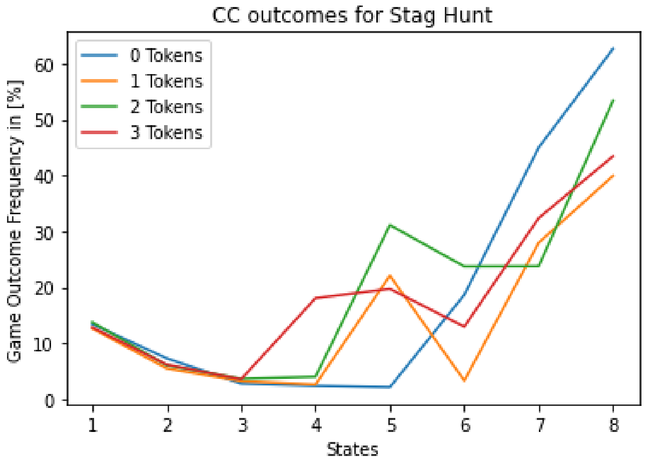
Sheet 2

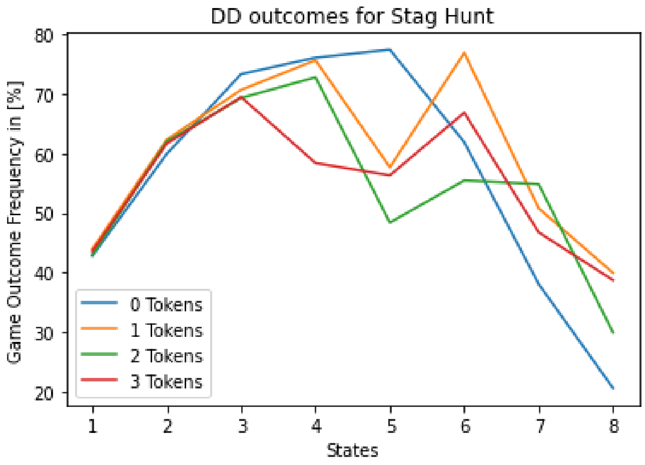
```python
from Model import Model
from GameHistoryAnalysis import GameHistoryAnalysis
import matplotlib.pyplot as plt


# Investigation into token number relative to cooperative epochs for 1, 2, 3, 4 Tokens, averaged over
number_of_agents = 50
generations = 1000
game = 'SH'
learning_mechanism = 'GA'
tokens = 4
timeout = 100
iterations = 5
states = 8

cooperative_epochs_by_states_by_tokens = []
avg_percent_outcomes_by_states_by_tokens = []
for s in range(states):
    print('Starting Computation for computational capacity of ' + str(s))
    cooperative_epochs_by_tokens = []
    avg_percent_outcomes_by_tokens = []
    for t in range(tokens):
        print('Starting Computation for communication tokens: ' + str(t))
        total_coop_epochs = 0
        avg_percent_outcomes = [0, 0, 0]
        for i in range(iterations):
            print('Starting Iteration ' + str(i))
            model = Model(number_of_agents, s+1, t+1, generations, learning_mechanism, timeout, game)
            model.run_model()
            gha = GameHistoryAnalysis(model.game_history, number_of_agents)
            total_coop_epochs += gha.cooperative_epochs

            total = sum(gha.outcome_frequency)
            percent_outcomes = [(gha.outcome_frequency[m]/total) * 100 for m in range(3)]
            avg_percent_outcomes = [avg_percent_outcomes[j] + percent_outcomes[j] for j in range(3)]
        avg_percent_outcomes = [avg_percent_outcomes[k]/iterations for k in range(3)]
        avg_percent_outcomes_by_tokens.append(avg_percent_outcomes)
        cooperative_epochs_by_tokens.append(total_coop_epochs/iterations)
    avg_percent_outcomes_by_states_by_tokens.append(avg_percent_outcomes_by_tokens)
    cooperative_epochs_by_states_by_tokens.append(cooperative_epochs_by_tokens)
```

```
Starting Computation for computational capacity of 0
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for computational capacity of 1
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for computational capacity of 2
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for computational capacity of 3
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for computational capacity of 4
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for computational capacity of 5
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for computational capacity of 6
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for computational capacity of 7
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
```

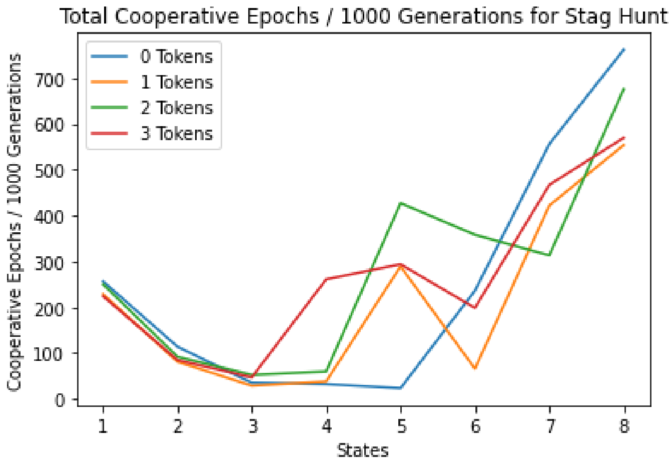
```
for t in range(tokens):
    plt.plot([s+1 for s in range(states)], [avg_percent_outcomes_by_states_by_tokens[p][t][0] for p i
plt.ylabel('Game Outcome Frequency in [%]')
plt.xlabel('States')
plt.title('CC outcomes for Stag Hunt')
plt.legend()
plt.show()
```



```
for t in range(tokens):
    plt.plot([s+1 for s in range(states)], [avg_percent_outcomes_by_states_by_tokens[p][t][1] for p i
plt.ylabel('Game Outcome Frequency in [%]')
plt.xlabel('States')
plt.title('DD outcomes for Stag Hunt')
plt.legend()
plt.show()
```



```
for t in range(tokens):
    plt.plot([s+1 for s in range(states)], [cooperative_epochs_by_states_by_tokens[s][t] for s in rar
plt.ylabel('Cooperative Epochs / 1000 Generations')
plt.xlabel('States')
plt.title('Total Cooperative Epochs / 1000 Generations for Stag Hunt')
plt.legend()
plt.show()
```

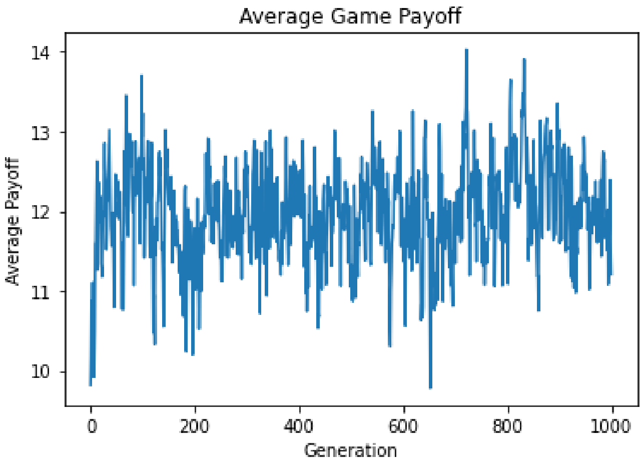


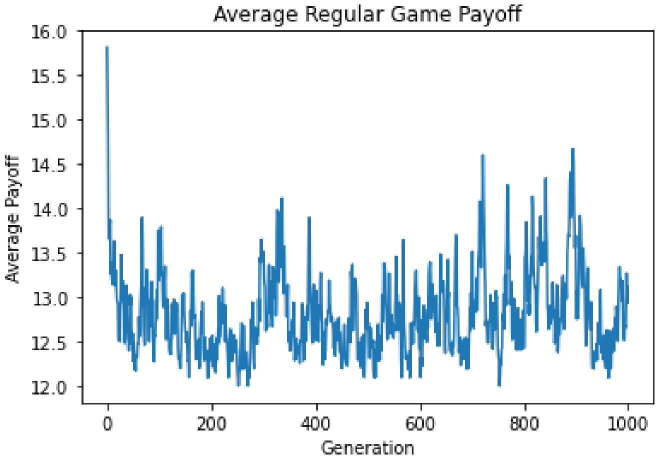
```
print(avg_percent_outcomes_by_states_by_tokens)
print(cooperative_epochs_by_states_by_tokens)
```

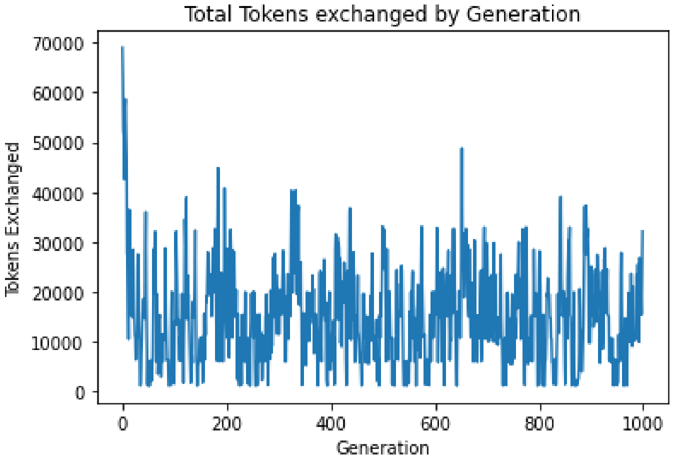
```
[[[13.437080492847187, 42.79415846504355, 43.76876104210926], [12.654586785166 21, 43.93823050701119,
[[256.6, 228.6, 250.2, 224.2], [114.2, 81.4, 92.2, 85.0], [35.4, 29.0, 52.6, 47.2], [32.6, 38.2, 60.0
```

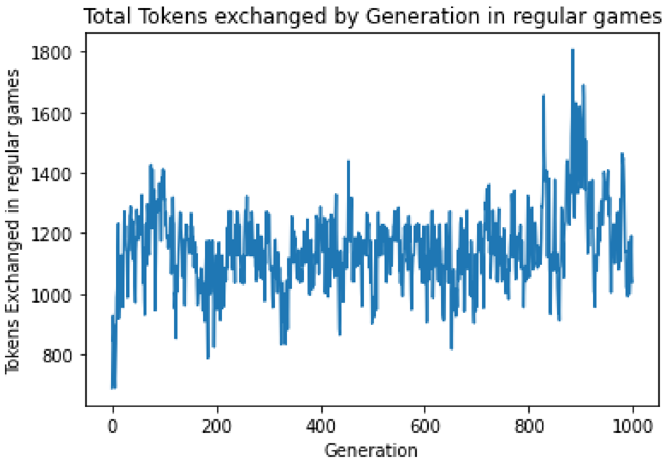
Sheet

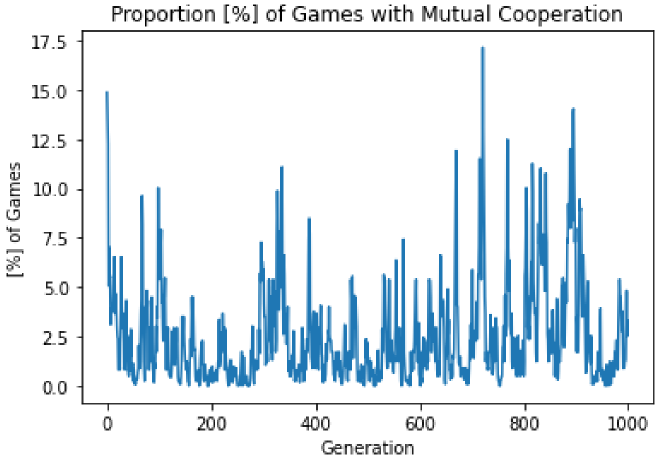
```
#%%

from Model import Model
from GameHistoryAnalysis import GameHistoryAnalysis
import matplotlib.pyplot as plt

#%%

number_of_agents = 50
generations = 1000
game = 'SH'
learning_mechanism = 'GA'
agent_computation_capacity = 4
tokens = 3
timeout = 100

model = Model(number_of_agents, agent_computation_capacity, tokens, generations, learning_mechanism,
model.run_model()

#%%

gha = GameHistoryAnalysis(model.game_history, number_of_agents)
gha.plot_average_payoff()
gha.plot_average_regular_game_payoff()

#%%

gha.plot_total_communication()
gha.plot_total_regular_communication()

#%%

gha.plot_cooperation_percentage()

#%%

gha.plot_average_chat_length()
gha.plot_average_regular_chat_length()

#%%

gha.plot_number_of_unique_conversations()


#%%

gha.plot_NCD_agents()

#%%

gha.plot_CRC_agents()

#%%

gha.plot_CD_agents()
```

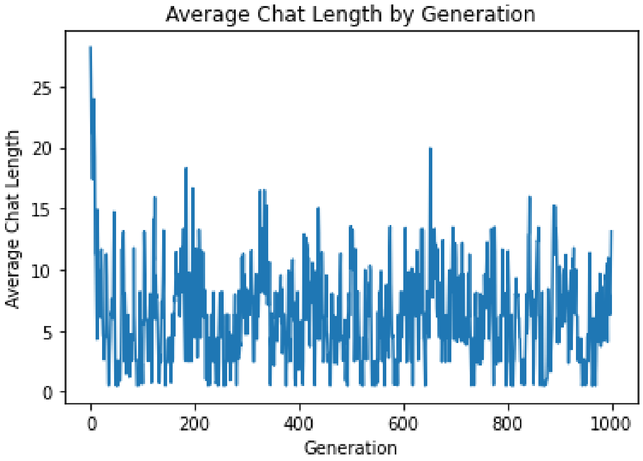

Average Game Payoff
Average Payoff
10
11
12
13
14
0
200
400
600
800
1000
Generation

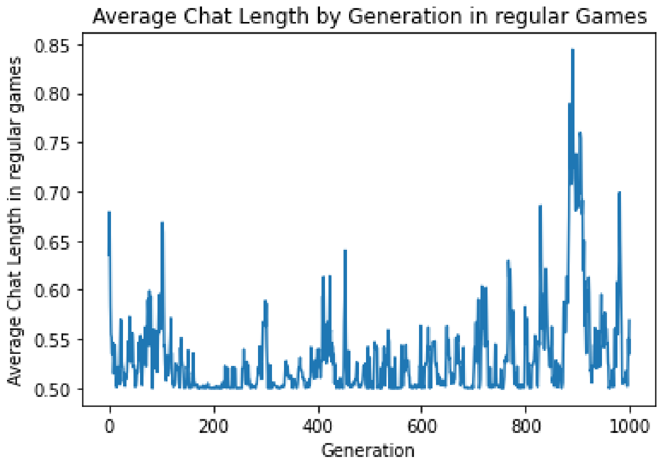

Average Regular Game Payoff
Average Payoff
12.0
12.5
13.0
13.5
14.0
14.5
15.0
15.5
16.0
0
200
400
600
800
1000
Generation

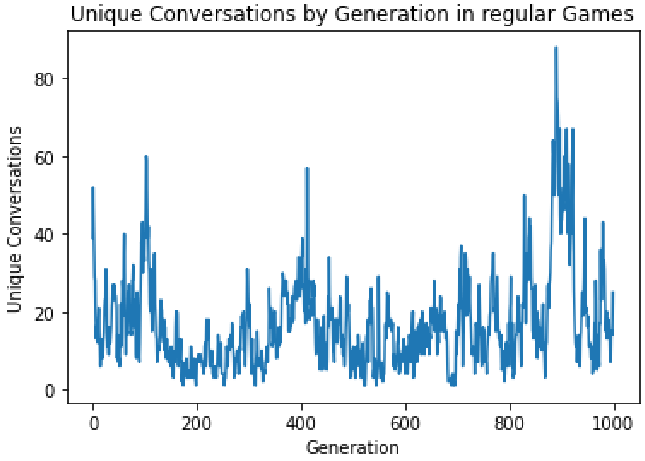

Total Tokens exchanged by Generation
Tokens Exchanged
0
10000
20000
30000
40000
50000
60000
70000
0
200
400
600
800
1000
Generation

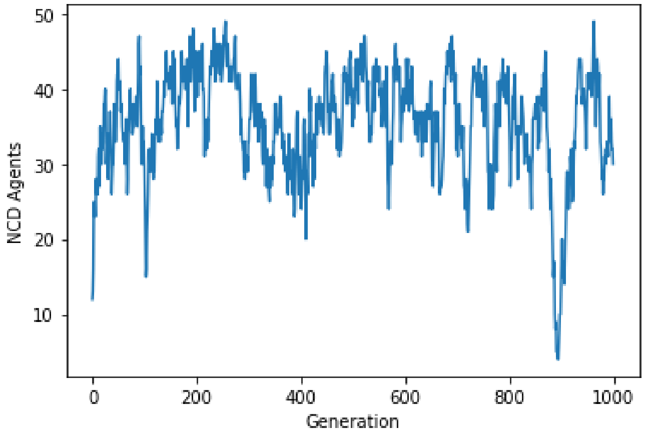

Total Tokens exchanged by Generation in regular games

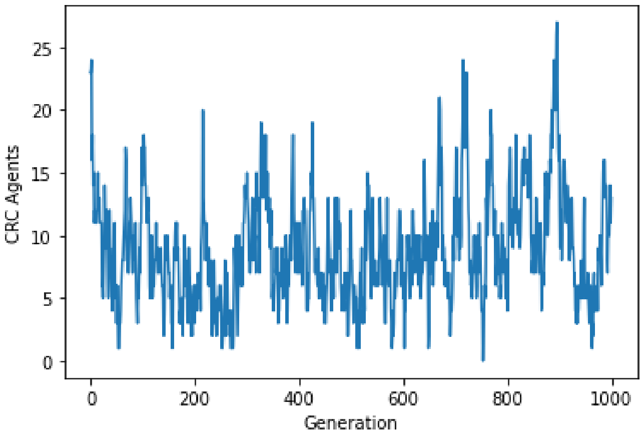

Proportion [%] of Games with Mutual Cooperation

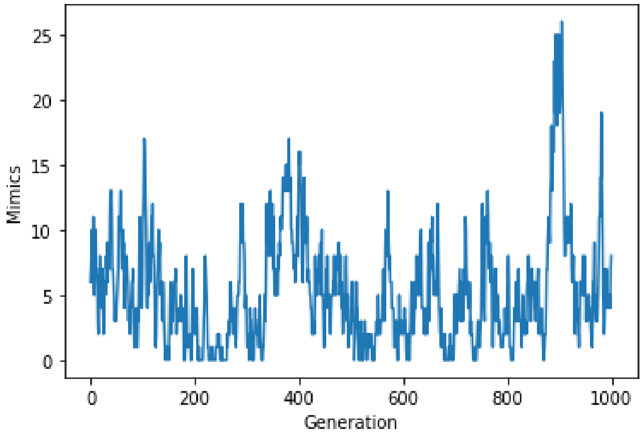

Average Chat Length by Generation


Average Chat Length by Generation in regular Games
Average Chat Length in regular games
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0
200
400
600
800
1000
Generation


Unique Conversations by Generation in regular Games
Unique Conversations
80
60
40
20
0
0
200
400
600
800
1000
Generation


NCD Agents
50
40
30
20
10
0
200
400
600
800
1000
Generation


CRC Agents
Generation
0
5
10
15
20
25
0
200
400
600
800
1000


Mimics
Generation
0
5
10
15
20
25
0
200
400
600
800
1000

Sheet 3

```
#continue from 8 states onward


number_of_agents = 50
generations = 1000
game = 'SH'
learning_mechanism = 'GA'
tokens = 4
timeout = 100
iterations = 5
states = 13

cooperative_epochs_by_states_by_tokens = []
avg_percent_outcomes_by_states_by_tokens = []
for s in range(9, states):
    print('Starting Computation for computational capacity of ' + str(s))
    cooperative_epochs_by_tokens = []
    avg_percent_outcomes_by_tokens = []
    for t in range(tokens):
        print('Starting Computation for communication tokens: ' + str(t))
        total_coop_epochs = 0
        avg_percent_outcomes = [0, 0, 0]
        for i in range(iterations):
            print('Starting Iteration ' + str(i))
            model = Model(number_of_agents, s+1, t+1, generations, learning_mechanism, timeout, game)
            model.run_model()
            gha = GameHistoryAnalysis(model.game_history, number_of_agents)
            total_coop_epochs += gha.cooperative_epochs

            total = sum(gha.outcome_frequency)
            percent_outcomes = [(gha.outcome_frequency[m]/total) * 100 for m in range(3)]
            avg_percent_outcomes = [avg_percent_outcomes[j] + percent_outcomes[j] for j in range(3)]
        avg_percent_outcomes = [avg_percent_outcomes[k]/iterations for k in range(3)]
        avg_percent_outcomes_by_tokens.append(avg_percent_outcomes)
        cooperative_epochs_by_tokens.append(total_coop_epochs/iterations)
    avg_percent_outcomes_by_states_by_tokens.append(avg_percent_outcomes_by_tokens)
    cooperative_epochs_by_states_by_tokens.append(cooperative_epochs_by_tokens)
    print(avg_percent_outcomes_by_states_by_tokens)
    print(cooperative_epochs_by_states_by_tokens)
```

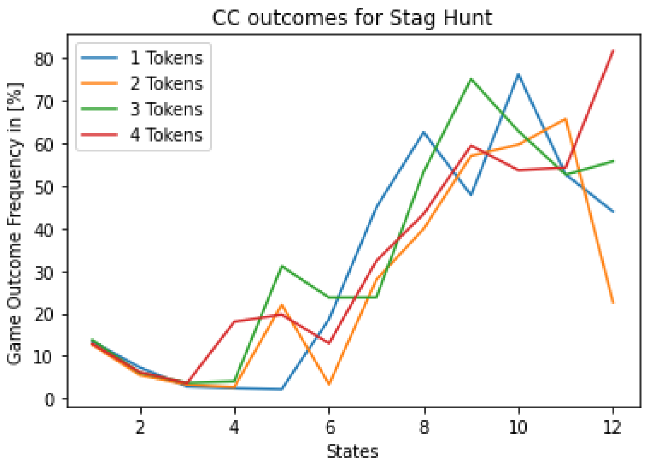
```
Starting Computation for computational capacity of 9
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
[[[47.810021823429665, 39.620570178051175, 12.569407998519159], [57.088657428484716, 26.4121409473535
[[563.8, 721.6, 877.6, 768.2]]
Starting Computation for computational capacity of 10
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
[[[47.810021823429665, 39.620570178051175, 12.569407998519159], [57.088657428484716, 26.4121409473535
[[563.8, 721.6, 877.6, 768.2], [885.6, 724.2, 751.8, 747.0]]
Starting Computation for computational capacity of 11
Starting Computation for communication tokens: 0
Starting Iteration 0
```

Sheet 4

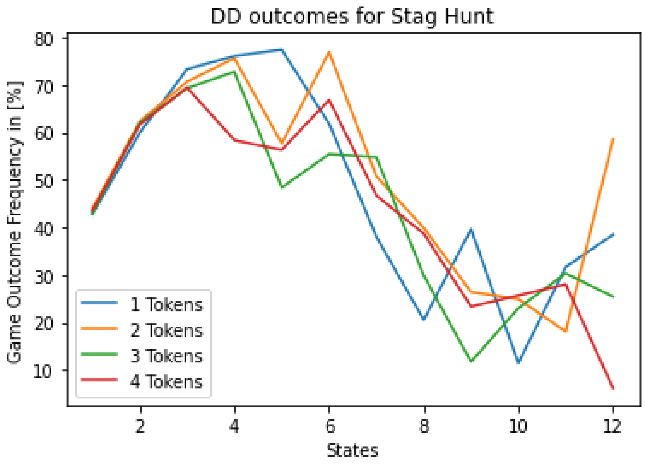
```python
from Model import Model
from GameHistoryAnalysis import GameHistoryAnalysis
import matplotlib.pyplot as plt


number_of_agents = 50
generations = 1000
game = 'SH'
learning_mechanism = 'GA'
tokens = 4
timeout = 100
iterations = 5
states = 12

cooperative_epochs_by_states_by_tokens = [[256.6, 228.6, 250.2, 224.2], [114.2, 81.4, 92.2, 85.0], [3
avg_percent_outcomes_by_states_by_tokens = [[[13.437080492847187, 42.79415846504355, 43.7687610421092
for s in range(11, states):
    print('Starting Computation for computational capacity of ' + str(s))
    cooperative_epochs_by_tokens = []
    avg_percent_outcomes_by_tokens = []
    for t in range(tokens):
        print('Starting Computation for communication tokens: ' + str(t))
        total_coop_epochs = 0
        avg_percent_outcomes = [0, 0, 0]
        for i in range(iterations):
            print('Starting Iteration ' + str(i))
            model = Model(number_of_agents, s+1, t+1, generations, learning_mechanism, timeout, game)
            model.run_model()
            gha = GameHistoryAnalysis(model.game_history, number_of_agents)
            total_coop_epochs += gha.cooperative_epochs

            total = sum(gha.outcome_frequency)
            percent_outcomes = [(gha.outcome_frequency[m]/total) * 100 for m in range(3)]
            avg_percent_outcomes = [avg_percent_outcomes[j] + percent_outcomes[j] for j in range(3)]
        avg_percent_outcomes = [avg_percent_outcomes[k]/iterations for k in range(3)]
        avg_percent_outcomes_by_tokens.append(avg_percent_outcomes)
        cooperative_epochs_by_tokens.append(total_coop_epochs/iterations)
    avg_percent_outcomes_by_states_by_tokens.append(avg_percent_outcomes_by_tokens)
    cooperative_epochs_by_states_by_tokens.append(cooperative_epochs_by_tokens)
    print(avg_percent_outcomes_by_states_by_tokens)
    print(cooperative_epochs_by_states_by_tokens)
```

```
Starting Computation for computational capacity of 11
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
[[[13.437080492847187, 42.79415846504355, 43.76876104210926], [12.65458678516621, 43.93823050701119,
[[256.6, 228.6, 250.2, 224.2], [114.2, 81.4, 92.2, 85.0], [35.4, 29.0, 52.6, 47.2], [32.6, 38.2, 60.0
Starting Computation for computational capacity of 12
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
[[[13.437080492847187, 42.79415846504355, 43.76876104210926], [12.65458678516621, 43.93823050701119,
[[256.6, 228.6, 250.2, 224.2], [114.2, 81.4, 92.2, 85.0], [35.4, 29.0, 52.6, 47.2], [32.6, 38.2, 60.6
```

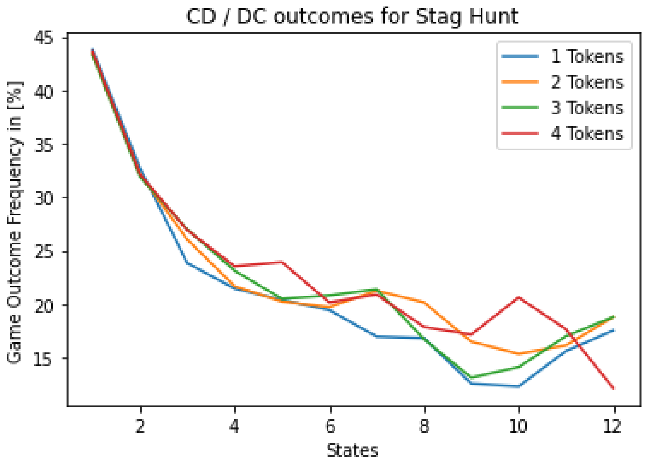
```
states = 12
for t in range(tokens):
    plt.plot([s+1 for s in range(states)], [avg_percent_outcomes_by_states_by_tokens[p][t][0] for p i
plt.ylabel('Game Outcome Frequency in [%]')
plt.xlabel('States')
plt.title('CC outcomes for Stag Hunt')
plt.legend()
plt.show()
```



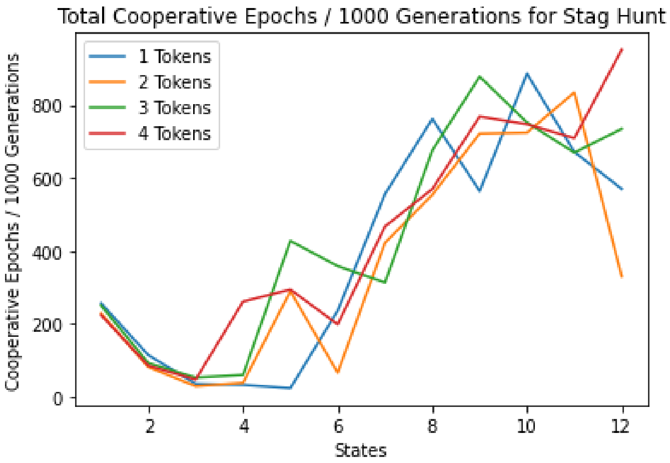
```
for t in range(tokens):
    plt.plot([s+1 for s in range(states)], [avg_percent_outcomes_by_states_by_tokens[p][t][1] for p i
plt.ylabel('Game Outcome Frequency in [%]')
plt.xlabel('States')
plt.title('DD outcomes for Stag Hunt')
plt.legend()
plt.show()
```



```
for t in range(tokens):
    plt.plot([s+1 for s in range(states)], [avg_percent_outcomes_by_states_by_tokens[p][t][2] for p i
plt.ylabel('Game Outcome Frequency in [%]')
plt.xlabel('States')
plt.title('CD / DC outcomes for Stag Hunt')
plt.legend()
plt.show()
```



```
for t in range(tokens):
    plt.plot([s+1 for s in range(states)], [cooperative_epochs_by_states_by_tokens[s][t] for s in rar
plt.ylabel('Cooperative Epochs / 1000 Generations')
plt.xlabel('States')
plt.title('Total Cooperative Epochs / 1000 Generations for Stag Hunt')
plt.legend()
plt.show()
```


Total Cooperative Epochs / 1000 Generations for Stag Hunt
1 Tokens
2 Tokens
3 Tokens
4 Tokens
Cooperative Epochs / 1000 Generations
0
200
400
600
800
2
4
6
8
10
12
States