

Курс «Вычислительные алгоритмы теории автоматического управления».

Лекция 7. Методы условной оптимизации. Задачи целочисленного программирования.

Многокритериальная и глобальная оптимизация.

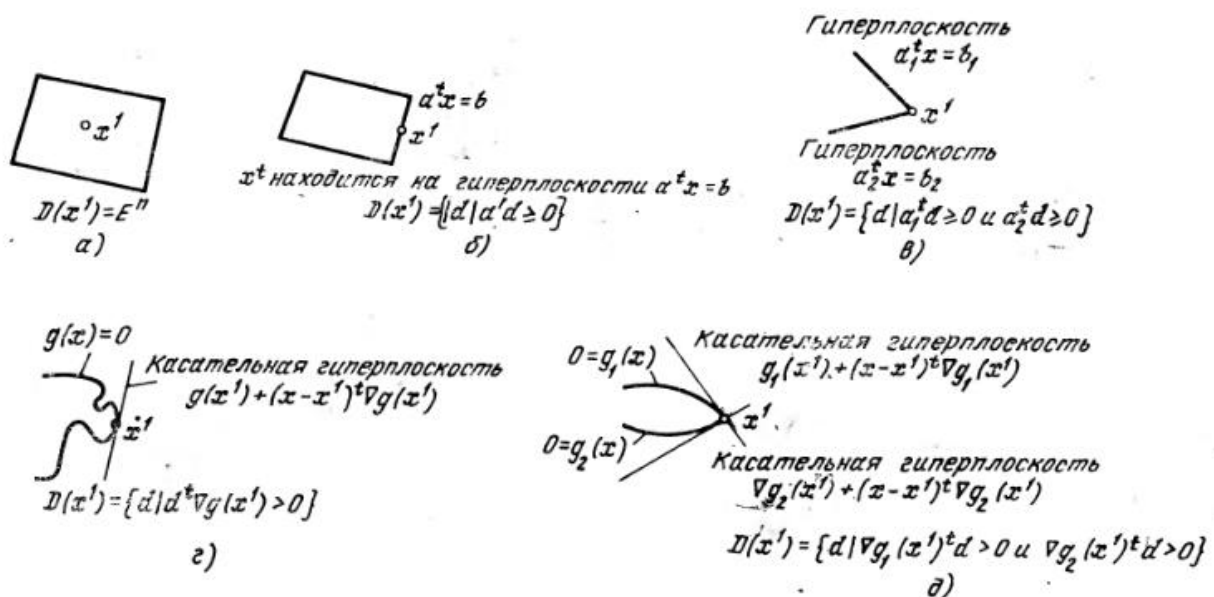
Необходимые и достаточные условия задач условной оптимизации. Седловые точки и функция Лагранжа. Численные методы и алгоритмы условной оптимизации. Методы штрафных функций. Многокритериальная оптимизация. Задачи целочисленного программирования. Задачи целочисленного программирования.. Проблема поиска глобального экстремума. Эволюционные алгоритмы. Программные пакеты системы Matlab для решения оптимизационных задач.

Необходимые и достаточные условия задач условной оптимизации.

Теорема, о необходимых условиях решения задачи безусловной оптимизации

$f(x) \rightarrow \min, x \in \mathbb{R}^n$, гласит, что если точка x^* - локальное решение, то выполняется равенство $\nabla f(x^*) = 0$. Однако, если имеются ограничения вида $x \in D \subset \mathbb{R}^n$, то движение в процессе процесса поиска, по направлению градиента $\nabla f(x^*)$, может достигнуть границы области D и, тогда доказательство теоремы станет неверным.

Пусть точка x является допустимой точкой. Определим возможное направление в точке x , как любое направление h , обладающее свойством, что $x + \alpha \cdot h$ находится в допустимом множестве $D \subset \mathbb{R}^n$ для всех достаточно малых α . То есть направление h является возможным в точке x , если существует такое значение $\sigma > 0$, что для $\forall \alpha, 0 \leq \alpha \leq \sigma$, точка: $x + \alpha \cdot h \in D$. Ниже на рисунке приведены примеры множеств возможных направлений.



Лемма. /Зангвилл с42/. Пусть точка x^* является оптимальной точкой решения задачи

$f(x) \rightarrow \min, x \in D(x) \subset \mathbb{R}^n$, где $D(x^*) = \{x \in \mathbb{R}^n : g_i(x) \leq 0, \|x - x^*\| \leq \varepsilon, i = 1, 2, \dots, m\}$, где $\varepsilon > 0$. Тогда $(\nabla f(x^*)) \cdot h \geq 0, \forall h \in D(x^*)$.

Доказательство. Допустим противоположное, то есть $\exists h \in D(x^*)$ такое, что $(\nabla f(x^*)) \cdot h < 0$. Тогда, по малое перемещение от точки x^* в направлении h уменьшало бы и значение целевой функции $f(x)$, что противоречит оптимальности точки x^* . Грубо говоря, в оптимальной точке производная целевой функции по направлению указывает на ее увеличение в любом возможном направлении.

Будем называть множество \bar{D} замыканием множества D , если любая точка \bar{D} является предельной для точек D . Очевидно, если D - замкнутое множество, то $D = \bar{D}$. К сожалению, множество $D(x^*)$ не обязательно замкнуто, так как некоторые касательные направления могут не входить в него (см. рисунок выше).

Следствие. Пусть точка x^* является оптимальной точкой для задачи $f(x) \rightarrow \min, x \in D(x^*) \subset \mathbb{R}^n$. Тогда $(\nabla f(x^*)) \cdot h \geq 0, \forall h \in \bar{D}(x^*)$.

Доказательство. Направление $h_\infty \in \bar{D}(x^*)$ может быть представлено как предельное для направлений $h^{(k)}$ из множества $h \in D(x^*)$. Так как по приведенной выше лемме

$$(\nabla f(x^*)) \cdot h^{(k)} \geq 0, \forall h^{(k)} \in D(x^*) \text{ для всех значений } k, \text{ то переход к пределу дает, что } (\nabla f(x^*)) \cdot h^\infty \geq 0.$$

Рассмотрим более детально множество $\bar{D}(x^*)$. В допустимой точке x ограничения могут быть разбиты на два множества: на те, которые активны в этой точке $\{g_i(x) = 0\}$, и на те, которые неактивны $\{g_i(x) < 0\}$. Обозначим через $I(x)$ множество индексов ограничений, активных в точке $x: (g_i(x) = 0), i \in I(x)$. Чтобы выразить $\bar{D}(x^*)$ через ограничения, необходимо рассмотреть лишь активные ограничения. Определим множество:

$$\tilde{D}(x^*) = \{h: (\nabla g_i(x))^T \cdot h \leq 0, \forall i \in I(x)\}.$$

Однако, существуют случаи, когда имеются направления в множестве $\tilde{D}(x^*)$, которые не принадлежат $\bar{D}(x^*)$. К счастью они редко встречаются на практике. Поэтому предположение, что $\tilde{D}(x^*) = \bar{D}(x^*)$ обычно вызывает небольшую потерю общности. Это предположение называется условием регулярности.

Отсюда, если точка x^* - оптимальная точка, то условие регулярности предполагает, что $\tilde{D}(x^*) = \bar{D}(x^*)$. В явной форме это условие можно записать в следующем виде:

$$\tilde{D}(x^*) = \bar{D}(x^*) = \{h: (\nabla g_i(x))^T \cdot h \leq 0, \forall i \in I(x^*)\}.$$

Теорема. (Условия Куна-Таккера о необходимых условия минимума в задаче условной оптимизации). /Зангвилл с44/. Рассмотрим задачу нелинейного программирования:

$$f(x) \rightarrow \min, x \in D(x^*) \subset \mathbb{R}^n,$$

где $D(x^*) = \{x \in \mathbb{R}^n: g_i(x) \leq 0, \|x - x^*\| \leq \varepsilon, i = 1, 2, \dots, m\}$ и все функции $f(x), g_i(x)$

дифференцируемые. Пусть точка x^* - оптимальная точка (решение) и предположим, что выполняются условие регулярности. Тогда справедливы следующие соотношения:

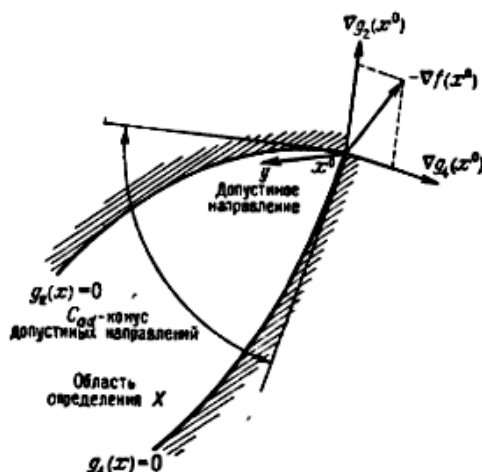
существуют множители $\lambda_i \geq 0, i = 1, 2, \dots, m$ такие, что $\lambda_i \cdot g_i(x^*) = 0, i = 1, 2, \dots, m$ и:

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i \cdot \nabla g_i(x^*) = 0.$$

Соотношения $\lambda_i \cdot \nabla g_i(x^*) = 0$, $i = 1, 2, \dots, m$ и $\nabla f(x^*) + \sum_{i=1}^m \lambda_i \cdot \nabla g_i(x^*) = 0$ называются

условиями Куна-Таккера. По существу, эти условия дают формальную формулировку интуитивной концепции: в оптимальной точке производная целевой функции по любому направлению должна указывать на ее рост. Геометрическая интерпретация условий Куна-Таккера заключается в том, что в оптимальной точке x^* градиент $\nabla f(x^*)$ лежит в геометрическом конусе градиентов ограничений, активных в точке x^* .

Иллюстрация условий Куна — Таккера на двумерном примере. В точке x^0 насыщенными ограничениями являются ограничения g_2 и g_4 и $I^0 = \{2, 4\}$.



Условия Куна-Таккера являются необходимыми, но недостаточными. Однако, если целевая функция $f(x)$ и ограничения $g_i(x)$ являются выпуклыми, то условия Куна-Таккера являются достаточными.

Седловые точки и функция Лагранжа.

Рассмотрим другой подход к формулировке необходимых и достаточных условий задачи нелинейного программирования. Пусть имеется задача следующего типа:

$$f(x) \rightarrow \min, \quad x \in D(x^*) \subset \mathbb{R}^n,$$

где $D(x^*) = \{x \in \mathbb{R}^n : g_i(x) \leq 0, \quad \|x - x^*\| \leq \varepsilon, \quad i = 1, 2, \dots, m\}$.

Поставим в соответствии каждому i -му ограничению действительное число $\lambda_i \geq 0$, называемое множителем Лагранжа. Функция Лагранжа, отвечающая задаче минимизации, есть по определению функция:

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i \cdot g_i(x).$$

Говорят, что пара (x^*, λ^*) есть седловая точка функции $L(x, \lambda)$, если выполняются следующие соотношения:

$$L(x^*, \lambda^*) \leq L(x, \lambda^*), \quad L(x^*, \lambda) \leq L(x^*, \lambda^*),$$

для $\forall x \in D(x^*), \quad \forall \lambda \geq 0$.

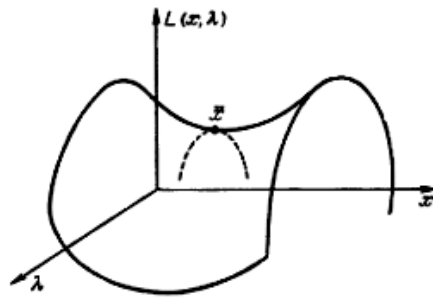


Иллюстрация понятия седловой точки: \bar{x} минимизирует $L(x, \bar{\lambda})$ на S ; $\bar{\lambda}$ максимизирует $L(\bar{x}, \lambda)$ на \mathbb{R}^+

Теорема (О свойствах седловых точек функции Лагранжа). /Мину с158/

Точка (x^*, λ^*) является седловой для функции $L(x, \lambda)$ в том и только в том случае если выполняются следующие соотношения:

$$L(x^*, \lambda^*) = \min_{x \in D(x^*)} (L(x, \lambda)), \quad g_i(x^*) \leq 0, \quad \lambda_i^* \cdot g_i(x^*) = 0, \quad i = 1, 2, \dots, m.$$

Теорема (о достаточности условия седловой точки функции Лагранжа).

Если точка (x^*, λ^*) есть седловая точка функции $L(x, \lambda)$, то точка x^* есть глобальный оптимум задачи:

$$f(x) \rightarrow \min, \quad x \in D(x^*) \subset \mathbb{R}^n,$$

где $D(x^*) = \{x \in \mathbb{R}^n : g_i(x) \leq 0, \quad \|x - x^*\| \leq \varepsilon, \quad i = 1, 2, \dots, m\}$.

Теорема (О существовании седловой функции в выпуклом случае).

Пусть функции $f(x), g_i(x)$ выпуклые, множество

$D(x^*) = \{x \in \mathbb{R}^n : g_i(x) \leq 0, \quad \|x - x^*\| \leq \varepsilon, \quad i = 1, 2, \dots, m\}$ выпуклое и существует такое $x \in D(x^*)$, что $g_i(x) < 0$. Тогда, если задача $f(x) \rightarrow \min, \quad x \in D(x^*) \subset \mathbb{R}^n$, где

$D(x^*) = \{x \in \mathbb{R}^n : g_i(x) \leq 0, \quad \|x - x^*\| \leq \varepsilon, \quad i = 1, 2, \dots, m\}$ имеет оптимальное решение x^* , то существует такой вектор множителей $\lambda^* \geq 0$, что (x^*, λ^*) есть седловая точка функции Лагранжа $L(x, \lambda)$.

При поиске седловой точки функции Лагранжа можно выделить прямую и двойственную задачи.

Прямая задача состоит в поиске седловой точки (x^*, λ^*) путем решения задачи:

$$L(x^*, \lambda^*) = \min_{x \in D(x^*)} (L(x, \lambda)), \quad g_i(x^*) \leq 0, \quad \lambda_i^* \cdot g_i(x^*) = 0, \quad i = 1, 2, \dots, m$$

Для постановки двойственной задачи предварительно определим в области $\lambda \geq 0$ функцию $w(\lambda)$ с помощью следующего условия:

$$w(\lambda) = \min_{x \in D} (L(x, \lambda)).$$

Такое условие выполняется, если функции $f(x), g_i(x)$ будут непрерывными в области $D(x^*)$, причем множество $D(x^*)$ компактно. Тогда поиск седловой точки можно осуществить, решая задачу:

$$\min_{\lambda} (w(\lambda)) = \max_{\lambda} \left(\min_{x \in D(x^*)} (L(x, \lambda)) \right), \lambda \in \mathbb{R}^{m+}.$$

Такую задачу будем называть дуальной (двойственной) задачей для задачи нелинейного программирования:

$$f(x) \rightarrow \min, x \in D(x^*) \subset \mathbb{R}^n.$$

Следует отметить важное свойство: дуальная функция $w(\lambda)$ вогнута, как и функция λ .

Численные методы и алгоритмы условной оптимизации.

Большинство существующих методов в нелинейном программировании можно разделить на два основных класса:

- прямые методы;
- методы, использующие понятие двойственности.

Прямые методы характеризуются тем, что они имеют дело непосредственной задачей. Методы порождают последовательность решений, обеспечивая монотонное убывание целевой функции. То есть, если итерационный процесс прерывается, эти методы обеспечивают приближенное решение, удовлетворяющее ограничениям. Однако, с другой стороны, эти методы являются неудобными, так как требуют в качестве начального значения задания внутренней точки из области ограничений, что является достаточно сложной задачей в общем случае, а, кроме того, для них бывает затруднительно обеспечить свойство глобальной сходимости.

В противоположность этому двойственные методы являются более «сильными» и обеспечивают глобальную сходимость. Зато эти методы не дают решения в ходе процесса поиска – решение может быть получено только в конце процесса сходимости.

Прямые методы.

Главными прямыми алгоритмами являются методы возможных направлений, методы проекционного градиента и обобщенного приведенного градиента. /Мину с168/

Основная схема прямых алгоритмов заключается в использовании методов без ограничений с учетом требований, диктуемых наличием ограничений. В методах возможных направлений выбирается начальная точка, удовлетворяющая ограничениям, и затем отыскивается возможное перемещение h такое, чтобы:

- малое перемещение в этом направлении не выводило из множества $D(x^*)$ допустимых решений;
- целевая функция $f(x)$ в выбранном направлении строго убывала.

Затем производится перемещение в полученном направлении до получения новой точки, лучшей, чем предыдущая точка. Например, отыскивается минимум целевой функции по направлению h с учетом границ множества $D(x^*)$. То есть, ищутся направление и масштабный коэффициент шага одномерного поиска с учетом следующих условий:

$$(\nabla f(x)) \cdot h \rightarrow \min, (\nabla g_i(x))^T \cdot h \geq 0.$$

Однако этот выбор, обладая преимуществом простоты, приводит к серьезным затруднениям. Прежде всего, при нелинейных ограничениях, даже очень малое перемещение может приводить к выходу из множества решений. Поэтому после каждого шага одномерного поиска необходимо осуществлять специальную процедуру возвращения в допустимую область. Это, кроме резкого вычислительного усложнения и потери универсальности, может привести и к нарушению условий сходимости процесса поиска. Это может произойти из-за того, что в процессе организации поиска, с учетом только имеющихся активных ограничений, новое ограничение также станет активным и выбранное направление h может резко измениться.

где $x^{(k)}$ текущее решение и $g_i(x^{(k)}) = 0$ текущие активные ограничения. Далее, на основе базиса линейного оператора, реализуемого с помощью матрицы B , осуществляется разложение вектора оптимизируемых параметров $x^{(k)}$ на базисные (зависимые) параметры (вектор $x_B^{(k)}$) и внебазисные (независимые) переменные (вектор $x_P^{(k)}$). Затем строится вектор приведенного градиента по следующей формуле:

$$u_p = \frac{\partial f(x^{(k)})}{\partial x_P^{(k)}} - \left(\frac{\partial f(x^{(k)})}{\partial x_B^{(k)}} \right) \cdot \left(\frac{\partial g(x^{(k)})}{\partial x_B^{(k)}} \right)^{-1} \cdot \left(\frac{\partial g(x^{(k)})}{\partial x_B^{(k)}} \right).$$

На основе анализа градиентов по каждому из векторов формируется необходимое направление поиска, и строятся соответствующие итерационные переходы к новой точке. Данный класс методов характеризуется достаточно сложными вычислениями и, из-за использования процедуры линеаризации функций ограничений, ограниченной сходимостью.

Оптимизация с ограничениями при помощи решения уравнений Куна-Таккера.

Принцип данных методов состоит в прямом решении уравнений Куна - Таккера с помощью алгоритма Ньютона **/Мину с191/**.

Эти методы могут рассматриваться как исходно-двойственные, так как действуют одновременно в пространстве исходных переменных и в пространстве множителей Куна - Таккера (двойственных переменных). При условиях регулярности функций $f(x)$ и $g_i(x)$ данные методы обеспечивают суперлинейную или квадратичную скорость сходимости. Однако, для обеспечения глобальной сходимости эти методы должны комбинироваться с другими алгоритмами.

Построим для задачи:

$$f(x) \rightarrow \min, \quad x \in D(x^*) \subset \mathbb{R}^n, \quad D(x^*) = \{x \in \mathbb{R}^n : g_i(x) \leq 0, \quad \|x - x^*\| \leq \varepsilon, \quad i = 1, 2, \dots, m\}$$

условия Куна-Таккера:

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i \cdot \nabla g_i(x^*) = 0, \quad \lambda_i \cdot \nabla g_i(x^*) = 0, \quad i = 1, 2, \dots, m.$$

Тогда поиск решения исходной задачи сводится к решению нелинейной системы уравнений с $(n + m)$ неизвестными (x, λ) . Метод Ньютона состоит в том, чтобы, исходя из точки $(x^{(k)}, \lambda^{(k)})$, линеаризовать систему уравнений в окрестности точки $x^{(k)}$ и определить точку $(x^{(k+1)}, \lambda^{(k+1)})$ как решение линеаризованной системы уравнений:

$$\begin{aligned} & \nabla f(x^{(k)}) + \sum_i \lambda_i^{(k)} \nabla g_i(x^{(k)}) + [\nabla^2 f(x^{(k)}) + \sum_i \lambda_i^{(k)} \nabla^2 g_i(x^{(k)})](x^{(k+1)} - x^{(k)}) + \\ & + \sum_i (\lambda_i^{(k+1)} - \lambda_i^{(k)}) \cdot \nabla g_i^T(x^{(k)}) \cdot (x^{(k+1)} - x^{(k)}) = 0, \end{aligned}$$

$$g_i(x^{(k)}) + \nabla g_i^T(x^{(k)}) \cdot (x^{(k+1)} - x^{(k)}) = 0.$$

Введем функцию Лагранжа:

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i \cdot g_i(x).$$

Если взять гессиан от функции Лагранжа в точке $(x^{(k)}, \lambda^{(k)})$, то можно записать следующее соотношение:

$$H^{(k)} = \nabla_{xx}^2 L(x^{(k)}, \lambda^{(k)}) = \nabla^2 f(x^{(k)}) + \sum_{i=1}^m \lambda_i^{(k)} \cdot \nabla^2 g_i(x^{(k)}).$$

Соответственно, запишем матрицу якобиана от функций $g_i(x)$ в точке $x^{(k)}$:

$$J^{(k)} = (\nabla g_1(x^{(k)}), \nabla g_2(x^{(k)}), \dots, \nabla g_m(x^{(k)}))^T = \frac{\partial g}{\partial x}(x^{(k)}).$$

Тогда можно записать в следующую систему:

$$\begin{pmatrix} H^{(k)} & (J^{(k)})^T \\ J^{(k)} & 0 \end{pmatrix} \cdot \begin{pmatrix} x^{(k+1)} - x^{(k)} \\ \lambda^{(k+1)} - \lambda^{(k)} \end{pmatrix} = \begin{pmatrix} -\nabla f(x^{(k)}) - (J^{(k)})^T \cdot \lambda^{(k)} \\ -g(x^{(k)}) \end{pmatrix}.$$

Или, проведя преобразования, получим следующие уравнения:

$$\begin{pmatrix} H^{(k)} & (J^{(k)})^T \\ J^{(k)} & 0 \end{pmatrix} \cdot \begin{pmatrix} x^{(k+1)} - x^{(k)} \\ \lambda^{(k+1)} - \lambda^{(k)} \end{pmatrix} = \begin{pmatrix} -\nabla f(x^{(k)}) \\ -g(x^{(k)}) \end{pmatrix}.$$

Положим $y^{(k)} = x^{(k+1)} - x^{(k)}$. Тогда систему можно записать, как следующее итерационное соотношение:

$$\begin{aligned} H^{(k)} \cdot y^{(k)} + (J^{(k)})^T \cdot \lambda^{(k+1)} &= -\nabla f(x^{(k)}) \\ J^{(k)} \cdot y^{(k)} &= -g(x^{(k)}) \end{aligned}$$

Легко заметить, что $y^{(k)}$ есть решение задачи квадратичной оптимизации:

$$\begin{aligned} \min & \left(\frac{1}{2} \cdot y^T \cdot H^{(k)} \cdot y + \nabla f(x^{(k)}) \cdot y \right) \\ & J^{(k)} \cdot y + g(x^{(k)}) = 0 \end{aligned}$$

При этом, $\lambda^{(k+1)}$ есть не что иное, как оптимальный двойственный вектор этой квадратичной задачи. Такое расширение алгоритма Ньютона называется методом Вильсона.

Для решения последней задачи квадратичной оптимизации можно воспользоваться, например, алгоритмом приведенного градиента. Для ограничений, имеющих тип только неравенств, задачу можно записать в следующем виде:

$$\begin{aligned} \min & \left(\frac{1}{2} \cdot y^T \cdot H^{(k)} \cdot y + \nabla f(x^{(k)}) \cdot y \right) \\ & J^{(k)} \cdot y + g(x^{(k)}) \leq 0 \end{aligned}$$

Методы штрафов.

Общий принцип таких методов заключается в замене исходной задачи на решение последовательности экстремальных задач без ограничений. Класс таких методов условно делится на группу методов штрафов (внешние и внутренние штрафы) и группу методов двойственности по Лагранжу. В данном курсе будем рассматривать только группу методов штрафов.

Метод внешних штрафов.

Рассмотрим общий принцип методов штрафов */Мину с 199, Фиакко-МакКормик с 57/*. Пусть имеется задача условной оптимизации вида:

$$f(x) \rightarrow \min, \quad x \in D(x^*) \subset \mathbb{R}^n, \quad D(x^*) = \{x \in \mathbb{R}^n : g_i(x) \leq 0, \quad \|x - x^*\| \leq \varepsilon, \quad i = 1, 2, \dots, m\}.$$

Пусть имеется функция $h: \mathbb{R} \rightarrow \mathbb{R}$ следующего вида:

$$h(y) = \begin{cases} h(y), & y \leq 0, y \in \mathbb{R} \\ h(y) = \infty, & y > 0 \end{cases}$$

Рассмотрим следующую задачу без ограничений:

$$\varphi(x) = f(x) + H(x) \rightarrow \min, \quad x \in \mathfrak{R}^n,$$

где функция H , называется функцией штрафа и определяется соотношением:

$$H(x) = \sum_{i=1}^m h(g_i(x)).$$

Очевидно, что функция штрафа H по построению является разрывной функцией, как и функция φ . Однако, можно применить и непрерывные и непрерывно дифференцируемые функции штрафа. В, частности, можно принять:

$$h(y) = \begin{cases} h(y), & y \leq 0, y \in \mathfrak{R} \\ h(y) = y^2, & y > 0 \end{cases}.$$

Заменим решение такой задачу на решение экстремальной задачи без ограничений:

$$\varphi(x, r) = f(x) + r \cdot H(x) \rightarrow \min, \quad x \in \mathfrak{R}^n, r \in \mathfrak{R}.$$

Здесь величина $r \in \mathfrak{R}$ и называется коэффициентом штрафа. Такую задачу оптимизации будем называть задачей с внешним штрафом. Если, рассмотреть решение $x(r)$ задачи при значении $r \rightarrow \infty$, то можно сделать предположение, что $x(r) \rightarrow x^*$.

Рассмотрим снова исходную задачу оптимизации:

$$f(x) \rightarrow \min, \quad x \in D(x^*) \subset \mathfrak{R}^n.$$

Запишем для ее решения x^* модифицированные условия Куна - Таккера:

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i \cdot \nabla g_i(x^*) = 0, \quad \lambda_i \cdot \nabla g_i(x^*) = r, \quad i = 1, 2, \dots, m.$$

Решим уравнения $\lambda_i \cdot \nabla g_i(x^*) = r, \quad i = 1, 2, \dots, m$ относительно λ_i и подставим полученные выражения в первое уравнение. Тогда получим следующее соотношение:

$$\nabla f(x^*) + \sum_{i=1}^m \frac{r}{g_i(x(r))} \cdot \nabla g_i(x(r)) = 0.$$

Данное уравнение определяет градиент следующей функции (называемой логарифмической функцией штрафа или внутренней функцией штрафа):

$$L(x, r) = f(x) + r \cdot \sum_{i=1}^m \ln(g_i(x)),$$

которая обращается в нуль в точке $x(r)$. То есть в точке $x(r)$ выполнены необходимые условия минимума для задачи без ограничений с функцией $L(x, r)$. Если, рассмотреть решение $x(r)$ при значении $r \rightarrow 0$, то видно, что решение $x(r) \rightarrow \bar{x}^*$ будет совпадать с точкой минимума функции

при функции $f(x)$ при отсутствии штрафа $H(x, r) = r \cdot \sum_{i=1}^m \ln(g_i(x))$.

Теорема (о внешних штрафных функциях). /Милу с2011/. Пусть $H: \mathfrak{R}^n \rightarrow \mathfrak{R}$ функция внешнего штрафа такая, что $H(x) \geq 0$ и удовлетворяющая условиям $H(x) = 0$ на множестве $D(x) = \{x: g_i(x) \leq 0, i = 1, 2, \dots, m\}$. Пусть также функции $f(x), H(x)$ являются непрерывными, а множество $D(x)$ замкнуто и выполняется одно из двух условий: $f(x) \rightarrow \infty$ при $\|x\| \rightarrow \infty$

или $D(x)$ ограничено, функция $H(x) \rightarrow \infty$ при $\|x\| \rightarrow \infty$. Тогда при $r \rightarrow \infty$ будут справедливы следующие утверждения:

а) последовательность точек $x(r)$ имеет, по крайней мере, одну точку сгущения и всякая точка сгущения этой последовательности есть (глобальное) оптимальное решение задачи:

$$f(x) \rightarrow \min, \quad x \in D(x^*) \subset \mathbb{R}^n, \quad D(x^*) = \{x \in \mathbb{R}^n : g_i(x) \leq 0, \quad \|x - x^*\| \leq \varepsilon, \quad i = 1, 2, \dots, m\};$$

$$b) \quad H(x(r)) \xrightarrow{r \rightarrow \infty} 0.$$

(без доказательства).

Метод внутренних штрафов (метод барьерных функций).

Главный недостаток предыдущего методов внешних штрафных функций заключается в том, что оптимальная точка x^* аппроксимируется последовательностью решений $\{x^{(k)}\}$, полученных при различных коэффициентах штрафа $r^{(k)}$, не принадлежащих множеству $D(x)$. Именно поэтому, были предложены другие методы штрафной функции, при которых траектория движения к оптимальной точке располагается внутри множества $D(x)$.

Пусть множество $D(x)$ является непустым, и каждая граничная точка есть предел последовательности точек, лежащих внутри $D(x)$. Рассмотрим следующую функцию:

$$B(x) = \sum_{i=1}^m \frac{1}{g_i(x(r))}.$$

Она удовлетворяет условиям $B(x) \geq 0$ для точек внутри множества $x \in \text{int}(D(x))$ и $B(x) \rightarrow \infty$, если точка x стремится к границе множества $D(x)$. Кроме того, при непрерывных функциях $g_i(x)$ функция $B(x)$ непрерывна внутри множества $D(x)$. Рассмотрим следующую функцию:

$$\varphi(x, r) = f(x) + r \cdot B(x),$$

где r – коэффициент штрафа.

Если функция f непрерывна на множестве $D(x)$ и выполняется одно из условий:

$f(x) \rightarrow \infty$ при $\|x\| \rightarrow \infty$ или множество $D(x)$ ограничено. Тогда при любом $r > 0$ функция $\varphi(x, r)$ имеет минимум на множестве $D(x)$, причем $D(x^*, r) \in \text{int}(D(x))$.

Таким образом, выбирая начальное значение коэффициента r_1 , находим минимум функции $\varphi(x, r_1)$. При этом, будем использовать в качестве начальной точки $x^{(0)} \in \text{int}(D(x))$.

Заметим, что при минимизации $\varphi(x, r_1)$ процесс поиска не может пересечь границу множества $D(x)$, так как при приближении к ней функция $\varphi(x, r_1) \rightarrow \infty$. Если величина $r_1 \cdot B(x^{(1)})$ достаточно мала, то тогда точка $x^{(1)}$ является хорошим приближением оптимума функции f . В противном случае, выбираем значение $r_2 < r_1$, и ищем следующее приближение точки $x^{(2)}$. Процесс продолжается до получения приемлемой аппроксимации оптимальной точки.

Теорема (о внутренних штрафных функциях). Пусть $B: \mathbb{R}^n \rightarrow \mathbb{R}$ функция внутреннего штрафа, удовлетворяющая условиям:

$$B(x) \geq 0, \quad \forall x \in \text{int}(D(x));$$

$B(x) \rightarrow \infty$, если точка x стремится к границе множества $D(x)$;

функция $B(x)$ непрерывна на множестве $\text{int}(D(x))$.

Пусть также f непрерывная функция и выполняется одно из двух условий:

$f(x) \rightarrow \infty$ при $\|x\| \rightarrow \infty$ или множество $D(x)$ ограничено. Тогда при $r \rightarrow 0$ имеем:

а) последовательность точек $x(r)$ имеет, по крайней мере, одну точку сгущения и всякая точка сгущения этой последовательности есть (глобальное) оптимальное решение задачи

$$f(x) \rightarrow \min, x \in D(x^*) \subset \mathbb{R}^n, D(x^*) = \{x \in \mathbb{R}^n : g_i(x) \leq 0, \|x - x^*\| \leq \varepsilon, i = 1, 2, \dots, m\}$$

б) $r \cdot B(x(r)) \rightarrow 0$.

Следует отметить, что для достижения высокой точности требуется выбор, либо больших (внешние штрафы), либо слишком малых (внутренние штрафы) коэффициентов штрафа, что приводит к плохой обусловленности задач поиска и вычислительным трудностям.

Определенные сложности представляет для методов внутренних штрафов выбор начальной точки $(x^{(0)}, r_0)$.

Рассмотрим процедуру начального выбора коэффициента штрафа r . Так как справедливо равенство $\varphi(x, r) = f(x) + r \cdot B(x)$, то можно записать следующее соотношение:

$\nabla \varphi(x, r) = \nabla f(x) + r \cdot \nabla B(x)$. Квадрат нормы вектора $\nabla \varphi(x, r)$ будет равен:

$$(\nabla f(x)) \cdot (\nabla f(x))^T + 2r \cdot \nabla f(x) \cdot B(x) + r^2 \cdot (\nabla B(x))^T \cdot \nabla B(x).$$

Минимум этой нормы достигается при значении:

$$r = \frac{-\nabla f(x) \cdot \nabla B(x)}{(\nabla B(x))^T \cdot \nabla B(x)}.$$

Уменьшить значение коэффициента штрафа можно с помощью следующего итерационного процесса $r_{k+1} = r_k / c$, где $c = 10$. **/Банди с118/**

Многокритериальная оптимизация.

Требования технического задания на проектирование систем управления обычно носят многокритериальный характер. Например, при заданной структуре системы обеспечить такие настройки регулятора, при которых система будет иметь наименьшее время затухания переходных процессов при минимальных значениях перерегулирования. При этом, требуется, чтобы ошибка выхода из-за воздействия на вход системы случайного возмущающего воздействия, была также минимальной.

Таким образом, задача оптимизации системы сводится к поиску экстремума не скалярной целевой функции, а векторной целевой функции вида $f = (f_1, f_2, \dots, f_m)^T$, где m – количество оптимизируемых скалярных критериев. Критерии f_i , $i = 1, 2, \dots, m$ будем называть частными или локальными, а критерий f – векторным критерием. По своему характеру критерии делятся на количественные и качественные. В дальнейшем будем рассматривать только количественные критерии, предполагая, что для качественных критериев уже разработана шкала интервальная измерений.

Выбор оптимального решения из множества всех решений $x \in X \subseteq \mathbb{R}^n$ сводится к выбору оптимальной оценки из множества достижимых оценок вида:

$$Y = f(X) = \{y \in \mathfrak{R}^m : y = f(x), x \in X \subseteq \mathfrak{R}^n\}.$$

Множество $Y \subseteq \mathfrak{R}^m$ будем называть критериальным.

Между тем, многие из скалярных критериев являются противоречивыми: улучшение одного из них при изменении оптимизируемых параметров приводит к ухудшению других. Возникает проблема выбора разумного компромисса, то есть определение такого допустимого решения x^* , при котором все скалярные критерии, входящие в векторную целевую функцию, будут принимать приемлемые значения.

Введем совокупность предпочтений, которые представим в виде бинарных отношений порядка для каждого k -го частного критерия. /Подиновский с46/

Введем понятие отношения (строгого) предпочтения: $x^i \succ_k x^j$, если $f_k(x^i) < f_k(x^j)$. То есть решение x^i (строго) предпочтительнее решения x^j , если критерий $f_k(x^i)$ более предпочтителен критерия $f_k(x^j)$.

Отношение безразличия или эквивалентности: $x^i \propto_k x^j$, если $f_k(x^i) = f_k(x^j)$. То есть решение x^i эквивалентно решению x^j , если критерий $f_k(x^i)$ эквивалентен критерию $f_k(x^j)$.

Отношение нестрогого предпочтения, которое означает, что $x^i \succeq_k x^j$, если $f_k(x^i) \leq f_k(x^j)$.

То есть решение x^i не менее предпочтителен, чем x^j .

Будем предполагать в дальнейшем, что частные критерии независимы по предпочтению.

При рассмотрении различных решений задачи оптимизации приходится устанавливать отношение сравнения \mathbb{Q} между различными векторными целевыми функциями:

$$\begin{aligned} (f_1, f_2, \dots, f_m) \mathbb{Q} (f_1^1, f_2^1, \dots, f_m^1), \quad f_1 \leq f_1^1; \\ (f_1^1, f_2^1, \dots, f_m^1) \mathbb{Q} (f_1^1, f_2^1, f_3^1, \dots, f_m^1), \quad f_2 \leq f_2^1; \\ \dots\dots\dots \\ (f_1^1, f_2^1, \dots, f_{m-1}^1, f_m^1) \mathbb{Q} (f_1^1, f_2^1, f_3^1, \dots, f_{m-1}^1, f_m^1), \quad f_m \leq f_m^1 \end{aligned}$$

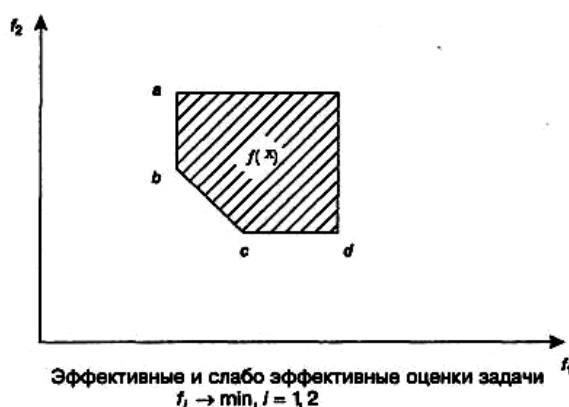
В этом случае на множестве оценок $Y \subseteq \mathfrak{R}^m$ векторных оценок вводят отношение нестрогого предпочтения \triangleleft , совпадающее с частичным порядком векторов f . Соответственно можно ввести отношение и на множестве решений: $x^i \triangleleft x^j$, если $f_k(x^i) \triangleleft f_k(x^j)$.

Оценки, минимальные по отношению \triangleleft , будем называть эффективными или оптимальными по Парето. Множество, включающее все эффективные решения, будем обозначать через $P(X)$ и называть множеством Парето. Очевидно, что для $\forall \bar{x} \in P(x)$ и $\forall \tilde{x} \notin P(x)$ будут справедливы неравенства $f_i(\bar{x}) \leq f_i(\tilde{x}), i = 1, 2, \dots, m$, причем, по крайней мере, одно из неравенств строгое. Очевидно, что решение многокритериальной задачи следует искать среди множества $P(X)$.

Данное утверждение носит название принципа Парето. В противном случае, всегда найдется точка x , оказывающаяся более предпочтительной, независимо от важности отдельных частных критериев. Оценки, минимальные по отношению \triangleleft , будем называть слабо эффективными или слабо оптимальными по Парето.

Иначе говоря, решение называется слабо эффективным, если оно не может быть улучшено сразу по всем критериям. Это множество будем называть множеством слабо эффективных оценок и обозначать через $S(X)$. Очевидно, что справедливо следующее соотношение: $P(X) \subseteq S(x)$.

На рисунке показано множество $S(X)$ состоящее из отрезков (a,b,c,d). Множество $P(X)$ состоит из отрезка (b,c).



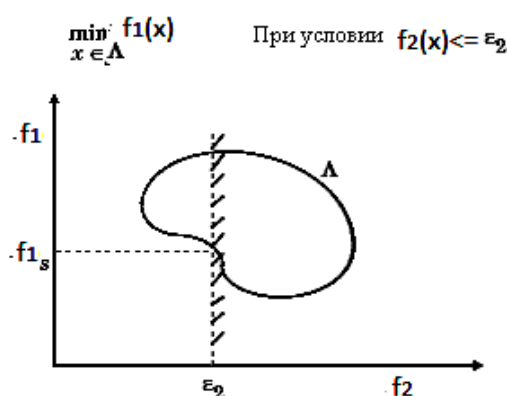
Хотя эффективное решение, чаще всего не единственное, все-таки множество эффективных решений значительно уже, чем исходное множество всех решений. Поэтому построение множества эффективных решений (или их оценок) является одним из первых этапов решения задачи многокритериальной оптимизации. В инженерной практике обычно проблему многокритериальной оптимизации решают путем ее сведения к некоторому единому скалярному критерию.

Метод главного критерия или ε ограничений.

В методе главного критерия выбирается одна скалярная функция, входящая в состав векторного критерия, наиболее важная, с точки зрения проектировщика. На остальные составляющие векторной целевой функции накладываются дополнительные ограничения, определяющие их допустимую область варьирования. В этом случае задача безусловной оптимизации сводится к задаче условной оптимизации вида:

$$f_j(x) \rightarrow \min, \quad x \in D = \{0 \leq f_i(x) \leq \varepsilon_i, i = 1, 2, \dots, j-1, j+1, \dots, m\}$$

где ε_i — некоторые заданные значения.



Однако проблемой данного метода является подходящий выбор ε_i , который мог бы гарантировать допустимость некоего решения. Следующий недостаток данного метода заключается в необходимости использования жестких ограничений, которые не всегда являются адекватными для точного построения задаваемых целей. Подобные методы строятся на основе выбора некоего приоритета для задаваемых целей. Процедура оптимизации выполняется в соответствии с выбранными приоритетами и в пределах допустимости принятых границ.

Метод весовых коэффициентов.

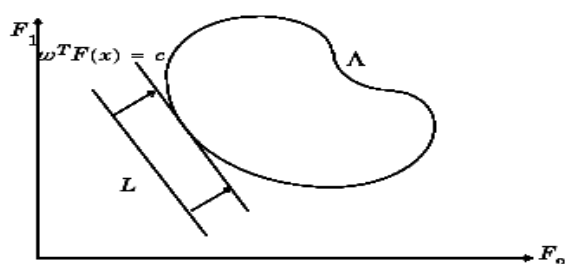
Данный метод основан на линейном объединении всех частных критериев в один в форме:

$$F(x) = \sum_{i=1}^m w_i \cdot f_i(x) \rightarrow \min_{x \in X},$$

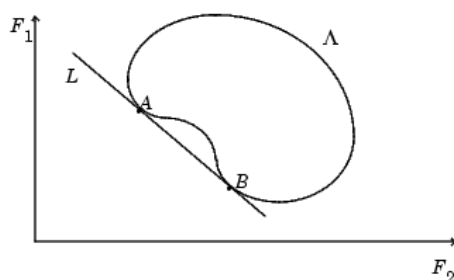
где $w_i > 0$, $\sum_{i=1}^m w_i = 1$. При этом предполагается, что все $f_i(x) \geq 0$ при $x \in X \subseteq \mathbb{R}^n$ /Трифонов/.

Далее уже к данной задаче оптимизации уже может быть применен стандартный алгоритм оптимизации для скалярного критерия. В этом случае рассматриваются взвешенные коэффициенты для каждой из выбранных целей. Взвешенные коэффициенты необязательно должны напрямую соответствовать относительной значимости соответствующей цели или принимать во внимание взаимовлияние между конкретно выбранными целями. Более того, границы не улучшаемых решений могут быть не достигнуты, так что определенные решения являются по существу недостижимыми.

Данный метод имеет следующую геометрическую интерпретацию. Рассмотрим случай двух взятых целей, как это представлено на рисунке. Отобразим линию $w^T F(x) = c$ в пространстве целевых функций. Минимизацию уравнения можно интерпретировать как поиск такого значения c , при котором линия L будет как раз касаться границы области при проведении данной процедуры решения задачи вне выбранной области. Путем подбора весов возможно, по существу, задать наклон линии L таким образом, что, в свою очередь, приводит к искомому решению в точке касания линии L с границей области поиска решения.



Проблема выпуклости становится актуальной в случае, когда нижняя граница области не является выпуклой. В этом случае множество не ухудшаемых решений между точками A и B не является достижимым при использовании подобной процедуры.



Задачи целочисленного программирования.

Основной класс задач дискретной (цифровой) оптимизации составляют такие задачи, в которых на значения всех или части переменных наложено требование их целочисленности. То есть, общую задачу целочисленного программирования можно сформулировать в виде:

$$f(x) \rightarrow \text{extr}, \quad x \in D(x) \subseteq \mathbb{Z}^n.$$

В ряде случаев, когда $D(x) \subseteq \mathbb{Z}^{n_1} \otimes \mathbb{R}^{n_2}$, где $n_1 + n_2 = n$, такие задачи называются задачами частичного целочисленного программирования. В частности, задача параметрической идентификации линейного стационарного объекта в классе передаточных функций

$W(s) = \frac{B_m(b, s)}{A_n(a, s)}$, сводится к минимизации некоторого критерия невязок наблюдаемых

(измеряемых) и вычисляемых величин выхода и, определению, как целочисленных положительных значений m, n , так и действительных оценок векторов параметров a, b .

Довольно часто, в качестве ограничений, в целочисленных экстремальных задачах, участвуют логические условия типа «да-нет». Тогда для формальной записи задачи оптимизации приходится вводить булевы переменные из множества $\{0, 1\}$ /Сергиенко с10/.

Для многих задач линейного и квадратичного программирования с целочисленными коэффициентами разработаны алгоритмы получения точного решения. Однако такие алгоритмы характеризуются экспоненциальной временной сложностью и относятся к классу NP-полных задач. Поэтому для задач оптимизации целочисленного программирования разработаны достаточно эффективные приближенные алгоритмы, которые позволяют за счет снижения точности в требуемых пределах значительно снизить трудоемкость получения решения (так называемые ε -приближенные алгоритмы).

Среди точных методов решения задач целочисленного программирования получили методы отсечения, идея которых впервые была предложена Данцигом и в дальнейшем развита в работах Гомори. В основном эти методы используются для решения задач целочисленного или частично целочисленного линейного программирования.

Для решения этих же задач также широко используются методы ветвей и границ, которые основаны на следующей вычислительной схеме: последовательном рассмотрении всех решений из конечного множества вариантов и замене полного их перебора сокращенным, за счет отбрасывания неперспективных решений, направленным перебором.

Значительная часть приближенных алгоритмов решения базируется на использовании вычислительных схем точных процедур, таких как методы отсечений, ветвей и границ, последовательного анализа вариантов и прочие. В качестве основных характеристик приближенных алгоритмов на практике чаще всего используются абсолютная и относительная погрешности целевой функции:

$$\varepsilon_1 = f(x^*) - f(x), \quad \varepsilon_2 = \frac{|f(x^*) - f(x)|}{|f(x^*)|},$$

где x - допустимое решение задачи, называемое приближенным ее решением; x^* - точное оптимальное решение задачи.

К приближенным методам относятся также такие алгоритмы, как метод вектора спада, метод направляющих окрестностей и метод случайного поиска.

Точные методы решения задач целочисленного программирования.

Метод последовательного анализа вариантов.

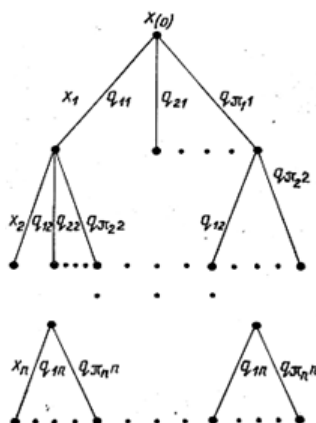
Методы этого класса основаны на пошаговом конструировании решений и отсеке в процессе такого конструирования тех, которые не могут быть достроены до оптимальных.

Рассмотрим следующую задачу дискретной оптимизации:

$$\min(f(x) : x \in D(x) \subseteq \mathbb{Z}^n), \quad D(x) = \{x \in Q^n : g_i(x) \leq 0, i=1:m\}, \quad (21)$$

где $D(x) = \{x \in Q^n : g_i(x) \leq 0, i=1, 2, \dots, m\}$, $Q^j = \{q_{1j}, q_{2j}, \dots, q_{\pi_j j}\}$ - некоторое заданное конечное множество. Обозначим множество всех решений данной задачи через Ω . Решение называется допустимым, если оно принадлежит множеству $D(X)$. Множество всех допустимых

решений обозначим через Ω_f . Тогда вектор $x_{(p)} = (x_1, x_2, \dots, x_p)^T$, где $p < n$, будем называть частичным решением, если $x_j \in q_j, j = 1, 2, \dots, p$. Если, при этом, данный вектор может быть достроен до допустимого решения $x_{(n)} = (x_1, x_2, \dots, x_p, x_{p+1}, \dots, x_n)^T$, то будем называть его допустимым частичным решением. Множество всех частичных решений такой задачи схематически можно изобразить в виде дерева решений H . Вершинам его однозначно соответствуют частичные решения, причем висячим вершинам – полные решения ($p = n$).



Процесс решения задачи, с помощью методов последовательного анализа решений, можно представить как схему продвижения по дереву H , начиная с его корня, отвечающего частичному решению $x_{(0)}$ /Ковалев с79/. Чтобы задать алгоритм решения конкретной задачи, необходимо определить правило выбора частичных решений, подлежащих развитию на каждом шаге. Правило сводится к построению набора σ элиминирующих тестов, которые осуществляют отсев тех частичных решений, которые не могут быть достроены ни до допустимых решений, ни до оптимальных решений. Формирование таких тестов осуществляется на основе специфик решаемой задачи, то есть заданием целевой функции и множеством $D(X)$. Применяя к любому подмножеству $h \in \Omega$ (содержащему как полные, так и частичные решения) тесты, можно выявить в множестве h недопустимые решения.

Рассмотрим простейший пример, иллюстрирующий метод последовательного анализа решений.

$-x_1 \cdot x_2 \cdot x_3 \rightarrow \min; x_1 + x_2 + x_3 = N, -x_1 < 0, -x_2 < 0, x_2 \leq 3, -x_3 < 0; x_1, x_2, x_3 \in \mathbb{N} / (0)$, где $N = 7$.

Набор допустимых векторов $x_1 \in h_1$, удовлетворяющих условию $x_1 \in D(x)$, состоит из следующего списка: $\{(1), (2), \dots, (7)\}$. Соответственно список $x_2 \in h_2$ имеет следующий вид $\{(1,1), (1,2), (1,3), (2,1), \dots, (6,3)\}$ и так далее. Введем элиминирующий тест ξ_1 , определяемый для первого шага, следующим условием $x_1 \leq N$. В дальнейшем ветви дерева H , не прошедшие данный тест, будем помечать кружками.

Для первого шага алгоритма ($k = 1$) тест $\xi_{1,1}$, входящий в состав теста ξ_1 , определяется условиями $N - x_1^{(1)} > 0, x_1^{(1)} > 0$. В соответствии с этим тестом ветвь (7) имеет пустое множество Δh_1 решений. Для второго шага решений ($k = 2$) тест $\xi_{1,2}$ определяется условиями

$N - x_1^{(2)} - x_2^{(2)} > 0, x_1^{(2)} > 0, x_2^{(2)} > 0$. В соответствии с этим тестом ветви $\{(4,3), (5,2), (5,3), (6,1), (6,2), (6,3)\}$ имеют пустое множество Δh_2 решений.

Определим теперь элиминирующий тест ξ_2 , основанный на вычислении нижней границы (оценки) для оптимального значения целевой функции $f(x)$ на каждом из множеств Ω_f .

На втором шаге рассмотрения дерева решений, значение целевой функции $f(x)$ можно определить с помощью следующего равенства: $f(x_{(2)}) = -(N - x_1^{(2)} - x_2^{(2)}) \cdot x_1^{(2)} \cdot x_2^{(2)}$, где $x_{(2)} \in \Omega_f$. В

соответствии с известным неравенством $x_1 + x_2 \geq 2\sqrt{x_1 \cdot x_2}$, и условиями

$N - x_1 - x_2 > 0, x_1 > 0, x_2 > 0$, можно записать следующее соотношение:

$$-(N - x_1 - x_2) \cdot x_1 \cdot x_2 \leq -N \cdot (-x_1 - x_2) \cdot \frac{1}{4} \cdot (x_1^2 + x_2^2) \leq -\frac{1}{4} \cdot (N - x_1 - x_2) \cdot (x_1 + x_2)^2.$$

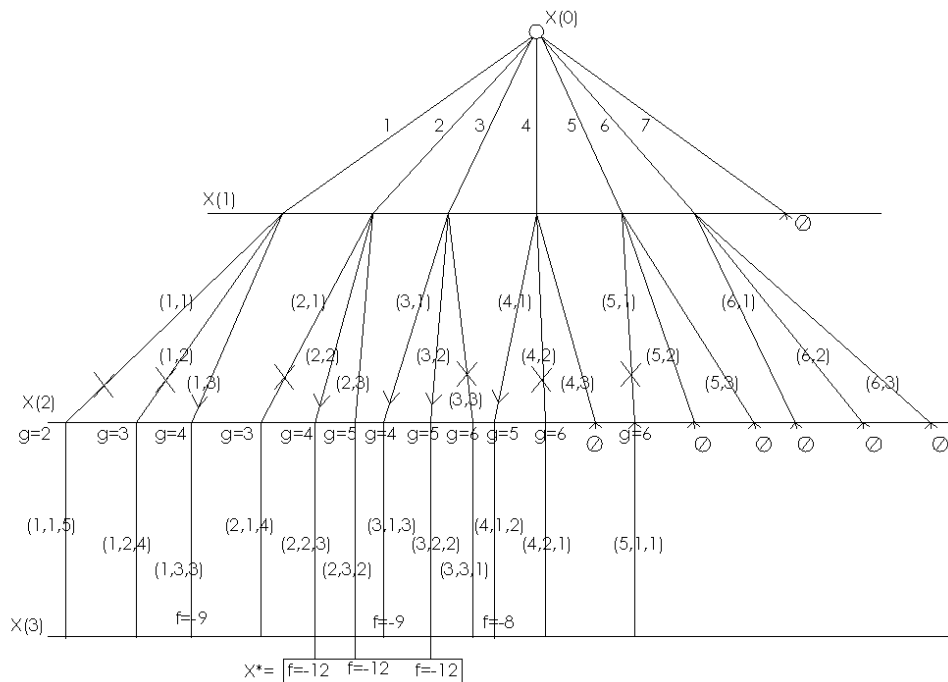
Функция $U(y) = (N - y) \cdot y^2$ имеет максимум при значении $y = \frac{2N}{3}$. Отсюда вытекает, что

нижняя граница для оптимального значения $f(x^*)$ будет достигаться при выполнении неравенства

$$\left\lceil \frac{2N}{3} \right\rceil \leq x_1 + x_2 \leq \left\lfloor \frac{2N}{3} \right\rfloor, \text{ то есть, для данного конкретного примера: } 4 \leq x_1 + x_3 \leq 5.$$

Соответственно можно выделить ветви дерева решений H , для которых необходимо вычислять значения целевой функции на 3-ем шаге (ветви дерева, по которым будет проводиться дальнейший анализ, отмечены стрелками).

В результате анализа полученных 5-ти вычислений (концевых вершин) определяется множество оптимальных точек x^* . Число вариантов перебора при последовательном анализе решений, как видно, значительно сокращается по сравнению со схемой полного перебора. Стратегия поиска задается следующей совокупностью правил ξ_1 (для первого шага) и ξ_2 (для второго шага).



Методы ветвей и границ.

Для всех алгоритмов метода ветвей и границ характерно применение следующего основного принципа: последовательное использование конечности множества вариантов решения задачи, и замена полного их перебора, сокращенным, так называемым частичным перебором. Полного перебора удастся избежать за счет отбрасывания неперспективных множеств решений, то есть таких, которые заведомо не могут содержать искомого оптимального решения задачи. Этот принцип реализуется путем разбиения всего множества допустимых решений на подмножества и построения оценок, позволяющих сделать вывод о том, какие из полученных подмножеств, стоит принимать для дальнейшего рассмотрения. В основе любой удачной реализации такого алгоритма обычно лежит максимальное использование специфики решаемой задачи.

Использование общих схем для решения задач целочисленного программирования позволяет, как правило, описать в сжатом виде основные этапы вычислительного процесса. Однако, для успешного применения таких схем при решении конкретных задач требуется определенная детализация, учитывающая конкретную специфику процесса оптимизации. Одной из таких общих схем является конструкция вычислительного процесса, предложенная Джефрионом и Марстеном для решения частичной целочисленной задачи линейного программирования:

$$F(x, y) = c_1x + c_2y \rightarrow \min, \text{ при ограничениях } Ax + By \geq b; x, y \geq 0,$$

где $x \in \mathbb{Z}^n$ - целочисленный вектор, $y \in \mathbb{R}^m$, а матрицы A, B и вектора c_1, c_2, b имеют соответствующие размеры.

Введем следующие основные понятия метода ветвей и границ. Под релаксацией задачи $P = \{c_1x + c_2y \rightarrow \min\}$ с допустимой областью $G = \{Ax + By \geq b; x, y \geq 0\}$, определяемой соответствующими неравенствами понимается переход к задаче \bar{P} с той же целевой функцией, но с некоторой допустимой областью $\bar{G} \supset G$. Задачи \bar{P} и P имеют следующие свойства:

- а) если задача \bar{P} не имеет допустимых решений, то их не имеет и задача P ;
- в) если оптимальное решение задачи \bar{P} удовлетворяет условиям $Ax + By \geq b; x, y \geq 0$, то оно одновременно является и оптимальным решением задачи P .

Очевидно, что имеет место следующее неравенство:

$$\min_{(x,y) \in G} F(x, y) \geq \min_{(x,y) \in \bar{G}} F(x, y).$$

Для релаксации задач указанного вида, чаще всего, на практике используется отбрасывание условий целочисленности. Получаемая в результате такого допущения задача \bar{P} сравнительно легко поддается решению с помощью хорошо разработанных методов линейного программирования.

Под ветвлением для некоторой задачи принято понимать разбиение всего множества допустимых ее решений на непересекающиеся подмножества. Кандидатом, на произвольном шаге вычислительного процесса, называется задача, которую необходимо «ветвить», а подзадачи, получаемые в результате этого ветвления, называют потомками данной задачи.

Под стратегией в алгоритмах ветвей и границ понимается порядок выбора задач-кандидатов для их последующего анализа. Рекордом назовем значение функции $F(x, y)$, которое соответствует наилучшему из всех полученных решений задач-кандидатов к данному этапу

вычислений. Текущее значение рекорда при описании схемы метода обозначим $\overline{F}(x, y)$.

На практике используются различные способы ветвления. В частности, часто применяют дихотомическое ветвление, когда на некоторую целочисленную переменную накладывают дополнительно два противоречивых ограничения вида:

$$x_i \leq [x_i] \text{ и } x_i \geq [x_i] + 1,$$

где x_i - одна из нецелочисленных компонент в оптимальном решении (\bar{x}, \bar{y}) текущего кандидата, а $[.]$ – целая часть числа.

Под зондированием кандидата будем понимать процесс, если в результате решения релаксированного кандидата придем к одному из следующих выводов.

1. Релаксационный кандидат не имеет допустимых решений.
2. Значение функции $F(x, y)$ для данного релаксационного кандидата удовлетворяет неравенству:

$$F(x, y) \geq \overline{F(x, y)}.$$

То есть данный кандидат не имеет таких допустимых решений, которые были бы лучшими по сравнению с рекордным решением.

3. Получено оптимальное решение релаксационного кандидата, которое одновременно является допустимым решением для кандидата.

В том случае, когда зондирование кандидат не привело ни к одному из приведенных выше выводов, зондирование кандидата прекращается и осуществляется процесс его ветвления, а полученные потомки включаются в список кандидатов.

На основании введенных определений, описание вычислительного процесса Джефффриона-Марстена можно представить в следующей последовательности.

1. Открыть список задач-кандидатов путем внесения в него задачи P . Положить рекорд $\overline{F(x, y)} = \infty$.
2. Провести анализ списка задач-кандидатов с целью выяснения его непустоты.
 - 2.1. Если список не пуст, то перейти к шагу 3.
 - 2.2. В случае пустоты списка, работа по методу заканчивается. Текущее рекордное решение является оптимальным, если оно соответствует значению рекорда $\overline{F(x, y)} < \infty$. В противном случае, исходная задача не имеет допустимого решения.
3. В качестве текущего кандидата выбрать, по какому-либо правилу с использованием выбранной стратегии, одну задачу из списка.
4. Выбрать релаксацию текущего кандидата.
5. Провести анализ задачи-кандидата с целью выяснения у нее допустимых решений.
 - 5.1. Решить релаксированную задачу.
 - 5.2. Если множество допустимых решений кандидата пусто, то перейти к шагу 2. В противном случае сделать проверку условия: есть ли у кандидата допустимые решения лучшие, чем рекордное. При положительном ответе – перейти к шагу 5.3 ; при отрицательном – исключить из списка данную задачу и перейти к шагу 2.
 - 5.3. Определить, получено ли оптимальное решение кандидата. Если нет, то перейти к шагу 5.4. В противном случае перейти к шагу 2, пересчитав и запомнив значение нового рекорда.
 - 5.4. Проверить по выбранной стратегии, следует ли продолжать зондирование кандидата? Если нет, то перейти к шагу 6. В противном случае модифицировать релаксацию рассматриваемого кандидата и перейти к шагу 5.1.
6. Осуществить разветвление кандидата и включить его потомки в список кандидатов, исключив, при этом, из него самого кандидата. Перейти к шагу 2.

Следует отметить, что эффективное решение многих практических задач целочисленного программирования большой размерности с помощью алгоритмов метода ветвей и границ проблематично, так как оно существенно зависит от конструктивной интерпретации всех основных пунктов приведенной схемы.

Рассмотрим задачу о ранце с одним ограничением, которая формулируется следующим образом /Колпаков с 2/. Задано количество n предметов. Предмет под индексом i

характеризуется весом w_i и ценой $p_i, i = 1:n$. Требуется положить в ранец грузоподъемностью C набор предметов максимальной стоимости. Неформальное описание может быть записано следующим образом:

$$F(x) = \sum_{i=1}^n p_i x_i \rightarrow \max, \text{ при ограничениях } \sum_{i=1}^n w_i x_i \leq C,$$

где $x_i \in \{0,1\}; w_i, p_i > 0; i = 1:n$.

При решении задачи о ранце размерности n полным перебором потребуется просмотреть в наихудшем случае 2^n булевых векторов.

Рассмотрим более эффективный метод ветвей и границ. В качестве оценочной задачи рассмотрим линейную релаксацию задачи о ранце, которая получается заменой дискретных ограничений линейными. Такая задача решается за линейное, относительно переменных, количество операций (метод Данцига решения задачи линейного программирования). Решение релаксационной задачи достигается на наборе переменных x_i , содержащем не более одного дробного (не целого) значения. Такую переменную будем называть дробной переменной. Ветвление состоит в разбиении исходной задачи на две, путем присваивания одной из переменных значений 0 и 1, соответственно. Отсев подзадачи будем производить в следующих случаях.

1. Оценочная задача не имеет решения.
2. Решение оценочной задачи не превосходит наилучшее значение целевой функции (рекорда), из найденных на данный момент значений.
3. Оптимальное решение, рассматриваемой релаксационной задачи, является полностью целочисленным. Тогда это решение, очевидно, является оптимальным решением текущей подзадачи. Если на допустимом решении, целевая функция принимает значение, превосходящее рекорд, то рекордом становится новое значение целевой функции.

Пример.

Найти $2x_1 + 2x_2 \rightarrow \max$, при ограничениях $2x_1 + 2x_2 \leq 3; x_1, x_2 \in \{0;1\}$.

Шаг 1. Задача релаксации \bar{P} имеет решение $x_1 = 1; x_2 = 1/2$. Значение оценки $F(x) = 3$.

Шаг 2. Осуществляем ветвление задачи P на две подзадачи P_0, P_1 , полученные присваиванием дробной переменной x_2 значений 0 и 1 соответственно: $P_0: 2x_1 \rightarrow \max; 2x_1 \leq 3; x_1 \in \{0;1\}$ и

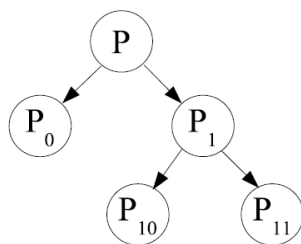
$P_1: 2x_1 \rightarrow \max; 2x_1 \leq 1; x_1 \in \{0;1\}$.

Шаг 3. Решаем релаксационную задачу \bar{P}_0 . Ее решение $x_1 = 1$ является целочисленным и, следовательно, дает допустимое решение задачи P_0 . Итоговое решение имеет вид $x_1 = 1; x_2 = 0$, значение целевой функции (рекорд) равен 2.

Шаг 4. Решаем релаксационную задачу \bar{P}_1 . Ее решение $x_1 = 1/2$ и оценка целевой функции при этом составляет 3. Так как $3 > 2$, то данная задача не удовлетворяет условию отсева и подвергается дальнейшему ветвлению. Присвоим переменной x_1 значения 0 и 1 и ветвлением построим две новые задачи P_{10}, P_{11} с пустым множеством переменных (переменные уже фиксированы для каждой задачи).

Шаг 5. Вторая \bar{P}_{11} задача несовместна, так как нарушаются ограничения. Итоговое решение, получаемое из задачи \bar{P}_{10} , имеет вид $x_1 = 0; x_2 = 1$ и оценка целевой функции равна 2.

Таким образом, максимум целевой функции равен 2. Дерево решений имеет 3 концевые вершины, то есть сложность решения задачи равна 3.



Приближенные методы целочисленного программирования.

Точные методы при решении целочисленных задач большой размерности, к которым относится большинство практически важных задач проектирования систем управления, не могут обеспечить получение окончательного результата за приемлемое время. Поэтому для их решения часто используют приближенные методы.

Одним из эффективных приближенных методов является метод вектора спада, который достаточно успешно используется для решения задач полностью целочисленного и частично целочисленного программирования, а также для решения задач комбинаторного типа.

Введем предварительно следующие определения.

Пусть $M \subseteq \mathbb{Z}$ есть дискретное множество, а $f(x)$ - произвольная действительная функция, которая определена на множестве M . Тогда точка $x \in M$ является точкой локального минимума функции $f(x)$ относительно окрестности радиуса r , если для любой точки y из шара $y \in U_M(x, r)$ выполняется неравенство $f(x) \leq f(y)$ и $U_M(x, r) \setminus \{x\} \neq \emptyset$. Очевидно, что если точка $x \in M$ не является точкой локального минимума функции $f(x)$ относительно шара радиуса $r_1 > 0$, то она не будет таковой и относительно окрестности любого радиуса r_2 , где $r_1 \leq r_2$. Приближенные методы решения целочисленного программирования состоят обычно в поиске точки локального минимума функции $f(x)$, заданной на множестве M , относительно некоторой окрестности радиуса $r > 0$. Будем называть векторную функцию $\Delta_M^r(x)$, определенную на дискретном множестве M , вектором спада функции $f(x)$, относительно окрестности (шара) радиуса $r > 0$, если выполняются следующие условия.

1. Значение функции $\Delta_M^r(x)$ в каждой точке $x \in M$ является l -мерным действительным вектором ($l = l(x, r)$).
2. Точка $x \in M$ является точкой локального минимума функции $f(x)$ тогда и только тогда, когда все компоненты вектора $\Delta_M^r(x)$, удовлетворяют условиям $\Delta_i(x) \geq 0, i = 1:l$.
3. Если точка $x \in M$ не является точкой локального минимума функции $f(x)$ относительно окрестности $U_M(x, r)$, то с помощью вектора спада можно определить точку $x^* \in U_M(x, r)$ такую, что $f(x^*) \leq f(x)$.

То есть, в соответствии с определением вектор $\Delta_M^r(x)$ позволяет для каждой точки $x \in M$ найти направление уменьшения (спада) значений функции $f(x)$ в шаре (окрестности) $U_M(x, r)$.

Рассмотрим общую схему метода вектора спада.

1. Исходя из особенностей решаемой задачи, выбираем некоторое начальное приближение $x^{(0)} \in M$ и задаем величину радиуса $r > 0$.

2. Задаем некоторую последовательность радиусов $\{r_1, r_2, \dots, r_n\}$, удовлетворяющую соотношениям $0 < r_1 < r_2 < \dots < r_n = r; m \geq 1$.

3. Полагаем $h = 0$.

4. На каждом $(h + 1)$ -м шаге алгоритма выполняем следующие действия.

4.1. Полагаем $k = 1$.

4.2. Рассматриваем окрестность $U_M(x^{(h)}, r_k)$.

4.3. По значениям компонент вектора спада $\Delta_M^{r_k}(x^{(h)})$ определяем, является ли $f(x^{(h)})$ локальным минимумом функции $f(x)$ в шаре (окрестности) $U_M(x^{(h)}, r_k)$. Если да, то при $k < m$, заменив $k \leftarrow k + 1$, переходим к пункту 4.2. При значении $k = m$, переходим к п.5. В противном случае, если значение функции $f(x^{(h)})$ не является локальным минимумом, то переходим к п.4.4.

4.4. По значениям компонент вектора спада $\Delta_M^{r_k}(x^{(h)})$ находим в окрестности $U_M(x^{(h)}, r_k)$ точку $x^{(h+1)}$, для которой $f(x^{(h+1)}) < f(x^{(h)})$. Заменяем $h \leftarrow h + 1$ и переходим к п.4.

5. Заканчиваем работу алгоритма, так как точка $x^{(h)}$ является искомой точкой локального минимума функции $f(x)$ относительно окрестности радиуса $r > 0$.

Теорема (достаточные условия сходимости метода вектора спада) /Сергиенко с101/.

Если функция $f(x)$, определенная на дискретном множестве M , удовлетворяет следующим условиям:

1. Функция $f(x)$ ограничена снизу на M , то есть $f(x) \geq c$, где $c = \text{const}$ для всех точек $x \in M$;
2. Имеет место неравенство $|f(x^1) - f(x^2)| \geq \delta$, где $\delta = \text{const}$ для любых точек x^1, x^2 из множества M таких, что $f(x^1) \neq f(x^2)$. Тогда, при любом выборе начального приближения $x^{(0)} \in M$, величины радиуса $r > 0$ и последовательности $\{r_1, r_2, \dots, r_n\}$, вычислительный процесс по методу вектора спада сходится за конечное число шагов, не превышающее величины $\frac{f(x^{(0)}) - c}{\delta}$.

Вычисления вектора спада предполагает определение направления из центральной точки окрестности $U_M(x, r)$ в точку наименьшего значения. Эту процедуру можно осуществить, как методом аппроксимации поверхности функции $f(x)$, по вычисленным значениям в некоторых точках, принадлежащих заданной области (шару), так и по численной аппроксимации градиента в центральной точке.

Проблема поиска глобального экстремума.

Задачу глобальной оптимизации можно записать в следующем виде:

$$F(x) \rightarrow \min_{x \in X},$$

где многоэкстремальная функция цели $F(x)$ является невыпуклой и обладающей необходимой степенью гладкости; $x \in X \subset \mathbb{R}^n$.

Назовем множество $X_{opt} = \arg \min_{x \in X} F(x)$ - множеством оптимальных решений, а множество:

$$X_{Lopt} = \{x \in X, \exists \varepsilon > 0 : F(x) \leq F(\tilde{x}), \forall \tilde{x} \in B_\varepsilon(x) \cap X\},$$

где $B_\varepsilon(x) = \{\tilde{x} \in \mathbb{R}^n : |\tilde{x} - x| < \varepsilon\}$ - множеством локально-оптимальных решений.

Тогда задача глобальной оптимизации состоит в отыскании точки $x \in X_{opt}$. Причем, обычно, имеет место соотношение $X_{opt} \neq X_{loc_opt}$. В силу широты класса многоэкстремальных функций задача глобальной оптимизации, в общем случае, является неразрешимой, то есть нельзя гарантировать, что решение задачи будет получено за конечное число шагов. Поэтому часто отказываются от требования разрешимости задачи и ищут не точное решение, а некоторую его оценку. При этом необходимо иметь некоторый критерий приемлемости получаемой оценки.

Способ решения задачи глобальной оптимизации можно представить в виде итеративного процесса, который порождает последовательность точек в соответствии с предписанным набором правил, который включает и критерий останова такого процесса. Довольно часто глобальное решение находят, перебрав все локальные решения оптимизационной задачи. Однако такая задача, как правило, является достаточно трудоемкой. Поэтому, обычно, применяют другой подход, когда находят и анализируют только часть локальных решений, а затем доказывают, что оставшиеся локальные минимумы не влияют на точность решения.

Поскольку априори не известно, где именно в области X лежит точки глобального оптимума, то обычно применяют некоторую стратегию, которая позволяет искать эти точки по всему множеству X . Выбрав какую-то точку $x^{(i)}$, обычно предполагают, что в окрестности ее существует лучшее значение функции $F(x_{lok}^{(i)}) < F(x^{(i)})$. Для поиска такой точки $x_{lok}^{(i)}$ применяют метод локальной спуска или процедуру локальной техники. Важным условием нахождения алгоритмом некоторого приближения является сходимость генерируемой им последовательности точек $\{x_{lok}^{(i)}\}$ к приемлемой оценке глобального оптимального решения.

Все известные методы глобальной оптимизации можно разделить на две категории: детерминированные и стохастические. Детерминированные методы получают глобальное решение посредством организации процедуры поиска на всем допустимом множестве. Поэтому при возрастании размерности задачи резко возрастает трудоемкость таких методов. Стохастические алгоритмы позволяют в определенной степени уйти от проблем детерминированных алгоритмов. Однако применение стохастических алгоритмов не дает гарантии нахождения приемлемой оценки.

Численные методы оптимизации многоэкстремальных одномерных функций.

Рассмотрим методы отыскания глобального экстремума функций одной переменной. Пусть функция $f(x)$ удовлетворяет на отрезке $X = [a, b]$ условию Липшица:

$$|f(x_1) - f(x_2)| \leq L |x_1 - x_2|,$$

где $L > 0$ - некоторая известная постоянная, и имеет в точке $x^* \in X$ минимум, то есть

$x^* = \arg \min_{x \in X} f(x)$. Предполагается, что в общем случае, точка глобального минимума может быть

не единственной.

Рассмотрим задачу приближенного отыскания минимального значения функции $f(x^*)$

исходя из предположения, что задано либо число N вычислений функции $f(x)$, либо точность δ отыскания значения минимума.

Метод перебора.

Пусть задано число N вычислений. Зададим точки вычисления функции $f(x)$ следующим образом: $x_1 = a + \frac{(b-a)}{2N}; x_2 = a + 3\frac{(b-a)}{2N}; \dots; x_N = a + (2N-1)\frac{(b-a)}{2N}$ /Сухарев с53/.

Вычислим значения $f(x_1), f(x_2), \dots, f(x_N)$ и в качестве приближения к точке минимального значения примем точку: $\hat{x} = \arg \min_{i=1:N} f(x_i)$. Нахождение точки \hat{x} можно решить одним из

методов сортировки. Очевидно, что погрешность такого метода будет не превосходить $L \frac{(b-a)}{2N}$, где $L > 0$ - постоянная Липшица.

Метод ломанных (метод Пиявского).

Пусть $x_1 = a; x_2 = b$ и известны результаты измерений $f(x_1), f(x_2), \dots, f(x_k)$. В силу соотношения $|f(x_i) - f(x_k)| \leq L |x_i - x_k|$ получим:

$$f(x) \geq f(x_i) - L |x - x_i|, x \in X, i = 1, 2, \dots, k.$$

Тогда справедливо следующее соотношение:

$$f(x) \geq \phi_k(x) \stackrel{\text{def}}{=} \max_{i=1, \dots, k} \{f(x_i) - L |x - x_i|\}.$$

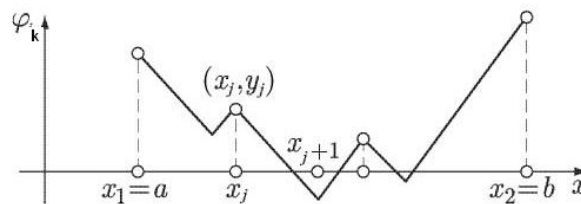
Функция $\phi_k(x)$ является точной минорантой для функций, удовлетворяющих условию

$|f(x_i) - f(x_k)| \leq L |x_i - x_k|$ и принимающих в точках x_1, x_2, \dots, x_k значения

$y_1 = f(x_1), y_2 = f(x_2), \dots, y_k = f(x_k)$. Тогда точка x_{k+1} выбирается по правилу:

$$x_{k+1} = \arg \min_{x \in X} \phi_k(x).$$

В качестве приближения \hat{x} к искомому значению x^* минимума после N вычислений принимается величина $\hat{x} = \min_{i=1, 2, \dots, N} f(x_i)$.



Погрешность N вычислений метода не превосходит величины $|\min_{i=1, 2, \dots, N} f(x_i) - \min_{x \in X} \phi_N(x)|$.

Метод покрытий.

Пусть требуется обеспечить отыскание минимума с точностью не хуже δ . Обозначим через

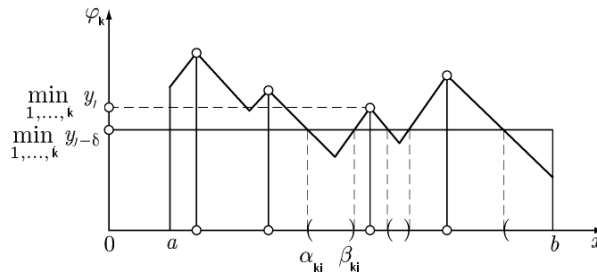
$N_0 = \frac{(b-a)}{r}$ число отрезков длины $r = \frac{2\delta}{L}$, которыми можно покрыть отрезок $X = [a, b]$. Зададим

следующие точки $x_i = \{a + (2i-1)\frac{(b-a)}{2N_0}\}, i = 1 : N_0$. Нетрудно убедиться, что отрезки длины r с

центром в точках сетки $\{x_i\}$ покрывают $X = [a, b]$. Пусть вычисления проведены в точках

x_1, x_2, \dots, x_k , то есть известны значения $y_1 = f(x_1), y_2 = f(x_2), \dots, y_k = f(x_k)$. Опишем $(k+1)$ -й шаг алгоритма. Для этого введем обозначение $X_k = \{x \in X : \phi_k(x) < \min_{j=1,2,\dots,k} (y_j) - \delta\}$, где

$\phi_k(x) \stackrel{\text{def}}{=} \max_{i=1,\dots,k} \{f(x_i) - L|x - x_i|\}$. Очевидно, что множество X_k является объединением не более $k+1$ -го промежутков. Обозначим число этих промежутков через величину m_k .



Пусть j -ый промежуток есть $(\alpha_{kj}, \beta_{kj})$, $l_{kj} = \beta_{kj} - \alpha_{kj}$. Обозначим через N_k минимальное число отрезков длины r , которыми можно покрыть множество X_k . Для определения числа N_k можно

записать следующее соотношение $N_k = \sum_{j=1}^{m_k} N_{kj}$, где $N_{kj} = \lceil \frac{l_{kj}}{r} \rceil$. Построим новую сетку с

следующими точками $x_{k+1} \in \bigcup_{j=1}^{m_k} \left\{ \alpha_{kj} + (2k-1) \frac{(\beta_{kj} - \alpha_{kj})}{2N_{kj}}; k=1, 2, \dots, N_{kj} \right\}$. Отрезки длины r с

центрами в точках новой сетки покрывают область X_k . Процесс останавливается на s -ом шаге, если $X_s = \emptyset$. При этом величина $\min_{k=1,2,\dots,s} (f(x_k))$ принимается в качестве приближения. Требуемая точность, при этом, будет обеспечиваться, так как выполняется соотношение:

$$\min_{x \in X} \phi_s(x) \leq \min_{x \in X} f(x) \leq \min_{j=1,2,\dots,s} y_j.$$

При реализации алгоритмов перебора, ломаных и покрытий требуется знать значение константы Липшица $L > 0$, которое на практике часто не известно. В качестве оценки значения L можно принять величину \hat{L}_k , вычисляемую, после k шагов, по следующей формуле:

$$\hat{L}_k = \sigma_k \max_{i1, i2 \leq k} \frac{|f(x_{i1}) - f(x_{i2})|}{|x_{i1} - x_{i2}|},$$

где $\sigma_k \geq 1$ коэффициент запаса. Однако, следует учитывать, что модифицированные таким способом алгоритмы уже не гарантируют решение задачи с заданной точностью.

Детерминированные методы многомерной глобальной оптимизации.

Одномерные методы перебора, ломаных и покрытий могут быть обобщены для задач многомерной многоэкстремальной оптимизации. Как уже отмечалось, основная трудность решения многомерных многоэкстремальных задач вызывается ростом вычислительных затрат при увеличении размерности пространства, содержащего область поиска X .

Если минимизируемая функция является липшицевой и решение задачи осуществляется по методу перебора, то с ростом размерности число узлов соответствующей сетки увеличивается экспоненциально. Задачу иногда удастся упростить за счет применения эвристического способа разбиения допустимого множества, как это делается в методе ветвей и границ (например, на

параллелепипеда), выбором вида оценок оптимального значения целевой функции на подмножестве, а также выбором стратегии измельчения разбиения.

Решение задачи поиска глобального экстремума методами покрытий или ломаных является значительно более экономным, чем перебор. Однако для этих методов возникает еще и вторая трудность: процедура построения оптимальных покрытий сложных многомерных областей является тяжелой вычислительной задачей, методы решения которой, известны лишь для некоторых случаев /Стронгин с154/. Для более простых, не оптимальных покрытий, задача существенно упрощается, но с ростом размерности увеличивается не только сложность вычислений, но и падает точность поиска. В частности, для метода Пиявского (метода ломаных) основная сложность в многомерном случае заключается в решении вспомогательных аппроксимирующих многоэкстремальных задач.

Для упрощения решения задачи многоэкстремальной многомерной оптимизации вида

$$F(x) \rightarrow \min_{x \in X},$$

где $x \in X \subset \mathbb{R}^n$, важную роль играет наличие некоторых специальных свойств минимизируемой функции $F(x)$. Эти свойства можно отнести к двум видам: свойства первого вида позволяют свести задачу оптимизации к одно- экстремальной; свойства второго вида – обеспечивают уменьшение (редукцию) размерности задачи.

Свойства первого вида обуславливает, либо наличие «глубокого», либо «широкого» глобального минимума по сравнению с локальными минимумами. В первом случае можно перейти к процедуре оптимизации некоторой «сглаженной» унимодальной функции, а во втором случае локальный спуск из некоторого небольшого количества начальных точек будет, практически всегда, приводить к области точек глобального минимума.

Свойства второго вида связывают с возможностью решения задачи оптимизации функции $F(x)$ с решением последовательности «вложенных» одномерных задач:

$$\min_{x \in X} F(x) = \min_{x_1 \in [a_1, b_1]} \dots \min_{x_n \in [a_n, b_n]} F(x_1, \dots, x_n).$$

Такая схема называется обычно многошаговой схемой редукции размерности. Такое

представление справедливо для простейшего случая, когда $F(x) = \sum_{i=1}^n f_i(x_i)$. Для других функций

необходимо введение других упрощающих допущений. Введение допущения о том, что оптимизируемая функция является липшицевой, не позволяет обойти трудности связанные с экспоненциальным возрастанием вычислительной сложности при увеличении размерности задачи, но позволяет перейти к одномерной задаче оптимизации с ограничениями в виде итерационной системы нелинейных уравнений.

Другим важным классом алгоритмов глобальной оптимизации являются траекторные методы. Известно, что отрезок $[0, 1]$ вещественной оси может быть однозначно и непрерывно отображен на гиперкуб $D \subset \mathbb{R}^n$. Отображения такого рода обычно называют кривыми Пеано. Пусть $x(t), t \in [0, 1]$ - есть кривая Пеано. Тогда из непрерывности $F(x)$ и $x(t)$, а также равенства

$$D = \{x(t) : 0 \leq t \leq 1\}, \text{ следует, что } \min_{x \in D} F(x) = \min_{t \in [0, 1]} F(x(t)).$$

Схема редукции обеспечивает построение постепенно уплотняющейся под заданную точность неравномерной сетки сразу по всей области $D \subset \mathbb{R}^n$. Для построения образа $x(t), t \in [0, 1]$, то есть кривых (траекторий)

заполняющих область оптимизации $D \subset \mathbb{R}^n$, часто используют свойства решений систем

обыкновенных дифференциальных уравнений первого или второго параметра, зависящих от параметра.

Методы случайного поиска многомерной глобальной оптимизации.

Метод прямого случайного поиска для решения многоэкстремальных задач не используют для поиска глобального оптимума каких-либо итераций локального спуска. Точка $x^{(k+1)}$ очередной итерации выбирается согласно выражению:

$$x^{(k+1)} = x^{(k)} + \xi^{(k)},$$

где $\xi^{(k)}$ - есть реализация n -ой векторной случайной величины, распределение вероятностей которой определяется типом алгоритма случайного поиска */Растрингин с77/*.

Ряд способов описания таких алгоритмов связан с моделями соответствующих вероятностных автоматов. Последовательность $\{x^k\}$ можно рассматривать как реализацию случайного процесса, которая при возрастании k должна пересекаться с заданной окрестностью точки абсолютного минимума. Очевидно, правильно построенный алгоритм должен обеспечивать достаточно высокую вероятность попадания и длительность пребывания $\{x^k\}$ в этой окрестности. Модификацией метода прямого поиска является алгоритм контролируемого случайного поиска. Алгоритм начинает работу на некотором множестве S начальных точек, равномерно распределенных на допустимом множестве задачи. В процессе работы образуются новые испытываемые точки, которые замещают подмножество худших точек из множества S . Алгоритм останавливается, когда все полученные значения функции достаточно близки.

В методе мултистарта также формируется случайным образом некоторое начальное множество стартовых точек S . Однако, последовательно, из каждой стартовой точки запускается алгоритм локального спуска и, из полученного множества локальных решений выбирается наилучшее. Чаще всего запуск локальных алгоритмов спуска осуществляют из точек, равномерно распределенных на множестве допустимого поиска. Для повышения вычислительной эффективности метода мултистарта часто осуществляют группировку точек (образование кластера) дающих близкие решения локального поиска. Таким образом, используя понятие близости локальных решений формируется некоторая средняя точка, имеющая соответствующий весовой коэффициент и координаты центра кластера. Тогда, при очередной модификации, стартового множества S с помощью метода случайного поиска, будут учитываться, как средние точки кластеров, так точки допустимой области, не вошедшие ни в один из кластеров. К концу работы алгоритма, по сути, формируются области (кластеры) точек, содержащие, различные локальные экстремумы.

Агломеративную процедуру кластеризации локальных решений глобальной оптимизации удобно описывать с помощью графов близости. Такой граф представляет собой направленный нагруженный граф (топографический граф), в вершинах которого хранится информация о целевой функции в каждой точке множества S , а дуги помечены мерой близости между точками (кластерами). Использование описания процесса поиска глобального оптимума с кластеризацией облегчает реализацию алгоритма, как на последовательных машинах, так и на машинах с параллельной архитектурой.

Эволюционные алгоритмы оптимизации.

Эволюционные алгоритмы – это алгоритмы, разработанные в качестве методов решения оптимизационных задач и основанные на принципах природной эволюции. Такие алгоритмы обладают достаточно слабой сходимостью к глобальному решению, но хорошо обрабатывают

сильно зашумленные функции с большим числом локальных решений, не «залипая» на них, и находя действительно глобальный оптимум.

Различают различные алгоритмы: генетические алгоритмы, эволюционное программирование, эволюционные стратегии, генетическое программирование. Все они моделируют базовые положения в теории биологической эволюции — процессы отбора, мутации и воспроизводства. Поведение агентов (локальных решений) определяется окружающей средой. Множество агентов принято называть популяцией. Такая популяция эволюционирует в соответствии с правилами отбора в соответствии с целевой функцией, задаваемой окружающей средой (правилами алгоритма). Таким образом, каждому агенту (локальному решению) популяции назначается значение его пригодности в окружающей среде. Размножаются только наиболее пригодные виды. Рекомбинация и мутация позволяют изменяться агентам и приспособляться к среде.

Рассмотрим основные различия между эволюционными алгоритмами и генетическими алгоритмами.

Первое различие заключается в способе представления особей. Эволюционные алгоритмы оперируют векторами действительных чисел, тогда как генетические алгоритмы — двоичными векторами.

Второе различие между генетическими и эволюционными алгоритмами кроется в организации процесса селекции. При реализации эволюционного алгоритма формируется промежуточная популяция, состоящая из всех родителей и некоторого количества потомков, созданных в результате применения генетических операторов. С помощью селекции размер этой промежуточной популяции уменьшается до величины родительской популяции за счет исключения наименее приспособленных особей. Сформированная таким образом популяция образует очередное поколение. Напротив, в генетических алгоритмах предполагается, что в результате селекции из популяции родителей выбирается количество особей, равное размерности исходной популяции, при этом некоторые (наиболее приспособленные) особи могут выбираться многократно. При реализации эволюционного алгоритма особи выбираются без повторений. В эволюционных алгоритмах применяется детерминированная процедура селекции, тогда как в генетических алгоритмах она имеет случайный характер.

Третье различие между генетическими и эволюционными алгоритмами касается последовательности выполнения процедур селекции и рекомбинации (т.е. изменения генов в результате применения генетических операторов). При реализации эволюционного алгоритма вначале производится рекомбинация, а потом селекция. В случае выполнения генетических алгоритмов эта последовательность инвертируется. При применении эволюционных алгоритмов потомок образуется в результате скрещивания двух родителей и мутации. При выполнении генетических алгоритмов вначале производится селекция, приводящая к образованию переходной популяции, после чего выполняются генетические операторы скрещивания и мутации. *Четвертое различие* между генетическими и эволюционными алгоритмами заключается в том, что параметры генетических алгоритмов (такие, как вероятности скрещивания и мутации) остаются постоянными на протяжении всего процесса эволюции, тогда как при реализации эволюционного алгоритма эти параметры подвергаются непрерывным изменениям (так называемая самоадаптация параметров).

По мере развития генетических и эволюционных алгоритмов в течение последних лет существенные различия между ними постепенно уменьшаются. Например, в настоящее время при реализации генетических алгоритмов для решения оптимизационных задач все чаще применяется представление хромосом действительными числами и различные модификации генетических операторов, что имеет целью повысить эффективность этих алгоритмов.

Программные пакеты системы Matlab для решения оптимизационных задач.

В данной системе предусмотрено три пакета Optimization Toolbox, Genetic Algorithms and Direct Search Toolbox и Global Optimization Toolbox.

Пакет оптимизации (Optimization) – это библиотека функций расширяющая возможности системы MatLab для решения задач оптимизации малой, средней и большой размерности. Основным алгоритмом для решения задач без оптимизации является симплексный метод Нелдера - Мида и квазиньютоновские методы. Для решения задач с ограничениями используются алгоритмы квадратичного программирования, а также алгоритмы Ньютона –Рафсона, Левенберга - Маркарда и прочие. Для решения задач целочисленной оптимизации с линейными ограничениями используется метод ветвей и границ.

Пакет Global Optimization Toolbox позволяет быстро формулировать оптимизационную модель в аналитической среде Maple и затем, используя соответствующие численные алгоритмы глобальной оптимизации, находить решение поставленной задачи.

Пакет Genetic Algorithms and Direct Search Toolbox обеспечивает поиск точки глобального минимума с использованием гибко настраиваемого генетического алгоритма, а также прямой поиск точки оптимизации по заданному образцу.