ИССЛЕДОВАНИЕ МЕТОДОВ
БЕЗУСЛОВНОЙ КОНЕЧНОМЕРНОЙ ОПТИМИЗАЦИИ

Лабораторная работа № 2

МЕТОДЫ ПРЯМОГО ПОИСКА ЭКСТРЕМУМА ДЛЯ ФУНКЦИЙ МНОГИХ ПЕРЕМЕННЫХ

Целью работы являются изучение и моделирование в среде MATLAB различных методов прямого поиска минимума и сравнение эффективности их применения для конкретных целевых функций нескольких переменных.

1. Краткие теоретические сведения

1.1. Постановка задачи конечномерной безусловной оптимизации

Пусть имеется некоторый арифметический вектор $x = [x_1, \ldots, x_n]$ и некоторая функция $f(x)$, называемая целевой функцией и отражающая качество решения той или иной прикладной задачи. В силу эквивалентности двух типов оптимизационных задач (максимизации и минимизации) далее рассматривается задача конечномерной минимизации. Задача поиска минимума целевой функции формулируется в виде:

$$x^* = arg\min f(x),\ x \in X,$$

где X – множество допустимых решений, среди которых ищется точка x^*, дающая минимум $f(x)$ целевой функции.

Другая распространенная запись задачи минимизации:

$$f(x) \to \min_{x \in X}.$$

Когда $X=R^n$, где R^n – n-мерное евклидово пространство вещественных чисел, то говорят о конечномерной безусловной задаче минимизации, т.е. целевая функция $f(x)$ имеет только несколько (в данном случае n) аргументов, и множество допустимых решений X есть все пространство R^n.

Ниже приводится краткое описание алгоритмов методов прямого поиска минимума для функции n переменных.

1.2. Метод покоординатного спуска (метода Гаусса-Зайделя)

Идея метода заключается в последовательном поиске точки минимума $x^* \in R^n$ целевой функции $f(x)$ вдоль каждой координаты x_i, $i=1,\dots,n$. Соответствующий алгоритм содержит следующие шаги.

1. Положить $k = 1$, $i=1$, задать точность вычислений ε и выбрать точку начального приближения

$$x^{(0)} = [x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}] \in R^n .$$

2. На итерации с номером k для точки $x^{(k)}$ произвести поиск минимума вдоль направления координаты x_i и найти соответствующую точку $x^{(i,k)}$. Это означает, что функция многих переменных становится как бы функцией одной переменной $f(x_i) = f(x_1^{(k)}, \dots, x_i, \dots, x_n^{(k-1)})$, здесь координаты $x_j^{(k)}$; $j=\overline{1,i-1}$, и $x_j^{(k-1)}$; $j=\overline{i+1,k}$, определены на настоящей, k-й, и предыдущей, (k-1)-й, итерациях соответственно. Точка $x^{(i,k)}$, которая обеспечивает минимум функции $f(x)$ в данном i-м направлении, находится с помощью любого из известных методов одномерной оптимизации (см. описание лабораторной работы № 1 [7]).

3. Если выполняются условия окончания поиска $\left\| x^{(k+1)} - x^{(k)} \right\| \le \varepsilon$, то осуществляется останов алгоритма и $x^{(i,k)} \approx x^*$. В противном случае

проверяется: если $i<n$, то переход к шагу 2, положив при этом $i=i+1$; иначе также переход к шагу 2, но $i=1$, $k=k+1$.

Таким образом, вышеописанная процедура выполняется по всем координатам.

1.3. Симплексный метод

1. Выбрать базовую точку x^0. Задать масштабный множитель α. Вычислить

$$\delta_1 = \left[\frac{(n+1)^{1/2} + n - 1}{n\sqrt{2}} \right] \alpha;$$

$$\delta_2 = \left[\frac{(n+1)^{1/2} - 1}{n\sqrt{2}} \right] \alpha .$$

Определить остальные вершины симплекса:

$$x^i = \begin{cases} x_j^o + \delta_1, & если \quad j \neq i, \\ x_j^o + \delta_2, & если \quad j = i. \end{cases} \quad i, j = 1, 2, \ldots, n.$$

2. На k-й итерации определить $x_c = \frac{1}{n} \sum_{\substack{i=1 \\ j \neq 1}}^{n+1} x^i$, где x^j - вершина с наибольшим значением функции. Отразить x^j относительно x_c:

$$x = 2x_c - x^j.$$

3. Проверка условий окончания поиска. Если условие сходимости выполнено, то останов алгоритма. Если не выполнено условие сходимости или некоторая вершина не исключается на протяжении более чем $M = 1{,}65n + 0{,}05n^2$ итераций, то необходимо уменьшить размеры симплекса, построить новый симплекс, выбрав в качестве

базовой точку, которой соответствует минимальное значение целевой функции, и перейти к шагу 2.

1.4. Метод Нелдера–Мида

1. Задать точность вычислений ε и параметры α, β, γ. Выбрать аргументы целевой функции $x_1, \ldots, x_{n+1}$ и вычислить ее значения в этих точках $f_1, \ldots, f_{n+1}$.
2. Найти x_h, x_g, x_z, f_h, f_g, f_z; f_h - наибольшее, f_g - следующее за ним, f_z - наименьшее значение функции.
3. Найти $x_0 = \dfrac{1}{n}\sum_{i \neq h} x_i$.
4. Найти x_r, f_r, отразив точку x_h относительно x_0:

$$x_r = (1 + \alpha)\, x_0 - \alpha x_h.$$

5. Если $f_r < f_z$, то производится растяжение симплекса, и находятся

$$x_e = \gamma x_r + (t - \gamma)\, x_0, f_e = f(x_e).$$

6. Если $f_e < f_z$, то $x_h = x_e$.

Проверка на сходимость осуществляется по следующему принципу. Необходимо вычислять оценку дисперсии значений целевой функции

$$\sigma^2 = \sum_{i=1}^{n+1} (f_i - \bar{f})^2 / (n-1), \text{ где } \bar{f} = \frac{\sum f_i}{n+1}.$$

и сравнивать с заданной точностью ε. Если σ < ε, то сходимость достигнута, и производится останов алгоритма. В противном случае переход к шагу 2.

7. Если $f_e \geq f_z$, то $x_h = x_r$.

Проверка на сходимость. Если сходимость достигнута, то останов алгоритма. В противном случае перейти к шагу 3.

8. Если $f_r > f_z$, но $f_r \leq f_g$, то $x_h = x_r$.

Проверка на сходимость. Если сходимость не достигнута, то возвратиться к шагу 2.

9. Если $f_r > f_z$ и $f_r > f_g$, то перейти к шагу 10.

10. Если $f_r > f_h$, то перейти к шагу 11.

Если $f_r < f_h$, то $x_h := x_r$ и $f_h = f_r$. Перейти к шагу 11.

11. Так как $f_r > f_h$, то производится переход к шагу сжатия:

$$x_c = \beta\, x_h + (1 - \beta)x_0.$$

12. Если $f_c < f_h$, то $x_h = x_c$, и если сходимость не достигнута, то возвратиться к шагу 2.

13. Если $f_c > f_h$, то перейти к шагу 14.

14. Уменьшить размерность симплекса $x_i = (x_i + x_z)/2$. Вычислить f_i для i=1,…, n+1. Проверить на сходимость. Если условие сходимости не выполняется, то возвратиться к шагу 3.

1.5. Метод Хука–Дживса

Условные обозначения:

x^K - текущая базовая точка;

x^{K-1} - предыдущая базовая точка;

x_p^{K+1} - точка, построенная при движении по образцу;

x^{K+1} - следующая (новая) базовая точка.

1. Задаются начальная точка $x^{(0)}$, приращение Δ_i, $i = 1,\ldots, n,$ а также коэффициент уменьшения шага $\alpha > 1$ и параметр окончания поиска ε.

2. Проводится исследовательский поиск.

3. Если исследовательский поиск удачный (найдена точка с меньшим значением целевой функции), то выполнить переход к шагу 5. Иначе переход к шагу 4.

4. Проверка условий окончания поиска: если условие $\Delta_i < \varepsilon$ выполнено, то останов алгоритма. Иначе надо уменьшить приращение

$$\Delta_i = \frac{\Delta_i}{\alpha}, \quad i = \overline{1,n},$$

и перейти к шагу 2.

5. Провести поиск по образцу

$$x_p^{(K+1)} = x^K + (x^K - x^{K-1}).$$

6. Провести исследующий поиск, используя x_p^{K+1} в качестве базовой точки. Пусть x^{K+1} - полученная в результате поиска точка.

7. Если выполняется неравенство $f(x^{K+1}) < f(x^K)$, то положить $x^{K-1} = x^K$ и $x^K = x^{K+1}$. Перейти к шагу 5. В противном случае перейти к шагу 4.

2. Порядок выполнения лабораторной работы

2.1. Изучить предлагаемые методы конечномерной безусловной оптимизации, используя дополнительную литературу и конспект лекций, если необходимо.

2.2. В соответствии с вариантом задания, определенным преподавателем, в среде MATLAB составить программы, реализующие вышеописанные методы поиска (метод покоординатного спуска; симплексный метод; метод Нелдера–Мида; метод Хука–Дживса), и найти точку минимума функции $f(x)$ с заданной точностью ε, изменяемой в ходе исследования.

2.3. Оформить отчет о выполнении задания с приведением условия задачи, алгоритмов и программ указанных методов поиска, графиков зависимостей количества итераций от точности решения, таблицы результатов сравнения рассмотренных методов, заключения по результатам сравнения методов.

3. Варианты заданий

№	*Целевая функция*	*Точность* ε
1	$(x_1 + 3x_2)^2 + 3(x_3 - x_4)^2 + (x_2 - 4x_3)^4$	$10^{-3}, 10^{-5}, 10^{-10}, 10^{-15}$
2	$5(x_2 - x)^2 + (4 - x_1)^2$	$10^{-3}, 5*10^{-5}, 10*10^{-10}, 5*10^{-15}$
3	$(x_1 + 2x_2)^2 + (x_3 - x_4)^2 + 3\,(x_1 - x_4)^2$	$2*10^{-3}, 4*10^{-5}, 8*10^{-10}, 10^{-15}$
4	$(x_3 - x)^2 + (2 - x_2)^2$	$10^{-3}, 10^{-5}, 10^{-7}, 10^{-10}$
5	$(x_1 - x_2)^2 - 4(x_3 - x_4)^2 + (x_2 - 6x_3)^4 + 2(x_1 - x_4)^2$	$10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}$
6	$(x_1 - x_2)^2 + 4(x_1 - x_4)^2 + (x_2 + 3x_4)^4$	$5*10^{-3}, 2*10^{-5}, 5*10^{-10}, 2*10^{-15}$
7	$3(x_1 - x_2)^2 - 4(x_2 - x_3)^2 + (x_1 - 2x_3)^4$	$10^{-3}, 10^{-5}, 10^{-10}, 10^{-15}$
8	$(x_3 - x_2)^2 + (x_3 - x_4)^2 + (x_2 - 6x_1)^2$	$10^{-3}, 10^{-5}, 10^{-15}, 10^{-20}$
9	$(x_1 - x_2)^2 + 4(x_3 - x_4)^2 + (x_2 + 6x_4)^4 + 2(x_1 - x_3)^2$	$10^{-5}, 10^{-7}, 10^{-9}, 10^{-12}$
10	$3(x_1 - x_2)^2 + (x_3 - x_4)^2 + (x_2 + 26x_3)^4 + 4(x_1 - x_4)^2$	$10^{-3}, 10^{-5}, 10^{-7}, 10^{-9}$

4. Контрольные вопросы

1. Назовите достоинства и недостатки прямых методов поиска для функций и переменных.

2. В чем преимущество метода Хука–Дживса по сравнению с методом покоординатного спуска?

3. В каких случаях удобно использовать симплексный метод?

4. Обеспечивают ли эти методы глобальную сходимость?

5. Для решения каких задач целесообразно использовать метод Нелдера–Мида?

6. Дайте геометрическую иллюстрацию всех четырех методов оптимизации.

7. Какой из приведенных методов целесообразно использовать для оптимизации технологических процессов в условиях производства?