We’re here to talk about what changes we made to the original simplegame code, what experiment we ran, and the results of our experimentation.

We focused on the orders the players made based on the condition of their backlog and inventory. So, our first change was to change the myOrder equation. Originally, the command said that no matter what, myOrder is equal to demand + 1. We changed it so that only if the player has a backlog is myOrder demand + 1. Otherwise, the players just meet demand.

This is a graph of the original retailer cost, backlog, and inventory. We looked at the factory graphs and they follow the same pattern, so we’re going to look at the changes in retailer only. In any case, this is the graph produced by the simplegame code we were given. You can see that retailer cost and inventory start increasing and continue to do until the end of the time period.

Now this is a graph of the same parameters after the change in code. You can see that cost reaches an initial peak that is greater than the original code, but then it levels out, so the cumulative cost is still much less after the change in responses to demand based on backlog.

Then we started playing around with a combination of what we’ve already done and a response to existing inventory as well. We kept the “if there’s backlog, order demand + 1” command, and added that if the onhand number is greater than TotalDemand, then myOrder would be one less than demand. This way, the players can get rid of extra inventory.

Here is the graph produced from that change in code. There are oscillations in all three parameters. But the total stays mostly the same. Also, backlog goes oscillates from non-zero to zero, which doesn’t happen in the original.

We also tried to adjust orders by three: that is, have demand + 3 when there is backlog, and demand – 3 when there is extra inventory. As you can see, there is even greater oscillation here, and it is not preferable because the peak cost is greater, while the oscillations center around a value above ten, which means the cumulative cost is greater in this iteration.

So, basically, what we learned was that it’s better to just compensate for backlog and not worry about inventory. Because the two combined causes oscillations that at best center around the previous average.