作業四

404410039 李維哲

一、參數設定

testTime=每種實驗都作30次

nqueen=n-queen 此次使用8與50

in GA:

population=單次存在的人口數=20人

generation=演化次數=500代

mutationRate=突變率=0.005

二、實作流程

class queen內容:

class queen

{

public:

//constructor

queen(void); //default

queen(int boardSize); //有給Size

queen(int\* knownArray); //使用已知棋盤產生

queen(const queen& copyFrom); //copy constructor

//accessor

void setRow(int row,int col); //更改第row行的值為col

void setBoard(int\* newBoard); //整張換成新棋盤

int getCol(int row); //拿到該row的col值

int\* getBoard(void); //拿到整張圖的陣列

int getAttack(void); //拿到目前的attack數

//assignment(shallow copy)

queen& operator =(const queen& rtSide); //shallow copy

void printBoard(void); //印出目前棋盤(以int[]表示)

void printBoard\_map(void); //印出目前棋盤(以圖形表示)

int attack(); //取得Attack數

int evolution\_HC(void); //對單個queen作HC演化一次

private:

int\* board; //儲存起始位置

int boardSize; //儲存size

int attackNum; //儲存目前attack數

};

hill climbing:

隨機產生一棋盤

對棋盤作HC演化 直到搜到local min attack

HC演化:

neighbor定義:

一次只動一row 比如0123->1123.2123.3123 0023.0223.0323...

計算各neighbor的Attack數 找到最小值

(所以需掃nqueen\*nqueen-nqueen(扣掉自己)次)

若過程沒有其他更好Attack數 跳出(已為最佳解)

若無 更新此圖為最佳neighbor(作法是更改該row值為最佳col)

genetic algorithm

參數設定:

單次存在的人口數population

演化次數generation

突變率mutationRate

丟入GA演化函式

挑選兩親代配對 重複挑population次

挑選方法:tournament法

隨機挑隨機個取最好的attack值

回傳該狀態

作crossover

crossover:

隨機挑切點

i<=切點 子代基因=1號親代基因

i>切點 子代基因=2號親代基因

產出population個子代後

子代作mutation

mutation:

基因中的某點會隨突變率而有機會改變其值

突變基本單位為單個基因點

子代直接取代整個親代 成為新的親代

(使用Generational model(GGA))

以上函式內的步驟重複執行generation次

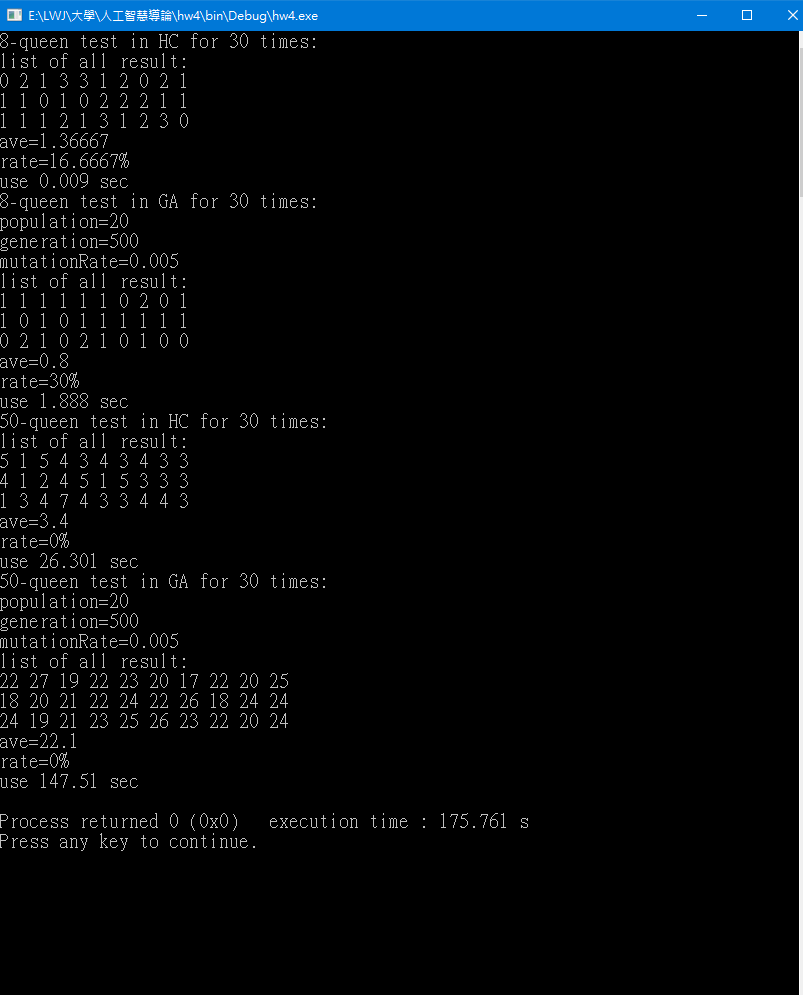
若有出現attack數==0的 跳出

產出最終結果

(結果為最終世代 還是有population個人)

從最終世代中挑attack數最小的 此圖為最終解

三、實驗結果:



8-queen HC:

average=1.366

success rate=16.66%

total running time=0.09sec

8-queen GA:

average=0.8

success rate=30%

total running time=1.88sec

50-queen HC:

average=3.4

success rate=0%

total running time=26.3sec

50-queen GA:

average=22.1

success rate=0%

total running time=147.51sec