**Project report**

**Authors**: Matej Luburić, Filip Šimek

**Problem Description**

Problem title is Capacitated vehicle routing problem with time windows. Given information about depot, vehicles and customers we need to find optimal solution in which all customers are served with respect to all defined constraints. A depot is starting point of all vehicle routes. Each vehicle has capacity limit. Each customer has defined coordinates, demand value, service, ready and due time. Ready and due times are time constraints in this problem. A vehicle is serving customers fulfilling their demand. Arrival time to each customer is calculated as maximum value between customer's ready time and sum of previous customer's arrival time, it's service time and upper ceiling of distance between two customers. The primary objective in this problem is to minimize number of vehicles needed to serve all customers. There is also a limit in number of vehicles that can be used. The secondary objective is to minimize the sum of distances on all routes. As there are many customers, finding exact solution to this problem is not an option, so we need to implement some optimization algorithm. Firstly, we construct an initial solution using greedy algorithm and then we optimize it using simulated annealing algorithm.

**Algorithm description**

Used languages: Java for project optimization, Python for visualisation

Solution representation: Customer input information are represented by a Customer class. These information is inmutable. Mutable customer information which is it's served time and position on route are represented in a CustomerCalc class with reference to corresponding customer. Vehicle is represented by Vehicle class. It has information about customers served and route length. Whole solution is represented by a Solution class that stores list of used vehicles.

Objective/fitness function: Solutions are compared primarly based on number of vehicles use din them. If these values are equal, better solution is one that has lower total route distance.

Initial solution construction:

Initial solution optimization:

**Algorithm pseudocode**