ModeFair Assesment

Mohd Bruce Lee bin Mustapa

Task 1

1. In this task I decided to use Genetic Algorithm for finding the list of the pattern.
2. This method involved several steps which is population initialization, fitness evaluation, parent selection, crossover, and mutation. We will go through each step
    a. Chromosome initialization
        - In this step, I used random generated number in a range to create the chromosome. The number must be not repeated and the first and the last number I put A and C as the constant which means the value at the middle of the array only to be shuffle.
    b. Fitness evaluation
        - In this step I used all the rule that nee to follow as evaluation point to measure the fitness. Firstly the arrangement of A, I, and C must be sequentially, then the chromosome will get extra mark. Next the node that passes one point that not being use will cause mark deducted vice versa if the point passed through is visited, extra mark will give.
    c. Parent selection
        - For parent selection we just take random two chromosome to be compete and produce child.
    d. Crossover
        - For crossover, this one use only one point switch which is two same values at two different chromosome and two different indexes will be switch. This the only method to avoid repeated value in the chromosome
    e. Mutation
        - For mutation, I just switch two values by switch their place. After that we evaluate it back then make comparison between 2 parent and 2 children to get the 2 best highest fitness value to survive in the generation.
3. There are several weakness and constraint
    a. The pattern generated is solid 9 node which do not include 7 or even lesser than that.
    b. The crossover can’t make to much change so the evolution is to small.
    c. The output is no optimized.
4. Here I attached a youtube video for better explanation - https://youtu.be/Nia8whr3nxs

Task 2

sample_solution.py | sample_data.json | MVP20.json | MVP30.json | MVP40.json | MVP50.json

sample_solution.py > ...

```
for i in range(20, 60, 10):
    car_types, depot, custs = gen_test_case(i)
    save_test_case(f"MVP{i}.json",car_types, depot, custs)

sample_test = {"path": f"sample_data.json", "best_ind": [], "best_score": 0, "avg_score": 0, "avg_time": 0}
start_time = time.time()
car_types, depot, custs = load_problem(sample_test['path'])
num_cars = len(custs) * 2
cars = [{"id": len(custs) + i + 1, "type": i // int(num_cars / 2)} for i in range(num_cars)]
POP_SIZE = 20
SELECTION_SIZE = 14
ELITE_SIZE = 4
NUM_GENS = 200
MUTATION_RATE = 0.9
CROSSOVER_RATE = 0.9 ; "changed here"
NUM_PTS = num_cars + len(custs)

ga = MVPGeneticAlgorithm(
    car_types, cars, custs, depot,
    POP_SIZE, SELECTION_SIZE, NUM_GENS,
    MUTATION_RATE, ELITE_SIZE, CROSSOVER_RATE, NUM_PTS,
)

ga.solve()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Final Result:
Total distance = 97.535 km
Total cost = RM 117.04
```

```
Final Result:
Total distance = 97.535 km
Total cost = RM 117.04

Vehicle 1 (Type A)
Round Trip Distance: 38.012 km, Cost: RM 45.61, Demand: 19
Depot -> C5 (10.483km) -> C1 (4.839km) -> C4 (4.021km) -> C3 (8.323km) -> Depot (10.347 km)

Vehicle 2 (Type A)
Round Trip Distance: 40.557 km, Cost: RM 48.67, Demand: 22
Depot -> C8 (11.612km) -> C9 (11.358km) -> C10 (5.060km) -> C7 (6.018km) -> Depot (6.508 km)

Vehicle 3 (Type A)
Round Trip Distance: 18.967 km, Cost: RM 22.76, Demand: 16
Depot -> C2 (9.072km) -> C6 (1.834km) -> Depot (8.061 km)

PS C:\Users\MohdBruceLee\Downloads\Sample Solution>
```

1. In this task, there are several changes I made but they all just minor changes just parameter tuning. Here the changes done:
   a. POP_SIZE = 20
   b. SELECTION_SIZE = 14
   c. ELITE_SIZE = 4
   d. NUM_GENS = 200
   e. MUTATION_RATE = 0.9
   f. CROSSOVER_RATE = 0.9