Eight Queens Puzzle: An Integer Linear Programming Approach

Mohamad Fares El Hajj Chehade

January 2023

1 Introduction

The Eight Queens Puzzle is a famous problem where 8 queen pieces are to be placed on an 8×8 chess board, and no queen can threaten another. The solution consists of the positions of these eight pieces on the chess board. Such a solution has been proven not to be unique. While finding one of these solutions can be done by conducting an eye test or through brute force, such methods become very cumbersome and time-consuming when trying to find multiple solutions.

In this work, we propose an automated method for finding all solutions to the Eight Queens Puzzle (or more generally, n queens on an $n \times n$ board) using an integer linear programming (ILP) approach. Although there is no objective function for the general problem, an ILP is powerful in finding feasible solutions to the problem, which in this case ensures that no queen threatens another queen. Hence, the problem is formulated with a dummy objective function.

First, an ILP is used to solve for one solution. After that, an iterative method is applied, where during each iteration, a new constraint is added that removes the previous solution from the feasible region. The iterative process continues until no more feasible solution exists, and the entire solution set has been covered.

You can find more information on the general Eight Queens Puzzle problem here.

2 Problem Formulation

In this section, the problem is formulated as an ILP optimization problem with a dummy objective function. The solution to this problem constitutes one of the many alternate solutions to the n Queens Puzzle. The Eight Queens Puzzle is obtained when $n = 8$.

- Let $x_{ij} = \{0,1\}$: A binary variable indicating whether there exists a queen on row i and column j where $i \in \{1, 2, ..., n\}$ and $j \in \{1, 2, ..., n\}$.

$$minimize \quad 1 \tag{1}$$

subject to:

- At most one queen on the same row and column (in reality, there should be exactly one queen on every row and column since the number of queens is equal to that of rows and of columns):

$$\forall i \quad \sum_{j=1}^{n} x_{ij} \leq 1 \tag{2}$$

$$\forall j \quad \sum_{i=1}^{n} x_{ij} \leq 1 \tag{3}$$

- At most one queen on the main diagonal:

$$\forall i \ \forall j \quad \sum_{k=max\{1-i,1-j\}}^{min\{n-i,n-j\}} x_{i+k,j+k} \leq 1 \tag{4}$$

The bounds for the summation are found by solving for the inequalities:

$$1 \leq i+k \leq n \tag{5}$$

$$1 \leq j+k \leq n \tag{6}$$

- At most one queen on the anti-diagonal:

$$\forall i \ \forall j \quad \sum_{k=max\{i-n,1-j\}}^{min\{i-1,n-j\}} x_{i-k,j+k} \leq 1 \tag{7}$$

The bounds for the summation are found by solving for the inequalities:

$$1 \leq i-k \leq n \tag{8}$$

$$1 \leq j+k \leq n \tag{9}$$

- Trivial enough, the number of queens placed on the board is n:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} = n \tag{10}$$

3 Solution Set Enumeration

The previous section finds one solution. In this section, an iterative procedure is suggested to capture all solutions, where in each iteration, the problem is solved again with the addition of a constraint that removes the previous solution from the feasible set.

If we let $\mathbf{x}$ be the solution of the previous iteration of size n, the added constraint at each iteration can be:

$$\sum_{r=1}^{n} x_r \leq n-1 \tag{11}$$

The iterative process can be summarized in the following pseudo-code:

Algorithm 1 Suggested Iterative Procedure

```
flag = True
while flag do
    Add the constraint in (11)
    x ← solve the new ILP
    if x ∈ S then                    ▷ S denotes the feasible set
        record x
    else
        flag = False
    end if
end while
```