Finance Project 1 - Linear Programming

Aidan Cremins, Peyton Lewis, Mark Moreno, Joe Morris

Fa22 OPTIMIZATION I

Table of Contents

I. Introduction

In this project, we are helping the firm fund its liabilities over the next eight years using a mix of bonds bought today and forwards bonds, which are bought at a future date and have no cost today. We seek to optimize the bond and forward bond configuration so that the cash inflows from the bonds in each of the eight years exactly match the cash outflows necessitated by the liabilities, without shorting any of the bonds. To make this inflow-outflow matching as optimal as possible for the firm, we want to minimize the upfront cost of buying the bonds in year 0. That is, we want to be able to finance all future liabilities by spending as little as possible today. We also want to finance the purchase of any forward bonds in a given period with the face values and coupon payments from that period so that we aren't carrying any excess cash balances between periods. Constructing a bond/forward mix in this way will allow the firm to not have to worry about buying assets in the future to cover these liabilities (assuming that they're fixed), and is hedged against interest rate changes.

II. Methodology

To begin this project, we read in the files for the liabilities and the given bonds to get a sense of which cash inflows we have at our disposal to match against the required cash outflows of the liabilities. Next, we set up a data frame which had a row for each bond-year combination for years 0 to 8. This allowed us to see for any given year what the price outflow for purchasing the bond would be (if applicable), what the inflow from coupon payments would be (if applicable), and what the inflow from receiving the face value back would be (if applicable). We then used this data frame to set up our objective function, which minimized the sum of the quantity of each bond times its ask price in year 0. Given that the price of bond 1 in year 0 is \$102, the price of bond 2 in year 0 is \$0 (since it's a forward bond), and the price of bond 3 in year 0 is \$99, the objective function was to minimize: $102x_1 + 0x_2 + 99x_3$…, etc. The complete objective function we sought to minimize is below:

$$102x_1 + 0x_2 + 99x_3 + 101x_4 + 98x_5 + 98x_6 + 0x_7 + 104x_8 + 100x_9 + 101x_{10} + 102x_{11} + 94x_{12} + 0x_{13}$$

In our code, we utilize negative cash flows to signal the payment for the bonds in period 0, or any subsequent period for the future bonds. Because of this, we effectively multiplied the objective function by -1, which had the effect of flipping the problem from a minimization problem to one in which we maximized. In effect, maximizing the negative cost was the same as minimizing the positive cash sum of the bond payments. This convention was applied so that having positive inflows of coupon payments and face values in future years would make more intuitive sense.

After establishing an objective function, we set up a series of 8 constraint equations that corresponded to the liabilities that needed to be exactly settled in years 1-8. In each of these equations, there was a decision variable (x_i) corresponding to the quantity of each bond (1-13) that satisfied that particular constraint equation. These constraint equations were essentially the rows of the cash flows data frame described earlier in that in each equation, the 13 decision variables were multiplied by that particular cash inflow (coupon or face value) or cash outflow (purchase price for futures) for that bond-year combination. The coefficients on these constraint equations were used to construct the 8 x 13 coefficient matrix "A" that was used in the optimization algorithm. The coefficients used in the matrix are shown in **Figure 1** below where each row represents the constraints for years one to eight and each column corresponds to the decision variables x_1 to x_{13}. Any row-column intersection represents the per-unit cash inflow or outflow for that particular bond in that particular year.

	Bond 1	Bond 2	Bond 3	Bond 4	Bond 5	Bond 6	Bond 7	Bond 8	Bond 9	Bond 10	Bond 11	Bond 12	Bond 13
Liability													
Year 1	105.00	-100.00	3.50	4.00	2.50	4.00	0.00	9.00	6.00	8.00	9.00	7.00	0.00
Year 2	0.00	103.00	103.50	104.00	2.50	4.00	-98.00	9.00	6.00	8.00	9.00	7.00	0.00
Year 3	0.00	0.00	0.00	0.00	102.50	4.00	2.00	9.00	6.00	8.00	9.00	7.00	-91.00
Year 4	0.00	0.00	0.00	0.00	0.00	104.00	102.00	9.00	6.00	8.00	9.00	7.00	3.00
Year 5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	109.00	106.00	8.00	9.00	7.00	3.00
Year 6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	108.00	9.00	7.00	3.00
Year 7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	109.00	7.00	3.00
Year 8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	107.00	103.00

Figure 1: Inputs to Coefficient Matrix of Original Bonds

The "b" vector (right hand side vector) of our constraint equations was an 8 x 1 vector storing the estimated liability amounts that the firm will face in the next 8 years. Lastly, the operators connecting the left and right sides of our constraint equations were all "=" since we must exactly match our net cash flows (inflows - outflows) with each year's liability amount without having a cash excess or deficit in any given year. Once the objective function and constraints were developed, we used Gurobi to find the optimal solution to our problem. The following code blocks were used:

```
def create_parameters(liabilities, bonds, df_year) :

    # create an empty matrix for the parameters length of liabilities times length of bonds
    A = np.zeros((len(liabilities), len(bonds)))
    # create an empty array for the objective function length of bonds
    obj = np.zeros(len(bonds) )

    # loop through each year of the liabilities
    for index, row in df_year.iterrows() :
        # fill in obj function
        if row['Year'] == 0 :
            obj[int(row['Bond']) - 1] = row['Price Outflow'] + row['Coupon Inflow'] + row['FV Inflow']

        # add price outflow + coupon inflow + fv inflow to A matrix
        if row['Year'] > 0 :
            A[int(row['Year']) - 1][int(row['Bond'] - 1)] = row['Price Outflow'] + row['Coupon Inflow'] + row['FV Inflow']

    # create an empty array for the right hand side of the constraints
    b = np.zeros(len(liabilities))
    # fill in the right hand side of the constraints with the liabilities
    for index, row in liabilities.iterrows() :
        b[index] = row['Liability']

    # create an array for the direction of the constraints
    sense = np.array(['='] * len(liabilities))

    return A, obj, b, sense
```

```
def solve_model(A, obj, b, sense) :
    finMod = gp.Model()
    finMod_x = finMod.addMVar(len(obj)) # tell the model how many variables there are
    # must define the variables before adding constraints because variables go into the constraints
    finMod_con = finMod.addMConstrs(A, finMod_x, sense, b) # NAME THE CONSTRAINTS!!! so we can get information about them later!
    finMod.setMObjective(None,obj,0,sense=gp.GRB.MAXIMIZE) # add the objective to the model...we'll talk about the None and the 0

    finMod.Params.OutputFlag = 0 # tell gurobi to shut up!!
    finMod.optimize()

    return finMod.objVal, finMod_x.x, finMod, finMod_con
```

Finding the optimal solution was certainly important, but we also had to check how useful it would be. We mentioned in the introduction that our optimal solution holds if future liabilities are fixed. However, it's very likely that our estimate for these future liabilities, especially those well into the future, will be slightly off the true value. Since we're setting up a portfolio of bonds/forwards that is supposed to exactly fund these future liabilities, we want to make sure that small changes in future liabilities due to estimation errors don't completely ruin our self-funding strategy. To do so, we can look at the shadow prices from changing each year's liability by $1. If these shadow prices are high, we would want to be very cautious and buy additional bonds in year 0, as this would mean that even a small increase in future liabilities would necessitate a much larger purchase in year 0 to still match cash flows. On the other hand, a small decrease in future liabilities would leave the firm with lots of excess cash in future years, which could be problematic because they would have needlessly tied money up in buying year 0 bonds.

Another test of our cash flow matching portfolio optimization strategy was to collect data on real bonds, not just the ones that we were given. We used the *Wall Street Journal* website to find data about bonds maturing on 9/15 in the future, as accessed on October 7, 2022. Since there were only 3 such bonds, we expanded our data set to also include bonds maturing at some point in August in the future, so as to replicate bonds maturing on 9/15 as closely as possible while also ensuring that we would receive cash flows by 9/15 to meet the firm's cash flow needs. In all years except for year 8, there were multiple bonds maturing, so there was some optionality that we could optimize. Our optionality was a bit limited in the sense that there were no forward bonds in the real bonds dataset, so we had to purchase all bonds up front in year 0.

III. Results

A. Original Bond Analysis

After using Gurobi to find the optimal number of each bond to purchase in year 0, we noticed that there were 5 bonds where it wasn't optimal to purchase any amount of them. The exact layout of the optimal quantities and associated price outflows of the requisite bonds are shown in **Figure 2** below.

Bond	Optimal Quantity	Year 0 Expenditure
1	6,522.49	$665,294.16
2	-	$0.00
3	12,848.62	$1,272,013.02
4	-	$0.00
5	15,298.32	$1,499,235.15
6	15,680.78	$1,536,716.03
7	-	$0.00
8	12,308.01	$1,280,032.71
9	-	$0.00
10	12,415.73	$1,253,988.48
11	10,408.99	$1,061,716.54
12	9,345.79	$878,504.67
13	-	$0.00

Figure 2: Bond Selection and Year 0 Expenditures (Original Bonds)

We then realized that all 3 of the forward bonds (bonds 2, 7, and 13) in our consideration set were among these 5 bonds that Gurobi suggested we buy $0 of. A bit skeptical at this point, we replaced the true coupon payment value for one of the forward bonds so that it would be *impossible* not to be included in an optimal solution (i.e. $10 coupon). As expected, the forward bond was part of the new optimal solution. In order to fund the forward bonds, we have to purchase more bonds in year 0 to ensure that we have the extra funds to buy the forward bonds in future years, which only makes sense if the forward bonds are a really good deal, like when we changed the coupon. We also computed the yield to maturity (YTM) on each of the 13 bonds and found that the forward bonds (highlighted in gray) had poorer relative yields compared to bonds with similar maturities. The YTM results are shown in **Figure 3** below.

Bond	YTM	Bond	YTM
1	2.94%	8	8.00%
2	3.00%	9	6.00%
3	4.03%	10	7.79%
4	3.47%	11	8.61%
5	3.21%	12	8.05%
6	4.56%	13	5.08%
7	3.05%		

Figure 3: Bond Options vs. YTM

Therefore, it appears as though the forward bonds that we were given just aren't as attractive as the other non-forward bonds and that there was no underlying issue with our constraint equations, as constructed.

The other non-optimal bonds were bonds that shared a maturity date with at least one other bond. It made sense to only purchase the best bond for each maturity date rather than purchasing some of an inferior bond. Ultimately, our optimal solution requires paying $9,447,500.76 in year 0 to purchase bonds in order to fund our liabilities that total $12,300,000. Thus, our approach saves the firm roughly $2,852,499.24, which justifies why we created the dedicated portfolio.

As a sanity check, we also wanted to check that the optimal solution from Gurobi was doing what we wanted it to: exactly match cash outflows from liabilities with cash inflows from bonds. The plot below (**Figure 4**) confirms that this is what happens with our optimal solution as the net cash flows balance out the liabilities (shown in red) for each of the eight years:

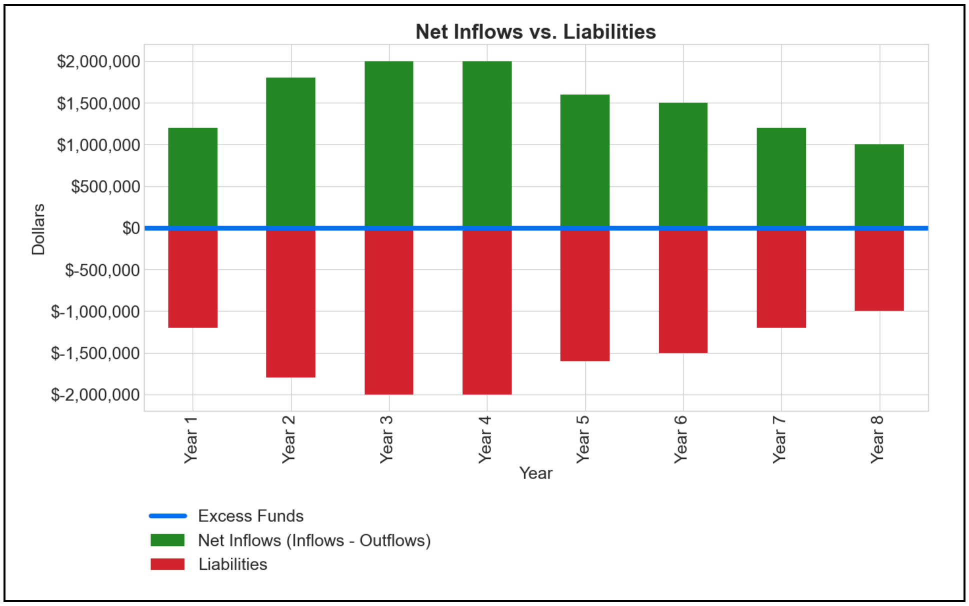

Figure 4: Net Inflows vs. Liabilities (Original Bonds)

B. Sensitivity Analysis

In order to investigate further, we performed sensitivity analysis on our optimized solution to see how altering the liability constraints affected our optimal value (expenditure today). Because the exact values of projected cash flows (i.e. liabilities) are not known by our firm with certainty, we compute the shadow prices associated with each of our eight constraints to determine how a $1 change in the expected future liabilities impacts the optimal cash expenditure in period 0, holding all else fixed. The results of this analysis are reproduced in **Figure 5** below.

	Lower Bound	Upper Bound	Range	Shadow Price	Liability
Year 1	$ 515,138.37	INFINITY	INFINITY	($0.97)	$ 1,200,000.00
Year 2	$ 470,168.21	$ 22,052,336.81	$21,582,168.60	($0.92)	$ 1,800,000.00
Year 3	$ 431,922.42	$ 31,062,103.33	$30,630,180.91	($0.91)	$ 2,000,000.00
Year 4	$ 369,199.31	$ 20,890,367.16	$20,521,167.85	($0.83)	$ 2,000,000.00
Year 5	$ 258,427.25	$ 10,751,333.43	$10,492,906.17	($0.65)	$ 1,600,000.00
Year 6	$ 159,101.43	$ 12,618,870.11	$12,459,768.68	($0.62)	$ 1,500,000.00
Year 7	$ 65,420.56	$ 11,972,949.71	$11,907,529.15	($0.53)	$ 1,200,000.00
Year 8	$0.00	$ 15,820,500.81	$15,820,500.81	($0.52)	$ 1,000,000.00

Figure 5: Sensitivity Analysis and Shadow Prices (Original Bonds)

As shown in the table, the shadow price for the first year's liability ($1,200,000) is -$0.97, indicating that a $1 increase in the expected future liability would increase the optimal cash outflow in period 0 by 97 cents (i.e. we need to spend 97 cents more today). The lower bound on this shadow price is seen to be $515,138.37 with an upper bound of infinity. Because the shadow price is non-zero, we know that the shadow price is binding within this range. Because the expected future liability could exist anywhere within this wide range and the shadow price would be the same, our company can be confident in the use of the -0.97 shadow price to model optimal cash outflows in the event that year 1 liabilities fluctuate in either direction. In the lower direction, liabilities could drop by approximately 57% ((1,200,000 -

515.138.37) / 1,200,000) before the shadow price is no longer a valid approximator of the resulting changes to year zero's optimal outflows. This appears to be a substantially large bound for the firm.

Looking at the other constraints, the shadow prices fall in magnitude as the years get further into the future, which is in line with expectation given the time value of money. A fluctuation in year 8 liabilities would alter the objective value by less (-$0.52) because at times farther in the future, a dollar is worth less and would therefore have a lessened impact on the year 0 optimal expenditures. Looking at the other years' shadow price information, it can be seen that in terms of the lower bounds, all liability estimates are relatively "safe" in that the lower bounds are in the order of hundreds of thousands, while the current estimated liabilities are in the order of millions. Unless the company severely overestimated its liabilities in these years, the shadow prices estimated should remain valid in the lower direction.

In the direction of the upper bound, a similar conclusion can be drawn. Looking at the firm's estimated liabilities, none exceed $2,000,000. The smallest upper bound appears for year 5 at $10,751,346.79, while the firm is currently estimating that year's liability to be $1,600,000. This would imply that the firm's actual year 5 liability would need to increase by roughly 571% in order for the shadow price to no longer be valid within this range. Given this information, we conclude that with a high degree of confidence, the shadow prices computed will be valid for the firm's operations within the next eight years. We used the code below to get the shadow prices and their bounds:

```python
# get the lower and upper bounds where the shadow prices are valid
low = [con.SARHSlow for con in model_con]
high = [con.SARHSup for con in model_con]
# get the shadow prices
pi = [con.pi for con in model_con]

# create a dataframe of the shadow prices, lower bounds, and upper bounds
df_con = pd.DataFrame({'Low': low, 'High': high, 'Pi': pi, 'Actual': A @ x}, index=['Year {}'.format(i + 1) for i in range(len(low))])

# create a range column, upper bound - lower bound
df_con['Range'] = df_con['High'] - df_con['Low']
# rename the columns for clarity
df_con.rename(columns={'Pi': 'Shadow Price', 'Actual':'Liability'}, inplace=True)
```

C. Real Bond Analysis

Using the real bond information provided on the *Wall Street Journal* website (as accessed on October 7, 2022), we were able to test that our program was optimizing correctly by giving it a new set of potential bonds to work with while maintaining the company's original liability structure. As mentioned earlier, the selection of bonds was limited to those maturing in each future year within the range of August 15 to September 15 so as to best model the company's needs to meet its liability obligations. This created a table with 34 potential bonds for the firm to choose from, ranging over the next eight years. For every year except for year 8, there exists optionality in the bond choices. An important assumption made was that although these bonds have semi-annual coupon payments, we treat the bonds as though they will pay coupons annually to simplify the analysis.

From the real bond analysis, we found that we still could indeed cover future cash outflows from liabilities with the inflows from these real bonds. Like our original bond analysis, **Figure 6** shows that in each period, the net cash flows exactly match the company's projected liabilities. The accompanying optimal quantities of each bond are shown in **Figure 7** below.

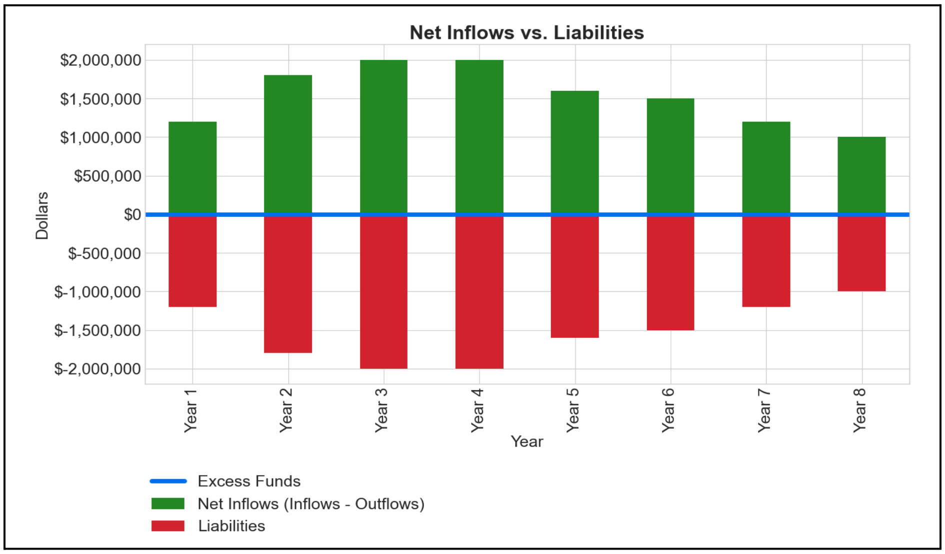

Figure 6: Net Inflows vs. Liabilities (Real Bonds)

Bond	Quantity	Optimal Expenditure	Bond	Quantity	Optimal Expenditure
1	0	$ -	18	0	$ -
2	0	$ -	19	0	$ -
3	7974.44	$ 807,444.45	20	17663.36	$ 1,594,118.29
4	0	$ -	21	0	$ -
5	0	$ -	22	0	$ -
6	0	$ -	23	0	$ -
7	0	$ -	24	0	$ -
8	0	$ -	25	13928.31	$ 1,533,367.75
9	0	$ -	26	0	$ -
10	0	$ -	27	0	$ -
11	0	$ -	28	0	$ -
12	0	$ -	29	13816.24	$ 1,482,068.15
13	14472.85	$ 1,335,988.58	30	0	$ -
14	0	$ -	31	0	$ -
15	0	$ -	32	0	$ -
16	16527.12	$ 1,769,228.30	33	11576.13	$ 1,101,052.41
17	0	$ -	34	9937.89	$ 777,361.49

Figure 7: Optimal Quantities and Associated Expenditures (Real Bonds)

However, this cash-flow matching came at a slightly higher cost at $10,400,629.42, meaning that we would only save $1,899,370.58, as opposed to the first solution. Also, like the analysis of the original bonds above, we only selected one bond per maturity year. Thus, although we had 34 bonds to choose from, we still only picked 8 to create our portfolio. Unlike the bonds given in our first section of analysis, we faced a limiting factor in year 8 with the real bonds, as there was only 1 bond with an August or September maturity up to 9/15/2030. This meant that in order to fund our liability in year 8, we had to buy bond 34 regardless of how good of an investment it was. This solution does have an added benefit that some of the bonds mature in August, so we could invest the cash flows in a short-term instrument (i.e. the money market) before we need the money on 9/15. Nevertheless, the additional proceeds from such short-term investments is extremely minimal.

Interestingly, we found a relatively similar result when looking at the YTM information on each bond in relation to its selection in the optimal solution or not as was found with the original bond analysis. When broken down into the maturity year groups, in 6 years out of 8 (not years 4 or 7), the bond selected was the one that had the highest yield, which is in accordance with expectation. The plot showing this information is below in **Figure 8**.

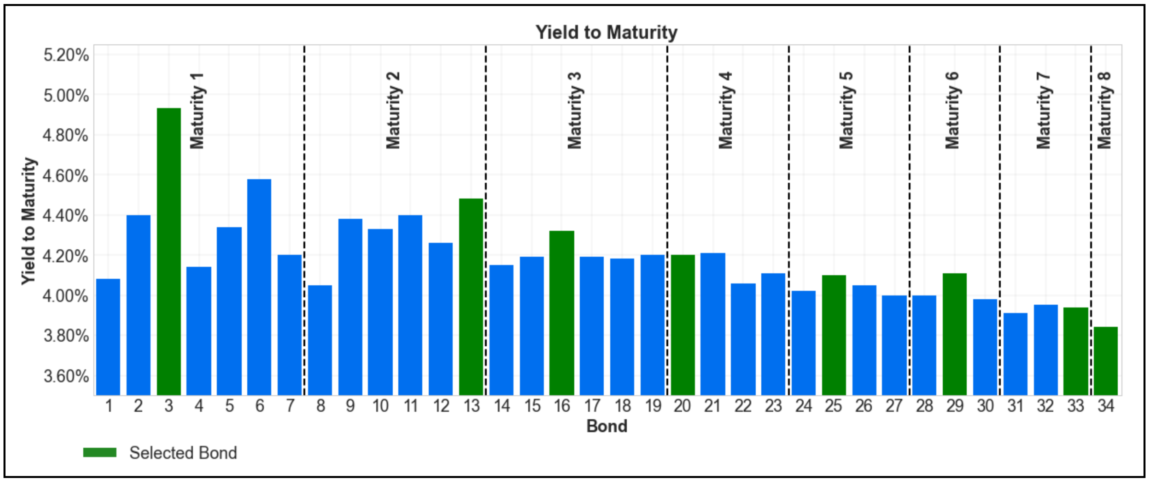

Figure 8: Maturity of Bond vs. YTM (and Selection)

IV. Conclusion

Overall, we were able to provide a way for the firm to cover its future liabilities for the next 8 years by paying a lump sum today to buy a portfolio of bonds/futures. This would be attractive to the firm to know that these liabilities will be covered, especially if there's some uncertainty regarding future cash flows from operations. Under such a scenario, the firm might not be able to pay back its liabilities and would either have to resort to costly short term financing or potentially even bankruptcy.

Our solution is a relatively robust one as the shadow prices for each given year were less than $1. This means that for say a $10,000 increase in liabilities, the optimal cost of bonds to buy in year 0 increases by less than $10,000. We certainly would recommend that the firm buy more bonds than optimal in year 0 to hedge against potential increases in liabilities. However, we don't have to worry about small increases in future liabilities suddenly ballooning the year 0 cost to an extreme amount as would've been the case if the shadow prices were greater than 1. Also, if future liabilities end up being less than estimated, the firm will have excess funds that year, but this is certainly better than facing a funding deficit. Additionally, we're confident that these shadow prices are stable enough to make such a conclusion above since the bounds for them are very wide and accommodate large deviations from the current liability estimates.

Our real bond analysis revealed that we could create a similar cash flow matching portfolio using actual U.S. Treasury Notes. However, the optimal solution was slightly more costly in year 0, which suggests that the given group of bonds that we initially optimized with are a good deal. Though we do receive the cash inflows slightly early, additional income from short-term investments would be so small that the original set of bonds still provides a dedicated portfolio at the lowest cost to the firm in year 0.