This document contains some of the supplementary material for

Title: Learning to Alternate
Journal: *Experimental Economics*
Authors: Jasmina Arifovic and John Ledyard

The corresponding author is Jasmina Arifovic, Simon Fraser University, arifovic@sfu.ca

Included in the materials are a toy example of how the IEL simulations proceed, a copy of the python program used to generate an IEL simulation, and a copy of the python program used for the Monte Carlo simulations.

There is also an excel file with the experiment data.

A toy example of an IEL simulation.

The IEL parameters we use are: J = 3, K = 2, mu_v = 0.8, mu_l = 0.8,

The number of rounds = 3

_____

This is round 0

The current strategy for player 0 is:
A = { [0.46 0.58] [0.79] [0.24] }
pi = [0.33 0.33 0.33]
The current plan = [0.24] and the current action = 0.24

The current strategy for player 1 is:
A = { [0.   0.36] [0.09 0.8] [0.09 0.06] }
pi = [0.33 0.33 0.33]
The current plan = [0.   0.36] and tfhe current action = 0.0

IEL for player 0 :
After update A= { [0.58 0.46] [0.79] [0.24] }
First is length experimentation.
Mutate the length of j= 1
The new plan is [0.79 1.]
Mutate the length of j= 2
The new plan is [0.24 1.]
After Length experimentation A= { [0.58 0.46] [0.79 1.] [0.24 1.] }
Next is value experimentation.
Mutate the value of j= 0
After value experimentation A= { [0.44 0.4] [0.79 1.] [
0.24 1] }
Next, update W = [3.64 8.99 8.99]
Next is replication.
For plan 0 j1= 2 j2= 2 : A 0 = [0.24 1.] , W 0 = 8.99
For plan 1 j1= 0 j2= 2 : A 1 = [0.24 1.] , W 1 = 8.99
For plan 2 j1= 0 j2= 0 : A 2 = [0.44 0.4] , W 2 = 3.64
After replication A= { [0.24 1.] [0.24 1.] [0.44 0.4] }
and W = [8.99 8.99 3.64]

IEL for player 1 :
After update A= { [0.36 0.] [0.8   0.09] [0.06 0.09] }
1 is continuing a plan. Further updating is unnecessary.

This is round 1

The current strategy for player 0 is:
A = { [0.24 1.] [0.24 1.] [0.44 0.4] }
pi = [0.42 0.42 0.17]
The current plan = [0.24 1.] and tfhe current action = 0.24

The current strategy for player 1 is:
A = { [0.36 0.] [0.8   0.09] [0.06 0.09] }
pi = [0.42 0.42 0.17]
The current plan = [0.36] and tfhe current action = 0.36

IEL for player 0 :
After update A= { [1.   0.24] [1.   0.24] [0.4   0.44] }
0 is continuing a plan. Further updating is unnecessary.

IEL for player 1 :
After update A= { [0.   0.36] [0.09 0.8] [0.09 0.06] }
First is length experimentation.
Mutate the length of j= 0
0 len=K The new plan is [0.36]
Mutate the length of j= 1
1 len=K The new plan is [0.8]
Mutate the length of j= 2
2 len=K The new plan is [0.06]
After Length experimentation A= { [0.36] [0.8] [0.06] }
Next is value experimentation.
Mutate the value of j= 0
Mutate the value of j= 1
Mutate the value of j= 2
After value experimentation A= { [0.36] [0.71] [0.12] }
Next, update W = [4.8   5.91 4.01]
Next is replication.
For plan 0 j1= 1 j2= 2 : A 0 = [0.71] , W 0 = 5.91
For plan 1 j1= 0 j2= 2 : A 1 = [0.36] , W 1 = 4.8
For plan 2 j1= 0 j2= 1 : A 2 = [0.71] , W 2 = 5.91
After replication A= { [0.71] [0.36] [0.71] }
and W = [5.91 4.8   5.91]

This is round 2

The current strategy for player 0 is:
A = { [1.   0.24] [1.   0.24] [0.4   0.44] }
pi = [0.42 0.42 0.17]
The current plan = [1] and tfhe current action = 1.0

The current strategy for player 1 is:

A = { [0.71] [0.36] [0.71] }
pi = [0.36 0.29 0.36]
The current plan = [0.36] and tfhe current action = 0.36

IEL for player 0 :
After update A= { [0.24 1.] [0.24 1.] [0.44 0.4] }
First is length experimentation.
Mutate the length of j= 0
0 len=K The new plan is [1]
Mutate the length of j= 1
1 len=K The new plan is [1]
After Length experimentation A= { [1] [1] [0.44 0.4] }
Next is value experimentation.
Mutate the value of j= 0
Mutate the value of j= 1
Mutate the value of j= 2
After value experimentation A= { [0.97] [0.99] [0.43 0.37] }
Next, update W = [5.72 5.72 5.56]
Next is replication.
For plan 0 j1= 0 j2= 1 : A 0 = [0.99] , W 0 = 5.72
For plan 1 j1= 2 j2= 0 : A 1 = [0.97] , W 1 = 5.72
For plan 2 j1= 0 j2= 0 : A 2 = [0.97] , W 2 = 5.72
After replication A= { [0.99] [0.97] [0.97] }
and W = [5.72 5.72 5.72]

IEL for player 1 :
After update A= { [0.71] [0.36] [0.71] }
First is length experimentation.
Mutate the length of j= 0
The new plan is [0.71 0.]
Mutate the length of j= 2
The new plan is [0.71 0.]
After Length experimentation A= { [0.71 0.] [0.36] [0.71 0.] }
Next is value experimentation.
Mutate the value of j= 0
Mutate the value of j= 1
After value experimentation A= { [0.74 0.] [0.41] [0.71 0.] }
Next, update W = [10.47 6.89 10.45]
Next is replication.
For plan 0 j1= 0 j2= 0 : A 0 = [0.74 0.] , W 0 = 10.47
For plan 1 j1= 0 j2= 1 : A 1 = [0.74 0.] , W 1 = 10.47
For plan 2 j1= 1 j2= 2 : A 2 = [0.71 0.] , W 2 = 10.45
After replication A= { [0.74 0.] [0.74 0.] [0.71 0.] }
and W = [10.47 10.47 10.45]

This game is over.

m1 + m2 = 1.28 m3 = 0.88

The python program for an IEL simulation

```
import random
from scipy.stats import truncnorm
import numpy as np
from sys import exit

# this section contains the basic IEL parameters and routines

J=180   #number of items in "considered set" S
muv = .033 # rate of mutation of value
sigmav=.05    # variance on the mutation of value # sigmav = float((su-sl))/10
mul = .0033       # rate of mutation of length
K =4    # max length of a "considered strategy"
L = K   # how far back to consider in computing foregone utility

#Types of inspiration:

type =[0,0] #type 1 is all "smart", type 2 is half "smart" half random.
                        #Any other produces  both random.
smart = [0,1] # What smart wants to play

#This initialization takes the types as given
def generalinitialization(I,J,K,sl,su,smart):
        W=[]
        St=[]

        for i in range(I):
                Sit=[]
                temp=[1]*len(range(J))
                W.append(temp)
                if type[i]==1:
                        Sit=[]
                        for j in range(J):
                                Sit.append(smart[:])
                elif type[i]==2:
                        if 2*(J/2)!=J:
                                exit("error J is not even")

                        else:
                                for j in range(J/2):
                                        Sittemp=[]
                                        k= random.randrange(K)
                                        for n in range(k+1):
```

```
                                        Sittemp.append(random.uniform(sl,su))
                                Sit.append(Sittemp)
                                Sit.append(smart[:])

                else:
                        for j in range(J):
                                Sittemp=[]
                                k= random.randrange(K)
                                for n in range(k+1):
                                        Sittemp.append(random.uniform(sl,su))
                                Sit.append(Sittemp)

                St.append(Sit)

        return St,W

def choiceprobabilitiesfori(utilities):
        choicepiti=[]
        e=min(utilities)
        if e <= 0:
                for j in range(J):
                        utilities[j] -= e-1
        sumw=sum(utilities)
        if sumw == 0:
                exit("error - sumw=0")
        for j in range(J):
                choicepiti.append(utilities[j]/float(sumw))
        return choicepiti


def selectionfori(some_list, probabilities):
        x = random.uniform(0, 1)
        cumulative_probability = 0.0
        for item, item_probability in zip(some_list, probabilities):
                cumulative_probability += item_probability
                if x < cumulative_probability: break
        return item
# item is the selected strategy

def foregoneutility(strategy,past_actions,player_name):
        #know L,t
        LL=min(L,t+1)
        v=0
        lenk = len(strategy)
        for d in range(LL):
                v+=utility(strategy[lenk-(d%lenk)-1],past_actions[t-d],player_name)
```

```
        return v/float(LL)

def Vexperimentationfori(strategyset):
                #value experimentation - know sigmav,sl,su,muv, mul
        for j in range(J):
                if onstrat[i]==0:
                        if random.uniform(0,1) < muv:
                                for k in range (len(strategyset[j])):
                                        centers = strategyset[j][k]
                                        r = (truncnorm.rvs((sl-centers)/float(sigmav),
                                                        (su-centers)/float(sigmav),
                                                        loc=centers, scale = sigmav, size
=1))
                                        strategyset[j][k] = np.array(r).tolist()[0]
        return strategyset

def Lexperimentationfori(strategyset,past_actions,player_name):
                #lengthexperimentation
        if K >1:
                for j in range(J):
                        if random.uniform(0,1) < mul:
                                if len(strategyset[j])==1:
                                        alt=strategyset[j][:]
                                        alt.append(1)
                                        strategyset[j].append(0)

        W0=foregoneutility(strategyset[j],past_actions,player_name)

        W1=foregoneutility(alt,past_actions,player_name)
                                        if W1>W0:
                                                strategyset[j]=alt
                                elif len(strategyset[j])==K:
                                        strategyset[j].pop(0)
                                        strategyset[j]
                                else:
                                        if random.uniform(0,1) < .5:
                                                strategyset[j].pop(0)
                                        else:
                                                alt=strategyset[j][:]
                                                alt.append(1)
                                                strategyset[j].append(0)

        W0=foregoneutility(strategyset[j],past_actions,player_name)

        W1=foregoneutility(alt,past_actions,player_name)
                                        if W1>W0:
```

```
                                                    strategyset[j]=alt

        return strategyset


def replicatefori(strategyset,utilities):
                        newS = [0]*J
                        newW = [0]*J
                        for j in range(J):
                                j1=random.randrange(J)
                                j2=random.randrange(J)
                                newS[j]=strategyset[j2][:]
                                newW[j]=utilities[j2]
                                if utilities[j1]> utilities[j2]:
                                        newS[j]=strategyset[j1][:]
                                        newW[j]=utilities[j1]
                        return newS,newW

                        #(newS,newW) = (replicated strategies, corresponding
                        #foregone utilities)

def updateWfori(Set,past_actions,player_name):
        W=[]
        for j in range(J):
                W.append(foregoneutility(Set[j],past_actions,player_name))
        return W

def updateStfori(Set):
        for strats in Set:
                item = strats.pop(0)
                strats.append(item)
        return Set
        #moves all strats in Set up by 1 (last played goes to the back)

#These are the parameters etc. for BoS

I=2  #number of subjects in a game

su = 1 #upper bound on the strategy set [sl,su]
sl = 0  #lower bound on strategy set [sl,su]

        #For BoS there is a basic 2x2 game matrix  [ 0payoffij, 1payoffij ] with 2
players

#pm = [ [0, 9,15,0],[0,15,9,0]]  #the symmetric payoff matrix
```

```
pm=[[3,9,20,3],[3,17,10,3]]   #the asymmetric payoff matrix

# a check for BoS
if len(pm) != I:
        exit("need utility parameters for all I")

alpha= ([[pm[0][3],pm[0][1]-pm[0][0],pm[0][2]-pm[0][0],pm[0][0]-pm[0][1]-
    pm[0][2]+pm[0][3]],[pm[1][3],pm[1][1]-pm[1][0],pm[1][2]-pm[1][0],
    pm[1][0]-pm[1][1]- pm[1][2]+pm[1][3]]]) # utility parameters


def utility(contemplated_action,actionvector, player_name):
                pactionvector=list(actionvector)
                pactionvector[player_name] = contemplated_action
                computed_utility =
(alpha[player_name][0]+alpha[player_name][1]*pactionvector[0]+

        alpha[player_name][2]*pactionvector[1] +

        alpha[player_name][3]*pactionvector[0]*pactionvector[1] )
                return computed_utility

# this is where the simulations start

# this section is for the simulation parameters

rounds=40   # number of rounds per run
runs = 1000  # number of runs in a  simulation

nash = 0 #initialize counts
alt = 0  #initialize counts

for sims in range(runs):

        random.seed()
        S=[]  #stores strategy sets for each run
        a=[]   #stores all actions for a run

#initialize for a run (i.e. one play of repeated game)
        onstrat=[0]*I
        currentstrat=[0]*I

        [St,W]= generalinitialization(I,J,K,sl,su,smart)

        for t in range(rounds):
                at=[]  #initializes actions for round t
```

```
        #this is the game play

                for i in range(I):
                        if onstrat[i] == 0:
                                p=choiceprobabilitiesfori(W[i])
                                currentstrat[i]=selectionfori(St[i],p)[:]
                        at.append(currentstrat[i][0])

                S.append(St)
                a.append(at)

        #this updates a player's "considered set"

                for i in range(I):
                        St[i]=updateStfori(St[i])
                        if len(currentstrat[i])>1:
                                del currentstrat[i][0]
                                onstrat[i] = 1
                        else:
                                onstrat[i]=0
                                St[i]=Lexperimentationfori(St[i],a,i)
                                St[i]=Vexperimentationfori(St[i])
                                W[i]=updateWfori(St[i],a,i)
                                (St[i],W[i])=replicatefori(St[i],W[i])

                #at this point we have S(t+1), W(t+1), onstrat(t+1), x(t+1).
                # we go to next t

        #record keeping before next simulation run
        temp = rounds -1
        m1 =(a[temp][0]-a[temp-1][0])**2 +(a[temp-2][0]-a[temp-3][0])**2
        m2 =(a[temp][1]-a[temp-1][1])**2 +(a[temp-2][1]-a[temp-3][1])**2
        m3=0
        for d in range(4):
                m3 += (a[temp - d][1]-a[temp-d][0])**2
        if (m1+m2 <1) and (m3> 3.5):
                nash +=1
                print a[temp][0],a[temp][1]
        if (m1+m2 >3) and (m3 >3.5):
                alt +=1

#reporting summary results of sims
print "nash = ",  nash/float(sims+1),
print "alternate =", alt/float(sims+1)
```

The python program for the Monte Carlo simulations

```
# Given a probability distribution on types, a number of pairs are matched
#and each plays a game.  That constitutes one observation (a block). Multiple
#blocks are run to generate a distribution.

import random
from scipy.stats import truncnorm
import numpy as np
from sys import exit

#IEL parameters

J=180   #number of items in "considered set" S
muv = .033 # rate of mutation of value
sigmav=.05    # variance on the mutation of value # sigmav = float((su-sl))/10
mul = .0033      # rate of mutation of length
K =2    # max length of a "considered strategy"
L = K   # how far back to consider in computing foregone utility

# IEL pieces

def generalinitialization(I,J,K,sl,su,smart):
        W=[]
        St=[]

        for i in range(I):
                Sit=[]
                temp=[1]*len(range(J))
                W.append(temp)
                if type[i]==1:
                        Sit=[]
                        for j in range(J):
                                Sit.append(smart[:])
                elif type[i]==2:
                        if 2*(J/2)!=J:
                                exit("error J is not even")

                        else:
                                for j in range(J/2):
                                        Sittemp=[]
                                        k= random.randrange(K)
                                        for n in range(k+1):
                                                Sittemp.append(random.uniform(sl,su))
                                        Sit.append(Sittemp)
```

```
                                        Sit.append(smart[:])

                else:
                        for j in range(J):
                                Sittemp=[]
                                k= random.randrange(K)
                                for n in range(k+1):
                                        Sittemp.append(random.uniform(sl,su))
                                Sit.append(Sittemp)

                St.append(Sit)

        return St,W

def choiceprobabilitiesfori(utilities):
        choicepiti=[]
        e=min(utilities)
        if e <= 0:
                for j in range(J):
                        utilities[j] -= e-1
        sumw=sum(utilities)
        if sumw == 0:
                exit("error - sumw=0")
        for j in range(J):
                choicepiti.append(utilities[j]/float(sumw))
        return choicepiti

def selectionfori(some_list, probabilities):
        x = random.uniform(0, 1)
        cumulative_probability = 0.0
        for item, item_probability in zip(some_list, probabilities):
                cumulative_probability += item_probability
                if x < cumulative_probability: break
        return item

def foregoneutility(strategy,past_actions,player_name):
        #know L,t
        LL=min(L,t+1)
        v=0
        lenk = len(strategy)
        for d in range(LL):
                v+=utility(strategy[lenk-(d%lenk)-1],past_actions[t-d],player_name)
        return v/float(LL)

def Vexperimentationfori(strategyset):
                #value experimentation - know sigmav,sl,su,muv, mul
```

```
        for j in range(J):
                if onstrat[i]==0:
                        if random.uniform(0,1) < muv:
                                for k in range (len(strategyset[j])):
                                        centers = strategyset[j][k]
                                        r = (truncnorm.rvs((sl-centers)/float(sigmav),
                                                        (su-centers)/float(sigmav),
                                                        loc=centers, scale = sigmav, size
                                                        =1))
                                        strategyset[j][k] = np.array(r).tolist()[0]
        return strategyset

def Lexperimentationfori(strategyset,past_actions,player_name):
                #lengthexperimentation
        if K >1:
                for j in range(J):
                        if random.uniform(0,1) < mul:
                                if len(strategyset[j])==1:
                                        alt=strategyset[j][:]
                                        alt.append(1)
                                        strategyset[j].append(0)

        W0=foregoneutility(strategyset[j],past_actions,player_name)
        W1=foregoneutility(alt,past_actions,player_name)
                                if W1>W0:
                                        strategyset[j]=alt
                                elif len(strategyset[j])==K:
                                        strategyset[j].pop(0)
                                        strategyset[j]
                                else:
                                        if random.uniform(0,1) < .5:
                                                strategyset[j].pop(0)
                                        else:
                                                alt=strategyset[j][:]
                                                alt.append(1)
                                                strategyset[j].append(0)

        W0=foregoneutility(strategyset[j],past_actions,player_name)
        W1=foregoneutility(alt,past_actions,player_name)
                                                if W1>W0:
                                                        strategyset[j]=alt

        return strategyset

def replicatefori(strategyset,utilities):
                        newS = [0]*J
```

```
                newW = [0]*J
                for j in range(J):
                        j1=random.randrange(J)
                        j2=random.randrange(J)
                        newS[j]=strategyset[j2][:]
                        newW[j]=utilities[j2]
                        if utilities[j1]> utilities[j2]:
                                newS[j]=strategyset[j1][:]
                                newW[j]=utilities[j1]
                return newS,newW

def updateWfori(Set,past_actions,player_name):
        W=[]
        for j in range(J):
                W.append(foregoneutility(Set[j],past_actions,player_name))
        return W

def updateStfori(Set):
        for strats in Set:
                item = strats.pop(0)
                strats.append(item)
        return Set

# this section is the basic game parameters used in these simulations (for BoS)
        # first the strategy set then the utility parameters and functions

I=2  #number of subjects in a game

su = 1 #upper bound on the strategy set [sl,su]
sl = 0  #lower bound on strategy set [sl,su]

#For BoS there is a basic 2x2 game matrix  [ 0payoffij, 1payoffij ] with 2 players

pm = [ [0, 9,15,0],[0,15,9,3]]  # pm is the payoff matrix

alpha= ([[pm[0][3],pm[0][1]-pm[0][0],pm[0][2]-pm[0][0],pm[0][0]-pm[0][1]-
    pm[0][2]+pm[0][3]],[pm[1][3],pm[1][1]-pm[1][0],pm[1][2]-pm[1][0],
    pm[1][0]-pm[1][1]- pm[1][2]+pm[1][3]]]) # utility parameters

def utility(contemplated_action,actionvector, player_name):
                pactionvector=list(actionvector)
                pactionvector[player_name] = contemplated_action
                computed_utility =
(alpha[player_name][0]+alpha[player_name][1]*pactionvector[0]+

        alpha[player_name][2]*pactionvector[1] +
```

```
        alpha[player_name][3]*pactionvector[0]*pactionvector[1] )
                return computed_utility

# this section is the basic IEL parameters used in these BoS simulations

#type =[0,0] #type 1 is all "smart", type 2 is half "smart" half random.
                        # type 0 is random.    Any other produces both random

smart = [0,1]

#probability any agent is type (0,1,2)
typeprobability= [.85,.15,0]

# this is where the simulations start

rounds=40   # number of rounds per run
runs = 200  # number of pairs in a  simulation
numblocks = 1

print "a block =", runs, "simulations"
print "there are", numblocks, "blocks"
print "probability type of (random, smart, half) =",
np.round(typeprobability,decimals=2)
print

print "block #, %Nash, %Alternate"
for block in range(numblocks):
        nash = 0 #initialize counts
        alt = 0  #initialize counts

        for sims in range(runs):

                random.seed()
                S=[]  #stores strategy sets for each run
                a=[]   #stores all actions for a run

                #Initialize for the type of each player in this pair
                z1=selectionfori([0,1,2], typeprobability)
                z2=selectionfori([0,1,2], typeprobability)
                type = [z1,z2]

#initialize for a run (i.e. one play of repeated game)
                onstrat=[0]*I
                currentstrat=[0]*I
                [St,W]= generalinitialization(I,J,K,sl,su,smart)
```

```
        for t in range(rounds):
        #print "round =", t
                at=[]  #initializes actions for round t

        #this is the round play

                for i in range(I):
                        if onstrat[i] == 0:
                                p=choiceprobabilitiesfori(W[i])
                                currentstrat[i]=selectionfori(St[i],p)[:]
                        at.append(currentstrat[i][0])

                S.append(St)
                a.append(at)

                for i in range(I):
                        St[i]=updateStfori(St[i])

                        if len(currentstrat[i])>1:
                                del currentstrat[i][0]
                                onstrat[i] = 1

                        else:
                                onstrat[i]=0
                                St[i]=Lexperimentationfori(St[i],a,i)
                                St[i]=Vexperimentationfori(St[i])
                                W[i]=updateWfori(St[i],a,i)
                                (St[i],W[i])=replicatefori(St[i],W[i])

        #at this point we have S(t+1), W(t+1), onstrat(t+1), x(t+1).
        # we go to next t

#record keeping before next simulation run
        temp = rounds -1
        m1 =(a[temp][0]-a[temp-1][0])**2 +(a[temp-2][0]-a[temp-3][0])**2
        m2 =(a[temp][1]-a[temp-1][1])**2 +(a[temp-2][1]-a[temp-3][1])**2
        m3=0
        for d in range(4):
                m3 += (a[temp - d][1]-a[temp-d][0])**2
        if (m1+m2 <1) and (m3> 3.5):
                nash +=1
        if (m1+m2 >3) and (m3 >3.5):
                alt +=1
    print block, nash/float(sims+1),alt/float(sims+1)
```

Experiment Data

Symmetric Payoff, Full Information

	Symmetric BoS with Public information 5/12/2013 (With Chart)																					
	Group 1		Group 2		Group 3		Group 4		Group 5		Group 6		Group 7		Group 8		Group 9		Group 10		Group 11	
Player ID	14	20	16	19	5	18	9	12	10	11	4	8	1	13	7	17	2	15	3	22	6	21
1	1	0	1	0.1	0	1	1	0	1	1	0	0.6	0	0.4	0	0	0.3	0.45	1	0	0.6	0.5
2	1	1	0	0	0	0	1	0	0	0.9	1	1	1	1	0	1	0.25	0.55	0	0.01	0.5	0.5
3	1	0	1	0	0	1	0	0	1	1	0	1	0	1	0	0	0	0.5	1	0	0.5	0.5
4	1	1	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1	1	0	0	0.5	0.5
5	1	0	1	0	0	1	0	0	1	1	0	1	0	0	0	0	0	1	1	1	0.5	0.5
6	1	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	1	0.5	0.5
7	1	0	1	0	0	1	0	0	0	0	0	1	0	0.9	0	0	0	1	1	0	0.5	0.5
8	1	0	0	1	1	0	0	0	1	1	1	0	0	0.9	0	0	1	0	0	1	0.5	0.5
9	1	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0.5	0.5
10	1	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	1	0	0.5
11	1	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0	0.5
12	1	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	1	0	1
13	1	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0.13	0	1	1	0	0	1
14	1	0	0	1	1	0	0	0	0	1	1	0	1	0	0	0.04	1	0	0	0	0	1
15	1	0	1	0	0	1	0	0	1	0	0	1	1	0	0	0.8	0	1	1	0	0	1
16	1	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0.07	1	0	0	1	0	1
17	1	0	1	0	0	1	0	0	1	0	0	1	0.2	1	0	0.08	0	1	1	0	0	1
18	1	0	0	1	1	0	0	0	0	1	1	0	0	1	0	0.01	1	0	0	1	0	1
19	1	0	1	0	0	1	0	0	1	0	0	1	0	1	0	0.03	0	1	1	0	0	1
20	1	0	0	1	1	0	0	0	0	1	1	0	0	1	0	0	1	0	0	1	0	1
21	0	0	1	0	0	1	0	1	1	0	0	1	0	0	0	0.09	0	1	1	0	0	1
22	0	0	0	1	1	0	0	1	0	1	1	0	1	0.1	0	0.01	1	0	0	1	0	1
23	0	0	1	0	0	1	0	0	1	0	0	1	1	0	0	0.06	0	1	1	0	0	1
24	0	1	0	1	1	0	0	0	0	1	1	1	0	0	0	0.11	1	0	0	1	0	1
25	0	1	1	0	0	1	0	1	1	0	0	1	0	1	0	0.02	0	1	1	0	0	1
26	0	1	0	1	1	0	0	1	0	1	1	0	0	1	0	0	1	0	0	1	0	1
27	0	1	1	0	0	1	0	0	1	0	0	1	1	1	0	0	0	1	1	0	0	0
28	0	1	0	1	1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	1	0	0
29	0	1	1	0	0	1	0	1	1	0	0	1	1	0	0	0	0	1	1	0	0	0
30	0	1	0	1	1	0	0	1	0	1	1	0	0	0	0	0	1	0	0	1	0	0
31	0	1	1	0	0	1	0	0	1	0	0	1	1	0.1	0	0	0	1	1	0	0	0
32	0	1	0	1	1	0	0	0	0	1	1	0	0	0.1	0	0	1	0	0	1	1	0
33	0	1	1	0	0	1	0	1	1	0	0	1	1	0.9	0	0	0	1	1	0	1	0
34	0	1	0	1	1	0	0	1	0	1	1	0	0	0.1	0	0	1	0	0	1	1	0
35	0	1	1	0	0	1	0	0	1	0	0	1	0	0.1	0	0	0	1	1	0	1	0
36	0	1	0	1	1	0	0	0	0	1	1	0	0	1	0	0	1	0	0	1	1	0
37	0	1	1	0	0	1	0	0	1	0	0	1	0	1	0	0	0	1	1	0	1	0
38	0	1	0	1	1	0	0	0	0	1	1	0	0	1	0	0	1	0	0	1	1	0
39	0	1	1	0	0	1	0	0	1	0	0	1	0	1	0	0	0	1	1	0	1	0
40	0	1	0	1	1	0	0	0	0	1	1	0	0	1	0	0	1	0	0	1	1	0

Symmetric BoS with Public information 5/3/2013 (With Chart)																		
Group 1		Group 2		Group 3		Group 4		Group 5		Group 6		Group 7		Group 8		Group 9		
7	12	14	16	8	15	4	10	2	5	11	13	3	9	1	17	6	18	
0.1	0.5	0	0	0.5	0.5	1	1	1	0	1	0	0.6	0.2	0.6	0	0	0.4	
0	0	0	0	0.4	0.5	0	1	0	0	0	0	0.3	0.2	1	0	0	0.9	
0	0.7	0	0	0.1	1	0	1	0	0.1	0.05	0	0	0.2	1	0	0	1	
0	0.3	0	1	0.1	1	0	1	0	0.5	0	0	1	0.4	1	0	0	1	
0	0	0	1	0.1	1	0	1	0	0.6	0.5	0	1	0.9	1	0	0	1	
0	1	0	1	0	0	0	1	0	1	0	1	0	0.1	1	0	0	1	
0	0	0	1	0.1	1	0	1	1	1	0	0	0	1	1	0	0	1	
0	0	0	1	0	1	0	1	1	0.6	0.09	1	0	0	1	0	0	1	
0	0.5	0	1	0	1	0	1	0	0.6	1	0	0	1	1	0	0	1	
0	1	0	1	0	1	0	1	0	0.7	0	1	1	0	1	0	0	1	
0	0	0	1	0	1	0	1	0	0.6	1	0	0	1	1	0	0	1	
0	0	0	1	0	1	0	1	0	0.6	0	1	1	0	1	0	0	1	
0	0	0	1	0	1	0	1	0	0.9	1	0	0	1	1	0	0	1	
0	0	0	1	0	1	0	1	0	0.9	0	1	1	0	1	0	0	1	
0	0	0	1	0	1	0	1	0	0.6	1	0	0	1	1	0	0	1	
0	0	0	1	0	1	0	1	0	0.6	0	1	1	0	1	0	0	1	
0	0	0	1	0	1	0	1	0	0.9	1	0	0	1	1	0	0	1	
0	0	0	1	0	1	0	1	0	1	0	1	1	0	1	0	0	1	
0	0	0	1	0	1	0	1	0	0.1	1	0	0	1	1	0	0	1	
0	0	0	1	0	1	0	1	1	1	0	1	1	0	1	0	0	1	
0	0	0	0	0	0	0	0	0	0.9	1	0	0	1	0	0	0	1	
1	0	0	0	0	0	0	0	0	0.9	0	1	1	0	0	0	0	1	
0	0.5	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	
1	0.5	0	0	0	0	1	0	0	0.9	0	1	1	0	0	0	0	1	
0	1	0	0	0	0	1	0	0	0.9	1	0	0	1	0	1	0	1	
1	0	0	0	0	1	1	0	0	0.9	0	1	1	0	0	1	0	1	
0	1	0	0	0	1	1	0	0	0.9	1	0	0	1	0	1	0	1	
1	0	0	0	0	1	1	0	0	0.9	0	1	1	0	0	1	0	1	
0	1	0	0	0	1	1	0	0	0.6	1	0	0	1	0	1	0	1	
1	0	0	0	0	1	1	0	0	0.7	0	1	1	0	0	1	0	1	
0	1	0	0	0	1	1	0	0	0.7	1	0	0	1	0	1	0	1	
1	0	0	0	0	1	1	0	0	0.7	0	1	1	0	0	1	0	1	
0	1	0	0	0	1	1	0	0	0.8	1	0	0	1	0	1	0	1	
1	0	0	0	0	1	1	0	0	0.8	0	1	1	0	0	1	0	1	
0	1	0	0	0	1	1	0	0	0.9	1	0	0	1	0	1	0	1	
1	0	0	0	0	0	1	0	0	0.9	0	1	1	0	0	1	0	1	
0	1	0	0	0	0	1	0	0	0.9	1	0	0	1	0	1	0	1	
1	0	0	0	0	0	1	0	0	0.9	0	1	1	0	0	1	0	1	
0	1	0	0	0	0	1	0	0	0.9	1	0	0	1	0	1	0	1	
1	0	1	0	0	0	1	0	0	0.9	0	0	1	0	0	1	0	1	

Symmetric Payoff, Dark Information

Symmetric BoS with NO public information 4/19/2013 (With Chart)																								
	Group 1		Group 2		Group 3		Group 4		Group 5		Group 6		Group 7		Group 8		Group 9		Group 10		Group 11	
Player ID	5	20	8	19	3	14	4	17	2	9	10	11	6	18	21	22	1	16	13	15	7	12
1	1	0	0	0	0.5	0	0	0.5	1	0	0	1	0.2	0.7	0.5	0	1	0	1	1	1	0
2	0	0	0	0	0.3	0	1	0.6	0.5	0	0	0.5	0	0.8	1	1	1	0	0.5	0.7	0	0
3	1	1	0	0	1	1	0	0.7	1	0	0	0.7	0.1	1	1	0	1	0	0.2	0.7	1	1
4	0	0.5	0	0	1	0	1	0.5	1	0	0	0.75	0	1	0	0	1	0	0.5	0.7	0	0.5
5	1	0.5	0	0	1	0	0	0.9	0	0	0	0.6	0	1	1	0	1	0	0.65	0.7	1	0
6	0	0	0	0	1	1	1	0.9	1	0	0	0	0	1	1	0	1	0	0.2	0.65	0	1
7	1	0.5	0	0	1	0	1	1	1	0	0	0.85	0	1	1	0	1	0	0	0.65	1	0
8	0	0.5	0	0	0	0	0	0.6	1	0	0	0.37	0	1	1	0	1	0	0.1	0.7	0	1
9	1	0	0	0	1	0	1	0.7	1	0	0	1	0	0	1	0	1	0	0	0.7	1	0
10	0	1	0	0.5	1	1	0	0.7	1	0	0	1	0	0	1	0	1	0	0.5	1	0	1
11	1	0	0	0.5	0	0	0.2	0.8	0	0	0	1	1	0	1	0	1	0	0	0.9	1	0
12	0	1	0	1	0	1	0.8	0.8	0	0	0	1	1	0	0	0	1	0	0.1	0.9	0	1
13	1	0	0	0	1	0	0.8	0.7	0	0	0	1	1	0	0	0	1	0	0.5	0.95	1	0
14	0	1	0	1	0	1	0.7	0.6	0	1	0	1	1	0	0	0	1	0	0.5	0.85	0	1
15	1	0	0	1	1	0	0.6	0.5	0	1	0	1	1	0	0	0.5	1	0	0	0.9	1	0
16	0	1	0	1	0	1	1	0.6	0	1	0	1	0.9	0	0	1	1	0	0.2	0.8	0	1
17	1	0	0	1	1	0	0.8	0.4	0	1	0	1	0.8	0	0	1	1	0	0.1	0.8	1	0
18	0	1	0	1	0	1	0.2	0.3	0	1	0	1	0.2	0	0	1	1	0	0	0.9	0	1
19	1	0	0	1	1	0	0.5	0.6	0	1	0	1	0.1	1	0	1	1	0	0.1	0.8	1	0
20	0	1	0	1	0	1	0.5	0.6	0	1	0	1	0	1	0	1	1	0	1	0.7	0	1
21	1	0	0	1	1	0	0	0.8	0	1	0	1	0	1	1	1	1	0	0	0.6	1	0
22	0	1	0	1	0	1	1	0.9	0	1	0	1	1	1	1	0	1	0	0.15	0.7	0	1
23	1	0	0	1	1	0	0	0.6	0	1	0	1	0	1	1	0	1	0	0.05	0.75	1	0
24	0	1	0	1	0	1	1	0.6	0	1	0	1	0	1	1	0	1	0	0	0.75	0	1
25	1	0	0	1	1	0	0	0.5	0	1	0	1	0	0.9	1	0	1	0	0	0.8	1	0
26	0	1	0	1	0	1	1	0.5	0	1	0	1	0.8	0.8	1	0	1	0	0	0.81	0	1
27	1	0	0	1	1	0	0	0.5	0	1	0	1	1	0	1	0	1	0	0	0.81	1	0
28	0	1	0	1	0	1	0	0.7	0	1	0	1	1	0	1	0	1	0	0	1	0	1
29	1	0	0	1	1	0	0	0.7	0	1	0	1	1	0	1	0	1	0	0	1	1	0
30	0	1	0	1	0	1	1	0.7	0	1	0	1	1	0	0	0	1	0	0	1	0	1
31	1	0	0	1	1	0	0	0	0	1	0	1	1	0	0	1	1	0	0	1	1	0
32	0	1	0	1	0	1	0	0	0	1	0	1	0.8	0	0	1	1	0	0	1	0	1
33	1	0	0	1	1	0	0.6	1	0	1	0	1	0	1	0	1	1	0	0	1	1	0
34	0	1	0	1	0	1	0.7	1	0	1	0	1	0	1	0	1	1	0	0	1	0	1
35	1	0	0	1	1	0	1	0.6	0	1	0	1	0	1	0	1	1	0	0	1	1	0
36	0	1	0	1	0	1	1	0.6	0	1	0	1	0	1	0	1	1	0	0	1	0	1
37	1	0	0	1	1	0	0	0.6	0	1	0	1	0	1	0	1	1	0	0	1	1	0
38	0	1	0	1	0	1	1	0.5	0	1	0	1	0.2	0.8	0	1	1	0	0	1	0	1
39	1	0	0	1	1	0	0	0.5	0	1	0	1	1	0	0	0	1	0	0	1	1	0
40	0	1	1	1	0	1	0	0	0	1	0	1	1	0	0	0	1	0	0	1	0	1

Symmetric BoS with NO public information 3/15/2013 (With Chart)										
Group 1		Group 2		Group 3		Group 4		Group 5	
4	6	5	7	9	10	2	8	1	3
0	0	0	0	1	1	0	0	0	0
0.5	0	0.7	1	0	0	0.5	0.5	0.7	0
1	1	1	0	0	1	0.5	1	1	0
0.5	0	1	0	0	0	0	0	1	0
0.5	0	0	0	1	1	0.5	1	1	0
1	0	0	0	1	0	0	0.5	1	0
0.5	0	0	0	0	1	0	0	1	0
1	0	0	0	1	0	0	1	1	0
1	0	1	0	0	1	0	1	1	0

Asymmetric Payoff, Dark Information

(No Information Beyond their Own Payoffs) Asymmetric Battle of the Sexes (No Information Beyond their Own Payoffs)

	11/1/2012 (No Chart)						11/2/2012 (No Chart)									
	Group 1		Group 2		Group 3		Group 1		Group 2		Group 3		Group 4		Group 5	
Player ID	1	4	2	3	5	6	8	9	1	3	4	10	2	6	5	7
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1
2	0.1	1	0.25	0.5	1	0.8	0.5	0	1	1	0	1	0	0.5	1	1
3	0.05	0.7	0.45	0.3	0	0.9	0.5	0.5	1	1	0.45	0	0	0	1	1
4	0.01	0.5	0.67	0	0.5	1	0	0.75	1	1	0.5	0.2	0	0	1	1
5	0	0.6	0.33	0.6	0	1	1	0.5	0	1	0.4	0.5	0	1	1	1
6	0.01	0.45	0.5	0.8	1	1	0.7	0.5	0	0	0.45	0.75	0	0	1	1
7	0.01	0.45	0.8	1	0.9	0.2	0.2	0.5	0	1	0.5	1	0	0	1	1
8	0.01	0.4	0.4	1	1	1	0.1	0.25	0	1	0.3	1	0	0	1	1
9	0.01	0.35	0.45	0.8	1	0.19	0	0	0	0	0.25	1	0	0	1	1
10	0.5	0.3	0.55	1	0.99	1	0	0	0	1	0.55	1	0	0	1	1
11	0.01	0.28	0.9	0.75	1	0.15	0	0.5	0	0	0.6	1	0	0	1	1
12	0.01	0.24	0.2	0.7	1	1	0	0.5	0	1	0.65	1	0	0	0	1
13	0.01	0.22	0.25	0.9	0	0.14	0	0.75	0	0	0.7	1	0	0	0	1
14	0.01	0.17	0.15	0.4	1	1	0	1	0	1	0.75	1	0	0	0	1
15	0.01	0.11	0.11	0.7	0	1	0	1	0	1	0.8	1	0	0	0	1
16	0.01	0	0.18	0.9	0.6	1	1	1	0	1	0.85	1	0	0	0	0
17	0.01	1	0.16	0	0	0.77	1	0.8	0	1	0.9	1	0	0	0	0
18	0.01	0.02	0.44	0.3	1	1	0.6	0.8	0	1	1	1	0	0	0	1
19	0.01	0	0.33	0.5	0.95	0.18	0.8	0.75	1	1	1	1	0	0	0	1
20	0.01	0	0.3	0.9	0	1	0.4	0.75	1	1	1	1	0	0	0	1
21	0.01	0	0.1	0.7	0.05	1	0.5	0.8	1	1	0.2	1	0	0	0	1
22	0.01	0	0.5	0.8	1	1	0.7	0.8	1	1	1	1	0	0	0	1
23	0.01	0	0.8	0.8	1	0.15	0.1	0.9	1	1	1	1	0	0	0	0
24	0.01	0	0.9	0.8	0	1	0.9	0	1	1	1	1	0	0	0	0
25	0.01	0	0.15	1	0	1	0.1	0.75	1	1	1	1	0	0	0	0
26	0	0	0.25	0.9	1	1	0	0.75	1	1	1	1	0	0	0	0.5
27	0	0	0.25	0.8	1	1	0	0.75	1	1	1	1	0	0	0	0
28	0	0	0.25	0.7	1	1	0	0.5	1	1	1	1	0	0	0	0
29	0	0	0.45	0.6	1	1	0	0.2	1	1	1	1	0	0	0	0
30	0	0	0.35	0.55	1	1	0	0	1	1	1	1	0	0	0	0
31	0	0	0.3	0.7	1	1	0	0	1	1	1	1	0	0	0	0
32	0	0	0.3	0.8	1	1	0	0	1	1	1	1	0	0	0	0
33	0	0	0.33	0.8	1	1	0	1	1	1	1	1	0	0	0	0
34	0	0	0.33	0.8	1	1	1	1	1	1	1	1	0	0	0	0
35	0	0	0.33	0.8	1	1	1	1	1	1	1	1	0	0	0	0
36	0	0	0.4	0.7	1	1	1	0	1	1	1	1	0	0	0	0
37	0	0	1	0.7	1	1	0	0	1	1	1	1	0	0	0	0
38	0	0	0.78	0.65	1	1	0	0	1	1	1	1	0	0	0	0
39	0	0	1	0.7	1	1	0	1	1	1	1	1	0	0	0	0
40	0	0	0.67	0.72	1	1	1	1	1	1	1	1	0	0	0	0

Asymmetric Battle of the Sexes (No Information Beyond their Own Payoffs)

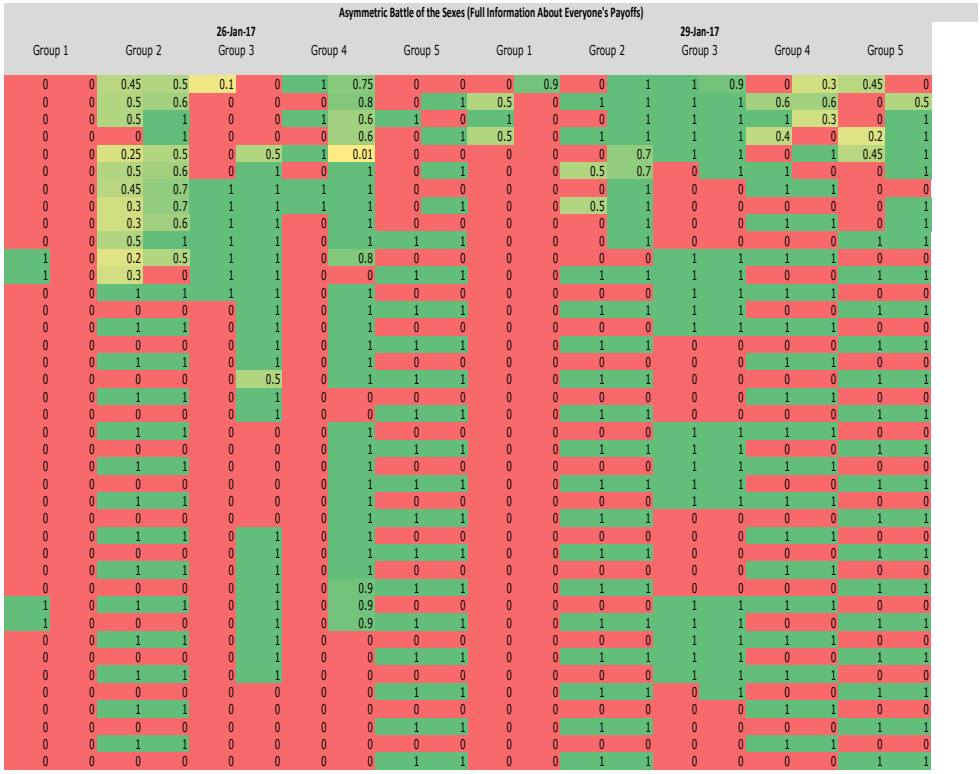
11/9/2012 (With Chart)								11/16/2012 (With Chart)									
Group 1		Group 2		Group 3		Group 4		Group 1		Group 2		Group 3		Group 4		Group 5	
1	2	5	6	4	8	3	7	6	9	2	7	1	8	5	10	3	4
0	0	0	0.9	0.7	1	0.5	0.6	0	1	0.1	1	0	0.4	0.3	1	0	1
0	1	0.3	0.9	0.25	1	0	0.6	0	0.5	0.3	0.5	0	0.4	0.7	1	0	1
0	0	0.15	0.8	0.4	1	0	0.5	0	0.7	0	1	0	0.3	1	1	0.5	1
0	1	0.25	0.7	0.4	0	0	0.3	0	0.3	0.3	0.7	0	0	1	1	1	1
0	1	0.35	0.6	0.35	1	0	0.4	0	0.1	0.5	0.9	0	0	1	1	1	1
1	0.7	0.42	0.5	0.45	0	0	0.1	0	0	0.7	0.96	0	0	1	1	1	1
0.5	0.8	0.5	0.55	0.5	1	0	0	0	1	0.9	1	0	0	1	1	1	1
0.1	0.7	0	0.55	0.2	1	0	0.8	0	0	0.9	1	0	0	1	1	1	1
0	1	0.1	0.7	0.4	1	0	0.5	0	0	0.9	0.95	0	0	1	1	1	1
0.4	0.6	0.15	1	0.45	1	0	0.4	0	0	0.7	0.9	0	0	1	1	1	1
0.6	0.8	0	1	0.35	1	0	0	0	0	1	1	0	0	1	1	1	1
0.7	0	0.05	1	0.5	0	0	0	0	0	1	1	0	0	1	1	1	1
0.2	1	0.4	0.95	0.5	1	0	0	0	0	1	1	0	0	0.9	1	1	1
1	0.8	0.3	0.9	0.5	1	0	0	0	0	0.5	1	0	0	1	1	1	1
0	0.95	0.35	0.85	0.75	1	0	1	0	0	0.6	0.9	0	0	1	1	1	1
0	0.7	0.55	0.9	0.85	1	0	0	0	1	0.6	0.9	0	0	0	1	1	1
0	0.5	0.28	0.85	1	1	0	0	0	0	0.6	1	0	0	0	1	1	1
0	0.8	0.35	0.85	1	1	0	0	0	0	0.5	1	0	0	1	0	1	1
0	0.3	0.3	0.8	1	1	0	0	0	0	0.5	0	0	0	0	1	1	1
0	0.8	0.28	0.8	1	1	0	0	0	0	0.2	1	0	0	0	1	1	1
0	0.7	0.32	1	1	1	0	0	0	0	0.2	0.5	0	0	0	1	1	1
0	0.8	0.88	1	1	1	0	0	0	0	0.1	0.6	0	0	0	1	1	1
0	0.3	0.7	0.6	1	1	0	0	0	0	0.1	0.65	0	0	0.5	1	1	1
0	0.7	0	0.7	1	1	0	0	0	0	0.1	0.7	0	0	0	1	1	1
0	0.8	1	1	1	1	0	0	0	0	0.1	0.8	0	0	0	0	1	1
0	0.6	0	1	1	1	0	0	0	0	0.1	0.4	0	0	0	0	0	1
0	0.7	0	1	1	1	0	0	0	0	0.1	0.3	0	0	0	0	0	1
0	1	0.1	1	1	1	0	0	0	0	0.1	0.2	0	0	0	0	0	1
1	0	0.1	0.75	1	1	0	0	0	0	0.1	0.1	0	0	0	0	0	0
0	1	0.1	0.85	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0.12	0.9	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0.12	0.95	1	1	0	0	0	0	0	0.1	0	0	0	0	0	0
1	0	0.16	1	1	1	0	0	0	0	0	0.3	0	0	0	0	0	0
0	0.7	0.17	0.9	1	1	0	0	0	0	0	0.5	0	0	0	0	0	0
0	0.8	0.18	0.8	1	1	0	0	0	0	0	0.7	0	0	0	0	0	0
0	0.3	0.15	0.7	1	1	0	0	0	0	0	0.9	0	0	0	0	0	0
0	0.2	0.1	0.8	1	1	0	0	0	0	0	1	0	0	0	0	0	0
0	0.2	0.17	0.9	1	1	0	0	0	0	0	1	0	0	0	0	0	0
0	0.1	0.16	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0.19	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Asymmetric Battle of the Sexes Dark					
1/28/2013 (With Chart)					
Group 1		Group 2		Group 3	
2	5	3	4	1	6
0	0.7	0	1	0	0.7
0.1	1	0.7	0.5	0	0.3
0	1	0.3	0.45	0	0
0.5	1	0.25	0.55	0	0
1	1	0.35	0.2	0	0
1	1	0.35	0.2	0	0
0	1	0.35	0	0	0
1	0	0.2	0.3	0	0
0.5	0	0.3	0.2	0	0
0	0	0.25	1	0	0
0	0	0.25	1	0	0
0	0	0.25	1	0	0
0	1	1	1	0	0
0	1	1	0	0	0
0	1	1	0	0	0
0	1	0	0	0	0
1	1	0	1	0	0
1	1	1	1	0	0
1	1	0	0	0	0
1	0	1	1	0	0
0	0	0	0	0	0
0	0	1	1	0	0
0	0	0	0	0	0
0	1	1	1	0	0
0	1	0	0	0	0
1	1	1	1	0	0
1	1	0	0	0	0
1	1	1	1	0	0
1	0	0	0	0	0
0	0	1	1	0	0
0	0	0	0	0	0
0	0	1	1	0	0
1	1	0	0	0	0
1	1	1	1	0	0
1	1	0	0	0	0
0	0	1	1	0	0
0	0	0	0	0	0
0	0	1	1	0	0
1	1	0	0	0	0
1	1	1	1	0	0

Asymmetric Payoff, Full Information

Asymmetric Battle of the Sexes (Full Information About Everyone's Payoffs)

Date	16-Jan-17												19-Jan-17									
	Group 1		Group 2		Group 3		Group 4		Group 5		Group 1		Group 2		Group 3		Group 4		Group 5	
Player ID																				
1	0.7	1	0.5	1	0.2	1	1	0.6	0.6	0	1	0.8	0	1	0	0	0	0.5	0.5	0
2	0	1	1	1	0.5	1	0.4	0.6	0	0.6	1	1	0.7	0.5	0	1	0.1	1	0	1
3	0	0.5	0	1	1	0	0	0.5	0	1	1	1	0.5	0.5	1	0	0	0.7	1	0
4	0.5	0	0	1	0	1	0	0.7	0	1	1	1	0.5	0.6	0	1	0	0.7	1	1
5	0.3	0.5	0	0	0	0	0.1	0.9	0.6	1	1	1	0.5	0.7	0	0	0	0.5	0	0
6	0.3	0.5	0	0	0	1	0.9	0.95	1	1	1	1	0.4	0.8	1	1	0	0	1	1
7	0.2	0	0	0	0	0	1	0.95	1	0.6	1	1	0.2	0.7	0	0	0	1	0	0
8	0	0.4	0	1	0	1	1	0.95	0	1	1	1	0.3	0.4	1	1	0	1	1	1
9	0	0.2	0	0.5	0	1	1	0.95	0.6	0	1	1	0.4	0.2	0	0	0	1	0	0
10	0	0	0	0	0	1	1	0.95	0	1	1	1	0.5	0.3	1	1	0	1	1	1
11	0	0	0	0	1	1	1	1	0	0	1	1	0.5	0.2	0	0	0	1	0	0
12	0	0	0	0	1	1	1	1	0	1	1	1	0.5	0.4	1	1	0	1	1	1
13	0	0	0	0	1	1	0	1	0	0	1	1	0.5	0.6	0	0	0	1	0	0
14	0	0	0	0	0	1	0.1	0.95	1	1	1	1	0.5	0.6	1	1	0	1	1	1
15	0	0	0	0	1	0	1	0.99	0	0	1	1	0.4	0.6	0	0	0	0.9	0	0
16	0	0	0	0	0	0	1	1	1	1	1	1	0.4	0.5	1	1	0	0.9	1	1
17	0	0	0	0	0	1	0.5	1	0	0	1	1	0.5	0.4	0	0	0	1	0	0
18	0	0	0	0	1	1	0.5	1	1	1	1	1	0.4	0.3	1	1	0	0.9	1	1
19	0	0	0	0	1	0	0.4	1	0	0	1	1	0.4	0.4	0	0	0	0.9	0	0
20	0	0	0	0	0	1	0.3	1	1	1	1	1	0.4	0.5	1	1	0	0.9	1	1
21	0	0	0	0	1	0	1	0.8	0	0	1	1	0.5	0.6	0	0	0	0.9	0	0
22	0	0	0	0	1	1	0.8	1	1	1	0	1	0.4	0.9	1	1	0	1	1	1
23	0	0	0	0	0	0	0	0.9	0	0	0	0	0.5	0.5	0	0	0	1	0	0
24	0	0	0	0	0	1	0.1	0.9	1	1	0	0	0.5	0.5	1	1	0	1	1	1
25	0	0	0	0	0	1	0.15	0.95	0	0	0	0	0.5	0.6	0	0	0	1	0	0
26	0	0	0	0	0	1	0.1	0.95	1	1	0	0	0.4	0.6	1	1	0	1	1	1
27	0	0	0	0	0	1	0.2	0.98	0	0	0	0	0.4	0.5	0	0	0	1	0	0
28	0	0	0	0	0	1	1	0.99	1	1	0	0	0.5	0.5	1	1	0	0.6	1	1
29	0	0	0	0	1	1	1	1	0	0	0	0	0.5	0.5	0	0	0	0.6	0	0
30	0	0	0	0	0	1	1	1	1	1	0	0	0.5	0.5	1	1	0	0.6	1	1
31	0	0	0	0	1	1	1	1	0	0	0	0	0.5	0.5	0	0	0	0.6	0	0
32	0	0	0	0	1	1	1	1	1	1	0	0	0.5	1	1	1	0	1	1	1
33	0	0	0	0	1	1	1	1	0	0	0	0	0.4	0.5	0	0	0	1	0	0
34	0	0	0	0	1	1	1	1	1	1	0	0	0.5	0.5	1	1	0	1	1	1
35	0	0	0	0	1	1	1	1	0	0	0	0	0.5	0.5	0	0	1	1	0	0
36	0	0	0	0	1	1	1	1	1	1	0	0	0.5	0.5	1	1	1	1	1	1
37	0	0	0	0	1	1	1	1	0	0	0	0	0.5	0.5	0	0	1	1	0	0
38	0	0	0	0	1	1	1	1	1	1	0	0	0.5	0.7	1	1	1	1	1	1
39	0	0	0	0	1	1	1	1	0	0	0	0	0.4	0.5	0	0	1	1	0	0
40	0	0	0	0	1	1	1	1	1	1	0	0	0.5	1	1	1	1	1	1	1


Asymmetric Battle of the Sexes (Full Information About Everyone's Payoffs)
26-Jan-17
29-Jan-17
Group 1
Group 2
Group 3
Group 4
Group 5
Group 1
Group 2
Group 3
Group 4
Group 5