Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №3.3

з дисципліни
«Інтелектуальні вбудовані системи»

на тему
«ДОСЛІДЖЕННЯ ГЕНЕТИЧНОГО АЛГОРИТМУ»

Виконала:

студентка групи ІП-84
Скрипник Єлена Сергіївна
номер залікової книжки: 8422

Перевірив:

викладач
Регіда Павло Геннадійович

Київ 2021

Основні теоретичні відомості:

Генетичні алгоритми служать, головним чином, для пошуку рішень в багатовимірних просторах пошуку.

Можна виділити наступні етапи генетичного алгоритму:

- (Початок циклу)
- Розмноження (схрещування)
- Мутація
- Обчислити значення цільової функції для всіх особин
- Формування нового покоління (селекція)
- Якщо виконуються умови зупинки, то (кінець циклу), інакше (початок циклу).

Розглянемо приклад реалізації алгоритму для знаходження цілих коренів діофантового рівняння $a+b+2c=15$.

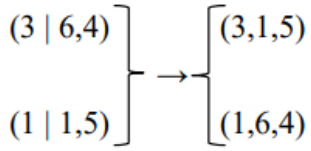
Згенеруємо початкову популяцію випадковим чином, але з дотриманням умови – усі згенеровані значення знаходяться у проміжку від одиниці до $y/2$, тобто на відрізку [1;8] (узагалі, границі випадкового генерування можна вибирати на свій розсуд):

(1,1,5); (2,3,1); (3,4,1); (3,6,4)

Отриманий генотип оцінюється за допомогою функції пристосованості (fitness function). Згенеровані значення підставляються у рівняння, після чого обраховується різниця отриманої правої частини з початковим y. Після цього рахується ймовірність вибору генотипу для ставання батьком – зворотня дельта ділиться на сумму сумарних дельт усіх генотипів.

$1+1+2\cdot5=12$	$\Delta=3$	$\frac{\frac{1}{3}}{\frac{27}{24}} = 0{,}7$
$2+3+2\cdot1=7$	$\Delta=8$	$\frac{\frac{1}{8}}{\frac{27}{24}} = 0{,}11$
$3+4+2\cdot1=9$	$\Delta=6$	$\frac{\frac{1}{6}}{\frac{27}{24}} = 0{,}15$
$3+6+2\cdot4=17$	$\Delta=2$	$\frac{\frac{1}{2}}{\frac{27}{24}} = 0{,}44$

Наступний етап включає в себе схрещування генотипів по методу кросоверу – у якості дітей виступають генотипи, отримані змішуванням коренів – частина йде від одного з батьків, частина від іншого, наприклад:



Лінія кросоверу може бути поставлена в будь-якому місці, кількість потомків також може вибиратися. Після отримання нових генотипів вони перевіряються функцією пристосованості та створюють власних потомків, тобто виконуються дії, описані вище.

Ітерації алгоритму відбуваються, поки один з генотипів не отримає $\Delta=0$, тобто його значення будуть розв'язками рівняння.

Завдання:

Налаштувати генетичний алгоритм для знаходження цілих коренів діофантового рівняння ax1+bx2+cx3+dx4=y. Розробити відповідний мобільний додаток і вивести отримані значення. Провести аналіз витрат часу на розрахунки.

Лістинг програми:

```
import 'package:flutter/material.dart';
import './Genetic.dart';

void main() {
  runApp(MaterialApp(
    home: HomePage(),
  ));
}

class HomePage extends StatefulWidget {
  @override
  _HomeAppState createState() => _HomeAppState();
}

class _HomeAppState extends State<HomePage> {
  int page = 0;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: getAppBar(),
      body: getBody(),
    );
  }

  Widget getBody() {
    List<Widget> pages = [
      GeneticPage(),
    ];
    return IndexedStack(
      index: page,
      children: pages,
```

```
    );
  }

  Widget getAppBar() {
    return AppBar(
      backgroundColor: Colors.teal,
      title: Container(
        child: Row(
          children: <Widget>[
            Text(
              "Лабораторна робота 3.3",
              style: TextStyle(fontSize: 20.0),
            ),
          ],
        ),
      ),
    );
  }
}

class GeneticPage extends StatefulWidget {
  GeneticPage({Key key}) : super(key: key);

  @override
  State<StatefulWidget> createState() {
    return __GeneticPageState();
  }
}

class __GeneticPageState extends State<GeneticPage> {
  double x1;
  double x2;
  double x3;
  double x4;
  int time;

  final aController = TextEditingController();
  final bController = TextEditingController();
  final cController = TextEditingController();
  final dController = TextEditingController();
  final yController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(body: getBody());
  }

  Widget getBody() {
    return Center(
      child: ListView
        (children: <Widget>[
        Padding(
          padding: const EdgeInsets.symmetric(vertical: 10.0),
        ),
        Text('ax1 + bx2 + cx3 + dx4 = y',
        style: TextStyle(
          fontSize: 20.0,
```

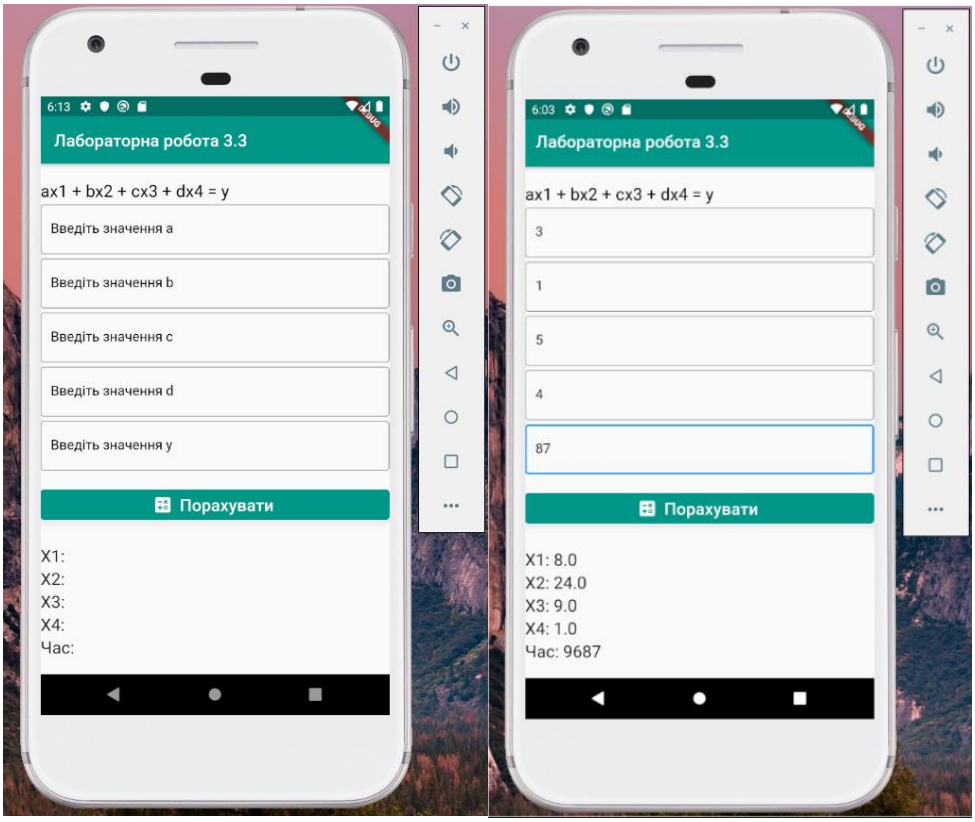
```
  fontFamily: 'Roboto',
  color: Colors.black,
),
),
SizedBox(
  height: 5.0,
),
Container(
  width: 300,
  child: TextField(
    controller: aController,
    decoration: InputDecoration(
      hintText: 'Введіть значення a',
      hintStyle: TextStyle(color: Colors.black),
      border: OutlineInputBorder(),
    ),
  ),
),
SizedBox(
  height: 5.0,
),
Container(
  width: 300,
  child: TextField(
    controller: bController,
    decoration: InputDecoration(
      hintText: 'Введіть значення b',
      hintStyle: TextStyle(color: Colors.black),
      border: OutlineInputBorder(),
    ),
  ),
),
SizedBox(
  height: 5.0,
),
Container(
  width: 300,
  child: TextField(
    controller: cController,
    decoration: InputDecoration(
      hintText: 'Введіть значення c',
      hintStyle: TextStyle(color: Colors.black),
      border: OutlineInputBorder(),
    ),
  ),
),
SizedBox(
  height: 5.0,
),
Container(
  width: 300,
  child: TextField(
    controller: dController,
    decoration: InputDecoration(
      hintText: 'Введіть значення d',
      hintStyle: TextStyle(color: Colors.black),
      border: OutlineInputBorder(),
```

```
    ),
  ),
),
SizedBox(
  height: 5.0,
),
Container(
  width: 300,
  child: TextField(
    controller: yController,
    decoration: InputDecoration(
      hintText: 'Введіть значення y',
      hintStyle: TextStyle(color: Colors.black),
      border: OutlineInputBorder(),
    ),
  ),
),
Padding(
  padding: const EdgeInsets.symmetric(vertical: 16.0),
  child: ElevatedButton.icon(
    label: Text('Порахувати',
    style: new TextStyle(
      fontSize: 20.0,
    ),
    ),
    icon: Icon(Icons.calculate),
    style: ElevatedButton.styleFrom(
      primary: Colors.teal,
      shadowColor: Colors.black26,
      elevation: 5,
    ),
    onPressed: () {
      var a = int.parse(aController.text);
      var b = int.parse(bController.text);
      var c = int.parse(cController.text);
      var d = int.parse(dController.text);
      var y = int.parse(yController.text);

      var startDate = DateTime
          .now()
          .microsecondsSinceEpoch;
      var gen = new Genetics();
      var result = gen.diophantEquation(
          equation: [a, b, c, d, y], populationSize: 10);
      print(result.length);
      if (result.length == 4) {
        setState(() {
          x1 = result[0].toDouble();
          x2 = result[1].toDouble();
          x3 = result[2].toDouble();
          x4 = result[3].toDouble();
          time = DateTime
              .now()
              .microsecondsSinceEpoch - startDate;
        });
      }
    }
```

```
        ),
      ),
      SizedBox(
        height: 5.0,
      ),
      Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Padding(
            padding: const EdgeInsets.only(top: 4.0),
            child:
            Text('X1: ${this.x1 ?? ' '}', style: TextStyle(fontSize: 20)),
          ),
          Padding(
            padding: const EdgeInsets.only(top: 4.0),
            child:
            Text('X2: ${this.x2 ?? ' '}', style: TextStyle(fontSize: 20)),
          ),
          Padding(
            padding: const EdgeInsets.only(top: 4.0),
            child:
            Text('X3: ${this.x3 ?? ' '}', style: TextStyle(fontSize: 20)),
          ),
          Padding(
            padding: const EdgeInsets.only(top: 4.0),
            child:
            Text('X4: ${this.x4 ?? ' '}', style: TextStyle(fontSize: 20)),
          ),
          Padding(
            padding: const EdgeInsets.only(top: 4.0),
            child: Text('Час: ${this.time ?? ' '}',
                style: TextStyle(fontSize: 20)),
          ),
        ],
      ),
    ]),
  );
 }
}
```

Результат виконання:



Висновки:

На даній лабораторній роботі було здійснено ознайомлення з принципами реалізації генетичного алгоритму, вивчено та досліджено особливості даного алгоритму з використанням засобів моделювання і сучасних програмних оболонок.