



Stock Portfolio 股票投資組合

運用基因演算法最佳化投資組合

期末報告 第四組 組員：

1092518陳俊邑 0613124黃薏臻 0613138林可芸 0813317江詠筑

目錄 / CONTENTS

01 

研究背景與研究目的

03 

研究方法與實驗設定

02 

文獻回顧與研究貢獻

04 

實驗結果與實驗結論



01

研究背景與研究目的

Research background and Research purpose

研究背景



- 投資組合可分散投資風險，得到預期報酬。
- 股票選擇太多，但時間和金錢資源有限。
- 例：疫情下的波動股市。





研究動機

- 投資需求及財務管理興趣。
- 大眾可以彈性選擇的最佳化股票投資組合程式。
- 應用課堂所學之基因演算法
- 股票投資最佳化的實作。
- 例：長輩的投資問題。



研究目的 / OUR GOALS

更有效率

快速整理台灣股票市場
所有上市上櫃股票
基礎最佳報酬率與價格



課程實作

利用基因演算法每一代
交配與突變的演化
挑出最佳投資組合



簡單優化

使用者無需具備
股市技術分析能力就
可以開始投資



投資

得到的投資組合
相對高報酬低風險
適合長期投資



02

文獻回顧與研究貢獻

literature review and research contribution



Harry Markowitz(1952)

現代投資組合理論 Modern Portfolio Theory

1. **風險 & 報酬**:正相關
2. **分散投資**多樣化的資產可**減少**投資組合承受**風險**

基因演算法應用於投資組合



績效高

Y. Xia , Shouyang.W & Xiaotie.D (2001)
在風險方面, 使用風險共變異數矩陣之**基因演算法**模型的
標準偏差小於效用函數模型所得出之投資組合。

實用性高

卞志祥 (1996)

以台灣加權股價指數為標的物, 建構以**基因演算法**作為分析選股的模型。

高崇勛 (2005)

採用**基因演算法**求個別涵蓋不同實際交易限制的投資組合選股模型。

研究貢獻



建構模型

具科學根據且有效率的客觀選股程式



提升能力

投資者的股票投資組合之決策與獲利



符合需求

協助投資者根據其投資限制提供資訊



精準投資

給予投資最佳的股票投資組合之參考



03

實驗設定與研究方法

Experimental settings and Research methods

實驗設定 / Experiment setting



編碼

排列型編碼



重要運算

比較不同權重



解碼

適應度計算



可行範圍

懲罰項設計

實驗設定 / Experiment setting



編碼

排列型編碼



解碼

適應度計算

```
def initPop():                                # 初始化群體
    return np.random.randint(len(GA_cov), size=(NUM_CHROME, NUM_BIT))

iteration 199: x = [280 917  1 478 840 792 669 906 950 739 412 221  30 811 372 337 561 199
 302 428], y = 2.353294
best return 1.042420, best std 0.011740, best cost 915.102000
```

比較不同權重

懲罰項設計

實驗設定 / Experiment setting



編碼

排列型編碼



重要運算

比較不同權重



解碼

適應度計算



可行範圍

懲罰項設計

實驗設定 / Experiment setting



編碼

排列型編碼



解碼

適應度計算



重要運算

比較不同權重

```
parameter_weight = {"return":3,  
                    "cost":1,  
                    "std":3}
```



可行範圍

懲罰項設計

```
PENALTY_COST = 1000
```

實驗設定 / Experiment setting

輸入資料/INPUT
以讀檔匯入大量股票資訊並
用矩陣整理



輸出資料/OUTPUT
一組最佳解矩陣
此投資組合報酬率
與大盤比較

```
NUM_ITERATION = 200
NUM_CHROME = 100
NUM_BIT = 20
Pc = 0.75
Pm = 0.1
```

課程實作
基因演算法
調整最佳參數



目標
1.報酬率
2.風險
3.目標價格的差距

研究方法 / research method

01

匯入資料

格式化期望研究時間
範圍的所有股票

來源：TEJ資料庫



02

演算法

適應度函數



03

目標檢核

視覺化所得結果
與大盤比較

目標：可調整



研究方法 / research method

01

匯入資料



原始資料

	證券代碼	簡稱	年月日	收盤價(元)	報酬率%
0	0050	元大台灣50	20170103	60.9271	0.1393
1	0051	元大中型100	20170103	22.6522	-0.1171
2	0052	富邦科技	20170103	37.7551	0
3	0053	元大電子	20170103	25.9755	0.5453
4	0054	元大台商50	20170103	18.6354	-0.0944
...
1259654	Y8886	未含金電	20210521	15547.9200	2.199
1259655	Y8887	TR未金電	20210521	30406.3600	2.199
1259656	Y8888	不含金融	20210521	14256.4600	1.6348
1259657	Y9997	報酬指數	20210521	31532.3100	1.6188
1259658	Y9999	加權指數	20210521	16302.0600	1.6188

1259659 rows x 5 columns

資料範圍
20170103
至
20210521

研究方法 / research method

長寬資料轉換

01
匯入資料 

```
data2 = pd.pivot_table(data, values="收盤價(元)", index="年月日", columns="證券代碼")  
data2
```

證券代碼	0050	0051	0052	0053	0054	0055	0056	0057	0061	006203	006204
年月日											
20170103	60.9271	22.6522	37.7551	25.9755	18.6354	12.5787	18.6680	41.99	15.96	28.4171	40.9646
20170104	60.8848	22.6256	38.0653	26.1428	18.6883	12.5787	18.7326	42.22	16.04	28.7531	40.9646
20170105	61.4356	22.8734	38.2336	26.2309	18.8204	12.6483	18.7972	42.22	16.04	28.7531	41.2065
20170106	61.5627	22.9176	38.1982	26.3278	18.8997	12.6657	18.8456	42.64	16.00	28.9082	41.3947
20170109	61.3932	22.7938	38.6413	26.4158	18.9613	12.6483	18.8133	42.58	16.00	28.9255	41.3947
...
20210517	126.5000	46.8500	113.6000	60.0000	28.1100	19.7200	32.7500	84.45	23.92	58.8000	78.1500
20210518	132.3000	49.7200	119.5000	62.9000	29.4000	20.5600	34.0100	89.00	23.93	62.0000	81.8000
20210519	131.4000	50.4500	118.7000	62.2500	29.5700	20.6800	34.1000	89.20	23.60	61.6500	81.5000
20210520	130.5000	49.9500	117.5500	61.6000	29.4900	20.6600	33.8600	88.00	23.80	61.7000	81.3000
20210521	132.3500	51.0000	119.1500	62.5000	29.9200	20.8000	34.0000	89.40	23.73	62.2500	82.8500

1069 rows x 1258 columns

來源：TEJ資料庫

研究方法 / research method

01

匯入資料



時間格式調整

```
data2.index = pd.to_datetime(data2.index, format="%Y%m%d")
data2
```

證券 代碼	0050	0051	0052	0053	0054	0055	0056	0057	0061
年月 日									
2017-01-03	60.9271	22.6522	37.7551	25.9755	18.6354	12.5787	18.6680	41.99	15.96
2017-01-04	60.8848	22.6256	38.0653	26.1428	18.6883	12.5787	18.7326	42.22	16.04
2017-01-05	61.4356	22.8734	38.2336	26.2309	18.8204	12.6483	18.7972	42.22	16.04
2017-01-06	61.5627	22.9176	38.1982	26.3278	18.8997	12.6657	18.8456	42.64	16.00
2017-01-09	61.3932	22.7938	38.6413	26.4158	18.9613	12.6483	18.8133	42.58	16.00
...
2021-05-17	126.5000	46.8500	113.6000	60.0000	28.1100	19.7200	32.7500	84.45	23.92
2021-05-18	132.3000	49.7200	119.5000	62.9000	29.4000	20.5600	34.0100	89.00	23.93

基因演算法範圍

20180101

至

20200101

研究方法 / research method

02

演算法



適應度函數

```
def fitFunc(x):                                # 適應度函數
    total_var = 0
    total_return = 0
    total_cost = initial_cost.iloc[x].sum()
    total_weight = initial_cost/total_cost
    for i in x:
        total_var += total_weight.iloc[i]**2 * GA_cov.iloc[i,i]
        total_return += total_weight.iloc[i] * final_return.iloc[i]
    for i in itertools.combinations(x,2):
        total_var += 2*total_weight.iloc[i[0]]*total_weight.iloc[i[1]]*GA_cov.iloc[i[0],i[1]]
```

研究方法 / research method

02

演算法



適應度函數

```
for i in x:
    total_var += total_weight.iloc[i]**2 * GA_cov.iloc[i,i]
    total_return += total_weight.iloc[i] * final_return.iloc[i]
for i in itertools.combinations(x,2):
    total_var += 2*total_weight.iloc[i[0]]*total_weight.iloc[i[1]]*GA_cov.iloc[i[0],i[1]]
```

證券代碼	0050	0051	0052	0053	0054	0055
0051	0.000059	0.000112	0.000065	0.000067	0.000068	0.000036
0052	0.000094	0.000065	0.000183	0.000108	0.000074	0.000040
0053	0.000087	0.000067	0.000108	0.000117	0.000081	0.000043
0054	0.000067	0.000068	0.000074	0.000081	0.000104	0.000042
0055	0.000043	0.000036	0.000040	0.000043	0.000042	0.000043

列舉所有組合並計算

研究方法 / research method

02 演算法

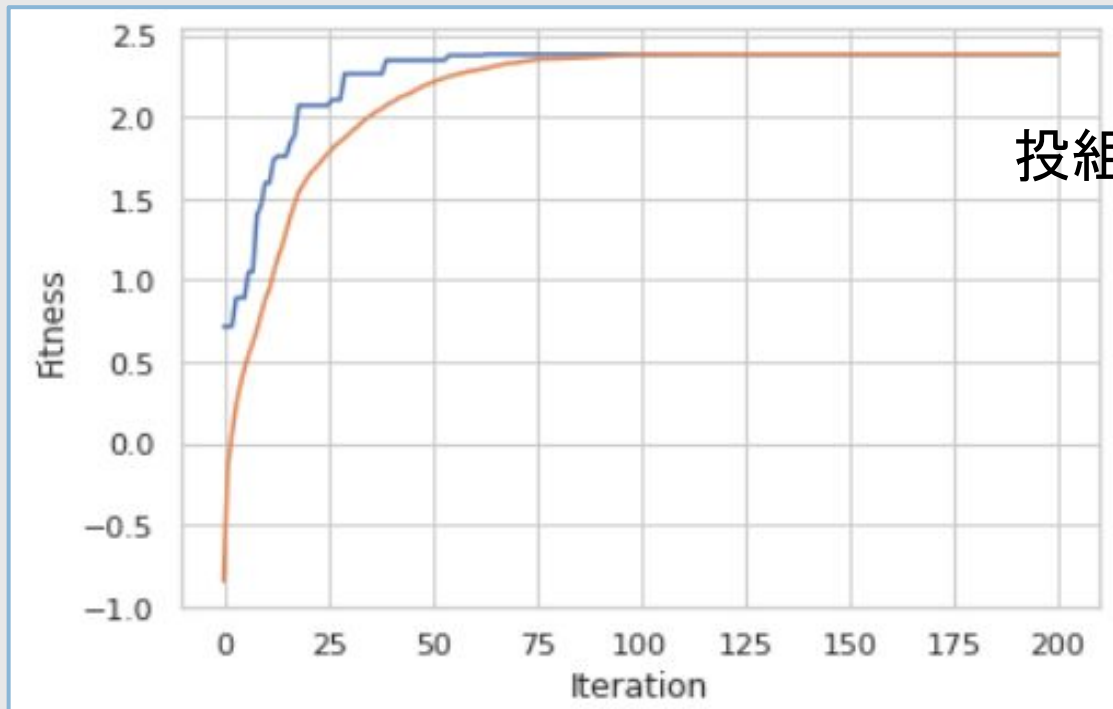


適應度函數(續)


```
budget_deduct = 0
if total_cost > 1000:
    budget_deduct = 2 * (total_cost - 1000) / 1000
elif total_cost < 1000:
    budget_deduct = (1000 - total_cost) / 1000
std_deduct = (np.sqrt(total_var) - min_std) / (max_std - min_std)
count_duplicates = NUM_BIT - len(np.unique(x))
return total_return * parameter_weight["return"]
       - std_deduct * parameter_weight["std"]
       - budget_deduct * parameter_weight["cost"]
       - 0.1 * count_duplicates
```

研究方法 / research method

適應度迭代結果



03

目標檢核 

投組效果檢驗範圍

20200101

至


20210521

研究方法 / research method

適應度迭代結果

20200101

03

目標檢核 



投組效果檢驗範圍

20200101

至

20210521

04

實驗結果與實驗結論

Experimental results and experimental conclusions

實驗結果

題目設定：100萬→20張股票



20%

1. 報酬

期望報酬率
別人恐懼我貪婪



60%

2. 成本

買這20檔股票的金額
與100萬差距越小
權重越高



80%

3. 風險

報酬率的標準差
風險趨避或愛好



實驗結果

/ Experiment Result





結論

- 實驗結果符合預期
- 基因演算法跑投資組合是有潛力的
- 若要實際應用，還須改善

參考文獻 / references

- Harry Markowitz, 1952. "Portfolio Selection", The Journal of Finance, Vol. 7, No. 1., pp. 77-91, March.
- Kallberg J.G., Ziemba W.T. ,1984. "Mis-Specifications in Portfolio Selection Problems", In: Bamberg G., Spremann K. (eds) Risk and Capital. Lecture Notes in Economics and Mathematical Systems, vol 227.
- Xia, Yusen & Wang, Shouyang & Deng, Xiaotie, 2001. "A compromise solution to mutual funds portfolio selection with transaction costs", European Journal of Operational Research, Elsevier, vol. 134(3), pp. 564-581, November.
- 卞志祥(1996), 台灣加權股價指數投資組合之基因演算法建構模型, 國立交通大學, 碩士論文。
- 高崇勛(2005), 使用基因演算法於考慮交易限制之投資組合最佳化模型, 國立高雄應用科技大學, 碩士論文。





Q&A THANKYOU !

謝謝老師及同學的聆聽 祝福大家一切平安順利

傳統均異模型(詳細推導過程)

當 $\sum_{i=1}^n w_i = 1$

$$u_p = E[R_p] = E\left[\sum_{i=1}^n w_i R_i\right] = \sum_{i=1}^n w_i E[R_i] = \sum_{i=1}^n w_i u_i$$

$$\sigma_p^2 = E[(R_p - u_p)^2] = E\left[\sum_{i=1}^n w_i (R_i - u_i)^2\right]$$

$$= E\left[\sum_{i=1}^n \sum_{j=1}^n w_i w_j (R_i - u_i)(R_j - u_j)\right]$$

$$= \sum_{i=1}^n \sum_{j=1}^n E[w_i w_j (R_i - u_i)(R_j - u_j)]$$

$$= \sum_{i=1}^n \sum_{j=1}^n w_i w_j \text{Cov}(R_i, R_j)$$

$$= \sum_{i=1}^n \sum_{j=1}^n w_i w_j \sigma_{ij} = w'[\sigma_{ij}]w$$

傳統均異模型 Mean-Variance Model

- 當 $\sum_{i=1}^n w_i = 1$,
- 平均值 $u_p = \sum_{i=1}^n w_i u_i$
 - 變異數 $\sigma_p^2 = w'[\sigma_{ij}]w$

$$\sigma_p = \sqrt{w'[\sigma_{ij}]w}$$

$$[\sigma_{ij}] = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1n} \\ \vdots & \ddots & \vdots \\ \sigma_{n1} & \cdots & \sigma_{nn} \end{bmatrix}$$

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$



傳統均異模型 Mean-Variance Model



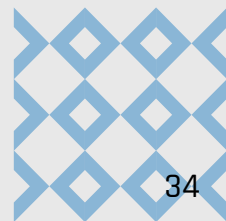
ρ_{ij} ：投資標的 i 及投資標的 j 的報酬率之相關係數

相關係數與共變異數關係： $\sigma_{ij} = \overset{=0}{\rho_{ij}} \sigma_i \sigma_j$

投資標的獨立不相關

共變異數矩陣 ➡ 對角矩陣 $[\sigma_{ij}] = \begin{bmatrix} \sigma_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{nn} \end{bmatrix}$

投資組合的報酬率變異數 ➡ $\sigma_p^2 = \sum_{i=1}^n w_i^2 \sigma_i^2$



傳統均異模型 Mean-Variance Model

考慮各投資標的皆為等權重 $\therefore w_i = \frac{1}{n}$

$$\sigma_p^2 = \sum_{i=1}^n w_i^2 \sigma_i^2 = \sum_{i=1}^n \frac{1}{n^2} \sigma_i^2 = \frac{1}{n} \sum_{i=1}^n \frac{1}{n} \sigma_i^2$$

$$n \rightarrow \infty \Rightarrow \sigma_p^2 \rightarrow 0$$

當投資組合的投資標的越多樣化、報酬相關性越低

➡ 投資組合的風險越小

