CSC 180-01 Intelligent Systems (Fall 2019)

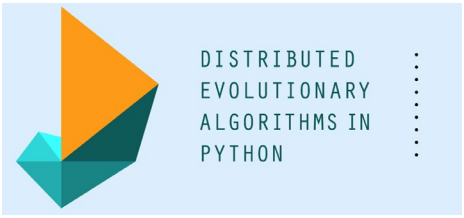
Mini-Project 4: Solving n-queens Problem using Genetic Algorithms

Due at 11 am, Friday, November 15, 2019

Demo Session: class time, Friday, November 15, 2019

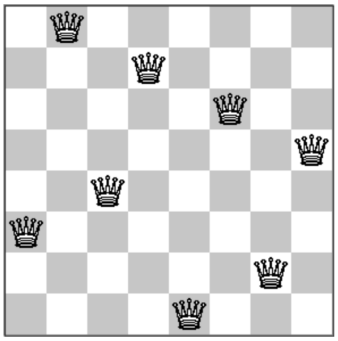
1. Problem Formulation

In this project, you practice with genetic algorithm by using Distributed Evolutionary Algorithms in Python (DEAP), the most popular Python library for evolutionary computation

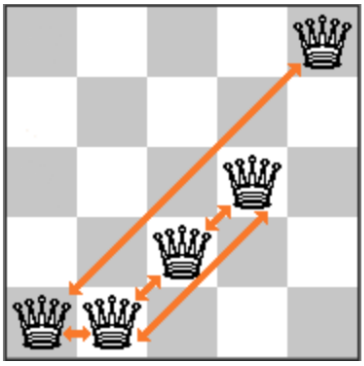


The n-queens problem was first invented in the mid-1800s as a puzzle for people to solve in their spare time, but now serves as a good tool for discussing computer search algorithms. **In chess, a queen is the only piece that can attack in any direction. The puzzle is to place a number of queens on a board in such a way that no queen is attacking any other.**

In this project, we use DEAP to solve the 8 queens puzzle. **The 8 queens puzzle is the problem of placing eight chess queens on an 8 × 8 chessboard so that no two queens attach each other.** The eight queens puzzle is an example of the more general N queens problem of placing n non-attacking queens on an n ×n chessboard, for which solutions exist for all natural numbers with the exception of n = 2 and n = 3.



One way we can describe the board above is to say it has a cost of 0, because there are 0 pairs of queens attacking each other. We can then generalize this to say **the cost of a given n-queens board is equal to the sum total number of distinct pairs of queens that are in the same row, column, or diagonal.** Consider this 5-queens puzzle. There are 5 pairs of queens attacking each other therefore the cost of this board is 5.



2. Major Challenges

When we use DEAP, there are two major challenges:

- How should we describe each board (a given arrangement of eight queens), i.e., what should the **best numeric representation of each board**?
- How to **write a (fitness) function to calculate the cost of any given board**?

3. Board Representation

In this project, let's compare two different board representations.

- **Position-indexed-based**: On an 8 × 8 board, each position will be represented as an intege r from 0 to 63.We use one integer to represent the position of a queen. Each board is a lis t of eight numbers (each number is taken from 0 to 63). For example: [14, 35, 51, 42, 1 2, 47, 62, 2]

```
show_grid([14, 35, 51, 42, 12, 47, 62, 2])
```

```
-|-|X|-|-|-|-|-|
---------------
-|-|-|-|X|-|X|-|
---------------
-|-|-|-|-|-|-|-|
---------------
-|-|-|-|-|-|-|-|
---------------
-|-|-|X|-|-|-|-|
---------------
-|-|X|-|-|-|-|X|
---------------
-|-|-|X|-|-|-|-|
---------------
-|-|-|-|-|-|X|-|
---------------
```

- **Row-indexed-based**: Each row of the board is indexed from 0 to 7. We place different queens on different rows from top to bottom. The sequence [a b c d ....] means that in 0-th row, a-th column, the queen is present and so on. Each board is a list of eight numbers (each number is taken from 0 to 7).

```
show_grid([2, 6, 2, 7, 6, 3, 4, 3])
```

```
-|-|X|-|-|-|-|-|
---------------
-|-|-|-|-|-|X|-|
---------------
-|-|X|-|-|-|-|-|
---------------
-|-|-|-|-|-|-|X|
---------------
-|-|-|-|-|-|X|-|
---------------
-|-|-|X|-|-|-|-|
---------------
-|-|-|-|X|-|-|-|
---------------
-|-|-|X|-|-|-|-|
---------------
```

4. Notebook

Write your code to complete the notebook on Canvas. Compare the two representations and think about which one is better. Print out the fitness of the best solution returned as well as the corresponding board for each representation.

5. Grading breakdown

Use the evaluation form on Canvas as a checklist to make sure your work meet all the requirements.

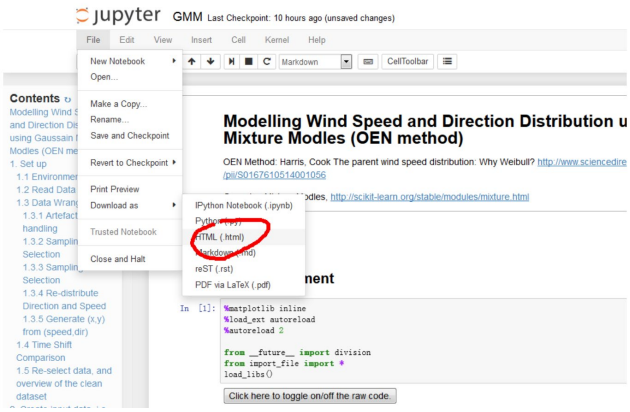
Implementation	**70 pts**
Your report	**20 pts**
In-class demo	**10 pts**

6. Teaming:

Students must work in teams with no more than 3 people. Think clearly about who will do what on the project. Normally people in the same group will receive the same grade. However, the instructor reserve the right to assign different grades to team members depending on their contributions. So you should choose partner carefully!

7. Deliverables:

(1) The HTML version of your notebook that includes all your source code. Go to "File" and then "Download as". Click "HTML" to convert the notebook to HTML.



5 pts will be deducted for the incorrect file format.

(2) **Your report in PDF format,** with your name, your id, course title, assignment id, and due date on the first page. As for length, I would expect a report with more than one page. Your report should include the following sections (but not limited to):

- **Problem Statement**
- **Methodology**
- **Experimental Results and Analysis**
- **Task Division and Project Reflection**

In the section "**Task Division and Project Reflection**", describe the following:

- who is responsible for which part,
- challenges your group encountered and how you solved them
- and what you have learned from the project as a team.

10 pts will be deducted for missing the section of task division and project reflection.

All the files must be submitted **by team leader** on Canvas before

11 am, Friday, November 15, 2019

NO late submissions will be accepted.

8. In-class Demo:

Each team member must demo your work during the scheduled demo session. Each team have **three minutes** to demo your work in class. Failure to show up in defense session will result in **zero** point for the project. The following is how you should allocate your time:

- Model/code design (1 minute)
- Findings/results (1 minute)
- Task division, challenges encountered, and what you learned from the project (1 minutes)

9. Think beyond the Project

- Can you come up with other possible board representations? Test their effectiveness using DEAP
- So far, we focus on 8-queens problem only. Can you find the solution to the general n-queens problem using DEAP for any N value?