

ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

д-р техн. наук, профессор

должность, уч. степень, звание

подпись, дата

Скобцов Ю.А.

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Генетическое программирование

по курсу: Эволюционные методы проектирования программно-информационных систем

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4132

подпись, дата

Н.И. Карпов

инициалы, фамилия

Санкт-Петербург 2024

СОДЕРЖАНИЕ

| | | |
|-----|---|----|
| 1 | Индивидуальное задание | 3 |
| 2 | Краткие теоретические сведения | 3 |
| 3 | Результаты выполнения работы | 3 |
| 3.1 | Исследование лучшего решения..... | 3 |
| 3.3 | Исследование решений при разных параметрах..... | 6 |
| 3.3 | Листинг программы | 8 |
| 4 | Вывод..... | 10 |

1 Индивидуальное задание

| № вв. | Вид функции | Кол-во пер-ых N | Промежуток исследования |
|----------|--|-----------------------|----------------------------|
| 6 | $f7(x)=\sum_{i=1:n}(-x(i) \cdot \sin(\sqrt{\text{abs}(x(i))}));$ | 10 | $-500 \leq x(i) \leq 500.$ |

2 Краткие теоретические сведения

В генетическом программировании (ГП) в качестве особи выступает программа, представленная в определенном формате, которая решает некоторую задачу. Часто это выполняется с использованием обучающих данных и индуктивного вывода. ГП очень близко к машинному обучению и поэтому в качестве фитнес-функции как правило выступают функции ошибки.

ГП работает с генетическим материалом переменной длины, что требует нестандартной формы представления генома и соответствующих генетических операторов. Программы состояются из переменных, констант и функций, которые связаны некоторыми синтаксическими правилами. Поэтому определяется терминальное множество, содержащее константы и переменные, и функциональное множество, которое состоит, прежде всего, из операторов и необходимых элементарных функций ($\exp(x)$, $\sin(x)$ и т.п.). Следует отметить, что терминалы и функции играют различную роль. Терминалы обеспечивают входные значения в систему (программу), в то время как функции используются при обработке значений внутри системы. Термины «функции» и «терминалы» взяты из древовидного представления, и соответствуют узлам древовидных (или графоподобных) структур.

3 Результаты выполнения работы

3.1 Исследование лучшего решения

В ходе выполнения программного решения были получены следующие результаты:

```
gen    nevals  min    avg
0      1000    393.759 inf
1      923     375.417 4.00921e+295
Лучшая особь: sin(x1)
```

Лучшая особь поколения 1



```
gen    nevals  min    avg
0      0       368.852 inf
1      935     364.792 inf
Лучшая особь: sin(sin(sqrt(cos(x5))))
```

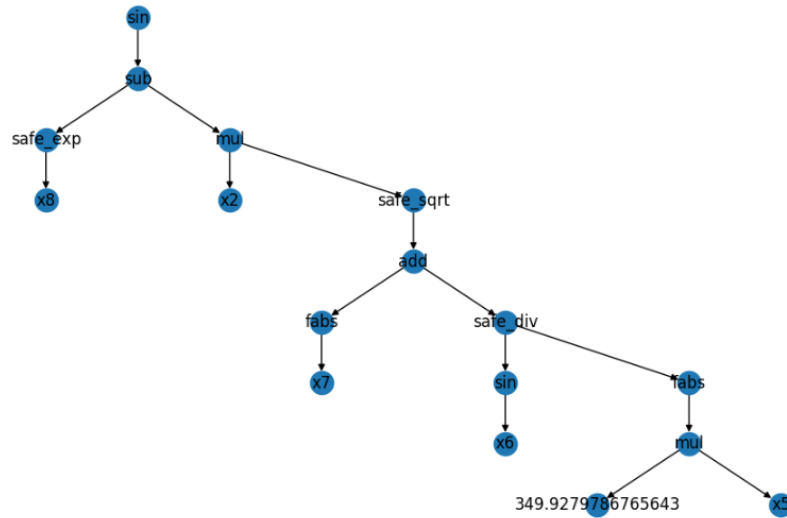
Лучшая особь поколения 11



| | | | |
|---|-----|---------|--------------|
| 0 | 0 | 341.586 | 1.73309e+213 |
| 1 | 922 | 341.586 | 2.54575e+212 |

Лучшая особь: $\sin(\text{sub}(\text{safe_exp}(x_8), \text{mul}(x_2, \text{safe_sqrt}(\text{add}(\text{fabs}(x_7), \text{safe_div}(\sin(x_6), \text{fabs}(\text{mul}(349.9279786765643, x_5))))))))))$

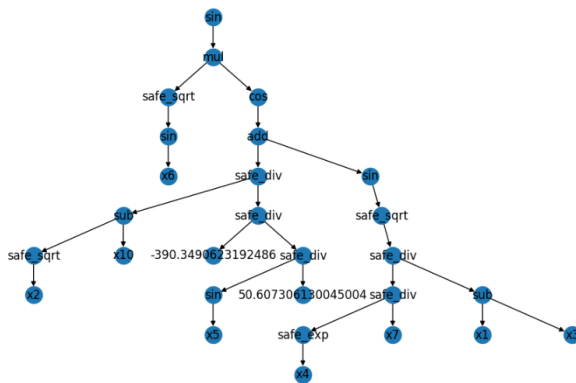
Лучшая особь поколения 21



| | | | |
|---|-----|---------|--------------|
| 0 | 0 | 374.818 | 1.45501e+211 |
| 1 | 937 | 364.334 | 2.81004e+213 |

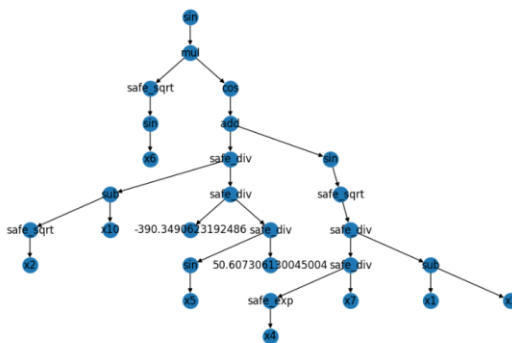
Лучшая особь: $\sin(\text{mul}(\text{safe_sqrt}(\sin(x_6)), \cos(\text{add}(\text{safe_div}(\text{sub}(\text{safe_sqrt}(x_2), x_{10}), \text{safe_div}(-390.3490623192486, \text{safe_div}(\sin(x_5), 50.607306130045004))))), \sin(\text{safe_sqrt}(\text{safe_div}(\text{safe_div}(\text{safe_exp}(x_4), x_7), \text{sub}(x_1, x_3)))))))))$

Лучшая особь поколения 91



Лучшая особь: $\sin(\text{mul}(\text{safe_sqrt}(\sin(x_6)), \cos(\text{add}(\text{safe_div}(\text{sub}(\text{safe_sqrt}(x_2), x_{10}), \text{safe_div}(-390.3490623192486, \text{safe_div}(\sin(x_5), 50.607306130045004))))), \sin(\text{safe_sqrt}(\text{safe_div}(\text{safe_div}(\text{safe_exp}(x_4), x_7), \text{sub}(x_1, x_3)))))))))$

Лучшая особь поколения 101



Лучший найденный индивид: $\sin(\text{mul}(\text{safe_sqrt}(\sin(x_6)), \cos(\text{add}(\text{safe_div}(\text{sub}(\text{safe_sqrt}(x_2), x_{10}), \text{safe_div}(-390.3490623192486, \text{safe_div}(\sin(x_5), 50.607306130045004))))), \sin(\text{safe_sqrt}(\text{safe_div}(\text{safe_div}(\text{safe_exp}(x_4), x_7), \text{sub}(x_1, x_3)))))))))$

В ходе поиска оптимального решения видно, что ошибка сильно уменьшается с возрастанием пройденных поколений, при этом само решение усложняется.

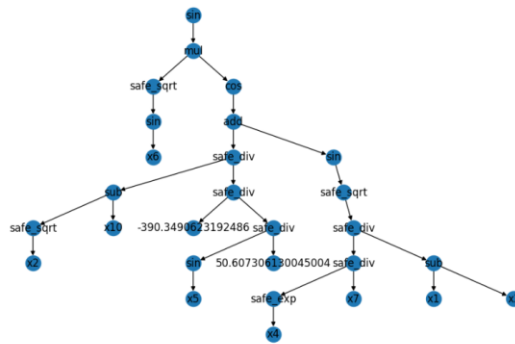
При этом даже лучшее решение (ошибка 340) далеко от искомой функции, ввиду того, что она – комбинаторно сложна.

3.3 Исследование решений при разных параметрах

В ходе работы было установлено, что размер популяции положительно влияет на точность, но негативно влияет на скорость вычислений:

Размер популяции 1000 (ошибка ~370):

Лучшая особь: $\sin(\text{mul}(\text{safe_sqrt}(\sin(x_6))), \cos(\text{add}(\text{safe_div}(\text{sub}(\text{safe_sqrt}(x_2)), x_{10}), \text{safe_div}(-390.3490623192486, \text{safe_div}(\sin(x_5), 50.607306130045004))))), \sin(\text{safe_sqrt}(\text{safe_div}(\text{safe_div}(\text{safe_exp}(x_4), x_7), \text{sub}(x_1, x_3))))))$
Лучшая особь поколения 101



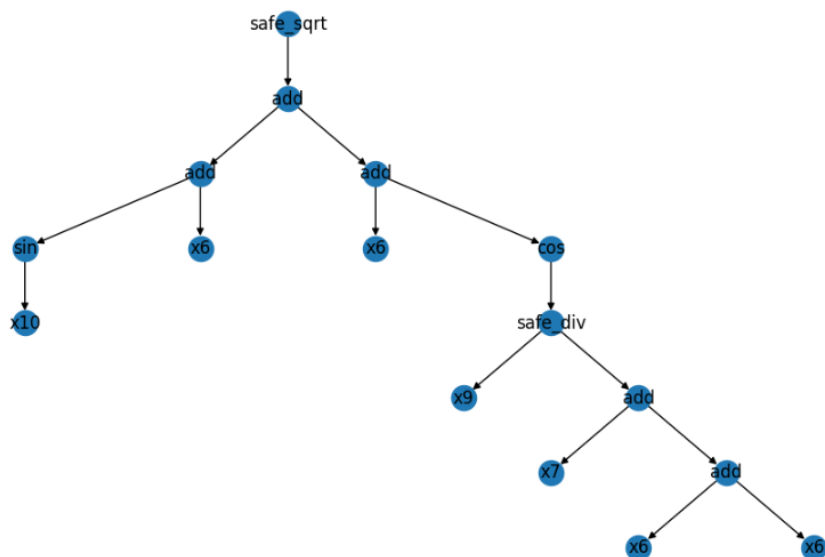
Лучший найденный индивид: $\sin(\text{mul}(\text{safe_sqrt}(\sin(x_6))), \cos(\text{add}(\text{safe_div}(\text{sub}(\text{safe_sqrt}(x_2)), x_{10}), \text{safe_div}(-390.3490623192486, \text{safe_div}(\sin(x_5), 50.607306130045004))))), \sin(\text{safe_sqrt}(\text{safe_div}(\text{safe_div}(\text{safe_exp}(x_4), x_7), \text{sub}(x_1, x_3))))))$

Размер популяции 10 (ошибка ~420):

| gen | nevals | min | avg |
|-----|--------|---------|---------|
| 0 | 0 | 430.846 | 484.19 |
| 1 | 10 | 422.055 | 468.365 |

Лучшая особь: $\text{safe_sqrt}(\text{add}(\text{add}(\sin(x_{10}), x_6), \text{add}(x_6, \cos(\text{safe_div}(x_9, \text{add}(x_7, \text{add}(x_6, x_6))))))))$

Лучшая особь поколения 101



Лучший найденный индивид: $\text{safe_sqrt}(\text{add}(\text{add}(\sin(x_{10}), x_6), \text{add}(x_6, \cos(\text{safe_div}(x_9, \text{add}(x_7, \text{add}(x_6, x_6))))))))$

Большая вероятность кроссинговера или мутации уменьшает шанс попасть в локальный минимум, но ухудшает сходимость:

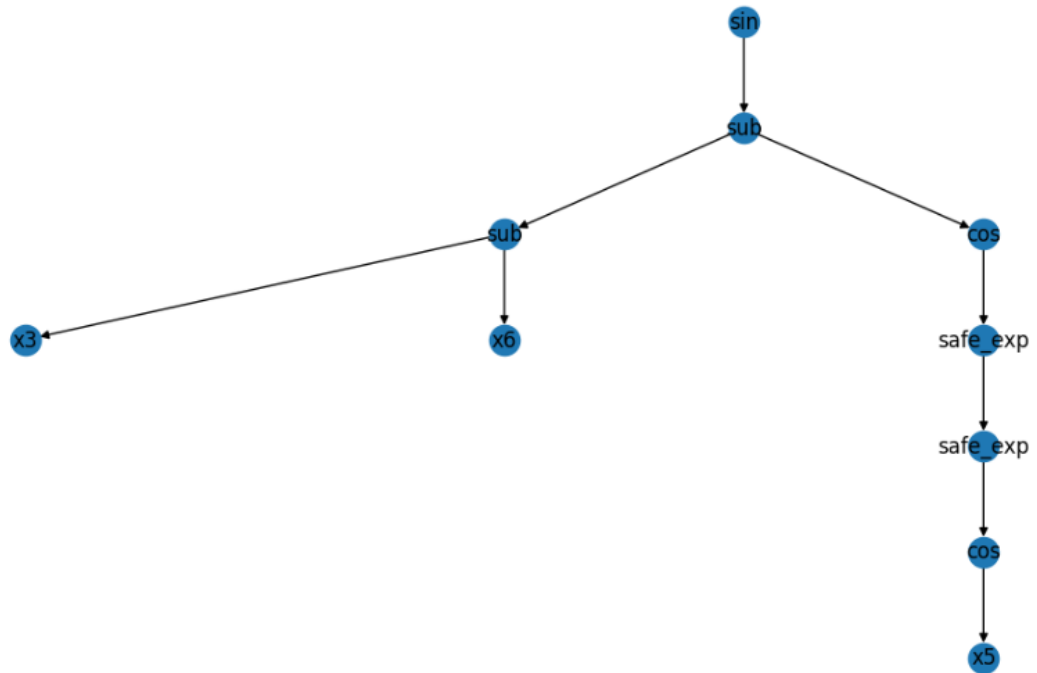
Ошибка при вероятности мутации 0,1; вероятности кроссинговера 0,5 составила 347,8

Лучшая особь: `sin(sub(sub(x3, x6), cos(safe_exp(safe_exp(cos(x5))))))`

| gen | nevals | min | avg |
|-----|--------|---------|--------------|
| 0 | 0 | 347.775 | 1.80885e+207 |
| 1 | 287 | 347.775 | 555.11 |

 Лучшая особь: `sin(sub(sub(x3, x6), cos(safe_exp(safe_exp(cos(x5))))))`

Лучшая особь поколения 101



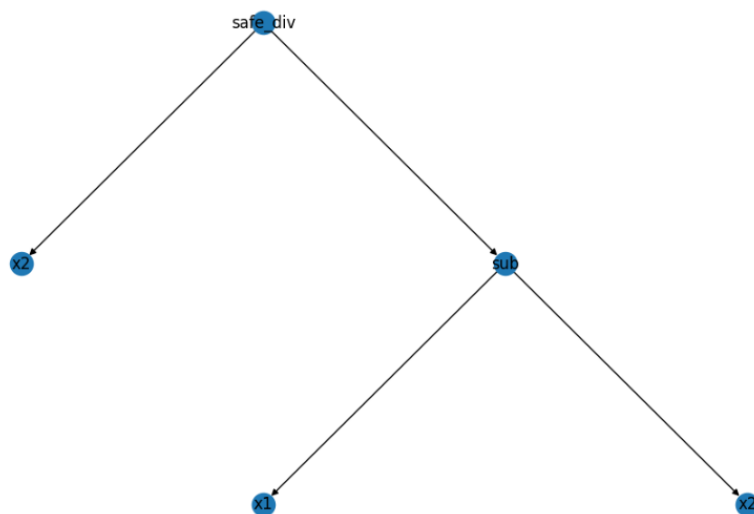
Ошибка при вероятности мутации 0,01; вероятности кроссинговера 0,5 составила 389,2:

Лучшая особь: `safe_div(x2, sub(x1, x2))`

| gen | nevals | min | avg |
|-----|--------|---------|-------------|
| 0 | 0 | 380.511 | 309829 |
| 1 | 456 | 389.247 | 3.1256e+198 |

 Лучшая особь: `safe_div(x2, sub(x1, x2))`

Лучшая особь поколения 101



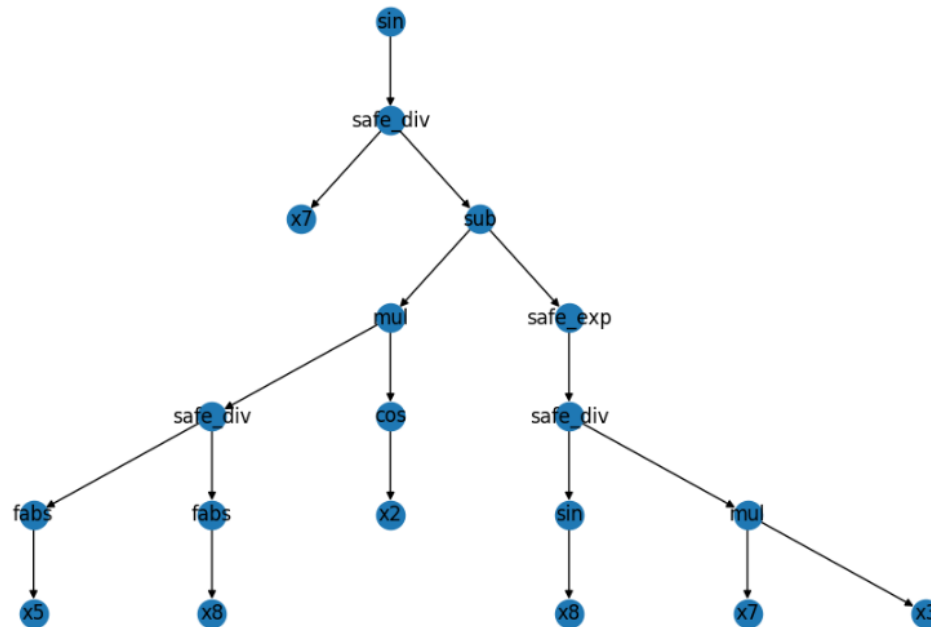
Лучший найденный индивид: `safe_div(x2, sub(x1, x2))`

Ошибка при вероятности мутации 0,2; вероятности кроссинговера 0,9 составила 367,4:

1 458 367.368 1.57555e+209

Лучшая особь: $\sin(\text{safe_div}(x_7, \text{sub}(\text{mul}(\text{safe_div}(\text{fabs}(x_5), \text{fabs}(x_8)), \cos(x_2)), \text{safe_exp}(\text{safe_div}(\sin(x_8), \text{mul}(x_7, x_3))))))$

Лучшая особь поколения 101



Лучший найденный индивид: $\sin(\text{safe_div}(x_7, \text{sub}(\text{mul}(\text{safe_div}(\text{fabs}(x_5), \text{fabs}(x_8)), \cos(x_2)), \text{safe_exp}(\text{safe_div}(\sin(x_8), \text{mul}(x_7, x_3))))))$

3.3 Листинг программы

```

import operator
import math
import random
import matplotlib.pyplot as plt
import networkx as nx

from deap import base, creator, gp, tools, algorithms

# Задаем функцию f7(x)
def f7(*x):
    return sum(-xi * math.sin(math.sqrt(abs(xi))) for xi in x)

# Создаем функциональное множество
pset = gp.PrimitiveSet("MAIN", arity=10) # N=10, входных переменных 10
pset.addPrimitive(operator.add, 2)
pset.addPrimitive(operator.sub, 2)
pset.addPrimitive(operator.mul, 2)
pset.addPrimitive(operator.truediv, 2)
pset.addPrimitive(abs, 1)
pset.addPrimitive(math.sin, 1)

```



```

pset.addPrimitive(math.cos, 1)
pset.addPrimitive(math.exp, 1)
pset.addPrimitive(math.sqrt, 1)
pset.addEphemeralConstant("rand", lambda: random.uniform(-500, 500))
pset.renameArguments(**{f"ARG{i}": f"x{i+1}" for i in range(10)})

# Определяем критерий оптимизации
creator.create("FitnessMin", base.Fitness, weights=(-1.0,))
creator.create("Individual", gp.PrimitiveTree, fitness=creator.FitnessMin)

# Базовые настройки
toolbox = base.Toolbox()
toolbox.register("expr", gp.genHalfAndHalf, pset=pset, min_=1, max_=3)
toolbox.register("individual", tools.initIterate, creator.Individual,
toolbox.expr)
toolbox.register("population", tools.initRepeat, list, toolbox.individual)

toolbox.register("compile", gp.compile, pset=pset)

# Фитнесс-функция
def evaluate(individual, target_function=f7):
    func = toolbox.compile(expr=individual)
    x_values = [[random.uniform(-500, 500) for _ in range(10)] for _ in
range(100)]
    errors = [
        abs(target_function(*x) - func(*x))
        for x in x_values
    ]
    return sum(errors) / len(errors),

toolbox.register("evaluate", evaluate)
toolbox.register("select", tools.selTournament, tournsize=3)
toolbox.register("mate", gp.cxOnePoint)
toolbox.register("mutate", gp.mutUniform, expr=toolbox.expr, pset=pset)

# Ограничение роста деревьев
toolbox.decorate("mate", gp.staticLimit(key=operator.attrgetter("height"),
max_value=17))
toolbox.decorate("mutate", gp.staticLimit(key=operator.attrgetter("height"),
max_value=17))

# Визуализация дерева
def plot_tree(individual, ax):
    nodes, edges, labels = gp.graph(individual)
    graph = nx.DiGraph()
    graph.add_nodes_from(nodes)
    graph.add_edges_from(edges)
    pos = nx.nx_agraph.graphviz_layout(graph, prog="dot")

```

```

nx.draw(graph, pos, with_labels=True, labels=labels, ax=ax)
ax.set_title("Дерево выражения")

# Основной цикл
random.seed(42)
population = toolbox.population(n=50)
halloffame = tools.HallOfFame(1)
stats = tools.Statistics(lambda ind: ind.fitness.values)
stats.register("min", min)
stats.register("avg", lambda x: sum(x) / len(x))

for gen in range(10): # Количество поколений
    population, logbook = algorithms.eaSimple(
        population,
        toolbox,
        cxpb=0.5,
        mutpb=0.2,
        ngen=1,
        stats=stats,
        halloffame=halloffame,
        verbose=True,
    )

    # Визуализация лучшего индивида в текущем поколении
    best_individual = halloffame[0]
    fig, ax = plt.subplots(figsize=(12, 8))
    plot_tree(best_individual, ax)
    plt.show()

print("Лучший найденный индивид:", halloffame[0])

```

4 Вывод

В ходе работы была создана программа для решения задачи символьной регрессии с помощью генетических алгоритмов. Полученные решения далеки от оптимальных ввиду большой сложности выбранной исходной функции.