University of Melbourne

MAST90014 Optimisation for Industry

Home services Routing Optimisation

Group 8

Gyu Hwan Park 998174

Jinlu Luo 775511

Xueming Li 1129369

Weitian Zhang 1103145

Rui Li 1225896

Jean Ma 1028582

Contents

Abstract

The growing demand for home service expands the market for home service business. Verified Market Research [2021] approximated the global home service market in 2018 to roughly $282 billion and forecasted it to reach $1,133.4 billion by 2026 with annual increase of nearly 19% from 2019 to 2026. While making such a big cake, it is vital to ensure the scheduling of services is efficient and effective. In our project, we aim to minimise the hiring cost while serving all the clients. An initial Mixed Integer Programming (MIP) model and four extensions are explored. Different sets of constraints are implemented to reflect the aim of each extension. One extension is investigated using the matheheuristic approach. Another one modifies the objective to consider total hiring cost by hours. Late arrival penalty is also taken into account as one of the extensions. Lastly, we consider allowing multiple types of services requested by each client.

1 Introduction

The demand for home services has been rising rapidly in recent years. Various types of services can be categorised as home services such as home cleaning, home maintenance, appliance repair, plumbing, etc. Such growing demand expands the market for home service business. With the development of modern technology, home service can be easily booked and arranged by home service company online. Technically, the company receives the booking information of a specific type of service from the clients and employs the service providers accordingly. It is important for the company to ensure the schedule of the service is efficient and effective. In other words, the hiring cost of service provided is the main factor to be considered in our project. The hiring cost is associated with many aspects like the service duration, wait times and the travel time between locations. In some small-scaled companies, such time-consuming and complicated schedule work is created manually. As a consequence, companies might waste some of the resources and may not be able to schedule the service in an optimal way. Our project is designed to help the company improve the efficiency and effectiveness of schedule task and thus save more cost. This comes into a picture of a Vehicle Routing Problem (VRP) which is a mixed problem of integer programming and optimization.

In our project, we formulated our problem as a VRP with routing and time constraints and implemented linear programming to obtain the optimal solutions. The objective is to minimize the total hiring cost of service providers while serving all the clients.

In the first part of our project, we developed a Mixed Integer Programming (MIP) model to solve the problem. Various constraints are set as required in the problem specifications.

In the second part, we made four extensions based on our problem in part one. The first extension is to implement a matheheuristic that scales significantly better to larger-sized problems. This was done by combining a fast, approximate heuristic algorithm with the exact MIP model in part one. Different sizes of dataset were used for run-time comparison. The second extension is to use an hourly hiring cost of providers so that the objective function would also consider minimising the providers' wait times when they arrive at clients early. The third one is to apply late arrival penalty in the model if the provider does not arrive on time. The last modification allows each client to request multiple types of services throughout the day, while each provider is able to provide one type of service from the set of all service types.

2 Literature Review

The relevant literature mainly focuses on two areas of optimisation, which are cost minimization and travel time minimization. In our cases, we pay more attention on the former one. As suggested by Decerle et al. [2018], a mixed integer programming model using time constraints associated with time window restrictions can be used for VRP. They provided multiple types of home healthcare service to clients. Synchronization constraints were considered as providers sometimes provide multiple services simultaneously in a day. They pointed out soft time window is more flexible than hard time window which might cause an overload of penalty cost in the objective function when taking synchronization constraints into account. In our report, we adapt a similar approach in introducing penalty for late arrivals to clients that soften the strict time constraints and allow greater flexibility and feasibility.

As proposed by Kergosien et al. [2009], a client may request multiple services from different providers, given that each provider only provides one type of service. Further, we identified an Australian government's initiative, the Community Visitors Scheme (CVS) [Department of Health, 2021] that could be framed as a multiple-services VRP problem like Kergosien et al. [2009]. These cases gave us idea of extending our original problem to another version which allows the clients to request multiple services.

Other than the integer programming model, a heuristic approach like artificial neutral network (NN) can also be used to address the VRP. A NN model which uses a modified version of Hopfield network model is proposed by E et al. [1989]. However, they have found the heuristic approaches might not always give feasible and optimal solutions as expected. The stability of this approach needs to be improved. In this report, we implement a rather simple, intuitive greedy heuristic algorithm and explore the effect of using the approximate solution obtained as the starting feasible solution for the initial MIP model.

3 Problem formulation

We propose several assumptions to simplify the home services routing problem in the context of VRP.

- We consider the company 90014 Home services (14HS) that accepts service requests of a given type and hire service providers.
- Service requests have a start time, a location and a duration.
- All service providers are available at 14HS at their specified work start time.
- Travel times between all possible locations are known.
- A service provider can start its service as soon as it arrives at the customer's location.

- A service provider must reach clients on or before the agreed time.
- All service providers must come back to 14HS by a specific time of a day no matter the total work duration.
- A service provider starts working at the hour it becomes available.
- Only one type of service is provided.
- Our goal is to minimize the total hiring cost while serving all clients.

3.1 Problem instance generation

We were only provided with one feasible example data with 6 clients, 5 providers, and associated data values. We developed a systematic procedure to randomly generate feasible problem instances of any desired client set size. The procedure is as follows:

1. Define the coordinate of 14HS as $(0, 0)$ and the radius r.
2. Randomly generate the coordinates of any given number of clients within the circle of radius r centred at 14HS, $(0, 0)$.
3. Compute the euclidean distance between each pair of locations, including 14HS and all the clients. Round up the distance to the nearest integer. This forms the values of t_{ij} for each $i, j \in N'$.
4. For all providers, of given size, we randomly generate the hiring cost between 10 and 25. Start times are generated randomly between a given range with specified probability mass values. This forms the values of f_i and w_i for $i \in M$.
5. For all clients, of given size, we set the duration as 1 for easier feasibility. We randomly generate the service start time such that it is no smaller than $\min_{j \in N} t_{0j}$. This ensures all clients are reachable by the service start time. The service start times is also no larger than l - ($\max_{i \in N} t_{i0}$ + 1), so the provider is able to return back to 14HS by hour l after its last service with duration 1 hour. This forms the values of d_i and s_i for $i \in N$.

Note the above procedure guarantees feasibility of the generated problem if we set the sizes of the provider set and client set equal, and make all providers available at hour 0. In general, having a large probability mass of providers' start time being 0 will tend to make the problem feasible given that the size of providers set is large enough. Throughout the report, unless otherwise specified, we use a fixed providers set of size 100, start times generated from [0,1,2,3] according to probabilities [0.5, 0.2, 0.15, 0.15].

3.2 Mathematical programming model

We formulate the problem as an MIP as below.

Data

- l : Hour that all providers need to be back at 14HS by.
- $N = \{2, \ldots, n+1\}$: Set of n clients.
  - s_i: Service start time for client $i \in N$.
  - d_i: Service duration for client $i \in N$.
- $N' = \{1\} \cup N$: Set of $n+1$ locations (14HS and n clients) where 1 represents 14HS.
- $M = \{1, \ldots, m\}$: Set of m service providers.
  - f_i: Hiring cost of provider $i \in M$ for the whole day.
  - w_i: Earliest available staring hour of provider $i \in M$.
- t_{ij} : Travel times between any pair of locations $i, j \in N'$.

Variables

- x_{ijk}: Binary variable equal to 1 if provider k travels from location i to location j, 0 otherwise.
- u_{ik}: Potential variable for subtour-elimination constraints, associated with location $i \in N'$ for provider $k \in M$.

Model

We wish to search for the routes (assignment of clients in order) for each service provider such that the total hiring cost of service providers is minimised.

$$Z = \min \sum_{k \in M} \sum_{j \in N} f_k \, x_{ijk}$$

subject to the following constraints:

1. All clients are served exactly once by one provider:

$$\sum_{i \in N'} \sum_{k \in M} x_{ijk} = 1, \quad j \in N$$

2. Route continuity:

$$\sum_{i \in N'} x_{ipk} = \sum_{j \in N'} x_{pjk}, \quad k \in M, p \in N$$

3. Not all providers need to work (serve at least one client):

$$\sum_{j \in N} x_{1jk} \leq 1, \quad k \in M$$

4. At lease one provider needs to work:

$$\sum_{j \in N} \sum_{k \in M} x_{1jk} \geq 1$$

5. Miller-Tucker-Zemlin subtour elimination constraints:

$$u_{ik} - u_{jk} + |N| x_{ijk} \leq |N| - 1, \quad i, j \in N, k \in M$$

6. Providers must not arrive later than the start time of their first client:

$$x_{1jk}(w_k + t_{1j}) \leq s_j, \quad k \in M, j \in N$$

7. Subsequent service times must be respected:

$$x_{ijk}(s_i + t_{ij} + d_i) \leq s_j, \quad i \in N, j \in N, k \in M$$

8. All providers must return to 14HS before the ending hour l:

$$x_{i1k}(s_i + t_{i1} + d_i) \leq l, \quad i \in N, k \in M$$

9. Potential variable constraint:

$$1 \leq u_{ik} \leq |N|, \quad i, j \in N', k \in M$$

10. Integrality constraint:

$$x_{ijk} \in \{0, 1\}, \quad i, j \in N', k \in M$$

Implementation

As the objective function and all constraints are linear in terms of the variables, implementation of the MIP model was straight-forward using `JuMP`, with both optimizers `Gurobi` and `Cbc`.

Solutions

Solutions to the problem are defined as a sequence of (i, j, k) tuples, representing the set of travels from location i to location j for provider k. These represent the optimal feasible set of routes of selected providers from which the total hiring cost is calculated. Visualization of sample solutions are shown in Figure 1.

In Figure 1, the node labelled 1 represents 14HS. Each cycle that starts at 14HS

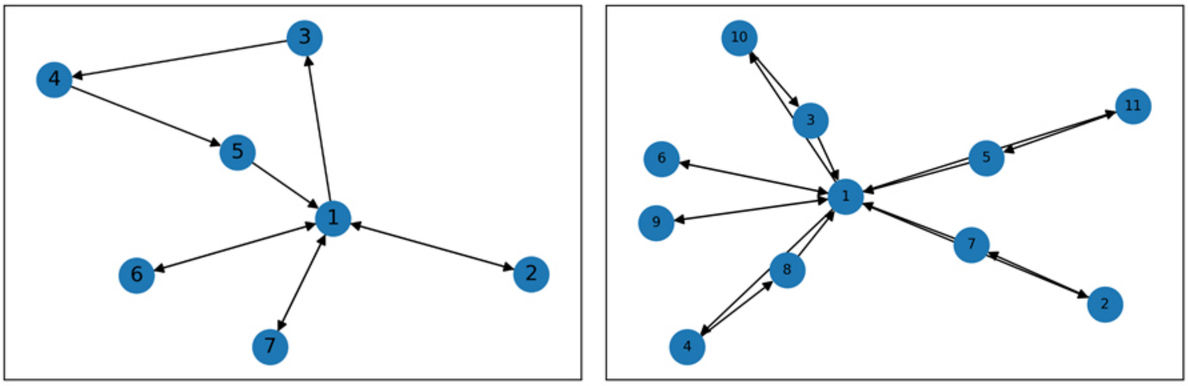

Figure 1: Optimal solutions for 6 clients (left), 10 clients (right)

and returns to 14HS represents the route of each hired provider. Note that in these visualisation plots, the coordinates of each node (including 14HS and clients) are randomly generated, so they do not respect the exact coordinates. On the left of Figure 1 we show an optimal set of routes that achieve an optimal cost of $45 for the example data 1 provided to us with 6 clients and 5 providers. On the right, is an optimal solution for the randomly generated problem with 10 clients with cost $65. The data for this example can be found in the supplementary code files submitted. An example of exact timelines of travel routes for feasibility checking is shown in Section 4.2.

3.3 Heuristic algorithm

In addition to the mathematical programming model, we also developed a heuristic algorithm that can solve the same problem. Algorithm (1) shows the pseudo-code for the heuristic algorithm.

The output of the algorithm are two arrays: 1) containing the served clients in order 2) containing the number of clients served for each provider in sorted array M. We use these arrays to extract the routes for each provider who is assigned to serve at least one client. Note that the heuristic algorithm is likely to provide a non-optimal (in terms of total hiring cost) set of routes of providers to serve all clients, but always a feasible one.

Algorithm 1: Heuristic algorithm

Data: $l, N, \mathbf{s}, \mathbf{d}, M, \mathbf{f}, \mathbf{w}, T$

Result: servedClients: array of served clients in order,
numServed: array of number of clients served for each provider in M

```
Sort M according to (f, w)
Sort N according to s
Initialize empty array servedClients = [ ]
Initialize empty array numServed = [ ]
for m ∈ M do
    location = 1
    i = 1
    for n ∈ N do
        if n ∉ servedClients then
            timeArrived = w[m] + t[location, n + 1]
            timeFinished = d[n] + s[n]
            if s[n] ≥ timeArrived, timeFinished ≤ l − t[n + 1, 1] then
                Append n to servedClients
                w[m] = timeFinished
                location = n + 1
                numServed[i] += 1
            end
        end
    end
    i += 1
    if length(servedClients) = |N| then
        break
    end
end
```

The key steps in the heuristic is the initial sorting of array of providers M, and array of clients N. M is sorted according to the hiring cost $\mathbf{f}$ first, and the ties for $\mathbf{f}$ are broken according to the providers' time of availability $\mathbf{w}$. This ensures the heuristic to consider assigning lower-cost, early-starting providers to clients first, which works towards our goal of minimising the total hiring cost while serving all clients. The subsequent procedures of the heuristic ensure that we assign feasible routes, where for each client, the provider assigned must not arrive later than the service start time and it must have enough time to return to 14HS after the service is finished.

The theoretical benefit of a heuristic compared to a full mathematical programming model is in the run-time. The heuristic is an approximate method that provides a heuristically reasonable feasible solution but not the optimal one, so it is likely to solve problems faster. The Big-O run-time of our heuristic algorithm is $O(q|N|)$ where $|N|$ is the number of clients and q is the number of providers who are assigned at least one client. Our main purpose of developing the heuristic is for implementing a matheheuristic version of the MIP model. Run-time and objective value comparisons of MIP, heuristic and matheheuristic are explored in detail in Section 4.1.

4 Extensions on Vehicle Routing Problem

We developed four extensions of simplified VRP discussed in Section 3. The first is a matheheuristic combining the MIP and heuristic, to better scale to larger problems and still give optimal solutions. This extension is directly related to the mechanical details of original problem, and does not change any constraints nor the objective function. The second is to modify the objective to minimise the total hourly hiring cost of providers, instead of constant daily-cost. Another extension is allowing service providers to arrive later than agree time but with an hourly late penalty. The fourth extension is to allow for multiple-service types requested by each client. These modifications expand the settings in which our models can adapt to.

4.1 Modification 1 - Matheheuristic

It is known that the VRP is an NP-hard problem. In our case, the run-time of MIP model will increase exorbitantly with the sizes of clients and providers. This motivates a more efficient method that scales better to larger-sized problem instances than the MIP model.

We developed a matheheuristic that combines Algorithm 1 and the original MIP model in Section 3. Algorithm 2 shows the pseudo-code for the matheheuristic.

The inputs to the matheheuristic are the standard data parameters. It outputs an optimized MIP model object which we can obtain the optimized objective and variable values from.

Algorithm 2: Matheheuristic algorithm

Data: $l, N, \mathbf{s}, \mathbf{d}, M, \mathbf{f}, \mathbf{w}, T$

Result: Optimized MIP `JuMP` object

1 Run the heuristic (Algorithm 1) using data inputs and obtain servedClients, numServed.

2 Set $x\_start_{ijk} := find\_x(servedClients, numServed, i, j, k)$ for each $i \in N', j \in N', k \in M$.

3 Set up the MIP model variables, objective and constraints shown in Section 3.1.

4 Initialize $x_{ijk} = x\_start_{ijk}$ for each $i \in N', j \in N', k \in M$.

5 Run optimizer on this MIP model with above initialization.

Key steps of the matheheuristic lie in lines 2 and 4 of Algorithm 2. In line 2, we compute whether provider k travels from location i to j in the solution obtained by the heuristic. In line 4, we initialize the start value of the binary variables x_{ijk} in the MIP as the corresponding x_start_{ijk} values. By doing so, we are warm-starting the MIP with the approximate solution obtained by the heuristic. This will eliminate the time required for the MIP to find its initial feasible solution, which could take a long time for large-sized problems.

4.1.1 Comparison of methods

We now compare the run-times and the objective values obtained from the four different methods: MIP with `Gurobi` optimizer, MIP with `Cbc` optimizer, heuristic and matheheuristic. Throughout the experiment, we used a fixed, randomly generated provider set of size 100. The initial problem instance had a randomly generated client set of size 10, and more clients were added incrementally to create larger problem instances. The sizes of client sets explored were: 10, 25, 50, 100, 120 and 150.

Table 1 in Appendix Section A shows the run-time values and the objective value achieved in the parenthesis. Figure 2 visually shows the run-times of four methods for increasing client set sizes. As expected, we observe exponentially-increasing trends in the run-times of MIP approaches, especially with the `Cbc` optimizer. The run-time of heuristic is the fastest, but the objective values (total hiring cost) obtained are between 5-17% higher than optimal. The matheheuristic, shown in red, has the advantages of both the heuristic and MIP. It is very fast compared to plain MIP and also always provides optimal objective values. It is in fact almost 32 times faster than running plain MIP with `Gurobi` optimizer. This experiment hence verifies that our implementation of the matheheuristic is indeed beneficial and allows us to scale to client sets of size 150 at least.

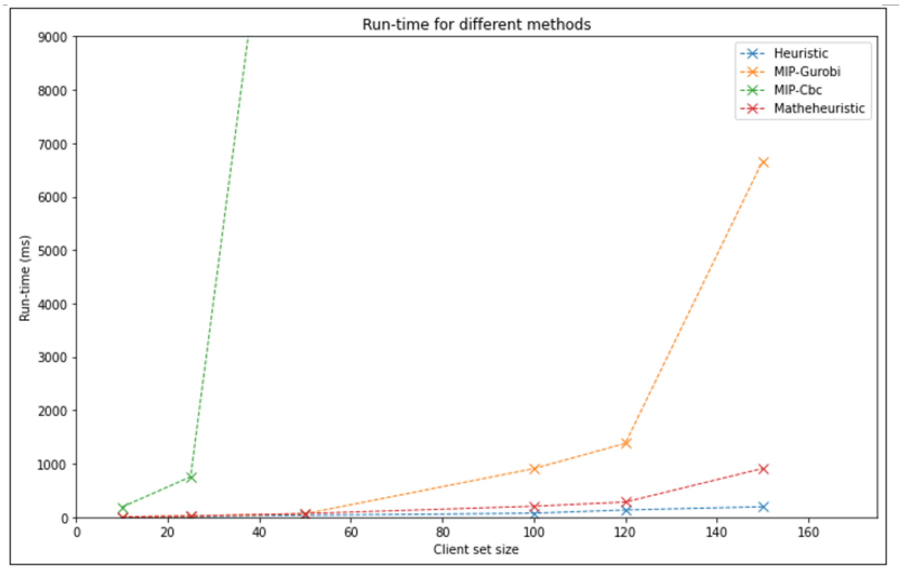

Figure 2: Run-time comparison plot. Matheheuristic is very fast and optimal. Heuristic is fastest but not always optimal. Plain MIP approaches are always optimal but very slow.

4.2 Modification 2 - Hourly cost objective

In this section, we modify the definition of data parameter f_i. Here, we define f_i to be the **hourly** hiring cost of provider $i \in M$. Rest of the data and variables remain unchanged.

Such modification of f_i implies that now the minimum hiring cost to serve all clients not only depends on the assignment of which providers to work but also the number of working hours for each. The working hours is defined as hour in which the provider returns back to 14HS and its work start/availability time. The motivation of such change is to more closely reflect real-life scenarios where service providers are paid by the number of working hours. We expect that this modification could change the optimal routes obtained from the original MIP model. As the time spent on waiting to start a service is now taken into account as well, we will prefer less waiting time. This is desirable as it will lead to a reduction in total hourly hiring cost to serve all clients.

Model

$$Z = min \sum_{k \in M} \sum_{j \in N} x_{1jk} f_k \left[\sum_{i \in N'} \sum_{j \in N'} x_{ijk}(t_{ij} + d_i + max(0, s_j - s_i - t_{ij} - d_i)) \right]$$

subject to the same constraints with the MIP in Section 3.1.
The new objective minimizes the total hiring cost (hourly) of service providers. That is, it minimizes the work hours of service provider which includes the travel times, service times and waiting times between serving clients if arrived early.

Implementation
The changed objective now involves a non-linear operation, $max(0, s_j - s_i - t_{ij} - d_i)$. This can easily be linearized, as all the terms involved in the operation are not model variables, but static data parameters. We can pre-compute the value of $max(0, s_j - s_i - t_{ij} - d_i)$ for each $i, j \in N'$ and replace the non-linear max operation in the objective. This results in a linear MIP model as desired.

Solution

Left of Figure 3 below illustrates an optimal solution obtained using the original MIP model from Section 3.1, where provider 1 needs to wait for 3 hours at client 1. The total cost, if we were to pay the providers for the number of worked hours, is \$290. With the new objective, we obtain the solution shown on the right. The set of providers and route assignments are changed. Provider 1 is now removed and provider 5 is introduced to work instead. Although providers 1 and 5 have equal hourly wage, provider 5 only spends 1 hour for waiting, and so we are able to serve all clients with only \$260. The exact timelines of travel routes for the solution on the right is provided in Appendix Section B.

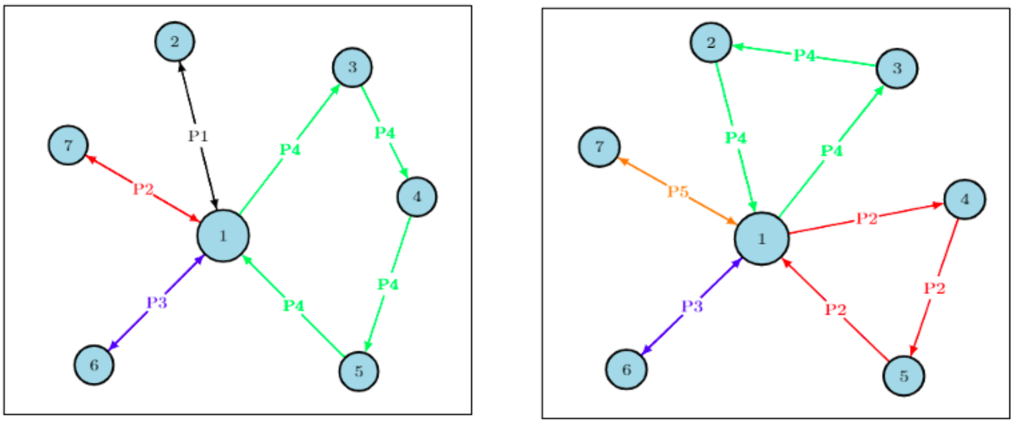

Figure 3: Optimal solutions for example data 1 using MIP from section 3.1 (left), MIP with new hourly cost objective (right)

4.3 Modification 3 - Penalizing late arrivals

For the model in Section 4.2 the providers must arrive no later than the service start times, and so the model is very strict and is not flexible. To allow for more flexibility and to provide feasible solutions to what was infeasible for the previous model in Section 4.2, we remove the need for the providers to arrive only before or exactly at the service start times. Instead, we penalize late arrivals to clients to encourage being on time, unless impossible to do so. Extra hourly penalty fee will apply if the service provider reaches the customers later than agreed time. This could be thought as a compensation paid to the client for being late.

The VRP model with late penalty inherits the same model format as the hourly model in Section 4.2. We introduce additional data parameters and variables below. Additional constraints related to these data parameters and variables are also included. Note that the constraints (6), (7), (8) from the MIP model in Section 3.1 that are related to service times are removed in this model.

Data

- P: hourly penalty cost.

Variable

- tl_{jk}: late arrival penalty hour of provider k at client j.
- te_{jk}: early arrival hour of provider k at client j.
- td_{ijk}: Actual departure time of provider k from node i to node j.

Model

The new objective not only minimise the total hiring cost(hourly) of service providers, but also minimise the late hour penalty cost. That is minimises the work hours of service provider and minimise the tardiness hour. The objective function becomes:

$$Z = min \sum_{i \in N'} \sum_{k \in M} f_k x_{i1k}(td_{i1k} + t_{i1} - w_k) + \sum_{j \in N'} \sum_{k \in M} tl_{jk} P \sum_{i \in N'} x_{ijk}$$

subject to the following constraints:

1. Either arrive early, arrive on time or late. If arrive on time or before agreed time, then late penalty must be zero, vice versa.

$$tl_{jk} \times te_{jk} = 0, \;\; j \in N', k \in M$$

2. Service provider depart from depot if they work.The actual departure time from depot is the first available time of providers.

$$td_{1jk} = x_{1jk} \times w_k, \;\; j \in N', k \in M$$

3. The departure time of provider k from node i to j is the sum of client available time, tardiness hour and service duration at client.

$$td_{ijk} = x_{ijk} \times (s_i + tl_{ik} + d_i), \;\; i \in N', j \in N', k \in M$$

4. All providers must come back to depot before time l.

$$x_{i1k} \times (td_{i1k} + t_{i1}) \leq l, \;\; i \in N', k \in M$$

5. The definition of actual arrival time at node j.

$$x_{ijk} \times (td_{ijk} + t_{ij}) = x_{ijk} \times (s_i + tl_{jk} - te_{jk}), \;\; i \in N', j \in N, k \in M$$

6. Integrality constraints.

$$tl_{jk}, te_{jk}, td_{jk} \in \mathbb{Z}_+, \;\; j \in N', k \in M$$

Implementation

The objective and constraints 1, 3, 4 and 5 involve products of pairs of model variables. This means that now we have a Mixed Integer Bilinear Program (MIBP) model instead of MIP. `Gurobi` optimizer internally linearizes such non-linear constraints using the Reformulation Linearization Technique (RLT) cuts technique [Gurobi Optimization, 2019]. This is a form of a cutting plain technique that helps solve such non-linear programs to global optimality [Bliek et al., 2014].

Solution

This model not only provides the same solution as the model in Section 4.2 for already feasible problems, but also provides a feasible solution to those problems that were infeasible by the hourly model in Section 4.2. The late penalty model is used to solve the data shown in Tables 2, 3, and 4 in Appendix Section C. We manually created this data so that it is infeasible by the hourly model, but feasible by the late penalty model. The late penalty model provides a solution with optimal objective value of $338 and the following late arrivals:

```
Provider 1 at customer 3 with tardiness hour 7.0
Provider 1 at customer 8 with tardiness hour 4.0
Provider 1 at customer 10 with tardiness hour 2.0
Provider 6 at customer 6 with tardiness hour 1.0
Provider 6 at customer 9 with tardiness hour 2.0
Provider 7 at customer 2 with tardiness hour 7.0
Provider 7 at customer 5 with tardiness hour 2.0
Provider 7 at customer 7 with tardiness hour 5.0
```

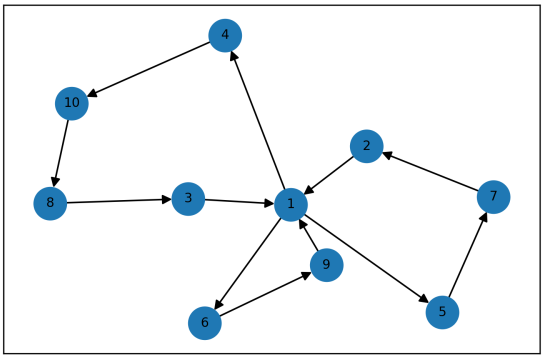

Figure 4: Optimal solution for example data 4.3 using model with penalty relaxation

The set of routes obtained is visualized in Figure 4 above.

4.4 Modification 4 - Multiple service types

Our last extension is motivated by a real community service program called the Community Visitors Scheme (CVS) [Department of Health, 2021], provided by the Department of Health. One of the aims of CVS is to improve the life quality of multicultural elderly people living in aged care facilities by arranging regular visits to provide language support in their mother-tongue language. Imagine a situation where in an aged care facility, there are multiple elderly people from different language backgrounds. Such aged care facility may request support in multiple languages to account for all residents. We have a group of language support workers as service providers who can satisfy this need. This situation can be framed as an instance of VRP that allows multiple-service requests from clients. We can think of each client as an aged care facility, requesting one or more types of services (language(s) support), and the providers are language support workers where one provider is able to provide one type of service (one language).

Additional assumptions that are posed under the multiple-services VRP model, compared to the hourly VRP model in Section 4.2, are listed below.

Assumptions

1. There is a fixed set of all service types, and one provider is able to provide one type of service.

2. Each client can request one or more types of services.
3. If multiple-services are requested by a client, these services all start at the same start time specific for the client. The duration for each service type can vary.

Data

- $\mathcal{O}$, this is the 14HS headquarters.
- l, the time of the end of the work day (when all service providers should be back at 14HS headquarters).
- $B = \{1, \ldots, b\}$, the set of service types.
- A^b, the set of clients who need service type b.
- H^b, the set of service providers who provide service type b.
- d_i^b, the service duration of b_{th} service type for client i.
- s_i, the service start time of client i.
- w_k^b, the time k^{th} b-type service provider is available at 14HS.
- c^b: The fixed cost of using one b-type service provider.
- g^b: The cost of using one b-type service provider per hour.
- t_{ij}^b: The time required for the b-type service provider to travel from i to j.

Variables

- x_{ijk}^b, equal to 1 if k^{th} b-type service provider travels from i to j, 4-dimensional variable.
- $u_{i,k}$: 'Potential' variable for subtour-elimination constraints, associated with location $i \in N'$ for provider $k \in M$.

Model

$Z = min \sum_{b \in B} \sum_{k \in H^b} \sum_{j \in A^b} x_{0jk}^b c^b +$

$\sum_{b\in B}\sum_{k\in H^b}\sum_{i\in\mathcal{O}\cup A^b}\sum_{j\in\mathcal{O}\cup A^b} g^b x^b_{ijk}(t^b_{ij}+d^b_i+max(0,s_j-s_i-t^b_{ij}-d^b_i))$

The new objective aims to minimize the total hiring cost of service providers (fixed and hourly). A difference from previous parts is the extra dimensions introduced by the service type sets for providers and clients. The constraints are as follows:

1. Ensure the each client should be served.

$$\sum_{j\in\mathcal{O}\cup A^b}\sum_{k\in H^b} x^b_{ijk}=1, i\in A^b, b\in B$$

2. b-type service provider that provide service to client j must arrive at j. After service (follow constraint 1), b-type service provider which provide service to j client must leave from j.

$$\sum_{i\in\mathcal{O}\cup A^b} x^b_{ijk}=\sum_{i\in\mathcal{O}\cup A^b} x^b_{jik}, i\in A^b, k\in H^b, b\in B$$

3. Not all providers need to work.

$$\sum_{j\in A^b} x^b_{1jk}\leq 1, k\in H^b, b\in B$$

4. Service providers should not later than start time of first client.

$$x^b_{0jk}(w^b_k+t_{0j})\leq s_j, i,j\in\mathcal{O}\cup A^b, k\in H^b, b\in B$$

5. Ensure each service provider can arrive at next client on time.

$$x^b_{ijk}(s_i+d^b_i+t^b_{ij})\leq s_j, i,j\in\mathcal{O}\cup A^b, k\in H^b, b\in B$$

6. Each service provider should come back to 14HS by l.

$$x^b_{i0k}(s_i+d^b_i+t^b_{i0})\leq l, i\in\mathcal{O}\cup A^b, k\in H^b, b\in B$$

7. Subtour elimination.

$$u_{ik} - u_{jk} + |A^b| x_{ijk} \leq |A^b| - 1, \quad i, j \in A^b, k \in H^b, b \in B$$

8. Integrality constraints.

$$1 \leq u_{ik} \leq |N|, \quad i, j \in A^b, k \in H^b$$

$$x^b_{ijk} \in \{0, 1\}, \quad i, j \in A^b, k \in H^b, b \in B$$

Implementation

Similar to the hourly model in Section 4.2, all the constraints in this model are linear. Only the $max(0, s_j - s_i - t^b_{ij} - d^b_i)$ in the objective function operation is non-linear. The same approach can be applied to linearize this operation. We can pre-compute all values of $max(0, s_j - s_i - t^b_{ij} - d^b_i)$ and explicitly substitute these values in the objective function.

Solution

We manually created a small feasible problem instance consisting of 6 clients, 15 providers and 3 different service types. Three clients have two-types of services requested, and the rest only single service type. The full data is shown in Appendix Section D.

Figure 5 below visualizes the optimal solution obtained by our multiple-services MIP model. We see that 8 different providers hired. For clients with 2 service types requested, different providers' routes are in different colours. We successfully assigned two providers to nodes 2, 6 and 7.

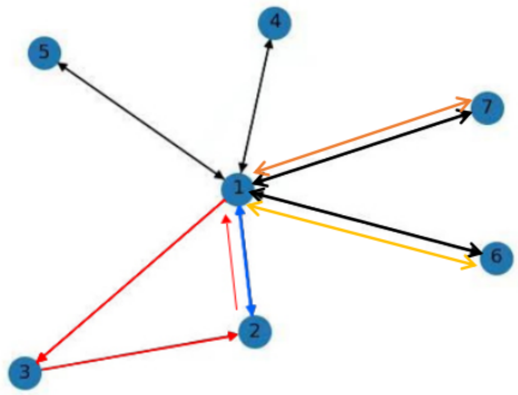

Figure 5: Optimal solution for obtained from multiple—services VRP model

5 Discussion

Limitations and Improvements

- In all our MIP models, we used the standard Miller-Tucker-Zemlin constraints for subtour elimination. However, as our VRP instances involve both the routing and time constraints, it would be possible to develop subtour constraints that directly use the data on durations, service times, travel times etc. We expect such change would reduce the run-time of MIP models greatly, as the dimension of constraints reduce by a factor of M.

- Our matheheuristic model is based on a simple heuristic algorithms. More advanced metaheuristics that can escape local optima have not been considered. To improve the performance and possibly the run-times, methods such as Tabu search and column generation should be explored.

- In the model with late hour penalty, we allow provider to have unlimited tardiness hours. However, this isn't allowed in real world and unlimited tardiness hour will lead to a loss of reputation. For future improvement, we can limit the maximum tardiness hour for more practicality at the cost of reduction in flexibility. It would be interesting to experiment the trade-off between such aspects.

- Our last modified model allows a client to request multiple services from different providers. In this case, each provider is restricted to provide a single type of service. However, in real case, for example, a language supporter could speak multiple different languages. The providers are likely to provide various types of services. Therefore, multiple services for both clients and providers can be considered.

6 Conclusion

In this report, we modelled the home services routing optimisation problem as a mixed integer programming model, with various extensions more related to real-life scenarios developed. The task required us to optimise the routes to minimize total hiring cost and serve all clients at the same time. We framed the problem as a variant of the classic VRP.

Besides the MIP model, a heuristic algorithm with successful implementation is also presented so that we can compare the behaviours between two methods and figure out their pros and cons. In general, MIP is good when dealing with small sized problem but becomes extremely slow at large scale. In contrast, heuristic performs well in terms of efficiency, however with less accuracy. Therefore, combining them together, we explore a matheheuristic algorithm which is fast and stable. In addition, exploring smarter metaheheuristic methods is a promising direction future research.

Our first modification of the plain MIP was changing the daily-paid hiring cost to hourly-paid. We do this because we noticed that there is a service provider after whose arrival instead of starting working immediately, 'waits' until the service start time. This extension successfully changed such solutions with large waiting times, and thus reduced costs greatly. Secondly, late penalty is added to allow for greater model flexibility. For example, sometimes a provider may not be able to arrive to the client on time due to traffic congestion or other unpredictable diversions. In such case, we could pay a compensation to the client for being late. With this new penalty term and deletion of previous early arrival constraint, we can solve problems that used to be infeasible using original MIP settings. We also suggest exploring suitable maximum late arrival times so that the model can become more practical. Motivated by an existing program called Community Visitors Scheme (CVS) provided by the Department of Health of Australian government, the last modification allows multiple services to be requested by clients. We assumed one provider offers one type of service and one client needs one or more service types. In the optimisation visualisation, we can clearly see that different service needs are satisfied indicated by different colours. However, in real life it is natural for one provider to supply multiple services as well. Thus this can also be explored in the future.

Overall, this home services routing optimisation problem was studied in the structure of VRP and satisfactory results have been achieved. Several extensions are made to better reflect real-life scenarios.

A Section 4.1.1

Table of run-times with achieved objective values in parenthesis. N/A if run-time exceeded 5 minutes.

Table 1: Run-times and objective values for the experiment

Client size	MIP-Cbc	MIP-Gurobi	Heuristic	Matheheuristic
10	188.202 (64)	5.045 (64)	3.719 (75)	5.123 (64)
25	754.438 (207)	18.495 (207)	13.065 (218)	21.242 (207)
50	17140 (494)	61.755 (494)	32.597 (543)	65.745 (494)
100	N/A	907.870 (1207)	73.098 (1272)	203.646 (1207)
120	N/A	1380 (1511)	133.502 (1517)	281.627 (1511)
150	N/A	6664 (1611)	194.021 (1744)	910.616 (1611)

B Section 4.2

Timeline of each provider generated by our code, for feasibility checking.

1. Solution description on data1 (Cost = 260)

    - Service provider 2 is hired for **5** hours at a cost of **50** and at:

        3.0hr: It is available at **HS14** and leaves to client **7**.

        5.0hr: It arrives at client **7**.

        5.0hr: It serves for **2** hrs.

        7.0hr: It finishes the service at client **7** and leaves to **HS14**.

        8.0hr: It arrives back at **HS14**.

    - Service provider 3 is hired for **3** hours at a cost of **24** and at:

        4.0hr: It is available at **HS14** and leaves to client **6**.

        5.0hr: It arrives at client **6**.

        5.0hr: It serves for **1** hrs.

6.0hr: It finishes the service at client **6** and leaves to **HS14**.

7.0hr: It arrives back at **HS14**.

- Service provider 4 is hired for **8** hours at a cost of **96** and at:

  0.0hr: It is available at **HS14** and leaves to client **3**.

  1.0hr: It arrives at client **3**.

  1.0hr: It serves for **1** hrs.

  2.0hr: It finishes the service at client **3** and leaves to client **2**.

  4.0hr: It arrives at client **2**.

  4.0hr: It serves for **2** hrs.

  6.0hr: It finishes the service at client **2** and leaves to **HS14**.

  8.0hr: It arrives back at **HS14**.

- Service provider 5 is hired for **6** hours at a cost of **90** and at:

  2.0hr: It is available at **HS14** and leaves to client **4**.

  3.0hr: It arrives at client **4**. It arrives early and waits until service start time.

  4.0hr: It serves for **1** hrs.

  5.0hr: It finishes the service at client **4** and leaves to client **5**.

  6.0hr: It arrives at client **5**.

  6.0hr: It serves for **1** hrs.

  7.0hr: It finishes the service at client **5** and leaves to **HS14**.

  8.0hr: It arrives back at **HS14**.

C Section 4.3

Table 2: General Data of Data-set 4.3

Work day duration	12
Number of providers	9
Number of clients	8

Table 3: Providers and Clients Data of Data-set 4.3

Provider	cost	15	10	8	12	15	7	6	9	18
	start time	0	3	4	0	2	2	4	1	0
Client	Start time	1	1	4	6	2	2	2	3	1
	Duration	1	1	1	1	1	2	1	3	1

Table 4: Distance Data of Data-set 4.3

0	3	1	1	1	1	2	2	2	3	1
2	0	2	3	4	3	2	3	2	4	1
1	2	0	1	2	3	2	1	4	4	4
1	3	1	0	1	4	4	1	3	1	4
1	4	3	1	0	1	2	4	2	2	3
1	3	3	2	1	0	1	2	2	2	4
1	1	2	3	3	1	0	4	2	1	3
2	1	2	2	2	3	2	0	3	1	1
4	4	1	3	3	4	2	2	0	2	3
1	2	2	2	3	2	2	1	2	0	1
1	3	4	3	1	2	2	1	3	1	0

D Section 4.4

Data

Work day duration: 8

Number of providers: 15

Number of clients: 6

Start time of each client: 4, 1, 4, 6, 5, 5

Number of service types: 3

The clients who require 1st type service: 2, 3, 6

The clients who require 2nd type service: 5, 7

The clients who require 3rd type service: 2, 4, 6, 7

The number of provider for each type service: 5

Start time of each provider:0, 3, 4, 0, 2

The cost for each type service: 15, 10, 8

Duration: the time of each service for each client. Each column represents the service type. Each row is a client.

Duration		
2	0	1
1	0	0
0	0	1
0	1	2
1	0	0
0	1	2

Distance:

0	3	1	1	1	1	2
2	0	2	3	4	3	2
1	2	0	1	2	3	2
1	3	1	0	1	4	4
1	4	3	1	0	1	2
1	3	3	2	1	0	1
1	1	2	3	3	1	0

References

C. Bliek, P. Bonami, and A. Lodi. Solving mixed-integer quadratic programming problems with ibm-cplex : a progress report. 2014.

J. Decerle, O. Grunder, A. H. El Hassani, and O. Barakat. A memetic algorithm for a home health care routing and scheduling problem. *Operations research for health care*, 16:59–71, 2018.

Department of Health. Community visitors scheme (cvs), 2021. URL https://www.health.gov.au/initiatives-and-programs/community-visitors-scheme-cvs.

W. E, H. J, and M. RC. A neural network algorithm for the multiple traveling salesmen problem. *Cybernetics*, 61:11–9, 1989.

Gurobi Optimization. Products of variables in mixed integer programming, 2019. URL https://www.youtube.com/watch?v=HS5HVCMORbo&ab_channel=GurobiOptimization.

Y. Kergosien, C. Lente, and J.-C. Billaut. Home health care problem: An extended multiple traveling salesman problem. *Multidisciplinary International Conference on Scheduling : Theory and Applications*, 2009.

Verified Market Research. Global home services market by deployment, by repairs & maintenance, by home improvement type, by geographic scope and forecast, 2021. URL https://www.verifiedmarketresearch.com/product/global-home-service-market-size-and-forecast-to-2025/.