Санкт-Петербургский политехнический университет Петра Великого
Институт прикладной математики и механики
Кафедра «Прикладная математика»

ОТЧЁТ ПО ЛАБОРАТОРНЫМ РАБОТАМ ПО ДИСЦИПЛИНЕ «ВЫЧИСЛИТЕЛЬНЫЕ КОМПЛЕКСЫ»

Выполнил
студент группы 3630102/70201

Крупкина Дарья

Проверил
к. ф.-м. н., доцент

Баженов Александр Николаевич

Санкт-Петербург
2020

Содержание

Список таблиц

Список иллюстраций

1 Постановка задачи

Для демонстрации интервальной глобальной минимизации использовать функцию:

$$function[Z, WorkList] = globopt0(X). \tag{1}$$

Она возвращает значение глобального экстремума Z и рабочий список WorkList. Работа алгоритма построена на последовательном сужении множества, на котором строится оптимум.

1.1 Задача 1

Рассмотреть пример из лекционного материала. Построить рабочий список, построить график сужения интервала.

1.2 Задача 2

Взять пример с [1] и изучить сходимость.

2 Теория

2.1 Алгоритм GlobOpt

Алгоритм для глобальной минимизации функции GlobOpt оперирует с рабочим списком ζ, в котором будут храниться все брусы, получающиеся в результате дробления исходного бруса области определения на более мелкие подбрусы.
Одновременно с самими подбрусами будем хранить в рабочем списке и нижние оценки областей значений целевой функции по этим подбрусам, так что элементами списка ζ будут записи-пары вида:

$$\zeta : (Y, y), \text{где} Y \subseteq X, y = f(Y). \tag{2}$$

Далее каждый шаг алгоритма состоит в извлечении из этого списка бруса, который обеспечивает рекордную (т. е. наименьшую) на данный момент оценку минимума снизу, его дроблении на более мелкие подбрусы, оценивании на них целевой функции, занесении результатов обратно в рабочий список.

2.2 Функция Растригина

Имеет вид:

$$f_R = x^2 + y^2 - \cos 18 \cdot x - \cos 18 \cdot y \tag{3}$$

Минимум функции достигается при значении аргумента x = (0, 0) и равен 2.

2.3 Cross-in-tray

Имеет вид:

$$f_C = -0.0001[|\sin(x)\sin(y)\exp(|100 - \frac{\sqrt{x^2+y^2}}{\pi}|)| + 1]^{0.1} \tag{4}$$

Минимум функции достигается при значении аргумента x = (±1.3494, ±1.3494) и равен 2.0626.

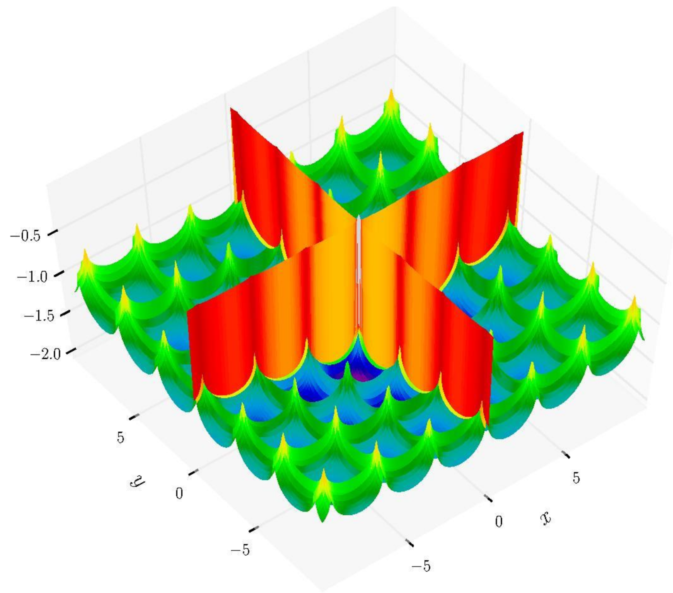

Рис. 1: Cross-in-tray function

3 Реализация

Лабораторная работа выполнена с помощью встроенных средств языка программирования Matlab. Использованы библиотеки IntLab для интервальной арифметики и оптимизатор globopt0. Исходный код лабораторной работы приведён в приложении в виде ссылки на репозиторий GitHub.

4 Результаты

4.1 Задача 1

Рабочий список будет состоять из n структур, содержащих в себе значения бруса и целевой функции, где n - число итераций.

Для выражения зависимости ширины от итерации будем рассматривать ширину как сумму диаметров двух интервальных компонент бруса.
Начальное значение ширины будет: $d([-5,5]) + d([-5,5]) = 20$. Приведем сам график:

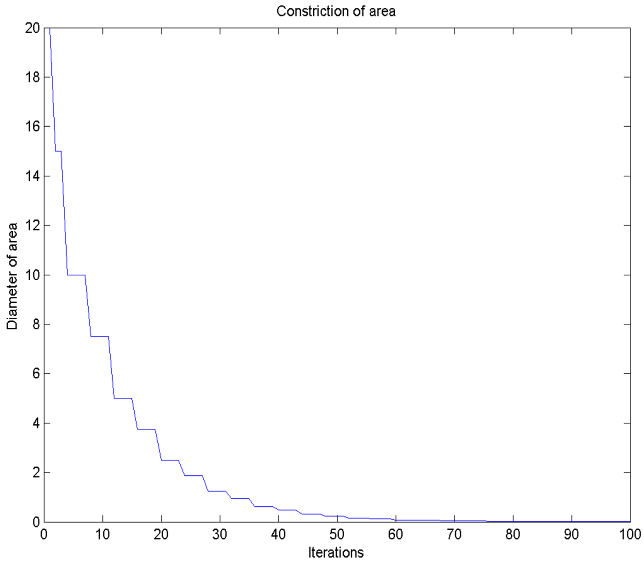

Рис. 2: Сужение бруса для функции Растригина

4.2 Задача 2

Рассмотрим тот же оптимизатор для функции (4). Минимум cross-in-tray равен 2.0626. Рассмотрим зависимость значения целевой функции от числа итераций.

Число итераций	$min(f_C(x, y))$
10	-2.2026
100	-2.0821
200	-2.0711
1000	-2.0640

Таблица 1: Зависимость вычисленного минимума функции cross-in-tray от числа итераций

Как видно из (1), с увеличением числа итераций значение целевой функции приближается к реальному и для 1000 итераций globopt находит достаточно точное значение.

Рассмотрим теперь для 1000 итераций зависимость абсолютной погрешности $|x_* - x_n|$ найденного значения на данной итерации x_n и значения минимума x_* от числа итераций.

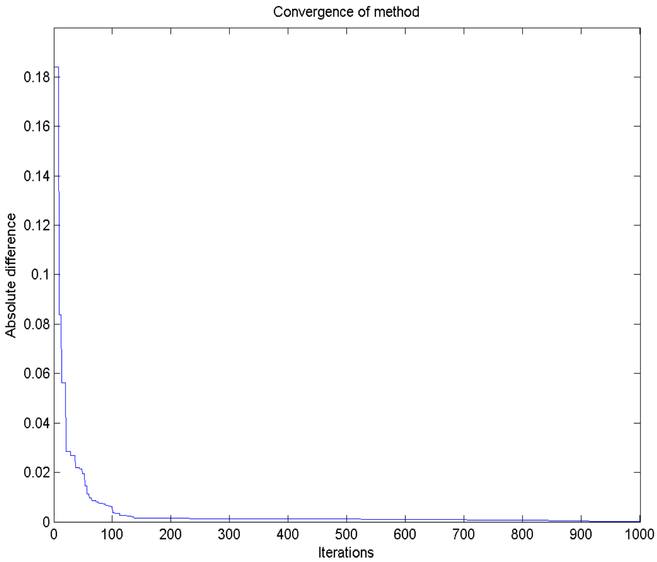

Рис. 3: Зависимость абсолютной погрешности от числа итераций для cross-in-tray

Тот же график сходимости с логарифмическом масштабе по оси ординат:

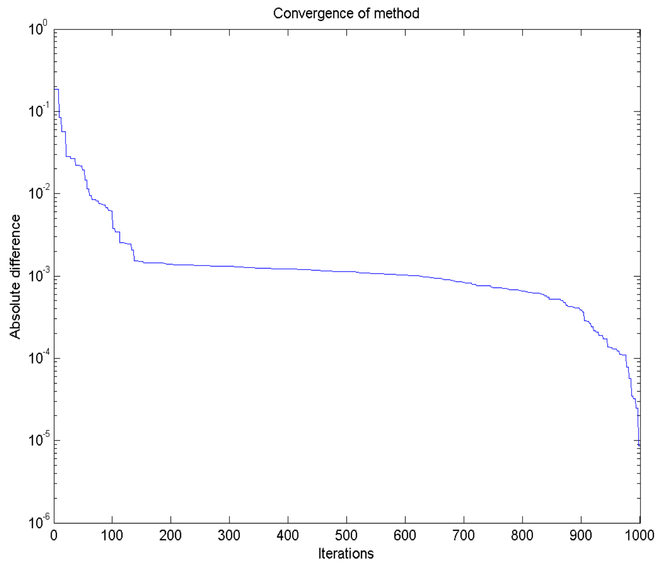

Рис. 4: График сходимости метода для cross-in-tray с логарифмическим масштабом по оси ординат

Видно, что метод обладает достаточно слабой сходимостью: нужно сделать порядка 200 итераций, чтобы добиться погрешности 10^{-3}.
Также рассмотрим траекторию центров брусов для последних 100 итераций, чтобы убедиться, обходит ли метод овраг:

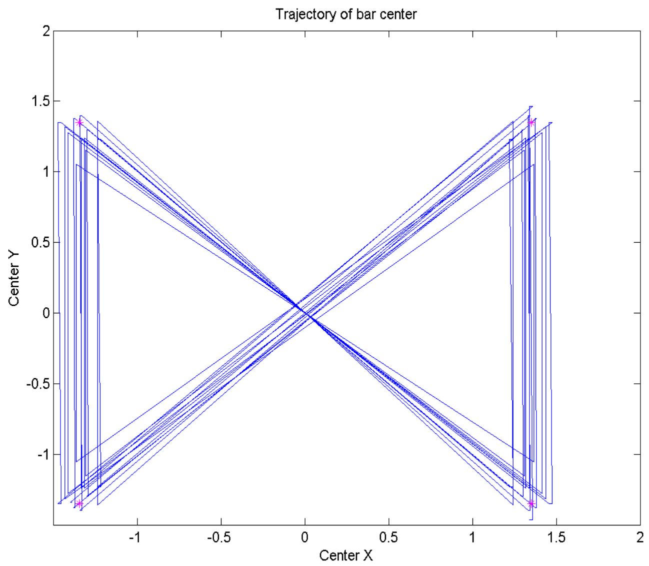

Рис. 5: Траектория центров брусов

Отдельно отмечены точки экстремума.

5 Приложения

Код программы на GitHub, URL: `https://github.com/DariaKrup/Computational_complexes`

Список литературы

[1] Функции для оптимизации. https://en.wikipedia.org/wiki/Test_functions_for_optimization