QueensProblem_MONKOUN

April 28, 2022

1 The 8-QueenProblem

Pour effectuer ce travail, le code créé précédemment a été readapté pour les problèmes quelconques où les X se sont pas forcement binaires dans GeneticAlgoGeneral_.py

[ ]:
```
#import GeneticAlgo.py as Genetic

import sys
sys.path.append("/GeneticAlgoGeneral_")
import GeneticAlgoGeneral_ as Genetic
```

1.1 Représentation du X

X sera une liste comme X=[1,3,4,2,6,7,4,0] Pour chaque élement de la liste, l'index est la position sur la première dimension et la valeur la position sur la deuxième dimension

1.2 La fonction objectif

[ ]:
```
def fitness_score(X):
    score = 0
    #La fonction objetif est positive
    #onc au fur et à mesure qu'on évolue dans les boucles, elles augmentent
    #contiinue arrete la boucle et pénalise le score, en l'empechant d'augmenter

    for ligne in range(len(X)):
        colonne = X[ligne]

        for autre_ligne in range(len(X)):
            #autre=colonne = X[autre_ligne]

            #On va pénaliser le fait qu'il y ait une autre reine sur la ligne
            #Elle gardera alors un score moins important
            if autre_ligne == ligne:
                print("Non respecté ici")
                continue
```

```
            #On va pénaliser le fait qu'il y ait une autre reine sur la colonne
            #Elle gardera alors un score moins important
            if X[autre_ligne] == colonne:
                print("Non respecté ici")
                continue

            #On va pénaliser le fait qu'il y ait une autre reine sur la␣
↪diagonale
            #Elle gardera alors un score moins important
            if autre_ligne + X[autre_ligne] == ligne + colonne:
                print("Non respecté ici")
                continue
            #On va pénaliser le fait d'avoir une autre reine sur la /diagonale
            #Elle gardera alors un score moins important
            if autre_ligne - X[autre_ligne] == ligne - colonne:
                print("Non respecté ici")
                continue


            #On incrémente le score de 1 dans la boucle
            #Quand la reine n'attaque aucune autre jusque là
            score += 1

    #Diviser le score par 2 à cause la comutativité
    return score/2
```

1.3 Test de la fonction objectif

[ ]:
```
X=[1,2,3,4,5,6,7,8]
fitness_score(X)
```

```
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
```

Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici

```
Non respecté ici
```

[ ]: 0.0

[ ]:
```
X=[0,5,1,4,6,3,2,2]
fitness_score(X)
```

```
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
```

[ ]: 26.0

[ ]:
```
X=[0,5,1,4,6,3,7,2]
fitness_score(X)
```

```
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
```

[ ]: 27.0

[ ]:
```
X=[6,1,3,5,7,2,4,6]
fitness_score(X)
```

```
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
```

```
Non respecté ici
Non respecté ici
Non respecté ici
```

[ ]:
```
27.0
```

[ ]:
```
X=[0,1,3,5,7,2,4,6]
fitness_score(X)
```

```
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
Non respecté ici
```

[ ]:
```
27.0
```

Interprétation: En vrai, on doute qu'une solution puisse respecter toutes les conditions mais, on aimerait trouver celle qui offre le meilleur compromis. Et le meilleur score a l'air d'etre 27

1.4 La fonction objectif sans les affichages internes

[ ]:
```
def fitness_score2(X):
    score = 0


    for ligne in range(len(X)):
        colonne = X[ligne]

        for autre_ligne in range(len(X)):

            if autre_ligne == ligne:
                continue

            if X[autre_ligne] == colonne:
                continue

            if autre_ligne + X[autre_ligne] == ligne + colonne:
                continue

            if autre_ligne - X[autre_ligne] == ligne - colonne:
                continue
```

```
            score += 1

    #Diviser le score par 2 à cause la comutativité
    return score/2
```

1.5 Application

[ ]:
```
(a,b,c,d,e)=Genetic.main(pop_size=20, nb_parents=2,nb_generations=10,
      nb_survivants=20, fonction_=fitness_score2,plage_valeurs=7,binaire=False)
```

[ ]:
```
a
```

[ ]:
```
array([[1, 5, 2, 0, 7, 0, 4, 6],
       [6, 3, 1, 7, 7, 0, 2, 5],
       [1, 5, 2, 0, 7, 3, 4, 6],
       [3, 3, 7, 4, 1, 5, 2, 6],
       [3, 5, 5, 1, 6, 4, 2, 7],
       [1, 6, 2, 7, 0, 3, 5, 2],
       [2, 5, 7, 0, 1, 3, 6, 6],
       [2, 5, 1, 6, 0, 3, 7, 5],
       [1, 7, 5, 3, 0, 0, 3, 6],
       [2, 7, 5, 3, 7, 0, 4, 6],
       [3, 7, 4, 1, 0, 5, 5, 2],
       [3, 5, 7, 1, 1, 5, 2, 6],
       [7, 1, 6, 2, 0, 6, 4, 5],
       [2, 7, 1, 3, 0, 0, 4, 4],
       [2, 5, 1, 3, 7, 0, 4, 1],
       [5, 3, 6, 0, 7, 4, 4, 7],
       [7, 5, 3, 1, 7, 5, 2, 6],
       [3, 5, 2, 4, 0, 7, 4, 6],
       [1, 3, 4, 0, 7, 0, 2, 6]])
```

[ ]:
```
b
```

[ ]:
```
array([27., 27., 27., 27., 26., 26., 26., 26., 26., 26., 26., 26.,
       26., 26., 26., 26., 26., 26.])
```

Ici les quatre meilleurs ont une fitness de 27