CSC 180-01 Intelligent Systems (Fall 2022)

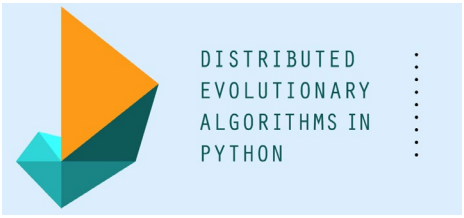
Project 4: Solving N-Queens Problem using Genetic Algorithms

Due at 10:00 am, Friday, November 18, 2022

Peer Review: class time, Friday, November 18, 2022

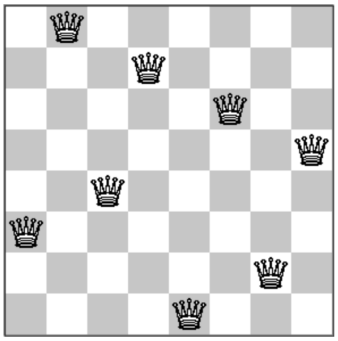
1. Problem Formulation

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

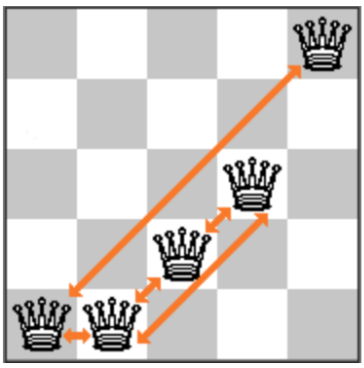


The n-queens problem was first invented in the mid-1800s as a puzzle for people to solve in their spare time, but now serves as a good tool for discussing computer search algorithms. **In chess, a queen is the only piece that can attack in any direction. The puzzle is to place a number of queens on a 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 queens on an 8 × 8 chessboard so that no two queens attack 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 encode each board (a given arrangement of eight queens) using numbers, 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. Chessboard Representation

In this project, let us compare two different board representations.

- **Position-indexed-based**: On an 8 × 8 board, each position will be represented as an integer from 0 to 63.We use one integer to represent the position of a queen. Each board is a list of e ight numbers (each number is taken from 0 to 63). For example: [14, 35, 51, 42, 12, 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 provided notebook on Canvas. For each representation, show the chessboard with eight queens you end up with having fewest conflicts as well as its fitness value.
Note: you may try different GA parameters and run the code a few times to achieve the perfect chessboards without any conflicts.

Compare the two chessboard representations and describe which one is better, e.g., in terms of ease of coding or final solution quality.

Try different mutation and crossover operations and vary the number of generations and the population size to see the changes. Write your findings in the report.

5. Grading breakdown

You may feel this project is described with some certain degree of vagueness, which is left on purpose. In other words, **creativity is strongly encouraged**. Your grade for this project will be based on the soundness of your design, the novelty of your work, and the effort you put into the project.

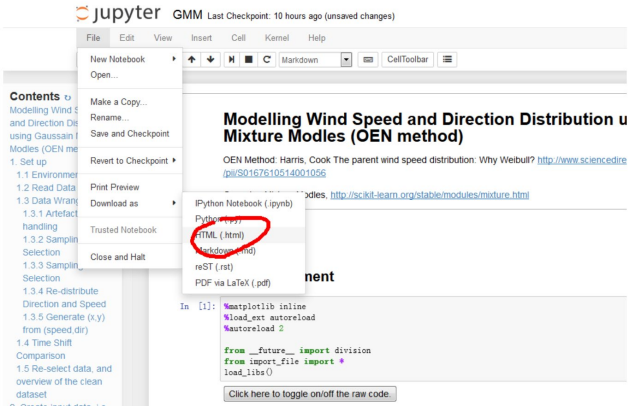
Use the evaluation form on Canvas as a checklist to make sure your work meet all the requirements.

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.

To submit your notebook and the report, go to Canvas "Assignments" and use "Project X (submit your code and report here)". Use the evaluation form on Canvas as a checklist to make sure your work meet all the requirements.

(3) Link to your video presentation shared to the discussion board. Each team have **three minutes** to demo your work. Failure to submit the video presentation 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)

To submit the link to your video presentation, go to Canvas "Discussions" and use "Post Your Presentation for Project X Here". Share your link by replying directly to my main discussion post.

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

10:00 am, Friday, November 18, 2022

NO late submissions will be accepted.

8. Think beyond the Project

- Can you come up with other possible chessboard representations? Compare 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?
- Can you try some other games?

9. Peer Review:

During the class after the deadline, please review and comment on the presentations from other teams by replying to their posts. It is a great chance for you to learn from other people's work. Please be nice, and provide constructive, specific feedbacks. You will become a better, more effective learner when you found yourself in a community of active learners!