Part 2 – Setting: Community Visitors Scheme program

<https://www.health.gov.au/initiatives-and-programs/community-visitors-scheme-cvs#cvs-for-aged-care-service-providers>

* There are old people (immigrants maybe) living in aged care facilities in Australia.
* Some of these people are not good at English, and they need language support in their own mother-tongue language.
* We have language support workers (service providers) who can speak one or many languages.
* These providers go on to help the old people (clients) in aged care facilities, for some duration of hours (1 hour, 2 hours, etc.).

**Providers**: language support workers

* Each provider can speak multiple languages (provide multiple types of service):
  + E.g.

|  |  |
| --- | --- |
| Type of service to provide (language) | Provider (worker) indicator |
| Service 1 (Chinese) | 3, 5, 6 |
| Service 2 (Korean) | 1, 3 |
| Service 3 (Italian) | 2, 4 |
| … |  |

* + Provider 1 can only provide service 2, provider 3 can provide services 1 and 3, etc.

**Clients**: Aged care facilities

* Each client (aged care facility) can have 1 or multiple people needing the language support service.
  + E.g

|  |  |
| --- | --- |
| Type of service to receive (language) | Client (facility) indicator |
| Service 1 (Chinese) | 1, 3 |
| Service 2 (Korean) | 2 |
| Service 3 (Italian) | 3 |
| … |  |

* + Client 1 requires services 1 and 3 (maybe client 1 has two old people requiring our service), client 2 requires service 2 only (maybe client 2 has only 1 old person requiring our service), etc.

**Report rough outline**

* Intro
* Literature review
* Part 1
  + MIP
  + Heuristics
* Part 2: aged care facility (CVS program)
  + **Extension** 1: Hourly
  + **Extension** 2: penalty
  + **Extension** 3: Client multiple service types
  + Model 4: Combine all extensions together, or, we can make extension 3 already contain extensions 1 and 2.
* Conclusion/summary

Until 10th next meeting:

* Weitian: write code for his multi-service part 2.
* Xueming: fix up the problem for provider working multiple times.
* Jean: Penalized version for part 1 extension, automate creating the excel file from Julia solution paths for visualisation.
  + Example excel file formats for vertices and edges are in overleaf <https://www.overleaf.com/project/608250528b54e9d3f82894dc>.
  + If it is easier for you, we do not have to make the diagrams on overleaf.
    - A possibility is using networkx in python?
      * E.g.) https://stackoverflow.com/questions/20133479/how-to-draw-directed-graphs-using-networkx-in-python
* Gyu Hwan: Work on comparing large size problems for part 1 vs heuristics.
  + Write pseudo code for heuristics algorithm to include in report.
* Claire: Writing report (intro, literature review, etc.).
* Rui: powerpoint structure, edit overleaf, etc.

**Week 10 meeting**

What-if analysis

* What if penalty?
* What if more providers, etc.
* What if hourly pay?
* What if multiple services?

Until 17th next meeting:

Presentation

* Intro
* Part 1: (not too long)
  + MIP
  + Solution diagrams
  + Heuristic
  + Run time comparison.
    - Table or a plot
* Part 2:
  + Extension 1: hourly – quick
  + **Extension** 2: penalty
    - Motivation: problems may not always be feasible, so we want a model that can more flexibly handle problem instances
    - Can solve feasible problems the same as part 1
    - Can also make infeasible problems feasible by using penalty
    - (more flexible)
    - Maybe show all constraints first, then focus on about penalty (additional/interesting) constraints bit more in detail in next slide
    - Show it can solve feasible problems in the same way as part 1 (can use data1, part 1 feasible, part 2 feasible -> same objective?), show it can also solve infeasible problems (part 1 infeasible, part 2 feasible, can have a graph).
  + **Extension** 3: Multiple services for clients
    - Motivation/example: real world example of **CVS program**
      * Clients are aged care facilities
      * Multiple language support
    - Client can have multiple service types required (language)
    - Talk about additional constraints compared to part 1
    - (or differences from part 1)
    - Maybe show all constraints first, then focus on additional/interesting constraints/differences from part 1 bit more in detail in next slide
    - Visualisation of working solution.
    - Explain in practical way
  + Extension 4: combining heuristics and MIP in part 1 (if it works)
    - Can boost run-time of MIP models.
* Improvements
  + Improvement: multiple service providers as well if we had more time
    - describe
    - Can mention we have an idea of modifying the data.
  + Combine extensions 1,2 possible too.