**LearningAgents branch on GitHub -> EquiLearn\learningAgents\src\optim\_PGM\_base**

## How to run?

Open “Optim,Play” in JupyterNotebook

In cells 4 and 5, low\_cost and high\_cost strategies are defined. These strategies can be static or can be loaded from a neural network (saved in NNs folder).   
for static strategy, the strategy functions have already been defined in “environmentModelBase” class. So they can easily be defined with this syntax:

Strategy(StrategyType.static, NNorFunc=em.NAME\_OF\_FUNCTION, name="NAME\_YOU\_LIKE", firstPrice=FIRST\_PRICE\_IF\_NEEDED)  
  
For loading from NN, the name of the NN file can be passed to “create\_nn\_strategy(name)” function and the strategy object will be returned.

After defining a list of low-cost strategies and a list of high-cost strategies, then you create a BimatrixGame object by setting the strategies:

bimatrix\_game = BimatrixGame.BimatrixGame(low\_strategies, high\_strategies)

and then you can run the tournament by calling the “run\_tournament()” method in the BimatrixGame file and all the equilibria that led to adding a new strategy will be returned at the end.

equilibria=BimatrixGame.run\_tournament(bimatrix\_game=BIMATRIX\_GAME\_OBJECT,number\_rounds= NUMBER\_OF\_ROUNDS)

## Where are the games and strategies stored?

The current game is stored in the base folder as “game.txt” and will be updated after each new strategy is added.

The new trained strategies (neural networks) are saved as in the “NNs” folder.

The details of each trained strategy will also be written in “results.xlsx” file in the base folder.

Please do NOT keep the “game.txt” or “results.xlsx” files open when you are running the project because the new data can not be written on them and will raise an error which stops the project.

## How to change the parameters?

All the main parameters’ values are set in the “globals.py” and can be changed in there.