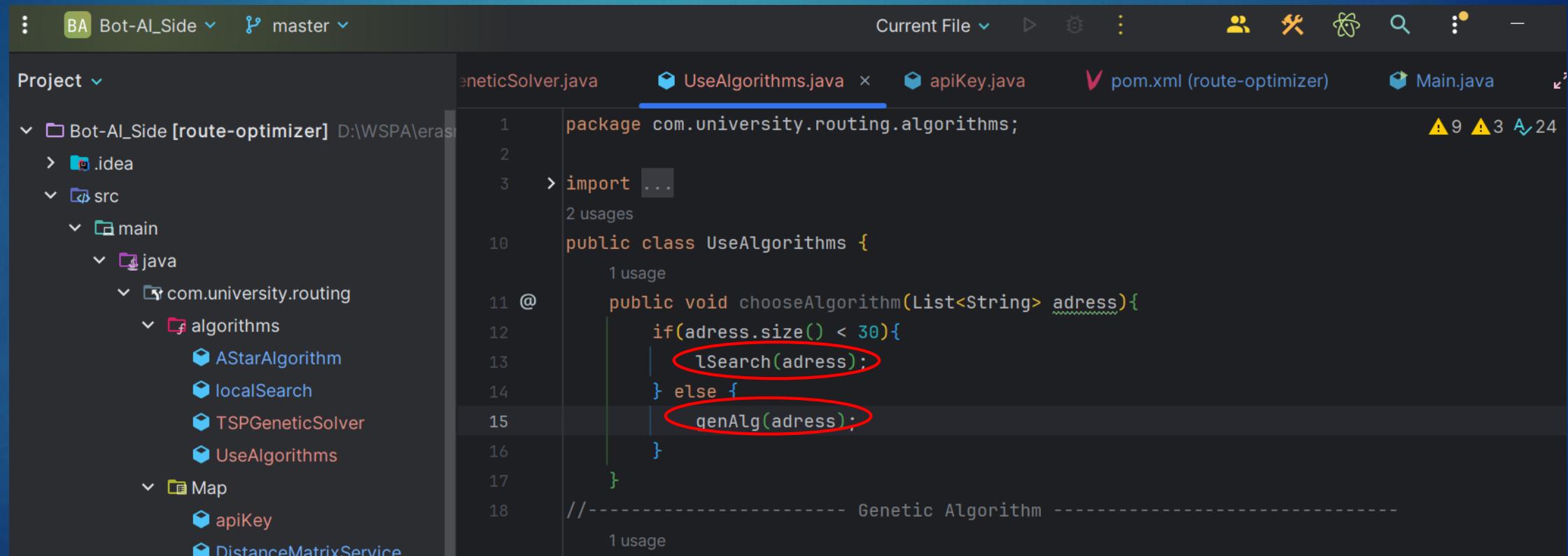


Algorithm side

We have identified 3 algorithms : A*, local search and genetic algorithms.

Depending on the number of addresses, the Local Search algorithm (if there are fewer than 30 addresses) or the Genetic Algorithm (if there are 30 or more addresses) will be used to find the best way to visit all the addresses and return to the starting point



The screenshot shows a Java code editor in an IDE. The project structure on the left includes a .idea folder, a src folder containing main and java packages. The java package contains subfolders com.university.routing.algorithms, com.university.routing, Map, apiKey, and DistanceMatrixService. Inside com.university.routing.algorithms, there are files AStarAlgorithm.java, localSearch.java, TSPGeneticSolver.java, and UseAlgorithms.java. The UseAlgorithms.java file is currently selected. The code in UseAlgorithms.java is as follows:

```
package com.university.routing.algorithms;
import ...;
public class UseAlgorithms {
    public void chooseAlgorithm(List<String> adress){
        if(adress.size() < 30){
            lSearch(adress);
        } else {
            genAlg(adress);
        }
    }
}
```

Two lines of code are highlighted with red circles: "lSearch(adress);" at line 13 and "genAlg(adress);" at line 15. The code editor interface includes tabs for other files like GeneticSolver.java and Main.java, and various status indicators at the top right.