Contents Page

Part 1. Mathematical Model

1.1. Objective Function and Variables

The goal is to minimize the objective function which considers the costs (in terms of time) of the services provided by a facility management company. The cost is calculated with the number of visits demanded per customer (h_i) annually, and the distance between the customer itself and the facility (d_{ij}). The facility allocated for each customer is assigned a binary variable X_{ij}, which means the cost between facility *j* and customer *i* is incurred only if facility *j* is allocated to customer *i.*

$$Min \sum_{i \in I} \sum_{j \in J} h_i \cdot d_{ij} \cdot X_{ij}$$

Decision variable**:** X_{ij}: New allocations for customer *i* in facility *j*.

$$X_{ij} = \begin{cases} 1, & customer\ i\ is\ allocated\ to\ facility\ location\ j \\ 0, & otherwise \end{cases}$$

1.2. Constraints

Constraint (2): Each customer *i* will be allocated to only one facility *j,* and *i* represents the set of demand nodes (customers in the case).

Constraint (3): The total demand for customer visits h_i allocated to each facility location *j* is no more than the capacity of the facility location c_j, and *j* represents the set of facility locations.

Constraint (4): The binary constraint and *j* represents the set of facility locations.

Constraint (5): For each customer *i,* the distance (measured as the driving time between customer *i* and facility location *j*) shortened by reallocating facility location is greater or equal to the reallocation cost itself (in terms of time)

Constraint (6): The binary reallocation variable that indicates if customer *i* is reallocated or not. If no reallocation is required, then a_{ij} would be equal to Xij , and consequently, Ri would be equal to "0" (no cost involved). Otherwise, Ri would be equal to "1" and the final cost is determined by *"T"*. The specific formula can be found in Appendix 5.1

1.3. Parameters

c_j: Capacity of location *"j".*

d_{ij}: Distance from customer *"i"* to facility *"j".*

h_i: Demand of customer *"i"* in terms of the number of visits required per year.

T: Reallocation cost in terms of time.

a_{ij}: Initial allocation for customer "*i"* in facility "*j"*.

1.4. Assumption

The statement describes certain assumptions that are being made for the Van Dorp mechanics' scheduling and allocation. It is assumed that each mechanic can visit only one customer per day and will not travel from one customer to another. The capacity of each factory and the demand of each customer is known and stable. The driving time between all customers and facilities is known and proportional to the distance, and external factors such as traffic and weather will not be considered. The reallocation cost of different customers is fixed and not affected by distance or factory. Finally, the initial allocation is feasible and known, satisfying the capacity constraints.

Part 2. Constructive and Improvement Heuristic Algorithm

2.1. Randomized Adaptive Greedy Adding Algorithm

The randomized adaptive greedy addition algorithm is a variant of the greedy algorithm for solving optimization problems. In this algorithm, it starts by initializing the facilities, their capacities, customer demands, costs, and an empty solution. The first step is to rank the customers in descending order (represented as d_{ij}), i.e., assigning priority to those with higher demand. A shortlist is then created, the size of which can be determined by a pre-determined parameter or by choosing a random value from a certain range. Based on this value, the number of customers to be included in the shortlist in this loop is determined. These customers will then be selected from the remaining highest-demand customers. Once the shortlist is created, the algorithm randomly selects a customer from this list to be assigned to the nearest facility. This process is repeated until all customers have been assigned. Random selection ensures that the algorithm explores different options and avoids duplicating customer selection. (The flow chart of GA is in Appendix 5.2)

The motivation of the randomized adaptive greedy adding algorithm is to balance the exploration of different options and the exploitation of customers with higher costs. Additionally, it can reduce the computation time with the best performance, so that it can efficiently allocate customers to facilities and minimize the total distance traveled.

2.2. First Improvement Algorithm

The first improvement heuristic algorithm starts by using the current allocation at Van Dorp and examining each demand node (i.e., customer) to check if reallocating them to a different facility could reduce the allocation cost (represented by $hi * di + T$). It randomly selects one customer from the whole dataset and uses a loop to check whether the current allocation cost is larger than the minimum allocation cost. If so, the algorithm then iterates through all facilities until it finds the first facility that makes the new allocation cost less than the current allocation cost and has enough capacity. Then the customer is assigned to this new facility. Even though the second or third feasible improvement may be much better than the first one, it always reallocates customers to the first one. This process is repeated until no further improvements can be made, and the final solution is returned. Notably, each customer may be repeatedly selected until all nodes can no longer be improved. (The flow chart of FI is in Appendix 5.2)

The motivation behind this algorithm is that by exploring the neighborhood of the current solution, the algorithm can potentially find a better solution by making small changes to the current allocation. Besides, the first improvement can reduce the scope and complexity of the problem by searching locally, rather than globally on a large scale.

2.3. Assumption for FI and GA Algorithms

The problem assumes that each customer is allocated only one open and operational facility and that the input data is accurate and reliable. It is also assumed that the local search algorithm will use a stopping parameter such that it stops when there is no improvement to the solution have been made in the latest cycle (when x=0).

Part 3. Application of Constructive Heuristic Algorithms

3.1. Data Preparation

The Randomized Adaptive Greedy Adding Algorithm is chosen as the constructive heuristic approach for this company. Despite the existence of other faster heuristics algorithms, this version has some literature validation regarding its performance in terms of a near-optimal solution using a reasonable runtime (Haket *et al*., 2020).

For the specific case of Prodnav Ltd., some assumptions are made as follows:

- ✓ Customers are grouped into local authorities; therefore, population data only considers "Unitary Authority" as customers.
- ✓ The demand from the customers is proportional to the population (all ages) with a rate of 0.01%. The total number of data is 374. Nevertheless, as the question only requires a study of the analysis of customers in Great Britain, it is necessary to remove customer data for Northern Ireland from the dataset, which ends up with 363 customers in the dataset.
- ✓ The distance matrix is calculated by the geodesic function with the coordinates of the customers and the facilities. The value is expressed as the distance (km) divided by the average speed of 60 km/hrs., resulting in time (hours).
- ✓ The constructive heuristic algorithm aims at a solution that starts with a null solution. So, it is not necessary to consider reallocation costs (both tangible and intangible) when all customers are allocated from 0.

3.2. Initial Solution Based on GA in Case

To prepare the data and perform the algorithm, the customers are sorted in descending order by the calculated demand, and a shortlist is created by extracting the top N random customers from this list. A randomly selected customer from this shortlist will be assigned to the closest possible facility. After the first customer has been assigned, it is removed from the full dataset and a second shortlist of the top N customers with random numbers is created. Then, one customer is randomly selected to find the optimal solution. The process will be repeated until the complete list of customers is assigned to the facility. The algorithm utilizes a random factor to decrease the likelihood of repeatedly proposing the same solution, while also prioritizing the reduction of travel time for high-demand customers. The resulting total travel time after applying the algorithm is reported to be 61,547.304 hours. The described methodology provides a potentially useful approach for optimizing the allocation of customers to facilities and may have broader applications in various domains.

Part 4. Improvement and Optimization

Based on the GA solution, we employed the result of GA using FI algorithm and the global optimal solution (OM). At this point, reallocation costs need to be considered and can no longer be ignored.

As the company does not provide any information for the reallocation cost, we adopted the parameter settings proposed by Haket *et al.* (2020) and set the reallocation cost per customer as 6 hours, assuming the company's distribution capacity and technological level remain constant. However, after including the reallocation cost, the outcomes for FI, OM, and GA remained the same (61,547.304) due to capacity limitations. Specifically, we discovered that multiple facilities had achieved full capacity, rendering further improvements unfeasible as the nearby facilities were unable to accommodate additional customers, leading to the local optimum.

This dilemma has been presented in the FI algorithm. When the algorithm selected a random customer and attempted to assign the customer to the facility with a lower cost, it found that the facility is already full, making reallocation impossible. Although swapping between two full facilities might lead to better results, this already defeats the definition of the FI algorithm.

For further improvement of the heuristic, we initially considered releasing capacity as a means of improving results. The table in Appendix 5.4 shows the number of reallocated customers with no new capacity, 10%, 20%, 50%, and 100% additional capacity, and the corresponding value of the objective function. For instance, when the capacity of all facilities increases by 10%, 31 customers are reallocated to better facilities and the result of FI drops to 44,791.334 hours. With more capacity relaxation, fewer customers are reallocated, because most customers are already at the optimal solution, which causes the rate of FI to slow down (Appendix 5.5).

Subsequently, we evaluated the impact of changes in reallocation costs on FI algorithm results based on 10% additional capacity. Reallocation cost decreases as the company's operational and distribution capacity improves, thus, we tested the reallocation cost from 6h to 0h (Appendix 5.6). The results demonstrate the reduction in the objective function as reallocation costs decrease. Therefore, it is crucial for the company to seek to reduce allocation costs in its future operations.

5. Appendix

5.1. The Objective Function and Constraint of Van Drop

$$\text{Min} \sum_{i \in I} \sum_{j \in J} h_i \cdot d_{ij} \cdot X_{ij} \qquad (1)$$

subject to

$$\sum_{j \in J} X_{ij} = 1, \qquad \forall i \in I, \qquad (2)$$

$$\sum_{i \in I} h_i \cdot X_{ij} \leq c_j, \qquad \forall j \in J, \qquad (3)$$

$$X_{ij} \in \{0,1\}, \qquad \forall i \in I,\ \forall j \in J, \qquad (4)$$

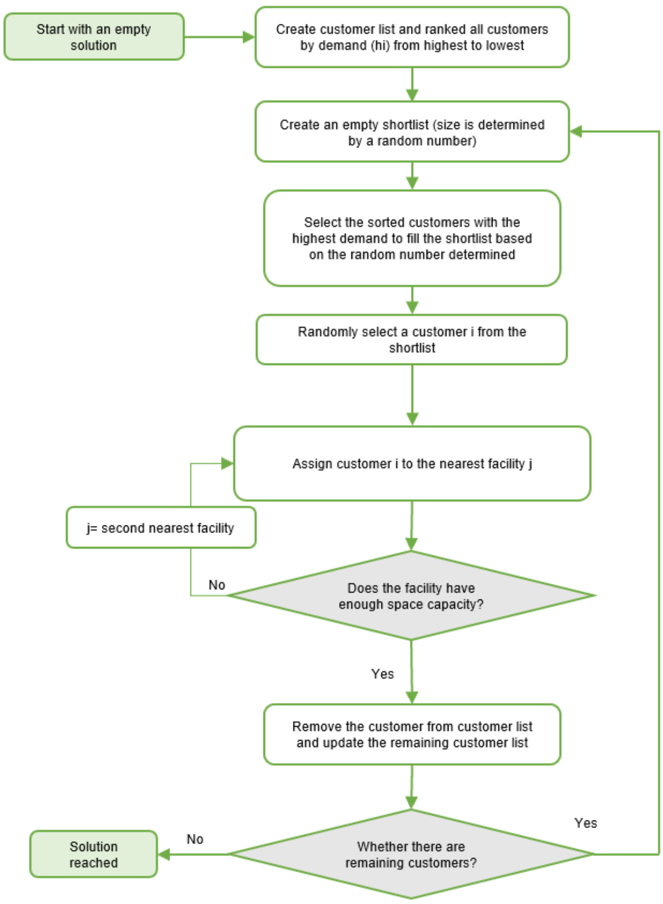
$$\sum_{j \in J} h_i \cdot d_{ij} \cdot (a_{ij} - X_{ij}) \geq R_i \cdot T, \qquad \forall i \in I, \qquad (5)$$

where

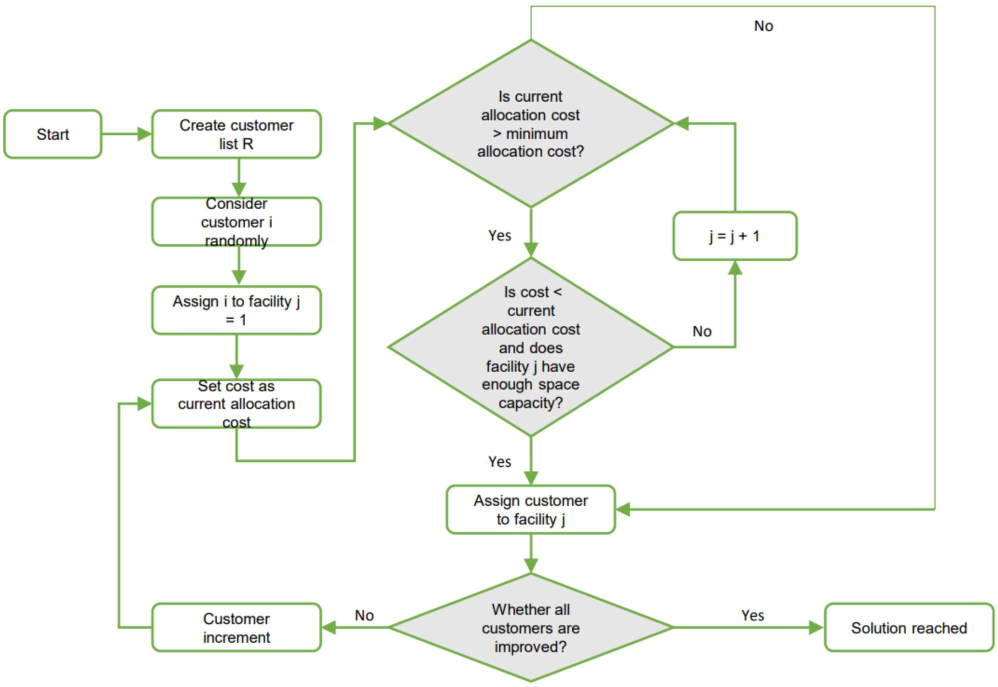
$$R_i = 1 - \sum_j a_{ij} X_{ij} \forall i, \qquad \forall i \in I. \qquad (6)$$

$$R_i = \begin{cases} 1, & customer\ i\ is\ reallocated \\ 0, & otherwise \end{cases}$$

5.2. The Flow Chart of the GA Algorithm



5.3. The Flow Chart of the FI Algorithm

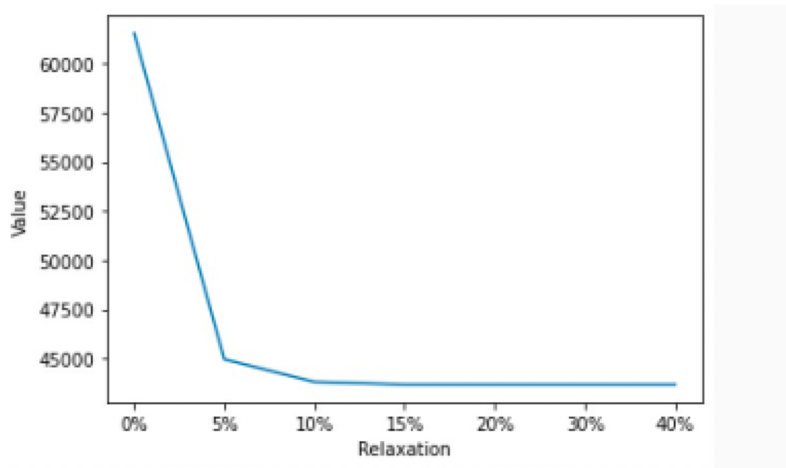


5.4. The Change of the Objective Value for Different Capacity Increments

When the Reallocation Cost is 6 hours

Capacity Relaxation	No Additional Capacity	10% (*1.1)	20% (*1.2)	50% (*1.5)	100% (*2)
Reallocated customers	---	31	3	0	0
Objective Value obtained by FI	61,547.30	44,791.33	43,717.16	43,691.98	43,691.98

5.5. Graph of Additional Capacity Versus Optimization Result



5.6. How do Reallocation Costs Affect FI Results

With an additional 10% capacity

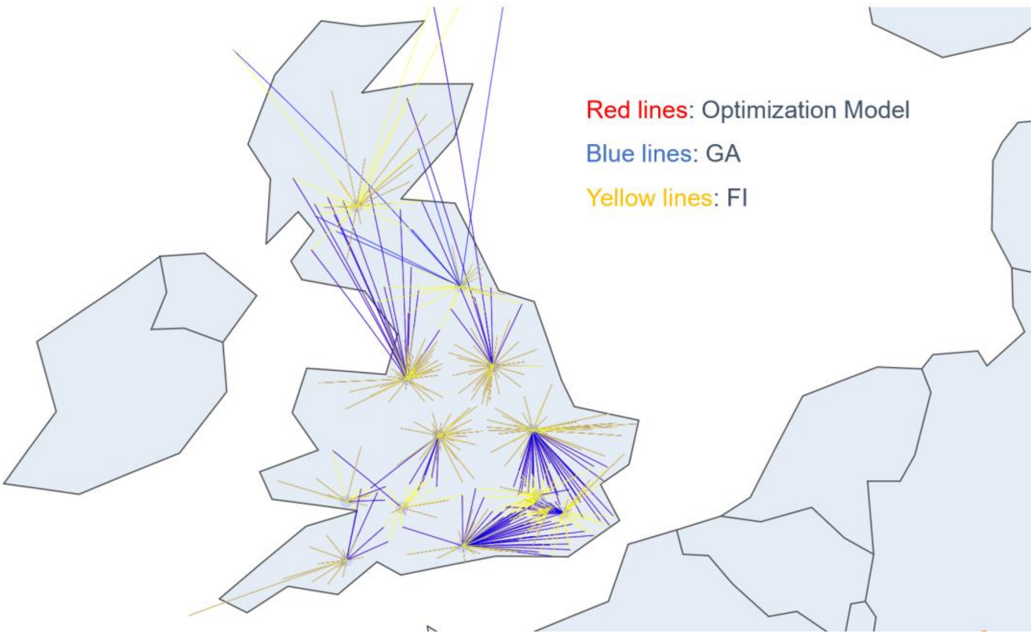
Reallocation Cost	0 hour	1 hour	2 hours	3 hours	4 hours	5 hours	6 hours
First Improvement Solution	44,665.12	44,711.33	44,714.20	44,837.11	44,844.44	44,844.44	44,791.33

5.7. Summary of Algorithm Results

```
Status: Optimal
Optimal Solution will be:  61547.30360324038
The result of GA algorithms is:
 61547.30360324042
The first improvement (No additional capacity) of GA algorithms value is:
 61547.30360324042
The first improvement (10% additional capacity) of GA algorithms value is:
 44791.3341816187
The number of reassigned is: 31
The first improvement (20% additional capacity) of GA algorithms value is:
 43717.158526220934
The number of reassigned is: 3
The first improvement (50% additional capacity) of GA algorithms value is:
 43691.98254924619
The number of reassigned is: 0
The first improvement (100% additional capacity) of GA algorithms value is:
 43691.98254924619
The number of reassigned is: 0
The improvement values of GA algorithm for First Improvement in the case of different % of additional capacity
are::
 16755.969421621718 17830.145077019486 17855.32105399423 17855.32105399423
The result of Random Algorithm is:
 63866.328789854
The improvement value of Random Algorithm doing First Improvement(No additional capacity) is:
 0.0
The number of reassigned is: 89
```

(Data source: Python Code of Group 13)

5.8. Comparison of Customer Allocation Under Three Algorithms



6. References

Haket, C., van der Rhee, B. and de Swart, J., 2020. Saving time and money and reducing carbon dioxide emissions by efficiently allocating customers. *INFORMS journal on applied analytics*, 50(3), pp.153-165.