Integer Programming Assignment 2

Solving the Vehicle Routing Problem (VRP) faster

Due Monday 15 June by 9:00am

This assignment is marked out of 10.

Improve your code for solving the VRP, i.e., <your upi>_veh_rout_func.py, by adding advanced branching and heuristics to close the bound gap for VRPs as quickly as possible. Marks are allocated for this assignments as follows:

- 1 paragraph explaining advanced branching used = 2 marks
- 1 paragraph explaining heuristics used = 2 marks
- 1 paragraph explaining experiments performed to determine final options = 2 marks
- Performance of <your upi>_veh_rout_func.py (including defined options) on 8 randomly generated files = 4 marks (1/2 for each file)

I encourage you to use figures and tables if they help your explanatory paragraphs.

As in IP Assignment 1, the randomly generated files are created by changing the seed on the example files and potentially changing the maxdist and ncurr values for examples #2 and #3 respectively. For every file that your code:

1) produces the correct answer for
2) within a reasonable time frame

you will get ½ a mark. Reasonable time frame = less time than the model answer code (which will not be highly tuned).

Hints

1) The following code snippet was found via Google and partitions a list, e.g., of location, into n pieces.

```
def partition(list_in, n):
    random.shuffle(list_in)
    return [list_in[i::n] for i in range(n)]
```

2) The following code snippet can sort a dictionary by value, e.g., assignment values.

```
s = {k: v for k, v in sorted(w.items(),
                             key=lambda item: item[1],
                             reverse=True)}
```

By looking at the (i, j, k) assignment variable values in descending order and adding i and j to vehicle k's route only when it "makes sense" (you will need to decide what is sensible) you can use a branch-and-bound node's solution to guide a heuristic.

3) You are welcome to use heuristics that you find for VRPs online.