Development of Efficient Electricity Distribution Model

Course: Algorithm Analysis and Design

Team AM

Aryan Kharbanda (2019101018)
Mayank Jain (2019101023)
Umang Srivastava (2019101090)
Arth Raj (2019101094)
Kshitij Mishra (2019111014)

Overview

Electricity companies face a lot of issues regarding the efficient power supply to all their customers as well as the customers have issues regarding their needs of power especially during peak hours.

Our aim is to develop and implement algorithms to improve this situation and implement better, renewable resources for generation of electricity at household level (two way transmission).

For companies to increase their profit margin, we implemented dynamic pricing.

SUGGESTIONS BY PROFESSOR AND MENTOR

- Develop a dynamic pricing model
- Develop a two-way electricity distribution model where the consumer can also supply in specific scenarios

CHALLENGES

1. Clustering of Houses
2. Load Shedding
3. Static and Dynamic Pricing
4. Two-way electricity Distribution

Clustering of Houses

Most of the residential colonies and areas in India are scattered and don't follow a specific distribution, while western countries have a well planned city layout and structure. Hence it becomes a problem in India to cluster houses in an area which follows a specific pattern of electricity consumption.

GOAL

To make clusters of houses which are near to each other in a city by making localities.

SOLUTION

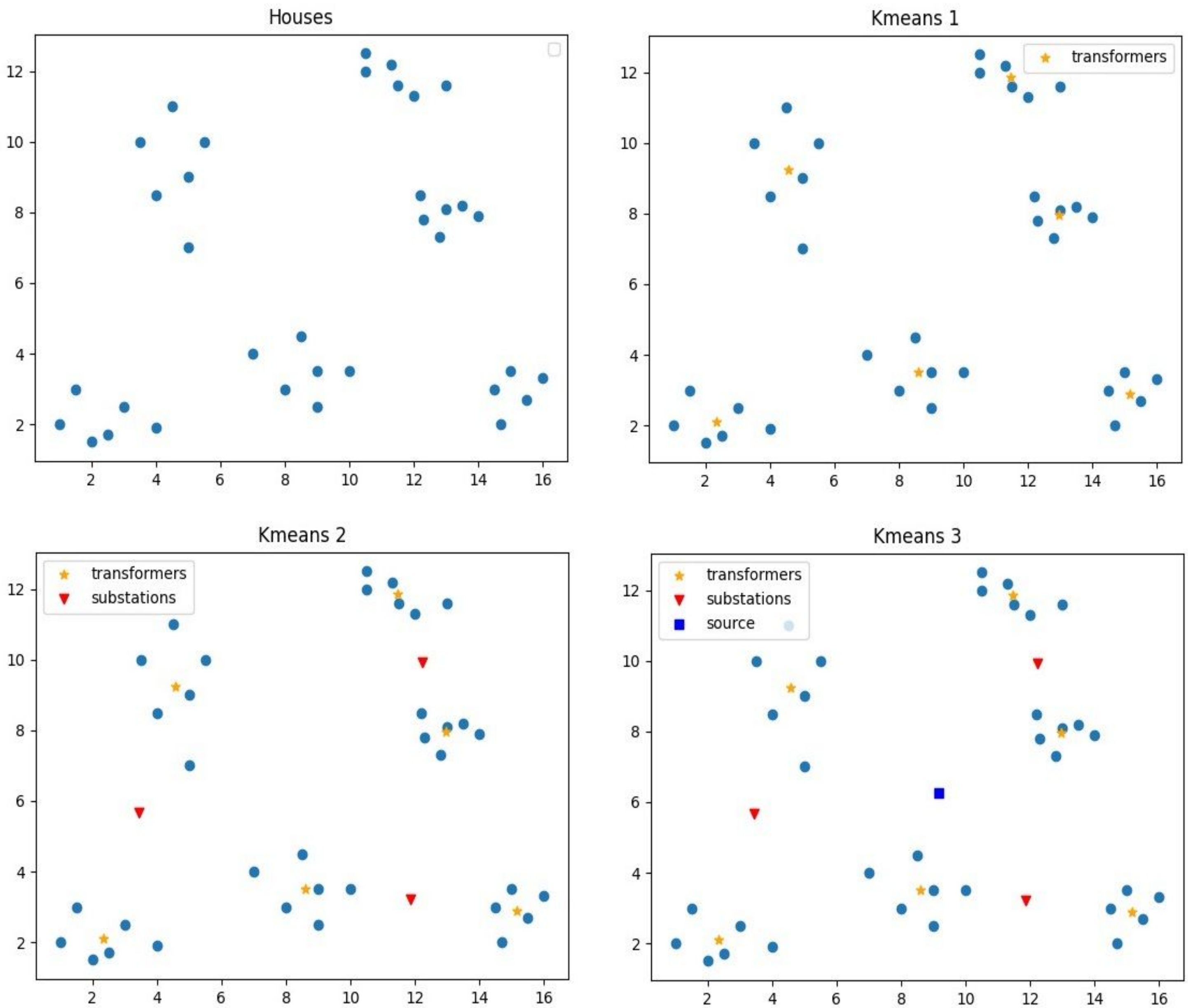
KMeans algorithm has been used to address the problem. We use geographical coordinates of a given set of houses as inputs. KMeans algorithm is an iterative algorithm that tries to partition the dataset into k pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. We categorize these houses into groups.

DESIGN

The algorithm works as follows:

- First we initialize k points, called means, randomly.
- We categorize each item to its closest mean and we update the mean's coordinates, which are the averages of the items categorized in that mean so far.
- We repeat the process for a given number of iterations and at the end, we have our clusters

OUTPUT


Houses
Kmeans 1
transformers
Kmeans 2
transformers
substations
Kmeans 3
transformers
substations
source

Load Shedding

PROBLEM

The second major problem related to electricity in India is unavailability of required amounts of electricity. While western grid and southern grid are surplus with electricity, northern and eastern grids are production deficits. This is solely due to large demand for electricity in households during peak hours of the day. This problem is dealt with by illogically cutting powers of localities. Can we find mathematical ways to find localities to manage this?

GOALS

- Finding an algorithm which helps in solving the above mentioned problem.
- Making the algorithm work for the clusters of houses so that load shedding can be done.

SOLUTION

Our first approach is the **Greedy algorithm** to minimise the number of houses to be cut-off. Meters in houses calculate the current power consumption and with this data, greedy can be applied. A better approach would be to maximise profit by using **Knapsack algorithm**. Values of weight and benefit will be analogous to current consumption and profit, which is calculated by multiplying the consumption with respective prices according to the meter installed. Arbitrary data : ₹4 for 2kW meter, ₹4.5 for 5kW meter and ₹6 for 10kW meter. Considering the larger scale on which these algorithms will be applied, it is better for the company to cut off the entire cluster (all houses connected to a transformer). This is carried out by Knapsack using weight and benefit as the cumulative consumption and profit for a cluster. This process is repeated after fixed intervals so we get new sets of localities which must be provided with electricity.

DESIGN

Algorithm: Greedy algorithm for load shedding

Input: Set S of n houses, such that house i has meter type value m_i with price p_i and consumption w_i . m_i has values from a set V. Total instantaneous available electricity is W.
Output: Number of houses that receive electricity N and profit as P. (arr[ ][ ] is taken as buffer)

```
N ← 0
P ← 0
for i ← 0 to n do
    arr[i][0] ← i + 1
    arr[i][1] ← w_i
sort arr[i][1]
for i ← 0 to n do
    if W ≤ 0 then
        return P, N
    N ← N + 1
    W ← W - arr[i][1]
    for j ← 0 to n do
        if arr[arr[i][0]][1] = m_j then
            P ← arr[i][1] * p_j
```

Algorithm: Knapsack algorithm for load shedding

Input: Set S of n houses, such that house i has positive benifit b_i which is profit and positive integer weight w_i which is consumption of the house; total available electricity for consumption is W.
Output: For w = 0,...., W, maximum benefit K[i][w] of a subset of S with total weight w.

```
for i ← 0 to n do
    for j ← 0 to W do
        K[i][j] ← 0
for i ← 0 to n do
    for j ← 0 to W do
        if i = 0 or j = 0 then
            K[i][j] ← 0
        else if w[i-1] ≤ j then
            K[i][j] ← max(b[i-1] + K[i-1][W-w[i-1]] , K[i-1][j])
        else then
            K[i][j] ← K[i-1][j]
```

Time and Space Analysis :-

Greedy Algorithm for load shedding

Time complexity = O(nlogn)

Space Complexity = O(n)

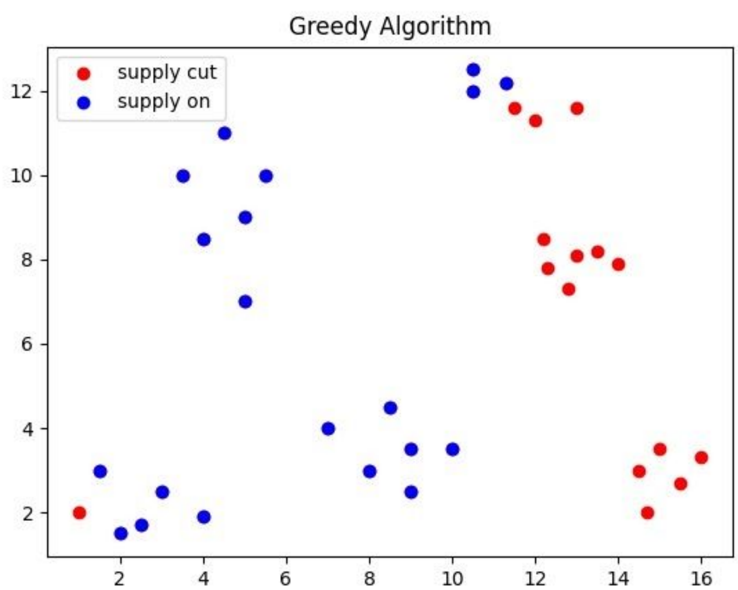
Knapsack algo for load shedding

Time Complexity (n*W) = $O(n^2)$

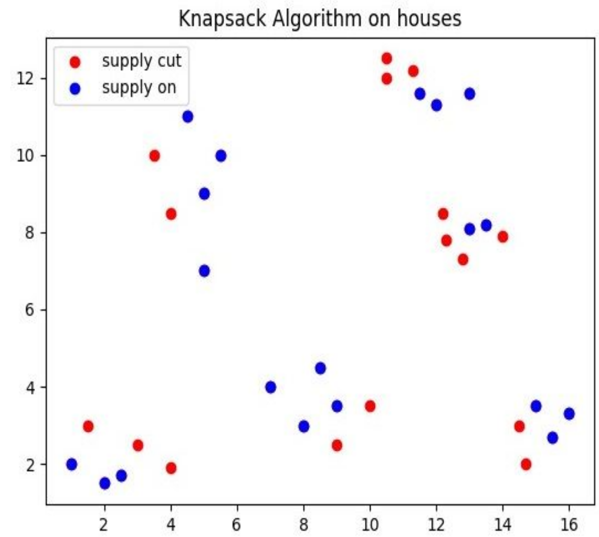
Space Complexity (n*W) = $O(n^2)$

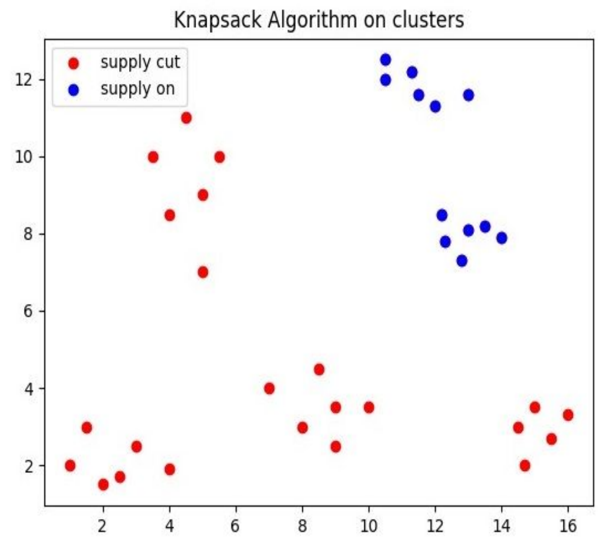
OUTPUT

GREEDY:



KNAPSACK:





SIMULATION For Load Shedding

We also created a simulation to check load shedding algorithms. It generates random values of average power consumption per hour in appropriate ranges for a day. The output is shown as follows.

According to Greedy Algorithm

Hour	Profit	Number of houses
00	2445.0	23
01	2531.5	19
02	2567.0	20
03	2492.0	21
04	2841.5	18
05	2420.5	17
06	2631.5	19
07	2713.0	19
08	2717.0	18
09	2535.0	17
10	2540.5	18
11	2484.5	17
12	2461.0	20
13	2386.5	21
14	2615.0	18
15	2582.0	20
16	2632.0	18
17	2589.5	20
18	2576.0	18
19	2550.5	19
20	2505.5	18
21	2533.5	18
22	2621.0	19
23	2602.5	22

According to Knapsack by house Algorithm

Hour	Profit	Number of houses
00	2803.5	15
01	3000.0	8
02	2992.0	12
03	2992.0	10
04	2968.5	14
05	3000.0	7
06	3000.0	7
07	3000.0	8
08	3000.0	8
09	3000.0	8
10	3000.0	6
11	3000.0	8
12	3000.0	9
13	3000.0	7
14	3000.0	7
15	2998.5	10
16	2992.5	12
17	3000.0	9
18	2994.0	12
19	3000.0	10
20	3000.0	8
21	2955.0	11
22	3000.0	9
23	2992.0	11

According to Knapsack by cluster Algorithm

Hour	Profit	Number of clusters
00	2513.0	2
01	2219.0	2
02	2474.0	2
03	2428.0	2
04	2186.0	2
05	2254.5	1
06	2459.5	2
07	1734.0	1
08	2133.0	1
09	1825.0	1
10	1786.0	1
11	1902.0	1
12	2671.5	2
13	2451.0	2
14	2569.0	2
15	2281.5	2
16	2176.5	2
17	2188.0	2
18	2137.5	2
19	2399.5	2
20	1816.0	1
21	1672.0	1
22	2443.0	2
23	2238.5	2

As we can see, the number of houses is maximum in greedy algorithm, but profit is lower.

But knapsack prioritizes benefit, so we get more profit but at the expense of more houses being cut.

Pricing Model

PROBLEM

The traditional pricing model followed for consumers is based upon the slots which depends on consumption. For instance, 10-100 units cost ₹x, then 100-150 units cost ₹y and so on. This slab based pricing method does not account for wastage of electricity as well as very uneven pricing based on consumption demands. It is logical to decide price based on consumption demands as different topography has different consumption patterns.

GOALS

- Develop 2 pricing models: Static and Dynamic.
- Base the pricing on the consumption pattern of a cluster rather than a fixed slab.

SOLUTION

We develop two pricing models: Static Pricing and Dynamic Pricing. Static Pricing includes different slabs that have a fixed price per unit and is used for houses that opt for it. Dynamic Pricing depends on various factors like the cost of production, fuel costs, infrastructure costs, etc. The city is divided into clusters of static and dynamic pricing, each applied to clusters based on the consumer demands.

Static Pricing vs Dynamic Pricing

DESIGN

Algorithm: Dynamic Pricing fo Electricity Consumption

Input: fuels = {coal, gas, diesel}, cost of fuel required for producing 1 unit and its percentage in production is C_i and P_i and where i ϵ fuels, hourly consumption of last 7*24 hours as A[168] and consumption of last hour as cons_last_hour and fixed price as fixed_price.
Output: Dynamic Price as dynamic_price for current hour

repeat
    for $j \leftarrow 0$ **do**
        for $k \leftarrow 0$ **to** $k \leftarrow 168$ **do**
            sum ← sum + A[k]
        average_consumption ← sum / 168
        for i **in** fuels **do**
            price_per_unit ← price_per_unit + C_i * P_i
        price_per_unit ← price_per_unit + fixed_price
        dynamic_price = (cons_last_hour / average_consumption) * price_per_unit
        print dynamic_price
    $j \leftarrow j + 1$
until program is stopped

Algorithm: Static Pricing for Electricity Consumption

Input: S_i for four paritions of consumption slab where $0 < i \leq 3$, P_j for prices of respective slabs where $0 < j \leq 4$ and last month's consumtion as consumption.
Output: Static Price as static_price for last month

if consumtion $\leq S_1$ **do**
    static_price = consumption * P_1
    return static_price
else if $S_1 <$ consumtion $\leq S_2$ **do**
    static_price = S_1 * P_1 + (consumption - S_1) * P_2
    return static_price
else if $S_2 <$ consumtion $\leq S_3$ **do**
    static_price = S_1 * P_1 + (S_2 - S_1) * P_2 + (consumption - S_2) * P_3
    return static_price
else if $S_3 <$ consumtion **do**
    static_price = S_1 * P_1 + (S_2 - S_1) * P_2 + (S_3 - S_2) * P_3 + (consumption - S_3) * P_4
    return static_price

Time and Space Analysis :-

Static Pricing

Time complexity = O(n)

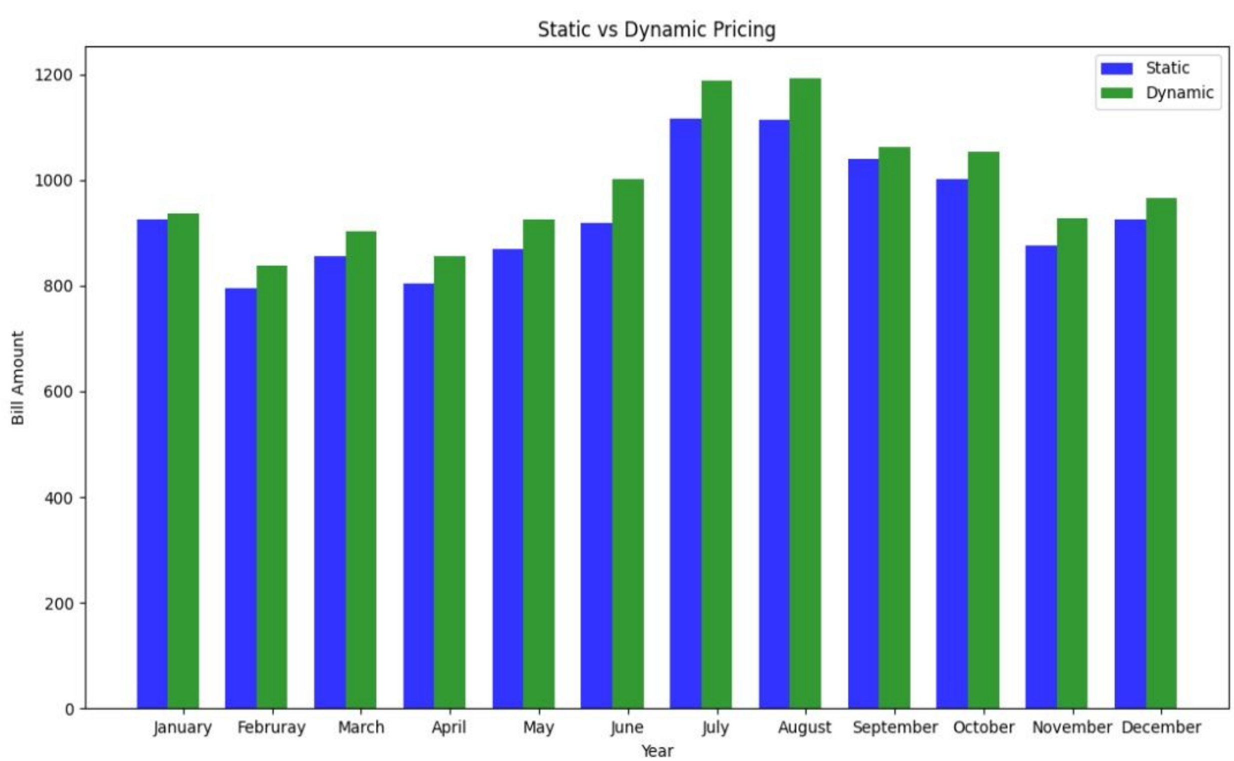
Space Complexity = O(n)

Dynamic Pricing

Time Complexity = O(n)

Space Complexity = O(n)

OUTPUT



The graph represents a comparison of monthly bills calculated by using the dynamic and static price of a particular house throughout the year .

For calculating dynamic price for an hour, firstly we calculate a base price which depends upon the price of fuel (coal, diesel, natural gas, etc) and their percentage contribution in total electricity production. Infrastructure costs are also taken into account while calculating base price. Previous hour demand and average consumption per hour calculated from the past 7 days are also required for calculating price per unit for a given hour .

Static price is set equal to existing rate of electricity in the state of residence; in our case, Uttar Pradesh.

Two - Way distribution of Electricity

PROBLEM

Under normal distribution method, there is one supplier and many consumers. That is, the relationship for supply to demand is one to many. This has several drawbacks as there is much load on one head as well as there is no promotion of renewable sources of energy on grass root level. In this modern era, the focus is on futuristic sources of energy, minimising wastage and pollution that occurs during production of electricity.

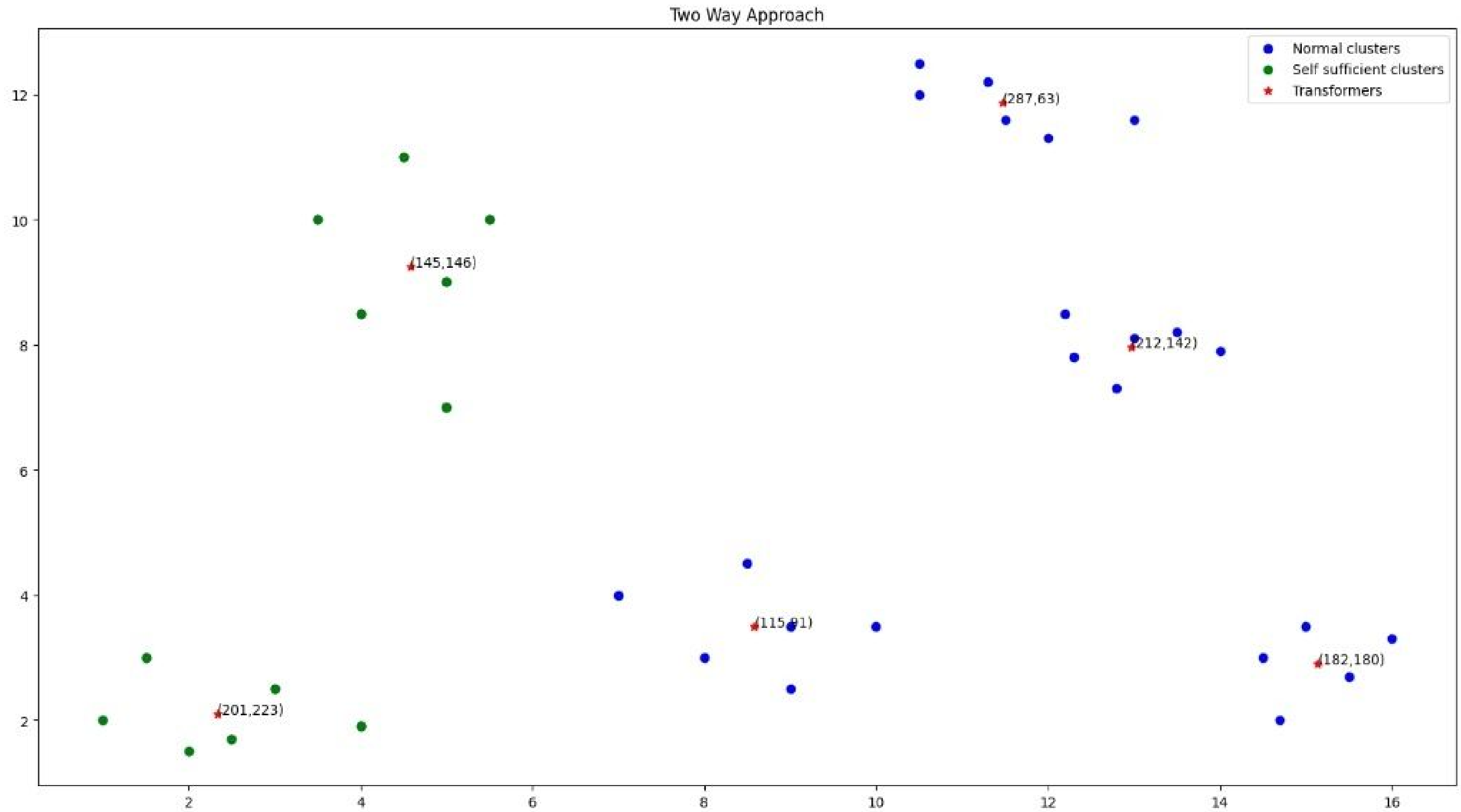
GOALS

- Develop a two-way electricity distribution system where the consumer itself can produce electricity.
- Implement futuristic methods of production of electricity.

SOLUTION

We design a two-way electricity distribution model where the consumer (by renewable methods like solar energy) can distribute in its cluster when the demand increases. This not only reduces the demand load on the supplier but is also beneficial for the environment. Also, this reduces the bill for the consumer as it can use the electricity developed in-house.

Output



Two Way Approach
Normal clusters
Self sufficient clusters
Transformers
(287,63)
(145,146)
(212,142)
(115,91)
(182,180)
(201,223)