Лабораторная работа № 3

Тема: Решение задач методом случайного поиска.

Задание:

Найти минимум функции:

$$y=\cos(x)+\frac{1}{b}\cos(a*x+1)+\frac{1}{b^2}\cos(a^2*x+2)+\frac{1}{b^3}\cos(a^3*x+3)+\frac{1}{b^4}\cos(a^4*x+4)$$

где a — кол-во букв в фамилии,

b — введённое пользователем значение в диапазоне [2, 10] **.**

Ход работы:

1. Организовать ввод пользователя: Стартовая точка (x_0) и параметр b. Проверить правильность ввода параметров.
2. Реализовать предлагаемый метод случайного поиска. Программа должна выводить промежуточную информацию, когда происходит «улучшение» функции, в виде:
   <#> <y> <x> $<\alpha>$
   (итерация) (функционал) (аргумент) (текущее значение шага)
3. Протестировать программу. Т.к. стартовая точка может быть «далека» от «хорошего» решения (локального или глобального минимума), для поиска решения может понадобиться несколько запусков программы. Провести тесты с разными значениями начального шага α . Определить оптимальный вариант комбинации (поиска, сходимости): (α , *c*).

Предлагаемый алгоритм для решения задачи:

1. Задать начальную точку x_0 . Задать начальный шаг α .
2. Вычислить значение функционала y(x_0).
3. Сгенерировать новую точку x_1 по закону:
   $x_1 = x_0 + \alpha * rand$,
   где: α - шаг, *rand* – случайное число равномерно распределенное в диапазоне (-1,1).
4. Вычислить значение функционала y(x_1).
5. Если y(x_1) < y (x_0) , тогда принимаем $x_0 = x_1$ и $y^* = y(x_1)$.
6. Изменяем шаг: $\alpha = c\alpha$, где $c \in (0.75, 0.999)$. Параметр *c* характеризует процесс замедления интенсивности поиска, т.е. чем дольше идёт поиск, тем меньше «разброс» в генерации новой точки (x_1). В п.1 алгоритма можно увеличить параметр шага, что бы на начальных этапах попытаться найти глобальный экстремум.
7. Проверка условия останова: Если условие останова выполняется, тогда завершаем работу алгоритма, иначе переход к **п. 3**. Условия: а) Последние n итераций не происходило «улучшение» функции (n — параметр терпения); б) Значение шага стало очень маленьким и дальнейший поиск может не дать лучших результатов ($\alpha < \alpha_{\min}$, можно определить $\alpha_{\min} = 0.0001$); в) Ограничение на количество итераций N_{max}.

Примечание 1. Данный подход относится к классу «жадных» алгоритмов, т.к. принимается лучшее решение на каждом шаге.

Примечание 2. Правила изменения напоминают градиентный метод, только новая точка выбирается в некоторой окрестности текущей точки с радиусом α .

Примечание 3. Алгоритм показывает хорошие результаты, если заменить генератор распределения для *rand* на нормальное (0, σ).

Примечание 4. Как можно изменить алгоритм, если разрешить функции иногда «ухудшать» свое значение?