МЕТОД СЖИМАЮЩИХСЯ МНОГОГРАННИКОВ

Цельработы: освоить методику численной оптимизации методом сжимающихся многогранников (Нелдера-Мида).

1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1. 1. Постановка задачи многомерной оптимизации

Рассмотрим задачу поиска минимума функции n переменных:

$$f(x_1,x_2,\ldots,x_n) \ \min, \tag{1}$$

В дальнейшем более удобной будет эквивалентная постановка: минимизация функции одного векторного переменного:

$$f(x) \to \min, \quad x \in \mathbf{R}^n .$$

1.2. Метод сжимающихся многогранников (Нелдера-Мида)

В методе Нелдера-Мида текущее приближение задается в виде n+1 точки в n-мерном пространстве. Такое количество выбрано потому, что это минимальное количество точек, не лежащих на одном линейном подпространстве (на одной линии, в одной плоскости и т.д.). Эти точки лежат в вершинах многогранника, размеры и положение которого неявным образом задают величину шага и направление поиска. С каждой итерацией метода одна или несколько вершин многогранника меняют свое положение, при этом многогранник сжимается или растягивается, приближаясь к искомой точке минимума.

Алгоритм

1) **Инициализация**

Задается n+1 начальная точка $x_1, x_2, \ldots, x_{n+1}$. Примечание: здесь и далее - x_1 означает i-й n-мерный вектор. Обычно это вектора (0,0,…0), (1,0,…,0),(0,1,…,0), … , (0,0,…,1).

В этих точках вычисляются значения функции:

$$f_1 = f(x_1),\ f_2 = f(x_2),\ldots,\ f_{n+1} = f(x_{n+1}).$$

2) **Сортировка**.

Из вершин многогранника выбираем три точки: x_h с наибольшим (из выбранных) значением функции f_h, x_g со следующим по величине значением f_g и x_1 с наименьшим значением функции f_1. Целью дальнейших манипуляций будет уменьшение по крайней мере f_h.

3) **Отражение**

Найдём центр тяжести всех точек, за исключением самой худшей из них, x_h:

$$x_c = \frac{1}{n} \sum_{i=1, i \neq h}^{n+1} x_i \ .$$

Вычислять в этой точке целевую функцию не нужно.

Отразим точку x_h относительно x_c с коэффициентом α (при $\alpha = 1$ это будет центральная симметрия), получим точку x_r и вычислим в ней функцию: $f_r = f(x_r)$. Координаты новой точки вычисляются по формуле $x_r = x_c - \alpha(x_h - x_c)$. Предлагается использовать $\alpha = 1$.

4) Далее сравниваем значение f_r со значениями f_h, f_g, f_1:

4.1)Если $f_r < f_1$, то есть если отраженная точка лучше всех имеющихся, то производим **растяжение**. Для этого строится новая точка $x_e = x_c + \beta(x_r - x_c)$ и в ней вычисляется значение функции $f_e = f(x_e)$, чтобы оценить успешность растяжения. Предлагает использовать $\beta = 2$

Если $f_e < f_1$, то есть растяжение удачно, то заменяем точку x_h на x_e и заканчиваем итерацию (на шаг 6).

Иначе (растяжение ухудшило результат) заменяем точку x_h на x_r и заканчиваем итерацию (на шаг 6).

4.2) Иначе, если $f_1 \leq f_r < f_g$, то заменяем точку x_h на x_r и переходим на шаг 6.

4.3) Иначе, если $f_g \leq f_r < f_h$, то меняем обозначения $x_r \leftrightarrow x_h, f_r \leftrightarrow f_h$ местами и переходим на шаг 5.

4.4). Иначе (если $f_r \geq f_h$, то есть когда отраженная точка хуже всех остальных), то переходим на шаг 5.

5) **Сжатие**.

К этому шагу алгоритм приходит, когда ни одна попытка сделать шаг не привела к успеху. Возможно, это означает, что шаг слишком велик, и нужно его уменьшить. Для этого строим точку $x_s = x_c + \gamma(x_h - x_c)$ и вычисляем в ней значение f_s. Предлагается использовать $\gamma = 1/2$. После этого оцениваем эффективность сжатия

5.1). Если $f_s < f_h$, то есть если есть улучшение, то заменяем точку x_h на x_s и переходим на шаг 6.

5.2). Иначе, если $f_s \geq f_h$, то производим **глобальное сжатие** — пропорциональное уменьшение так, чтобы точка с наименьшим значением функции x_1 осталась на своем месте, а остальные приблизились бы к ней вдвое.

Новые точки вычисляются по формуле: $x_i \to x_i + (x_i - x_1)/2$ для всех точек, кроме x_1 (которая не меняется). Для каждой новой точки вычисляется значение целевой функции.

6). Последний шаг — **оценка погрешности**. Оценка может выполняться по разному. В данной работе предлагается в качестве погрешности использовать максимальное из попарных расстояний между точками:

$$r = \max_{\substack{i=1..n+1,\\ j=1..n+1,\\ i\neq j}} d(x_i, x_j),$$

где d(a,b) – расстояние между точками.

Если оценка погрешности превышает заданную точность, то осуществляется возврат на шаг 2, иначе алгоритм завершается.

2. ЗАДАНИЕ ПО РАБОТЕ

В лабораторной работе требуется на любом языке программирования реализовать решение заданной 2-мерной оптимизационной задачи, используя метод сжимающихся многогранников.

Для выполнения элементарных операций над векторами необходимо либо воспользоваться существующими библиотеками, доступными для выбранного языка (например, numpy для языка python), либо написать собственные подпрограммы сложения и умножения вектора на число.

В результате выполнения программа должна вывести таблицу из четырех столбцов:

- номер итерации i,
- текущее наилучшее приближение x_1 (два столбца)
- оценку погрешности r_i.

3. СОДЕРЖАНИЕ ОТЧЕТА

Отчет по лабораторной работе должен содержать следующие элементы:

1. Постановка задания.
2. Данные варианта.
3. Аналитическое решение оптимизационной задачи.
4. Текст программы.
5. Результаты выполнения программы, оформленные в виде таблицы из 4 столбцов: итерация, текущее наилучшее приближение (две координаты), погрешность. Если число шагов велико, привести в отчете только последние

15 шагов.

6. Выводы.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1) Оцените скорость сходимости (линейная, сверхлинейная, квадратичная) по полученным результатам работы программы.

2) Сколько раз, максимально и минимально, вычисляется целевая функция за одну итерацию алгоритма?

3) Как изменится эффективность алгоритма, если взять очень малое значение параметра γ ? очень больше значение параметра β ?

5. ВАРИАНТЫ ЗАДАНИЙ

№	Уравнение	Значения свободных параметров A, B	
		A	B
1	$(x_1-A)^2+(x_2-B)^2$	1	2
2	$2(x_1-A)^2+(x_2-B)^2$	1	1
3	$3(x_1-A)^2+5(x_2-B)^2$	2	1
4	$(x_1-A)^4+(x_2-B)^4$	1	2
5	$(A-x_1)^2+B(x_1^2-x_2)^2$	3	4
6	$(x_1-A)^2+(x_2-B)^2$	5	-1
7	$2(x_1-A)^2+(x_2-B)^2$	-3	2
8	$3(x_1-A)^2+5(x_2-B)^2$	1	7
9	$(x_1-A)^4+(x_2-B)^4$	2	4
10	$(A-x_1)^2+B(x_1^2-x_2)^2$	3	2
11	$(x_1-A)^2+(x_2-B)^2$	4	4
12	$2(x_1-A)^2+(x_2-B)^2$	3	3
13	$3(x_1-A)^2+5(x_2-B)^2$	1	2
14	$(x_1-A)^4+(x_2-B)^4$	-1	-1
15	$(A-x_1)^2+B(x_1^2-x_2)^2$	-1	-1
16	$(x_1-A)^2+(x_2-B)^2$	3	2
17	$2(x_1-A)^2+(x_2-B)^2$	4	4
18	$3(x_1-A)^2+5(x_2-B)^2$	3	3
19	$(x_1-A)^4+(x_2-B)^4$	1	2
20	$(A-x_1)^2+B(x_1^2-x_2)^2$	1	20
21	$(x_1-A)^4+(x_2-B)^4$	2	-1

Точность, с которой необходимо искать решение уравнения, одинакова для всех вариантов и равна 10^{-6}.