In order to validate the proposed project, this chapter discuss the results obtained by applying it in the real case scenario of Campolide. The implementation and tests of the proposed GA were done using the Python 2.7 programming language on a MacBook Pro from late 2013 with the following specifications: 2.4Ghz Core i5-4258U processor with 3MB L3 cache, 4GB of 1600MHz DDR3 RAM and macOS 10.13 High Sierra. To decide on the genetic algorithm required parameters addressed in the previous chapter, multiple runs were made and their results were analyzed. The next paragraphs will present the chosen test cases over the possible parameters combinations and the result of the algorithm execution with these combination of parameters.

The settings of the EC algorithm are the following: Number of generations: 50000 (except for the instance A32k5, where only 10000 generations were required); Population size: 200; Tournament selection with tourney size: 5; Elitist strategy; Crossover rate: 0.75; Mutation rates: swap: 0.05; inversion: {0.1, 0.15}; insertion: 0.05; displacement: {0.15, 0.2}. For every set of parameters we performed 30 runs with the same initial conditions and with different random seeds. All initial populations were randomly generated according to the following algorithm: **GVR: a New Genetic Representation for the Vehicle Routing Problem**

In order to execute the GA, the user should adjust the values of the following six parameters: *Iterations, Population, Children per Generation, Mutation Policy, Mutation Probability* and *Diversity Threshold.* The values of the parameters of this algorithm can be selected from the following set of values:

* *Iterations* G {100,000, 1,000,000},
* *Population* e {20, 50, 72, 100, 144, 300},
* *Children per Generation* G {1,2, 3, 4, 5},
* *Mutation Policy* G {Swap two loading spots. Reverse sub tour},
* *Mutation Probability* G {0, 0.05, 0.1, 0.2, 0.4, 0.8} and
* *Diversity Threshold* G {0, 0.01, 0.05, 0.1, 0.5}

In order to cover all the possible solutions that the aforementioned parameter settings can produce, there are 3,6D0 different combinations, which make the testing process even more difficult. The set of values for the GA parameters presented above is common in the literature and this is why they have chosen. **Genetic Algorithms for Municipal Solid Waste Collection and Routing Optimization**