INITIAL PLAN

FINDING THE OPTIMAL JOB ASSIGNMENT FOR A HOME-HELP SERVICE

CM3202 One Semester Individual Project – 40 credits

Author – Naomi Davidson

Supervisor – Richard Booth

Moderator – Natasha Edwards

PROJECT DESCRIPTION

REAL-WORLD CONTEXT

Companies offering home-help visits to elderly clients create rotas, with care staff starting and finishing each working day at their own homes and travelling to visit a number of client homes at given times during the day. These job assignments are often time consuming and complicated to create manually, and can result in the undesirable situation of carers covering more miles than necessary travelling backwards and forwards to visit clients living in locations across the area in which they work.

OVERALL AIM

The aim of this project is to develop and implement an algorithm using integer linear programming to solve the described real-world problem. This algorithm must create optimal job assignments for carers employed by a home-help company of any size, with a fixed number of support workers who must visit a fixed number of homes, spending a set amount of time at each home. To be optimal, these jobs will be assigned in such a way that the total distance travelled is minimal, as well as being fair, with each carer having an approximately equal workload. In addition, the assignments should adhere as much as possible to any patient preferences over the timings of the visits they receive (e.g., "not before 10am", "between 1pm and 2pm"), and take into account other factors such as recurring carer-client pairings, carer hours, etc.

PROJECT AIMS AND OBJECTIVES

CORE SOLUTION

The foundational aim is to build the core functionality of the algorithm: a given number of carers will each be assigned a weekly rota, with jobs distributed evenly between carers, and with total distance travelled for all carers minimised. The implementation should have the following inputs and output:

INPUTS

- Set of carers with home locations
- Set of clients with home locations
- Duration of visits
- Number of hours in a working day
- Number of days in a working week

OUTPUT

- Weekly rota for each carer which is efficient and fair to all carers, while fitting specified client needs.
  Efficient: total distance travelled by all carers has been minimised
  Fair: An equal share of work between carers

APPROACH

I aim to achieve this by:

- Defining the problem in plain English
- Understanding it as a specific variant of the vehicle routing problem (VRP) and/or multiple travelling salesperson problem (MTSP) optimisation problems
- Expressing this solution mathematically as an integer linear program (ILP)
- Implementing the ILP in Python, using Gurobi as a solver to efficiently explore the solution space

ADVANCED SOLUTION

Once the central functionality has been achieved, adjustments to the objective function and/or further constraints can then be added to the ILP to improve the relevance of the solution to the real-world business problem. The following are examples of potential inputs and outputs:

ADDITIONAL INPUTS

- Previous rota
- Unforeseen circumstance such as a carer needing to stay longer with client to wait for an ambulance
- Set of carers to include a corresponding number of contracted hours
- Set of carers to include limits on carer availability (working certain times of day, days off, etc.)
- Set of clients to include corresponding needs/preferences (if any) for visit day(s) and/or time(s)
- Set of clients to include corresponding needs/preferences for visits by a same-gendered carer
- Set of clients to include corresponding needs for number of carers per visit
- Set of clients to include corresponding appointment lengths

ENHANCED OUTPUT

- Weekly rota as in the core solution, but additionally configured to:
  - Have similarities to an existing rota (to ensure consistency for both clients and carers)
  - Share work fairly, according to varied carer hours
  - Explicitly account for travel time in rota
  - Increase the fairness of the solution by reducing the longest distance travelled by any one carer, without dramatically increasing the total distance for all carers

ETHICS

No research for my project will require ethical approval, and the data used to test the algorithms will be created by me and have no connection to real individuals.

RISKS

Although this project is grounded in a real-world example, its focus on the theoretical aspect of the problem and a lack of reliance on outside sources for data mean that it is relatively low risk.

One slight risk is that of proposed tools such as Gurobi not being suitable, or having issues using the software. This is a manageable risk as in the event of an issue with using Gurobi, a local search algorithm (e.g., genetic algorithms, tabu search) as encountered in the Combinatorial Optimisation model can be implemented to help explore the solution space.

Any risk of lost work due to hardware failures will be largely mitigated by a commitment to store all files on OneDrive, where they will be automatically backed up and available both online through OneDrive and locally on the system on which they were last accessed.

WORK PLAN

My supervisor and I have agreed to have a meeting every Thursday afternoon for half an hour to review progress from the week and to discuss and adapt plans for the following week(s) as required.

I am also aware of other university commitments, particularly in weeks 3, 10 and 11, and have considered these when devising my work plan.

The following table sets out the weekly milestones for this project:

Week	*Focus*	*Tasks/Milestones*	*Deliverables*
1	*Initial Plan*	Write initial plan Install Gurobi and obtain an academic license	Initial plan
2 *3*	*Background*	Use Google scholar to research relevant optimisation problems and solutions, e.g. Multi-depot MTSP Gain familiarity with Gurobi, or another form of local search Create sample datasets	Sample datasets
4	*Core Algorithm*	Formulate basic problem as ILP – choose variables, objective function, and basic constraints **Progress Review Meeting with supervisor**	Mathematical formulation of problem
5	*Core Implementation*	Create program using Python and the gurobipy library to solve the ILP given input, produce rotas as output Verify the effectiveness of the core solution	Basic program
6 *7* *8*	*Additional Features*	Identify interesting and/or important functionality to add to the program Update ILP and python code accordingly Repeat **Progress Review Meeting with supervisor (Week 7)**	Advanced program
9	*Evaluation*	Experiment with different datasets Investigate solutions for correctness Visualise solutions	Solution visualisations
Easter		*Overflow time for unfinished tasks*	
10	*Report writing*	Write up final report including background and literature review	Literature review
11	*Report writing*	Complete final report	Draft final report
12	*Final report*	Proof-read final report	Final report