IPD Refcard

Philippe MATHIEU
philippe.mathieu@univ-lille.fr
Univ. Lille, CNRS, Centrale Lille, UMR 9189 – CRIStAL (SMAC) – FRANCE

Imports

```
from game import *
from ipd import *
from strategies import *
```

Game

Creation

```
ipd_scores = [(3, 3), (0, 5), (5, 0), (1, 1)]
ipd = game.Game(ipd_scores, ["C", "D"])
```

Display

```
ipd.prettyPrint()
```

Nash and Pareto optima

```
ipd.getNash()     # provide index
ipd.getPareto()   # provide index
```

The classical matrix

		Player II: Cooperate	Player II: Defect
Player I	Cooperate	R=3 R=3	T=5 S=0
	Defect	S=0 T=5	P=1 P=1

Meetings

Execution

```
sA = Tft()
sB = Periodic("D","allD")
m = Meeting(g,sA,sB,20) # default 1000
m.run()
```

Score of the first strategy

```
m.s1_score
```

Display rounds

```
m.prettyPrint(20)
```

Tournament

Execution

```
t = Tournament(g, getMem(1,1), length=20)  # default  1000
t.run()
```

Python display limits

```
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
```

Display all the score matrix

```
t.matrix
```

Display only the results

```
t.matrix['Total']
```

Display the 10 best

```
t.matrix['Total'][0:10]
```

Get one of the winners (ex æquo possibles)

```
t.matrix.index[0]
```

Get the best score

```
t.matrix['Total'][0]
```

Ecological competitions

Execution

```
e = Ecological(t, pop=50) # default 100
e.run()
```

Python display limits

```
pd.set_option('display.max_rows', None)
```

History of the evolutions

```
e.historic
```

Display the 3 bests

```
e.historic.iloc[-1,0:3]
```

Get the best score

```
e.historic.iloc[:,0]
```

Display all those still alive

```
e.historic.iloc[-1][e.historic.iloc[-1]>0]
```

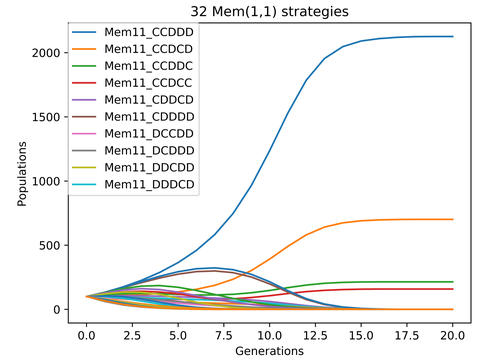
Graphical aspects

```
e.drawPlot()                # on-screen display
e.drawPlot(save='fig.pdf') # save in a file
e1.drawPlot(5,None)         # only 5 lines in legend
```

Genotype Mem(1,2) - plot Mem(1,1)

My two first plays C C

Me-1	She-2	She-1	
C	C	C	C
C	C	D	D
C	D	C	C
C	D	D	D
D	C	C	D
D	C	D	C
D	D	C	D
D	D	D	D



Classical equivalences

```
Mem(0,0,'C','AllC')
Mem(0,0,'D','AllD')
Mem(1,0,'cDC','PerCD')
Mem(1,0,'dDC','PerDC')
Mem(0,1,'cCD','Tft')
Mem(0,1,'dCD','Mistrust')
Mem(1,1,'cCDDD','Spiteful')
Mem(1,1,'cCDDC','Pavlov')
Mem(0,2,'ccCCCD','Tf2t')
Mem(0,2,'ccCDDD','Hard_tft')
Mem(1,2,'ccCCCDCDDD','Slow_tft')
Mem(1,2,'ccCDCDDCDD','Winner12')
Mem(1,2,'ccCDCDDDDD','Spiteful_cc')
```

```
Proba('C',1,1,1,1,'AllC')
Proba('D',0,0,0,0,'AllD')
Proba('C',0,0,1,1,'PerCD')
Proba('D',0,0,1,1,'PerDC')
Proba('C',1,0,1,0,'Tft')
Proba('D',1,0,1,0,'Mistrust')
Proba('C',1,0,0,0,'Spiteful')
Proba('C',1,0,0,1,'Pavlov')
```

Sizes of strategy sets

Name	Size
memory(0,1)	$2^1 * 2^2 = 8$
memory(1,0)	$2^1 * 2^2 = 8$
memory(1,1)	$2^1 * 2^4 = 32$
memory(2,0)	$2^2 * 2^4 = 64$
memory(1,2)	$2^2 * 2^8 = 1024$
memory(2,1)	$2^2 * 2^8 = 1024$
memory(2,2)	$2^2 * 2^{16} = 262144$

Name	Size
proba(K=1)	$2 * 2^4 = 32$
proba(K=2)	$2 * 3^4 = 162$
proba(K=4)	$2 * 5^4 = 1250$
proba(K=5)	$2 * 6^4 = 2592$
proba(K=8)	$2 * 9^4 = 13122$