

此次實作以GA+SSE的方式來解決花的種類問題

正解的SSE計算出的結果為89.2974

可見SSE與正解的關係不為完全正相關。（SSE過大的情形可能導致求出解的正確率下降）

此次實作發現與之前01問題最大的不同是，01問題進行Evaluation時只需考慮1是否更多，問題較為簡單

而這次需考慮到各個種類、各項資料的SSE總和，問題較為複雜

一開始實作在crossover時是以單一點交換的方式來進行(與之前解01問題時一樣的方式)，來解決此次問題，發現在很前面的Evaluation下就會收斂，收斂約在500-600之間。後來思考後發現若是把diversity設高，也就是更為隨機的方式進行crossover（此次實作利用兩點交換方式再加上4點隨機點交換的方式進行crossover），效果顯著。

————— 以下為PR分析

利用PR將重複多次的點鎖住不進行計算，此次實作設定可以自訂evaluation，在重複次數達到自訂的evaluation下則將該點鎖住。

接著利用Evaluation的總數不變（POP和Iteration的數量也不改變）但PR的自訂evaluation改變來探討PR在不同的evaluation設定下對時間的影響，詳見下表

Evaluation的總數為POP和iteration相乘

實驗表格以POP為600，iteration為200，Run為30的情況下進行實驗及討論
Data Set 為IRIS

表一、不同PR Evaluation的實驗數據表

PR_evaluation	PR_Lock_point Quantity	AVG_SSE	Best_SSE	AVG_Accuracy	Time(s)	SSE_Time(s)
120	121	160	113	73.04%	126.403	78.112
600	122	169	131	62.311	124.464	76.94
1200	121	162	100	71.24%	127.202	78.957
3000	121	142	89	72.78%	129.69	81.35
6000	121	111	86	76.36%	136.66	88.36
12000	109	87	79	83.8%	169.568	120.224
-	0	80	78	83.47%	206.39	156.86

註解：-為沒有啟動PR機制（詳細作法：設定的Evaluation超過pop*iteration，因此程式架構皆完全相同)

表二計算公式

下表的time(%)以上表無啟動 P R 機制的總時間來做100%，時間越少所佔百分比越小
(PR_evaluation-AVG_Time/PR_AVG-time)

下表的SSE(%)為運用PR後造成 S S E 變差的比率(PR_evaluation-AVG_SSE)/未啟用PR_AVG_SSE

以 S S E 來評斷精準性原因為，SSE和花種的準確率並無完全正相關，而程式是以SSE的方式來進行評斷的，因此採用SSE為精準性，較能判斷程式的實際情形。

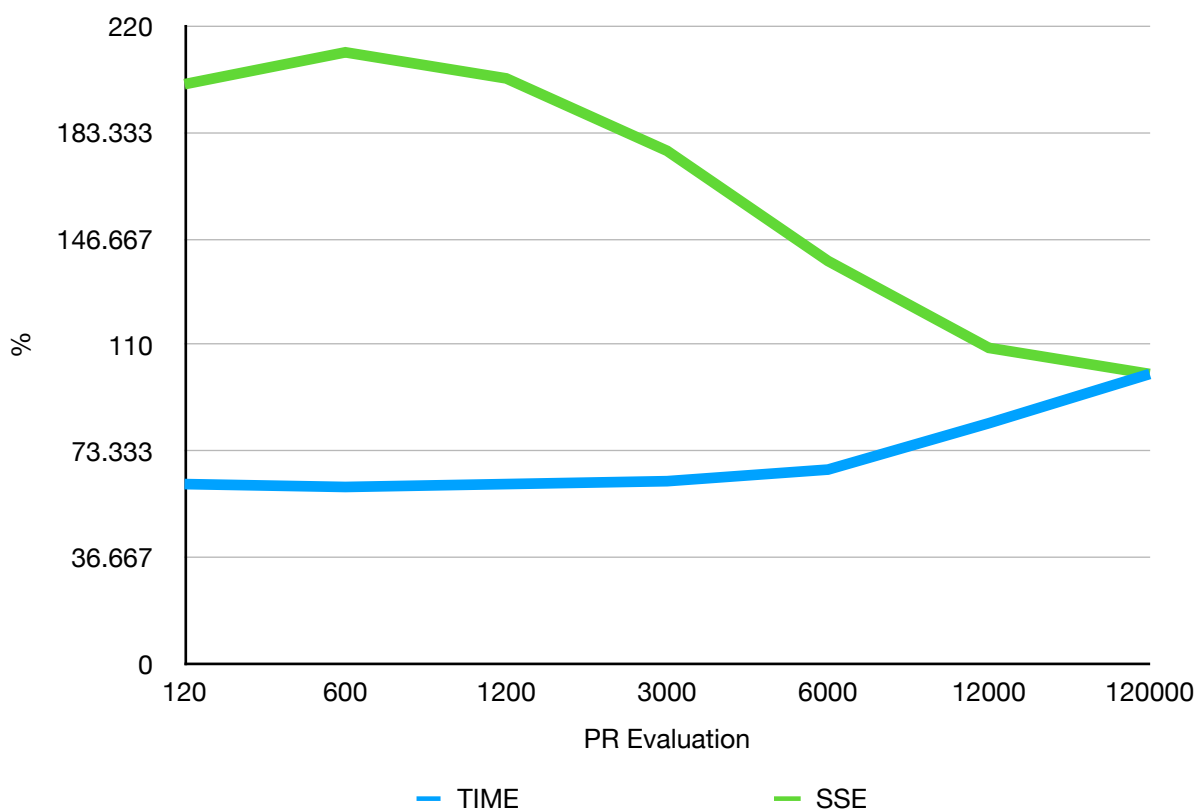
紅色底色的PR為評估過時間和 S S E 準確度後，較佳的採用方式

若以總Evaluation(120000)和PR_Evaluation(6000)的比值來做判定，以evaluation的 5%時來啟動 P R 機制的效果最好(有效縮短時間，結果也不錯)

表二、無PR機制和不同PR機制下的時間及SSE差值百分比表

PR_evaluation	120	600	1200	3000	6000	12000
Time(%)	62%	61%	62%	63%	67%	83 %
SSE (%)	200%	211%	202%	177%	139%	109%

無PR機制和不同PR機制下時間及SSE差值百分比關係圖()



結論

由上圖可以看出越早啟動PR機制，SSE的值會越高，時間則會越少。

但在120、600、1200、3000時的時間差異並不大，但SSE卻有很大的差異，以上圖看的話若要兼顧精確度和時間，3000及6000會是較好的選擇，以evaluation的比例來看大約為2.5%-5%(3000/120000-6000/120000)