INDR 220: Introduction to Computing for Operations Research
Homework 6: The Non-attacking Knights Problem
Deadline: January 16, 2023, 11:59 PM

In this homework, you will implement a Python script that solves the non-attacking knights problem using CPLEX. The task is to find the maximum number of knights that can be placed on an $M \times N$ chessboard such that no knight attack some other knight. The decision variables can be formulated as

$$x_{ij} = \begin{cases} 1 & \text{if a knight is placed at location } (i,j), \\ 0 & \text{otherwise.} \end{cases}$$

The integer linear programming formulation of this problem becomes

$$\begin{aligned} \text{maximize} \quad & z = \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij} \\ \text{subject to:} \quad & x_{ij} + x_{kl} \leq 1 \quad \forall (i,j,k,l) \text{ where } (i,j) \text{ and } (k,l) \text{ are attacking locations} \\ & x_{ij} \in \{0,1\} \quad i = 1,2,\ldots,M; \;\; j = 1,2,\ldots,N. \end{aligned}$$

An example of the non-attacking knights problem with a 2×6 chessboard can be given as

$$\begin{aligned} \text{maximize} \quad & z = x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} \\ \text{subject to:} \quad & x_{11} + x_{23} \leq 1 \\ & x_{12} + x_{24} \leq 1 \\ & x_{13} + x_{25} \leq 1 \\ & x_{13} + x_{21} \leq 1 \\ & x_{14} + x_{26} \leq 1 \\ & x_{14} + x_{22} \leq 1 \\ & x_{15} + x_{23} \leq 1 \\ & x_{16} + x_{24} \leq 1 \\ & x_{11} \in \{0,1\} \quad x_{12} \in \{0,1\} \quad x_{13} \in \{0,1\} \quad x_{14} \in \{0,1\} \quad x_{15} \in \{0,1\} \quad x_{16} \in \{0,1\} \\ & x_{21} \in \{0,1\} \quad x_{22} \in \{0,1\} \quad x_{23} \in \{0,1\} \quad x_{24} \in \{0,1\} \quad x_{25} \in \{0,1\} \quad x_{26} \in \{0,1\}. \end{aligned}$$

An optimum solution of the example problem with a 2×6 chessboard is as follows:

$$\begin{array}{llllll} x_{11}^{\star} = 1 & x_{12}^{\star} = 1 & x_{13}^{\star} = 0 & x_{14}^{\star} = 0 & x_{15}^{\star} = 1 & x_{16}^{\star} = 1 \\ x_{21}^{\star} = 1 & x_{22}^{\star} = 1 & x_{23}^{\star} = 0 & x_{24}^{\star} = 0 & x_{25}^{\star} = 1 & x_{26}^{\star} = 1 \end{array}$$

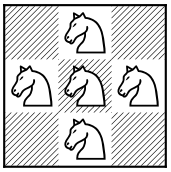
An example of the non-attacking knights problem with a 3×3 chessboard can be given as

$$\begin{aligned} \text{maximize} \quad & z = x_{11} + x_{12} + x_{13} + x_{21} + x_{22} + x_{23} + x_{31} + x_{32} + x_{33} \\ \text{subject to:} \quad & x_{11} + x_{32} \leq 1 \end{aligned}$$

$$
\begin{aligned}
& x_{11} + x_{23} \leq 1 \\
& x_{12} + x_{33} \leq 1 \\
& x_{12} + x_{31} \leq 1 \\
& x_{13} + x_{32} \leq 1 \\
& x_{13} + x_{21} \leq 1 \\
& x_{21} + x_{33} \leq 1 \\
& x_{23} + x_{31} \leq 1 \\
& x_{11} \in \{0,1\} \quad x_{12} \in \{0,1\} \quad x_{13} \in \{0,1\} \\
& x_{21} \in \{0,1\} \quad x_{22} \in \{0,1\} \quad x_{23} \in \{0,1\} \\
& x_{31} \in \{0,1\} \quad x_{32} \in \{0,1\} \quad x_{33} \in \{0,1\}.
\end{aligned}
$$

An optimum solution of the example problem with a 3×3 chessboard is as follows:

$$
\begin{aligned}
& x^\star_{11} = 0 \quad & & x^\star_{12} = 1 \quad & & x^\star_{13} = 0 \\
& x^\star_{21} = 1 \quad & & x^\star_{22} = 1 \quad & & x^\star_{23} = 1 \\
& x^\star_{31} = 0 \quad & & x^\star_{32} = 1 \quad & & x^\star_{33} = 0
\end{aligned}
$$



Implement your algorithm to solve the non-attacking knights problem in a single interactive Python notebook using Azure Lab Services. Your notebook should include at least the following function definition that takes the number of rows and columns as parameters and returns the solution found.

```
def nonattacking_knights_problem(M, N):
    #implement your algorithm here
    return(X_star, obj_star)
```

What to submit: You need to submit your source code in a single file (`.py` file that you will download from Azure Lab Services by following "File" / "Download as" / "Python (.py)" menu items) named as `STUDENTID.py`, where `STUDENTID` should be replaced with your 7-digit student number.

How to submit: Submit the file you created to Blackboard. Please follow the exact style mentioned and do not send a file named as `STUDENTID.py`. Submissions that do not follow these guidelines will not be graded.

Late submission policy: Late submissions will not be graded.

Cheating policy: Very similar submissions will not be graded.
