Delivery Vehicle Routing System

- **Due:** *11:59 pm 27/10/2024 (End of Week 12)*
- **Contributes** 50% of your final result
- **Group Assignment – Group of 2-4 students**

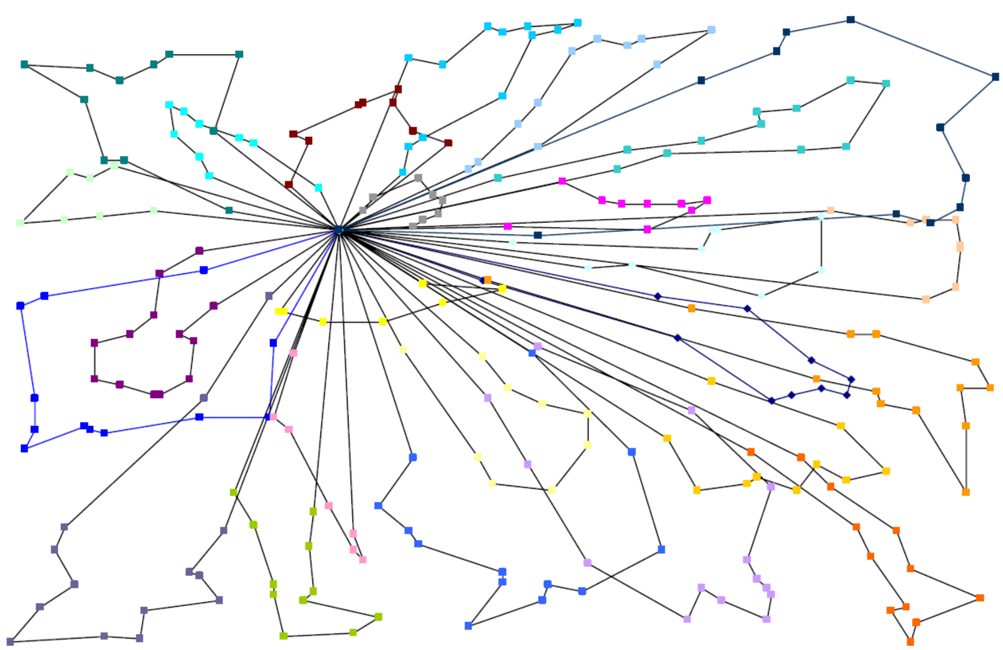
Summary

You need to implement and demonstrate a system for the Vehicle Routing Problem. The description of this problem can be found at: https://developers.google.com/optimization/routing/vrp
This is a well-known problem in both academia and industry (see also the following page: https://www.altexsoft.com/blog/business/how-to-solve-vehicle-routing-problems-route-optimization-software-and-their-apis/). You will develop a solution for a couriers company with a number of delivery vehicles. There are many versions of this problem but we will start with the basic one and suggest a number of extensions to enable the student to do extra research to achieve higher marks.

The basic system will assume that all parcels to be delivered are located at the company warehouse and all delivery vehicles also depart from the company warehouse and return back to the warehouse after delivering all their parcels. It will involve the following roles:

- **Master Routing Agent (MRA)**
  - Collects capacity constraints from other agents (i.e. from delivery agents)
  - Receives as input the list of parcels to be delivered to the customers' locations
  - Produces the routes for all the delivery vehicles
  - Sends the individual routes to delivery agents
- **Delivery Agent (DA)** (***any number***)
  - Send its own capacity constraint to the MRA
  - Receives the individual schedule

A solution assigns a route to each DA so that all parcels will be delivered and the DAs finish their routes at the warehouse. The MRA aims to find the optimal solution.



Constraints:

- The MRA can use any search/optimisation technique from the unit (e.g. GA, ACO, PSO, CSP etc.) or combinations of them. Note that you may use the Google OR-Tools library as a baseline solution to test your own solution but you must implement a search/optimisation technique of your own for this project. Another powerful open-source tool that you can reference to is OptaPlanner: https://www.optaplanner.org/learn/useCases/vehicleRoutingProblem.html. Please check out their videos for some cool applications of constraint optimisation solver. Again, you may use the OptaPlanner library as a baseline solution to test your own solution but you must implement a search/optimisation technique of your own.
- The DAs and the MRA must use an appropriately defined interaction protocols for communication/coordination. Please make sure that you'll provide a sequence diagram for the implemented service calls in your report.
- The agents can use any standard content language (for inter-agent communication)
- A GUI will be available for the user input, parameter settings and visualisation (and a configuration file for the defaults)
- You can assume that the cost/time to go from point A to point B is calculated according to the straight-line distance between A and B.

System requirements

- There are options to allow both the *automatic creation of the input list of delivery items (i.e., their locations) by taking the number of items* and *loading the input list of delivery items from a text file*.
- **Basic Requirement 1:** The capacity is specified by a number (i.e., the number of items the vehicle can carry). You cannot assume that the total capacity of all vehicles is equal or more than the number of items to be delivered. *That is, the solution must work when the total capacity of all vehicles is smaller than the number of items to be delivered*. A solution is optimal when the company delivers the largest number of items using the smallest total travel distances of all vehicles. Between the two objectives: **Number of items delivered** and **Total travel distances**, we prioritise on the **Number of items delivered**. *For instance, if Solution 1 can deliver 21 items with the total travel distance of 500 and Solution 2 can only deliver 20 items with the total travel distance of 400 then Solution 1 is better than Solution 2.*
- **Basic Requirement 2:** An additional constraint must be satisfied: each delivery vehicle v can only travel a maximum distance d_v.
- **Extension.Option1:** Vehicle routing problems with time windows (VRPTWs) – see: https://developers.google.com/optimization/routing/vrptw. You'll need to do your own research to figure out how to capture the input and other necessary parameters.
- **Extension.Option2:** We would like to build a system for Amazon-Uber: The MRA (e.g., Amazon) publishes to all DAs the list of items to be delivered and the DAs (drivers) will bid for the items they want to deliver. That is, a DA tells the MRA how much it wants to be paid for delivering item i by calculating its optimal routes for delivering the items it bids on. The MRA then awards the item to the lowest bidder.

Project requirements

- Source code maintained on Git based VCS (Github/Bitbucket/GitLab/…). You must provide read-only access to the tutor/lecturer
- Running illustrative demo of a working prototype (please refer to **Marking Scheme** for details on functionality that needs to be implemented)
- Project report (8-10 pages) that includes the following sections
  - Cover Page (with team details) and a Table of Contents (TOC)
  - Introduction
  - Overall system architecture

- Implemented interaction protocols,
- Implemented search/optimization techniques,
- Scenarios/examples to demonstrate how the system works,
- Some critical analysis of the implementation, and
- Summary/Conclusion.

➢ Presentation (video).

Marking Scheme

Requirements	***Mark***
Task 1: Basic interaction between DAs and MRA working fully according to a well-defined interaction protocol (with a clear sequence diagram). DAs can (a) submit their capacities according to the capacity specification, (b) MRA can receive requests from DAs and take an input list of delivery items to generate the route assignment for each DA, (c) MRA notifies DAs of their respective routes.	10
Task 2 (Basic Requirement 1): MRA uses any optimization technique to generate the optimal route assignment.	20
Task 3 (Basic Requirement 2): MRA uses any optimization technique to generate the optimal route assignment with the additional constraints (DAs' maximum travel distances).	20
GUI: Route Assignment (with associated cost for each route) Visualization Dashboard	10
Project Report	10
Project Presentation (Video, 6-8 minutes)	10
	80
Research Component (can be done by the whole team, a sub-team, or an individual) There are many potential extensions in this project. **Extension (Option 1)** and **Extension (Option 2)** are two examples but there are many more. Choose one and get your tutor's approval then complete it very well.	Up to 20
	100/100
You need to follow good programming practice (e.g., well-designed, well-structured codes with clear and helpful comments). Failure to do so get penalty.	**Up to -20**
You need to demonstrate the progress you make every week to your tutor by following a clear project plan the team have agreed with your tutor after the team has been formed. Failure to do so will get a penalty.	**Up to -50**

NOTE: Individual marks will be proportionally adjusted based on each team member's overall contribution to the project as indicated in the '*Who did what*' declaration.

Submission

At least one member of the team must submit the entire project (code + report) as a .zip file to Canvas by 11:59pm of 27/10/2024. Create a single zip file with your code and a working version of your system. Standard late penalties apply – 10% for each day late, more than 5 days late is 0%.

You must also provide your tutor read only access to your git repository within 1 week of forming teams.

The video **(6-8 minute duration)** should be submitted to Canvas by at least one member of the team by 11:59pm on Tuesday 29/10/2024.