security

February 21, 2025

[6]:

```
import numpy as np

def ComputeValue(M):
    M = np.array(M)

    ps = np.linspace(0, 1, 101)
    best_value = -np.inf
    best_p = 0
    for p in ps:
        payoff_col1 = p * M[0, 0] + (1 - p) * M[1, 0]
        payoff_col2 = p * M[0, 1] + (1 - p) * M[1, 1]
        security_value = min(payoff_col1, payoff_col2)  #col want to minimize
        if security_value > best_value: #row want to find the maximum
            best_value = security_value
            best_p = p
    P1 = np.array([best_p, 1 - best_p])
    V1 = best_value


    qs = np.linspace(0, 1, 101)
    best_value2 = np.inf
    best_q = 0
    for q in qs:
        payoff_row1 = q * M[0, 0] + (1 - q) * M[0, 1]
        payoff_row2 = q * M[1, 0] + (1 - q) * M[1, 1]
        security_value2 = max(payoff_row1, payoff_row2)  #row want to maximize
        if security_value2 < best_value2: #col want to find the minimum
            best_value2 = security_value2
            best_q = q
    P2 = np.array([best_q, 1 - best_q])
    V2 = best_value2

    return P1, P2, V1, V2
```

1.

[7]:

```
M = np.array([[1, 3],[4, 2]])
```

```
P1, P2, V1, V2 = ComputeValue(M)
print("M:","\n",M)
print("P1:", P1)
print("V1:", V1)
print("P2:", P2)
print("V2:", V2)
```

```
M:
 [[1 3]
 [4 2]]
P1: [0.5 0.5]
V1: 2.5
P2: [0.25 0.75]
V2: 2.5
```

2. M1 solution

[8]:

```
import scipy.io as scio

data = scio.loadmat('security_strategy_files\examples.mat')
# print(data.keys())
M1 = data['M1']
P1, P2, V1, V2 = ComputeValue(M1)
print("M1:", "\n", M1)
print("P1:", P1)
print("V1:", V1)
print("P2:", P2)
print("V2:", V2)
```

```
M1:
 [[1 5]
 [3 2]]
P1: [0.2 0.8]
V1: 2.6
P2: [0.6 0.4]
V2: 2.6
```

M2 solution

[9]:

```
M2 = data['M2']
P1, P2, V1, V2 = ComputeValue(M2)
print("M2:", "\n", M2)
print("P1:", P1)
print("V1:", V1)
print("P2:", P2)
print("V2:", V2)
```

```
M2:
 [[7 8]
 [4 6]]
```

```
P1: [1. 0.]
V1: 7.0
P2: [1. 0.]
V2: 7.0
```

M3 solution

[10]:

```
M3 = data['M3']
P1, P2, V1, V2 = ComputeValue(M3)
print("M3:", "\n", M3)
print("P1:", P1)
print("V1:", V1)
print("P2:", P2)
print("V2:", V2)
```

```
M3:
 [[7 4]
 [2 9]]
P1: [0.7 0.3]
V1: 5.5
P2: [0.5 0.5]
V2: 5.5
```

3. In M2, each player's mixed security strategy is pure security strategy. These strategies are dominant compared with other strategies.