

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по учебной практике
Тема: ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Студенты гр. 2384

Преподаватель

Соц Е.А.
Поглазов Н.В.
Жангиров Т.Р.

Санкт-Петербург
2024

Цель работы.

Почти полностью реализовать генетический алгоритм, какой-то функционал может отсутствовать.

Задание.

Для заданного полинома $f(x)$ (степень не больше 8) необходимо найти параметры ступенчатой функции $g(x)$ (высота “ступеней”), которая приближает полиномиальную функцию, то есть минимизировать расстояние $|f(x) - g(x)|$ между функциями на заданном интервале $[l, r]$. Количество и длина ступеней вводятся пользователем.

Выполнение работы.

GUI.

В GUI была добавлена возможность просмотра работы генетического алгоритма как по шагам (кнопка next), так и самое последнее поколение (кнопка fast-forward). Также добавлена возможность автоматического обновления состояния работы ГА (кнопка play). Частота обновления измеряется в миллисекундах и задается пользователем в углу контрольной панели.

Также было добавлено новое диалоговое окно для настройки гипер параметров алгоритма.

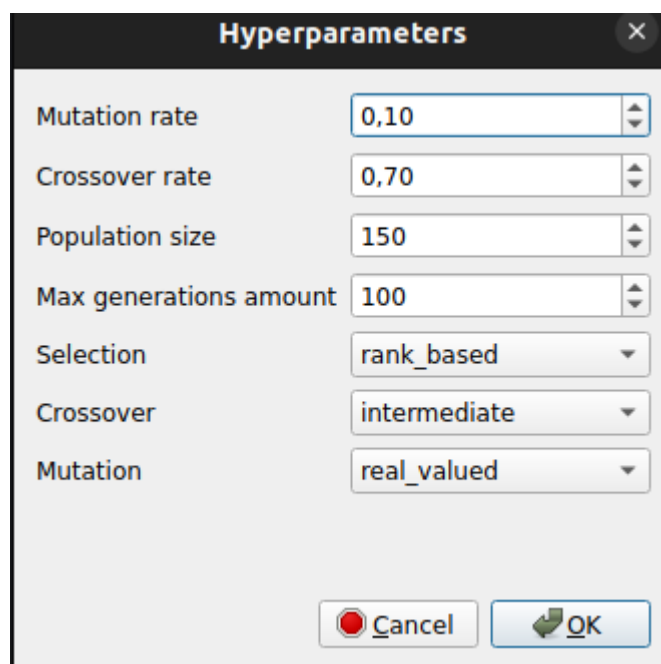


Рис.1. Диалоговое окно настройки гипер параметров.

Оно имеет следующие поля:

- Вероятность мутации - число типа double $\in [0, 1]$.
- Вероятность скрещивания - число типа double $\in [0, 1]$.
- Размер популяции - натуральное число.
- Максимальное число поколений - натуральное число.
- Выбор операторов селекции, скрещивания и мутации.

После нажатия кнопки ОК эти данные сохраняются в формате json и используются для GeneticHyperInitializer.

Кнопка для вызова этого окна была добавлена в ParametersDock.

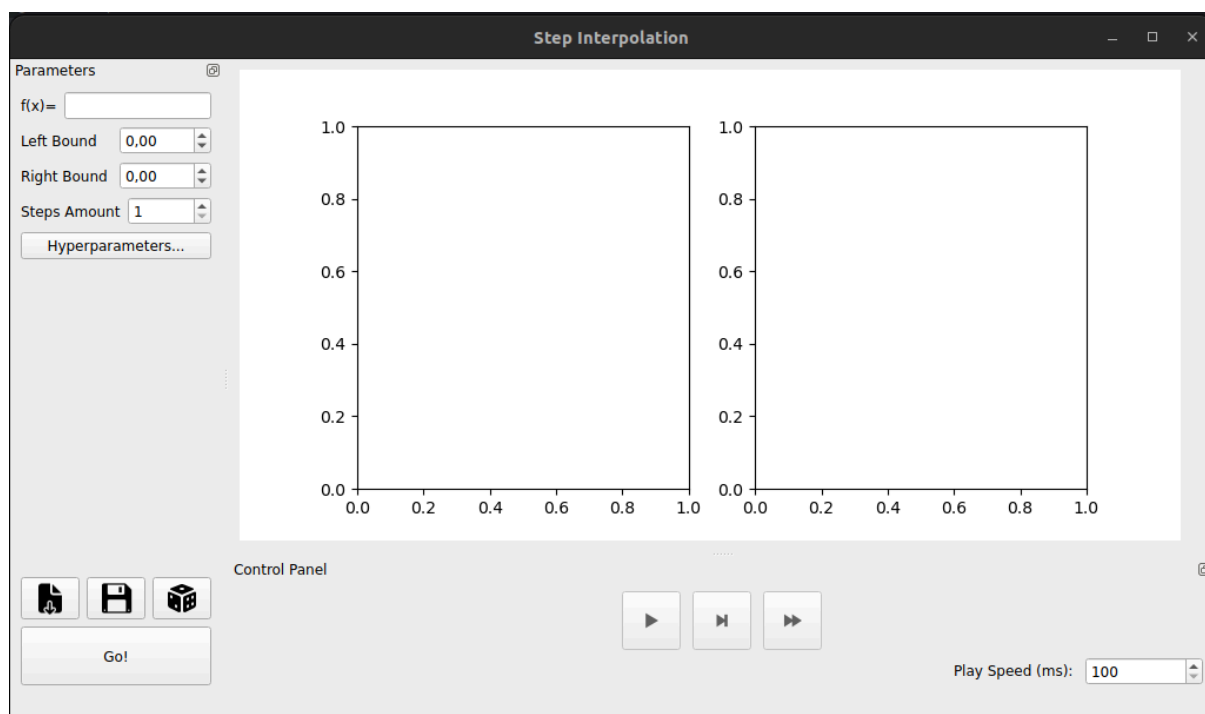


Рис.2. Текущий вид интерфейса.

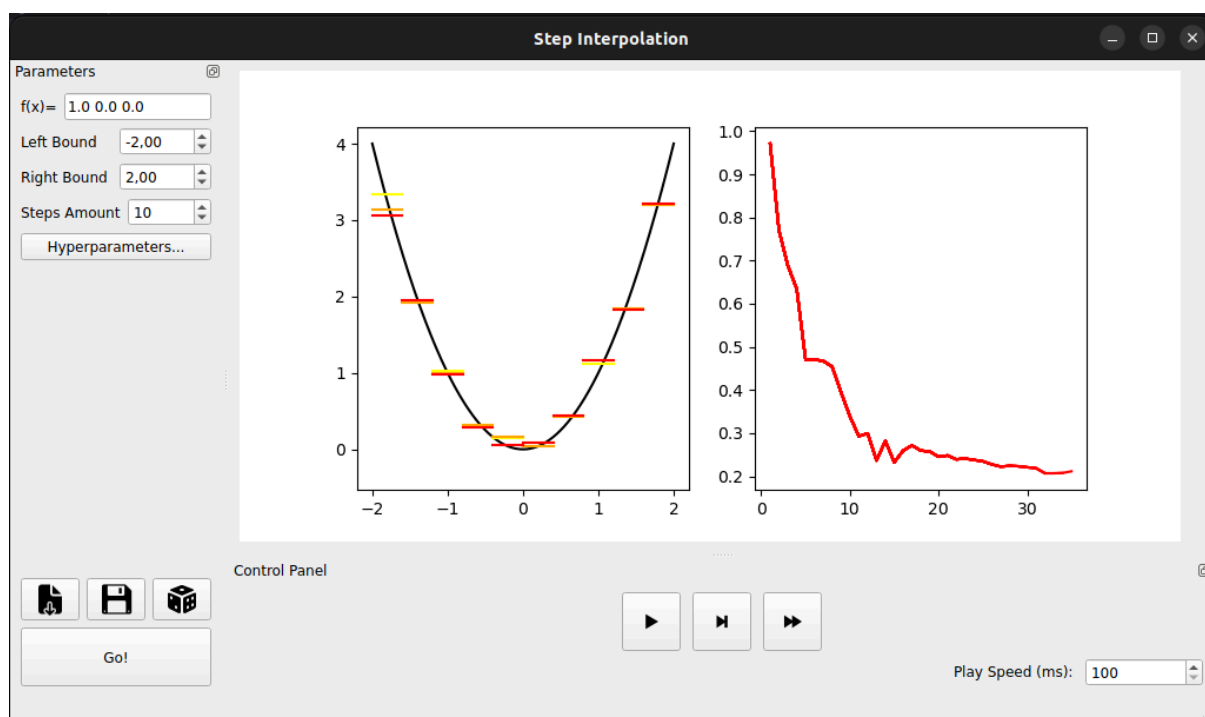


Рис. 3. Демонстрация работы алгоритма.

На графике с многочленом также отображаются три лучших решения (красный - наилучшее, оранжевый - топ 2, желтый - топ 3). Справа находится график целевой функции, который также обновляется с каждым шагом.

Рефакторинг кода.

Для соблюдения принципа единственной ответственности класс MainWindow был разбит на несколько вспомогательных классов:

- ControlPanelDock - контрольную панель (с кнопками для управления ходом ГА) было решено вынести в отдельный DockWidget, чтобы добавить гибкости интерфейсу и переложить на него ответственность по управлению сессией.
- PlotManager - класс отвечающий за отрисовку многочлена и ступенчатых функций на холсте. Хранит сам объект холста MplCanvas.
- GeneticAlgorithmManager - класс отвечающий за инициализацию генетического алгоритма. Хранит объект GeneticAlgorithm.
- SessionManager - класс отвечающий за обновление сессии (обновление графиков и настроек ГА). Хранит объекты PlotManager, GeneticAlgorithmManager а также настройки сессии (словарь SettingsData).

Также реструктуризации подвергся класс ParametersDock (был разбит на FileDialogManager для запуска диалогов сохранения/загрузки данных сессии, ParametersValidator для проверки правильности введенных параметров и SettingsManager для взаимодействия с SettingsData), а также инициализация класса GeneticAlgorithm была разбита на два класса:

- GeneticHyperInitializer - инициализатор гиперпараметров.
- GeneticSessionInitializer - инициализатор параметров сессии.

Добавление новых операторов ГА.

Для проведения эксперимента (какие операторы лучше использовать в поставленной задаче) были реализованы дополнительные операторы: отбор рулеткой и мутация обменом.

Метод рулетки для отбора представляет собой вероятностный отбор: чем лучше приспособленность индивидуума, тем больше вероятность, что в качестве родителя выберется именно этот индивидуум. Возвращает два выбранных родителя.

Метод мутации обменом: случайным образом выбираются два гена и меняются местами. Данный метод применяется к потомку после скрещивания с заданной вероятностью. Возвращает измененный (или прежний) индивидуум.

Вывод.

Были реализованы дополнительные операторы: мутации и отбора.