

Университет ИТМО

Институт прикладных компьютерных наук
Глубокое обучение и генеративный искусственный интеллект

ОТЧЕТ ПО 3-Й ЛАБОРАТОРНОЙ РАБОТЕ
курса
«Эволюционные вычисления»

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ДЛЯ ЗАДАЧИ ОПТИМИЗАЦИИ
НЕПРЕРЫВНОЙ ФУНКЦИИ

Студент:
Группа № М4130

Батурина Ксения Александровна

Санкт-Петербург 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 РЕАЛИЗАЦИЯ И РЕЗУЛЬТАТЫ	4
1.1 Реализация операторов	4
1.1.1 Оператор инициализации	4
1.1.2 Оператор кроссовера	4
1.1.3 Оператор мутации	4
1.2 Результаты экспериментов	5
2 ВОПРОСЫ	6
2.1 Что важнее, кроссовер или мутация?	6
2.2 Как влияет значение параметра «размер популяции» на производительность и эффективность алгоритма?	6
2.3 Важно ли знать область определения переменных целевой функции?	6

ВВЕДЕНИЕ

Код доступен в репозитории на GitHub: https://github.com/xeniabaturina/ITMO_EVOL/tree/main/lab3.

Цель работы:

Получить навыки разработки и анализа эволюционных операторов генетического алгоритма для решения задачи оптимизации непрерывной вещественнозначной функции.

Задачи работы:

1. Скачать проект, основанный на фреймворке Watchmaker.
2. Реализовать необходимые операторы.
3. Провести эксперименты.
4. Ответить на вопросы:
 - Что важнее, кроссовер или мутация?
 - Как влияет значение параметра «размер популяции» на производительность и эффективность алгоритма?
 - Важно ли знать область определения переменных целевой функции?

1 РЕАЛИЗАЦИЯ И РЕЗУЛЬТАТЫ

1.1 Реализация операторов

Для решения задачи были реализованы требуемые операторы. Перед проведением экспериментов были дополнительно проведены запуски с различными значениями используемых параметров для определения их оптимальных значений.

1.1.1 Оператор инициализации

Класс `MyFactory` используется для инициализации начальной популяции, позволяя создавать случайных особей для эволюции.

Создается новая особь (вектор типа `double[]`), представляющая собой случайный набор значений в указанном диапазоне для каждой координаты вектора.

Параметры: граничные значения координат векторов

1.1.2 Оператор кроссовера

В данной реализации для каждой координаты вектора родительских особей создаются две новые особи.

Каждая новая координата рассчитывается как линейная комбинация соответствующих координат родительских особей с разными весами. Один родитель вносит больший вклад в нового потомка, чем другой.

Таким образом, это линейный кроссовер.

Параметры: коэффициент для балансировки влияния каждого из родителей

1.1.3 Оператор мутации

Выбор стратегии мутации зависит от вероятности, которая рассчитывается с учетом номера текущего поколения. Если случайное значение меньше порога, применяется равномерная мутация; иначе — мутация методом Гаусса.

В случае равномерной мутации каждый индивид в популяции может изменить свои значения с определенной вероятностью, получая новые случайные значения в заданном диапазоне.

Мутация методом Гаусса также изменяет значения индивидов, но с использованием гауссовского распределения для случайных приращений. Эта мутация более вероятна в более поздних поколениях.

Таким образом, оператор мутации поддерживает гибкую стратегию, обеспечивая в начальных поколениях больше *exploration*, а в последующих — большую степень *exploitation*.

Параметры: стандартное отклонение, коэффициент вероятности стандартной мутации или методом Гаусса.

1.2 Результаты экспериментов

В соответствии с заданием была проведена серия запусков для решения задачи с размерностями 2, 10, 20, 50 и 100. В таблице ниже приведены размер популяции и количество итераций алгоритма для всех запусков и расчет средних значений.

Таблица 1 — Результаты экспериментов по производительности алгоритма при увеличении размерности проблемы.

Размер проблемы	Размер популяции	Количество итераций	Результат
2	20	83,2	9,96
10	20	957,9	9,58
20	20	2438,6	9,54
50	10	5462,5	9,53
100	5	9963,3	9,36

2 ВОПРОСЫ

2.1 Что важнее, кроссовер или мутация?

Кроссовер вводит элемент обмена информацией между родительскими особями и позволяет комбинировать положительные стороны нескольких решений. Это может потенциально ускорить сходимость алгоритма.

Мутация важна для сохранения разнообразия в популяции и избегания преждевременной сходимости к локальным оптимумам.

Эффективность алгоритма часто заключается в балансе между кроссовером и мутацией. Слишком интенсивное использование кроссовера может привести к потере разнообразия в популяции, тогда как избыточная мутация может замедлить сходимость.

2.2 Как влияет значение параметра «размер популяции» на производительность и эффективность алгоритма?

Увеличение размера популяции может улучшить разнообразие генетического материала и помочь избежать преждевременной сходимости к недостаточно оптимальным решениям. Однако слишком большой размер популяции может привести к увеличению вычислительной сложности и замедлению работы алгоритма. Поэтому необходим баланс между размером популяции и доступными вычислительными ресурсами для достижения оптимальной производительности.

2.3 Важно ли знать область определения переменных целевой функции?

Да. Если область определения известна, можно настроить параметры алгоритма, что может значительно улучшить его эффективность. Без этого алгоритм может тратить ресурсы на поиск решений за пределами допустимой области, что может привести к долгому времени сходимости и в целом будет неэффективно.