TECHNICAL REPORT

1. Introduction

The project aims to maximize profit for Metal Co. based on their weekly anticipated demand and production capacity. Daily operation includes processing up to 500 tons of steel to manufacture six different products. The products are shipped out on the same day. Any products that are put on backorder or left in inventory are associated with additional costs.

By optimizing the production schedule and inventory policy for Metal Co. using PuLP, a Python-based linear programming modeler, the projected weekly profit increases by 12.8% from 2 million to 2.256 million.

2. Assumptions

The start of the production cycle is Monday and the factory is open all days of the week. The factory produces the products during the day. Unless there is a need for inventory, all products are shipped out by day's end.

3. Method

We used PuLP as the linear programming modeler and Python Pandas to create a dataframe from the data in the Excel file. There are 168 variables and 103 constraints.

3.1 Decision Variables

The unit of all variables is **tons of steel**.
`p` ranges from 0 to 5, being Nails, Screws, Pipe, Flashing, Rebar, and Conduit respectively.
`d` ranges from 0 to 6, being Monday, Tuesday, Wednesday, Thursday, and Friday respectively.

Variable	Meaning
`produced[p,d]`	The amount of product `p` produced on day `d`
`inventory[p,d]`	The amount of product `p` put in inventory on day `d`
`backorder[p,d]`	The amount of product `p` backordered on day `d`
`sold[p,d]`	The amount of product `p` sold (i.e. shipped out) on day `d`
`inventory[p,d-1]`	The amount of product `p` coming from inventory on the day before day `d`
`backorder[p,d-1]`	The amount of product `p` used to fulfill backorder of the day before day `d`

3.2 Parameters

The unit of all parameters is **per ton of steel**.

`sellprice[p]`	The selling price per ton of product `p`
`mfgcost[p]`	The manufacturing cost per ton of product `p`
`inventorycost`	The cost per day left in inventory
`materialcost`	The price of raw steel

3.3 Constraints

- Each day, the company can process up to 500 tons of steel, which means the sum of the amount of all products `p` on day `d` is less than or equal to 500. The code below shows these constraints for all 7 days of the week.

```
for d in range(0, len(days)):
    my_lp_problem += pulp.lpSum([produced[p,d] for d in products]) <= 500
```

- Each day, the amount of the product sold cannot exceed the demand of that product. The code below shows this constraints for 6 products in all 7 days of the week.

```
for p in range(0, len(products)):
    for d in range(0, len(days)):
        my_lp_problem += sold[p,d]  <= demands[p][d]
```

- Each day, the amount of product produced that day + the inventory from the previous day + the backorder of that day must equal to the amount of product sold that day + the amount put in inventory that day + the amount of product that fulfills the backorder of the previous day. The code below shows these constraints for 6 products in 6 days (Tuesday to Sunday) of the week. The code for Monday is essentially the same, but it needed to be tweaked a little bit (shown in the code file) because PuLP doesn't take `d-1` if `d` starts from 0.

```
for p in range(0, len(products)):
    for d in range(1, len(days)):
        my_lp_problem += produced[p,d] + inventory[p,d-1] +
backorder[p,d] == sold[p,d] + inventory[p,d] + backorder[p,d-1]
```

- There is no ending inventory, which means all of the inventory for Sunday (d=6) is 0.

```
for p in range(0, len(products)):
    my_lp_problem += inventory[p,6] == 0
```

- There is no backlog at the end, which means all backorder for Sunday (d=6) is 0.

```
for p in range(0, len(products)):
    my_lp_problem += backorder[p,6] == 0
```

3.4 Objective Function

The objective is to maximize profit for Metal Co. The profit for each week is a summation of daily profit, which comprises the income of six products sold, subtracted by the costs of manufacturing and raw materials, the cost of keeping items in inventory, and the cost of having backorders.

```
my_lp_problem += pulp.lpSum([sold[p,d]*sellprice[p] for p in products for d in
days])
  - pulp.lpSum([produced[p,d]*mfgcost[p] for p in products for d in days])
  - inventorycost*pulp.lpSum([inventory[p,d] for p in products for d in days])
  - pulp.lpSum([backorder[p,d]*0.02*sellprice[p] for p in products for d in
days])
  - materialcost*pulp.lpSum([produced[p,d] for p in products for d in days ])
```

4. Analysis

The optimal solution provides a production schedule that meets the constraint of maximum production capacity (less than 500 tons processed per day), while raising productivity for other days of the week. Table 1 shows that Monday and Friday's total demands exceed production capacity by 140-150%, while other weekdays like Tuesday, Wednesday, and Thursday have less than 500 tons of demands, therefore we can leverage the lower demands to increase production on these days in order to supply the high demands on Monday and Friday.

Table 1. Anticipated Daily Demands (D) and Recommended Production (R) (Units: Tons)

	Monday		Tuesday		Wednesday		Thursday		Friday		Saturday		Sunday	
	D	R	D	R	D	R	D	R	D	R	D	R	D	R
Nails	50	50	40	40	60	60	40	90	50	0	30	30	30	30
Screws	90	90	70	70	80	80	70	70	70	70	40	40	40	40
Pipe	90	90	60	60	70	125	60	5	80	80	40	40	30	30
Flashing	70	70	60	60	80	80	40	160	120	0	60	60	30	30
Rebar	250	50	75	270	75	80	75	175	300	200	75	75	75	75
Conduit	150	150	0	0	0	0	0	0	150	150	0	0	0	0
Total	**700**	**500**	**305**	**500**	**365**	**425**	**285**	**500**	**770**	**500**	**245**	**245**	**205**	**205**

Table 2 shows that to fulfill the high demand for Friday, we can take advantage of the low demand on Thursday to produce an extra 50 tons of nails, 120 tons of flashings, and 100 tons of rebars and store them in inventory.

Table 2. Inventory Schedule

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Nails	0	0	0	50	0	0	0
Screws	0	0	0	0	0	0	0
Pipe	0	0	55	0	0	0	0
Flashing	0	0	0	120	0	0	0
Rebar	0	0	0	100	0	0	0
Conduit	0	0	0	0	0	0	0

From Table 3, for a high demand day like Monday and it is the start of the production cycle, backorders are inevitable, we recommend prioritizing rebar to be in backorder due to its lowest selling price compared to other products. This will minimize the cost associated with backordering. As a result, we will need to backorder 200 tons of rebars for Monday, and 5 tons of rebars for Tuesday.

Table 3. Backorder Schedule

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Nails	0	0	0	0	0	0	0
Screws	0	0	0	0	0	0	0
Pipe	0	0	0	0	0	0	0
Flashing	0	0	0	0	0	0	0
Rebar	200	5	0	0	0	0	0
Conduit	0	0	0	0	0	0	0

5. Recommendation and Future Work

5.1 Recommendation

I recommend the following strategies to maximize profit for Metal Co.:

- Metal Co. will gain an optimal profit of $2,256,410 for the anticipated week, a 12.8% increase compared to past operations, if they follow the production and inventory schedule laid out in Table 1
- Utilize resources on low-demand days to fulfill orders on high-demand days
- On high-demand days, prioritize producing other products over rebar, and put rebar on backorder to minimize the cost of backordering

5.2 Future Work

Since anticipated demands could change during the week, more future work can be done to predict daily demand (instead of weekly) and run the optimization everyday to make a production schedule for the next day.