Linear, Split

This section is used to introduce the ideas that are generalized in the Arbitrary Reciprocation Function section.

In[142]:=
```
Clear["Global`*"];
```

The function g is used to simplify the p0 payoff function below. This function is the sum of 1. the proportion amount of data sent from peer i to peer j *prior to* round t=0 and 2. the amount of data sent from i to j during t=0, given a resource distribution resources. It assumes a 'split' initial reputation (which accounts for the $\frac{1}{2}$ term). The second term is the calculation peer i does to weigh peer j using the linear reciprocation function (which is the same as the identity function).

The first term in this function is specific to the 'split' initial reputation distribution, and basically says that peer i will send half of its data to each of its peers in the first round. This is independent of the choice of (pure) reciprocation function, as the peers having the same reputation at t=0 means they have the same weights in the first round.
The second term is specific to both the linear reciprocation function and split initial reputation distribution. The RF determines the amount of data sent from peers j and k, which sets the quotient in this term, and the linear RF is the same as the identity function, which makes it unnecessary to pass each of the values that comprise the quotient into a reciprocation function.

In[143]:=
$$\texttt{g[resources\_, i\_, j\_, k\_] :=}\ \frac{\texttt{resources[[i]]}}{2} + \frac{\texttt{resources[[j]]}}{\texttt{resources[[j]]} + \texttt{resources[[k]]}}\ \texttt{resources[[i]];}$$

p0 gives the payoff of peer 0 in round t+1, given a resource distribution resources. The second parameter, d, is the deviation from the linear reciprocation strategy. This is dependent on both the linear RF and split initial reputation conditions.

In[144]:=
$$\texttt{p0[resources\_, d\_] :=}\ \frac{\frac{\texttt{g[resources,1,2,3]+d}}{\texttt{g[resources,2,1,3]}}}{\frac{\texttt{g[resources,1,2,3]+d}}{\texttt{g[resources,2,1,3]}} + \frac{\texttt{g[resources,3,2,1]}}{\texttt{g[resources,2,3,1]}}}\ \texttt{resources[[2]]} + \frac{\frac{\texttt{g[resources,1,3,2]-d}}{\texttt{g[resources,3,1,2]}}}{\frac{\texttt{g[resources,1,3,2]-d}}{\texttt{g[resources,3,1,2]}} + \frac{\texttt{g[resources,2,3,1]}}{\texttt{g[resources,3,2,1]}}}\ \texttt{resources[[3]];}$$

In[145]:=
```
rf = Identity;
```

In[146]:=
```
resources = {B0, B0, B0};
```

In[147]:=
```
p0[resources, d] // FullSimplify
```

Out[147]=
$$\frac{4\ \mathrm{B0}^3 - 2\ \mathrm{B0}\ d^2}{4\ \mathrm{B0}^2 - d^2}$$

In[148]:=
```
ArgMax[{p0[resources, d], B0 > 0,
  -resources[[2]]/(resources[[2]] + resources[[3]]) * resources[[1]] ≤ d ≤ resources[[1]] -
   resources[[2]]/(resources[[2]] + resources[[3]]) * resources[[1]]}, d, Reals] // FullSimplify
```

Out[148]=
```
{ 0              B0 > 0
{ Indeterminate  True
```

Arbitrary Reciprocation Function

In[149]:=
```
Clear["Global`*"];
```

In[150]:=
```
p0[g_, rf_, resources_, d_] :=
  rf[(g[rf,resources,1,2,3]+d)/g[rf,resources,2,1,3]] /
    (rf[(g[rf,resources,1,2,3]+d)/g[rf,resources,2,1,3]] + rf[g[rf,resources,3,2,1]/g[rf,resources,2,3,1]]) resources[[2]] +
  rf[(g[rf,resources,1,3,2]-d)/g[rf,resources,3,1,2]] /
    (rf[(g[rf,resources,1,3,2]-d)/g[rf,resources,3,1,2]] + rf[g[rf,resources,2,3,1]/g[rf,resources,3,2,1]]) resources[[3]];
```

Split

This section generalizes the previous to use any reciprocation function (rather than being specific to just the linear RF).

In[151]:=
```
gsplit[rf_, resources_, i_, j_, k_] :=
  resources[[i]] * (1/2 + rf[resources[[j]]/resources[[i]]] /
     (rf[resources[[j]]/resources[[i]]] + rf[resources[[k]]/resources[[i]]]));
```

Proportional

In[152]:=
```
initialSend[resources_, i_, j_] :=
  If[i == j, 0, resources[[i]] * resources[[j]] / Sum[If[l == i, 0, resources[[l]]], {l, 3}]];
initialLedgers[resources_] := Partition[
   initialSend[resources, #1, #2] & @@@ Tuples[{Range[3], Range[3]}], 3];
```

In[154]:=
```
gprop[rf_, resources_, i_, j_, k_] := initialLedgers[resources][[i]][[j]] +
   rf[resources[[j]]] / (rf[resources[[j]]] + rf[resources[[k]]]) resources[[i]];
```

Constant

In[155]:=
```
gconst[c_, rf_, resources_, i_, j_, k_] :=
```
$$\texttt{resources[[i]]} * \left(c + \frac{\texttt{rf}\left[\frac{\texttt{resources[[j]]}}{\texttt{resources[[i]]}}\right]}{\texttt{rf}\left[\frac{\texttt{resources[[j]]}}{\texttt{resources[[i]]}}\right] + \texttt{rf}\left[\frac{\texttt{resources[[k]]}}{\texttt{resources[[i]]}}\right]}\right);$$

In[156]:=
```
g1[rf_, resources_, i_, j_, k_] := gconst[1, rf, resources, i, j, k]
```

Test Cases

In[157]:=
```
p0s = {p0[gsplit, ##] &, p0[gprop, ##] &, p0[g1, ##] &};
(* payoff functions (one for each set of initial ledgers) *)
```

In[158]:=
```
resources = {B0, B0, B0}; (* resource dist *)
```

Linear RF

Homogeneous

In[159]:=
```
rf[d_] := Identity; (* linear RF *)
```

In[160]:=
```
Do[
```
$$d_{opt} = \texttt{ArgMax}\left[\left\{\texttt{p[rf, resources, d], B0} > 0, -\frac{\texttt{rf[resources[[2]]]}}{\texttt{rf[resources[[2]]] + rf[resources[[3]]]}}\right.\right.$$
$$\texttt{resources[[1]]} \le d \le \texttt{resources[[1]]} - \frac{\texttt{rf[resources[[2]]]}}{\texttt{rf[resources[[2]]] + rf[resources[[3]]]}}$$
$$\left.\left.\texttt{resources[[1]]}\right\}, \texttt{d, Reals}\right] \texttt{ // FullSimplify;}$$
```
 Print[d_opt],
 {p, p0s}]
```

$$\begin{cases} 0 & B0 > 0 \\ \text{Indeterminate} & \text{True} \end{cases}$$

$$\begin{cases} 0 & B0 > 0 \\ \text{Indeterminate} & \text{True} \end{cases}$$

$$\begin{cases} 0 & B0 > 0 \\ \text{Indeterminate} & \text{True} \end{cases}$$

Non-homogeneous

In[161]:=
```
resources = {B0, B1, B1};
```

In[162]:=
```
Do[dopt = ArgMax[{p[rf, resources, d], B0 > 0, B1 > 0,
      -rf[resources[[2]]]/(rf[resources[[2]]] + rf[resources[[3]]]) resources[[1]] ≤ d ≤
       resources[[1]] - rf[resources[[2]]]/(rf[resources[[2]]] + rf[resources[[3]]]) resources[[1]]},
     d, Reals] // FullSimplify ;
  Print[dopt],
  {p, p0s}]
```

$$\begin{cases} 0 & B1 > 0 \,\&\&\, B0 > 0 \\ \text{Indeterminate} & \text{True} \end{cases}$$

$$\begin{cases} 0 & B1 > 0 \,\&\&\, B0 > 0 \\ \text{Indeterminate} & \text{True} \end{cases}$$

$$\begin{cases} 0 & B1 > 0 \,\&\&\, B0 > 0 \\ \text{Indeterminate} & \text{True} \end{cases}$$

In[163]:=
```
Clear[rf];
```

Sigmoid RF

In[164]:=
```
resources = {B0, B0, B0}
```

Out[164]= {B0, B0, B0}

In[165]:=
```
rf[d_] := 1/(1 + Exp[1 - x]);
```

In[166]:=
```
Do[
  dopt = ArgMax[{p[rf, resources, d], B0 > 0, -rf[resources[[2]]]/(rf[resources[[2]]] + rf[resources[[3]]])
        resources[[1]] ≤ d ≤ resources[[1]] - rf[resources[[2]]]/(rf[resources[[2]]] + rf[resources[[3]]])
         resources[[1]]}, d, Reals] // FullSimplify;
  Print[dopt],
  {p, p0s}]
```

$$\begin{cases} 0 & B0 > 0 \\ \text{Indeterminate} & \text{True} \end{cases}$$

$$\begin{cases} 0 & B0 > 0 \\ \text{Indeterminate} & \text{True} \end{cases}$$

$$\begin{cases} 0 & B0 > 0 \\ \text{Indeterminate} & \text{True} \end{cases}$$

In[167]:=
```
Clear[rf];
```

Tanh RF

In[168]:=
```
rf[z_] := Tanh;
```

In[169]:=
```
Do[
  dopt = ArgMax[{p[rf, resources, d], B0 > 0, -rf[resources[[2]]]/(rf[resources[[2]]] + rf[resources[[3]]])
      resources[[1]] ≤ d ≤ resources[[1]] - rf[resources[[2]]]/(rf[resources[[2]]] + rf[resources[[3]]])
       resources[[1]]}, d, Reals] // FullSimplify;
  Print[dopt],
  {p, p0s}]
```

$$\begin{cases} 0 & B0 > 0 \\ \text{Indeterminate} & \text{True} \end{cases}$$

$$\begin{cases} 0 & B0 > 0 \\ \text{Indeterminate} & \text{True} \end{cases}$$

$$\begin{cases} 0 & B0 > 0 \\ \text{Indeterminate} & \text{True} \end{cases}$$