Table of Content

1. Introduction

Multi-facility service providers often struggle to allocate their customers efficiently among their facilities. This inefficiency results in increased time, money, and carbon dioxide emissions, which are detrimental to both the environment and the company. Therefore, the research done by Haket et al. (2020) incorporated reallocation costs and considered several heuristics to demonstrate how companies can optimize their customer allocation without relying on specialized software.

2. Mathematical model

In the Van Dorp case, the objective function aims to minimize the total distance travelled by mechanics to serve their respective customers allocated to their respective facilities. The set of customers is represented by 'i', where i={1,2,...,4888}, and the set of facility locations are represented by 'j', where j={1,2,...,14}.

The decision variable Xij is binary, representing whether customer i is newly allocated to facility j or not.

The model (See Appendix: A.1) includes four constraints and one criterion to restrict the decision variables. Constraint (2) states that each customer can be allocated to only one facility. Constraint (3) requires that the sum of customer visits should not exceed the allocated facility capacity, for all facilities. Constraint (4) specifies that the decision variable Xij must equal either 0 or 1, where Xij = 1 indicates that the customer 'i' should be allocated to facility 'j', and 0 otherwise. Constraint (5) stipulates that a customer may be reallocated only if the reallocation savings (reduction in total travelling time due to the potential reallocation) exceed the reallocation costs captured by RHS. Criterion (6) defines whether reallocation would happen. When customer i is reallocated, R equals 1 and 0 otherwise. The model includes two parameters: T and Cj. T denotes the reallocation cost, measured in time. The parameter Cj denotes the capacity of each facility 'j', expressed as the maximum number of customer visits facility 'j' can accommodate in one year.

The model makes many assumptions. Firstly, routing is not considered. Secondly, the facility and customer number are fixed, and mergers or acquisitions of facilities are not considered. Thirdly, the reallocation cost T is assumed to be independent of the customer demand. Fourthly, the cost is measured in terms of time, including operational and administrative costs. Finally, the model assumes that each customer is served by one facility.

3. Heuristics Algorithms

3.1 Construction Heuristics:

All construction Heuristics are built on this basic idea: To start with an empty solution and iteratively add each candidate to the solution in the best possible way to optimize the objective function value. In Van Dorp's case, each candidate represents each unique customer, which is added iteratively to the solution set.

Construction Heuristics are relatively quicker in generating a solution and are often used when the data is too large to explore all possible solutions and find the optimal solution. There are

various versions of the Construction Heuristic. The focus of this paper is to explain the randomized adaptive greedy adding (GA) version.

The application of the GA algorithm on a simplified version of Van Dorp's case is illustrated (See Appendix: B.1). Firstly, all unallocated candidates are sorted in descending order of their greedy function values. Greedy function Value refers to the characteristic value of the candidates which has the maximum influence in the objective function value (in Van Dorp's case, the greedy function value considered is the demand value). Next, a random number of the top candidates are selected from the list. This selection is also referred to as Restricted Candidate List (RCL) (Wikipedia, n.d.). From this list, a candidate is chosen randomly and is added to the solution in the best possible way that optimizes the objective function (in Van Dorp's case, the candidate is allocated to the nearest available facility). This process is repeated until all the candidates are added to the solution.

GA has the combined advantages of the savings regret version and the random adding version. Moreover, the only difference between GA and saving regret is the initial selection of a random number of top customers and selecting a customer from that subset list, instead of directly selecting the top customer. While the Savings Regret Version will only give one final solution at every iteration, GA may give different solutions, which makes the final solution less dependent on the data provided. However, it still ensures that the probability to select high-demand customers in the first iterations will be higher than selecting customers with low demand. It also ensures that the possibility to reach the optimum solution is always there, in cases when the savings regret version would always give a local optimum solution.

In overview, using GA increases the ability over the savings regret version to explore the search space and avoid getting stuck in local optima.

3.2 Improvement Heuristics

All Improvement Heuristics, on the other hand, run on an existing solution, wherein it iteratively selects each candidate and tries to reconfigure it in the solution space to further optimize the objective function value. This process goes on until there can no longer be an improvement in the objective function value or until the maximum number of Heuristic iterations is reached, which is also called the stopping criteria.

The application of FI algorithm on a simplified version of Van Dorp's case is illustrated (See Appendix: B.2). To summarize, firstly, a customer is selected randomly from the list of customers not yet selected for improvement. Next, its allocation to each facility is compared to its existing allocation, in the order of the index of the facilities, and is reallocated to the first available facility that the algorithm finds will improve the objective function value, on the condition that the constraints are met. This process is followed for all the remaining allocated customers. Once all customers are analysed and possibly reallocated, the heuristic is run again, this time on the updated solution. This loop goes on until the stopping criterion is met.

One of the primary advantages of this version particularly for highly capacitated cases, over other versions like the "best" version, is that the solution space is searched multiple times, as the generation of a "slightly better" solution gives room for an even better solution. This ensures high likelihood of reaching the global optimum solution, compared to versions like the "best" version, which takes all allocation decisions at an early stage, especially when the facilities are highly capacitated.

4. Constructive Heuristics - Random Adaptive Heuristic

4.1 Data Preparation

The nominal capacities of the 13 facilities were given in “facilities.xlsx”. Based on the names of the facilities’ local authorities, the details of their longitude and latitude were fetched from ‘local_authority_districts.xlsx’ by matching names. Regarding facilities that could not match, their names were searched online to pair the local authority and stored the result in ‘facilities_geo.xlsx’. The “local_authority_districts.xlsx.” had details of Local Authorities, and all the local authorities in the list were considered as locations where the customers reside. ‘All Ages’ column in “ukpopestimatesmid2021on2021geographyfinal.xlsx” was retrieved as population. Demand was calculated by multiplying the population of each Local Authority by 0.1%, which indicates the total visits that are to be made to each Local Authority. Based on the Haversine formula, which utilizes the longitude and latitude of both facilities and customers to determine the distance in kilometres between each pair, a Distance Matrix table was created.

4.2 Result

The Pseudocode for the GA is described (see Appendix C.1). After running the code in python, each customer was able to be allocated to a facility. The total allocation cost comes about to be approx. 4.5 million (in Km) considering one-way travel, which translates to a total of 1.125 billion Grams of CO2 Emission considering two-way travel in a year, using an average theoretical value of 125 g/km (Kadijk et al. 2015). This, as shown later, is also the global optimum solution.
The distribution of allocation costs of different customers is shown (See Appendix: C.2). The distribution is favourably rightly skewed, indicating that most customers had the least allocation costs.

Moreover, Capacity relaxation table for GA Heuristic is also illustrated (See Appendix: C.3). It could be noted that capacity is a tight constraint parameter.

5. Implementation of First Improvement Heuristic and Optimization Model

5.1 Data Preparation

Regarding the preparation of additional data, at first the remaining capacity available for reallocation was calculated for each facility, which was served as input data for First Improvement Heuristic. Secondly, the allocation matrix that was returned by the constructive heuristic algorithm, was used as input data for both OM and FI.

The reallocation cost (T) was set to zero at first to focus solely on minimising the distance between customers and facilities. Moreover, Reallocation cost was expressed in “1/Km” units, as the input data serving the models were in “Km” units. With any average speed assumed for the problem, the reallocation cost could be converted into its original time unit.

5.2 Sensitivity Analysis

After applying Optimization Model and improvement heuristics on the solution generated by GA, it was found that the total allocation cost remained the same. This suggests that GA already provided the global optimal solution. Furthermore, sensitivity analysis was conducted by changing the capacities and reallocation cost. Increasing capacities led to a decrease in cost for both optimization and FI (See Appendix: D.1), but FI's performance remained worse than OM model for all capacity relaxation levels (See Appendix: D.2). Howsoever, the potential savings in allocation costs are substantial for Prodnav Ltd case and leads to a considerable reduction in CO2 emissions if the facilities are completely incapacitated, compared to the base level capacity. On the other hand, Changing the reallocation cost did not affect the total allocation cost for either of the two (See Appendix: D.3), which can be reasoned out because both gave the same total allocation cost that GA gave when T = 0.

6. Future Improvement

There are many limitations to the Heuristics, OM formulation, and data preparation settings.

Firstly, it was assumed that reallocation cost is independent of customer demand, however, this is not applicable in the real world. Moreover, with the assumption that each customer should be allocated only to one facility is simplistic and makes the solution suboptimal. A better approach would be to consider that a customer can be allocated to more than one facility. The Optimization problem could be restated accordingly (See Appendix: A.2). In this OM formulation, instead of a binary variable, Xij is a continuous variable ranging from 0 to 1, which represents the proportion of demand of customer ‘i’ that is allocated to facility ‘j’. Moreover, the parameter ‘Ti’ is different for different customers. Here, ‘Ti’ would refer to the Reallocation Cost to reallocate all the customer ‘i’ demand. Here, its coefficient value, which is the summation of ‘diff ‘ for all j, would represent the proportion of the customer ‘i’ demand that was reallocated. Further Improvements can be made to the model such as addition of penalty terms for customers who refuse reallocation.

The FI pseudocode in the research paper did not consider reallocation costs when considering reallocation decisions, but this error was corrected in Python. FI could also have been implemented differently to reach the global optimum solution, such as by allowing customer replacement. Particularly while iterating over all the facilities, the reduction in objective function value by swapping the selected customer with the customer currently having the highest allocation cost for that facility should have been evaluated instead. This procedure would have ensured that FI also reaches the global optimum solution, which based on Prodnav Ltd characteristics, was not giving a global optimum solution when tried for different capacity relaxation levels.

The Haversine formula was used instead of the actual road distance. Moreover, there are potential delays in travelling time due to traffic congestion during peak hours. Consideration of both could be achieved using integrative programming tools such as Google Maps API.

7. Conclusion

To conclude, this project presents a comprehensive approach to addressing the facility location problem by offering insights into various heuristics that can be used to tackle the problem and highlighting areas for further research.

Reference

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. Available at: https://doi.org/10.1287/inte.2020.1028.

Kadijk G, Ligterink N, Spreen J (2015) TNO report: On-road NOx and CO2 investigations of Euro 5 light commercial vehicles. Retrieved January 29, 2020, https://www.tno.nl/media/4980/nox_and_co2_investigations_of_commercial_vehicles.pdf

Wikipedia. (n.d.). *Greedy randomized adaptive search procedure - Wikipedia*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Greedy_randomized_adaptive_search_procedure

Appendix

A. Optimization model formation

A.1 Proposed Optimization Model Formulation in the research paper

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

subject to

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

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

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

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

where

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

A.2 Improved version of the OM model

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

Subject to,

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

$$\sum_{i\in I} h_i . X_{ij} \le c_j, \qquad \forall j \in J$$

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

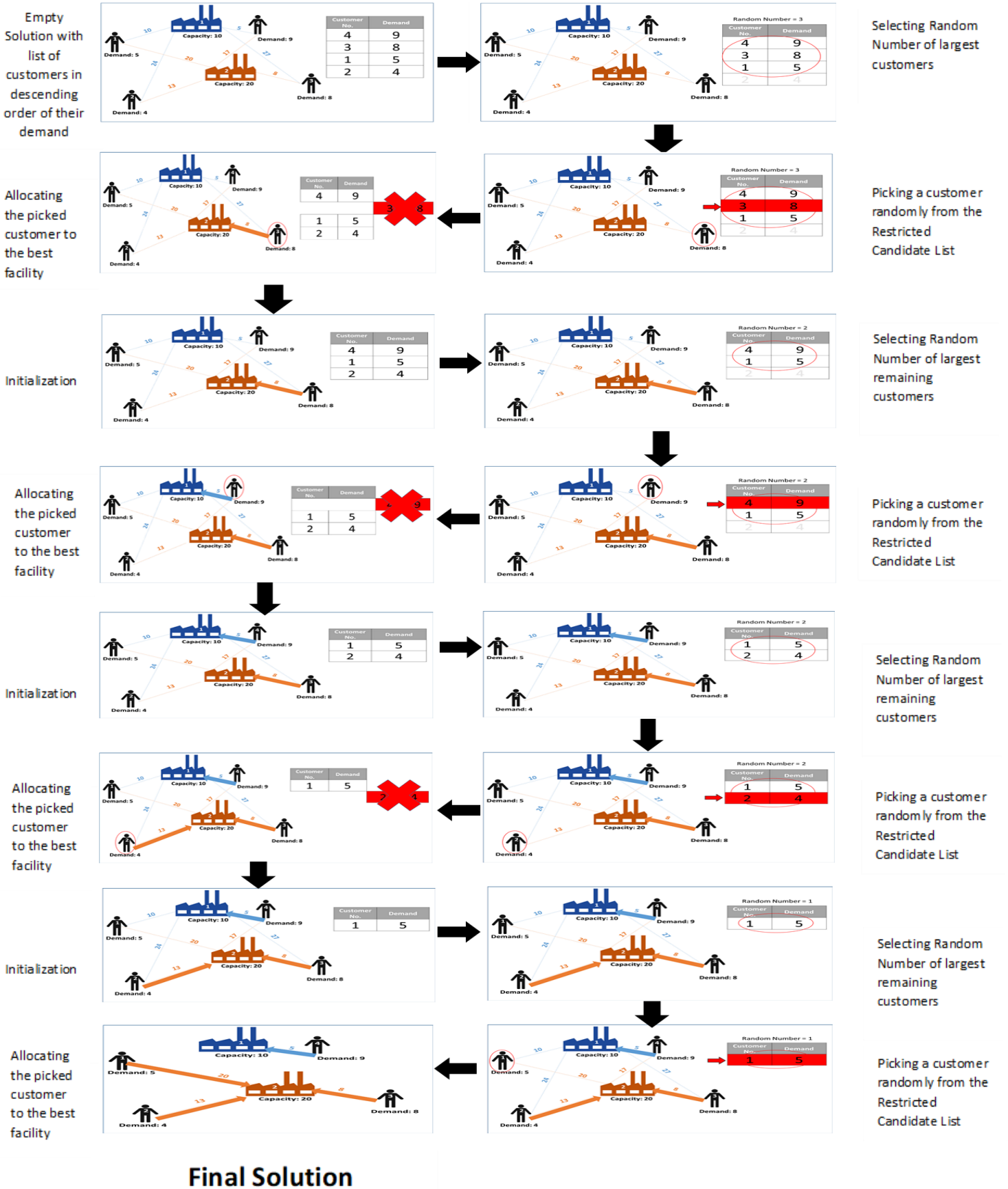
$$\sum_{j\in J} h_i . d_{ij} . (a_{ij} - X_{ij}) \ge T_i . \sum_{j\in J} diff_{ij}, \qquad \forall i \in I$$

$$diff_{ij} \ge 0, \qquad \forall i \in I, \forall j \in J$$

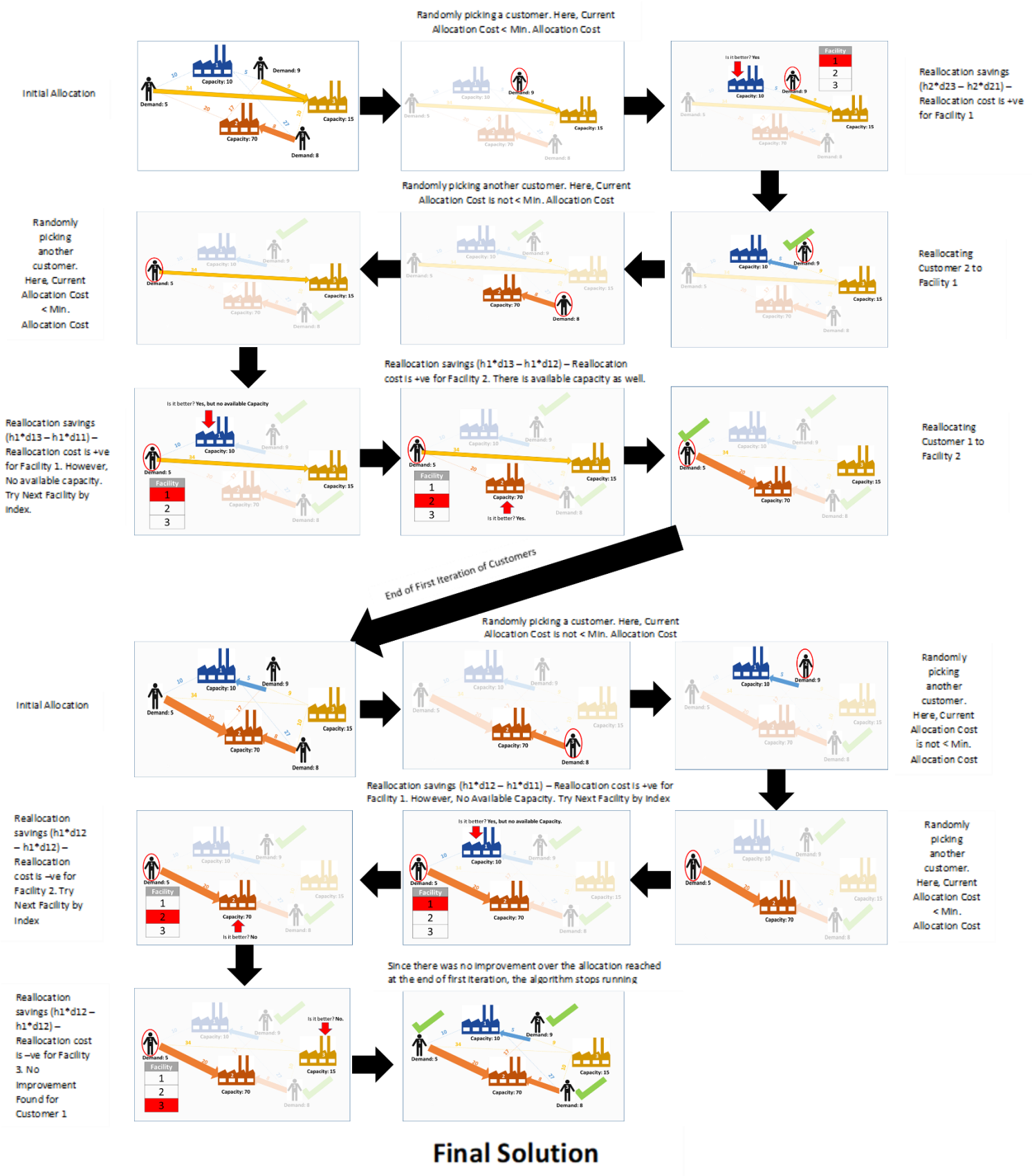
$$diff_{ij} \ge a_{ij} - X_{ij}, \qquad \forall i \in I, \forall j \in J$$

B. Illustrations for Heuristics

B.1 Greedy Adaptive Construction Heuristic Algorithm



B.2 First Improvement Heuristic Algorithm

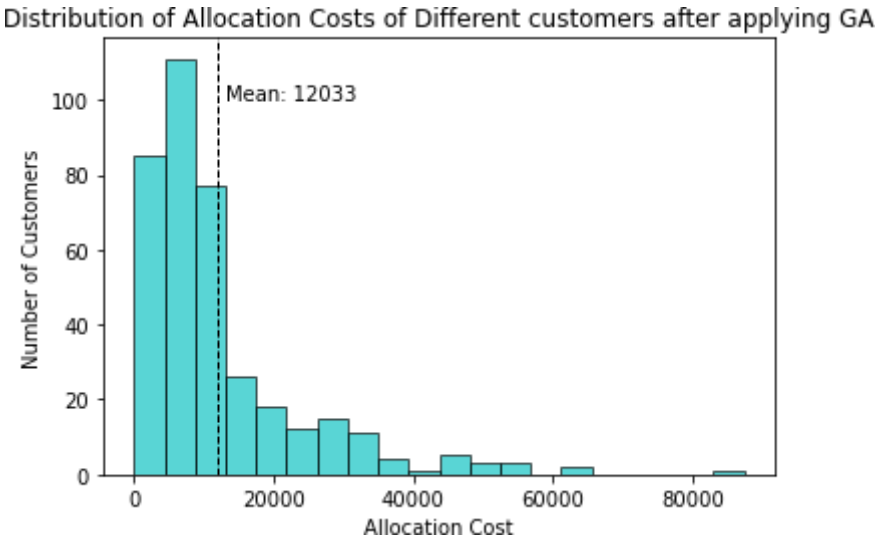


C. Construction Heuristic

C.1 Pseudo-code for Greedy Adaptive Construction Heuristic

```
Sub GreedyRandomAdaptive
1 Initiate capacities, demands, costs, empty solution and seed
2 Generate a copy 'l' of list of customers with their demand values
3 While 'l' is not empty, do
4       Sort 'l' in descending order of the demand values
5       Generate a random number 'n' between 1 and the number of rows in 'l'
6       Create Restricted Candidate List by taking the subset of the top 'n' customers from 'l'
7       Randomly pick a customer 'c' from the Restricted Candidate List
8       for j = 1 to k from lowest to highest cost of allocation do
9               if facility j has enough spare capacity then
10                      allocate customer 'c' to facility j
11                      delete customer c record from the copied list 'l'
12                      goto nextiteration
13              else: end if
14      next facility
15      if customer 'c' is not allocated then
16              delete customer c record from the copied list l
17              goto nextiteration
18      else: end if
19 nextiteration:
20 loop
21 return solution
End sub
```

C.2 Distribution of Allocation Costs of Different customers



C.3 Capacity relaxation table for GA construction heuristics

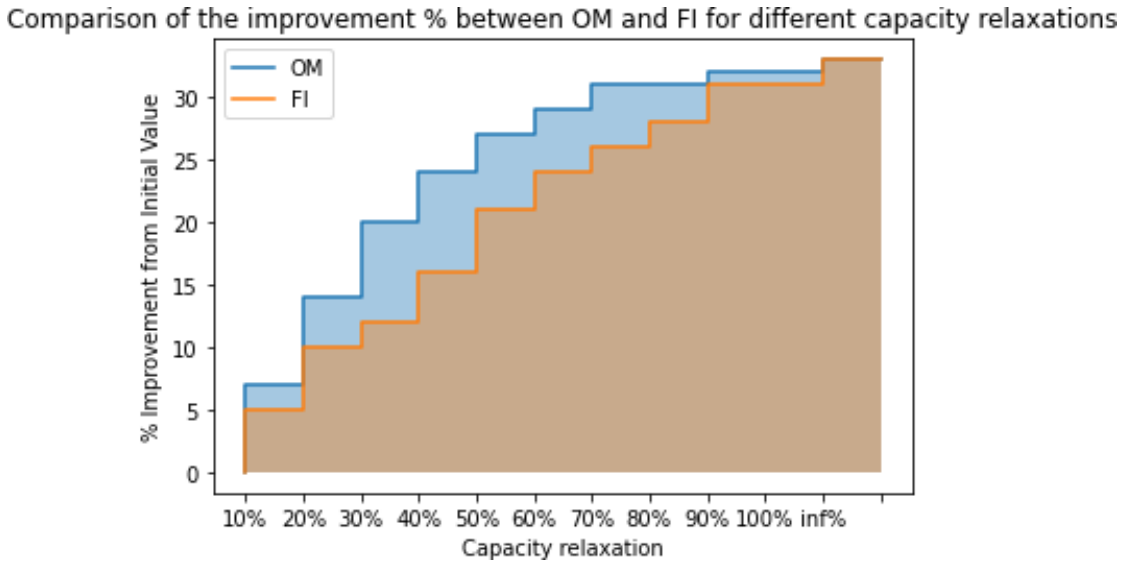
Capacity Increment	Total Distance Value	% change
0%	4500350	0%
10%	4229967	-6%
20%	4033393	-5%
30%	3844017	-5%
40%	3691241	-4%
50%	3519813	-5%

D. Sensitivity Analysis: Optimization and First Improvement Heuristic

D.1 Capacity Relaxation Table

Increment	Total Allocation Cost after incrementing Capacity for OM	% Gain over base case for OM	Total Allocation Cost after incrementing Capacity for FI	Number of FI improvement iterations	% Gain over base case for FI
0%	4500350	NA	4500350	0	NA
10%	4166261	7	4285174	1	5
20%	3853849	14	4050574	2	10
30%	3597447	20	3950368	1	12
40%	3411020	24	3771326	1	16
50%	3285229	27	3560732	1	21
60%	3191660	29	3440338	2	24
70%	3125411	31	3308433	3	26
80%	3086058	31	3225193	3	28
90%	3067897	32	3127664	4	31
100%	3056455	32	3108295	3	31
inf%	3018257	33	3018257	3	33

D.2Comparison of % Gain over base case between OM and FI for different capacity relaxation



D.3 Reallocation Cost Sensitivity Analysis

Increment	Total Allocation Cost after incrementing Reallocation Cost	% Gain over base case for OM	Total Allocation Cost after incrementing Reallocation Cost	Number of FI improvement iterations	% Gain over base case for FI
0%	4500350	NA	4500350	0	NA
10%	4500350	0	4500350	0	0
20%	4500350	0	4500350	0	0
30%	4500350	0	4500350	0	0
40%	4500350	0	4500350	0	0
50%	4500350	0	4500350	0	0
60%	4500350	0	4500350	0	0
70%	4500350	0	4500350	0	0
80%	4500350	0	4500350	0	0
90%	4500350	0	4500350	0	0
100%	4500350	0	4500350	0	0
inf%	4500350	0	4500350	0	0