



**Faculty of Engineering and Technology
Electrical and Computer Engineering
Department**

**ARTIFICIAL INTELLIGENCE, ENCS3340
Project #1**

**By: Osama Rihami
Number: 1190560
Section: 2**

Instructor: Dr. Aziz Qaroush

Date: 3/12/2021

Contents:

Abstract
Problem Formalization
Code Explanation: <ul style="list-style-type: none">1. Project Class2. Group Class3. Chromosome Class4. Population Class5. Genetics Class6. Driver Class7. Controller Class
Compilation
Extra

Abstract:

This project aims to use Genetic algorithms to build a “Graduation projects distribution” program.

The program will read inputs from two files:

- Groups selections (Excel File)
- Projects (PDF File).

Then it will assign a project for each group so that:

- The assignment must be closed as much as possible to the order of selection.
- The minimum number of conflicts that can occur between groups selections.

Problem Formalization:

For this project, I used Java language + scene builder.

In genetics algorithm, the chromosome represents a solution,

The solution here is an array of integers, the index of the array represents the group number (id), and the value of that index represents the topic choice for that group.

A population has many chromosomes sorted by fitness.

The fitness of each chromosome = the number of conflicts between group choices.

The population sorts chromosomes ascending by their fitness value.

If two chromosomes have the same fitness value, we compare them by the benefit value.

The benefit of each chromosome = for each index in the array

if it was the first choice add 3

if it was the second choice add 2

if it was the third choice add 1.

We sort them descending.

Code Explanation:

1. Project Class:

Each Project has a project number (pid),
project description that has the name and description of that project.
we use this class to define Project instances.

2. Group Class:

Each Group has id,
Student1, student2, and student3,
3 choices: option1, option2 and option3.
we use this class to define Group instances.

3. Chromosome Class:

A chromosome represents a solution,
Each one has an array (int[]) of genes, the index of the array represents the
group number (id), and the value of that index represents the topic choice
for that group.
Array list of Groups.
Fitness double, Benefit int.

initialize function (initalizeChrom): for every group in the array genes
choose a random value between the three options.
Calculate fitness (calcFitness), calculate benefit (calcBenefit),
Compare function that follows the order we explained before.

4. Population Class:

Each population has an array of chromosomes,
and Array list of Groups.

initialize function (initalizePopul): for every index in the array define a
chromosome and initialize it, then calculate its fitness and benefit.
At the end sort the chromosomes in the array by the compare function we
define before in chromosome class.

Sort by fitness function (sortByFitness).

5. Genetics Class:

This class represents the algorithm,
It has Static final integers:

- Population_size: number of chromosomes in the population.
- Num_of_best_chromosomes: number of chromosomes in the population that will not be changed during reproduction.
- Mutation_Rate: the rate to accept the mutation.
- Tournament_Size: the number of chromosomes selected during tournament selection.

and Array list of Groups.

Functions:

- Population reproduction (Reproduction): apply crossover and mutation over the population.
- Population crossover(populationCrossover): the first chromosomes are not changed; the remaining chromosomes are replaced by a crossover between two tournament selection chromosomes.
- tournament selection(tournamentSelection): selects number of chromosomes randomly from the old population and return the best one of them.
- Chromosome's crossover(chromosomeCrossover): create a new chromosome and for its genes array it selects each value randomly between the two old chromosomes.
- Population mutation (populationMutation): the first chromosomes are not changed, the remaining chromosomes we apply chromosome mutation on them.
- Chromosome's mutation (chromosomeMutation): create new chromosome and for each value in the genes array if (math.random) was below the mutation rate we choose the value randomly between the options for that group, if it was above the rate, it has the same value for the old chromosome.

6. Driver Class:

This class just starts the interface from the “layout.fxml” file.

7. Controller Class:

This is the main class.

First, it calls two functions:

Read excel file(readExcelFile): it reads the excel file and for each row, it defines a group instance and adds it to an array list, at the end, it returns the array list.

Read PDF file(readPdfFile): it reads the pdf file and for each row, it defines a Project instance and adds it to an array list, at the end, it returns the array list.

Is numeric (isNumeric): receive a string and check if it's an integer or not.

We initialize a new population and iterations counter.

We define the labels and text fields ... etc. for Javafx.

Button functions:

- Show projects(showProjects): reset the scroll pane at the top (same for all buttons), for each project in the arraylist print the value on the label.
- Show Groups(showGroups): for each group in the arraylist print the value on the label.
- Generate Solutions (generateSolutions): every time we click the button we define a new population, reset the counter, and while (the fitness value is not zero and the counter didn't reach the limit) we apply reproduction on the population and increase the counter, In the end, for each Chromosome in the population print its values, fitness, and benefit on the label.
- Show Best Solution (showBestSolution): we print the best chromosome values and their fitness on the label.

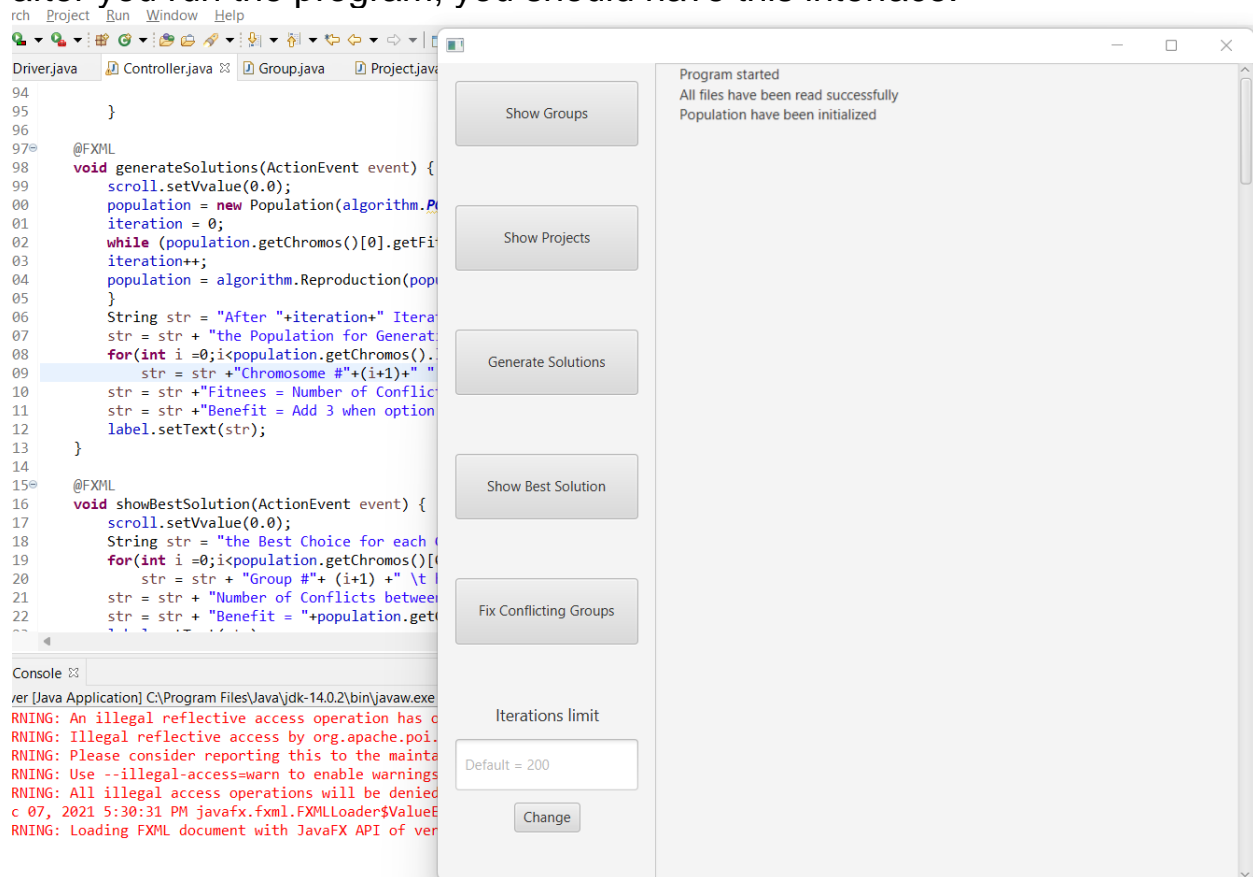
Note: other functions are described in EXTRA.

Compilation

Note that before running the program you need to add these libraries to the project:

- All JavaFX and FXML libraries.
- commons-collections4-4.1
- pdfbox-app-2.0.4
- poi-3.17
- poi-ooxml-3.17
- poi-ooxml-schemas-3.17
- xmlbeans-2.6.0

after you run the program, you should have this interface:



After you press Show Groups:

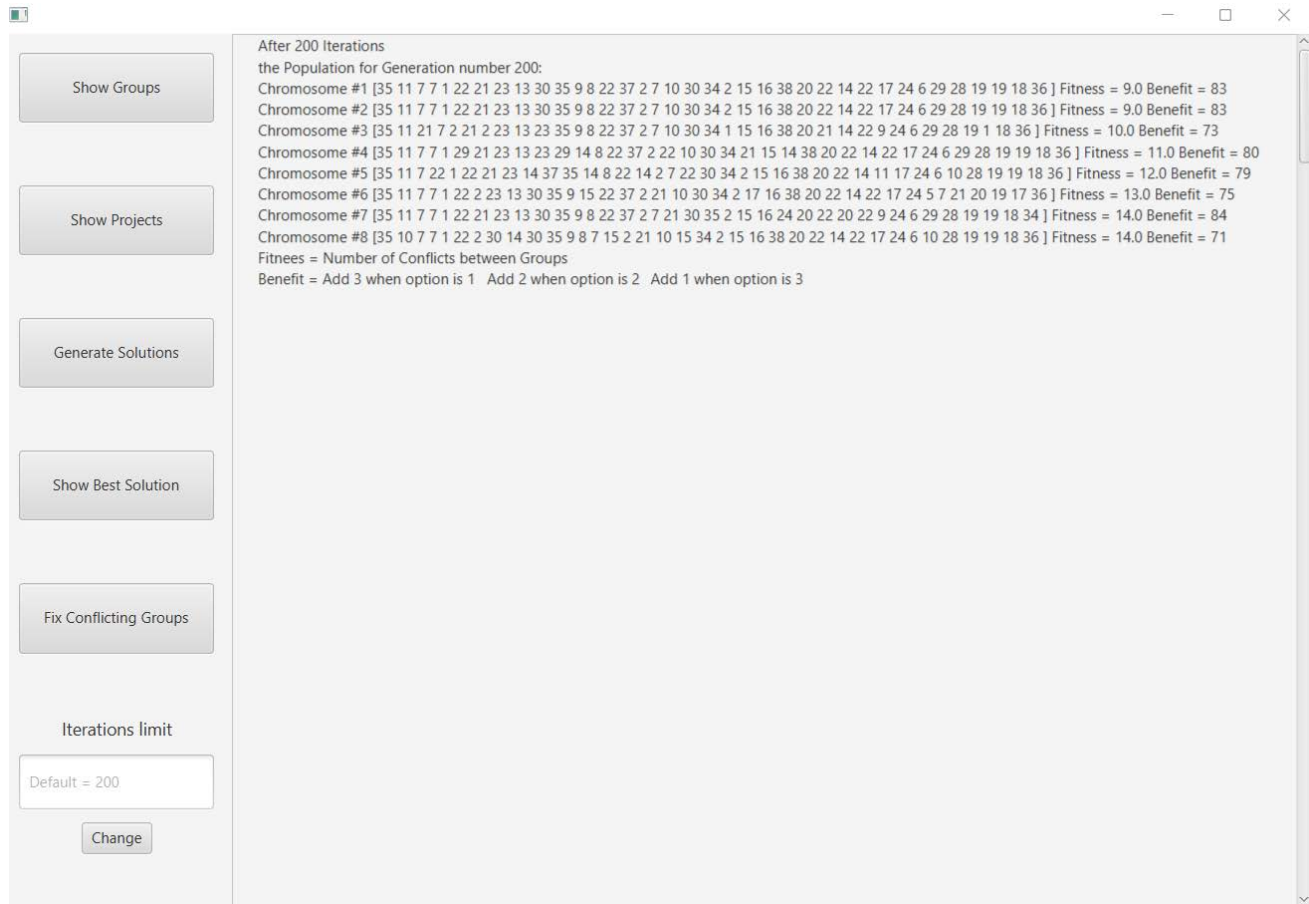
ID	Names	First Choice	Second Choice	Third Choice
Group ID: 1	Student Names: Tari, Issa, Jihad	Choices : 35	7	22
Group ID: 2	Student Names: Kheiria, Dina, Baraa	Choices : 11	21	10
Group ID: 3	Student Names: Dana, Lara, Mayar	Choices : 7	21	22
Group ID: 4	Student Names: Ameenah, Amr, Bashar	Choices : 7	22	35
Group ID: 5	Student Names: Saba, Safa, Mahmoud	Choices : 1	2	22
Group ID: 6	Student Names: Mohammad, Abdallah,	Choices : 22	21	29
Group ID: 7	Student Names: Najla, Yara, Shayma	Choices : 22	21	2
Group ID: 8	Student Names: Mohammad, Hussam, Mohammad	Choices : 9	23	30
Group ID: 9	Student Names: Qutayba, Abd, Hammam	Choices : 14	17	13
Group ID: 10	Student Names: Akram, Ahmad, Omar	Choices : 30	23	37
Group ID: 11	Student Names: Nazeeh, Noor, Qamar	Choices : 29	35	34
Group ID: 12	Student Names: Mohammad, Mariam, Omaima	Choices : 15	14	9
Group ID: 13	Student Names: Aminah, Mean, Haleema	Choices : 8	15	22
Group ID: 14	Student Names: Zahra, Tamara, Dalia	Choices : 22	21	7
Group ID: 15	Student Names: Malak, Haneen, Jenan	Choices : 15	14	37
Group ID: 16	Student Names: Jumanah, Nemah,	Choices : 2	7	7
Group ID: 17	Student Names: Laila, Samira, Abd	Choices : 7	22	21
Group ID: 18	Student Names: Farah, Najat, Marah	Choices : 21	10	22
Group ID: 19	Student Names: Hala, Sdeen, Jyana	Choices : 30	14	15
Group ID: 20	Student Names: Duha, Leena, Raghad	Choices : 29	34	35
Group ID: 21	Student Names: Haneen, Noor, Marah	Choices : 1	2	21
Group ID: 22	Student Names: Amneh, Tareq, Mahmoud	Choices : 17	15	8
Group ID: 23	Student Names: Issam, Mohammad, Ahmad	Choices : 14	16	15
Group ID: 24	Student Names: Yazan, Aziz, Dana	Choices : 23	24	38
Group ID: 25	Student Names: Abd, Muath, Suhaib	Choices : 20	14	19
Group ID: 26	Student Names: Mohammad, Besan, Abd	Choices : 22	21	29
Group ID: 27	Student Names: Sara, Israa, Raghad	Choices : 14	20	15
Group ID: 28	Student Names: Moayad, Aseel, Yazan	Choices : 22	29	11
Group ID: 29	Student Names: Huthaifa, Miqdaa, Mohammad	Choices : 17	9	30
Group ID: 30	Student Names: Tareq, Mahmoud,	Choices : 23	24	30
Group ID: 31	Student Names: Tatyana, gorge, Abd	Choices : 7	6	5
Group ID: 32	Student Names: Yara, Wafa, Sundos	Choices : 7	29	10
Group ID: 33	Student Names: Kamel, Kareen, Sara	Choices : 29	28	21
Group ID: 34	Student Names: Ahmad, Batool, Obada	Choices : 19	20	1
Group ID: 35	Student Names: Ahmad, Batool, Obada	Choices : 19	20	1
Group ID: 36	Student Names: Raya, Deena, Ragahd	Choices : 17	18	15
Group ID: 37	Student Names: Rana, Lama, Yaqeen	Choices : 7	34	36

After you press Show Projects:

The screenshot shows a web application interface with a sidebar on the left and a main content area on the right. The sidebar contains several buttons: 'Show Groups', 'Show Projects', 'Generate Solutions', 'Show Best Solution', 'Fix Conflicting Groups', and 'Iterations limit'. The 'Iterations limit' section has a text input field with 'Default = 200' and a 'Change' button. The main content area displays a table with two columns: 'Project Number' and 'Description'. The table lists seven projects, each with a brief description of its focus and goals.

Project Number	Description
Project # 1	Dr. Abualseoud Hanani: Audio Classification. Audio classification is among the most in-demand speech processing projects. As deep learning focuses on building a network that resembles a human mind, sound recognition is also essential. While image classification has become much advanced and widespread, audio classification is still a relatively new concept. The main idea of this project is to build a deep learning model that can accurately classify a given short duration sound (e.g. 10 seconds) into one of a predefined sound events such as music, human speech, vehicle, boom, crumpling, etc. Audio classification can be helpful in query based multimedia retrieval, acoustic scene analysis, and bio-diversity monitoring. For this project, we can work on Google AudioSet, which is a vast collection of labeled audio that they collected from YouTube videos. They all are 10-seconds long and are incredibly varied. We can use the audio files present in AudioSet to train and test our models. They are correctly labeled, so working with them is relatively more straightforward. There are currently 632 audio event classes and more than two million sound clips present in AudioSet (https://research.google.com/audioset/).
Project # 2	Dr. Abualseoud Hanani: Machine learning-based system for evaluating short stories understanding for Arabic speaking children. The main idea of this project is to develop a system that is able to evaluate the correctness of response of children (learners) for a given question upon a specific topic automatically. A set of carefully-selected multimedia prompts; text, picture, audio, or video describing specific topics, will be displayed to the child, and then he/she is asked to answer specific questions upon what he/she has seen and heard in his/her language. The learner response represents how well he/she understands the topic and his/her ability to formulate the ideas and the strength in the language. All of these skills can be analyzed and evaluated by the proposed automatic system in terms of the meaning and maybe the language. In addition to the general evaluation, the system gives the child a feedback about his/her weakness and strengths. The proposed system is a multi-disciplinary system which uses different technology in computer science such as Natural Language Processing (NLP), speech processing and machine learning. All prompts and responses will be in Arabic language. To make it clearer, to encourage the child to go on, the prompts could be an Arabic learning story (displayed in a form of text or audio or pictures) which talks about a specific idea. It will be divided into parts and after each part it will display a question to the child upon what's previously displayed to him/her. The child then is asked to answer the question from the story by speech in Arabic language to be evaluated to give a judgment of correct or incorrect answer.
Project # 3	Dr. Khader Mohammad: EEG brain scans model. EEG brain scans have been introduced as a way to detect emotions which opens doors beyond the medical field. Different techniques used in trying to classify Emotions (Boring, Calm, Horror, Funny) and authentication based on EEG signals. All sources of EEG Signal will be studied and analyzed from channel it was recorded in this research a Human emotions models based on EEG Signals will be build using python and other needed tools.
Project # 4	Dr. Khader Mohammad: A Hypered approach to verify ECG emotion extracted from ECG signal and face recognition. In this project you will work into emotion generation from brain and classify the effect of environment conditions on EEG Signals that effect emotions and authentications. Will use face recognition to verify the matching. This is project is a continuation of previous project where emotion is extracted from brain ECG signal with 85% accuracy. What we need is to raise the accuracy % and compare to the face emotion recognoin. We used python and machine learning to chive the goals.
Project # 5	Dr. Ahmad Alsadeh: Automatic Website Vulnerability Scanner Application. The purpose of this application is to build an automatic scanner for discovering, analyzing, testing, identifying, and reporting of website vulnerabilities of a target website.
Project # 6	Dr. Ahmad Alsadeh: IPv6 SEND Application for Mobile Devices. IPv6 SECure Neighbor Discovery (SEND) is an extension to countermeasure the security threats in IPv6 Neighbor Discovery Protocol (NDP). SEND provides address ownership proof, message protection, and router authorization capabilities. The current lack of robust implementations of SEND at the end user devices limits its deployment and leave the door open for the attacks against NDP. The main goal of this project is to provide an open-source implementation of SEND for mobile devices.
Project # 7	Dr. Mohammed Hussein: CarPale Application: Buy-Sell New & Used Cars, Prices & Offers in Palestine.

After you press Generate Solutions:



After it enters the while loop this is the output.

every time you click the button you will have different output.

Note that it shows the population for the current generation, so every time you change the population you will see the last one.

After you press Show Best Solution:

the Best Choice for each Group is :

Group #1	has Project # :35
Group #2	has Project # :11
Group #3	has Project # :7
Group #4	has Project # :7
Group #5	has Project # :1
Group #6	has Project # :22
Group #7	has Project # :21
Group #8	has Project # :23
Group #9	has Project # :13
Group #10	has Project # :30
Group #11	has Project # :35
Group #12	has Project # :9
Group #13	has Project # :8
Group #14	has Project # :22
Group #15	has Project # :37
Group #16	has Project # :2
Group #17	has Project # :7
Group #18	has Project # :10
Group #19	has Project # :30
Group #20	has Project # :34
Group #21	has Project # :2
Group #22	has Project # :15
Group #23	has Project # :16
Group #24	has Project # :38
Group #25	has Project # :20
Group #26	has Project # :22
Group #27	has Project # :14
Group #28	has Project # :22
Group #29	has Project # :17
Group #30	has Project # :24
Group #31	has Project # :6
Group #32	has Project # :29
Group #33	has Project # :28
Group #34	has Project # :19
Group #35	has Project # :19
Group #36	has Project # :18
Group #37	has Project # :36

Number of Conflicts between Groups = 9.0
Benefit = 83

These are the values for the best chromosome in our population.

Extra

1) Fix Conflicting Groups:

It shows you the group choices that are chosen before by another group (That creates conflict).

Also, it shows you the project numbers that are not chosen by anyone (Still available).

Then it automatically assigns both together so now the solution won't have any conflicts.

Groups IDs that are conflicting with other groups:
11 4 17 14 19 26 21 28 35

Project Numbers that are still available:
3 4 5 12 25 26 27 31 32 33

Group #11	is assigned for Project # :3
Group #4	is assigned for Project # :4
Group #17	is assigned for Project # :5
Group #14	is assigned for Project # :12
Group #19	is assigned for Project # :25
Group #26	is assigned for Project # :26
Group #21	is assigned for Project # :27
Group #28	is assigned for Project # :31
Group #35	is assigned for Project # :32

The solution after we press (Fix Conflicting Groups):

the Best Choice for each Group is :

Group #1	has Project # :35
Group #2	has Project # :11
Group #3	has Project # :7
Group #4	has Project # :4
Group #5	has Project # :1
Group #6	has Project # :22
Group #7	has Project # :21
Group #8	has Project # :23
Group #9	has Project # :13
Group #10	has Project # :30
Group #11	has Project # :3
Group #12	has Project # :9
Group #13	has Project # :8
Group #14	has Project # :12
Group #15	has Project # :37
Group #16	has Project # :2
Group #17	has Project # :5
Group #18	has Project # :10
Group #19	has Project # :25
Group #20	has Project # :34
Group #21	has Project # :27
Group #22	has Project # :15
Group #23	has Project # :16
Group #24	has Project # :38
Group #25	has Project # :20
Group #26	has Project # :26
Group #27	has Project # :14
Group #28	has Project # :31
Group #29	has Project # :17
Group #30	has Project # :24
Group #31	has Project # :6
Group #32	has Project # :29
Group #33	has Project # :28
Group #34	has Project # :19
Group #35	has Project # :32
Group #36	has Project # :18
Group #37	has Project # :36

Number of Conflicts between Groups = 0.0
Benefit = 58

Note that the benefit value has decreased because we assigned projects for groups that are not part of their choices.

2) Change iterations limit:

The user also can change the number of iterations that reproduce the population with the value he wants.

The screenshot shows a software interface with a left sidebar and a main content area. The sidebar contains several buttons: 'Show Groups', 'Show Projects', 'Show Population', 'Find Best Choices', and 'Find Conflicting Groups'. Below these buttons is a section labeled 'Iterations limit' which includes a text input field containing 'abc' and a 'Change' button. Below the 'Change' button, the text 'Invalid Input' is displayed. The main content area on the right displays two lists of numbers: 'Groups IDs that are conflicting with other groups:' followed by '4 16 14 26 17 32 29 28 35', and 'Project Numbers that are still available:' followed by '3 4 5 12 25 26 27 31 32 33'.

Show Groups

Show Projects

Show Population

Find Best Choices

Find Conflicting Groups

Iterations limit

abc

Change

Invalid Input

Groups IDs that are conflicting with other groups:
4 16 14 26 17 32 29 28 35

Project Numbers that are still available:
3 4 5 12 25 26 27 31 32 33