撈下來的資料格式：

|  | **sno** | **sna** | **tot** | **sbi** | **sarea** | **mday** | **lat** | **lng** | **ar** | **sareaen** | **snaen** | **aren** | **bemp** | **act** | **srcUpdateTime** | **updateTime** | **infoTime** | **infoDate** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 500101001 | YouBike2.0\_捷運科技大樓站 | 28 | 0 | 大安區 | 2022-05-16 14:38:04 | 25.02605 | 121.5436 | 復興南路二段235號前 | Daan Dist. | YouBike2.0\_MRT Technology Bldg. Sta. | No.235，Sec. 2，Fuxing S. Rd. | 28 | 1 | 2022-05-16 14:47:11 | 2022-05-16 14:47:24 | 2022-05-16 14:38:04 | 2022-05-16 |
| **1** | 500101002 | YouBike2.0\_復興南路二段273號前 | 21 | 1 | 大安區 | 2022-05-16 14:38:04 | 25.02565 | 121.54357 | 復興南路二段273號西側 | Daan Dist. | YouBike2.0\_No.273，Sec. 2， Fuxing S. Rd. | No.273，Sec. 2，Fuxing S. Rd. (West) | 20 | 1 | 2022-05-16 14:47:11 | 2022-05-16 14:47:24 | 2022-05-16 14:38:04 | 2022-05-16 |

盤點可用資源：

| **站名** |  |  | **？** |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **sna** | **tot** | **sbi** | **mday** | **bemp** | **act** | **srcUpdateTime** | **updateTime** | **infoTime** | **infoDate** |
| YouBike2.0\_捷運科技大樓站 | 28 | 0 | 2022-05-16 14:38:04 | 28 | 1 | 2022-05-16 14:47:11 | 2022-05-16 14:47:24 | 2022-05-16 14:38:04 | 2022-05-16 |
| YouBike2.0\_復興南路二段273號前 | 21 | 1 | 2022-05-16 14:38:04 | 20 | 1 | 2022-05-16 14:47:11 | 2022-05-16 14:47:24 | 2022-05-16 14:38:04 | 2022-05-16 |

sno(站點代號)、sna(中文場站名稱)、tot(場站總停車格)、sbi(可借車位數)、sarea(中文場站區域)、mday(資料更新時間)、lat(緯度)、lng(經度)、ar(中文地址)、sareaen(英文場站區域)、snaen(英文場站名稱)、aren(英文地址)、bemp(可還空位數)、act(場站是否暫停營運)

感興趣的問題：最佳化的車輛分配

空車率：1 - sbi/tot

滿車率：sbi/tot

假設一天有24個小時的空車率，想最大化（空車率最小的時段。），同時避免沒位可停。

目標：

* 最小化空車率
* 最小化人力資源（運送次數、時間、距離）
* 最大化服務範圍（設立新站點）：以經緯度、人口出發、set cover problem

限制：

* 滿足需求
* 有限資源：人力、每單位時間能調度的車輛數

困難：

* 怎麼估計該站點的需求？

每小時的空車數(tot - sbi)

* 怎麼定義最小化目標？

最小化滿車/空車的情形(i.e. 該小時的空車率小於10%，即定義為1，反之則0。)

* 車輛供給

估計目前的總車輛供給數目、並假設車輛總站的位置、以及每小時能運送的車輛數K（代替人力資源）。

一小時為單位

交通狀況、雨量、平日假日

假設：

空車率變動、台北市

模型：

數據複雜、不好量化：

抽象化：

最小

目前舊款Youbike 1.0還有1.1萬台車，新款Youbike 2.0則有1萬500台車，合計2萬1500台，平均一天總租借次數約10萬次。

5/25 討論：

站點跟站點間的距離

給一圖 G(V,E)

V 為站點的集合

E 為站點間（量化）的距離

Formulation:  
（給N個運送員建議的搬移路線，最小化運送員移動路徑（用時間跟距離當常數評估））

最短路徑問題、運送問題、空車率問題 (隱藏問題:也要有地方停車7:3?)

空車率 : 無車車位/全部車位

車輛變化率

空車率：預估在時刻中的空車率、決定最佳有車比率是多少、各站剩餘車輛與彼此距離、

(假設沒有暫時存放區、主要目標為調度現有車輛)

人力資源 : 各站分配(資料導向，各)

補車順序 : (定義空車比率，得出移動車輛數和目標移置地點，一次裝車上限)

min 運送距離總和（每次移動的時候）: 定義駐點

**Operations Research - Final proposal**

**youbike 補車動線規劃**

2022.05.25

周成康 B09705011 黃戎僔 B09705032 陳瑾叡 B09705016 吳驊祐 B09705009   
吳天冷 B08303135 李詠如 B07703084 劉德駿 B07801013  
李宗霖 B07702044 陳琳瑄 B06B02059

**一、情境敘述**

是不是常常有走在台大的路上，遇到想騎 youbike 附近卻找不到車的狀況呢？

但又時常看到youbike的調度車出現，員工辛苦地將腳踏車從停放數太多的地方，移動至較熱門因此空車率較高的站點。然而就算調度車已出勤許多次，使用者對於空車率的感受卻仍普遍有感，因此我們想替youbike 公司規劃調度動線與補車順序的規劃。

假設youbike的公司決定租用小卡車作為調度車，並重新規劃台大校園內的補車動線。為了最小化空車率[[1]](#footnote-0)與最小化公司成本，我們希望在平衡各站點車輛空車率的前提下，規劃不同調度車在單日中最佳化的調度路線。（站點區域可能會擴大到全台北市，實務組還在討論中）並將一天以一小時為單位，分為24個區間，每一個區間的時間單位換算後，即為調度車一次出勤可達之最遠距離。

由於主要目標是在平衡各點空車率之外，最小化運送成本，因此先將問題簡化成單一時間區間內的情形。再進一步討論連續時段時，每個時段需有多少支援的調度車輛以及調度路線。

**二、模型**

目標式：min )

懲罰：

限制式：

1. 每趟總和時間不可超過指定上限
2. 每台車的載運量有限（）
3. 各站點最後要滿足空車率小於（）成
4. 各站點的移入總量要 該站點需求
5. 各站點的移出總量要 該站點供給

**Set variable:**

1. 站點個數
2. 時段個數

**Desicion variables:**

1. 在時段時，站點間的運輸流量
2. 調度車數量

**Parameter**

1. 站點之間的距離
2. 換算後站點之間的所需時間, （ 為某時段的係數）
3. 在時段時，站點之間的移動成本（里程數、油錢）。
4. 站點總停車格數量
5. 在時段時，站點目前有車的數量
6. 每台調度車的可搬運容量
7. 租用一台調度車一天的成本
8. 調度車在單位時間內移動的距離（or 時間）上限
9. 假設運輸車上班時，只能由特定入口進入學校（如大門、新月台、舟山路、長興街口等），那麼運輸車進入校園的依據，即為離哪個入口的需求最高、且不可都擠在同個入口（？）。
10. 每一台運輸車都會有容量上限（K），因此在每個小時內，運輸車會有一連串的++或--的動作。例如：出發點大門（0）- (c:10) > 管一（+3）- (c:20) > 總圖（+30）
11. 先假設運輸車的單位出勤時間為1小時（變數），即在一小時內須將各站點的車子，從較多車輛的站點運送到較缺乏車量的站點。
12. 除了一天中第一個時段為指定調度車出發點外，在其他時段中，第 個時段第 台車的移動終點，即為第 個時段第 台車的移動起點。
13. 同時假設租車一次以一天為單位，而當一天中不同時段有不同的調度車需求時，需滿足最高數量。

因此我們將

1. 先 決定單日所需的最高調度車數量
2. 再 以該調度車數量作為整日擁有資源，進行最小化每台車時間成本的路線規劃。

二、模型

**Set variable:**

* 站點個數
* 時段個數

**外生變數：**

* 站點之間的距離
* 換算後站點之間的所需時間, （ 為某時段的係數）
* 在時段時，站點之間的移動成本（里程數、油錢）。
* 站點總停車格數量
* 在時段時，站點目前有車的數量
* 每台調度車的可搬運容量
* 租用一台調度車一天的成本
* 調度車在單位時間內移動的距離（or 時間）上限

間接計算出的變數：

* 卡車數量
* 每個站點要補多少台車

**Objective function:**

第一階段：求得最高需求時段需要的卡車數

* 計算24小時內數據中，各站點空車需求數最高的小時，並以該小時內的總需求除以每台調度車可載運的總量作為調度車數量的依據，即可求得 調度車數量
* 總需求（該小時） 每台調度車可載運的總量 = 調度車數量

第二階段：

* 以第一階段求得的卡車數作為上限 ，決定運送的路線與運送的量。
* 預計: minimum cost network flow (MCNF) 結合 assigment jobs

Decision Variable:

* 比較合理的假設
  + 綜合車子數量 與 運送成本
  + min 車輛總數\*車輛成本 + 各車的運輸成本 - 不做事懲罰？
  + 有張圖G（V, E）、有站點（V）、有移動成本 C(i, j)、但沒有流量限制（限制是在車子上不是圖上），因此應該讓他加總看看順序問題？？對一條路線而言，有個比率紀錄：（需求遞補/成本），如何從最有效益的角度去看？
* 這是沒那麼合理的假設()
  + 將所有站點分成 供給方 ( ) 與 需求方 ( )，將 台卡車作為中間倉儲，其中 個倉儲位於不同站點上。
  + 建立與這些站點與配車、配車與需求點間的距離矩陣、時間成本矩陣。  
    (在此種假設下，配車方式會隨調度車所在站點改變)
* 第台卡車的配給路線
* Let = flow size of arc (i, j)
* For each node i, the label (bi) means its supply quantity is bi.
* One supply node, two demand nodes, and two transshipment nodes.

為給定之Flow chat

假設有個站點

Assumption

補車搬上搬下的時間可忽略？

統一卡初始車出發點？

在每個時段時，卡車會選擇距離自己最近的站點出發，遇到等距的站點時，會選擇需求較缺的。（每個時段的移動距離、卡車裝載車數有限）

實際規劃路徑、assigment 問題

去排路線

設定開始時間為5:00

4臺：台大、淡水、新店、

定義成本

Objective value:

最小化公司成本 （假設租一台車一小時1000元，要最小化成本）

最小化空車率

最小化公司成本（量化：定義距離與人力、youbike 車 移動車的容量、空車率變化、平衡空車率、排路線、超過叫第二台車）、最小化空車率

動線、運送車的單位（運送單元、運送成本、單趟成本、路線成本）

一整天：叫車的成本、跑距離的成本

一小時看一次：該路線應該如何出發

綜合七天前資料的空車率拿來推估，得出類似班表。

==========

0525

分工：

實務組：(找資料、整理資料) 4

找資料(爬蟲+手動計算)

定義量化資料（定義量化方式、整理格式)

（內部再自行分工PPT與文件）

理論組：5

描述問題

建模型(數學)

寫程式

分析演算法性能（如CA2）

（內部再自行分工PPT與文件）

=============

實務組: 周成康、李詠如、吳驊祐、陳謹叡

**1 Introduction.**

You should always start by describing the background and motivation of your problem.

Then the problem should be defined. If possible, you should explain why the problem is interesting, i.e., challenging and important. Real-world observations are good for motivating your study. The problem should have a decision maker dealing with a complicated environment with various types of operations. You should highlight the trade-offs faced by the decision maker when choosing among alternatives.

### **3 Data collection.**

Describe the background of your case and the process of collecting information to generate the parameter values in your instance(s).

### **6 Conclusions.**

Say something to summarize the whole study and possible ways to improve it

============

理論組：李宗霖、吳天冷、黃戎僔、陳琳瑄、劉德駿

### **2 Problem description.**

Use words to describe your conceptual model and use formula to describe your mathematical model. Make your problem description precise.

### **4 Method.**

Tell us how you solve your problem, say, using Python to invoke Gurobi

Optimizer, implementing your own (heuristic) algorithm(s), etc

### **5 Results.**

Write down the results of solving your instance(s). If you have an executable plan, write it down. Some visualization is typically a plus. It will be better if you have some performance evaluation (say, your proposed solution may save how much time/money compared with the current solution).

最小化公司成本 & 最小化空車率（0-1）

空車率：

時段、站點、

[+1, -3, +4, -2]

平衡空車率？？計算？？

（量化：定義距離與人力、youbike 車 移動車的容量、空車率變化、平衡空車率、排路線、超過叫第二台車）、我們報問題敘述、建模、

**測試資料欄位**

| **資料欄位名稱** | **測試資料** |
| --- | --- |
| 各站點停車數量 | [1, 4, 10, …] |
| 各站點車柱總數 | [10, 20, 40, …] |
| 各站點需求  (+為過剩，-為需求) | [+10, −20, −40, …] |
| 從站點 i 調度車輛到站點 j 的成本 | [ [0 1000 2000 … ]  [1000 0 1000 … ]  …  [ 5000 4000 2200 ..]] |
| 從站點 i 與站點 j 的運輸時間 (單位：小時) | [ [0 0.2 0.6 … ]  [0.1 0 0.3… ]  …  [ 1.2 1.1 0.8 ...]] |

**5/30 提問：**

1. 我們想像使用者在輸入資訊後，公司可以在該時段內跑出在給定車輛數的情況下，這些車子的運送路線以及運送的搬上搬下量。
2. 兩種假設：
   1. 以站點間的需求移動為主要描述：  
      在假設各站點各有需求，變數表示從 i 點運到 j 點被移動的腳踏車量。另外有個變數表示將指派給第m台車移動。  
      Q1：一個站的不能被分給兩輛車的運送不太合理  
      Q2：在一個站點補不完其實可以留在調度車上不用放下去，但目前沒有考慮到「在調度車上」的量。  
      Q3：有嘗試想加入 m 個虛擬點，但無法評估各虛擬點到各站點的移動成本（因為車子虛擬點每個時間的位置不同）

Q4：讓所有站點可停車數都膨脹20臺（加上一台卡車可載運的容量上限）

* 1. 把每台車的放入與拿走分開：  
     每個站點 i 有某車 m 的放入量與拿走量  
     Q：不確定是否會出現同個站點拿下又拿上的問題，直觀上認為只需要其中一個，拿下又拿上其實等於不用拿。  
     Q：補滿後又被拿走的可能？（站點出現重複路線）

1. 我們希望在移動過後，得到的結果是可以從 串出第 m 台車的移動路線與班上搬下的數量。
   1. 移動路線希望可以從 串出來，在假設 2a 的情況下成立的話，就會是由的車子協助調度。
   2. 但不知道這些的結果會不會連起來、會不會出現瞬間跳躍、以及不知道移動順序與終點在哪邊。

5/30 老師給的回饋：

1. 參考VRP的列式
2. 參考TSP的constraints
3. 對某台車，限制沒經過的點拿上拿下的數量為0
4. 每個站點上的 flow in = flow out
5. 每台車的 flow in 、 flow out 小於容量上限（）
6. 限制車子開始與結束，保證會串成一串。
7. 分段做
   1. Phase-I：校內先分區
   2. Phase-II：排單線

**自行調整參數**

1. ：一台調度車容量上限。
2. ：單位時間內，租用一台調度車的成本。
3. ：安全範圍的停車率上限 。（比率）
4. ：安全範圍的停車率下限 。（比率）
5. ： 調度車在單位時間內移動的距離（or 時間） 上限。
6. : 懲罰係數（各站沒補足下限或補超過上限）
7. ：卡車虛擬點：*表示第 m台車在該時段最後的車上腳踏車數，（Final點的概念）虛擬點？？*

**Set variables:**

* 表示有 i 個站點。
* 表示有 m 台調度車。
* 表示有 t 個時段。

表示由 m 這台車，從 i 移動到 j 的量

= 1 表示由 m 這台車，從 i 移動到 j ， otherwise = 0。 （i != j）

= 各站沒補足下限的比率

= 補超過上限的比率

= 1 表示第 m 台車出動， otherwise = 0。（一天，目前的假設）

= 1 表示第 m 台車在第 t 個時段出動， otherwise = 0。（一小時）

不知道結果會不會連起來、會不會瞬間跳躍、以及不知道終點在哪

虛擬點還沒放完（晚點處理）

**目標式**

最小化 各車加總的（租用＋使用費）＋移動成本＋（沒補足或補超過）的懲罰

s.t.

1. 限制一個站點只能有一台車經過（進入）
2. 限制一個站點只能有一台車經過（移出）
3. 停車率（）要大於下界

, max { ,0}

1. 停車率（）要小於上界

, max { ,0}

1. 每趟總和時間（or距離）不可超過指定上限 L
2. 每台車的載運量有限（ A ）??
3. 各站點的 移入總量 要 0
4. 各站點的 移入總量+已有車數要 該站點 停放空間上限
5. 各站點的 移出總量 要 該站點停車數量

**New**

**自行調整參數**

* ：一台調度車容量上限。
* ：單位時間內，租用一台調度車的成本。
* ：安全範圍的停車率上限 。（比率）
* ：安全範圍的停車率下限 。（比率）
* ： 調度車在單位時間內移動的距離（or 時間） 上限。
* : 懲罰係數（各站沒補足下限或補超過上限）
* ：卡車虛擬點：*表示第 m台車在該時段最後的車上腳踏車數，（Final點的概念）虛擬點？？*

Set variables:

* 表示有 i 個站點。
* 表示有 m 台調度車。
* 表示有 t 個時段。

由 m 這台車，從 i 站點拿走的量。 （如果這個是 0 ）

由 m 這台車，從 j 站點放入的量。

/\* 表示由 m 這台車，從 i 移動到 j 的量 \*/

= 1 表示由 m 這台車，從 i 移動到 j ， otherwise = 0。 （i != j）

= 各站沒補足下限的比率差

= 各站補超過上限的比率差

= 1 表示第 m 台車出動， otherwise = 0。（一天，目前的假設）

= 1 表示第 m 台車在第 t 個時段出動， otherwise = 0。（一小時）

不知道結果會不會連起來、會不會瞬間跳躍、以及不知道終點在哪

虛擬點還沒放完（晚點處理）

**目標式**

最小化 各車加總的（租用＋使用費）＋移動成本＋（沒補足或補超過）的懲罰

s.t.

1. 限制一個站點只能有一台車經過（進入）??
2. 限制一個站點只能有一台車經過（移出）??
3. 停車率（）要大於下界  
   , max { ,0}
4. 停車率（）要小於上界

, max { ,0}

1. 每趟總和時間（or距離）不可超過指定上限 L
2. 每台車的在每個站點載運量有限（ A ）??
3. 每台車拿上車的量有限（ A ）
4. 每台車拿下車的量有限（ A ）
5. 各站點的 移入總量 要 0
6. 各站點的 移入總量+已有車數要 該站點 停放空間上限
7. 各站點的 移出總量 要 該站點停車數量

5/30 老師給的回饋：

1. 參考VRP的列式
2. 參考TSP的constraints
3. 對某台車，限制沒經過的點拿上拿下的數量為0
4. 每個站點上的 flow in = flow out
5. 每台車的 flow in 、 flow out 小於容量上限（）
6. 限制車子開始與結束，保證會串成一串。
7. 分段做 ?
   1. Phase-I：校內先分區
   2. Phase-II：排單線

05/30

**參數**

：一台調度車容量上限。

：安全範圍的停車率上限 。（比率）

：安全範圍的停車率下限 。（比率）

： 調度車移動的距離（或時間） 上限。(60 min)

: 懲罰係數（各站沒補足下限或補超過上限）

: 表示調度車從 i 移動到 j的成本。 (1 - 30 min)

Set variables: 表示有 i 個站點。

調度車從 i 站點拿走的量。 （如果這個是 0 ）

調度車放入 i 站點的量。

/\* 有m台腳踏車從 i 移到 j \*/

表示調度車有經過站點 i 。 ， otherwise = 0。

= 1 表示調度車從 i 移動到 j ， otherwise = 0。 （i != j）

表示這台車在第 i 站點做事前有 台車在車上。

= 站點 i 沒補足下限的比率差

= 站點 i 補超過上限的比率差

### P\_i?

**目標式**

最小化 移動成本＋（沒補足或補超過）的懲罰

s.t.

1. 車子的路線，兩個都走到才有機會連在一起（ 與 與 的關係 ）

1. 拿出、拿入後與該點的貨車上總量不可以超過貨車容量（xabz的關係）

3’’ 貨車上狀況：拿完之後的結果要等於下一次的開始

1. 第一個站點（手動挑，並命名為）### 應該是上一個小時的輸入?
2. 限制 a 跟 b 的 upper bound，沒經過站點 i 的話， 和 應該要是0。（ 和 是總車子數量）

1. 更新後的停車率（）要大於下界
2. 更新後的停車率（）要小於上界
3. 每趟總和時間（or距離）不可超過指定上限 L??
4. flow in = flow out ??
5. 各站點的 移入總量+已有車數要 該站點 停放空間上限
6. 各站點的 移出總量 要 該站點停車數量
7. 不可以跑過去又跑回來

1. 不可以從同一站點出發及抵達同一站點

1. 一個站點只能出發一次/抵達一次

1. 拿走a 與放下b不能同時大於零
2. 若有站點i放車或取車，就必定有從站點i出發或抵達站點i的路線

（或必須=1，看是抵達還是出發）

1. 在所有站點的調度車上的數量（）永遠小於等於從所有站點拿走的車的數量

（）

1. 對虛擬點的限制
2. detect cycle 問題(subtour)

6/2 琳瑄 （<https://reurl.cc/Lmebxe> ）

**參數**

：一台調度車容量上限。

：安全範圍的停車率上限 。（比率）

：安全範圍的停車率下限 。（比率）

： 調度車移動的距離（或時間） 上限。(60 min)

: 懲罰係數（各站沒補足下限或補超過上限）

: 表示調度車從 i 移動到 j的成本。 (1 - 30 min)

Set variables: 表示有 i 個站點。

調度車從 i 站點拿走的量。

調度車放入 i 站點的量。

表示調度車有經過站點 i 。 ， otherwise = 0。

= 1 表示調度車從 i 移動到 j ， otherwise = 0。 （i != j）

表示這台車在第 i 站點進站前有 台車在車上。

表示這台車在第 i 站點進站搬運後有 台車在車上

= 站點 i 沒補足下限的比率差

= 站點 i 補超過上限的比率差

**目標式**

最小化 移動成本＋（沒補足或補超過）的懲罰（！）

s.t.

1. 車子的路線，兩個都走到才有機會連在一起（ 與 與 的關係 ）（！）

1. 貨車上狀況：進站前結果＋搬運後要等於進站後（！）
2. 進 i 站後結果等於的進 j 站前調度車上數量（！）
3. 限制 a 跟 b 的 upper bound，沒經過站點 i 的話， 和 應該要是0。（ 和 是總車子數量）（！）

1. 更新後的停車率（）要大於下界（！）
2. 更新後的停車率（）要小於上界（！）
3. 每趟總和時間（or距離）不可超過指定上限 L（！）
4. flow in = flow out （！）
5. 各站點的 移入總量+已有車數要 該站點 停放空間上限（！）
6. 各站點的 移出總量 要 該站點停車數量（！）
7. 不可以跑過去又跑回來（！）

1. 不可以從同一站點出發及抵達同一站點（！）

1. 一個站點只能出發一次/抵達一次（！）
2. 拿走a 與放下b不能同時大於零（！）
3. 拿出、拿入後與該點的貨車上總量不可以超過貨車容量（xabz的關係）（！）

,

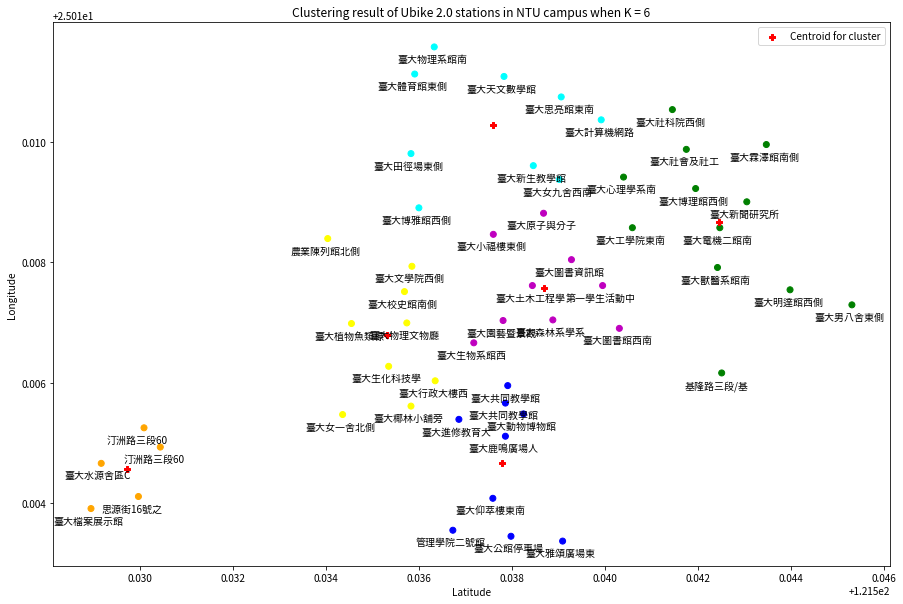
1. 對虛擬點的限制

（！）（目前gurobi跑不出來）

1. 若有站點i放車或取車，就必定有從站點i出發或抵達站點i的路線（差這個）（！）

（或必須=1，看是抵達還是出發）

**K-Means演算法**



| **Cluster No.** | **Cluster total space** | **Cluster occupied space** | **Cluster vacant space** | **Cluster stations** |
| --- | --- | --- | --- | --- |
| 0 | 127 | 8 | 119 | 5 |
| 1 | 264 | 108 | 154 | 12 |
| 2 | 245 | 84 | 145 | 9 |
| 3 | 136 | 26 | 110 | 9 |
| 4 | 152 | 72 | 72 | 9 |
| 5 | 192 | 24 | 168 | 9 |

K-means is one of the most popular Unsupervised Machine Learning Algorithms Used for Solving Classification Problems. K Means segregates the unlabeled data into various groups, called clusters, based on having similar features, common patterns.

In this case, every station is depicted on the graph according to their latitude and longitude. We have presumed that stations are split into 6 groups.

<需要補充K-means演算法過程[[2]](#footnote-1)>

We calculate Euclidean distance to calculate distances between data points and centroids.

<進度：可切分53x53的distance matrix成6個cluster内部n x n的distance matrix>

1. 單一站點的空車率：即該站點目前空位 / 該站點總車柱數 [↑](#footnote-ref-0)
2. [K Means Clustering Simplified in Python | K Means Algorithm (analyticsvidhya.com)](https://www.analyticsvidhya.com/blog/2021/04/k-means-clustering-simplified-in-python/) [↑](#footnote-ref-1)