CARP Project

ZhouYusheng 11612417

11612417@mail.sustc.edu.cn

1 Preliminaries

This project is doing some implementations to solve a capacitated arc routing problem(CARP).

1.1 Problem Description

Arc Routing is the process of selecting the best path in a network based on the route.The goal of CARP is to produce a route with the minimum cost.(1)

1.2 Problem Applications

One example is garbage collection, where some route might require a garbage collection while others don't, and to calculate the minimum cost.

2 Methodology

Despite the CARP rule is simple, the implementation is full for details handing and requires appropriate data structure to represent. This part describes the representation, the more detail of algorithm and the architecture used in the codes.

2.1 Notation

The project contains some important data:
s: the number of vertices in the graph

C: the capacity of a vehicle
d: the depot of a graph
mc: the cost between two vertices stored in a 2-dimensional matrix
md: the demand between two vertices stored in a 2-dimensional matrix
D: distance between two vertices

2.2 Architecture

Here list the functions in the Python file CARPsolver with their usage.
Open: read the given info, generate mc and md
Floyd: use Floyd algorithm calculate the minimum cost between two nodes
demandcost: store a route cost which demand is not zero in a matrix
sformat: store the route info in a list
PathScan: an algorithm to calculate the route and cost

2.3 Detail of Algorithm

Here show the detail of some significant algorithm.

Floyd: implement the Floyd algorithm to calculate the minimum cost between two vetices and store in mincost matrix.

Algorithm 1 Floyd

```
Input: mc,s
Output: mc //minimum cost
 1: for k in (1,s) do
 2:     for i in (1,s) do
 3:         for j in (1,s) do
 4:             if $mc[i][j] < mc[i][k] + mc[k][j]$ then
 5:                 $mc[i][j] \leftarrow mc[i][k] + mc[k][j]$
 6:             end if
 7:         end for
 8:     end for
 9: end for
10: return mc
```

Algorithm 2 PathScan

```
Input: mc,s,md,d,C
Output: Sum //total cost tR //total Route
 1: $k \leftarrow 0$
 2: copy all required arcs in list task
 3: while $task! = \phi$ do
 4:     $k \leftarrow k + 1$; $Rk \leftarrow \phi$; load(k); $cost(k) \leftarrow 0$; $\leftarrow 1$;
 5:     while $task! = \phi|D! =$ inf do
          $D \leftarrow$ inf;
 6:         for u in task — load(k)+md<C do
 7:             if di,beg(u) < d then $D \leftarrow di, beg(u)$; $D \leftarrow u$;
 8:             end if
 9:             if di,beg(u) == d then $D \leftarrow u$;
10:             end if
11:         end for
12:         add D in route Rk;
13:         remove arc D and its opposite from task;
14:         $load(k) \leftarrow load(k) + md$
15:         $cost(k) \leftarrow cost(k) + mc + d$
16:         $i \leftarrow end(D)$
17:     end while
18:     $cost(k) \leftarrow cost(k) + di1$
19: end whilereturn Sum,tR
```

3 Empirical Verification

Empirical verification is compared with given .dat dataset

3.1 Environment

This project is written in Python using IDE PyCharm. The libraries being used time, numpy and sys. Python version is 3.7

3.2 Performance and Check

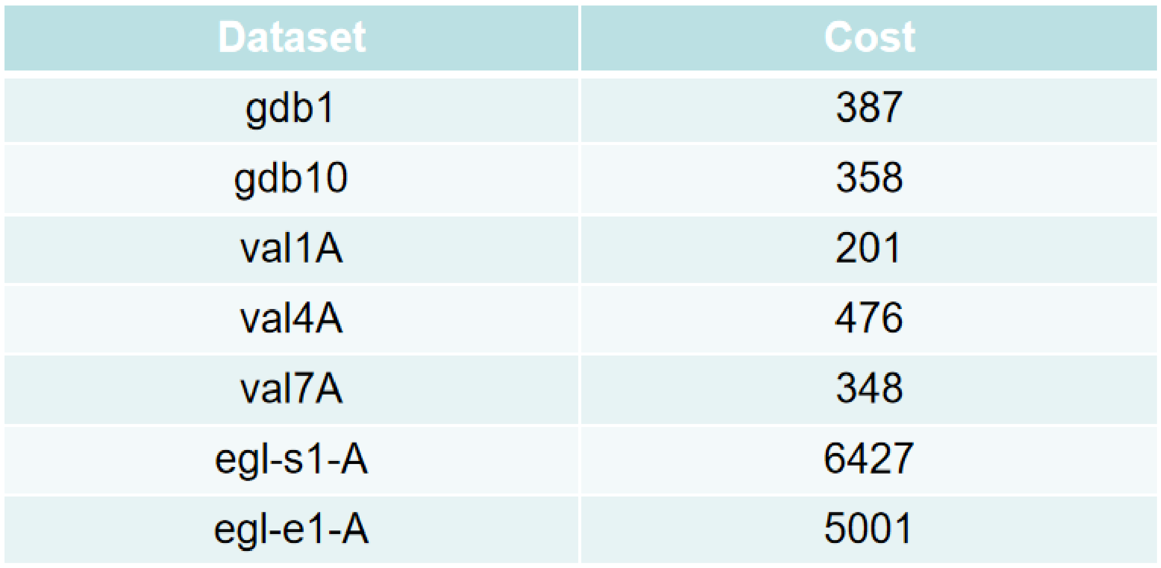
Empirical verification is using the CARP result check web(http://carp-judge.herokuapp.com) to make sure the results are right.

3.3 Data and data structure

Data being used in this project is seven dat files for test the codes. The data structures used in this project include list and array.

3.4 Result

The results of different dataset are showing below,for cost, the unit is second.



Dataset	Cost
gdb1	387
gdb10	358
val1A	201
val4A	476
val7A	348
egl-s1-A	6427
egl-e1-A	5001

Table 1: Results of data sets

3.5 Conclusion

The project is using Floyd algorithm and PathScan algorithm to implement a feasible and relatively accuracy solution to CARP. The disadvantage is comparing to other method, the results in this project is having higher cost,however the advantage of the method is it easy to understand and implement.

References

[1] Wikipedia, “Arc routing.” `https://en.wikipedia.org/wiki/Arcrouting`. Accessed November 21,2018.