

Defection and Cooperation amongst Prisoners

@NikoletaGlyn





Operational Research

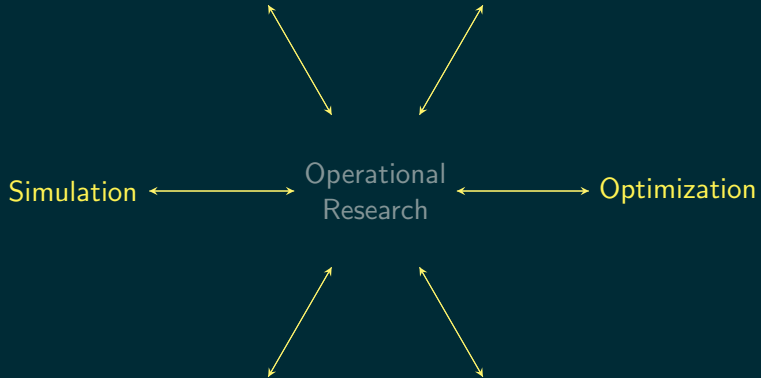
Marketing

Computing

Simulation \longleftrightarrow Operational Research \longleftrightarrow Optimization

Stochastic Modelling

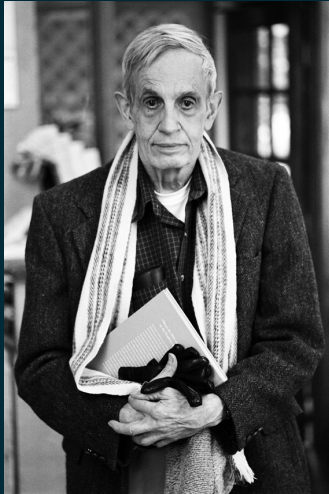
...



Game Theory

Traveller's Dilemma

$$\begin{bmatrix} (2, 2) & (4, 0) & (4, 0) \\ (0, 4) & (3, 3) & (5, 1) \\ (0, 4) & (1, 5) & (4, 4) \end{bmatrix}$$



$$\begin{bmatrix} (2, 2) & (4, 0) & (4, 0) \\ (0, 4) & (3, 3) & (5, 1) \\ (0, 4) & (1, 5) & (4, 4) \end{bmatrix}$$

The Prisoner's Dilemma



$$\begin{bmatrix} (3, 3) & (0, 5) \\ (5, 0) & (1, 1) \end{bmatrix}$$

Robert Axelrod

1980a: 14+1 strategies

1980b: 64+1 strategies

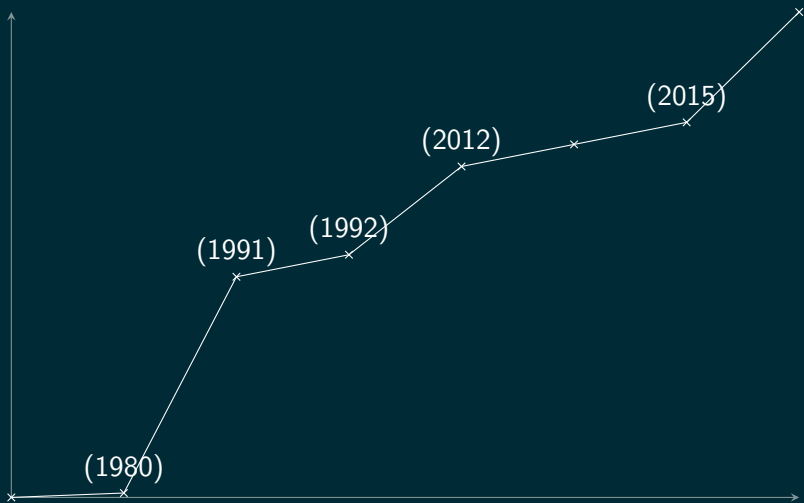
```
class TitForTat(Player):
    """
    A player starts by cooperating and then mimics the previous action of
    the opponent.

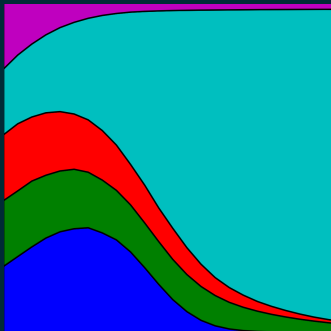
    Note that the code for this strategy is written in a fairly verbose
    way. This is done so that it can serve as an example strategy for
    those who might be new to Python.

    Names:

    - Rapoport's strategy: [Axelrod1980]_
    - TitForTat: [Axelrod1980]_
    """

    def strategy(self, opponent):
        """This is the actual strategy"""
        # First move
        if not self.history:
            return C
        # React to the opponent's last move
        if opponent.history[-1] == D:
            return D
        return C
```



Literature Review
Meta study of tournaments
Machine learning strategies
Evolution

@NikoletaGlyn

<https://github.com/Nikoleta-v3>

<https://github.com/Axelrod-Python/Axelrod>