

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.

Изучить генетические алгоритмы, описать решение поставленной задачи: представление данных, метрика качества, какой ГА будет использоваться. Сделать прототип GUI.

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.

Основные теоретические положения.

Задача генетического алгоритма (ГА) - найти оптимальное решение некоторой задачи, путем развития популяции потенциальных решений называемых индивидами. Решения итеративно оцениваются и используются для создания следующего поколения.

Генотип - описание одного индивида, набор генов, сгруппированных в одну хромосому.

Популяция - множество индивидов (потенциальных решений), которые хранит генетический алгоритм. Популяция отображает текущее поколение.

Функция приспособленности (целевая функция) - функция, которую необходимо оптимизировать в рамках решаемой задачи. На каждой итерации ГА рассчитывает приспособленность индивида для формирования нового поколения.

Отбор - формирование множества индивидуумов для использования следующего поколения.

Скращивание - процесс создания пары новых индивидуумов (потомков), путем комбинации хромосом двух родителей из текущей популяции.

Мутация - процесс случайного изменения хромосомы для поиска неисследованных областей и поддержания разнообразия популяции.

Выполнение работы.

Основные методы алгоритма.

Ген - неделимая часть хромосомы, отвечающая за конкретный признак. В поставленной задаче ген будет означать высоту ступени. Таким образом, хромосома - набор генов. Размер хромосомы равен количеству ступеней, которое вводит пользователь.

Целевая функция должна минимизировать расстояние между ступенчатой функцией и заданным полиномом, тогда для оценивания данного отклонения был выбран метод RMSE (корень среднеквадратической ошибки). Обоснование данного выбора заключается в легкой интерпретации значения такой ошибки (единицы измерения ошибки совпадают с единицами измерения длины ступени). Условиями остановки алгоритма являются использование заданного количества шагов и нахождение решения с минимальным значением ошибки относительно заданного значения. Таким образом, алгоритм всегда закончится, при этом будет найдено наилучшее относительно просмотренных решений, что говорит о корректности алгоритма. При этом инвариантом данного алгоритма будет являться размер популяции, который корректируется во время его работы.

Для отбора в генетическом алгоритме используется селекция среди родителей. Она происходит по правилу ранжированного отбора. Данное правило заключается в вероятностном выборе родителя по сформированному списку в указанном порядке, который определяет ранг каждого из выборки. По итогу селекционно будут выбраны два родителя для последующего скрещивания.

Для выбранной пары будем производить равномерное скрещивание: для каждого гена с равным шансом выбирается значение первого или второго родителя для первого потомка. Для второго потомка выбирается

ген того родителя, который не попал в первого потомка. Такой метод позволяет равномерно изменять решение, так как распределение максимумов и минимумов на заданной функции чередуется.

Еще одним методом скрещивания, который планируется реализовать в работе, является промежуточная рекомбинация. В данном методе предварительно определяется числовой интервал значений генов потомков, который должен содержать гены родителей. Потомки создаются по правилу:

$$Offspring = Parent1 + \alpha * (Parent2 - Parent1), \alpha \in [-d, 1 + d]$$

— случайное число; $d \geq 0$

Для каждого гена создаваемого потомка выбирается множитель α . Такой метод позволяет создавать новые особи, пригодность которых может быть лучше, чем пригодность родителей, так как значения генов новых особей отличаются от значений генов особей-родителей.

После скрещивания с заданной вероятностью к потомку применяется мутация. Для мутации особей с вещественными числами необходимо определить величину шага мутации - число, на которое изменится значение гена при мутации. Гены могут мутировать согласно данному правилу: новая переменная = старая переменная $\pm \alpha \cdot \delta$, где знаки $+$ или $-$ выбираются с равной вероятностью, $\alpha = 0.5 * \text{поисковое пространство}$ (интервал изменения данной переменной),

$$\delta = \sum_{i=1}^m \frac{a(i)}{2^i},$$

$a(i) = 1$ с вероятностью $1/m$, в противном случае $a(i) = 0$, m - параметр. Новая особь при такой мутации в большинстве случаев не намного отличается от старой: вероятность маленького шага мутации выше, чем вероятность большого шага.

Шаги: выбор родителей, скрещивание и мутация - производятся до тех пор, пока новое потомство не станет такого же размера, каким была изначальная популяция.

Создание пользовательского графического интерфейса (GUI).

Для создания GUI было решено использовать фреймворк PyQt5. Это решение обусловлено (помимо опыта работы с ним) простотой, популярностью фреймворка а также возможностью интеграции Matplotlib в виджеты.

Для начала было создано главное окно приложения (`main_window.py` и `main_window_ui.py`).

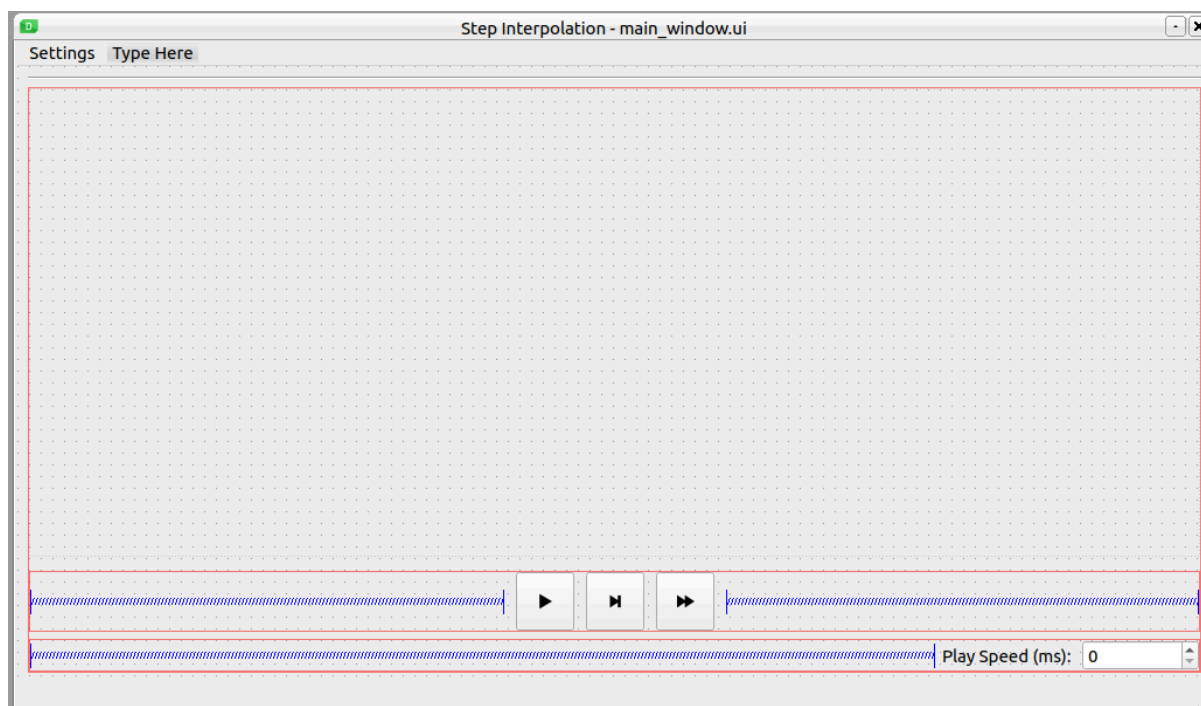


Рисунок 1. Верстка основного окна.

Большую часть окна занимает виджет `canvas`, в котором и будет отображаться `Figure`, создаваемая `matplotlib (mpl)`. Ниже располагаются кнопки запуска\паузы сеанса (после нажатия иконка меняется на паузу и обратно), перехода на шаг вперед (к следующему поколению) и перехода в конец сеанса. Еще ниже располагается `SpinBox` для изменения скорости обновления сеанса (в миллисекундах).

Для задания пользовательских параметров был создан DockWidget ParametersDock (parameters_dock.py и parameters_dock_ui.py). Этот виджет будет прикрепляться сбоку основного окна.

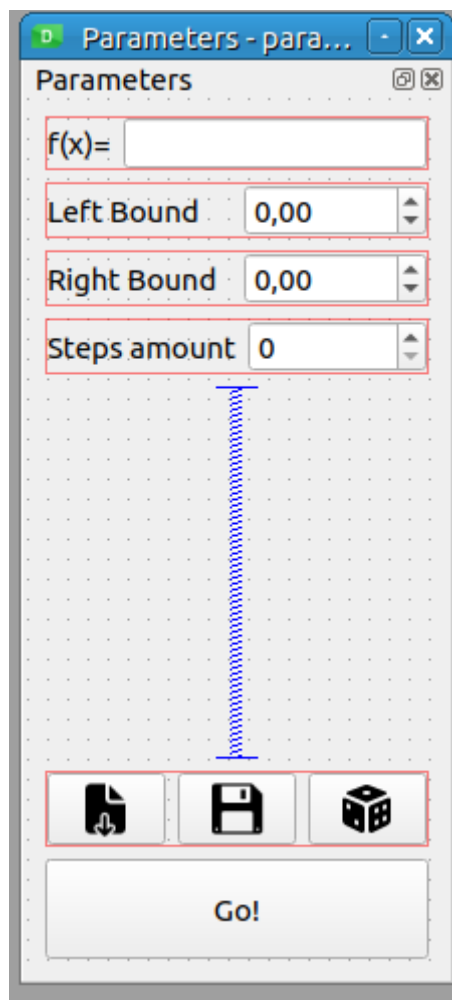


Рисунок 2. Верстка параметров.

Этот виджет содержит следующие поля для заполнения:

1. $f(x)$ - многочлен для приближения, степени не более 8. Представляется как набор вещественных чисел разделенных через пробел, которые представляют собой коэффициенты многочлена в порядке увеличения степени. (например многочлен $5x^2 + 0.2$ должен быть представлен как "0.2 0 5")
2. Left(Right) Bound - вещественное число, представляющее левую(правую) границу функции для приближения.

3. Steps Amount - натуральное число (≥ 2) количество “ступеней” приближающей функции.

Также внизу находятся кнопки для работы с параметрами:

1. Загрузка параметров из JSON файла.
2. Сохранение параметров в JSON файл.
3. Случайное задание параметров.

Кнопка запуска сеанса находится внизу виджета. При ее нажатии производится проверка всех полей, и вывод ошибки с описанием проблемы в случае некорректности данных. В случае успешной проверки эти данные передаются в основное окно для отрисовки функции.

Для реализации виджета для отрисовки функции, был использован класс `FigureCanvasQTAgg` из `mpl`. Это один из реализованных backend для отображения графиков, реализованный в библиотеке модуле `mpl`. Он используется в качестве родительского класса для `MplCanvas` (`mpl_canvas.py`)

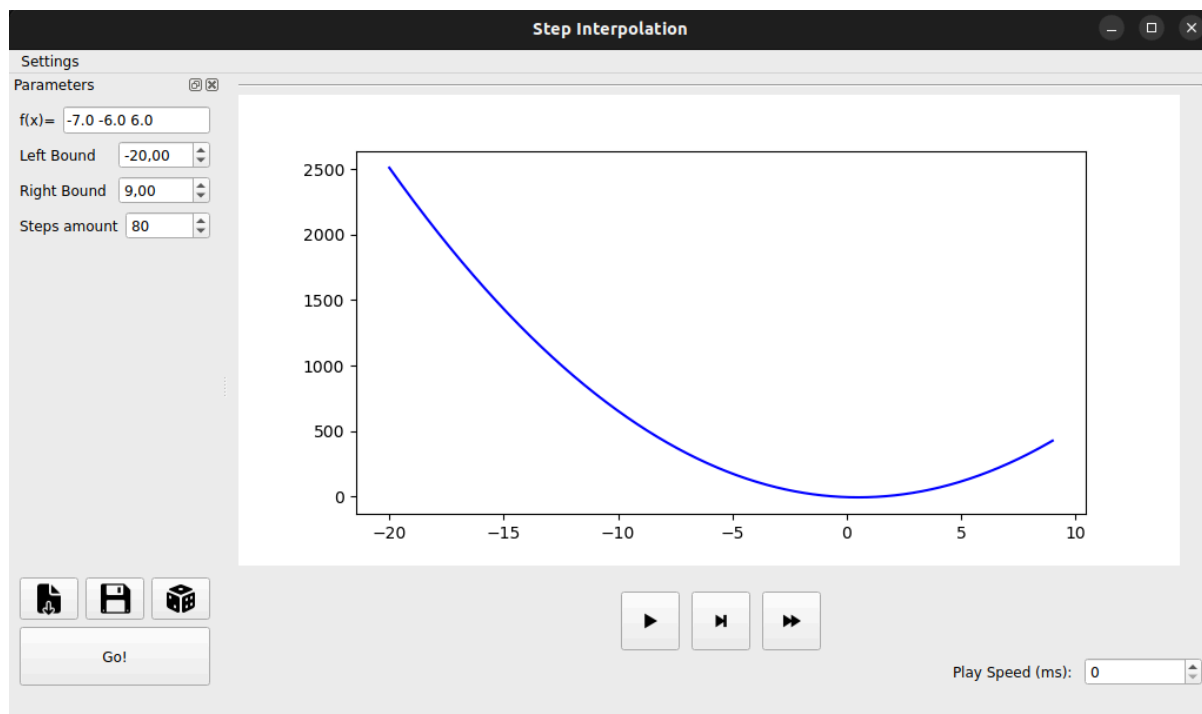


Рисунок 3. Текущий вид окна

Вывод.

В результате выполнения задания был изучен генетический алгоритм, а также прилагающиеся к нему, основанные на теории эволюции Дарвина, методы: селекции, скрещивании (кроссинговер), а также мутации. Было предложено два метода скрещивания, чтобы в последующем их сравнить экспериментально.

Также была сверстана реализована основная часть GUI.