# Work package Tabu Search 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_tabou.exe' on the command line without argument or with 4 arguments: 'algo_tabou.exe nb_iterations duration_tabou nb_cities distances_cities'.

## II. Program composition

3 classes (rechercheTabou ; solution ; random with the files .h et .cpp), the main function main.cpp, 2 input data files 'distances' and the makefile.

1. Function main()
2. Class solution
    - a. constructor solution()
    - b. function evaluer()
3. Class rechercheTabou
    - a. constructor RechercheTabou()
    - b. function voisinage_2_opt()
    - c. function optimiser()

## III. Running tests

*A. Tests with no Tabu duration on 10 cities problem with 3 iteration cases*

Input data:

Nb_iterations: 5, 10, 100
Duration_tabou: 0
Nb_cities: 10
Distances_cities: distances_entre_villes_10.txt

1. Run the algorithm 10 times for each Nb_iterations (<u>the global minimum is 3473</u>).
2. How many iterations it needs to converge? What is it doing after the convergence? Which are your conclusions?
3. Calculate the number of solutions to the 10 cities and 50 cities problems.
4. Calculate the number of neighbors for any solution with 2-opt for the 10 cities and 50 cities problems
5. How many solutions are visited by the algorithm before its convergence? Which are your conclusions about the algorithm performance?

*B. Tests with no Tabu duration on 50 cities problem with 3 iteration cases*

Input data:

Nb_iterations: 10, 100, 1000
Duration_tabou: 0
Nb_cities: 50
Distances_cities: distances_entre_villes_50.txt

1. Run the algorithm 10 times for each Nb_iterations (<u>the global minimum is 5644</u>).
2. How many iterations it needs to converge? What is it doing after the convergence? Which are your conclusions?
3. How many solutions are visited by the algorithm before its convergence? Which are your conclusions about the algorithm performance?

### *C. Tests with Tabu duration*

1. Run the algorithm 10 times with:

    Nb_iterations: 1000<br>
    Duration_tabou: 10<br>
    Nb_cities: 50<br>
    Distances_cities: distances_entre_villes_50.txt

2. How many times is improved the best solution?
3. How many local minima are visited by the algorithm? Does it converge? What are your conclusions?

### *D. Variation in Tabu duration*

1. How is coded the Tabu list? What is tabu? How is coded the Tabu duration? What is the maximal Tabu duration?
2. Run the algorithm 10 times with:

    Nb_iterations: 1000<br>
    Duration_tabou: 20, 40, 60, 100, 1000<br>
    Nb_cities: 50<br>
    Distances_cities: distances_entre_villes_50.txt

3. How many times is improved the best solution?
4. How many local minima are visited by the algorithm? Does it converge? What are your conclusions?
5. For your best value of Tabu duration, try different Nb_iterations to find the higher efficiency of the algorithm (higher % of success over 10 runs)?
6. What should be the impact of an infinite value for the Tabu duration?