Faculty of Computers &
Artificial Intelligence

Helwan University

Medicalinformatics
Department

Project *11*

"Vehicle Route problem"

A report project submitted
in partial fulfillment of the requirements for passing
the 1st semester 2021-2022 evaluation

In

Artificial Intelligence

By

Aya Ahmed Elsayed (20208052)

Ameena Mustafa Saad (20208038)

Basma Mohammed Fath (20208070)

Nada Essam Mohammed (20208254)

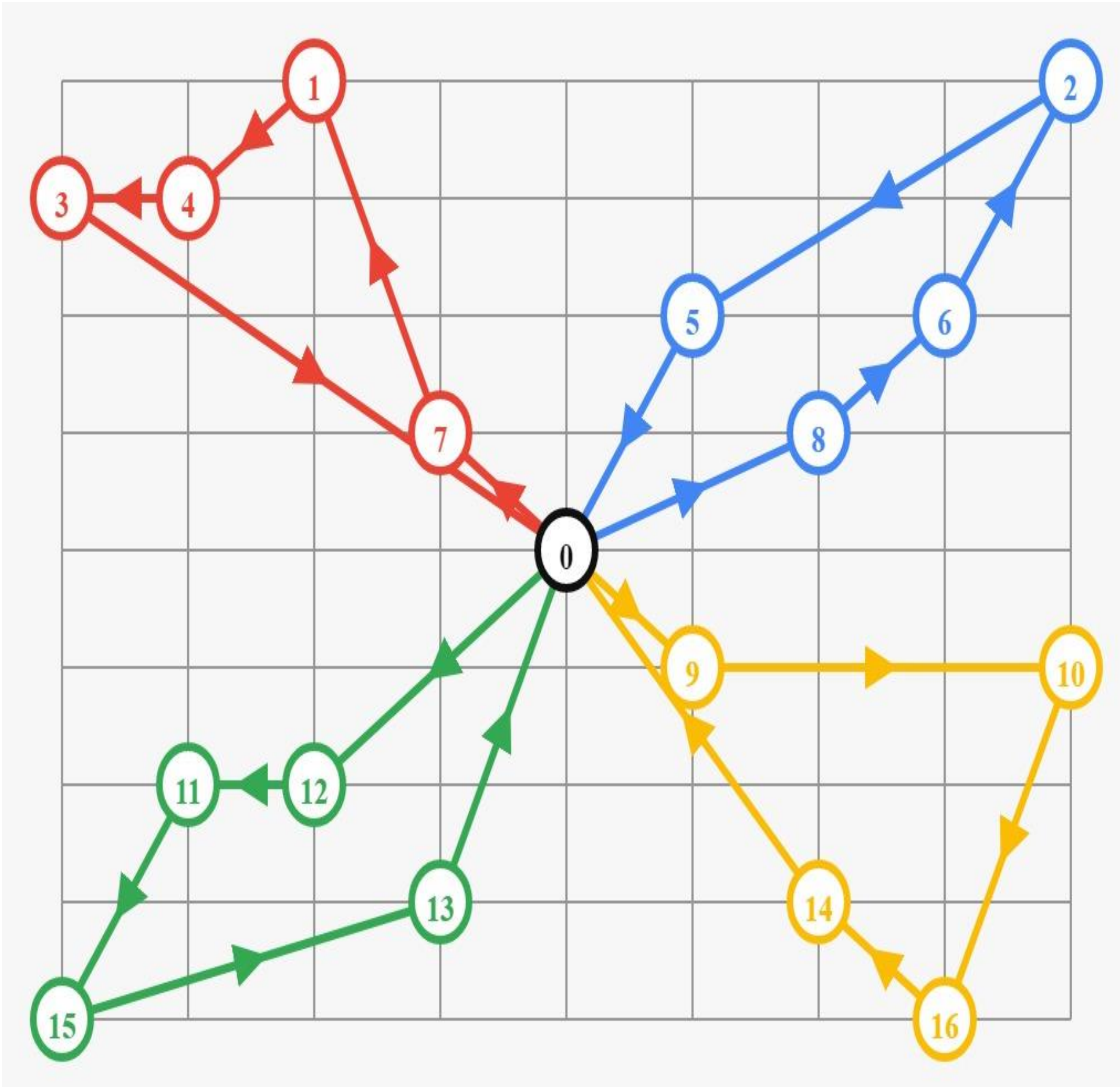
Yassmin Safwat Abdulrahman (20208293)

Supervised by

Dr. Amr S. Ghoneim

Desember 2021

Table of Contents


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

1. Abstract

Transportation planning has been established as a key topic in the literature and social production practices. An increasing number of researchers are studying vehicle routing problems.

(VRPs) and their variants considering real-life applications and scenarios. Furthermore, with the rapid growth in the processing speed and memory capacity of computers, various algorithms can be used to solve increasingly complex instances of VRPs.

In this study ,We reviewed recent research according to models and solutions, and divided models into three categories of customerrelated, vehicle-related, and depot-related models. We classified solution algorithms into exact, heuristic, and meta-heuristic algorithms.

We enabled future researchers to find our report easily and provide readers with recent trends and solution methodologies in the field of VRPs and some well-known variants.

2. Introduction

2.1 Project idea and overview

Problems related to the distribution of goods between warehouses and customers are generally considered as vehicle routing problems (VRPs).

VRPs are some of the most critical challenges faced by logistics companies, an increasing amount of research is focusing on VRPs.

Solving VRPs is computationally expensive and categorized ,because real world problems involve complex constraints such as time windows, time-dependent travel times (reflecting traffic congestion), multiple depots, and heterogeneous fleets.

These features introduce significant complexity and have dramatically evolved the VRP research landscape.

The processing speed and memory capacity of computers has grown rapidly, enabling the processing of increasingly complex instances of VRPs and widespread application of logistics distribution scenarios.

The number of VRP solution methods introduced in the academic literature has grown rapidly over the past few decades.

The VRP family can be considered as two combinatorial senses:

(1) the number of possible solutions, which grow exponentially with computer science and algorithm design.

(2) the number of conceivable problem variants, which also grow exponentially with avariety of problem attributes .

2.2 Applications in market for multi depot(VRP)

(1) Upper Rout problem

(2) Tour Solver

(3) Amazon

(4) Jumia

(5) OLX

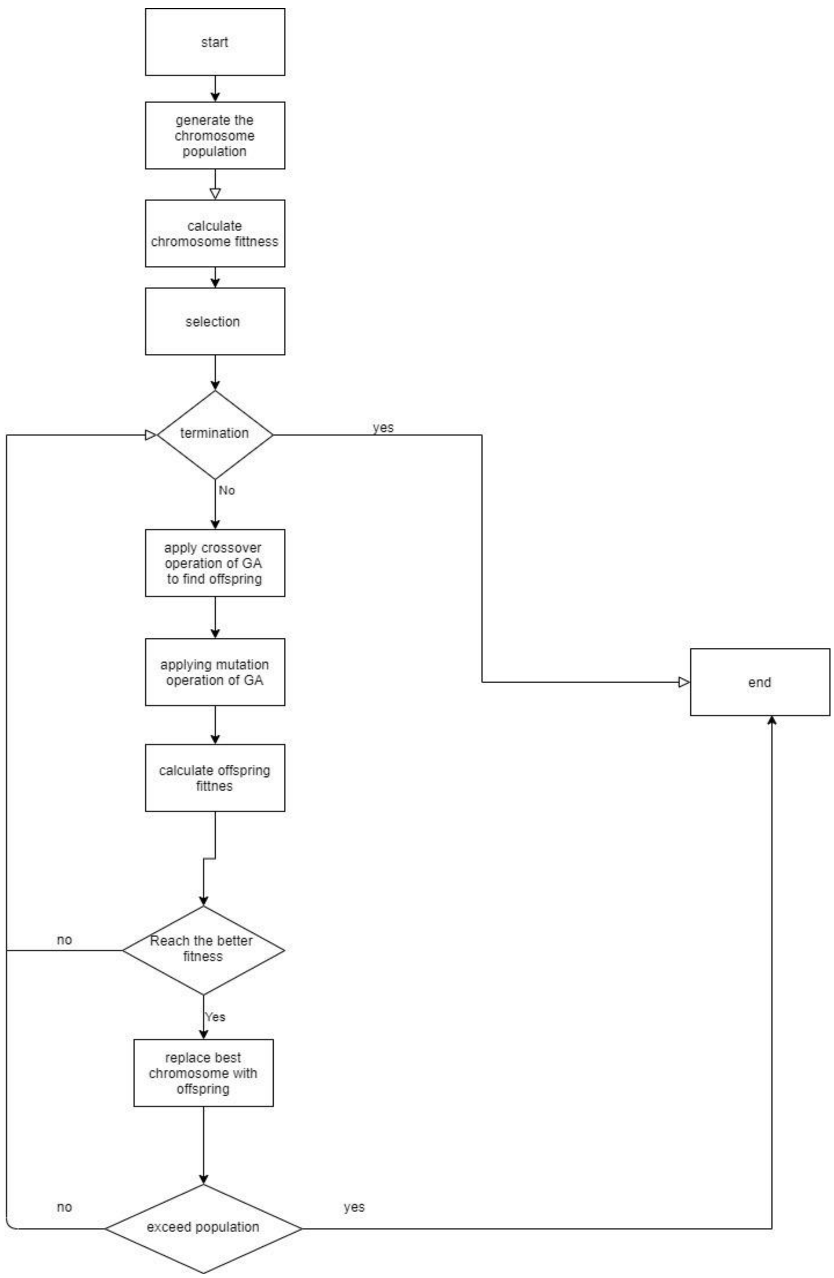
(6) Talabat

(7) Souq

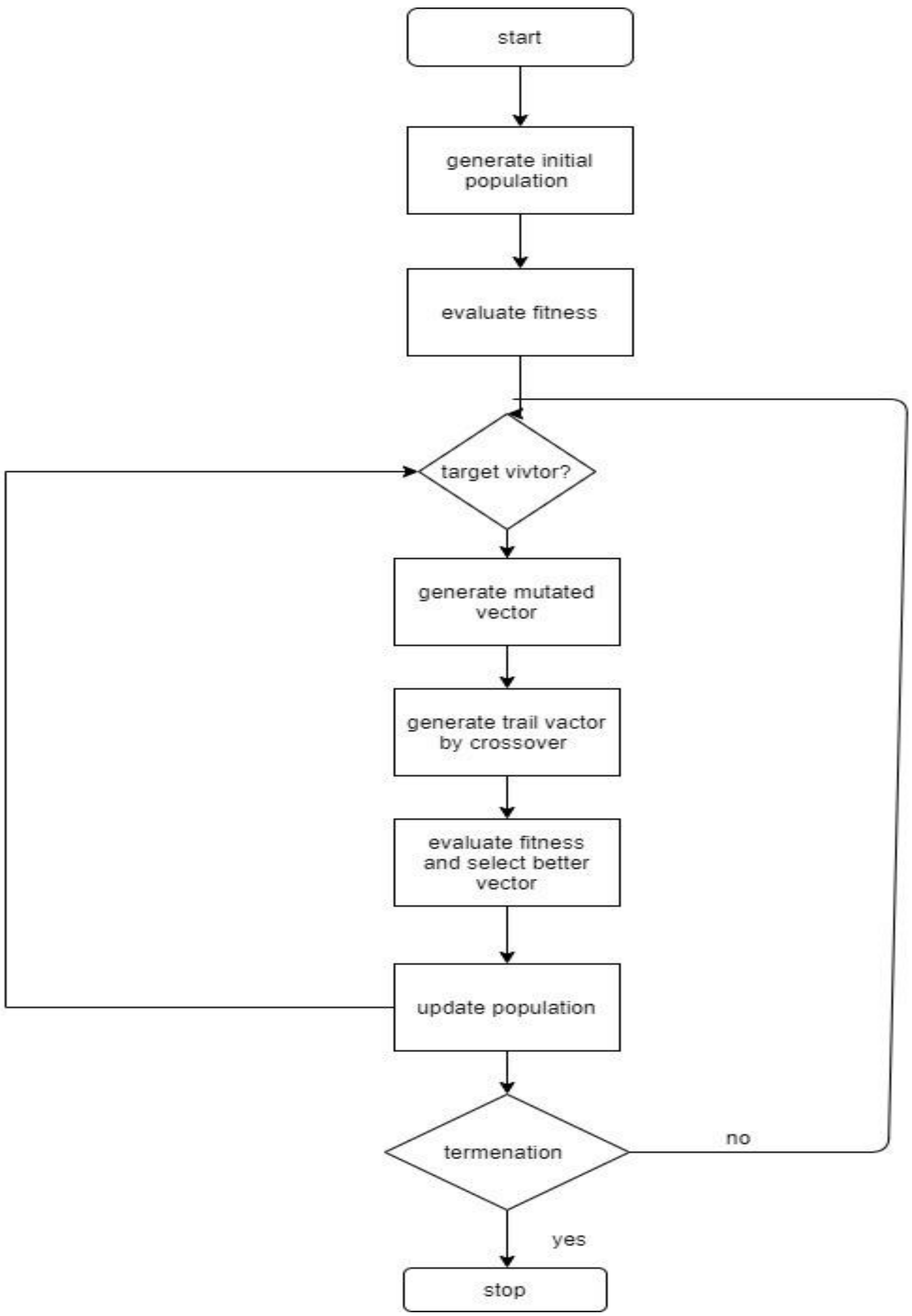
3. Proposed solution and Dataset

3.1 Main functionalities and features

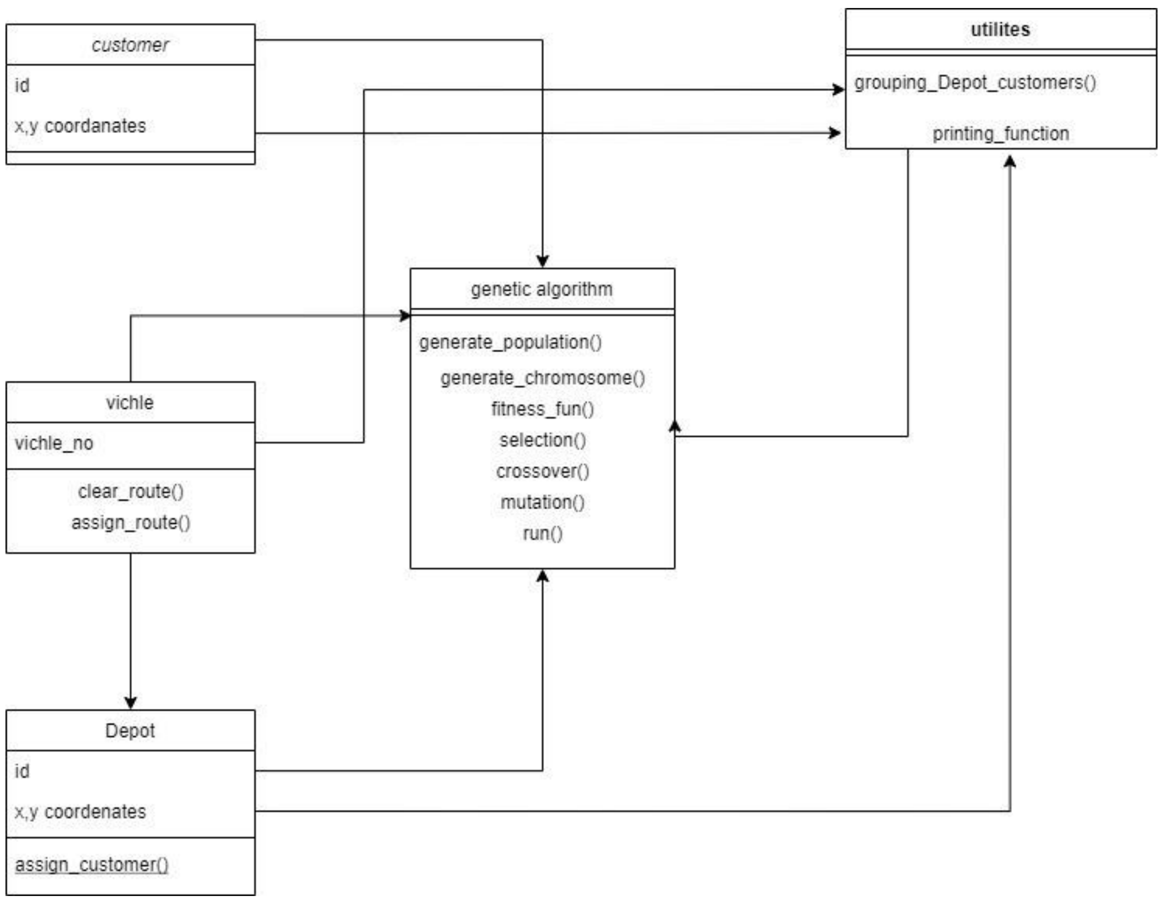
-Flow chart for genetic algorithms



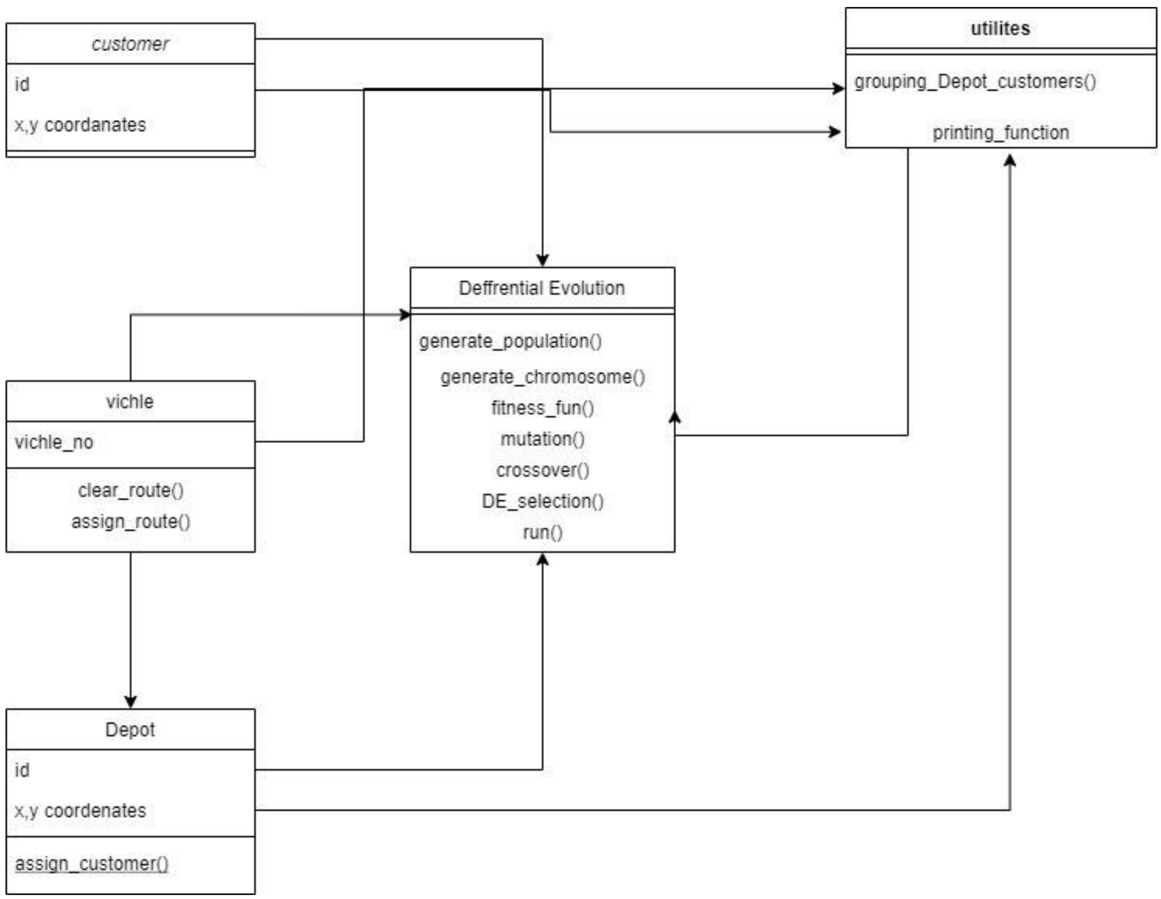
-Flow chart for differential evolution



-Class diagram of Genetic Algorithm



-class diagram for differential evolution



3.2 Methods and features

-Development platform

-Tools : Pycharm

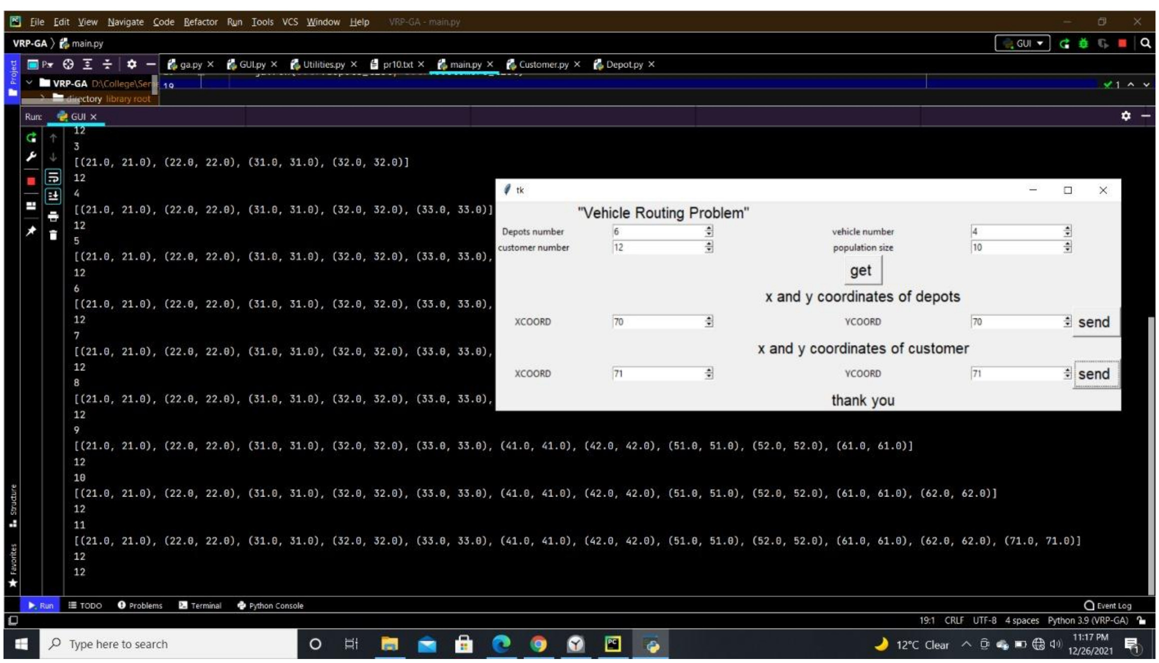
-Libraries : - numpy

- math

- random

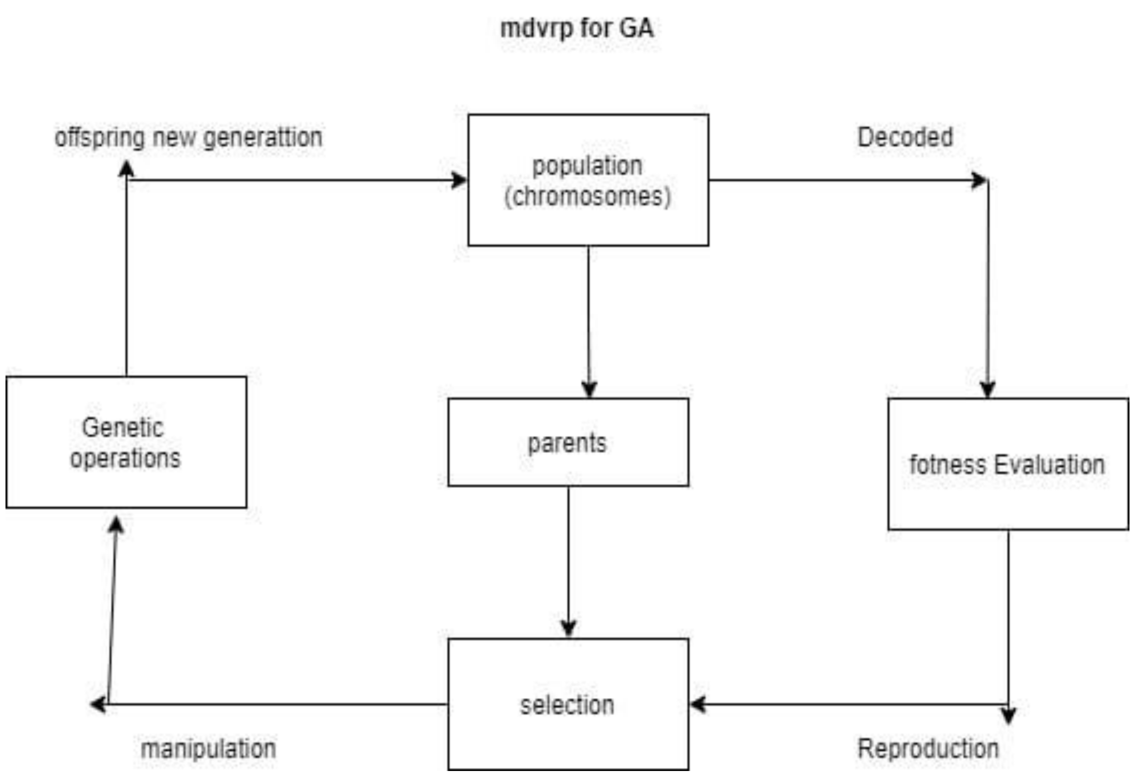
- pandas

-GUI

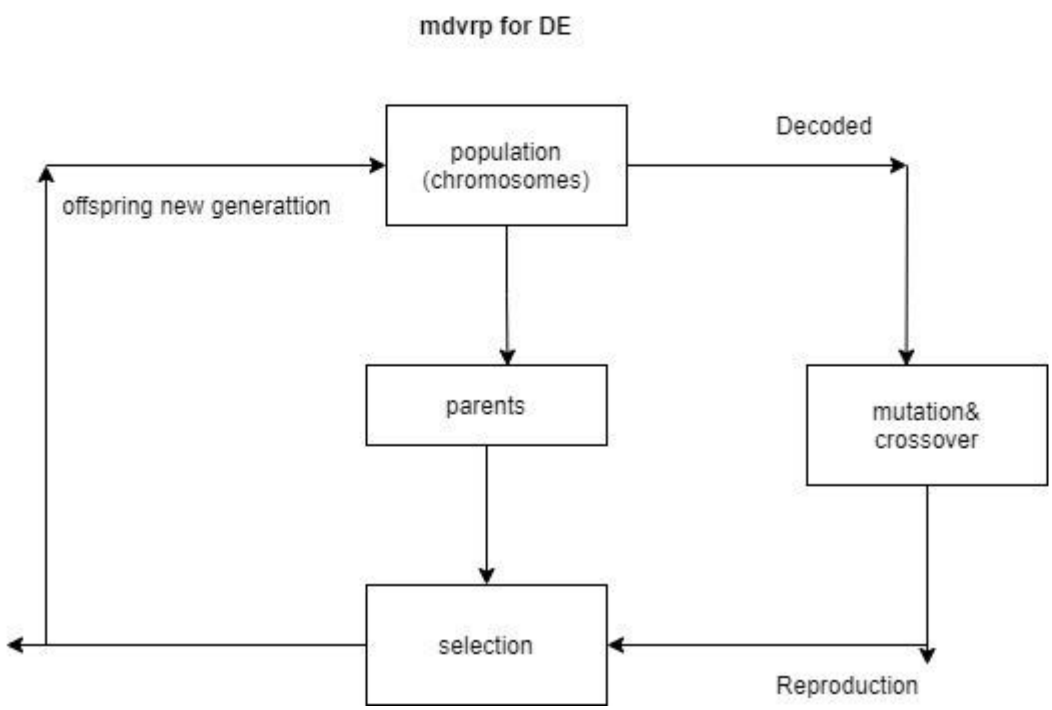


4. Applied Algorithms

4.1 details of genetic algorithm



4.2 details of differential evolution

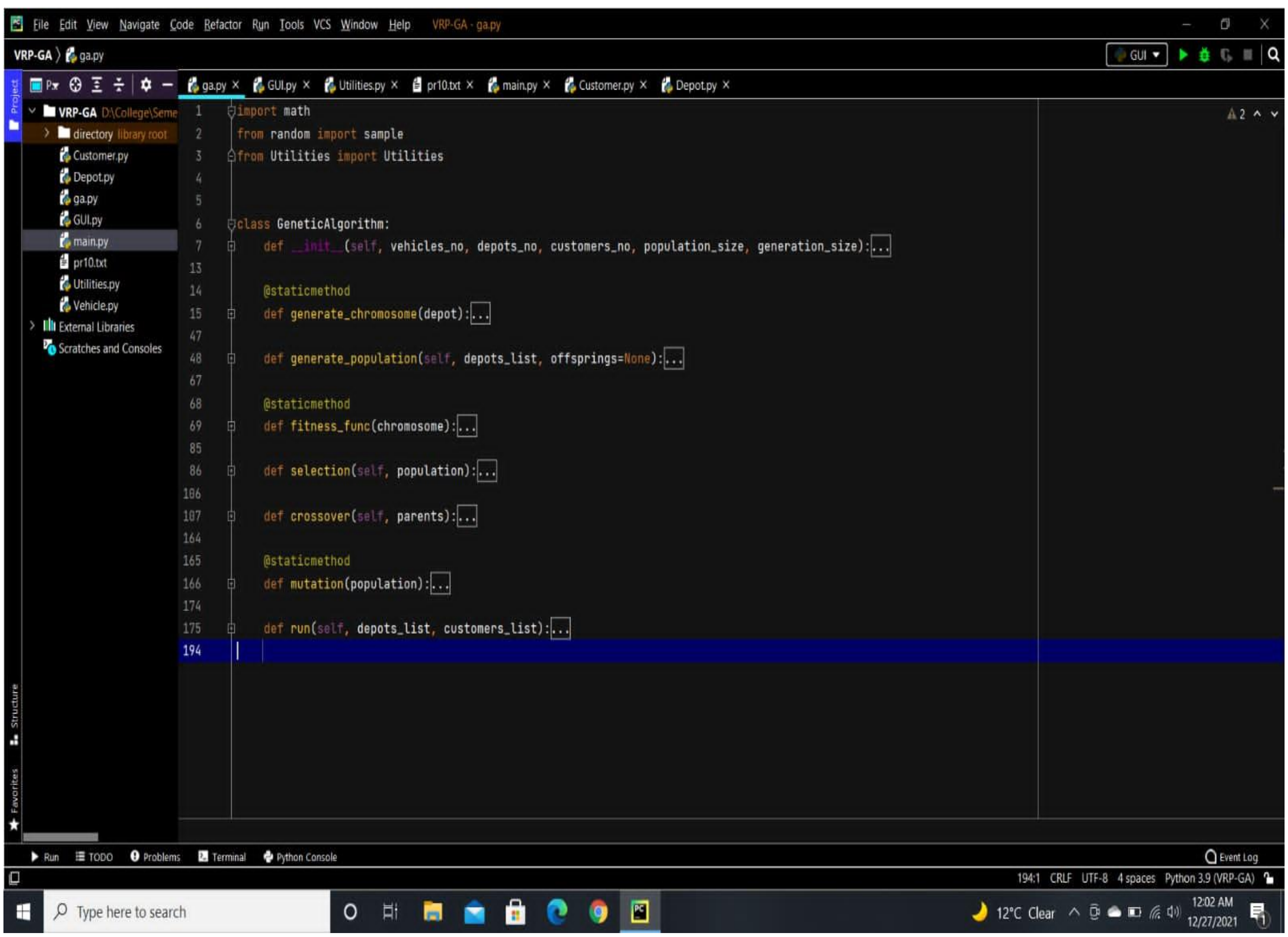


5. Expriments and results :

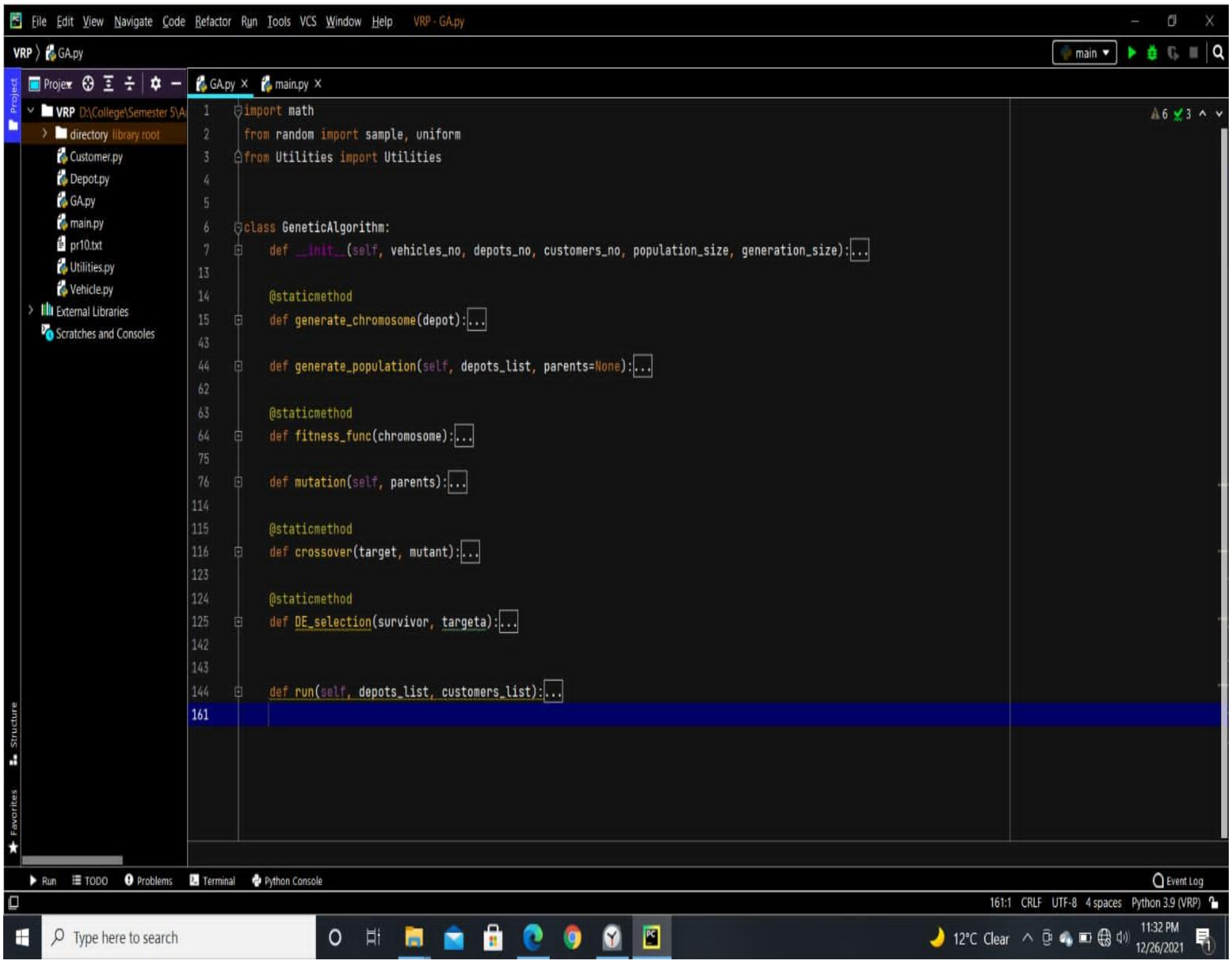
5.1 Testing and Results

Testing

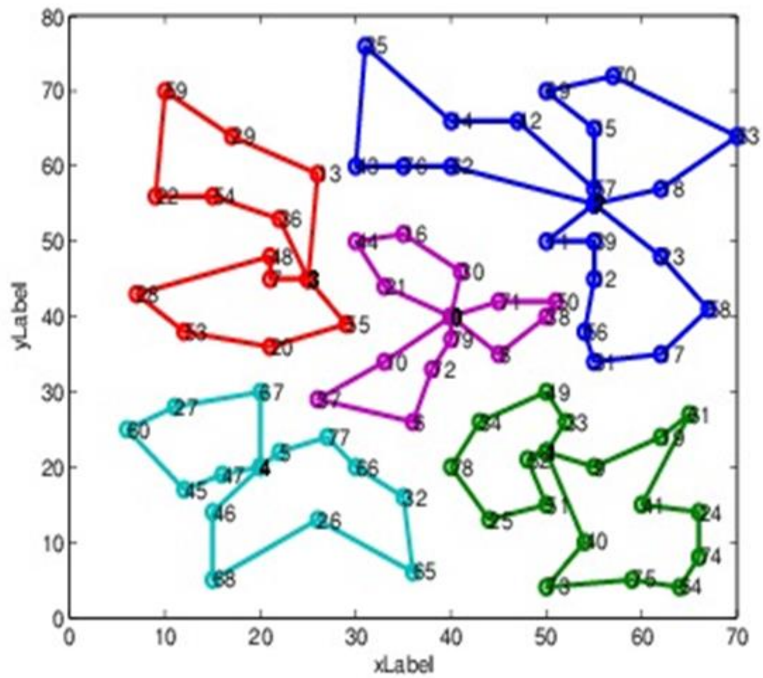
GA

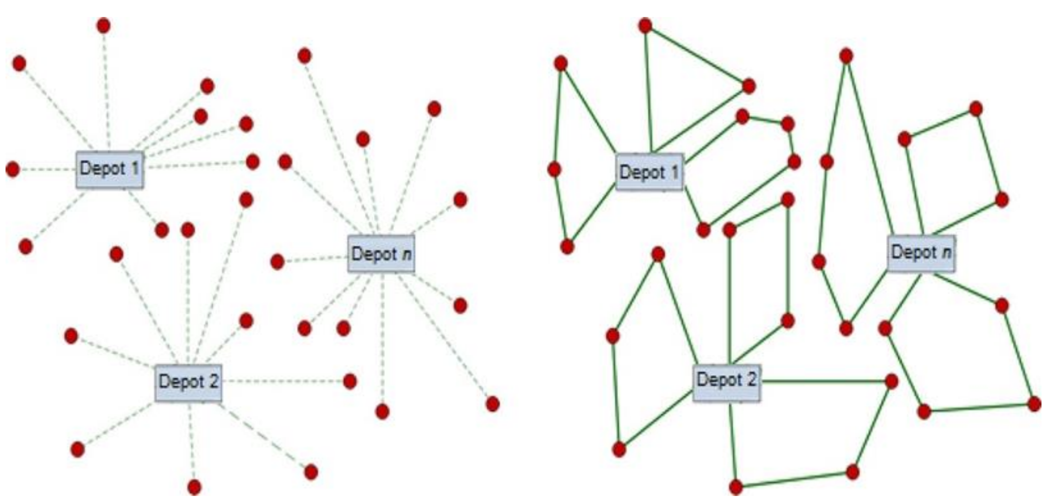


DE


File Edit View Navigate Code Refactor Run Tools VCS Window Help VRP - GA.py
VRP GA.py
GA.py main.py
VRP D:\College\Semester 5\A
directory library root
Customer.py
Depot.py
GA.py
main.py
pr10.txt
Utilities.py
Vehicle.py
External Libraries
Scratches and Consoles
import math
from random import sample, uniform
from Utilities import Utilities
class GeneticAlgorithm:
def __init__(self, vehicles_no, depots_no, customers_no, population_size, generation_size):...
@staticmethod
def generate_chromosome(depot):...
def generate_population(self, depots_list, parents=None):...
@staticmethod
def fitness_func(chromosome):...
def mutation(self, parents):...
@staticmethod
def crossover(target, mutant):...
@staticmethod
def DE_selection(survivor, targeta):...
def run(self, depots_list, customers_list):...
Run TODO Problems Terminal Python Console Event Log
161:1 CRLF UTF-8 4 spaces Python 3.9 (VRP)
Type here to search
12°C Clear
11:32 PM
12/26/2021

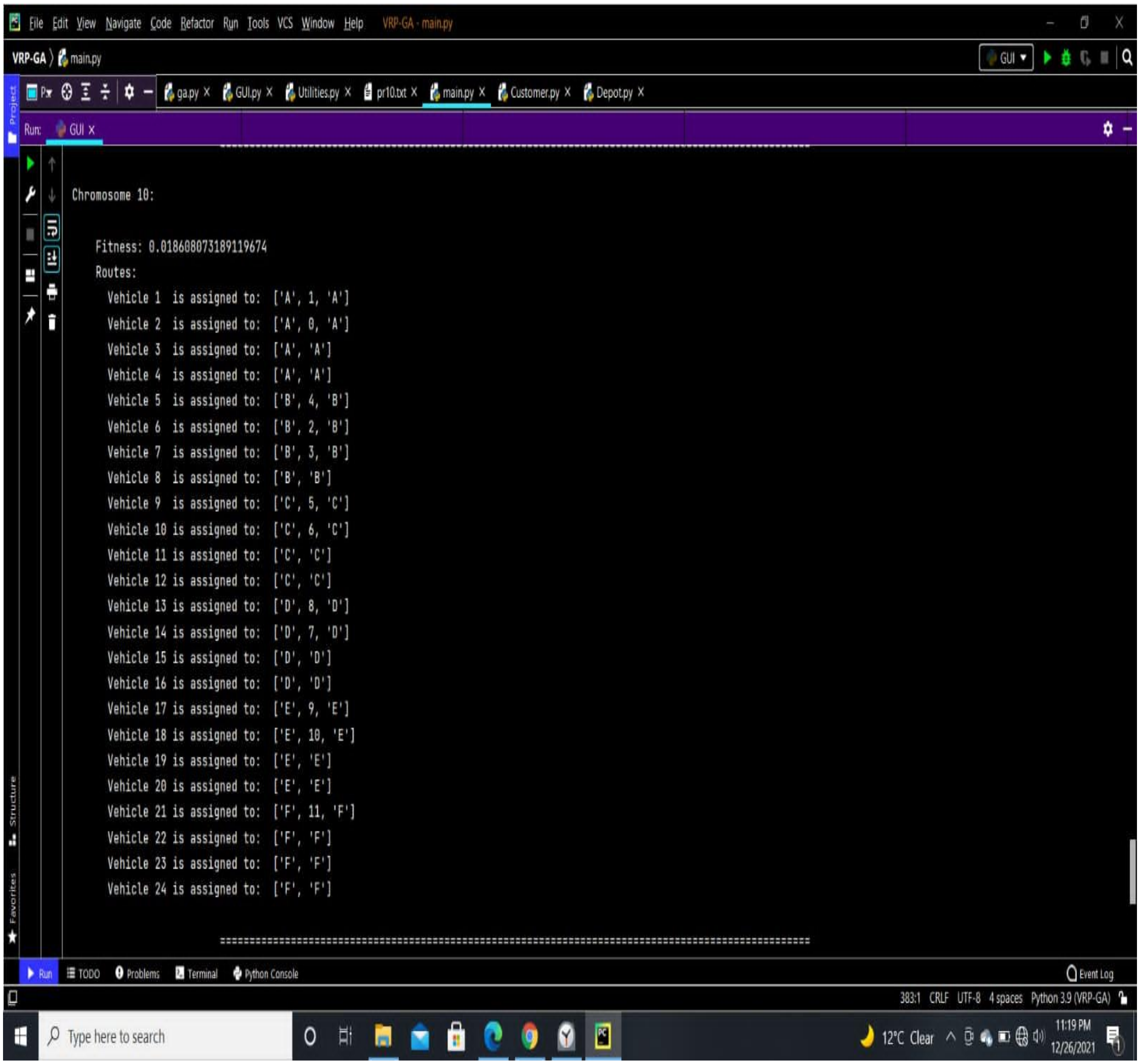
Results

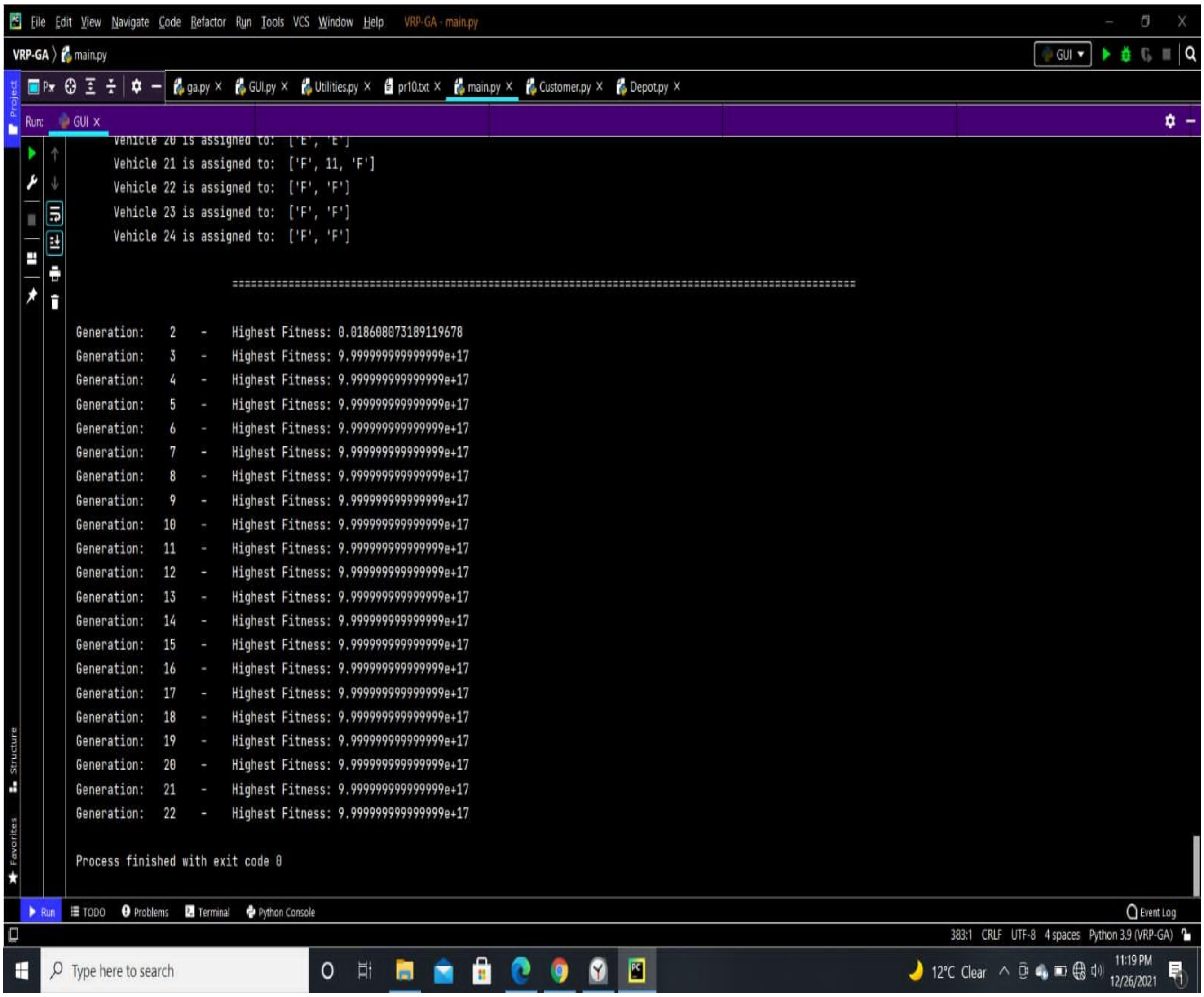




5.2 Samples of output

GA



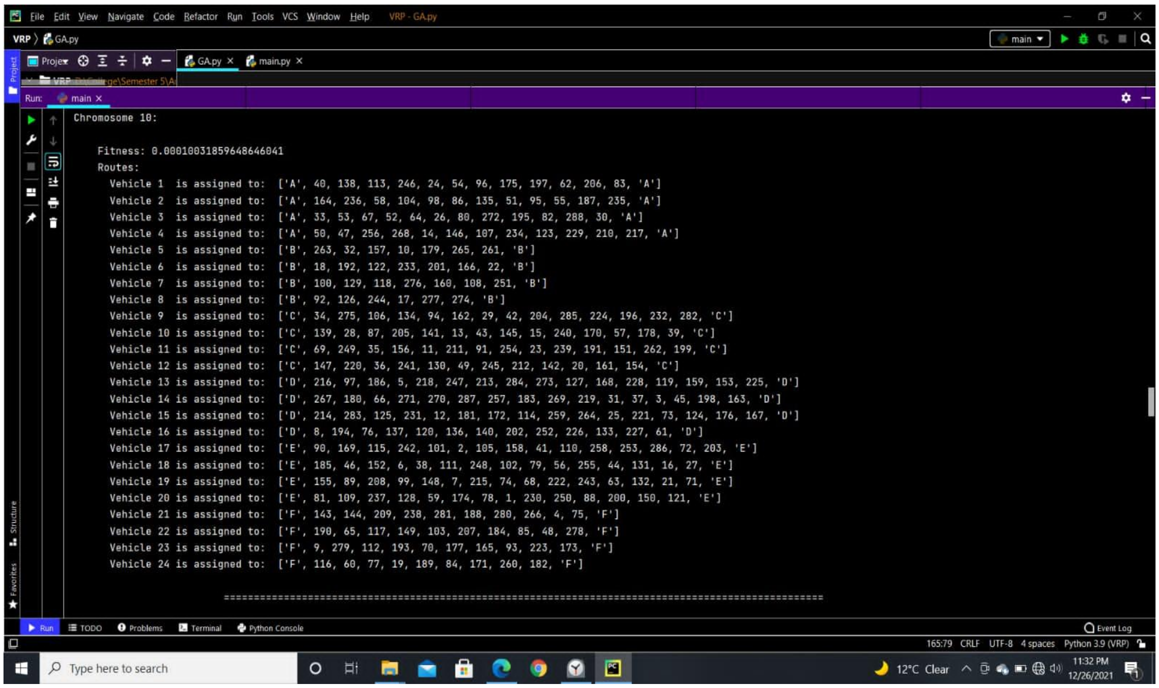


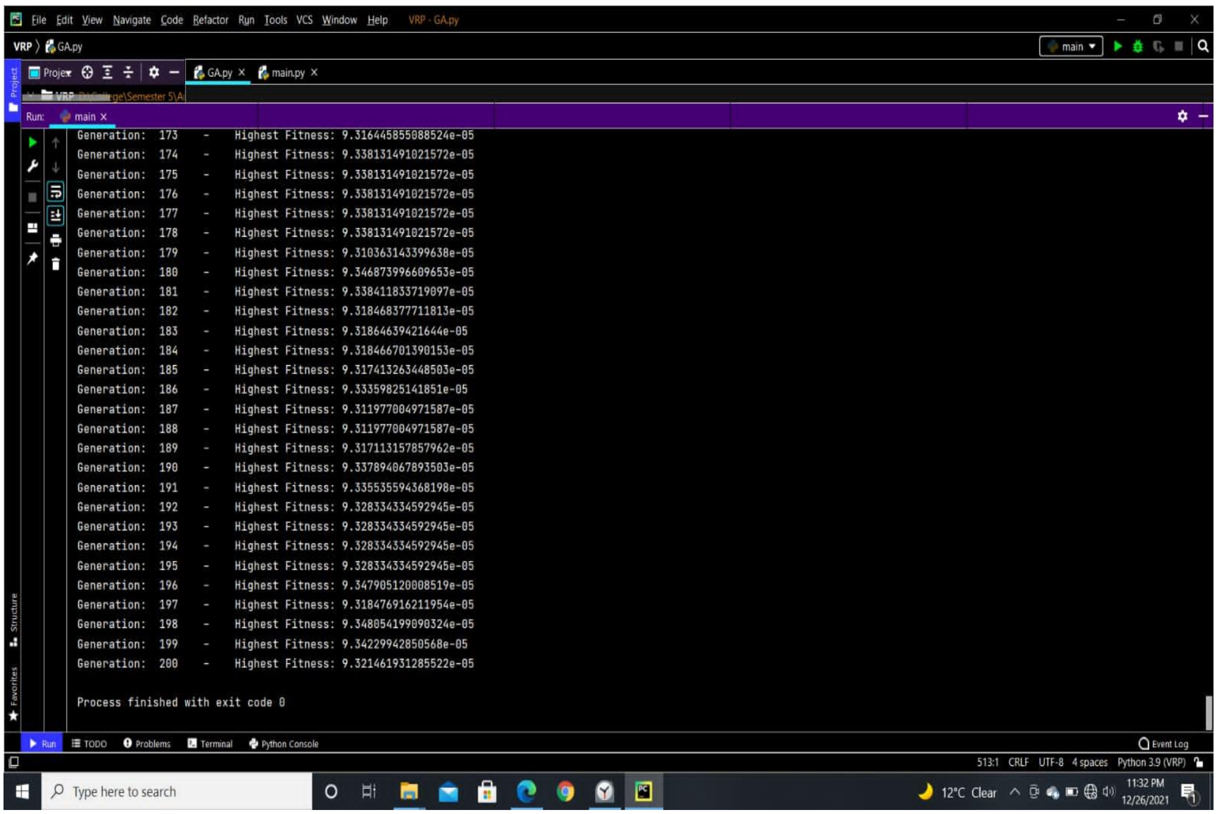
-The Algorithm stopped generating chromosomes after having reached chromosomes of the same fitness 20 times in a row.

-This is due to our conditions of termination which are either to reach 200 generations or generate 20 chromosomes of the same fitness in a row.

-The fitness reached a constant value quickly due to the lack of data.

DE


Chromosome 10:
Fitness: 0.00010031859648646041
Routes:
Vehicle 1 is assigned to: ['A', 40, 138, 113, 246, 24, 54, 96, 175, 197, 62, 206, 83, 'A']
Vehicle 2 is assigned to: ['A', 164, 236, 58, 104, 98, 86, 135, 51, 95, 55, 187, 235, 'A']
Vehicle 3 is assigned to: ['A', 33, 53, 67, 52, 64, 26, 80, 272, 195, 82, 288, 30, 'A']
Vehicle 4 is assigned to: ['A', 50, 47, 256, 268, 14, 146, 107, 234, 123, 229, 210, 217, 'A']
Vehicle 5 is assigned to: ['B', 263, 32, 157, 10, 179, 265, 261, 'B']
Vehicle 6 is assigned to: ['B', 18, 192, 122, 233, 201, 166, 22, 'B']
Vehicle 7 is assigned to: ['B', 100, 129, 118, 276, 160, 108, 251, 'B']
Vehicle 8 is assigned to: ['B', 92, 126, 244, 17, 277, 274, 'B']
Vehicle 9 is assigned to: ['C', 34, 275, 106, 134, 94, 162, 29, 42, 204, 285, 224, 196, 232, 282, 'C']
Vehicle 10 is assigned to: ['C', 139, 28, 87, 205, 141, 13, 43, 145, 15, 240, 170, 57, 178, 39, 'C']
Vehicle 11 is assigned to: ['C', 69, 249, 35, 156, 11, 211, 91, 254, 23, 239, 191, 151, 262, 199, 'C']
Vehicle 12 is assigned to: ['C', 147, 220, 36, 241, 130, 49, 245, 212, 142, 20, 161, 154, 'C']
Vehicle 13 is assigned to: ['D', 216, 97, 186, 5, 218, 247, 213, 284, 273, 127, 168, 228, 119, 159, 153, 225, 'D']
Vehicle 14 is assigned to: ['D', 267, 180, 66, 271, 270, 287, 257, 183, 269, 219, 31, 37, 3, 45, 198, 163, 'D']
Vehicle 15 is assigned to: ['D', 214, 283, 125, 231, 12, 181, 172, 114, 259, 264, 25, 221, 73, 124, 176, 167, 'D']
Vehicle 16 is assigned to: ['D', 8, 194, 76, 137, 120, 136, 140, 202, 252, 226, 133, 227, 61, 'D']
Vehicle 17 is assigned to: ['E', 90, 169, 115, 242, 101, 2, 105, 158, 41, 110, 258, 253, 286, 72, 203, 'E']
Vehicle 18 is assigned to: ['E', 185, 46, 152, 6, 38, 111, 248, 102, 79, 56, 255, 44, 131, 16, 27, 'E']
Vehicle 19 is assigned to: ['E', 155, 89, 208, 99, 148, 7, 215, 74, 68, 222, 243, 63, 132, 21, 71, 'E']
Vehicle 20 is assigned to: ['E', 81, 109, 237, 128, 59, 174, 78, 1, 230, 250, 88, 200, 150, 121, 'E']
Vehicle 21 is assigned to: ['F', 143, 144, 209, 238, 281, 188, 280, 266, 4, 75, 'F']
Vehicle 22 is assigned to: ['F', 190, 65, 117, 149, 103, 207, 184, 85, 48, 278, 'F']
Vehicle 23 is assigned to: ['F', 9, 279, 112, 193, 70, 177, 165, 93, 223, 173, 'F']
Vehicle 24 is assigned to: ['F', 116, 60, 77, 19, 189, 84, 171, 260, 182, 'F']


Generation: 173 - Highest Fitness: 9.316445855088524e-05
Generation: 174 - Highest Fitness: 9.338131491021572e-05
Generation: 175 - Highest Fitness: 9.338131491021572e-05
Generation: 176 - Highest Fitness: 9.338131491021572e-05
Generation: 177 - Highest Fitness: 9.338131491021572e-05
Generation: 178 - Highest Fitness: 9.338131491021572e-05
Generation: 179 - Highest Fitness: 9.310363143399638e-05
Generation: 180 - Highest Fitness: 9.346873996609653e-05
Generation: 181 - Highest Fitness: 9.338411833719097e-05
Generation: 182 - Highest Fitness: 9.318468377711813e-05
Generation: 183 - Highest Fitness: 9.31864639421644e-05
Generation: 184 - Highest Fitness: 9.318466701390153e-05
Generation: 185 - Highest Fitness: 9.317413263448503e-05
Generation: 186 - Highest Fitness: 9.33359825141851e-05
Generation: 187 - Highest Fitness: 9.311977004971587e-05
Generation: 188 - Highest Fitness: 9.311977004971587e-05
Generation: 189 - Highest Fitness: 9.317113157857962e-05
Generation: 190 - Highest Fitness: 9.337894067893503e-05
Generation: 191 - Highest Fitness: 9.335535594368198e-05
Generation: 192 - Highest Fitness: 9.328334334592945e-05
Generation: 193 - Highest Fitness: 9.328334334592945e-05
Generation: 194 - Highest Fitness: 9.328334334592945e-05
Generation: 195 - Highest Fitness: 9.328334334592945e-05
Generation: 196 - Highest Fitness: 9.347905120008519e-05
Generation: 197 - Highest Fitness: 9.318476916211954e-05
Generation: 198 - Highest Fitness: 9.348054199090324e-05
Generation: 199 - Highest Fitness: 9.34229942850568e-05
Generation: 200 - Highest Fitness: 9.321461931285522e-05
Process finished with exit code 0

6.Analysis , Didcussion , And future work

Advantages and Disadvantages

Comparing a differential evolution algorithm to a genetic algorithm is like comparing a screwdriver to a Swiss army knife. While screwdriver is an excellent tool for screwing and unscrewing screws, it will not do you any good when you want to cut a piece of meat. A Swiss army knife is a general purpose tool, but when trying to deal with screws, you still would rather have a proper screwdriver.

6.1 Genetic algorithms

- Advantages

1. **GA** search from a population of points, not a single point.

2. **GA** supports multi-objective optimization.

3. **GA** use payoff (objective function) information, not derivatives.

- Disadvantages

1. **GA** requires less information about the problem, but designing an objective function and getting the representation and operators right can be difficult.

2. Its really hard for people to come up with a good heuristic which actually reflects what we want the algorithm to do.

3. Its also hard to choose parameters like number of generations, population size etc. When we are working even though our heuristic was right we were not realizing it because we were running for a fewer generations.

6.2 Differential evolution

- Advantages

1. to find a minimum of a fitness function f(x):Rn→R . Individuals in the population of a differential evolution algorithm are vectors of real numbers.

2. you would be better of to use a differential evolution algorithm to measure fuel consumption on a vehicle model with respect to mass of an engine and a wheel size.

3. . It's the cumulative effect of three or more different changes that give the illusion of choice.

- Disadvantages

1. Some differential equations are not as well-behaved, and show singularities due to a failure to model the problem correctly, or a limitation of the model that was not apparent.

2. Some DE's can be solved analytically in closed-form, but most have to be approximated by numerical procedures, which can be unstable.

6.3 The future modifications

-Once the route assigned to each vechile in the depot the system try to predict time of the whole trip and display it to each customer(destinations) in addition to displaying time to the depot to regulate the timetable of each vechile the whole day.

-As a dispatcher , you want to know when your drivers are going off their routs , you can draw a line around the area in which a particular truck's deliveries for the day scheduled to occur ,and then receive automatic alerts if the GPS-tracking shows the driver outside of that zone.

-This gives you the oversight to spot any issues your drivers might be having and immmmediatly reach out of them to resolve the situation and get the rout back on track.

6.4 Literature Review

Kurnia, H., Wahyuni, E. G., Pembrani, E. C., Gardini, S. T., & Aditya, S. K. (2018, March). Vehicle Routing Problem Using Genetic Algorithm with Multi Compartment on Vegetable Distribution. In IOP Conference Series: Materials Science and Engineering (Vol. 325, No. 1, p. 012012). IOP Publishing.

Baptista, F., & Tavares, J. (2009). Bio-inspired Algorithms for the Vehicle Routing Problem, Ed.

Kunnapapdeelert, S. (2017). IMPROVED DIFFERENTIAL EVOLUTION ALGORITHMS FOR VEHICLE ROUTING PROBLEMS (Doctoral dissertation, Asian Institute of Technology).

Prayudani, S., Hizriadi, A., Nababan, E. B., & Suwilo, S. (2020, June). Analysis Effect of Tournament Selection on Genetic Algorithm Performance in Traveling Salesman Problem (TSP). In Journal of Physics: Conference Series (Vol. 1566, No. 1, p. 012131). IOP Publishing.

Tlili, T., Krichen, S., Drira, G., & Faiz, S. (2016). On Solving the Multi-depot Vehicle Routing Problem. In Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics (pp. 103-108). Springer, New Delhi.

Webb, G., & Yu, X. (Eds.). (2004). AI 2004: Advances in Artificial Intelligence: 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia, December 4-6, 2004, Proceedings (Vol. 3339). Springer