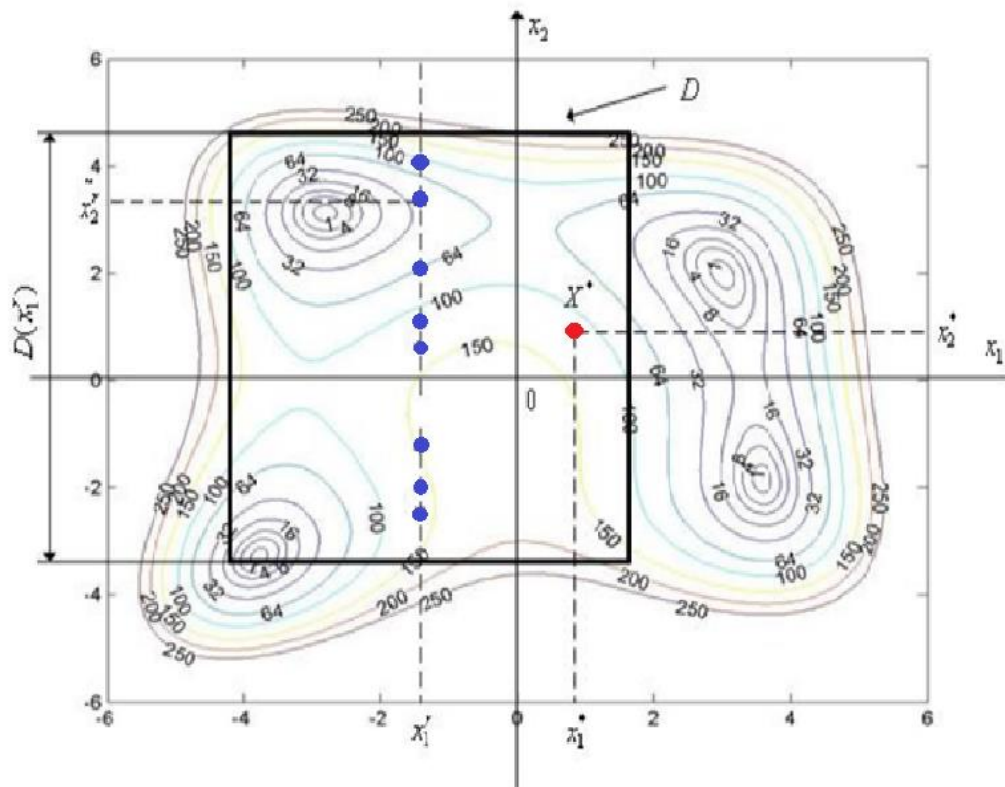
При $n \leq 3$ надежность алгоритма достаточно высока, а затраты на поиск значительно меньше затрат на полный перебор на той же сетке.

Схема комбинации метода с методом случайного поиска (n=2)

1. Задаем величины N_1, N_2 – количества испытаний при решении задач минимизации. Полагаем $r=1, k=1$.
2. С помощью какого-либо генератора случайных чисел, равномерно распределенных в интервале $[x_k^L, x_k^U]$, генерируем случайное число $x_1^0 = x_1^*$
3. Методом случайного поиска решаем задачу при $x_1 = x_1^*$ – находим точку $x_2^0 = x_2^*$ и вычисляем значение критерия оптимальности $\Phi^* = \Phi(x_1^*, x_2^*)$
4. Аналогично п.2 генерируем случайное число x_1^r
5. Методом случайного поиска решаем задачу при $x_1 = x_1^r$ – находим точку \tilde{x}_2^r и вычисляем значение критерия оптимальности $\Phi^r = \Phi(x_1^r, \tilde{x}_2^r)$
6. Если $\Phi^r < \Phi^*$, то выполняем присваивания $x_1^* = x_1^r, x_2^* = \tilde{x}_2^r, \Phi^* = \Phi^r$.
7. Если $r1 < N1$, то выполняем присваивание $r1=r1+1$ и переходим на п.4. Иначе принимаем точку (x_1^*, x_2^*) в качестве приближенного значения точки глобального минимума функции $\Phi(X)$ в области D или каким-либо из рассмотренных ранее методов организуем в окрестности указанной точки поиск локального минимума функции $\Phi(X)$ и заканчиваем вычисления.

Иллюстрация на примере функции Химмельблау (n=2) – см. рис.

Принято, что X^* – точка минимума функции $\Phi(X)$ в области D после $(r-1)$ -й итерации. Точки на прямой $x_1 = x_1^r$ случайным образом сгенерированы на r -й итерации.



Замечание. Рассмотренный метод, как и любой другой метод глобальной оптимизации, при отсутствии априорной информации о свойствах минимизируемой функции **не гарантирует нахождение глобального минимума.**

8 Редукция размерности с помощью развертки Пеано (сведение к задаче одномерной глобальной оптимизации)

Рассмотрим многомерную задачу глобальной условной оптимизации:

$$\min_{\bar{X} \in D} \Phi(\bar{X}) = \Phi(\bar{X}^*) = \Phi^*$$

где множество допустимых значений D задается только с помощью ограничений типа неравенств и представляет собой *гиперкуб с центром тяжести в начале координат и длиной ребра, равной 1*

$$D = \{\bar{X} \mid -0.5 \leq x_i \leq 0.5, \quad i = 1, \dots, n\}$$

Разбиение гиперкуба. Развертка Пеано

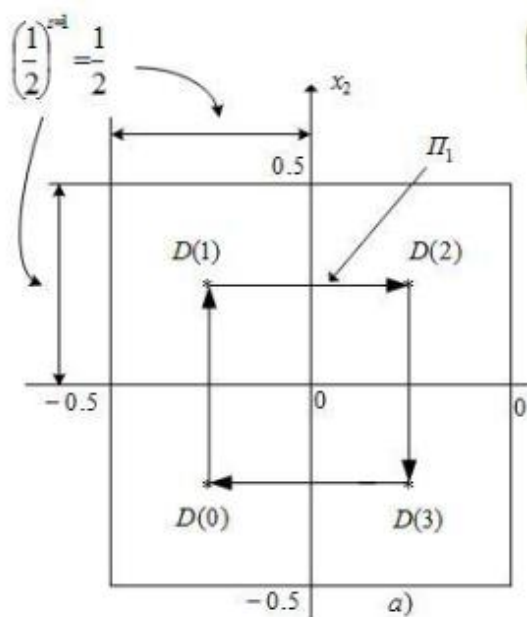
Шаг 1 ($s = 1$). Координатными плоскостями гиперкуб D разбивается на 2^n гиперкубов

первого разбиения с длиной ребра, равной $\frac{1}{2} = \left(\frac{1}{2}\right)^{s=1}$ (см. рис.а)

Пронумеруем их с помощью целочисленной переменной $z_1 \in [0, 2^n - 1]$ таким образом, чтобы гиперкубы с номерами, отличающимися на единицу, имели общую грань.

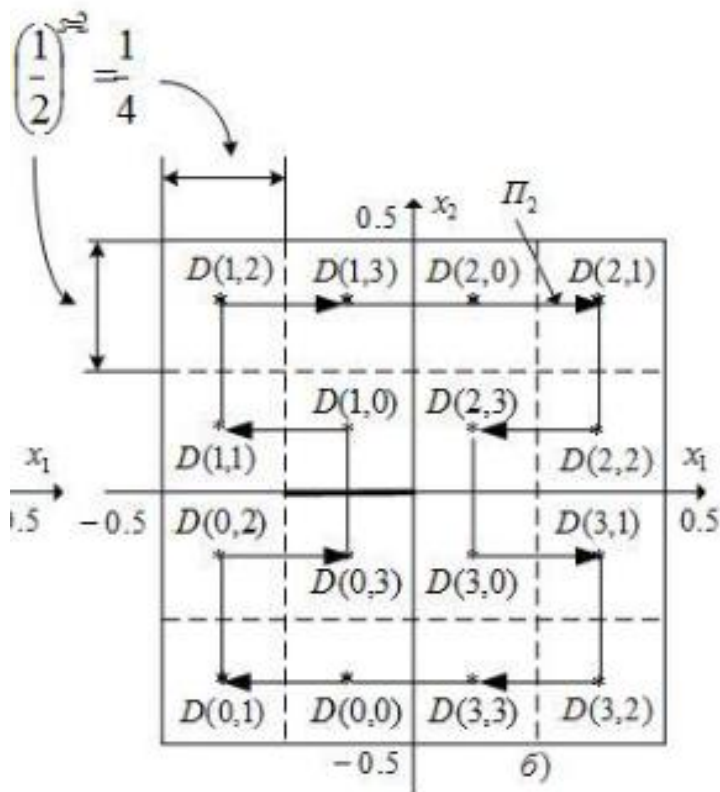
Соединим центры гиперкубов ломаной Π_1 в порядке введенной нумерации.

Гиперкуб первого разбиения с номером z_1 обозначим $D(z_1)$.



Шаг 2 ($s = 2$). По рассмотренной схеме каждый гиперкуб первого разбиения разобьем плоскостями, параллельными координатным плоскостям и проходящими через его

центр, на 2^n гиперкубов второго разбиения с длиной ребра, равной $\frac{1}{4} = \left(\frac{1}{2}\right)^{s=2}$ (рис.б)



Пронумеруем полученные гиперкубы с помощью переменной $z_2 \in [0, 2^n - 1]$ по тому же правилу, что и гиперкубы первого разбиения, с тем отличием, что нулевой гиперкуб второго разбиения, входящий в гиперкуб $D(z_1)$, должен иметь общую грань с $(2^n - 1)$ -ым гиперкубом второго разбиения, входящим в гиперкуб $D(z_1 - 1)$.

Соединим центры гиперкубов ломаной Π_2 в порядке введенной нумерации. Обозначим гиперкубы второго разбиения $D(z_1, z_2)$.

...

Шаг s. Аналогично шагу 2 разбиваем гиперкубы $(s-1)$ -го разбиения на гиперкубы s -го

разбиения с длиной ребра, равной $\left(\frac{1}{2}\right)^s$, нумеруем их с помощью переменной

$z_s \in [0, 2^n - 1]$, соединяем центры гиперкубов ломаной Π_s в порядке введенной нумерации и обозначаем $D(z_1, z_2, \dots, z_s)$.

Ломаная Π_s называется **разверткой Пеано**.

В пределе при $s \rightarrow \infty$ ломаная Π_s называется **кривой Пеано**. Кривая Пеано обладает тем свойством, что проходит через все точки гиперкуба и имеет в каждой точке излом.

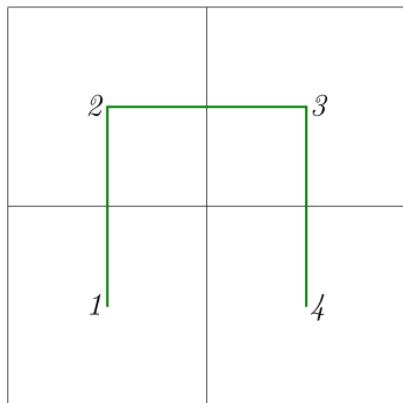


Fig. 1.

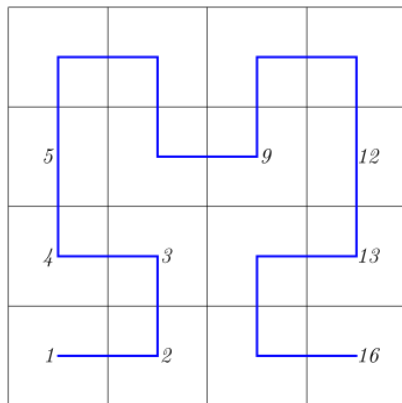


Fig. 2.

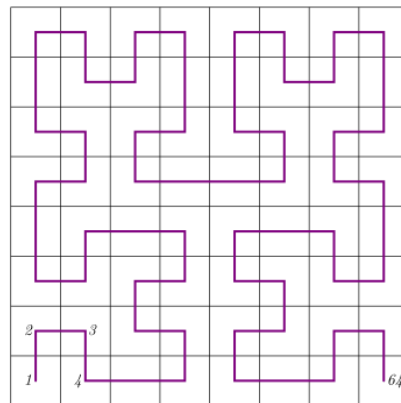
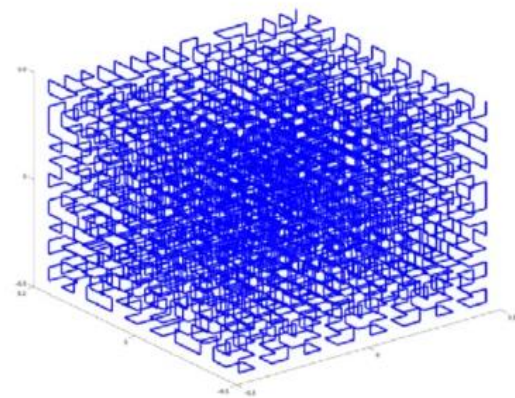
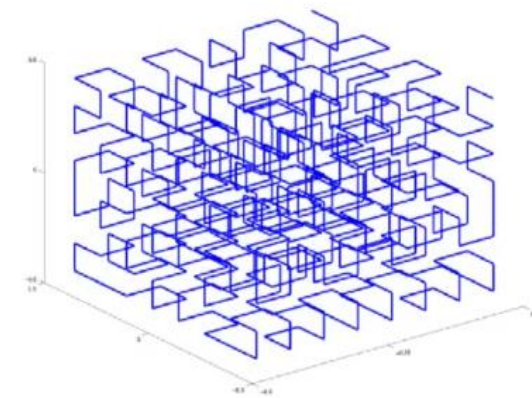
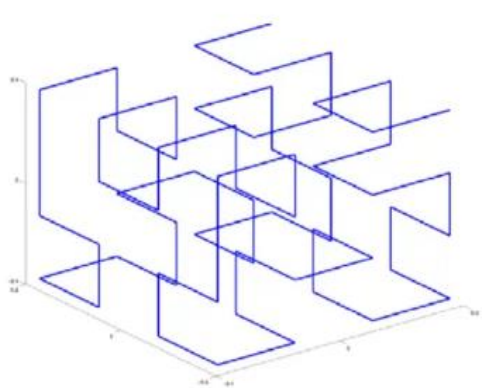
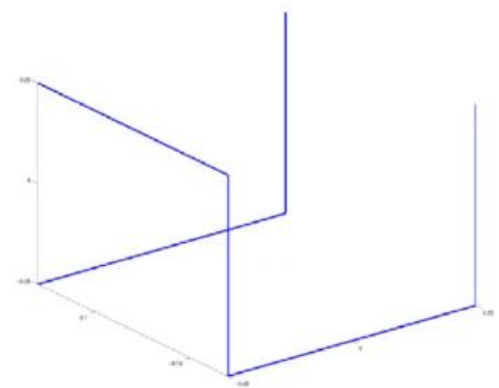
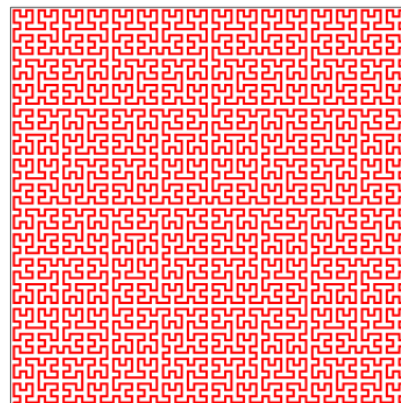
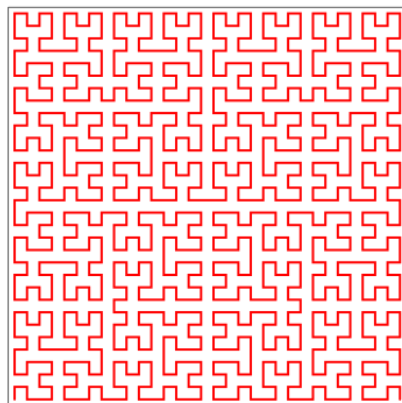
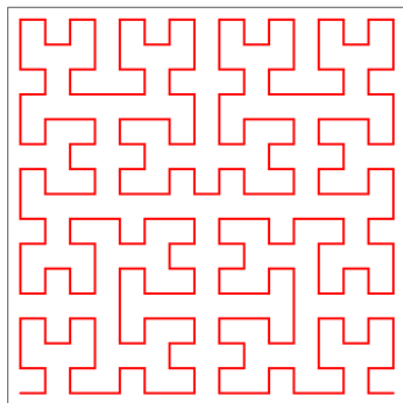


Fig. 3.



Разбиение отрезка [0, 1]

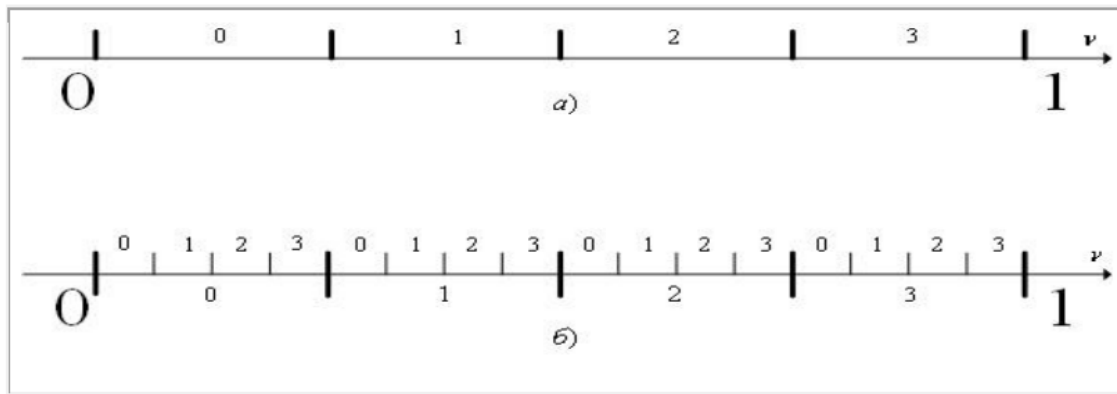
Шаг 1 ($s=1$). Разобьем отрезок $[0,1]$ на 2^n равных частей длиной $\left(\frac{1}{2^n}\right)^{s=1}$ (см. рис. а), пронумеруем их слева направо с помощью переменной $z_1 \in [0, 2^n - 1]$ и обозначим Δz_1 .

Шаг 2 ($s=2$).

Каждый из отрезков Δz_1 , $z_1 \in [0, 2^n - 1]$ разобьем на 2^n равных частей длиной $\left(\frac{1}{2^n}\right)^{s=2}$ (см. рис.б), пронумеруем их слева направо с помощью переменной $z_2 \in [0, 2^n - 1]$ и обозначим $\Delta(z_1, z_2)$.

...

Шаг n . Аналогично шагу 2 каждый из отрезков $(s-1)$ -го разбиения разобьем на 2^n равных частей длиной $\left(\frac{1}{2^n}\right)^{s=n}$, пронумеруем их слева направо с помощью $z_n \in [0, 2^n - 1]$ и обозначим $\Delta(z_1, z_2, \dots, z_n)$.



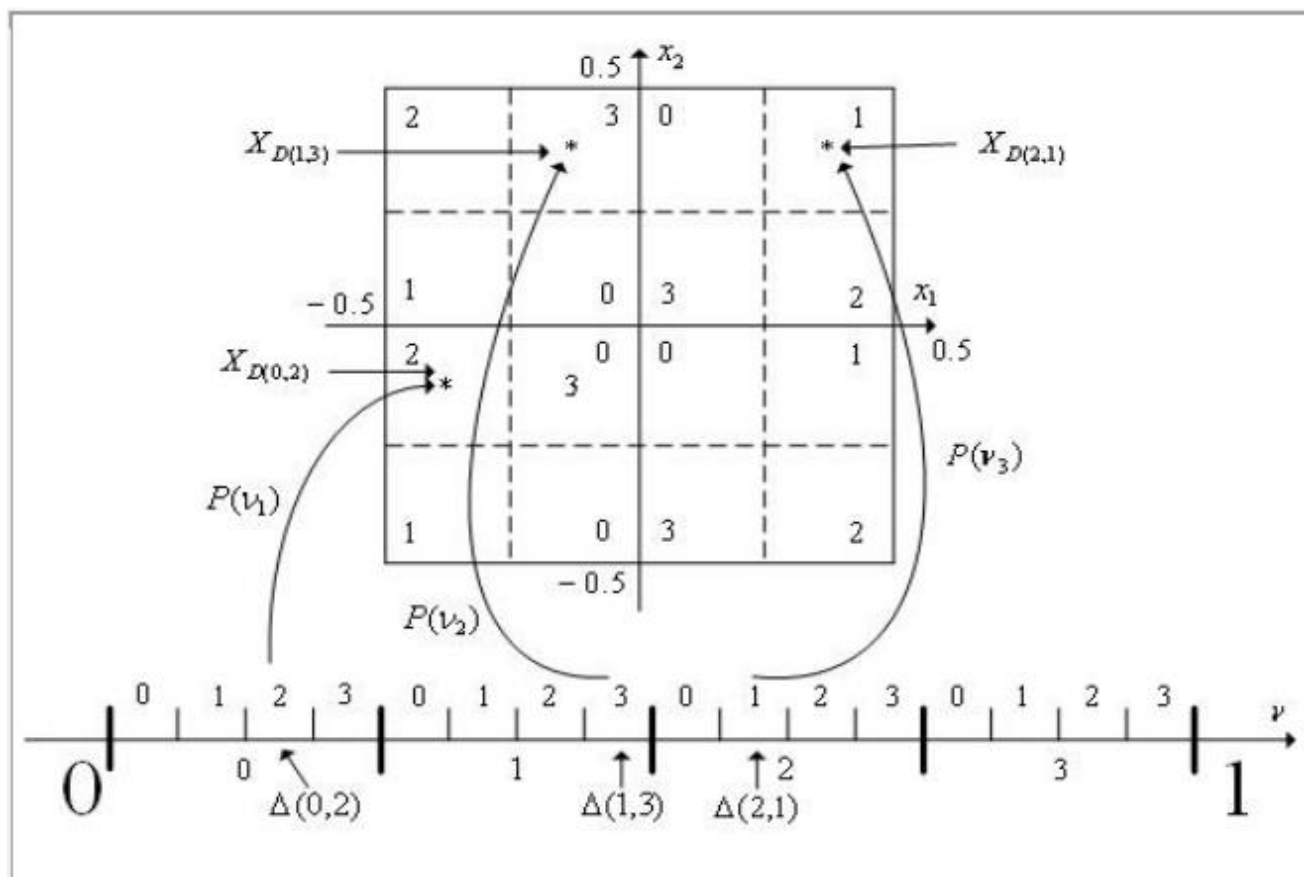
Отображение отрезка [0,1] на гиперкуб

Определим отображение точки v отрезка $[0,1]$ на гиперкуб D следующим образом: если точка $v \in \Delta(z_1, z_2, \dots, z_s)$, то соответствующая точка $X_{D(z_1, z_2, \dots, z_s)}$ является центром гиперкуба $D(z_1, z_2, \dots, z_s)$

Обозначим введенное отображение $P(v)$. Таким образом, если $v \in \Delta(z_1, z_2, \dots, z_s)$, то $P(v) = X_{D(z_1, z_2, \dots, z_s)} \in D(z_1, z_2, \dots, z_s)$ (см. рис).

На рис. любая точка $v_1 \in \Delta(0,2)$ отображается в центр гиперкуба $D(0,2)$ – точку $X_{D(0,2)}$

Аналогично, любая точка $v_2 \in \Delta(1,3)$ отображается в точку $X_{D(1,3)}$ и любая точка $v_3 \in \Delta(2,1)$ отображается в точку $X_{D(2,1)}$



В пределе при $s \rightarrow \infty$ введенное отображение отображает отрезок $[0,1]$ на кривую Пеано. Можно показать, что в пределе при $s \rightarrow \infty$ построенное **отображение является непрерывным и взаимнооднозначным.**

Утверждение

Пусть число $v \in [0,1]$ в двоичном представлении имеет вид

$$v_b = 0, a_1 a_2 \dots a_k \dots$$

Тогда первые S^n двоичных цифр числа $v_b = 0, a_1 a_2 \dots a_k \dots$ определяют разбиение $\Delta(z_1, z_2, \dots, z_s)$ отрезка $[0,1]$:

$$z_1 = \langle a_1, a_2, \dots, a_n \rangle$$

$$z_2 = \langle a_{n+1}, a_{n+2}, \dots, a_{2n} \rangle$$

...

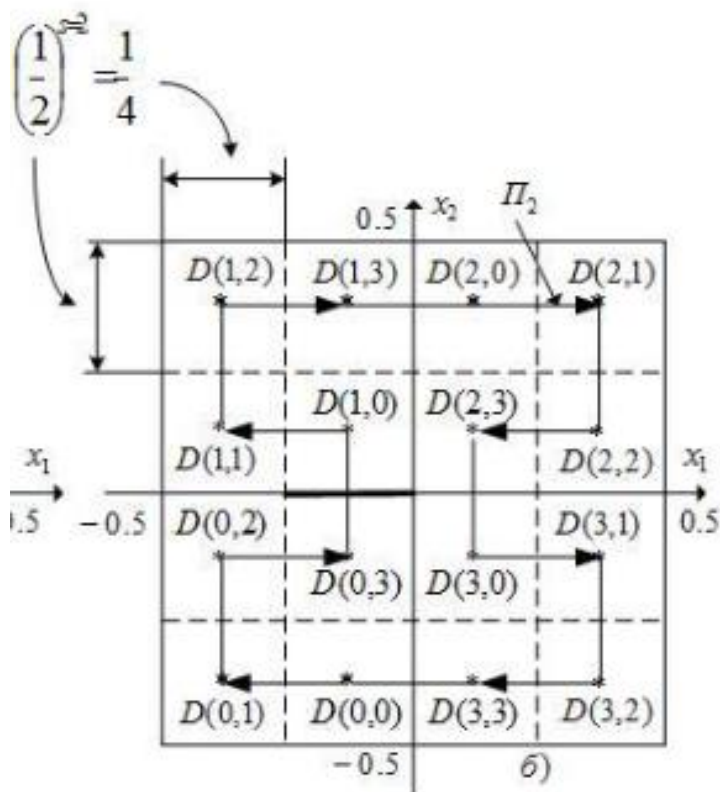
$$z_s = \langle a_{(s-1)n+1}, a_{(s-1)n+2}, \dots, a_{sn} \rangle$$

где $\langle * \rangle$ – операция преобразования двоичного числа в десятичное.

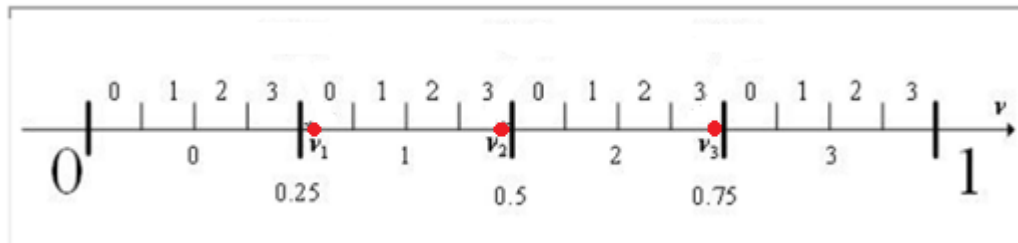
Пример

Пусть область D представляет собой квадрат ($n=2$).

Рассмотрим случай $s=2$.



На отрезке $[0,1]$ рассмотрим точки $v_1=0.26$, $v_2=0.49$, $v_3=0.74$ – см. рис.



Преобразуем $v_1=0.26$ в двоичную систему счисления (требуется найти $sn=2 \times 2=4$ знака):

$0.26 \times 2 = 0.52$ – запоминаем целую часть (0); $v_1=0.26=0.0\dots_b$

$0.52 \times 2 = 1.04$ – запоминаем целую часть (1); $v_1=0.26=0.01\dots_b$

$0.04 \times 2 = 0.08$ – запоминаем целую часть (0); $v_1=0.26=0.010\dots_b$

$0.08 \times 2 = 0.16$ – запоминаем целую часть (0); $v_1=0.26=0.0100\dots_b$

Итого, $v_1=0.26=0.0100\dots_b$

$$z_1 = \langle a_1, a_2 \rangle = \langle 01 \rangle = 1_d$$

$$z_2 = \langle a_3, a_4 \rangle = \langle 00 \rangle = 0_d$$

Таким образом, $v_1 \in \Delta(z_1, z_2) = \Delta(1, 0)$

Аналогично $v_2=0.49=0.0111\dots_b$

$$z_1 = \langle a_1, a_2 \rangle = \langle 01 \rangle = 1_d$$

$$z_2 = \langle a_3, a_4 \rangle = \langle 11 \rangle = 3_d$$

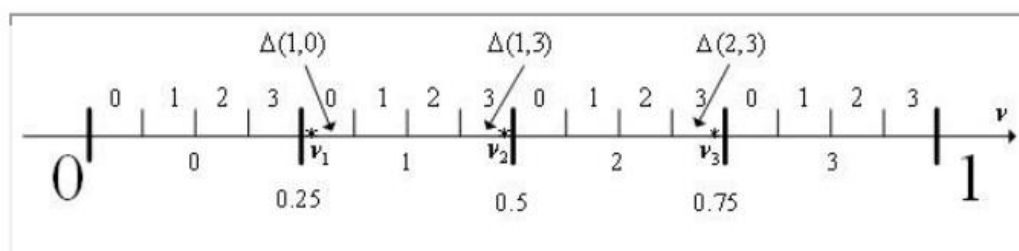
Таким образом, $v_2 \in \Delta(z_1, z_2) = \Delta(1,3)$

Аналогично $v_2 = 0.74 = 0.1011..._b$

$$z_1 = \langle a_1, a_2 \rangle = \langle 10 \rangle = 2_d$$

$$z_2 = \langle a_3, a_4 \rangle = \langle 11 \rangle = 3_d$$

Таким образом, $v_3 \in \Delta(z_1, z_2) = \Delta(2,3)$



Определим на отрезке $[0,1]$ функцию $\Phi(v) = \Phi(X) = P(v)$.

Отметим, что если функция $\Phi(X)$ является непрерывной функцией, то функция $\Phi(v)$ также непрерывна. Однако эта функция является НЕгладкой и МНОГОэкстремальной, даже если исходная функция $\Phi(X)$ гладкая и унимодальная.

Таким образом, с помощью развертки Пеано многомерная задача глобальной условной оптимизации сводится к **одномерной задаче** условной глобальной оптимизации

$$\min_{v \in [0,1]} \Phi(v) = \Phi(v^*) = \Phi^*$$

Решение задачи многомерной глобальной условной оптимизации с помощью развертки Пеано

Метод решения многомерной задачи глобальной условной оптимизации с использованием развертки Пеано называется **методом развертки Пеано** и может быть скомбинирован со всеми рассмотренными выше методами решения одномерных задач глобальной оптимизации.

Определение необходимого количества разбиений s

При заданной точности ϵ_{rsx} решения задачи по X необходимое количество s разбиений гиперкуба D может быть найдено из следующих соображений.

Гиперкуб s -го разбиения $D(z_1, z_2, \dots, z_s)$ имеет длину ребра, равную $\left(\frac{1}{2}\right)^s$

Максимальное расстояние точек этого гиперкуба до его центра равно половине диагонали гиперкуба, которая, очевидно, равна корню квадратному из суммы квадратов

длин n ребер гиперкуба, т.е. макс.расстояние до центра равно $\frac{1}{2} \sqrt{n \cdot \left(\frac{1}{2}\right)^{2s}}$

Таким образом, s может быть найдено из условия $\left(\frac{1}{2}\right)^{s+1} \cdot \sqrt{n} \leq \epsilon p s x$

Схема решения задачи многомерной оптимизации с помощью развертки Пеано

1. Для заданной точности $\epsilon p s x$ решения исходной задачи по X найти s – количество разбиений области D из условия

$$\left(\frac{1}{2}\right)^{s+1} \cdot \sqrt{n} \leq \epsilon p s x$$

2. Решить одномерную задачу оптимизации $\Phi(v)$, где $v \in [0,1]$.

При этом вычисления значений $\Phi(v)$ должны производиться по следующей схеме:

– для заданного $v \in [0,1]$ находим $s \times n$ цифр его двоичного представления

$$v_b = 0, a_1, a_2, \dots, a_{sn}, \dots$$

– определяем числа

$$z_1 = \langle a_1, a_2, \dots, a_n \rangle, z_2 = \langle a_{n+1}, a_{n+2}, \dots, a_{2n} \rangle, \dots, z_s = \langle a_{(s-1)n+1}, a_{(s-1)n+2}, \dots, a_{sn} \rangle$$

– в гиперкубе $D(z_1, z_2, \dots, z_s)$ определяем координаты его центра $X_{D(z_1, z_2, \dots, z_s)}$

– вычисляем значение критерия оптимальности $\Phi(X)$ в точке $X_{D(z_1, z_2, \dots, z_s)}$, которое и принимаем за значение $\Phi(v)$.

Замечание: Рассмотренный метод, как и любой другой метод глобальной оптимизации, при отсутствии априорной информации о свойствах минимизируемой функции не гарантирует нахождение глобального минимума.

9 Метод Монте-Карло (метод мультистарта)

Рассмотрим многомерную задачу глобальной условной оптимизации:

$$\min_{\bar{X} \in D \subset R^n} \Phi(\bar{X}) = \Phi(\bar{X}^*) = \Phi^*$$

где множество допустимых значений определяется как ограничениями типа неравенств, так и ограничениями типа равенств

$$D = \{\bar{X} \mid h_i(\bar{X}) = 0, g_j(\bar{X}) \geq 0, i \in [1, \dots, m], j \in [1, \dots, l]\}$$

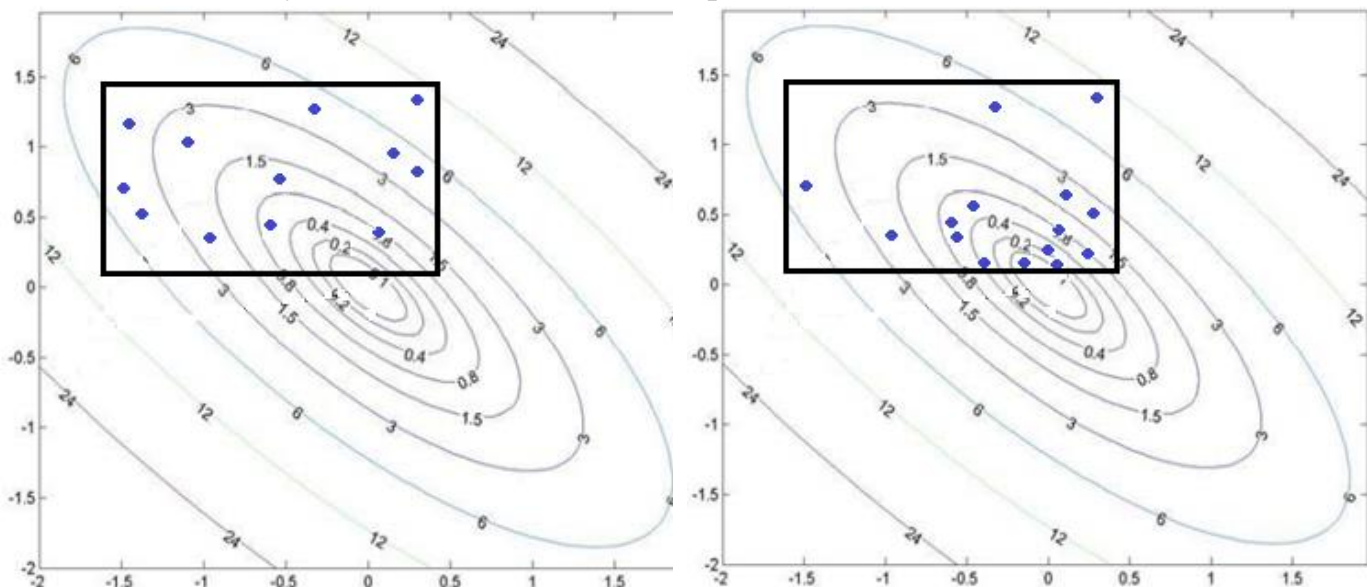
Метод Монте-Карло относится к классу прямых методов случайного поиска.

Схема метода Монте-Карло

1. Задаем общее количество испытаний N и полагаем счетчик числа итераций $r = 1$.
2. С помощью какого-либо программного генератора случайных чисел генерируем n компонент вектора $X^1 \in D$.
3. Вычисляем $\Phi(X^1)$ и полагаем $X^* = X^1$, $\Phi^* = \Phi(X^1)$, $r = r + 1$.
4. Аналогично п.2 генерируем случайную точку $X^r \in D$.
Вычисляем соответствующее значение критерия оптимальности $\Phi(X^r)$.
5. Выполняем следующие присваивания:
если $\Phi(X^r) < \Phi^*$, то $X^* = X^r$.
иначе $X^* = X^*$
6. Если $r < N$ полагаем $r = r + 1$ и переходим на п.4, иначе принимаем X^*, Φ^* в качестве приближенного решения задачи и заканчиваем вычисления.

Отметим, что в простейшем случае точки $X^r \in D$ генерируются *равномерно распределенными в области D* .

С целью сокращения вычислительных затрат и при наличии априорной информации о положении точки глобального минимума, целесообразно использовать законы распределения, в которых вероятность генерации точки в окрестности предполагаемого глобального минимума выше, чем вне этой окрестности.



Для локализации с помощью метода Монте-Карло глобального минимума с высокой вероятностью и точностью, требуется очень большое количество испытаний, поэтому метод Монте-Карло обычно комбинируют с каким-либо детерминированным методом локальной оптимизации.

Комбинация метода Монте-Карло с детерминированным методом локальной оптимизации

1. Задаем общее количество исходных случайных точек N .
2. С помощью какого-либо программного генератора случайных чисел генерируем N случайных точек X^1, X^2, \dots, X^N , принадлежащих множеству D .
3. Полагаем $r=1$.
4. Исходя из точки X^r , каким-либо многомерным методом условной оптимизации находим локальный минимум $(X^*)^r$ функции $\Phi(X)$ в окрестности этой точки и вычисляем $\Phi((X^*)^r) = (\Phi^*)^r$.
5. Если $r < N$ полагаем $r=r+1$ и переходим на п.4, иначе – переходим к следующему пункту.
6. Находим минимальное из чисел $(\Phi^*)^r, r=1,2,\dots,N$ – пусть $\min(\Phi^*)^r = (\Phi^*)^k$. Принимаем $(X^*)^k$ и соотв. значение $(\Phi^*)^r$ в качестве приближенного решения задачи и заканчиваем вычисления.