Define the 5 functions, chosen to match https://en.wikipedia.org/wiki/Beta_distribution#/media/-File:Beta_distribution_pdf.svg . Note that the labeling in this code (a, b, c, d, e) does not coincide with the labeling in Figure 1 (A, B, C, D, E). The correct mapping is given as comments in the code:

```
In[1]:= fa = BetaDistribution[1/2, 1/2] (* A in Fig. 1 *);
fb = BetaDistribution[5, 1] (* E in Fig. 1 *);
fc = BetaDistribution[1, 3] (* B in Fig. 1 *);
fd = BetaDistribution[2, 2] (* D in Fig. 1 *);
fe = BetaDistribution[2, 5] (* C in Fig. 1 *);
```

The following function takes in a sequence of (distribution, multiplier) pairs. It then computes the probability that the largest scaled utility drawn comes from the first given distribution.

```
In[6]:= probmax[dist1_, m1_, dist2_, m2_, dist3_, m3_, dist4_, m4_, dist5_, m5_] :=
  Integrate[PDF[dist1, x] * CDF[dist2, m1/m2 * x] * CDF[dist3, m1/m3 * x] *
    CDF[dist4, m1 / m4 * x] * CDF[dist5, m1 / m5 * x], {x, 0, 1}]
```

The next function takes in input as above, but computes the expected utility conditional on the agent having the largest scaled utility.

```
In[7]:= condexputil[dist1_, m1_, dist2_, m2_, dist3_, m3_, dist4_, m4_, dist5_, m5_] :=
  Integrate[x * PDF[dist1, x] * CDF[dist2, m1/m2 * x] * CDF[dist3, m1/m3 * x] *
     CDF[dist4, m1 / m4 * x] * CDF[dist5, m1 / m5 * x], {x, 0, 1}] /
   probmax[dist1, m1, dist2, m2, dist3, m3, dist4, m4, dist5, m5]
```

Multipliers generated in Python to equalize the probabilities:

```
In[8]:= ma = Rationalize[1., 0];
mb = Rationalize[0.8339729309082031, 0];
mc = Rationalize[2.0838546752929688, 0];
md = Rationalize[1.2098731994628906, 0];
me = Rationalize[2.0375137329101562, 0];
```

Using these multipliers, measure how far the probabilities of each agent having the largest scaled utility deviate from the optimal point of 1/5, which is always at most 2*10^-6.

```
In[13]:= N[probmax[fa, ma, fb, mb, fc, mc, fd, md, fe, me] - 1 / 5, 20]
N[probmax[fb, mb, fa, ma, fc, mc, fd, md, fe, me] - 1 / 5, 20]
N[probmax[fc, mc, fb, mb, fa, ma, fd, md, fe, me] - 1 / 5, 20]
N[probmax[fd, md, fb, mb, fc, mc, fa, ma, fe, me] - 1 / 5, 20]
N[probmax[fe, me, fb, mb, fc, mc, fd, md, fa, ma] - 1 / 5, 20]
```

Out[13]= $1.9926178471538369383 \times 10^{-6}$

Out[14]= $-1.9075898272784769614 \times 10^{-6}$

Out[15]= $2.3870431949109106351 \times 10^{-8}$

Out[16]= $1.1530367464121009604 \times 10^{-6}$

Out[17]= $-1.2619351982365700437 \times 10^{-6}$

Finally, we check the difference between the expected utility of agent i for an item conditioned on i receiving the item and i’s expected utility for an item without conditioning. This difference is much smaller for the agent with distribution fb, labeled E in the paper.

In[18]:=
```
N[condexputil[fa, ma, fb, mb, fc, mc, fd, md, fe, me] - Expectation[x, x ≈ fa], 20]
N[condexputil[fb, mb, fa, ma, fc, mc, fd, md, fe, me] - Expectation[x, x ≈ fb], 20]
N[condexputil[fc, mc, fb, mb, fa, ma, fd, md, fe, me] - Expectation[x, x ≈ fc], 20]
N[condexputil[fd, md, fb, mb, fc, mc, fa, ma, fe, me] - Expectation[x, x ≈ fd], 20]
N[condexputil[fe, me, fb, mb, fc, mc, fd, md, fa, ma] - Expectation[x, x ≈ fe], 20]
```

Out[18]= 0.41627014764957458444

Out[19]= 0.064716892528258072566

Out[20]= 0.29611394460090903177

Out[21]= 0.26294068032548731606

Out[22]= 0.22236473132350756367