A *Mathematica* exposition of a breach of contract model

Seth J. Chandler 2011

Introduction

If anyone is interested in what goes on in my Analytic Methods for Lawyers class, here' s what I expect my students to be able to follow by the end of the semester. That's the time of the course when we briefly address "Law and Economics." I expect them to be able to follow both the ideas and the Mathematica code shown here, or at least most of it.

The Contract "Game"

In their book Analytical Methods for Lawyers (2d ed.)(pp. 403-07), Prof. Howell Jackson and his colleagues discuss a model for breach of contract. The professors consider a situation in which a seller is making a desk. It will either cost $300 or $2000 to make the desk There' s a 90 % chance it will only cost $300 but a 10 % chance that it will cost $2000. The buyer is willing to pay $700 for the desk and values the desk at $1000. The idea is to develop a damage model that will result in the cellar efficiently performing his contract.

The diagram below presents an extensive form of the game.

Mathematica Code

I first set up a function **max** that takes a set of payoffs and finds the maximum payoff for the player whose has the right to make a decision.

In[1]:=

```
max[p_, payoffs_] := Part[payoffs, First@Ordering[Part[payoffs, All, p], -1]]
```

I then create a spectral measure (**f**) of the wealths of the player following some lottery. One can find information on spectral measures at this web site. Basically, however, a spectral measure can be thought of as the expected quantile of a distribution of wealths, where the quantiles are weighted according to a non-increasing probability distribution on the interval [0,1]. Examples of such weighting distributions are UniformDistribution[{0,1}], TruncatedDistribution[{0,1},ExponentialDistribution[2]] and many others, with each distribution representing a form of risk aversion. A key point is that spectral measures are akin to certainty equivalent wealth. If the outcomes are measured in dollars, the spectral measure is in dollars. It is legitimate to add spectral measures together to create a total spectral measure, or total wealth.

In[2]:=
```
f[ϕList_(* a list of weighting distributions *), playerIndex_Integer
  (* the playeer that gets to decide what action will be taken *),
  probList_(*the probability of each outcome *), outcomeList_
  (* the payoffs to the players from each outcome*)] :=
 With[{lotteryResults = Map[max[playerIndex, #] &, outcomeList]},
  MapThread[Expectation[Quantile[EmpiricalDistribution[probList → #1], p], p ≈ #2] &,
   {Transpose@lotteryResults, ϕList}]]
```

I can also create a variant of **f** that uses numeric computation rather than attempting symbolic evaluation.

In[3]:=
```
fN[ϕList_, playerIndex_Integer, probList_, outcomeList_] :=
 With[{lotteryResults = Map[max[playerIndex, #] &, outcomeList]},
  MapThread[NExpectation[Quantile[EmpiricalDistribution[probList → #1], p], p ≈ #2] &,
   {Transpose@lotteryResults, ϕList}]]
```

I can then define the outcomes from the contracts game, where d1 represents the damages the breaching party pays the non-breaching party in the event the cost of performance is $300 and d2 represents the damages thebreaching party pays the non-breaching party in the event the cost of performance is $2000. Notice that if the law is unable to distinguish the two situations or the finder of fact is unable to figure out which occurred, we can constrain d1=d2. Further note that we could have a function d with a more extensive domain that mapped the cost of performance into damages.

In[4]:=
```
outcomes[d1_, d2_, price_: 700, clo_: 300, chigh_: 2000, v_: 1000] :=
 {{{v - price, price - clo}, {-price + d1, price - d1}},
  {{v - price, price - chigh}, {-price + d2, price - d2}}}
```

In[5]:=
```
probs[α_: 0.9] := {α, 1 - α};
```

So, by way of example, I can now calculate the spectral measures for the players assuming the buyer (player 1) has a weighting function of TruncatedDistribution[{0,1},ExponentialDistribution[0.7]] and the seller (player 2) has a weighting function of TruncatedDistribution[{0,0.5},ExponentialDistribution[0.1]]. The code works tolerably fast.

In[6]:=
```
Timing[f[{TruncatedDistribution[{0, 1}, ExponentialDistribution[0.7]],
   TruncatedDistribution[{0, 0.5}, ExponentialDistribution[0.1]]},
  2, probs[0.9], outcomes[300, 200]]]
```

Out[6]=
```
{0.31782, {-413.43, 400.}}
```

Using numerical methods speeds things up.

In[7]:=
```
Timing[fN[{TruncatedDistribution[{0, 1}, ExponentialDistribution[0.7]],
   TruncatedDistribution[{0, 0.5}, ExponentialDistribution[0.1]]},
  2, probs[0.9], outcomes[300, 200]]]
```

Out[7]=
```
{0.058164, {-413.43, 400.}}
```

I can take the total of this and get the total wealth produced by this damages regime.

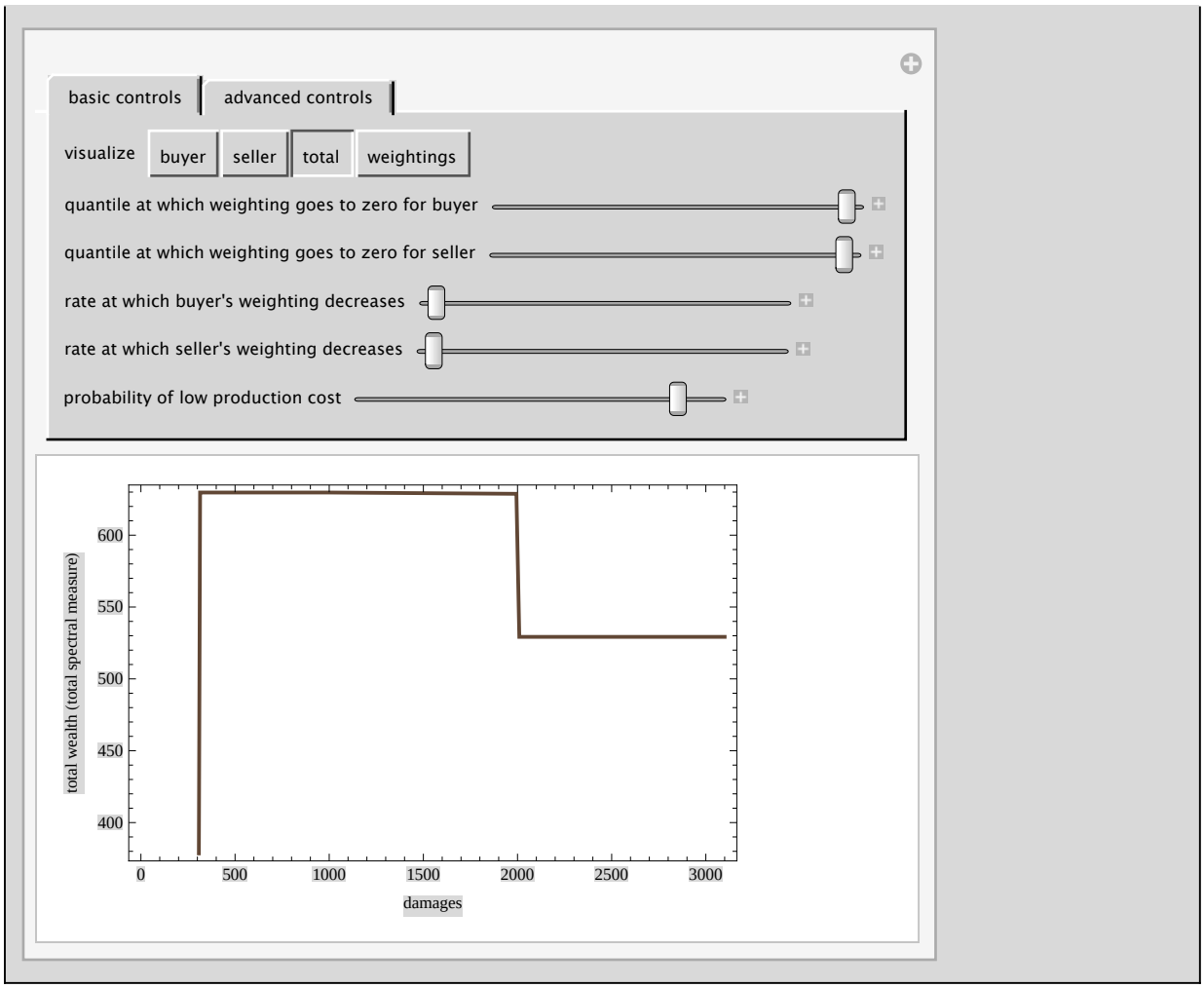
In[8]:=
```
Total[%[[2]]]
```

Out[8]=
```
-13.4295
```

Cost Invariant Damages

I can now produce a Manipulate that let' s one see how the damages regime relates to total wealth under a variety of scenarios. In this Manipulate, the damages are not permitted to depend on the cost of performance to the seller. This Manipulate must be used asynchronously and one must be patient. It will take a while to produce a respectable plot. You can choose to visualize the weighting functions, the spectral measure of the buyer, the spectral measure of the seller and the total spectral measure.

Out[9]=

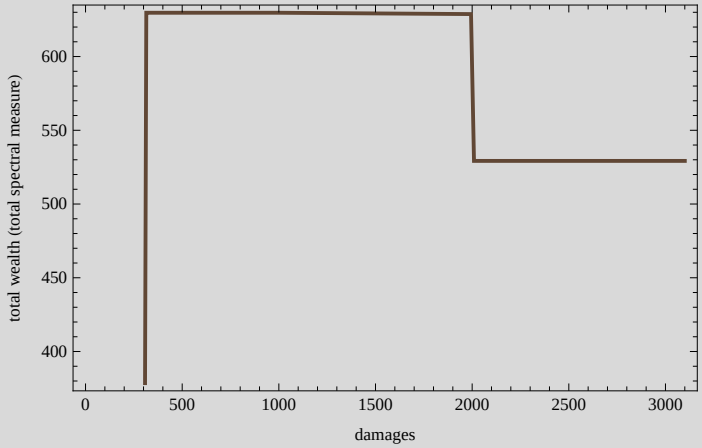


The graphic below illustrates a scenario in which both parties are minimally and equally risk averse damages from $400 to $2000 appear to maximize total wealth.

In[10]:=

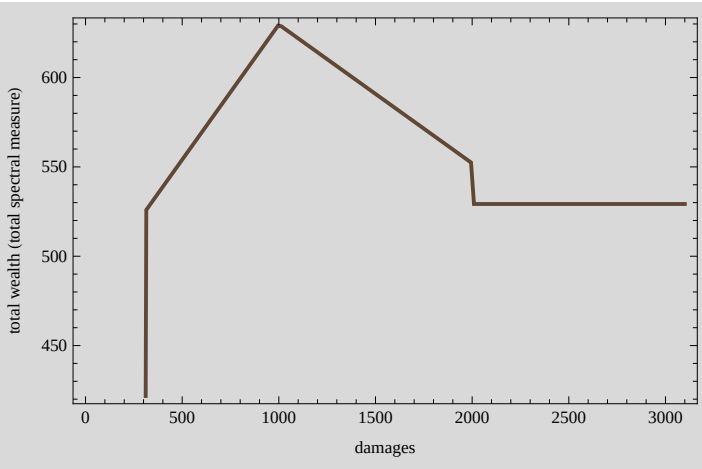
```
DynamicModule[{chigh = 2000, clo = 300, kB = 0.01`, kS = 0.01`, price = 700,
  visualize = "total", v = 1000, zB = 1, zS = 1, α = 0.9`}, Switch[visualize,
  "weightings", Row[
   {With[{bw = PDF[TruncatedDistribution[{0, zB}, ExponentialDistribution[kB]], p]},
     Plot[bw, {p, 0, 1}, ImageSize → 250, PlotStyle → ColorData[5][1]]],
    With[{bw = PDF[TruncatedDistribution[{0, zS}, ExponentialDistribution[kS]], p]},
     Plot[bw, {p, 0, 1}, ImageSize → 250, PlotStyle → ColorData[5][1]]]}],
  _, Plot[With[{bs$ = fN[{TruncatedDistribution[{0, zB}, ExponentialDistribution[kB]],
       TruncatedDistribution[{0, zS}, ExponentialDistribution[kS]]}, 2,
      probs[α], outcomes[d, d, price, clo, chigh, v]]}, Switch[visualize,
    "buyer", bs$[[1]],
    "seller", bs$[[2]],
    "total", Total[bs$]]], {d, 0, 3100}, Axes → False, Frame → True,
   FrameLabel → {"damages", "total wealth (total spectral measure)"},
   PlotStyle → Directive[{Thick, Switch[visualize,
     "buyer", ColorData[5][1],
     "seller", ColorData[5][2],
     _, ColorData[5][3]]}], MaxRecursion → 2]]
```

Out[10]=



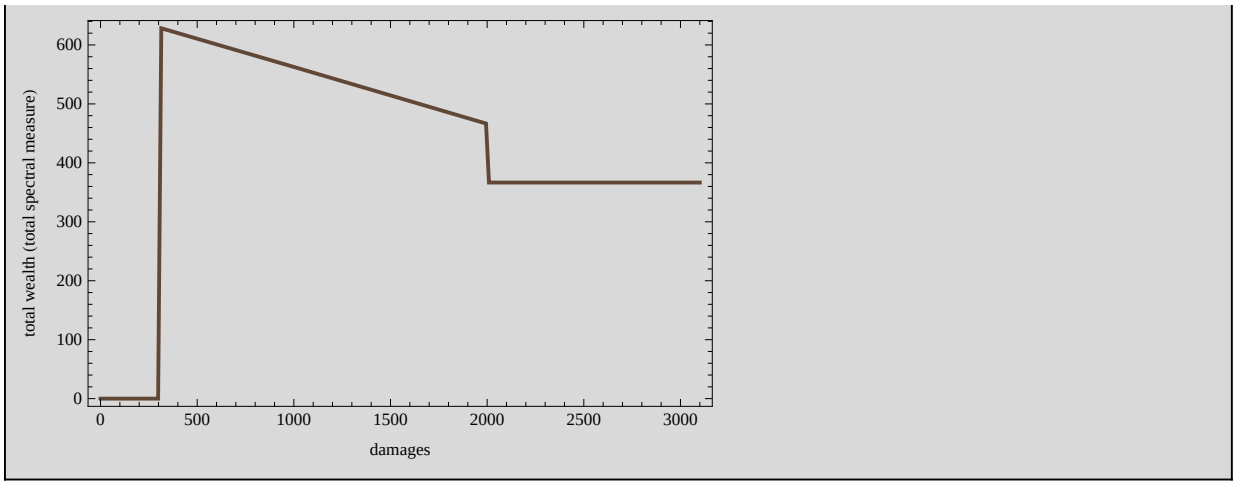
The graphic below shows a scenario in which the buyer is more risk averse than the seller. Now, total wealth is maximized when damages are $1, 000, the value of performance to the buyer. The risk averse buyer has, as it were, fully insured against loss.

Out[11]=



The graphic below shows a scenario in which the buyer is less risk averse than the seller. Now, total wealth is maximized when damages are about $300, the cost of low performance. The risk averse seller has now insulated itself from significant liability in the event the cost of performance ends up being high.

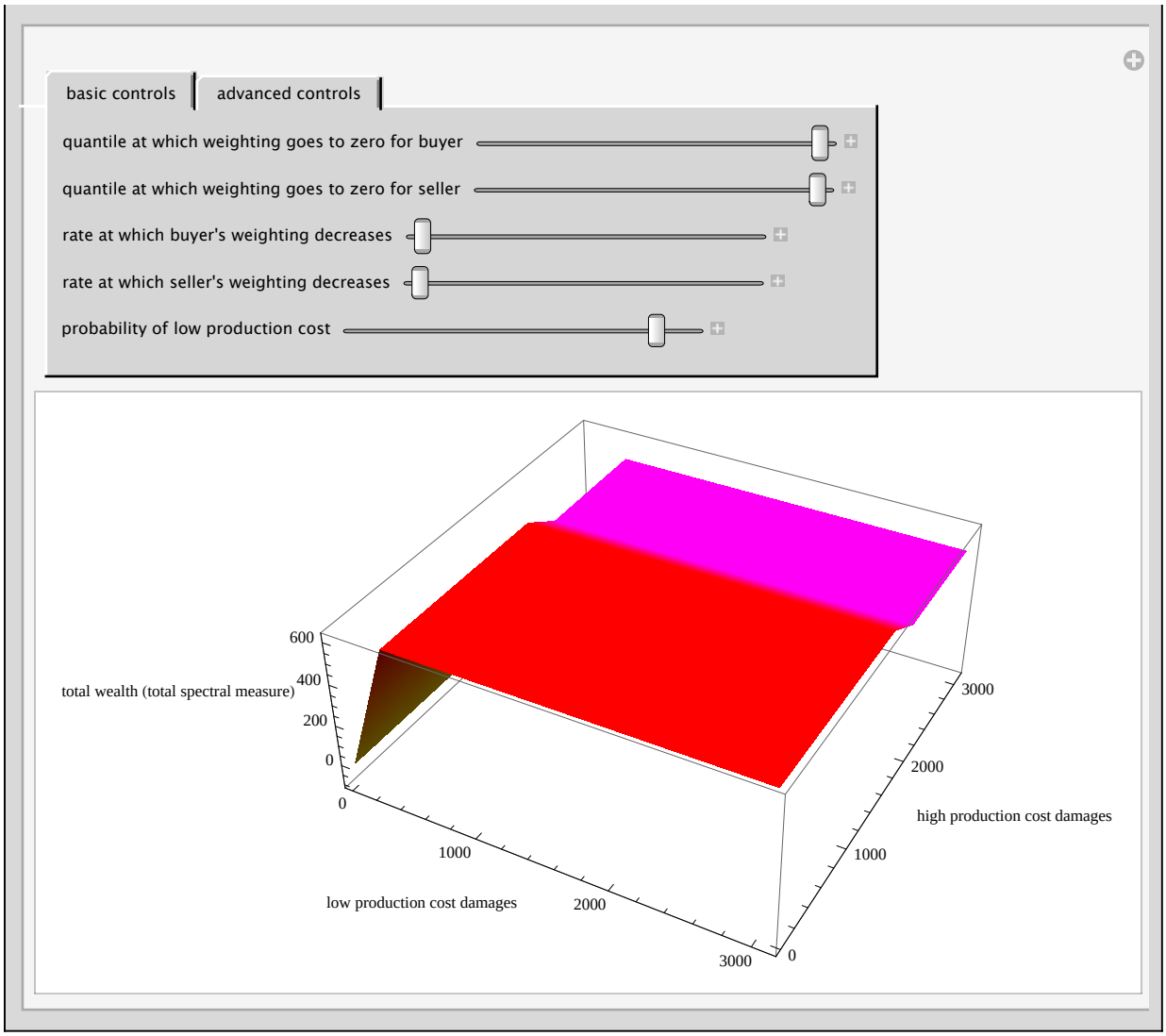
Out[12]=



Cost Variant Damages

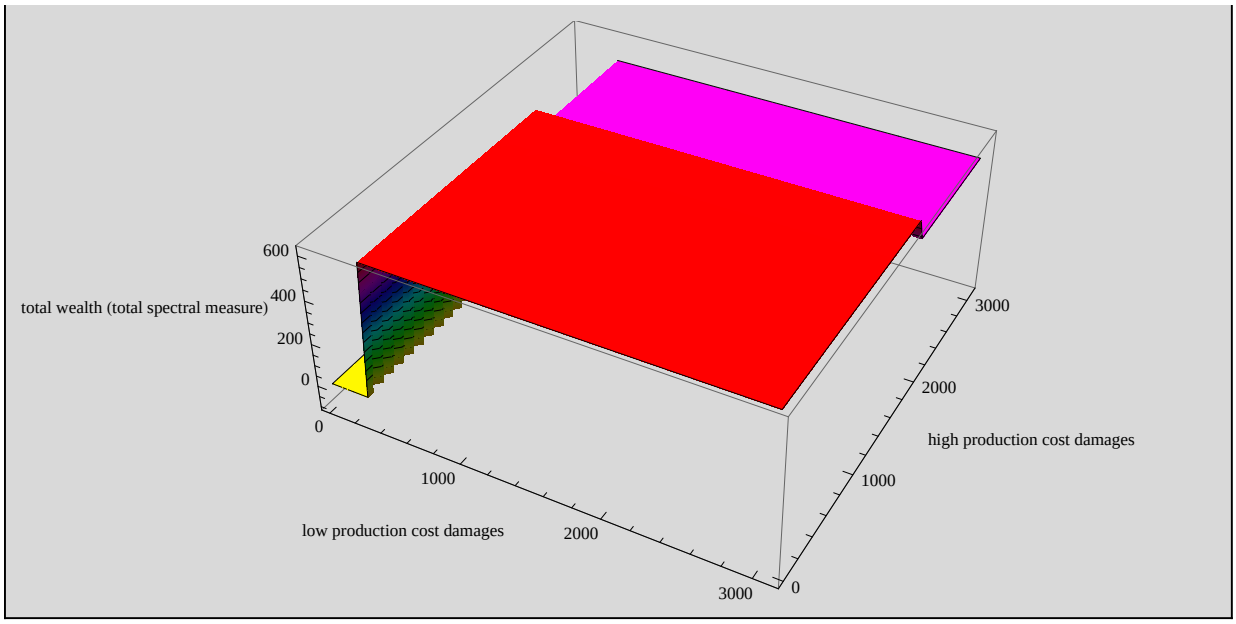
I can also produce a Manipulate in which the damage awards are permitted to vary depending on whether production costs were high or low. It could take a minute or so to draw the picture here depending on the speed of your computer, even more if you select “Quality” in the advanced setting. Be patient. If anyone has ideas how this computation could be accelerated, please contact me.

Out[13]=



Here' s a snapshot of some sample output.

Out[14]=



I actually think you can learn a lot about contract law from studying this simple model and using the interactive models presented here. E-mail me interesting outputs and your comments on their implications.