

Môn: Nhập môn Trí tuệ Nhân tạo

- Bài tập thực hành tuần: 6
- Sinh viên: Huỳnh Thị Bảo Trân
- MSSV: 19110482

Câu hỏi:

Viết chương trình cài đặt thuật toán xếp tám quân hậu trên bàn cờ kích thước 8×8 sao cho không có quân hậu nào có thể ăn những quân hậu còn lại, nghĩa là không có hai quân nào đứng trên cùng hàng, hoặc cùng cột hoặc cùng đường chéo.

Bài làm:

In []:

```
import sys
import random

Gene = 4
Mutation = 4
Range = 0
Vec = [(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 1), (1, -1), (1, 0), (1, 1)]
GaIdx = [4, 6]

def board(gene):
    for i in range(0, len(gene)):
        line = []
        for j in range(0, len(gene)):
            if j == gene[i]:
                line.append(1)
            else:
                line.append(0)
        print(line)

def make():
```

```

ini = []
for cnt in range(0, Gene):
    line = []
    for i in range(0, 8):
        val = random.randint(0, 7)
        line.append(val)
    ini.append(line)
return ini

def getCell(board, pos, ofst):
    posX = pos[0] + ofst[0]
    posY = pos[1] + ofst[1]
    if posX >= 0 and posY >= 0 and posX < 8 and posY < 8:
        val = board[posX][posY]
    else:
        val = -1
    return val

def calc(gene):
    fitness = 0
    board = []
    line = []
    for i in range(0, 8):
        line = []
        for j in range(0, 8):
            if j == gene[i]:
                line.append(1)
            else:
                line.append(0)
        board.append(line)

    for i in range(0, 8):
        for j in range(0, 8):
            val = getCell(board, (i, j), (0, 0))
            if val == 1:
                for vec in Vec:
                    for k in range(1, 8):
                        valof = getCell(board, (i, j), (vec[0]*k, vec[1]*k))
                        if valof == 1:
                            fitness += 1
                        elif valof == -1:

```

```

                                break
    return fitness

def simple(listg, rankl):
    newg = []
    for i in range(0, Gene):
        if i == rankl[3]:
            newg.append(listg[rankl[0]])
        else:
            newg.append(listg[i])

    updateg = []
    line1 = []
    line2 = []
    for i in range(0, 8):
        if i < GaIdx[0]:
            line1.append(newg[rankl[0]][i])
            line2.append(newg[rankl[1]][i])
        else:
            line1.append(newg[rankl[1]][i])
            line2.append(newg[rankl[0]][i])
    updateg.append(line1)
    updateg.append(line2)

    line1 = []
    line2 = []
    for i in range(0, 8):
        if i < GaIdx[1]:
            line1.append(newg[rankl[2]][i])
            line2.append(newg[rankl[3]][i])
        else:
            line1.append(newg[rankl[3]][i])
            line2.append(newg[rankl[2]][i])
    updateg.append(line1)
    updateg.append(line2)
    return updateg

def main(argv):
    glist = make()

```

```

fitness = []
for i in range(0, Gene):
    fitness.append(0)

loop = 0
while True:
    loop += 1
    idx = 0
    maxidx = []
    minidx = []

    if loop % Mutation == 0:
        geneidx = random.randint(0, Gene-1)
        posidx = random.randint(0, 7)
        valrand = random.randint(0, 7)
        glist[geneidx][posidx] = valrand

    for gene in glist:
        fitness[idx] = calc(gene)
        if idx == 0:
            maxidx = (fitness[0], 0)
            minidx = (fitness[0], 0)

        if maxidx[0] < fitness[idx]:
            maxidx = (fitness[idx], idx)

        if minidx[0] > fitness[idx]:
            minidx = (fitness[idx], idx)

        idx += 1
    minfit = min(fitness)
    if minfit <= Range:
        board(glist[minidx[1]])
        break

    ranktemp = []
    for i in range(0, Gene):
        if i != maxidx[1] and i != minidx[1]:
            ranktemp.append(i)

    if fitness[ranktemp[0]] > fitness[ranktemp[1]]:

```

```
        rankl = [minidx[1], ranktemp[1], ranktemp[0], maxidx[1]]
    else:
        rankl = [minidx[1], ranktemp[0], ranktemp[1], maxidx[1]]

    updatelist = []
    updatelist = simple(glist, rankl)
    glist = updatelist
```

```
main(sys.argv)
```

```
[0, 0, 0, 0, 1, 0, 0, 0]
[1, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 1]
[0, 0, 0, 1, 0, 0, 0, 0]
[0, 1, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 1, 0]
[0, 0, 1, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 1, 0, 0]
```