Read Me

The program is to solve the VRP problem that is the Vehicle Routing Problem. The codes are both written in Golang and Python. The main objective of the code is to minimize distance and maximize loads with constraints that the timing shouldn’t increase 12 hours and the mean cost for the drivers should decrease as much as possible.

**Input-**

Given the file named Training Problems that consists of loadNo pickup and dropoff as cartesian points a driver is said to have completed a transfer when he drives to the pickup location picks up the load and then drives off to the dropoff location and f=drops off the load. The time required to drive from one point to another in minutes is called Euclidean distance and the formula was given as-

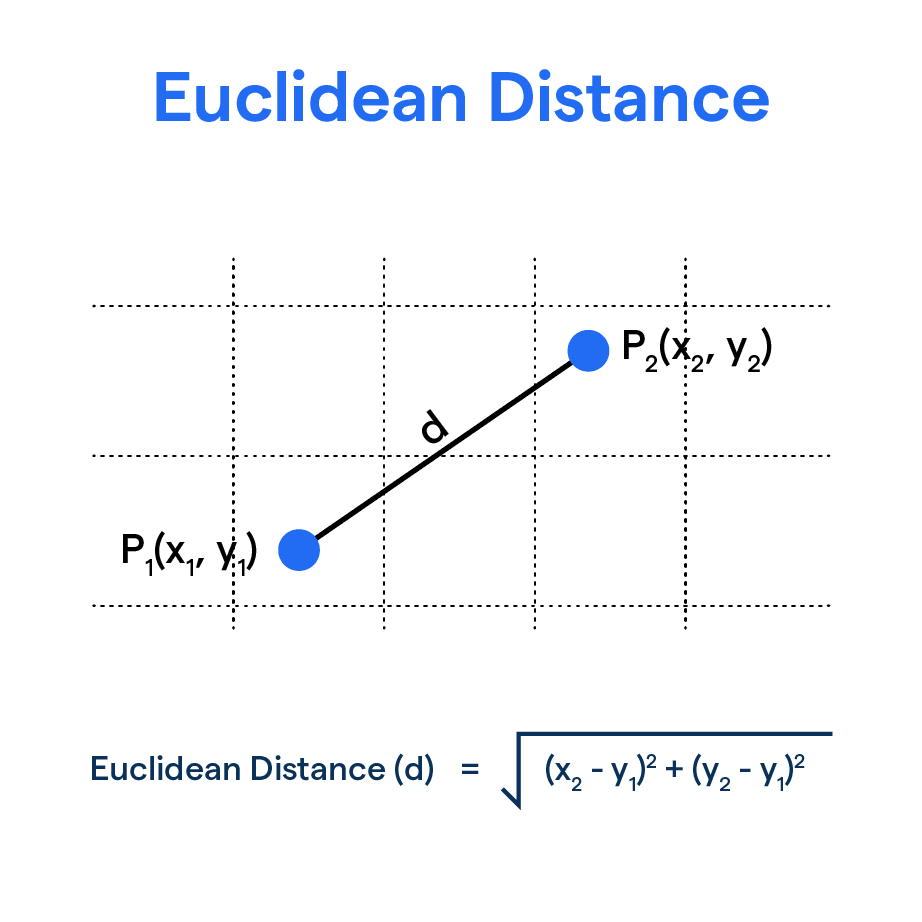
loadNumber pickup dropoff

1 (-50.1,80.0) (90.1,12.2)

2 (-24.5,-19.2) (98.5,1,8)

3 (0.3,8.9) (40.9,55.0)

4 (5.3,-61.1) (77.8,-5.4)



**Ways to run the code-**

**Python-**

* cd VortoProblemPython
* python3 assessment.py <input file path>

**Golang-**

* cd VortoProblemGolang
* Go to the main.go file
* Type the command-

**go run main.go <path to the directory consisting of the problems>**

**Output-**

The program writes a solution to stdout. The solution lists, on separate lines, each driver’s ordered list of loads as a schedule. An example solution to the above problem could be:

[1]

[4,2]

[3]

This solution means one driver does load 1, another driver does load 4 followed by load 2, and a final driver does load 3.

**Note-** A little about Clark and Wright Savings Algorithm-

The core of the algorithm Clarke and Wright Saving is calculating savings is measured by how much do the reduction of mileage and time spent by linking nodes that exist and make a route based on the value of the savings of the largest is the distance between the starting node and the destination node.

References-

<https://www.google.com/search?sca_esv=d4fe766562f09bfd&sxsrf=ADLYWIKhdNvMJba7NHjYvldcRoiIeeAM8Q:1721600313279&q=euclidean+distance+image&udm=2&fbs=AEQNm0Aa4sjWe7Rqy32pFwRj0UkWd8nbOJfsBGGB5IQQO6L3J_86uWOeqwdnV0yaSF-x2jpXXSZVlK6C0YPjHbsLu8HQWfm38FRPm5FVWQmVvNxikPuGUro0kULs737NLvYt_770cqPBb-7c9YW98tWSpANkOcZZylZvCbVkKD2SeH_fJOnPgXdUsVXCVyRh_eSp_T6GxUwRZE0bpoVyiYh4GfPMFOiP6w&sa=X&ved=2ahUKEwiv2fj4lLmHAxVKRTABHcgnCFkQtKgLegQIGBAB&biw=1200&bih=720&dpr=2#vhid=uN8WQ1cIki5quM&vssid=mosaic>

https://iopscience.iop.org/article/10.1088/1742-6596/2421/1/012045/pdf#:~:text=The%20core%20of%20the%20algorithm,node%20and%20the%20destination%20node.