Лабораторная работа №4:

Условная оптимизация

Дисциплина: Методы оптимизации

Цель работы: Изучение и практическое освоение методов условной оптимизации для решения задач нелинейного программирования с ограничениями.

Теоретическая часть

Введение в условную оптимизацию

Задачи условной оптимизации возникают, когда необходимо найти минимум или максимум функции при наличии ограничений на ее переменные. Математическая постановка задачи в общем виде формулируется как:

$\min_{x\in\mathbb{R}^n} f(x)$

при условиях:

$$\begin{aligned} &g_i(x) \leq 0, \quad i = 1, \ldots, m \\ &h_j(x) = 0, \quad j = 1, \ldots, k \\ &x_l \leq x \leq x_u \end{aligned}$$

где $f(x)$ - целевая функция, $g_i(x)$ - функции ограничений-неравенств, $h_j(x)$ - функции ограничений-равенств, x_l и x_u - нижние и верхние границы переменных.

Основные понятия

Допустимое множество - множество точек, удовлетворяющих всем ограничениям: $\Omega = \{x \in \mathbb{R}^n : g_i(x) \leq 0, h_j(x) = 0, x_l \leq x \leq x_u\}$

Активные ограничения - ограничения, которые выполняются как равенства в точке минимума. Различают:

- Активные ограничения-неравенства: $g_i(x^*) = 0$
- Активные ограничения-равенства: $h_j(x^*) = 0$

Локальный минимум - точка $x^* \in \Omega$ называется точкой локального минимума, если существует окрестность U этой точки такая, что $f(x^*) \leq f(x)$ для всех $x \in U \cap \Omega$.

Глобальный минимум - точка $x^* \in \Omega$ называется точкой глобального минимума, если $f(x^*) \leq f(x)$ для всех $x \in \Omega$.

Условия оптимальности Куна-Таккера

Для задачи нелинейной оптимизации с ограничениями-неравенствами:

$\min f(x) \quad \text{при условиях} \quad g_i(x) \leq 0, \quad i = 1, \ldots, m$

определим функцию Лагранжа:

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

где $\lambda_i \geq 0$ - множители Лагранжа.

Теорема Куна-Таккера (необходимые условия оптимальности):
Если x^* - точка локального минимума и выполнены условия регулярности (например, условие линейной независимости градиентов активных ограничений), то существуют множители Лагранжа $\lambda^* = (\lambda_1^*, \ldots, \lambda_m^*)$ такие, что:

1. **Стационарность**:
   $\nabla_x L(x^*, \lambda^*) = \nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) = 0$
2. **Допустимость**:
   $g_i(x^*) \leq 0, \quad i = 1, \ldots, m$
3. **Дополняющая нежесткость**:
   $\lambda_i^* g_i(x^*) = 0, \quad i = 1, \ldots, m$
4. **Неотрицательность**:
   $\lambda_i^* \geq 0, \quad i = 1, \ldots, m$

Для задач с ограничениями-равенствами и неравенствами условия дополняются соответствующими условиями для ограничений-равенств.

Методы условной оптимизации

1. Метод доверительной области (trust-constr)

Алгоритм метода:

1. **Инициализация**: x_0, Δ_0
2. **Аппроксимация**: построение квадратичной модели целевой функции и линейной модели ограничений
3. **Решение подзадачи**:
   $\min_{p \in \mathbb{R}^n} m_k(p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T B_k p$
   при условиях: $g_i(x_k) + \nabla g_i(x_k)^T p \leq 0$, $\|p\| \leq \Delta_k$
4. **Обновление доверительной области** на основе соотношения фактического и предсказанного уменьшения:
   $\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)}$

Параметры радиуса доверительной области:

- Δ_0 - начальный радиус доверительной области
- Δ_{max} - максимальный радиус доверительной области
- Δ_{min} - минимальный радиус доверительной области
- η_1, η_2 - пороговые значения для принятия шага ($0 < \eta_1 < \eta_2 < 1$)
- γ_1, γ_2 - коэффициенты уменьшения/увеличения радиуса ($0 < \gamma_1 < 1 < \gamma_2$)

Критерии сходимости:

- ϵ_g - допуск на норму градиента: $\|\nabla L(x^*, \lambda^*)\| \leq \epsilon_g$
- ϵ_c - допуск на выполнение ограничений: $\|c(x^*)\| \leq \epsilon_c$
- ϵ_x - допуск на изменение переменных: $\|x_{k+1} - x_k\| \leq \epsilon_x$

Параметры подзадачи:

- B_k - аппроксимация матрицы Гессе лагранжиана
- $\rho_k = \frac{f(x_k)-f(x_k+p_k)}{m_k(0)-m_k(p_k)}$ - показатель качества шага

В методе доверительной области **p** - это вектор шага, определяемый как решение подзадачи минимизации в пределах доверительной области:

$p_k = \arg\min_{p\in\mathbb{R}^n} m_k(p)$

где:

- $p \in \mathbb{R}^n$ - вектор шага из текущей точки x_k
- $m_k(p)$ - локальная квадратичная модель целевой функции

Свойства параметра p

1. Ограничение нормы:
$\|p\| \le \Delta_k$
где Δ_k - текущий радиус доверительной области

2. Структура квадратичной модели:
$m_k(p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T B_k p$

3. Условия оптимальности для p:
Для найденного p_k должны выполняться условия Каруша-Куна-Таккера:

$$\begin{aligned}(B_k + \lambda I)p_k + \nabla f(x_k) + A_k^T \mu = 0 \\ \mu_i(g_i(x_k) + \nabla g_i(x_k)^T p_k) = 0 \\ \mu_i \ge 0 \\ \|p_k\| \le \Delta_k\end{aligned}$$

Вычислительные аспекты

Определение длины шага:

- Если $\|p_k\| < \Delta_k$, то ограничение неактивно
- Если $\|p_k\| = \Delta_k$, то шаг достигает границы доверительной области

Связь с фактическим шагом:
Новая точка вычисляется как:
$x_{k+1} = x_k + p_k$

Критерий принятия шага:
$\rho_k = \frac{\text{actual reduction}}{\text{predicted reduction}} = \frac{f(x_k)-f(x_k+p_k)}{m_k(0)-m_k(p_k)}$

Геометрическая интерпретация

- **p** - направление и длина шага внутри доверительной области
- **p** балансирует между направлением наискорейшего спуска и учетом кривизны функции
- При малых Δ_k: p приближается к направлению наискорейшего спуска
- При больших Δ_k: p приближается к шагу Ньютона

Функция scipy.optimize.minimize()

Ключевые параметры:

- `gtol` - допуск на градиент (обычно 1e-8)
- `barrier_tol` - точность для барьерных методов
- `initial_tr_radius` - начальный радиус доверительной области
- `max_tr_radius` - максимальный радиус доверительной области
- `eta` - порог принятия шага (0 < η < 1)
- `initial_constr_penalty` - начальный коэффициент штрафа

Специфические параметры:

- `factorization_method` - метод факторизации матриц
- `hess_update_strategy` - стратегия обновления гессиана
- `subproblem_algorithm` - алгоритм решения подзадачи
- `verbose` - уровень детализации вывода (0-3)

2. Последовательное квадратичное программирование (SLSQP)

На каждой итерации решается квадратичная подзадача:

$$\begin{aligned}&\min_{d\in\mathbb{R}^n}\frac{1}{2}d^TH_kd+\nabla f(x_k)^Td\\&\text{при условиях:}\\&\nabla g_i(x_k)^Td+g_i(x_k)\leq 0\\&\nabla h_j(x_k)^Td+h_j(x_k)=0\end{aligned}$$

где H_k - аппроксимация матрицы Гессе лагранжиана.
Параметры квадратичной подзадачи:

- H_k - аппроксимация $\nabla^2_{xx}L(x_k,\lambda_k,\mu_k)$
- ϵ_H - параметр регуляризации гессиана
- τ - параметр для обеспечения положительной определенности

Критерии остановки:

- Первый порядок оптимальности: $\left\|\begin{bmatrix}\nabla f(x)+A(x)^T\lambda\\c(x)\end{bmatrix}\right\|\leq\epsilon_{opt}$
- Допуск дополняющей нежесткости: $\|\min(\lambda,-c(x))\|\leq\epsilon_{comp}$

Параметры обновления:

- α_k - длина шага, определяемая условием Вольфа или Армихо
- β - параметр уменьшения шага ($0<\beta<1$)

Параметр d

Математическое определение

В методе SLSQP **d** - это вектор направления поиска, определяемый как решение квадратичной подзадачи программирования:

$d_k = \arg\min_{d\in\mathbb{R}^n} q_k(d)$

где квадратичная подзадача имеет вид:

$$\begin{aligned}&\min_{d\in\mathbb{R}^n} \frac{1}{2} d^T H_k d + \nabla f(x_k)^T d\\&\nabla g_i(x_k)^T d + g_i(x_k) \leq 0 \quad (\text{неравенства})\\&\nabla h_j(x_k)^T d + h_j(x_k) = 0 \quad (\text{равенства})\end{aligned}$$

Свойства параметра d

1. Локальная аппроксимация:

- **d** представляет направление, которое минимизирует квадратичную аппроксимацию целевой функции
- Одновременно удовлетворяет линейным аппроксимациям ограничений

2. Условия оптимальности:
Для оптимального d_k выполняются условия ККТ:

$$\begin{aligned}H_k d_k + \nabla f(x_k) + \sum \lambda_i \nabla g_i(x_k) + \sum \mu_j \nabla h_j(x_k) = 0\\\lambda_i(\nabla g_i(x_k)^T d_k + g_i(x_k)) = 0\\\lambda_i \geq 0\\\nabla h_j(x_k)^T d_k + h_j(x_k) = 0\end{aligned}$$

Структура и компоненты

Составляющие направления:

- **Тангентальная компонента** - движение вдоль допустимого множества
- **Нормальная компонента** - движение toward допустимого множества

Связь с полным шагом:
Фактическое обновление переменных:
$x_{k+1} = x_k + \alpha_k d_k$
где α_k - длина шага, определяемая линейным поиском.

Вычислительные аспекты

Определение допустимости:

- Если $d_k = 0$ и условия ККТ выполнены, достигнута стационарная точка
- Направление **d** обеспечивает одновременное улучшение целевой функции и соблюдение ограничений

Свойства матрицы Гессе:

- H_k - положительно определенная аппроксимация гессиана лагранжиана
- Обеспечивает выпуклость квадратичной подзадачи

Геометрическая интерпретация

- **d** - направление, которое ведет к допустимой точке с улучшенным значением функции
- В окрестности решения **d** стремится к нулю
- Направление учитывает как градиент целевой функции, так и активные ограничения

Функция scipy.optimize.minimize()

Основные параметры:

- `ftol` - допуск по изменению функции (обычно 1e-6)
- `eps` - размер шага для численного дифференцирования
- `maxiter` - максимальное число итераций
- `disp` - флаг вывода результатов (0/1)

Параметры точности:

- `tol` - общая точность решения
- `iprint` - уровень вывода итерационной информации
- `finite_diff_rel_step` - относительный шаг для конечных разностей

3. Метод штрафных функций

Преобразование задачи условной оптимизации в безусловную:
$P(x) = f(x) + \mu \left[\sum_{i=1}^{m} \max(0, g_i(x))^2 + \sum_{j=1}^{k} h_j(x)^2\right]$

Параметры штрафной функции:

- μ - коэффициент штрафа в функции:
  $P(x, \mu) = f(x) + \mu \cdot \Phi(x)$
- $\Phi(x) = \sum_{i=1}^{m} [\max(0, g_i(x))]^p + \sum_{j=1}^{l} |h_j(x)|^p$

Стратегия увеличения штрафа:

- $\mu_{k+1} = \gamma \cdot \mu_k$, где $\gamma > 1$
- Критерий увеличения: $\|\nabla\Phi(x_k)\| > \epsilon_\mu$

Параметры точности:

- ϵ_μ - точность для штрафного параметра
- ϵ_P - допуск для штрафной функции: $\|\nabla P(x, \mu)\| \leq \epsilon_P$

Функция scipy.optimize.minimize()

Параметры штрафов:

- `mu` - коэффициент штрафа (начальное значение)
- `mu_update` - стратегия обновления коэффициента штрафа
- `penalty_type` - тип штрафной функции (квадратичная, точная и т.д.)

Параметры сходимости:

- `penalty_tol` - точность для штрафной функции
- `max_penalty_iter` - максимальное число итераций с увеличением штрафа
- `initial_penalty_param` - начальный параметр штрафа

4. Общие параметры для всех методов

Критерии остановки:

- `tol` - общий допуск сходимости
- `maxiter` - максимальное число итераций
- `disp` - вывод информации о процессе

Параметры производных:

- `jac` - метод вычисления якобиана {'2-point', '3-point', cs}
- `hess` - метод вычисления гессиана
- `finite_diff_rel_step` - относительный шаг для конечных разностей

Дополнительно:

- `callback` - функция обратного вызова
- `options` - дополнительные специфические параметры

Практическая часть

Задача 1: Минимизация функции Розенброка с ограничениями

Постановка задачи: Найти минимум функции Розенброка от двух переменных при наличии ограничений.

Функция Розенброка:
$f(x) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$

Ограничения:

- $0 \leq x_0 \leq 1$
- $-0.5 \leq x_1 \leq 2.0$
- $x_0 + 2x_1 \leq 1$
- $2x_0 + x_1 = 1$
- $x_0^2 + x_1 \leq 1$
- $x_0^2 - x_1 \leq 1$

Используемые методы

Метод доверительной области (trust-constr)

Теория метода:
Метод trust-constr основан на последовательном решении приближенных задач в доверительной области. На каждой итерации:

1. **Построение модельной задачи**: Целевая функция аппроксимируется квадратичной формой, а ограничения - линейными функциями:
   $m_k(p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T B_k p$
   где B_k - аппроксимация матрицы Гессе.
2. **Решение подзадачи**: Находится шаг p_k, минимизирующий $m_k(p)$ при условиях:
   $g_i(x_k) + \nabla g_i(x_k)^T p \leq 0$
   $\|p\| \leq \Delta_k$
3. **Обновление доверительной области**: Вычисляется отношение фактического уменьшения к предсказанному:
   $\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)}$
   На основе ρ_k корректируется размер доверительной области Δ_k.

Преимущества для данной задачи:

- Эффективно работает с нелинейными ограничениями
- Использует информацию о вторых производных (матрица Гессе)
- Устойчив к плохой обусловленности

Особенности реализации

Градиент и матрица Гессе функции Розенброка:

$$\nabla f(x) = \begin{bmatrix} -400x_0(x_1 - x_0^2) - 2(1 - x_0) \\ 200(x_1 - x_0^2) \end{bmatrix}$$

$$\nabla^2 f(x) = \begin{bmatrix} 1200x_0^2 - 400x_1 + 2 & -400x_0 \\ -400x_0 & 200 \end{bmatrix}$$