

# 郵件投遞成功率提升 — 最佳路徑規劃

---

孫君傳、劉正宇、簡辰安

# 理念—回歸原點

中華郵政有著穩固的基礎客戶與規模，也一直致力於諸如i郵箱、民間通路、便利商店合作等創新。

但除此之外，我們認為中華郵政可以回到原點，透過郵務數據的分析，提高郵件投遞的效率來：

- 縮短原先郵務士送件及善後的時間
- 解放人力資源以發展新業務

## 優勢

- 紙本郵件領先市場
- 員工數量多且優質
- 穩定的金融業務→支援大膽郵務決策

## 劣勢

- 易固化的組織結構
- 國營事業的公共責任

## 機會

- 資訊科技成熟
- 追求效率的郵務市場

## 威脅

- 其他物流公司對於環境的快速應變

# 摘要

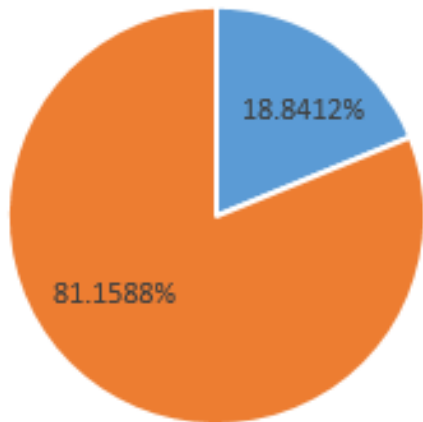
---

- 目的：提高郵件投遞成功率，減少重複投遞成本以增加郵務士及郵局整體工作效率
- 投遞成功率提升方案：
  - 找出影響投遞成功率的因素：資料視覺化分析與統計檢定
  - 獲得最佳投遞時間：提出透過預測模型、知識管理系統(KMS)及事先預約的方式，讓郵務士在投遞前了解手上郵件投遞狀況。
  - 發展最佳路線規劃：將投遞過程描繪成VRPWT問題，配合最佳投遞時間，計算出最佳的投遞路線。
- 成果：透過程式驗證方案可行性，確實提高投遞成功率以降低成本

# 一、背景資料分析

-郵件投遞成功率與影響投遞成功率之參數

整年約18.8%的郵件會投遞失敗

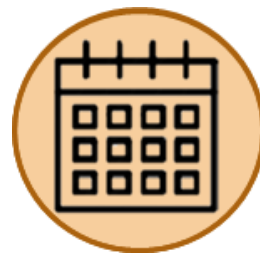


其中以下因素會影響投遞成功率



**雨量：**

雨量大到一定程度投遞成功率有顯著下降



**時段，星期幾：**

早上、中午高機率成功  
假日的成功率低於平日

# 時段

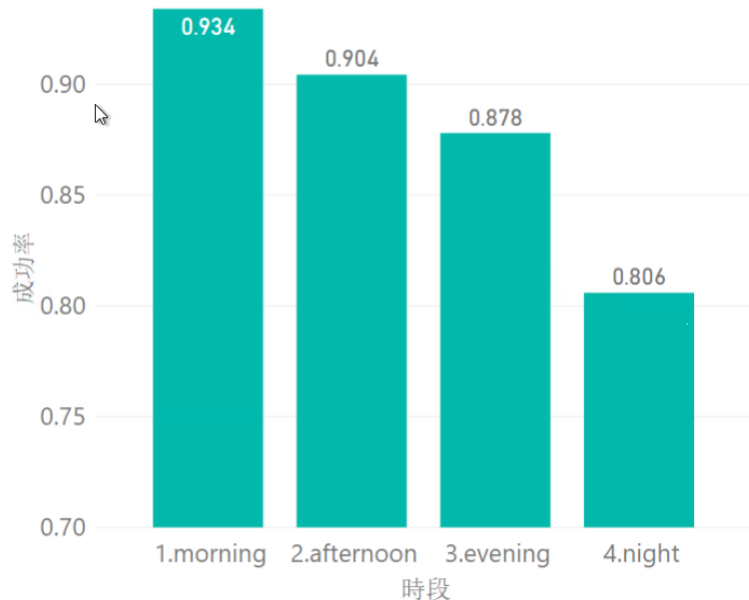
右圖可發現時間是影響成功率的顯著原因

高機率成功:  
早上、中午

高機率失敗:  
下午、晚上

可能原因:  
特定時段收件者比較不容易在家或無法收件

在各個時段中的投遞成功率



# 投遞日性質(星期幾)

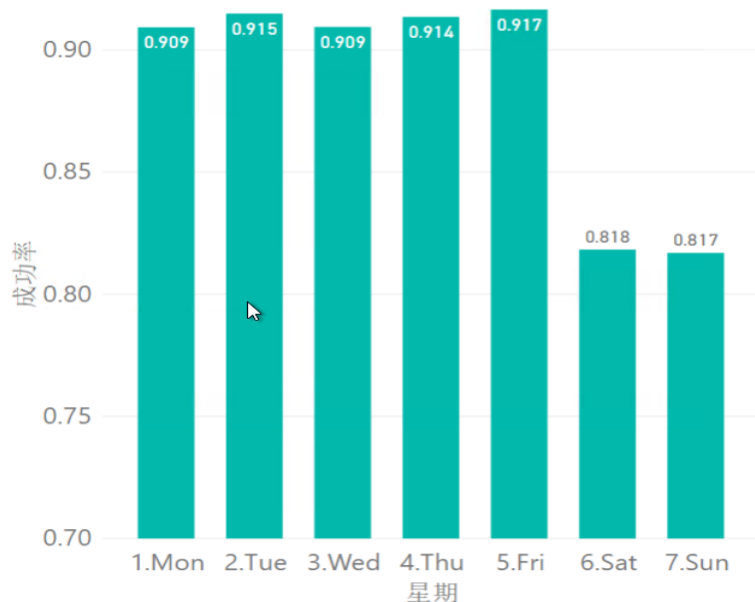
在一個星期中各日的投遞成功率

由右圖可以發現到

假日的成功率低於平日

可能原因：

部分民眾假日不會在家中，無法簽收郵件



# 天氣

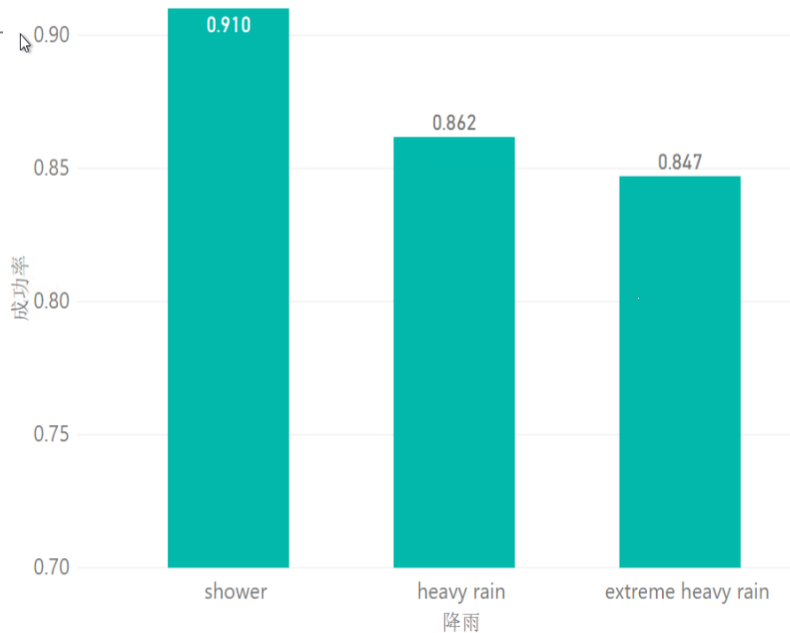
天氣是影響投遞的重要因素之一，從資料中可以看出：

雨量大到一定程度，投遞成功率有顯著下降

可能原因：

輕微下雨不太影響收件者與郵務士作息  
當雨大到一定程度，便會影響到郵務士  
遞送的難易度與客戶的行為。

在各種雨量中的投遞成功率



資料來源：中央氣象局所提供的雨量資料月報表

# 可能影響因素

---

我們希望找出統計證據，以證明不同的因素確實會導致不同的寄件成功率，如此便能蒐集相關變數，並建置相對應的預測模型。

Why

懷疑時間、日期等是影響寄件成功率的因素

How

使用 Pearson's Chi-Squared test of contingency table

What

找出統計證據，證明不同的變數會影響寄件成功率



# Chi-Squared test of contingency table

## 不同日

$H_0$ : 不同日的郵件寄送成功率 一樣  
 $H_1$ : 不同日的郵件寄送成功率不一樣

## 不同時段

$H_0$ : 不同時間的郵件寄送成功率 一樣  
 $H_1$ : 不同時間的郵件寄送成功率不一樣

## 不同雨量

$H_0$ : 不同天氣(雨量)的郵件寄送成功率 一樣  
 $H_1$ : 不同天氣(雨量)的郵件寄送成功率不一樣

by 中央氣象局所提供的雨量資料月報表

x1	x2	1.Mon	2.Tue	3.Wed	4.Thu	5.Fri	6.Sat	7.Sun
fail		10517	12978	10593	11134	12473	2806	3028
success		105373	139453	106318	117602	136929	12635	13510

Pearson's Chi-squared test

data: tab1  
X-squared = 3416, df = 6, p-value < 2.2e-16

x1	x2	1.morning	2.afternoon	3.evening	4.night
fail		21173	25527	6707	10122
success		300089	241414	48289	42028

Pearson's Chi-squared test

data: tab1  
X-squared = 9816.2, df = 3, p-value < 2.2e-16

x1	x2	extreme	heavy rain	heavy rain shower
fail		1826	805	60898
success		10117	5021	616682

Pearson's Chi-squared test

data: tab1  
X-squared = 716.5, df = 2, p-value < 2.2e-16

## 小結：

透過Pearson's Chi-squared test，我們有統計證據說明：不同時間點、日期和雨量所對應的投遞成功率並不相同，不論兩者間是相關還是因果關係，這些變數都是我們建置預測模型的重要參考。

## 二. 獲取最佳投遞時間之方法

01.

### 預測模型

蒐集影響投遞成供的資料，使用統計和機械學習模型，預測客戶收件的最佳時間。

02.

### 知識系統

整合資訊資源，包括郵務士自身經驗和投遞的歷史資料，紀錄客戶收件的最佳時間。

03.

### 事先預約

客戶主動預約收件時間、地點，提供郵務士更彈性的收件模式和精確時間。

考慮成功收件的時間，能減少重複投遞及善後的次數，進而降低運送成本  
接下來詳述以上三種收集最佳投遞時間的方案

# 方法1：預測模型

將投遞成功與否設為應變數，將更多可能影響的參數：日期、時間、氣溫、雨量、紫外線指數和 同時同區號的包裹數 設為自變數，由於成功與否為決定性問題(是/否)，我們選擇使用 **logistic regression** 建置模型。

Why

了解參數對成功率影響的程度，提供規劃投遞路線的依據

How

使用R language 建置 Logistic Regression 模型

What

預測郵件在不同時間點、星期、天氣等狀態下的投遞成功率

# Logistic Regression

## 參數

	success	time	week	address	temp	rain	UVI	amount
2109356	I4	evening	WED	80700	19.6	1	9	234107
2109357	I4	afternoon	FRI	80700	15.0	1	7	234107
2109358	H4	afternoon	THU	80200	23.7	7	8	170119
2109359	H4	afternoon	WED	80200	24.0	1	10	170119
2109360	I4	evening	THU	80200	16.5	3	6	170119
2109361	I4	evening	WED	80700	21.1	1	11	234107
2109362	H4	afternoon	WED	80700	15.5	4	7	234107
2109363	I4	afternoon	FRI	80700	15.0	1	7	234107
2109364	H4	afternoon	THU	80700	23.5	1	7	234107
2109365	I4	afternoon	FRI	80700	15.0	1	7	234107

資料來源：800584局號中郵件狀態代碼被登記為I4(成功)與H4(不成功)、且掛號號碼中有郵遞區號的投遞資料，共約97萬筆，每筆資料加上郵遞區號、當日平均氣溫、雨量、紫外線指數和同梯寄件數

時間：早上(9~11點)、下午(12~15)、傍晚(16~19)和晚上(剩餘時間)

訓練樣本：7成的資料作訓練樣本，剩下3成作比對

## model

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9161    0.1958    0.2623    0.3314    3.5482

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.300e-01  1.789e-02 -18.450 < 2e-16 ***
timeevening  3.366e+00  6.497e-03  518.096 < 2e-16 ***
timemorning  1.520e+00  1.564e-02  97.174 < 2e-16 ***
timenight   -5.604e+00  3.149e-01 -17.793 < 2e-16 ***
weekMON      6.614e-02  8.498e-03   7.783 7.07e-15 ***
weekSAT      8.900e-01  2.656e-02  33.503 < 2e-16 ***
weekTHU      3.676e-02  7.763e-03   4.736 2.18e-06 ***
weekTUE      3.836e-01  8.262e-03  46.427 < 2e-16 ***
weekWED      6.146e-01  8.415e-03  73.040 < 2e-16 ***
temp        -1.748e-02  8.109e-04 -21.556 < 2e-16 ***
rain        -1.778e-02  1.047e-03 -16.970 < 2e-16 ***
UVI          2.409e-02  1.108e-03  21.731 < 2e-16 ***
amount       3.025e-06  2.892e-08  104.597 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1402502  on 1385149  degrees of freedom
Residual deviance:  942039  on 1385137  degrees of freedom
AIC: 942065

Number of Fisher Scoring iterations: 7
```

# 檢查模型

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9161   0.1958   0.2623   0.3314   3.5482

Coefficients:
(Intercept) -3.300e-01  1.789e-02 -18.450 < 2e-16 ***
timeevening  3.366e+00  6.497e-03  518.096 < 2e-16 ***
timemorning  1.520e+00  1.564e-02  97.174 < 2e-16 ***
timenight   -5.604e+00  3.149e-01 -17.793 < 2e-16 ***
weekMON      6.614e-02  8.498e-03   7.783 7.07e-15 ***
weekSAT      8.900e-01  2.656e-02  33.503 < 2e-16 ***
weekTHU      3.676e-02  7.763e-03   4.736 2.18e-06 ***
weekTUE      3.836e-02  8.262e-03   4.627 < 2e-16 ***
weekWED      6.146e-01  8.415e-03  73.040 < 2e-16 ***
temp        -1.748e-02  8.109e-04 -21.556 < 2e-16 ***
rain        -1.778e-02  1.047e-03 -16.970 < 2e-16 ***
UVL         2.409e-02  1.108e-03  21.731 < 2e-16 ***
amount      3.025e-06  2.892e-08 104.597 < 2e-16 ***

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1402502  on 1385149  degrees of freedom
Residual deviance:  942039  on 1385137  degrees of freedom
AIC: 942065

Number of Fisher Scoring iterations: 7
```

Deviance residuals is centered at 0 (good)

Wald's test for all coefficients:  
all statistically significant

模型有82.8%的正確率

成功將成功案例預測成成功的準確度有9成

時間對成功率的影响較顯著

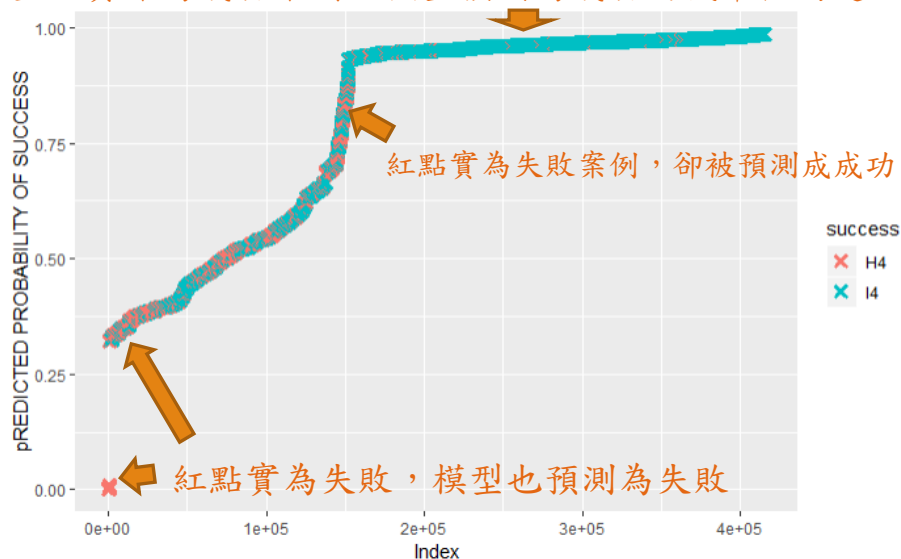
## Confusion Matrix and Statistics

Prediction	Reference	
	H4	I4
H4	43693	30357
I4	41189	300305

Accuracy : 0.8278  
95% CI : (0.8267, 0.829)  
No information Rate : 0.7957  
P-value [Acc > NIR] : < 2.2e-16  
Kappa : 0.444  
McNemar's Test P-value : < 2.2e-16  
Sensitivity : 0.5147  
Specificity : 0.9082  
Pos Pred Value : 0.5900  
Neg Pred Value : 0.8794  
Prevalence : 0.2043  
Detection Rate : 0.1051  
Detection Prevalence : 0.1782  
Balanced Accuracy : 0.7115  
'Positive' Class : H4

# 預測模型結論

藍點實際為成功案例，模型預測為成功的機率值高達9成



此圖為測試資料被歸類為成功的機率值，同前頁模型的 specificity 所示，此模型對成功郵件的辨識度較高。

- 準確度：0.8278
- 影響成功率參數：時間 > 星期 > 天氣
- 具代表性參數不足使目前模型不易預測會投遞失敗的郵件
- 建議提升模型品質的方法：
  - 使用不同模型如：SVM, random forest 和 MLP 等
  - 放入更精確的地理位置、去識別化用戶資訊等參數
  - 結合長期使用郵政服務情形的統計和推測人口結構

## 方法2：知識管理系統(KMS)

---

現在許多郵務士早已自己有一套運送習慣，知道哪些客戶什麼時候較容易送到，卻缺乏一個有系統的整合，因此可以建置一個資料庫，儲存並管理這些寶貴的經驗，好處如下：

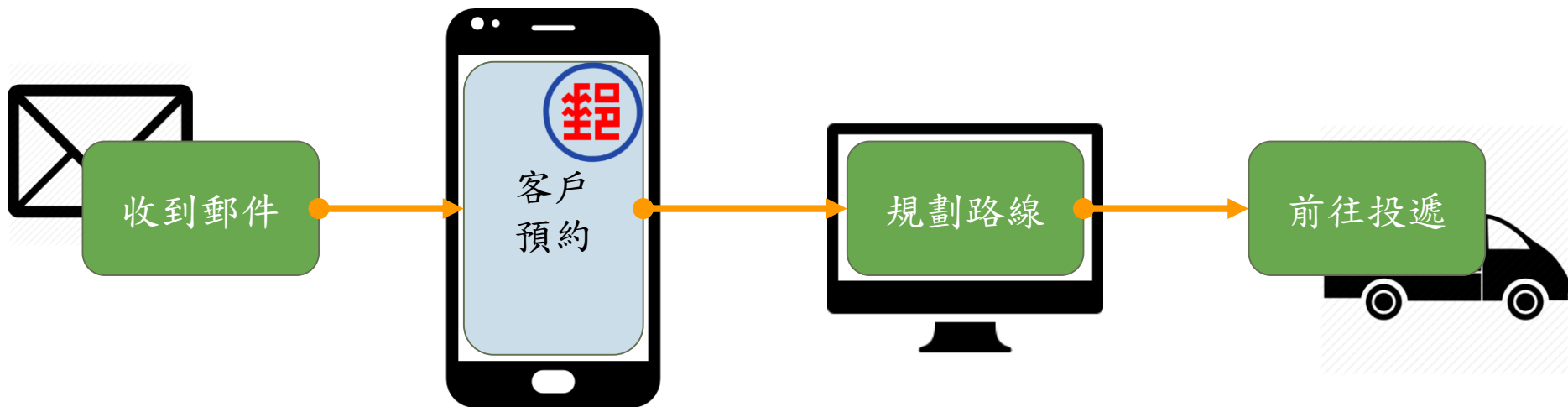
- 歷史資料可輔助完善方法1之預測模型
- 只需資料儲存及詢問郵務士，成本低廉
- 提升人力調度彈性：

彌平新手與老手的經驗差，新手上路或投遞區域更換效率不會變差，進而可依據當日郵件總量或郵局營運情況進行人力調派



## 方法3：APP事先預約

- 被動收集：由收件者**主動預約**希望的收件時間（比起預測時間區間，更為精準）
- 衍生應用：除了預約時間，更能**預約收件地點**（ex: 改到上班地點取貨，更為彈性）





### 三. 最佳路徑規劃

目標

確保郵差一次就成功投遞，進而減少重複投遞成本和增加郵差投遞效率

方法

結合客戶郵址、道路資訊、過去運送成本統計和給定送貨時間，將投遞過程變成整數線性規劃(ILP)的問題

結果

將運送過程視為  
**Vehicle Routing Problem with Time Windows** 問題，  
算出在不同時間點送達至不同客戶的最佳路線。

# 做法



## 道路距離

使用google提供的  
distance-matrix api



## 投遞時間

透過預測模型預測、  
知識庫的紀錄或事  
前預約而得到



## 整數規劃

使用solver計算：



或  
提供的

OR-tools等工具



## 結果呈現

得到抵達各點的時  
間和所屬郵差，可  
將結果輸出成圖表

## 模型

s.t. : $\cup$

$$\sum_k^V \sum_j^N x_{ijk} = 1, \quad \forall i \in C \quad \cup$$

$$\sum_j^N x_{0jk} = 1, \quad \forall k \in V \quad \cup$$

$$\sum_i^N x_{ihk} - \sum_j^N x_{hjk} = 0, \quad \forall h \in C, k \in V \quad \cup$$

$$\sum_i^N x_{i,n+1,k} = 1, \quad \forall k \in V \quad \cup$$

$$\sum_i^C \sum_j^N d_i x_{ijk} \leq q, \quad \forall k \in V \quad \cup$$

$$\sum_k^V \sum_j^N x_{0jk} \leq |V|, \quad \forall k \in V, j \in N \quad \cup$$

$$s_{ik} + t_{ij} - s_{jk} \leq M_{ij}(1 - x_{ijk}), \quad \forall i, j \in N, k \in V \quad \cup$$

$$a_i \leq s_{ik} \leq b_i, \quad \forall i \in N, k \in V \quad \cup$$

$$x_{ijk} \in \{0,1\}, \quad \forall i, j \in N, k \in V \quad \cup$$

$$s_{ik} \in N^+, \quad \forall i \in N, k \in V \quad \cup$$

# VRPWT 模型

## 決策變數

$$x_{ij}^k := \begin{cases} 1, & k_{th} \text{ 郵差是否從 } i_{th} \text{ 客戶端移動至 } j_{th} \text{ 客戶端} \\ 0, & o/w \end{cases} \quad \cup$$

$$s_{ik} := k_{th} \text{ 郵差開始服務 } i_{th} \text{ 客戶的時間} \quad \cup$$

## 決策目標

Obj:  $\cup$

$$\text{Min} \sum_k^V \sum_i^N \sum_j^N c_{ij} x_{ijk} \quad \leftarrow \text{總運輸成本}$$

## 模型參數

$$c_{ij} := \text{從 } i_{th} \text{ 客戶到 } j_{th} \text{ 客戶的交通成本} \quad \cup$$

$$t_{ij} := \text{從 } i_{th} \text{ 客戶到 } j_{th} \text{ 客戶的所需時間} \quad \cup$$

$$q := \text{車輛載重/空間} \quad \cup$$

$$d_i := i_{th} \text{ 客戶的計件重量/體積} \quad \cup$$

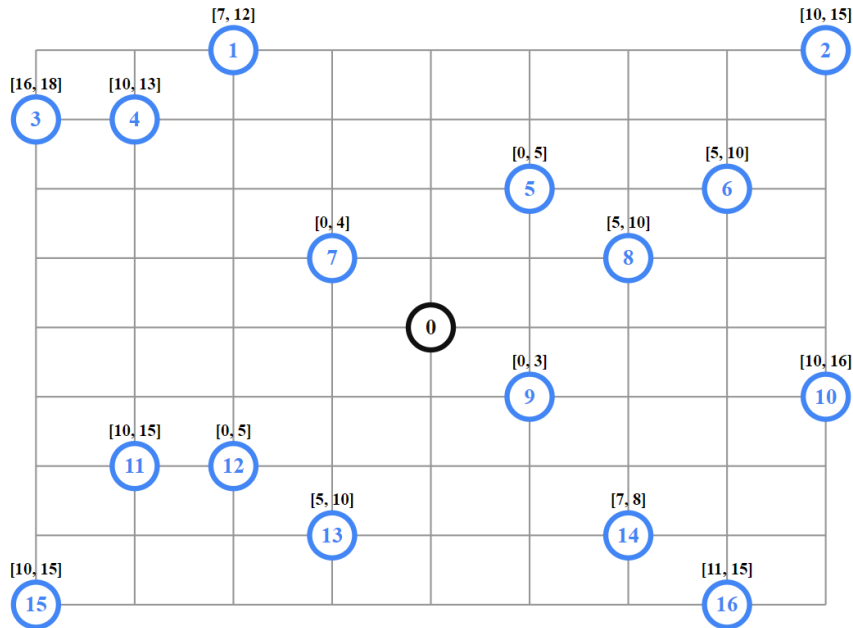
$$[a_i, b_i] := i_{th} \text{ 客戶的最佳投遞時間} \quad \cup$$

$$M_{ij} := \max\{b_i + t_{ij} - a_j\} (\text{移動時間上限}) \quad \cup$$

使用預測模型或  
過去歷史資料  
推測高機率投遞  
成功的時間區間

# GRAPH範例

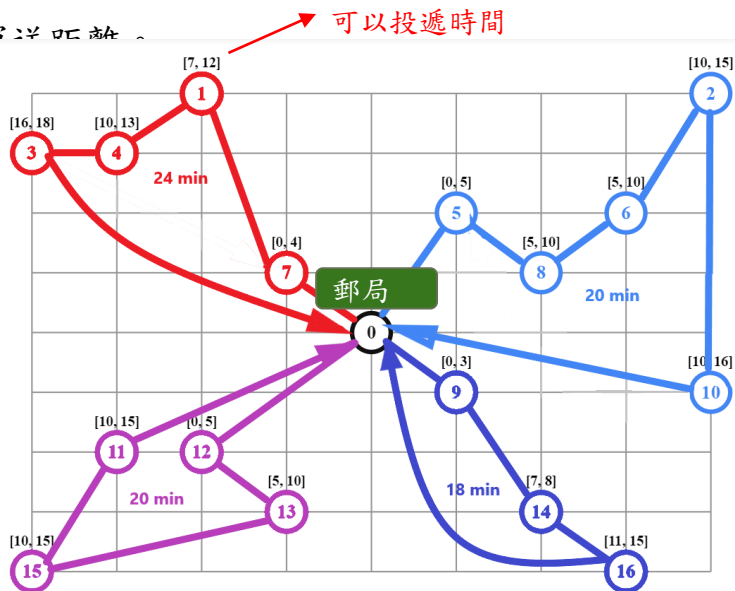
```
data['time_matrix'] = [  
    [0, 6, 9, 8, 7, 3, 6, 2, 3, 2, 6, 6, 4, 4, 5, 9, 7],  
    [6, 0, 8, 3, 2, 6, 8, 4, 8, 8, 13, 7, 5, 8, 12, 10, 14],  
    [9, 8, 0, 11, 10, 6, 3, 9, 5, 8, 4, 15, 14, 13, 9, 18, 9],  
    [8, 3, 11, 0, 1, 7, 10, 6, 10, 10, 14, 6, 7, 9, 14, 6, 16],  
    [7, 2, 10, 1, 0, 6, 9, 4, 8, 9, 13, 4, 6, 8, 12, 8, 14],  
    [3, 6, 6, 7, 6, 0, 2, 3, 2, 2, 7, 9, 7, 7, 6, 12, 8],  
    [6, 8, 3, 10, 9, 2, 0, 6, 2, 5, 4, 12, 10, 10, 6, 15, 5],  
    [2, 4, 9, 6, 4, 3, 6, 0, 4, 4, 8, 5, 4, 3, 7, 8, 10],  
    [3, 8, 5, 10, 8, 2, 2, 4, 0, 3, 4, 9, 8, 7, 3, 13, 6],  
    [2, 8, 8, 10, 9, 2, 5, 4, 3, 0, 4, 6, 5, 4, 3, 9, 5],  
    [6, 13, 4, 14, 13, 7, 4, 8, 4, 4, 0, 10, 9, 8, 4, 13, 4],  
    [6, 7, 15, 6, 4, 9, 12, 5, 9, 6, 10, 0, 1, 3, 7, 3, 10],  
    [4, 5, 14, 7, 6, 7, 10, 4, 8, 5, 9, 1, 0, 2, 6, 4, 8],  
    [4, 8, 13, 9, 8, 7, 10, 3, 7, 4, 8, 3, 2, 0, 4, 5, 6],  
    [5, 12, 9, 14, 12, 6, 6, 7, 3, 3, 4, 7, 6, 4, 0, 9, 2],  
    [9, 10, 18, 6, 8, 12, 15, 8, 13, 9, 13, 3, 4, 5, 9, 0, 9],  
    [7, 14, 9, 16, 14, 8, 5, 10, 6, 5, 4, 10, 8, 6, 2, 9, 0],  
]
```



# GRAPH範例

使用程式運行VRPTW規劃路線，可考慮到顧客有空的時間，  
避免投遞失敗，且可考慮人腦不易想出的路線，得以減少整體運送距離。

系統推薦路線

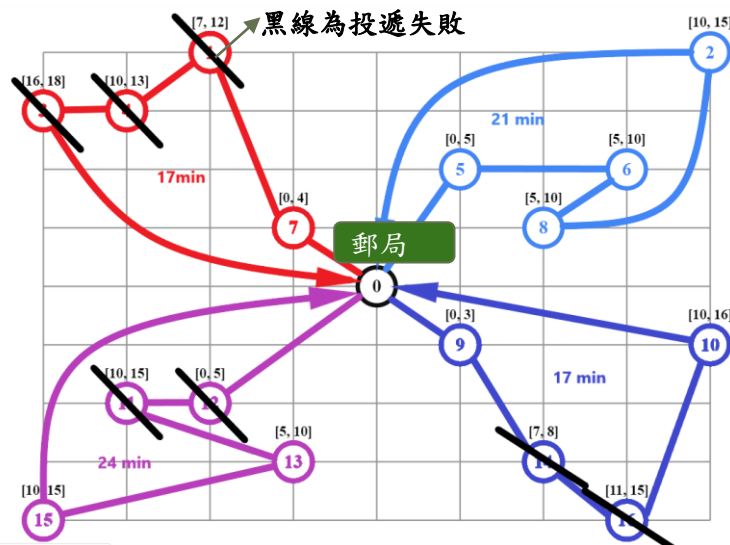


使用OR-tools計算各郵差抵達不同客戶的時間的結果

```
Route for vehicle 0:  
0 Time(0,0) -> 9 Time(3,4) -> 14 Time(7,8) -> 16 Time(14,14) -> 0 Time(21,21)  
Time of the route: 21min
```

使用greedy演算法模擬郵差實際投遞路線

1. 一味前往最近的地址，整體路線不一定最短
2. 不清楚投遞時間，會發生投遞失敗的個案



原始路線

# 實際範例

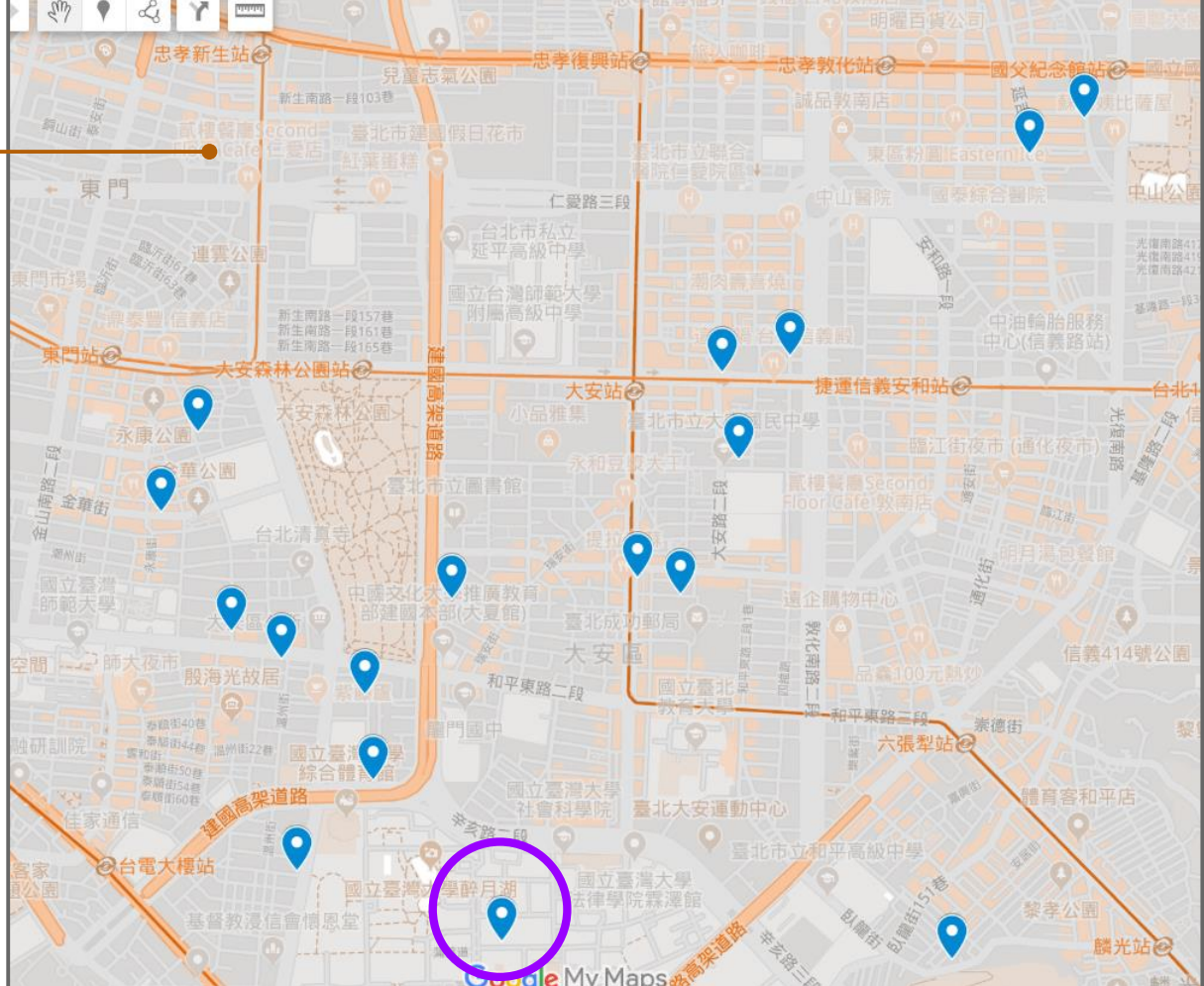
大安區(郵遞區號106)

隨機產生的16個點

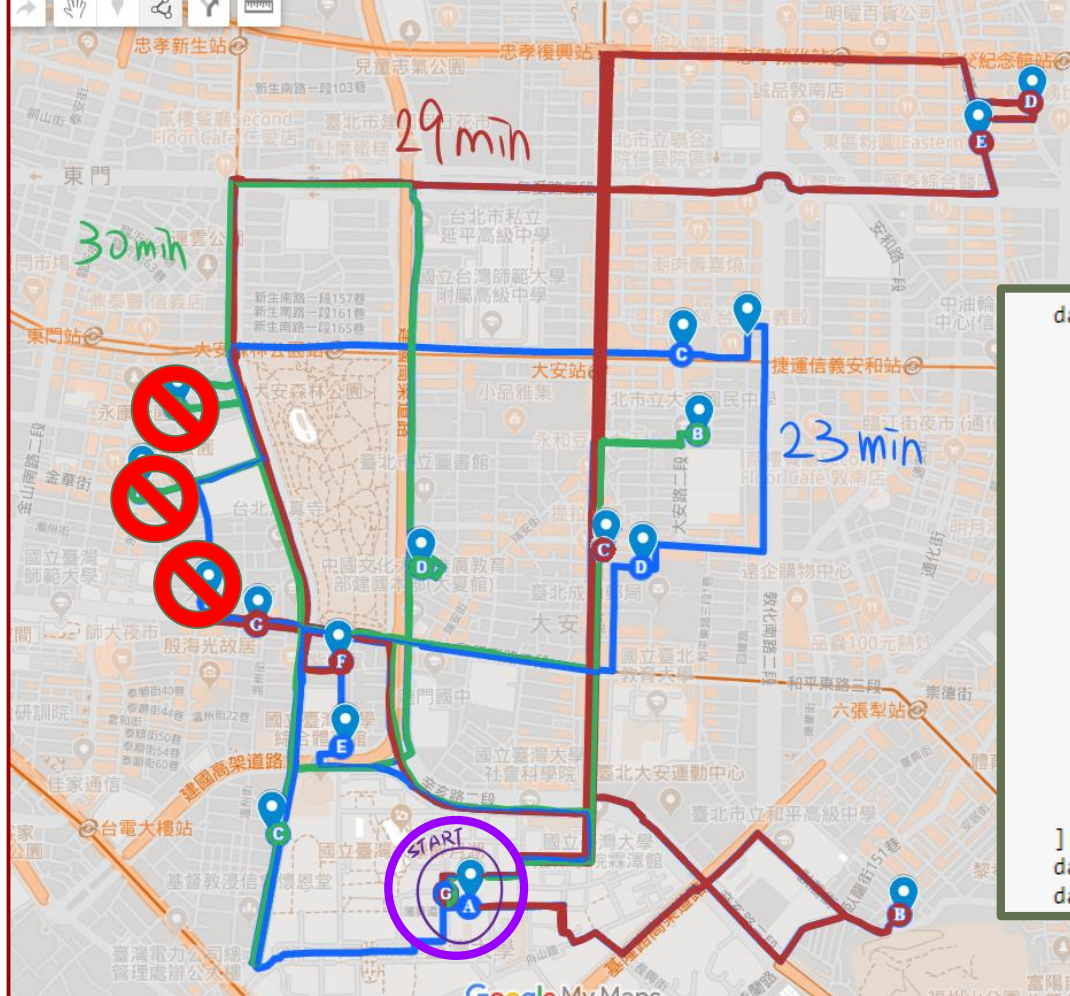
+

郵局

```
data['time_matrix'] = [  
    [ 0,189,515,259,611,746,369,943,941,577,504,597,230,409,761,722,130,833,729,602], # 0  
    [252, 0,465,209,561,741,319,979,929,527,303,385,367,209,561,615, 61,919,820,401], # 1  
    [595,677, 0,501,853,655,497,765,662,684,251,332,317,156,508,587,163,884,780,372], # 2  
    [243,324,390, 0,500,660,236,918,782,446,334,415,382,236,462,517,339,789,685,303], # 3  
    [567,660,809,470, 0,610,642,763,761,563,426,507,492,331,364,415,338,718,572, 61], # 4  
    [584,666,418,487,583, 0,504,390,287,257,537,619,371,440,536,176,457,491,388,209], # 5  
    [261,342,299,166,518,572, 0,876,778,358,586,668,432,492,525,186,498,503,400,150], # 6  
    [907,988,742,812,763,469,794, 0, 94,626,636,729,827,539, 76,673,673,723,721,575], # 7  
    [869,950,702,771,730,425,774,129, 0,541,448,530,282,350,576,451,368,601,498,327], # 8  
    [454,535,493,359,392,289,366,606,502, 0,454,507,767,428,309,610,601,807,805,522], # 9  
    [449,308,332,395,576,793,690,690,763,468, 0,304,383,437,592,730,715,850,556,844], # 10  
    [399,258,282,345,526,780,640,649,713,551,318, 0,182,236,391,662,514,649,604,665], # 11  
    [229,458,474,553,623,663,684,941,488,957,251,102, 0,195,375,634,491,596,562,612], # 12  
    [324,175, 73,222,403,707,519,580,590,604,412,278,200, 0,242,594,371,550,503,562], # 13  
    [730,581,520,556,538,686,593, 84,786,294,425,277,263,341, 0,462,266,452,389,465], # 14  
    [448,480,450,330,219, 47,136,671,162,683,401,433,403,283,171, 0, 89,624,115,636], # 15  
    [140, 61,140,193,348,620,472,606,561,622,462,437,424,354,161,233, 0,634,289,625], # 16  
    [737,770,766,645,565,426,514,822,523,889,761,612,598,625,597,642,584, 0,757,384], # 17  
    [717,750,735,614,503,331,428,728,447,856,311,344,314,193,265,499,313,664, 0,676], # 18  
    [453,305,291,284, 77,336,163,480,381,493,688,539,479,515,497,657,593,397,744, 0], # 19  
]  
for i in range(20):  
    for j in range(20):  
        data['time_matrix'][i][j] = data['time_matrix'][i][j]/60
```







出局: 2, 6, 10

the result of

V<sub>ehicle</sub>

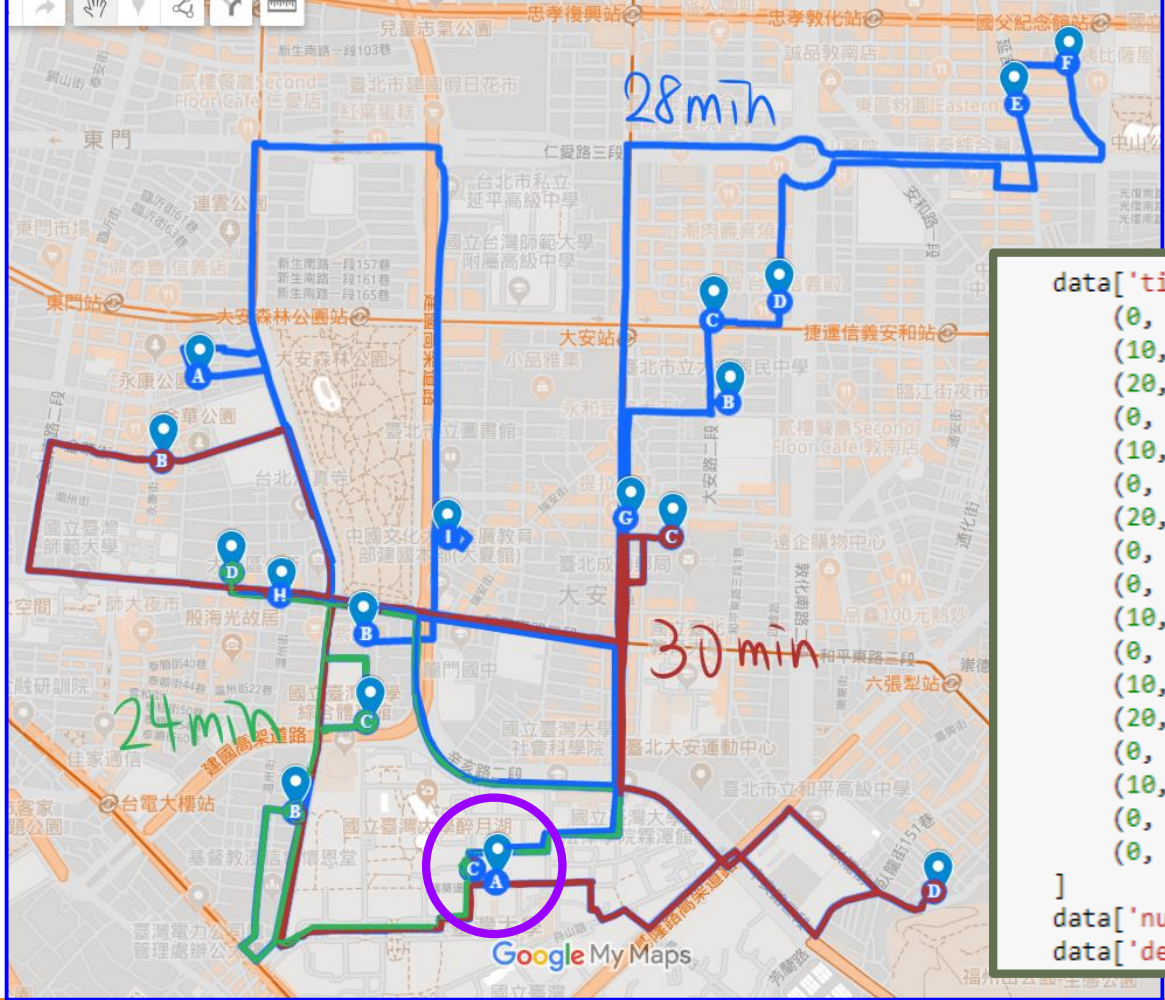
R<sub>outing</sub>

P<sub>roblem</sub>

total: 82min

```
data['time_windows'] = [
    (0, 120), # depot
    (10, 50), # 1
    (20, 60), # 2
    (0, 25), # 3
    (10, 30), # 4
    (0, 30), # 5
    (20, 30), # 6
    (0, 20), # 7
    (0, 20), # 8
    (10, 20), # 9
    (0, 20), # 10
    (20, 30), # 11
    (20, 30), # 12
    (0, 30), # 13
    (10, 40), # 14
    (0, 30), # 15
    (0, 30), # 16
]
```

```
data['num_vehicles'] = 4
data['depot'] = 0
```

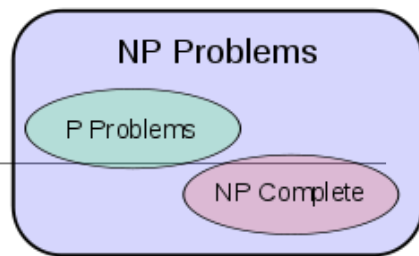


```
data['time_windows'] = [
    (0, 120), # depot
    (10, 50), # 1
    (20, 60), # 2
    (0, 25), # 3
    (10, 30), # 4
    (0, 30), # 5
    (20, 30), # 6
    (0, 20), # 7
    (0, 20), # 8
    (10, 20), # 9
    (0, 20), # 10
    (10, 30), # 11
    (20, 30), # 12
    (0, 30), # 13
    (10, 40), # 14
    (0, 30), # 15
    (0, 30), # 16
]
data['num_vehicles'] = 4
data['depot'] = 0
```

the result of  
**V**ehicle  
**R**outing  
**P**roblem with  
**T**ime  
**W**indows  
**total:82min**



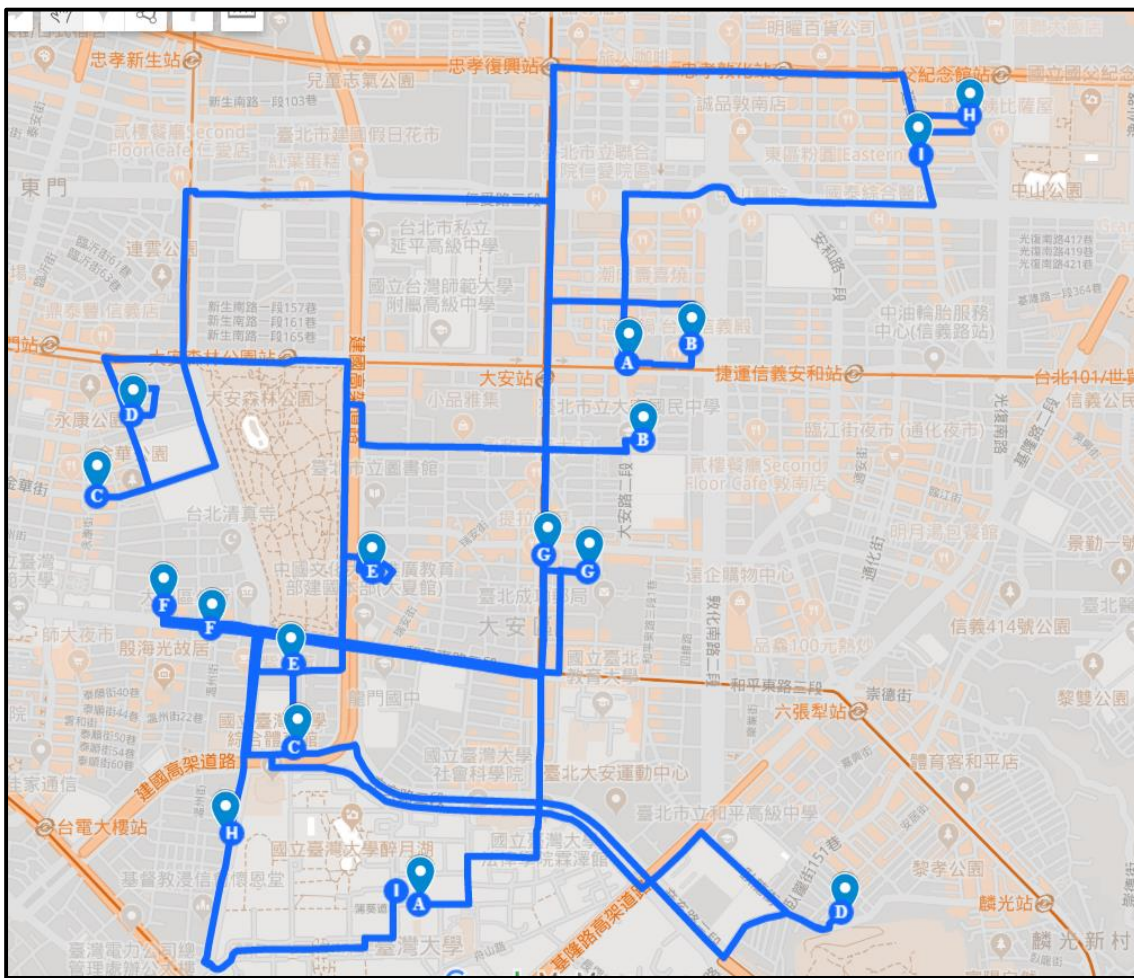
# 最佳路徑結論



- 參數：地址、路況、運送成本統計、郵務士數量、最佳投遞時間
- 成果：在指定時間內投遞的最佳路徑，1. 減少重複投遞和2. 降低運輸成本
- 限制：
  - 此問題為NP-complete，也就是說處理的數量越多，計算過程會越複雜，以至於使用低階配備會無法計算。
  - 對於少量客戶的郵遞區號(幾千戶)，可以透過solver (ex,gurobi, or-tools)直接運算。
  - 如果數量過多，則必須退而求其次使用 heuristic algorithmn，針對VRPTW問題，我們嘗試使用過Genetic algorithmn (1)、Ant colony algorithmn (2)等加入隨機變數的演算法，已逼近最佳答案，運算速度會快上許多，且答案都不錯。

(1) Vehicle Routing with Time Windows using Genetic Algorithms Sam R. Thangiah Artificial Intelligence and Robotics Laboratory Computer Science Department Slippery Rock University Slippery Rock PA 16057 U.S.A.

(2) Necula, R., Breaban, M., Raschip, M.: *Tackling Dynamic Vehicle Routing Problem with Time Windows by means of Ant Colony System*, 2017 IEEE Congress on Evolutionary Computation (CEC), 5-8 June, Donostia - San Sebastián, Spain, pp. 2480-2487, 2017



A  
N  
T  
  
C  
O  
L  
O  
N  
Y



the result of

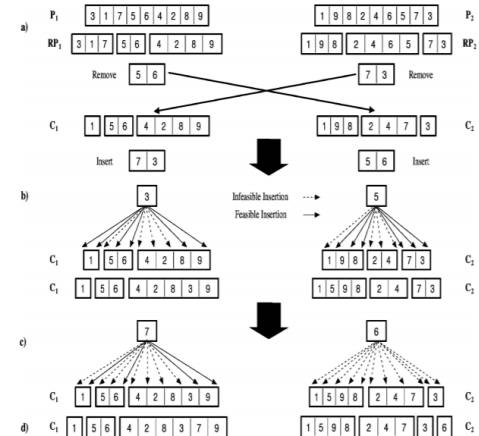
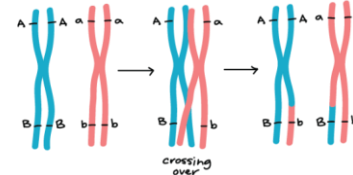
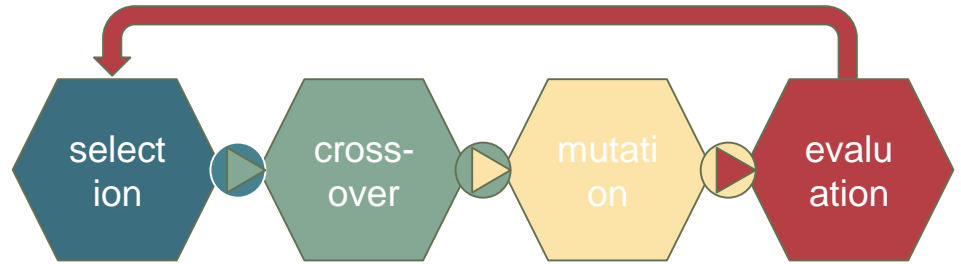
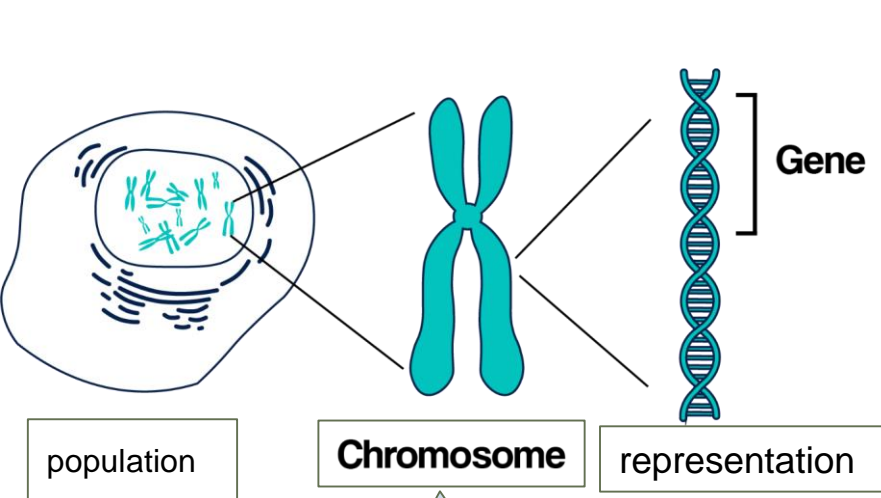
V<sub>ehicle</sub>

R<sub>outing</sub>

P<sub>roblem</sub>

with 1 car  
total:85min

# GENETIC ALGORITHM



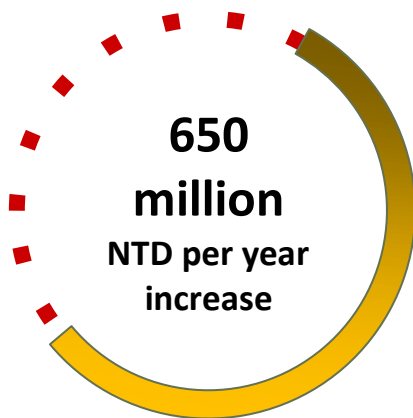
# 效益評估



● 最高提升  
投遞成功率

18.8%

(若取得最佳投遞時間使  
投遞幾乎不投遞失敗)



● 投遞失敗成本計算

失敗件數(3889萬件)  
x 每件花費時間(2.5分)  
x 每分鐘成本(6.73元)

= 654,324,250元



● 人力調度彈性

效率提升、不受經驗限制

● APP可帶來額外效益

ex: 推廣郵局服務...

每分鐘成本=郵務勞務費用/郵局員工數/工作時間(250天、8小時)  
=21,009,391,000/26000/250/8/60=6.73元

失敗案件花費時間=兩次投遞+電話通知=2.5分鐘

# 總結論

---

## ● 目標：

增進郵務士的投遞效率(18.8%失敗率)及減少後續處理投遞失敗的時間

## ● 最佳收件時間：

○ 預測模型、知識管理系統、事先預約APP

## ● 最佳路徑規劃：

給定收件時間並計算最佳路經，提高初次投件成功的機率的同時也壓低運送和時間成本，預估最高可節省6.5億元