CARP Project

Wang Zhiyuan 11610634
CSE
Computer Science and Technology
11610634@mail.sustc.edu.cn

1. Preliminaries

1.1. Software

For this project, I write it by python, and the extra packets I used is numpy.

This project aim to design a excellent enough solution of a Capacitated Arc Routing Problems

1.2. Algorithm

The algorithm I used is Genetic Algorithm, and I design 4 rules to do the pathscanning, design ulusoy, 2-opt, Merge-Split Operator to do the local-search and variation, and use a way that desrible in the paper [**?**] to do the crossover.

2. Methodology

2.1. Representation

In my code, to design the algorithm, I write it into three part:

- *pathscanning*: Do the pathscanning, get the initial solution.
- *localsearch*: For a ordered list, give a best split of it.
- *varition*: For a solution change it and apply the change if the solution can be better.
- *crossover*: Crossover two solution and generate a child solution.

2.2. Architecture

- Read data an storage:
    - Open the filename incoming as the parameter
    - Storage all the node and the cost of edge in a matrix, and this matrix in fact is the adjacency matrix of the graph.
    - Storage all the edge and its demand.
    - Storage all the edge that have demand in a set.
- Use floyd to the adjacency matrix of the graph.

- pathscanning: Random use the step below in each step.
    - maximize the distance
    - minimize the distance
    - maximize the term dem(t)/sc(t)
    - maximize the term dem(t)/sc(t)
- Choice 30 best result in the pathscanning and generate the initial population
- Do MS(Merge-Split) Operator to the population.
    - Merge and do split. Then use ulusory.
- Crossover
- Do MS Operator until the timeout.
    - Merge and do split. Then use ulusory.

2.3. Detail of Algorithm

For the detail of the algorithm, I will introduct it in five part.

2.3.1. Read data. When we read Data, I read the data from the file given, and storage the graph in two matrix, both two matrices can regard as the adjacency matrix of the graph, but on matrix is for the cost and one matrix is for the demand.

Then, I will storage all the information of the edges that have demand, including the two nodes and the cost. Why we need this set? When we do pathscanning, we need know when we have satisy all the required edges, the node is used to get the destination point and the next depot. The cost is used to calculation the final cost. Because for the required edge, we can't use the adjacency matrix after floyd. Because for the two node of the required edge, there may be other route between two nodes with less cost than the cost of the edge, then the value in the matrix will be the least coat, but we must use the cost in the edge. The last step is do floyed to the adiacency matrix.

2.3.2. pathscanning. After read data and storage, I will do the pathscanning and get the initial population. When I do the pathscanning, I will use the set that I storage the required edges. In each step, I will scan the set and choice the best edge. Firstly, I will choice the closest one, if there is many node have the least cost, I will apply four rules randomly.

2.3.3. .

3. Empirical Verification

3.1. Design

3.2. Performance

3.3. Result

3.4. Analysis

[?] [?]