GOGORO SITE SELECTION PROBLEM

TEAM O

B11611047 楊闳凱
B11705051 陳奕廷
B11702044 何娸瑋
B11702080 周榆庭

CONTENTS

INTRODUCTION

GoStation

- The largest battery swapping refueling system in Taiwan
- Easy access for users to swap depleted batteries for fully charged ones, ensuring continuous mobility without the need for lengthy charging times.

INTRODUCTION

site selection

high cost

efficiency

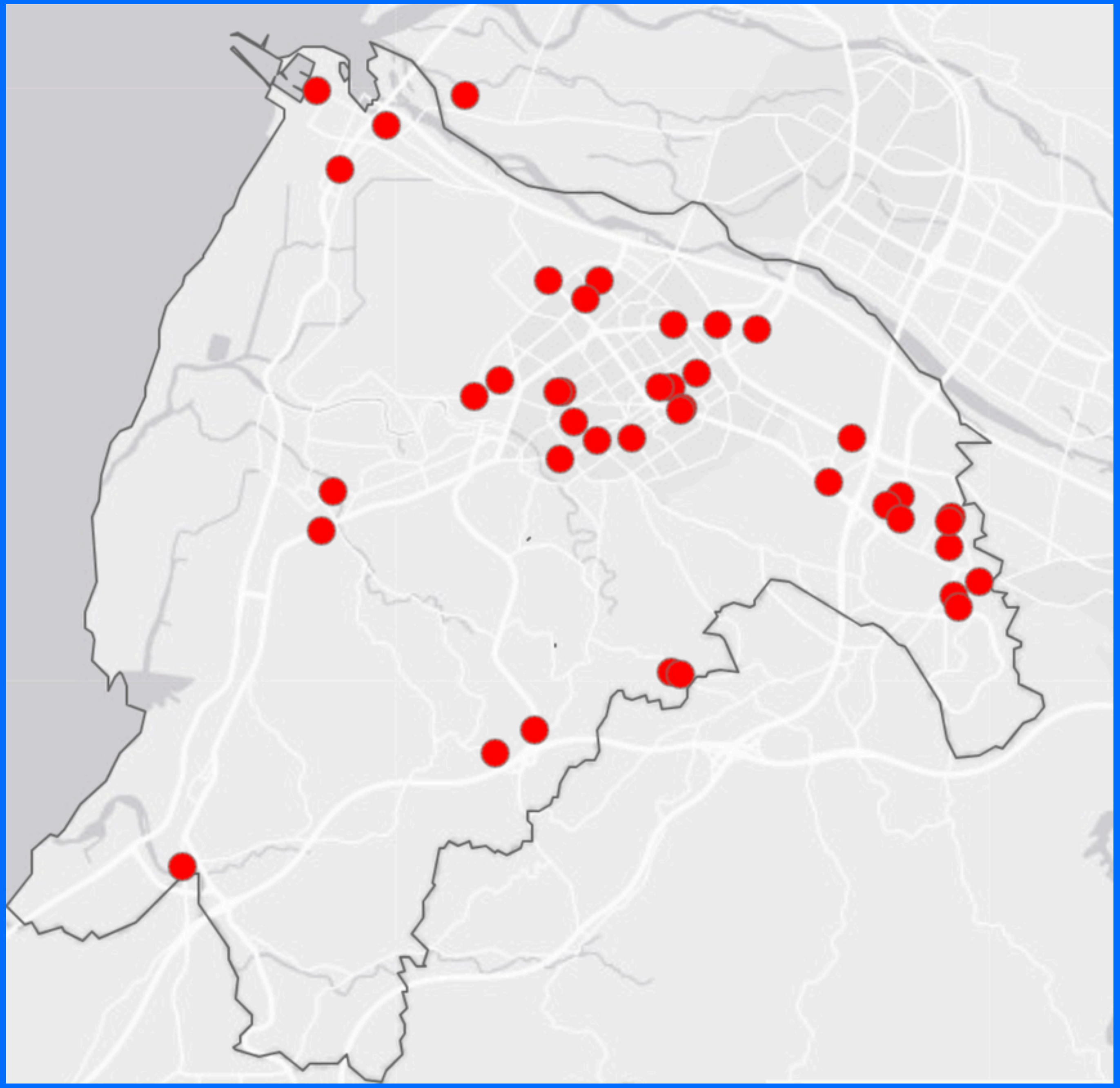
sustainability

convenience

02

PROBLEM DESCRIPTION

PHASE 1: GOSTAION OPTIMAZATION STRATEGY



- There are 42 GoStations in Hsinchu City. We are considering whether any of these GoStations can be removed due to the availability of stations with larger capacities.
- **Assumption:**
  - Every community's demand should be satisfied by the nearest GoStation.
  - If this condition is not met, the GoStation cannot be removed.
- **Goal:** Maximize the removal of GoStations without compromising service coverage.

PHASE 2: SHIPPING COST MINIMIZATION

How batteries would be dispatched to the remaining GoStations so that the battery shipping costs per day could be minimized while meeting each community's demand for fully charged batteries?

types	Description
shipping cost between GoStations	• Staff will transport the fully charged batteries from some GoStations to the others that are shortage in inventory • charge linearly according to the shipping distance • The maximum cost is \$1 per battery
extra battery cost to get batteries from warehouse	• get batteries from warehouse to cover shortage in supply • costs \$ 1,000 per battery
assign cost	• hire staff to schedule batteries lead to a fixed labor cost \$ 1 per period

03

DATA PROCESSING

DATA PREPARATION

We obtain raw material from the website called **Gogoro 神器** by the following steps:

DATA PREPARATION

We obtain raw material from the website called **Gogoro 神器** by the following steps:

Step 1

open the website as html file

Step 2

copy the graph illustrating the remaining battery numbers

Step 3

record the upper and lower bounds of time and battery of each graph, then run Python to get the real-time full battery data on all web pages.

Step 4

to reduce the number of removal Gostations in Phase 1, the supply should be underestimated and the demand should be overestimated as much as possible.

PRE-PROCESSING

Phase 1: supply and demand

Demand

- With one week-long data, calculate the supply of each data point and the maximum number of batteries at the gostation as the immediate demand
- took their daily peak time usage and added them together to divide the total number of periods adopted to obtain the average demand.

Supply

- beginning inventory: the first quantity of fully charged batteries in each period
- the additional batteries that can be provided during a period : the additional number of fully charged batteries within a period
- The sum of the two numbers is the total supply
- took the average supply during the peak hours as the data.

PRE-PROCESSING

Phase 2: supply and demand

3 batteries as the number that can be fully charged in half an hour

Sometimes newly charged battery will be consumed immediately. the average number of batteries that can be charged at a site in half an hour is about two to three, but it still varies by regions.

maximum number of fully charged batteries to 80% of the total

in many periods, the maximum number of fully charged batteries will not reach the maximum, which is designed to protect the batteries

the demand and supply of places where the number of fully charged batteries is the same and is less than 80% of the maximum batteries are added by three batteries

PRE-PROCESSING

Phase 2: supply and demand

Demand

- we assign the demand of each community to the remaining GoStations in the result of phase 1
- calculate the demand of the them at each period. Since the GoStations that has been removed will not be assigned demand, the demand will always be 0.

Supply

- The supply of this part will only count how many more batteries appear in this period than in the previous period
- Similarly, the supply data will only consider GoStations that will keep running after Phase 1.

04

MATHEMATICAL MODEL

PHASE 1: GOSTAION OPTIMIZATION STRATEGY

Parameters

There are n communities in Hsinchu City. Let $I = \{1, ..., n\}$ be the set of communities.

There are m GoStations in Hsinchu City. Let $J = \{1, ..., m\}$ be the set of GoStations.

Let D_i be the demand of batteries in community $i \in I$.

Let K_j be the capacity of GoStation $j \in J$.

Let d_{ij} be the distance between community $i \in I$ and GoStation $j \in J$.

Let $G = \sum_{i \in I} d_{ij} \quad \forall j \in J$.

Decision Variables

Let $x_j = 1$ if GoStation $j \in J$ will keep running, or 0 otherwise.

Let $y_{ij} = 1$ if the demand in community $i \in I$ is satisfied by facility $j \in J$, or 0 otherwise.

Let w_i be the distance between community $i \in I$ and its cloest GoStation.

PHASE 1: GOSTAION OPTIMIZATION STRATEGY

Formulation

$$\min \quad \sum_{j \in J} x_j$$

$$\text{s.t.} \quad \sum_{i \in I} D_i y_{ij} \le K_j x_j \quad \forall j \in J \qquad (1)$$

$$y_{ij} \le x_j \quad \forall j \in J \qquad (2)$$

(1) For each community, the demand must be less than the electricity capacity provided by each GoStation.

$$w_i \le d_{ij} x_j + G(1 - x_j) \quad \forall i \in I \quad \forall j \in J \qquad (5)$$

$$x_j \in \{0,1\} \quad \forall j \in J \qquad (6)$$

$$y_{ij} \in \{0,1\} \quad \forall i \in I \quad \forall j \in J \qquad (7)$$

PHASE 1: GOSTAION OPTIMIZATION STRATEGY

Formulation

$$
\begin{aligned}
\min \quad & \sum_{j \in J} x_j \\
\text{s.t.} \quad & \sum_{i \in I} D_i y_{ij} \le K_j x_j \quad \forall j \in J && (1) \\
& y_{ij} \le x_j \quad \forall j \in J && (2) \\
& \sum_{j \in J} y_{ij} = 1 \quad \forall i \in I \quad \forall j \in J && (3) \\
& w_i \le \sum_{j \in J} d_{ij} y_{ij} \quad \forall i \in I && (4) \\
& w_i \le d_{ij} x_j + G(1 - x_j) \quad \forall i \in I \quad \forall j \in J && (5) \\
& x_j \in \{0, 1\} \quad \forall j \in J && (6) \\
& y_{ij} \in \{0, 1\} \quad \forall i \in I \quad \forall j \in J && (7)
\end{aligned}
$$

(2) Examine the existience of GoStation and whether it can provide enough electricity

(3) Each community can only go to one Gostation for swapping batteries

PHASE 1: GOSTAION OPTIMIZATION STRATEGY

Formulation

$$\min \quad \sum_{j \in J} x_j$$

$$\text{s.t.} \quad \sum_{i \in I} D_i y_{ij} \leq K_j x_j \quad \forall j \in J \tag{1}$$

$$y_{ij} \leq x_j \quad \forall j \in J \tag{2}$$

$$\sum_{j \in J} y_{ij} = 1 \quad \forall i \in I \quad \forall j \in J \tag{3}$$

$$w_i \leq \sum_{j \in J} d_{ij} y_{ij} \quad \forall i \in I \tag{4}$$

$$w_i \leq d_{ij} x_j + G(1 - x_j) \quad \forall i \in I \quad \forall j \in J \tag{5}$$

$$x_j \in \{0,1\} \quad \forall j \in J \tag{6}$$

$$y_{ij} \in \{0,1\} \quad \forall i \in I \quad \forall j \in J \tag{7}$$

(4) (5)
The minimum distance between community i and charging station j is constrained to ensure optimal station allocation.

PHASE 2: MINIMIZING BATTERY SHIPPING COST

Parameters

Let $M = \{1, ..., 27\}$ be the set of GoStations and $T = \{1, ..., 48\}$ be the set of periods.
Let S_{mt} be the additional supply at GoStation $m \in M$ in the beginning of period $t \in T$,
D_{mt} be the demand of GoStation $m \in M$ at period $t \in T$,
$C_{mm'}$ be the cost shipping a battery from GoStation $m \in M$ to GoStation $m' \in M$.

Decision Variables

Let $z_{m,t,m',t+1}$ be the number of batteries transferred from GoStaiton $m \in M$ in the beginning of period $t \in T$ to GoStation $m' \in M$ in the beginning of period $t + 1 \in T$,
$f_{m,m',t,t+1} = 1$ if in period $t \in T$ staffs transfer batteries from GoStation $m \in M$ to GoStation $m' \in M$, or 0 otherwise,
E_m be the number of batteries added to GoStation $m \in M$ before period 1.

PHASE 2: MINIMIZING BATTERY SHIPPING COST

$$\min \sum_{m=1}^{42} \sum_{m'=1/\{m\}}^{42} \sum_{t=1}^{47} C_{mm'} z_{m,t,m',t+1} + \sum_{m=1}^{42} \sum_{m'=1/\{m\}}^{42} C_{mm'} z_{m,48,m',1} + \tag{1}$$

$$\sum_{m=1}^{42} \sum_{m'=1/\{m\}}^{42} \sum_{t=1}^{47} f_{m,t,m',t+1} + \sum_{m=1}^{42} \sum_{m'=1/\{m\}}^{42} f_{m,48,m',1} + 1000 \sum_{m=1}^{42} E_m \tag{2}$$

$$\text{s.t.} \quad \sum_{m'=1}^{42} z_{m',t,m,t+1} + S_{m,t+1} \geq D_{m,t+1} \quad \forall m \in M, \quad \forall t = 1, ..., 47 \tag{8}$$

$$\sum_{m'=1}^{42} z_{m',t,m,t+1} + S_{m,t+1} - D_{m,t+1} = \sum_{m'=1}^{42} z_{m,t+1,m',t+2} \quad \forall m \in M, \quad \forall t = 1, ..., 46 \tag{9}$$

PHASE 2: MINIMIZING BATTERY SHIPPING COST

shipping cost between GoStations

$$\min \sum_{m=1}^{42} \sum_{m'=1/\{m\}}^{42} \sum_{t=1}^{47} C_{mm'} z_{m,t,m',t+1} + \sum_{m=1}^{42} \sum_{m'=1/\{m\}}^{42} C_{mm'} z_{m,48,m',1} +$$

$$\sum_{m=1}^{42} \sum_{m'=1/\{m\}}^{42} \sum_{t=1}^{47} f_{m,t,m',t+1} + \sum_{m=1}^{42} \sum_{m'=1/\{m\}}^{42} f_{m,48,m',1} + 1000 \sum_{m=1}^{42} E_m$$

s.t. 42

assign cost

$S_{m,t+1} \geq D_{m,t+1} \quad \forall m \in M, \quad \forall t = 1,$

extra battery cost to get batteries from warehouse

42 42

add up the costs related to battery shippment

$z_{m,t+1,m',t+2} \quad \forall m \in M, \quad \forall t = 1, ..., 46 \qquad (9)$

PHASE 2: MINIMIZING BATTERY SHIPPING COST

(8) make sure we can satisfy all of the demand

beginning inventory

demand

$$\text{s.t.} \quad \sum_{m'=1}^{42} z_{m',t,m,t+1} + S_{m,t+1} \geq D_{m,t+1} \quad \forall m \in M, \quad \forall t = 1, ..., 47 \qquad (8)$$

the number of batteries become fully charged

PHASE 2: MINIMIZING BATTERY SHIPPING COST

$$\min \sum_{m=1}^{42} \sum_{m'=1/\{m\}}^{42} \sum_{t=1}^{47} C_{mm'} z_{m,t,m',t+1} + \sum^{42} \sum^{42} C_{mm'} z_{m,48,m',1} +$$

$$\sum_{m=1}^{42} \sum_{m'=1/\{m\}}^{42} \sum_{t=1}^{47} f_{m,t,m',t+1} +$$

(9) The ending inventory of one period should be the beginning inventory of the next period.

$$\text{s.t.} \sum_{m'=1}^{42} z_{m',t,m,t+1} + S_{m,t+1} \geq D_{m,t+1} \quad \forall m \in M, \quad \forall t = 1, ..., 47 \qquad (8)$$

$$\underbrace{\sum_{m'=1}^{42} z_{m',t,m,t+1} + S_{m,t+1} - D_{m,t+1}}_{\text{ending inventory of period t}} = \underbrace{\sum_{m'=1}^{42} z_{m,t+1,m',t+2}}_{\text{beginning inventory of period t + 1}} \quad \forall m \in M, \quad \forall t = 1, ..., 46 \qquad (9)$$

PHASE 2: MINIMIZING BATTERY SHIPPING COST

$$\sum_{m'=1}^{42} z_{m',48,m,1} + S_{m,1} + E_m \geq D_{m,1} \quad \forall m \in M \tag{10}$$

$$\sum_{m'=1}^{42} z_{m',48,m,1} + S_{m,1} + E_m - D_{m,1} = \sum_{m'=1}^{42} z_{m,1,m',2} \quad \forall m \in M \tag{11}$$

$$\sum_{m'=1}^{42} z_{m',47,m,48} + S_{m,48} - D_{m,48} = \sum_{m'=1}^{42} z_{m,48,m',1} \quad \forall m \in M \tag{12}$$

$$z_{m',t,m,t+1} \leq 549 f_{m,m',t,t+1} \quad \forall m \in M, \quad \forall m' \in M, \quad \forall t \in T \tag{13}$$

$$z_{m,t,m',t+1} \in \mathbb{N} \quad \forall m \in M, \quad \forall m' \in M, \quad \forall t \in T \tag{14}$$

$$f_{m,m',t,t+1} \in \{0, 1\} \quad \forall m \in M, \quad \forall m' \in M, \quad \forall t \in T. \tag{15}$$

PHASE 2: MINIMIZING BATTERY SHIPPING COST

$$\sum_{m'=1}^{42} z_{m',48,m,1} + S_{m,1} + \underline{E_m \geq D_{m,1}} \quad \forall m \in M \tag{10}$$

$$\sum_{m'=1}^{42} z_{m',48,m,1} + S_{m,1} + \underline{E_m - D_{m,1}} = \sum_{m'=1}^{42} z_{m,1,m',2} \quad \forall m \in M \tag{11}$$

$$\sum_{m'=1}^{42} z_{m',47,m} \tag{12}$$

extra battery amount to get enough batteries

(10) (11) Before the beginning of the first period, if there is no excessive supply in all of the GoStations, staff have to get batteries from warehouse.

PHASE 2: MINIMIZING BATTERY SHIPPING COST

(13) identify whether labor cost incurred in period t

$$m'=1 \qquad (10)$$

$$\sum_{m'=1}^{42} z_{m',48,m,1} + S_{m,1} + E_m - D_{m,1} = \sum_{m'=1}^{42} z_{m,1,m',2} \quad \forall m \in M \qquad (11)$$

$$\sum_{m'=1}^{42} z_{m',47,m,48} + \ldots \sum_{m'=1} \qquad (12)$$

1 if assign cost incurred in period t, or 0 otherwise

$$z_{m',t,m,t+1} \leq 549 f_{m,m',t,t+1} \quad \forall m \in M, \quad \forall m' \in M, \quad \forall t \in T \qquad (13)$$

$$z_{m,t,m',t+1} \in \mathbb{N} \quad \forall m \in M, \quad \forall m' \in M, \quad \forall t \in T \qquad (14)$$

$$f_m \qquad (15)$$

the number of batteries transffered at period t

05

HEURISTIC ALGORITHM

PHASE 1: GOSTAION OPTIMIZATION STRATEGY

```
function plan(abandoned, station):
    assumed_abandoned = abandoned + [station_name]

    if station not in assigned_station:
        RemoveStation(station)
        abandoned = assumed_abandoned
        return True

    floating_communities = CommunitiesUsesStation(station)
    each_community_demand = demand[station] / (size of floating_communiti

    for each community in floating_communities:
    #test if the demand of the community can be accepted by
    #the nearest station left. if not return false

    for station in stations:
        UpdateNewDemand()

    RemoveStation(station)
    abandoned = assumed_abandoned
    return True
```

Pseudo Code

PHASE 1: GOSTAION OPTIMIZATION STRATEGY

Calculate Load Ratios	Compute the ratio of the average full battery capacity to the maximum battery capacity for each station. Sort these ratios from smallest to largest
Function to Determine Removal	Assuming the station will be removed. Create an array, assumed_abandoned, containing both already removed stations and stations to be removed.
Check Community Demand	If all second closest stations can meet their respective community's demand, then remove the station. Otherwise, retain the station in operation.

PHASE 2: MINIMIZING BATTERY SHIPPING COST

```
while (not elements in extra are not all zero):
    reset extra for all stations to 0
    set left_battery of stations to their need of extra battery

    total_send_cost = 0

    for period in periods:
        total_battery = 0
        for each station in stations:
            station.left_battery =station.left_battery + supply at current period - demand at current period


        for each station in stations:
            if station.left_battery < 0:
                # check from the closest to the farthest station if any one can meet this
                # station’s need. If there isn’t any station that can meet the need, then
                # get the batteries from others in the distance order. Store the number of
                # taken stations to workers, the total distance of taken stations to send_distance,
                # and the amount of not available from other stations to  extra_needed.
                if extra_needed > extra[station]
                   extra[station] = extra_needed
                total_send_cost += workers + (send_distance / farthest_two_stations)


    if elements in extra is not all zero:
        for each station in stations:
            if extra[station] is in max of extra:
                station.etra += max(extra)
```

Pseudo Code

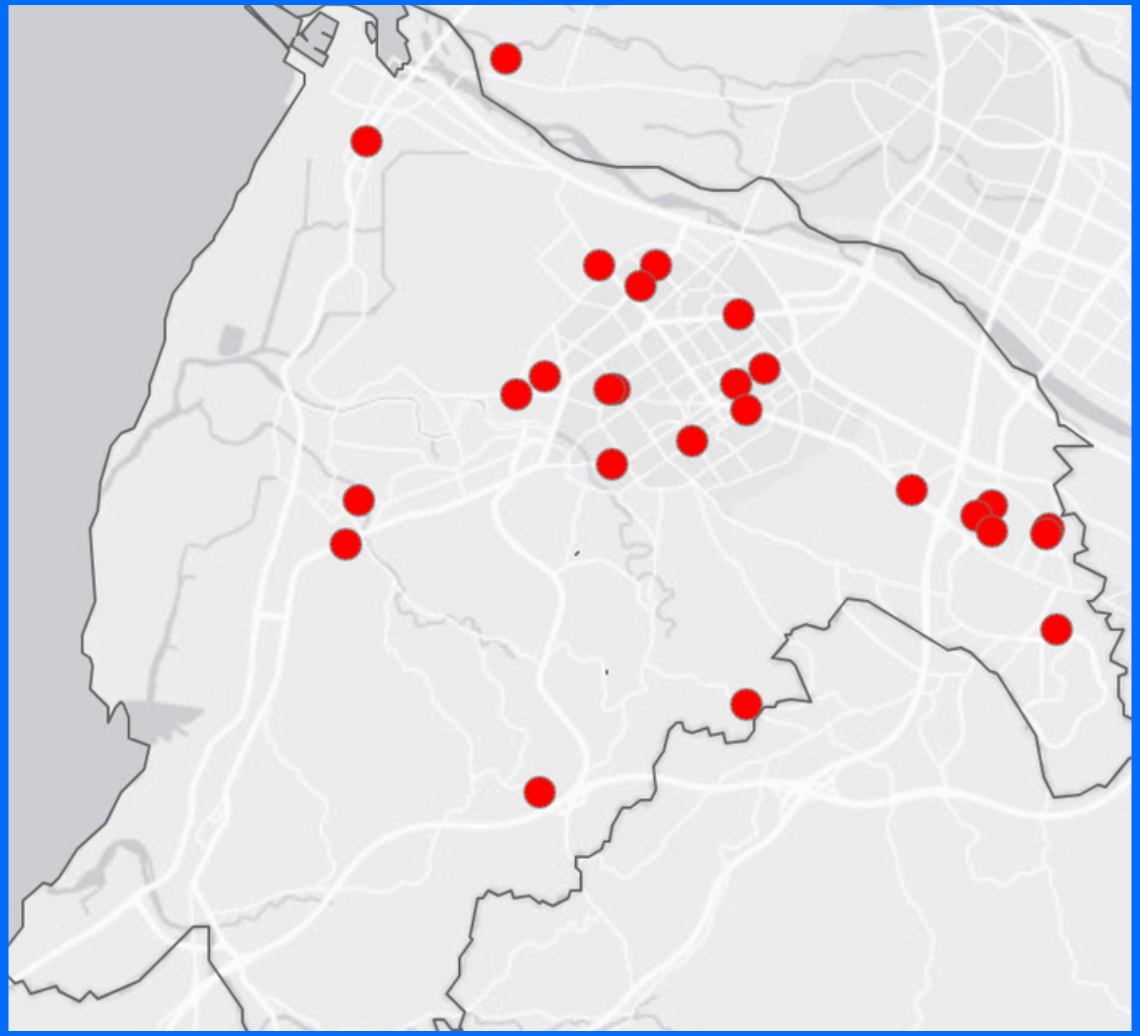
PHASE 2: MINIMIZING BATTERY SHIPPING COST

Demand Check	**Calculate the initial inventory and the inventory. If the inventory is negative, the replenishment process will begin.**
One-to-One or Many-to-One Replenishment	**Check the nearest GoStation for full demand. If unmet, proceed to the next. Use many-to-one replenishment if needed, iteratively reducing demand from multiple stations until fully satisfied.**
Extra Battery Calculation	**In special cases where GoStation batteries are insufficient to meet demand, we'll utilize extra batteries. The algorithm will then update replenishment records accordingly.**

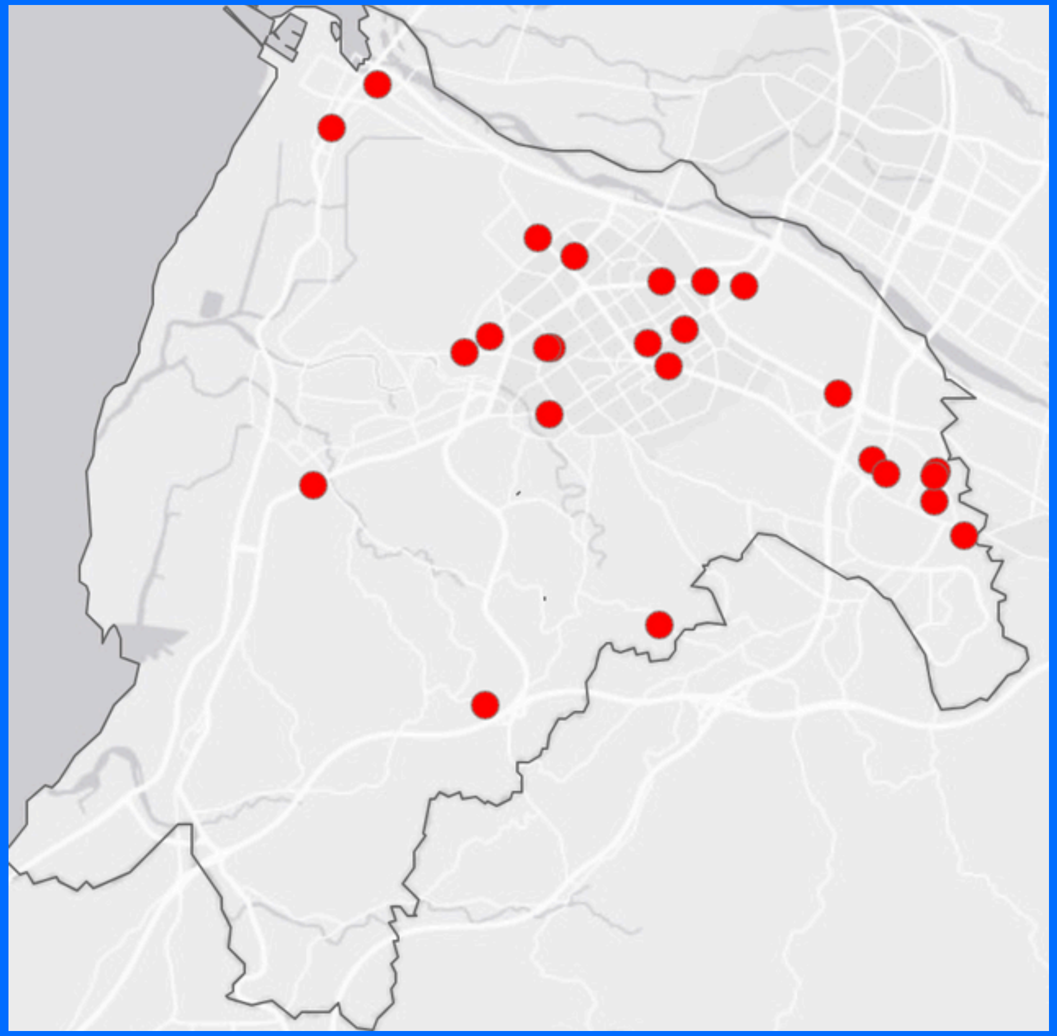
06

EVALUATION

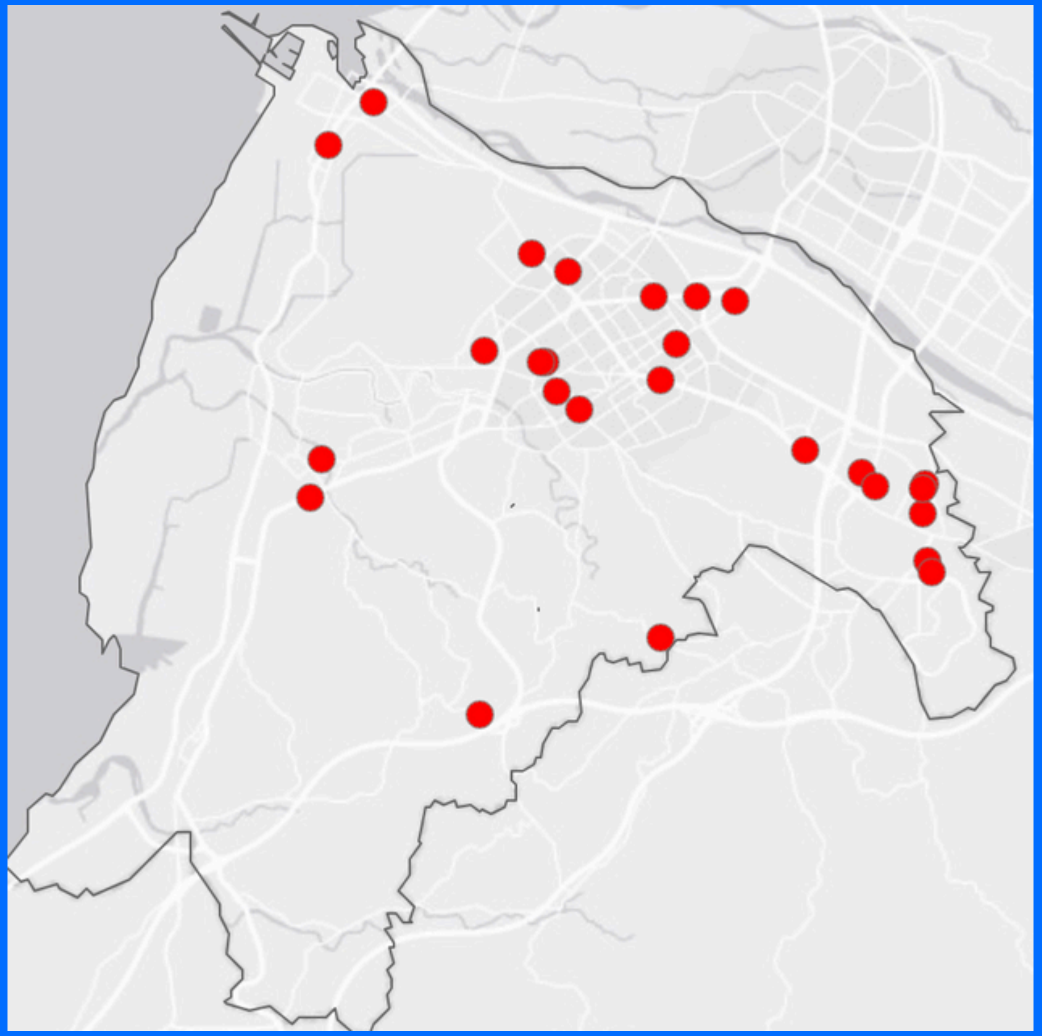
PHASE 1: GOSTATION OPTIMIZATION STRATEGY



Gurobi
Left: 27 stations



Heuristic
Left: 27 stations



Simple
Left: 28 stations

PHASE 1: GOSTATION OPTIMIZATION STRATEGY

Heuristic VS Simple

The only difference is that we haven't calculated the ratio of the average full battery capacity to the maximum battery capacity, and the algorithm will not follow any order.

PHASE 2: SHIPPING COST MINIMIZATION

optimized solution of gurobi

	A	B
1	z_{m, t , m', t+1}	value
2	z4,28,8,29	3
3	z7,26,0,27	15
4	z11,22,13,23	2
5	z12,14,13,15	6
6	z33,45,32,46	10

	A	B
1	E_{m}	value
2	E0	3
3	E1	0
4	E2	17
5	E3	16
6	E4	0

	A	B
1	f_{m, m', t, t+1}	value
2	f4,8,28,29	1
3	f7,0,26,27	1
4	f11,13,22,23	1
5	f12,13,14,15	1
6	f33,32,45,46	1

Shipping Times = 49

Extra Battery Cost = 620,000

Assign Times = 35

PHASE 2: SHIPPING COST MINIMIZATION

optimality gap

	shipping times	assign times
heuristic algorithm	78.7%	371.4%
simple algorithm	239.6%	894.3%

PHASE 2: SHIPPING COST MINIMIZATION

Heuristic VS Simple Heuristic

For a simple heuristic, extra batteries are intensively located in particular GoStations. This way, although the total amount of extra batteries of simple heuristic and heuristic algorithm is identical, it takes simple heuristic algorithm more shipping times and assign times to operate.

07 CONCLUSIONS

IMPROVEMENT

DATA ACQUISITION CHALLENGES

Due to factors like time and existing circuits, this makes supply prediction challenging. When a GoStation runs out of batteries, accurately determining remaining user charging needs becomes difficult.

DIFFICULTY IN ESTIMATING USER EXPERIENCE

Removing stations could force users to travel further to find a battery swap station, which could negatively impact user experience and, consequently, Gogoro's profitability.

OUTREACH RESERACH COVERAGE

Implement our solution in a wider areas, revising our algorithm to fit in various scenarios

THANK YOU