

CSE5319-001

SPEC TOPS THEORY / ALGORITHMS

Name: Venkata Susanth Bondalapati

Uid: 1001945746

Q1.

Q(1)	Rock	Paper	Scissors	Spock	Lizard
Rock	(1, -1)	(0, 1)	(1, 0)	(0, 1)	(1, 0)
Paper	(1, 0)	(-1, -1)	(0, 1)	(1, 0)	(0, 1)
Scissors	(0, 1)	(1, 0)	(-1, -1)	(0, 1)	(1, 0)
Spock	(1, 0)	(0, 1)	(1, 0)	(-1, -1)	(0, 1)
Lizard	(0, 1)	(1, 0)	(0, 1)	(1, 0)	(-1, -1)

this is correlated equilibrium

```
a payoff 0.444444 b payoff 0.555556 objective is 1
z distribution is:
(1 1 0)
(1 2 0.138889)
(1 3 0.194444)
(1 4 0.0555556)
(1 5 0)
(2 1 0)
(2 2 0)
(2 3 0)
(2 4 0.0555556)
(2 5 0.0555556)
(3 1 0)
(3 2 0)
(3 3 0)
(3 4 0.0555556)
(3 5 0.0555556)
(4 1 0)
(4 2 0)
(4 3 0)
(4 4 0)
(4 5 0)
(5 1 0.0555556)
(5 2 0.0277778)
(5 3 0.194444)
(5 4 0.111111)
(5 5 0)
Model has been successfully processed
```

this is coarse correlated equilibria = 0.5

```
a payoff      0.5 b payoff      0.5 objective is      1
z distribution is:
(1 1          0)
(1 2          0.5)
(1 3          0)
(1 4          0)
(1 5          0)
(2 1          0.25)
(2 2          0)
(2 3          0)
(2 4          0)
(2 5          0)
(3 1          0)
(3 2          0)
(3 3          0)
(3 4          0)
(3 5          0)
(4 1          0)
(4 2          0)
(4 3          0.25)
(4 4          0)
(4 5          0)
(5 1          0)
(5 2          0)
(5 3          0)
(5 4          0)
(5 5          0)
Model has been successfully processed
```

Q2
gambit Nash equilibrium

The screenshot shows the Gambit software interface. At the top, the title bar reads "Gambit - Untitled Strategic Game (unsaved changes)". Below the title bar is a menu bar with "File", "Edit", "View", "Format", "Tools", and "Help". A toolbar contains various icons for file operations, game representation, and analysis. The main window displays a game tree on the left and a payoff matrix on the right. The game tree shows two players, Player 1 and Player 2, each with a choice between W and N. The payoff matrix shows the payoffs for each combination of choices. Player 1's payoffs are (3, 4) for (W, W), (3, 2) for (W, N), (0, 1) for (N, W), and (0, 2) for (N, N). Player 2's payoffs are (0, 1) for (W, W), (1, 1) for (W, N), (1, 0) for (N, W), and (1, 0) for (N, N). A dialog box titled "Computing Nash equilibria" is open in the foreground, displaying the message "The computation has completed." and "Number of equilibria found so far: 3".

Gambit - Untitled Strategic Game (unsaved changes)

File Edit View Format Tools Help

Player 1
Payoff: 7/2

Player 2
Payoff: 7/2

	W		N	
W	3	3	5	4
N	4	5	2	2

Computing Nash equilibria

The computation has completed.

Number of equilibria found so far: 3

#	1: W	1: N	2: W	2: N
1	1	0	0	1
2	0	1	1	0
3	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{4}$

correlated equilibria

```
a payoff      5 b payoff      4 objective is      9
z distribution is:
(1 1      0)
(1 2      1)
(2 1      0)
(2 2      0)
Model has been successfully processed
```

Q3

To compute expected cost per agent in mixed Nash equilibrium Roughgarden 178 is modified for six agents with six edges.

***to get the output run the file Q3.cpp in c ++

// Analyze expected cost per agent for mixed Nash equilibrium.

// Roughgarden, p.178

//Q3 finding expected cost per agent

#include <stdio.h>

```
int main() {
    int binChoice[6];
    int i,sum=0;
    int ballCount[6];

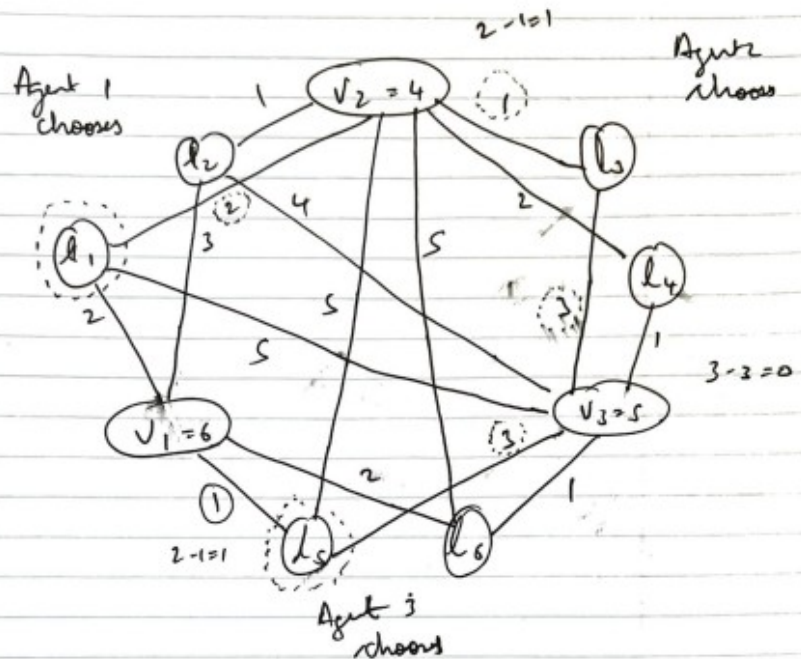
    // Generate each mapping for 6 agents to 6 edges
    for (binChoice[0]=0; binChoice[0]<6; binChoice[0]++)
        for (binChoice[1]=0; binChoice[1]<6; binChoice[1]++)
            for (binChoice[2]=0; binChoice[2]<6; binChoice[2]++)
                for (binChoice[3]=0; binChoice[3]<6; binChoice[3]++)
                    for (binChoice[4]=0; binChoice[4]<6; binChoice[4]++)
                        for (binChoice[5]=0; binChoice[5]<6; binChoice[5]++) {
                            // Clear the edges
                            for (i=0;i<6;i++)
                                ballCount[i]=0;
                            // Count agents for each edge
                            for (i=0;i<6;i++)
                                ballCount[binChoice[i]]++;
                            // Accumulate c(x)=x costs
                            for (i=0;i<6;i++)
                                sum+=ballCount[i]*ballCount[i];
                        }
    // 6 agents * number of choices for choosing bin simultaneously
    printf("expected cost per agent= %10.6f\n",((double) sum)/(6*6*6*6*6*6));
}
```

result :

Output

```
/tmp/dDDtCzLBE4.o
expected cost per agent=  1.833333
```

Q4.)



Agent	choice with its payoffs $\pi_i(s)$	$\sum_{i=1}^k \pi_i(s)$		
$L_1(1)$	$L_2(1)$	$L_3(1)$	10	2
$L_1(2)$	$L_2(1)$	$L_3(2)$	11	3
$L_1(1)$		$L_4(2)$	11	3
$L_1(2)$		$L_4(1)$	10	0
$L_2(2)$	$L_3(2)$	$L_5(2)$	10	2
$L_2(1)$	$L_3(1)$	$L_5(1)$	11	3
$L_2(2)$		$L_4(2)$	12	5
$L_2(1)$		$L_4(1)$	11	2