Advanced Forest Planning

Dr. Bettinger

Chaz Merritt

08/30/2024

Assignment 1- Linear Programming

I implemented the Linear Programming problem using a Python library for modelling LP’s called PuLP. The python file is attached, to run it yourself, assuming python is installed on your system, you can install the two dependencies, PuLP and Pandas, like this:

python -m pip install pulp

python -m pip install pandas

According to my formulation I achieved the optimal solution with a total objective value of 0, meaning the deviation from target was completely minimized for each period.

1. **My objective function was:**

prob += lpSum(deviation\_variables)

This simply means the sum of the “deviation variables” which were calculated as  
*for period in range(1, 9):  
 harvested\_volume = lpSum([  
 variables[f"stand\_{stand}period\_{period}"] \*  
 stands\_volume.loc[stands\_volume['Stand'] == stand, f'P{period}\_volume'].values[0]  
 for stand in range(1, 115)  
 ])  
  
 deviation = LpVariable(f"deviation\_period\_{period}", lowBound=0)  
  
 # Deviation constraints (absolute deviation)  
 prob += deviation >= harvested\_volume - target  
 prob += deviation >= target - harvested\_volume  
  
 deviation\_variables.append(deviation)*

To enforce the stand adjacency constraint, a binary set of decision variables was created in addition to the standard decision variables which are continuous from 0 to 40. The binary variables would allow me to enforce mutual exclusivity between adjacent stands. Then the rest of the constraints were calculated this way. Note that the age constraint adjusts for the change in age across periods using a 5-year increment. I did not have it written down if we used 5 years, 10 years, or something else between time periods.

1. **Constraints**

**# Can't harvest stands that are less than 35 years old based on stand data (stands age 5 years per period)**  
for index, row in stands\_volume.iterrows():  
 stand = int(row['Stand'])  
 age = row['Age']  
 for period in range(1, 9):  
 # Calculate the age of the stand at the beginning of the period  
 period\_age = age + (period - 1) \* 5  
 if period\_age < 35:  
 prob += variables[f"stand\_{stand}period\_{period}"] == 0  
 # Stand starts at less than 35 years old but becomes mature by the end of the time horizon  
 elif period\_age >= 35:  
 prob += variables[f"stand\_{stand}period\_{period}"] <= 40  
  
**# Enforce mutual exclusivity for adjacent stands**  
for index, row in adjacency.iterrows():  
 stand1 = row['stand']  
 stand2 = row['adjacent\_stand']  
 for period in range(1, 9):  
 prob += binary\_variables[f"y\_stand\_{stand1}period\_{period}"] + binary\_variables[f"y\_stand\_{stand2}period\_{period}"] <= 1  
  
**# Can only harvest each stand once across the entire time horizon**  
for stand in range(1, 115):  
 prob += lpSum([binary\_variables[f"y\_stand\_{stand}period\_{period}"] for period in range(1, 9)]) <= 1

1. **Accounting Rows**

I did not use Lindo systems, so the concept of “accounting rows” does not apply directly to my python implementation I don’t believe. However, the problem is identical. I calculated my deviations using the absolute difference instead of the square simply because it should be the same and I was having issues with maintaining linearity in PuLP. At any rate, this is how I calculated my deviation variables:

deviation\_variables = []  
  
for period in range(1, 9):  
 harvested\_volume = lpSum([  
 variables[f"stand\_{stand}period\_{period}"] \*  
 stands\_volume.loc[stands\_volume['Stand'] == stand, f'P{period}\_volume'].values[0]  
 for stand in range(1, 115)  
 ])  
  
 deviation = LpVariable(f"deviation\_period\_{period}", lowBound=0)  
  
 # Deviation constraints (absolute deviation)  
 prob += deviation >= harvested\_volume - target  
 prob += deviation >= target - harvested\_volume  
  
 deviation\_variables.append(deviation)

1. **Solution**

Lastly, a solution to the problem is presented on the last pages in the form of my abridged final output. This prints the values for the decision variables after running the solver, which converged to the optimal solution in a tenth of a second. Feel free to check the work and see that every stand for a given period does in fact add up to the target value of 11721.

Here is my abridged final output:

Optimal solution found.

stand\_2period\_1: 532.0 board feet

stand\_10period\_1: 600.0 board feet

stand\_13period\_1: 612.0 board feet

stand\_14period\_1: 848.0 board feet

stand\_19period\_1: 1116.0 board feet

stand\_37period\_1: 380.0 board feet

stand\_39period\_1: 680.0 board feet

stand\_54period\_1: 413.0000056 board feet

stand\_62period\_1: 644.0 board feet

stand\_69period\_1: 1156.0 board feet

stand\_81period\_1: 672.0 board feet

stand\_84period\_1: 1224.0 board feet

stand\_104period\_1: 1112.0 board feet

stand\_107period\_1: 856.0 board feet

stand\_111period\_1: 876.0 board feet

stand\_28period\_2: 344.0 board feet

stand\_38period\_2: 680.0 board feet

stand\_44period\_2: 1180.0 board feet

stand\_49period\_2: 1140.0 board feet

stand\_52period\_2: 620.0 board feet

stand\_67period\_2: 996.0 board feet

stand\_68period\_2: 553.0000025999999 board feet

stand\_72period\_2: 1176.0 board feet

stand\_89period\_2: 1032.0 board feet

stand\_92period\_2: 1332.0 board feet

stand\_95period\_2: 1160.0 board feet

stand\_99period\_2: 436.0 board feet

stand\_100period\_2: 1072.0 board feet

stand\_11period\_3: 1480.0 board feet

stand\_33period\_3: 1200.0 board feet

stand\_42period\_3: 1288.0 board feet

stand\_43period\_3: 1328.0 board feet

stand\_45period\_3: 1012.0 board feet

stand\_51period\_3: 1140.0 board feet

stand\_70period\_3: 1328.999994 board feet

stand\_74period\_3: 1528.0 board feet

stand\_112period\_3: 1416.0 board feet

stand\_22period\_4: 2172.0 board feet

stand\_25period\_4: 1700.9999952 board feet

stand\_48period\_4: 2168.0 board feet

stand\_55period\_4: 1348.0 board feet

stand\_73period\_4: 860.0 board feet

stand\_102period\_4: 1372.0 board feet

stand\_109period\_4: 2100.0 board feet

stand\_16period\_5: 436.0 board feet

stand\_24period\_5: 737.0000054 board feet

stand\_29period\_5: 428.0 board feet

stand\_31period\_5: 972.0 board feet

stand\_46period\_5: 1968.0 board feet

stand\_50period\_5: 1740.0 board feet

stand\_60period\_5: 2376.0 board feet

stand\_75period\_5: 1792.0 board feet

stand\_91period\_5: 464.0 board feet

stand\_94period\_5: 808.0 board feet

stand\_4period\_6: 2376.0 board feet

stand\_6period\_6: 680.0 board feet

stand\_8period\_6: 860.0 board feet

stand\_26period\_6: 348.0 board feet

stand\_56period\_6: 2480.0 board feet

stand\_63period\_6: 712.0 board feet

stand\_71period\_6: 1136.0 board feet

stand\_82period\_6: 920.0 board feet

stand\_93period\_6: 604.0 board feet

stand\_96period\_6: 464.99999900000006 board feet

stand\_105period\_6: 520.0 board feet

stand\_110period\_6: 620.0 board feet

stand\_17period\_7: 876.0 board feet

stand\_27period\_7: 1512.0 board feet

stand\_35period\_7: 884.0 board feet

stand\_65period\_7: 1460.0 board feet

stand\_76period\_7: 1960.0 board feet

stand\_80period\_7: 616.9999988000001 board feet

stand\_87period\_7: 1536.0 board feet

stand\_108period\_7: 2876.0 board feet

stand\_12period\_8: 1072.0 board feet

stand\_20period\_8: 1224.0 board feet

stand\_41period\_8: 1548.9999839999998 board feet

stand\_78period\_8: 1500.0 board feet

stand\_90period\_8: 1268.0 board feet

stand\_106period\_8: 2188.0 board feet

stand\_113period\_8: 2920.0 board feet