Work package Evolutionary Algorithm

I. Introduction

1. Copy the file and uncompress it.
2. Load the program in a C++ environment, compile by typing 'make' on the command line or using Code::Blocks for example, and run by 'algo_evo.exe' on the command line without argument or with 6 arguments: 'algo_evo.exe nb_generation population_size crossover_rate mutation_rate nb_cities distances_cities'.

II. Program composition

4 classes (chromosome, population, random and ae, with the files .h et .cpp), the main function main.cpp, 2 input data files 'distances' and the makefile.

1. Class chromosome
   attributes: genes, taille, fitness
   methods: constructor chromosome(), functions afficher(), evaluer(), ordonner(), copier(), identique(), echange_2_genes()…
2. Class population
   attributes: individus, taille_pop, ordre
   methods: constructor population(), functions statistiques(), nb_chromosomes_similaires(), similitude()…

3. Class random
   methods: randomize(), aleatoire()

4. Class ae
   attributes: nbgenerations, taille_pop, taux_croisement, taux_mutation, taille_chromosome, pop, les_distances
   methods: constructor ae(), optimiser(), croisement1X()…

III. Running tests

A. Tests on the number of generations on 10 cities problem

Input data:
   Number of generations: 50, 500, 1.000, 100.000
   Population size: 20
   Crossover rate: 0.8
   Mutation rate: 0.5
   Distances_cities: distances_entre_villes_10.txt

1. Run the algorithm 10 times for each number of generations (the global minimum is 3473).
2. How many generations it needs to converge? What is it doing after the convergence? What are your conclusions?
3. What are doing the crossover and mutation heuristics used in the algorithm?
4. Calculate the number of neighbors for any solution with the crossover heuristic and with the mutation heuristic for the 10 cities and 50 cities problems.
5. How many solutions are visited by the algorithm before its convergence? What are your conclusions about the algorithm performance?

B. Tests on the population size on 10 cities problem

Input data:
   Number of generations: 1.000

Population size: 5, 50, 100, 500
Crossover rate: 0.8
Mutation rate: 0.5
Distances_cities: distances_entre_villes_10.txt

1. Run the algorithm 10 times for each population size.
2. How many generations it needs to converge? What is it doing after the convergence? What are your conclusions?
3. How many solutions are visited by the algorithm before its convergence? What are your conclusions about the algorithm performance?

C. Tests on 4 combinations of population size and number of generations on 50 cities problem

Input data:

Number of generations: the two best found in the section A
Population size: the two best found in the section B
Crossover rate: 0.8
Mutation rate: 0.5
Distances_cities: distances_entre_villes_50.txt

1. Run the algorithm 10 times for each of the four combinations.
2. How many generations it needs to converge? What are your conclusions?
3. How many times is improved the best solution?
4. How many local minima are visited by the algorithm?

D. Crossover and mutation analysis on 50 cities problem

Keep the two best combinations of generation number and population size found in section C.
1. Run the algorithm 10 times with the crossover rate sets to 1 and the mutation rate sets to 0.
2. Run the algorithm 10 times with the crossover rate sets to 0 and the mutation rate sets to 1.
3. What are your conclusions on the performances between both cases and the case tested in section C with 0.8 and 0.5?
4. Which machine learning setting of the crossover and mutation rates might be defined to adapt them to minimize the time needed to get the optimal solution?

E. Selection analysis on 50 cities problem

Keep the best combination of algorithm parameters tested up to now.
1. What is doing the selection procedure used in the algorithm?
2. Use the random selection and replacement in the algorithm and run it 10 times
3. Use the ranking selection and replacement in the algorithm and run it 10 times for the ranking rate equal to 0, 1, 5, 100 and 1.000.
4. Compare and comment the results of the three selections: wheel, random and rank.