Data sheet for Tabu Search algorithm improvement

I. Reference results on 50 and 100 cities with constant Tabu Duration

1. From your current algorithm, keep the two-best constant Tabu Durations you observed on the 50 cities problem.
2. Run these two versions 10 times with:
    Data set: 50 cities
    Nb_iterations: 10,000
    Duration_tabou: the two-best constant Tabu Durations from your 1st report

    Data set: 100 cities
    Nb_iterations: 100,000
    Duration_tabou: the two-best constant Tabu Durations from your 1st report
3. Report for each data set and each run:
    The best solution fitness
    The iteration at which the best solution is founded
    The number of local minima visited by the algorithms

II. Random-based Tabu Duration (RTD)

Implement a Random-based Tabu Duration between 2 bounds: the Tabu Duration varies at each iteration during the search, randomly generated between 2 bounds (refer to page 61 section 1 of the course).

1. With n the number of cities, the bounds are:
    Lower bound: upper_integer(alpha*0.5*square root of n)
    Upper bound: upper_integer(alpha*1.5*square root of n)
    NB: upper_integer is corresponding to the mathematical ceiling function
2. Run the algorithm in the same conditions as I.2 for:
    Alpha: 1 and 5
3. Report the same observed data as I.3 and the mean Tabu Duration for each city (mean value of Tabu Durations on all moves involving that city)
4. Report your conclusions about the algorithm performance linked to RTD

III. Frequency-based Tabu Duration (FTD)

Implement a Frequency-based Tabu Duration: the Tabu Duration is initially fixed to a constant value, then it is incremented by a function of the number of times a variable is used for a move (refer to page 61 section 2 of the course) or the number of times a move is used (switch of two cities (i,j) in our case study).

1. Release FTD1: With n the number of cities, and n_{ij} the number of times the move (i,j) has been done, the function is:
    Tabu Duration: constant + upper_integer(alpha*n_{ij}/n)
2. Run the algorithm in the same conditions as I.2 for:
    Constant: 10
    Alpha: 1 and 5
3. Report the same observed data as I.3 and the mean Tabu Duration for each city (mean value of Tabu Durations on all moves involving that city)
4. Report your conclusions about the algorithm performance linked to FTD1
5. Release FTD2: Change the Tabu Duration function by:
    Tabu Duration: constant + upper_integer(alpha*n_i/n) where n_i is the number of times the city i is used for a move, whatever the exchange city j.
6. Run the algorithm in the same conditions as I.2 for:

Constant: 10
Alpha: 1 and 5

7. Report the same observed data as I.3 and the mean Tabu Duration for each city (mean value of Tabu Durations on all moves involving that city)
8. Report your conclusions about the algorithm performance linked to FTD2

IV. Which are your personal learning outcomes from this work?