Методы оптимизации в анализе данных. Вариант 14. Задание 5. Белоножко Илья Николаевич. КНиИТ 241

Задание. Реализовать на любом из языков программирования метод циклического покоординатного спуска, метод Хука-Дживса, метод случайного поиска. Найти минимум функции. Сравнить число итераций, необходимых для решения

14)

**Метод циклического покоординатного спуска**

Идея метода состоит в том, чтобы минимизировать исходную функцию по направлению первого базиса(для 2-мерной функции [1, 0]), потом второго([0, 1]), и так продолжать, пока функция при прохождении к каждому базису не будет меняться на малую величину. То есть, условием останова явялется . Полный алгоритм:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Программа:

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

**Метод Хука-Дживса**

В этом методе предварительно определяется этап исследующего поиска. То есть, чтобы найти направление, в котором функция убывает. Метод содержит 2 процедуры:

1. Исследующий поиск в окрестности данногй точки *x* для опеделения направления убывания функции *f(x)*. В результате получим точку .
2. Перемещение в направлении убывания *()*

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Если в результате случилось так, что , то проверяется норма приращения. Если она меньше заданной точности, то минимум найден. Если нет, полагают, что (где коэффициент дробления шага > 1) и повторяют исследующий поиск. Итоговый алгоритм метода Хука-Дживса:

*Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание*

Программа:

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

**Случайный поиск с возвратом при неудачном шаге**

Идея этого метода в том, чтобы делать шаг в случайном направлении, с заданной величиной шага. Если в новой точке функция меньше, то алгоритм продолжается из этой точки. Если число неудачных шагов достигает числа M, то величина шага уменьшается. Это продолжается, пока шаг не станет меньше заданной точности.

Алгоритм:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Программа:

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Результаты для каждого метода:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание