CS164 - Optimization Final Project

Minerva University

CS164 - Optimization Methods

Prof. J. Levitt

December 16, 2022

Table of Contents

Constrained Optimization & Linear Regression

Introduction

While Tesla's automotive manufacturing is impressive, the company is still evolving and consequently faces issues, including supply chain management. In 2016, Tesla planned to ramp up Model 3 electric car production to 500K vehicles annually by 2020. Considering that each vehicle has 3K parts, all of which need to be in the right place at the right time and procured for the right price at the right quality, it was not a surprise that the goal was not achieved immediately. This paper will quantify what is "right" by formulating an optimization problem that minimizes overall costs while helping Tesla manage its demand and supply curve.

Problem Description

After the robust ideation process (Appendix A), we formulate the problem that minimizes total costs associated with Model 3's production and distribution subject to the constraints:

- Logical constraints:
  - The amount of manufactured cars corresponds (≥) to the number of shipped cars.
  - The production capacity of each manufacturing plant must not be exceeded.
- Supply constraint:
  - Suppliers have enough raw materials to supply Model 3's manufacturing.
- Demand constraint:
  - Manufacturers produce enough cars to meet demand.

Satisfying these constraints, Tesla will not only find the overall costs of production and minimize them (i.e., first deliverable) but also will be able to order enough raw materials for manufacturers to produce the needed amount of cars (i.e., second deliverable).

The data will be taken from matrices created in Excel that show supplier stock, raw material costs and shipping, product requirements, production capacity, customer demand, production cost, and shipping costs (Appendix B). Matrices' columns and rows show the actual data about Tesla. We researched who are their main suppliers (e.g., BHP and Ganfeng Lithium Co.), what raw materials are needed for the production[1] and grouped them for simplification purposes[2], where Tesla's manufacturers are located (e.g., Fremont, US; Shanghai, China; Virginia, US), which countries buy Tesla in the biggest amount and what are the models that Tesla produces. As for the last one, we still focused only on Model 3 in our solution, but we left empty gaps for other models to have the possibility to expand the solutions that our code provides. The quantities itself was randomly assigned[3] due to the lack of publicly available data.

After we generate datasets, we transfer them to Python, transferring NaN values to 0 (Appendix C).[4] Then, we listed Decision Variables, using tools of OR Tools open library (Figure 1, Appendix D).

```
Manufactures:  ['Fremont, US', 'Shanghai, China', 'Virginia, Nevada']
Materials:  ['Metal', 'Plastics', 'Fur', 'Batteries Materials']
Suppliers:  ['BHP', 'Ganfeng Lithium Co.', 'Glencore', 'Modine Manufacturing Co.', 'Rohm And Haas Company']
Tesla Models:  ['Model 3', 'Model Y', 'Model X', 'Model S']
Countries Consumers:  ['USA', 'China', 'Europe', 'Other']
```

Figure 1. *The parameters of the chosen datasets.*

While these variables can be adjusted to find the combination that minimizes overall cost satisfying the constraint, possible confounding variables could include:

---

[1] We used the article about Tesla's life cycle.
[2] These included Metal, Plastics, Fur and Batteries Materials. The recent update showed that Tesla's leading supply chain issue was associated with computer chips. Thus, in the future model iteration, it would be great to specify computer chips as a separate raw material.
[3] We still tried to find some real data. For example, we calculated customer demand for Model 3 in different locations by finding overall Tesla demand in the USA, China and Europe in 2020. Then, we saw demand for Model 3 in Europe and used proportion to find demand in USA and China.
[4] In this assignment we focused on Model 3, but if we want to add models, we only need to modify pages "Product Requirements" and "Production capacity" in "CS164 - Final Project" sheet by additing other products.

- Changes in demand for Model 3: This can impact the number of cars to manufacture.
- Changes in the availability of labour: If there are shortages or excesses in the labour force available to manufacture and distribute Model 3s, it could impact the efficiency and cost of production and distribution.
- Changes in regulatory requirements: If there are changes in regulatory requirements related to the production or distribution of Model 3s, it could impact the cost and feasibility of the production and distribution process.

Solution Specification

The optimization problem in our scenario corresponds to the convex optimization, specifically the linear program. This is driven by the fact our cost function is linear and constraints are linear (which is convex by definition). The nature of the feasible region is driven by the constraints (as per the KKT conditions, the solution must lie on the vertex or edge of the constraints). As the feasible region is not empty, the optimal solution exists. Furthermore, as the problem is convex, the solution we find is an optimal solution, by definition.

Formulating a linear programming model using the variables above, we first state the objective function (Figure 2). Our task is to minimize the overall costs related to Model 3's production, so we consider the overall cost of materials and production (i.e., supplier stock * (raw material costs + raw material shipping costs), the overall production cost of each manufacture (i.e., # of cars manufactured * production cost) and overall shipping costs (i.e., customer demand * shipping costs).

```
# Objective function
cost = solver.Objective()

# Overall cost of materials and production
for manfufacture in Manufactures:
    for supplier in suppliers:
        for material in materials:
            cost.SetCoefficient(orders[(manfufacture, material, supplier)] ,
                                RMC.loc[supplier][material] + RMS.loc[supplier][manfufacture])

# Overal production cost of each manufacture
for manfufacture in Manufactures:
    for product in models:
        cost.SetCoefficient(production_volume[(manfufacture, product)], int(PC.loc[product][manfufacture]))

# Shipping cost to consumers
for manfufacture in Manufactures:
    for customer in consumers:
        for product in models:
            cost.SetCoefficient(delivery[(manfufacture, customer, product)], int(SC.loc[manfufacture][customer]))
```

Figure 2. *The objective function of the supply optimization problem.*

Then, we write constraints in the format that suits Ortools Linear Solver (Figure 3).

```
# Constraint 1: Amount of manufactured cars >= Number of shipped to the consumers
for product in models:
    for manfufacture in Manufactures:
        c = solver.Constraint(0, solver.infinity())
        c.SetCoefficient(production_volume[(manfufacture, product)] , 1)
        for customer in consumers:
            c.SetCoefficient(delivery[(manfufacture, customer, product)], -1)

# Constraint 2: Number of cars is enough for the demand (Lean Startegy)
for customer in consumers:
    for product in models:
        c = solver.Constraint(int(CD.loc[product][customer]),int(CD.loc[product][customer]))
        for manfufacture in Manufactures:
            c.SetCoefficient(delivery[(manfufacture,customer,product)], 1)

# Constraint 3: Suppliers can give materials needed for the production (i.e., have them in stock)
for supplier in suppliers:
    for material in materials:
        c = solver.Constraint(0, int(SS.loc[supplier][material]))
        for manfufacture in Manufactures:
            c.SetCoefficient(orders[(manfufacture, material, supplier)],1)

# Constraint 4: Manufactures have enough material to produce all models
for manfufacture in Manufactures:
    for material in materials:
        c = solver.Constraint(0,solver.infinity())
        for supplier in suppliers:
            c.SetCoefficient(orders[(manfufacture, material, supplier)],1)
            for product in models:
                c.SetCoefficient(production_volume[(manfufacture, product)], - PR.loc[product][material])

# Constraint 5: Manufactures capacities are not exceeded
for manfufacture in Manufactures:
    for product in models:
        c = solver.Constraint(0, int(PCAP.loc[product][manfufacture]))
        c.SetCoefficient(production_volume[(manfufacture, product)],1)
```

Figure 3. *The constraints present in our optimization problem.*

An optimal solution to this problem exists because the objective function and the constraints are well-defined and measurable. The problem has a clear goal and there are multiple possible solutions (i.e., different combinations of raw materials, suppliers' routes, etc.) that can be compared to determine the optimal solution.

Optimal Solution

The complete code can be found here.

Input:

```
# Determine the optimal overall cost
cost.SetMinimization()

if solver.Solve() == solver.OPTIMAL:
    print("Optimal Solution Found!")
print("Optimal Overall Cost: ", "%.2f" % solver.Objective().Value())
```

Output:

```
Optimal Solution Found!
Optimal Overall Cost:  26270.00
```

The optimal solution will help Tesla allocate a sufficient amount of money for Model 3 production.

Order for Suppliers

The application of this optimization problem goes beyond the optimal solution. The most important part is to know how this minimum budget is achieved: from which supplier Tesla needs to order materials, in which quantity, which manufacturer needs to produce cars, in which quantity and where to deliver them. Among all of these questions, we prioritize the first one, because the whole production is not possible until you have the resources to do it and because ram materials are the leading expense. We write the code that compares prices for every material from every supplier, checks the material's availability and produces the most optimal way of ordering the product (Figure 5, Appendix E).[5]

[5] **#contrained.**

```
Order for Supplier from Every Manufacture
Fremont, US :
   BHP :
         Metal : 20.00
         Plastics : 18.00
   BHP  Bill:  2600.00
   Ganfeng Lithium Co. :
         Metal : 10.00
   Ganfeng Lithium Co.  Bill:  850.00
   Glencore :
   Doesn't supply from this company.
   Modine Manufacturing Co. :
         Fur : 6.00
         Batteries Materials : 30.00
   Modine Manufacturing Co.  Bill:  3360.00
   Rohm And Haas Company :
   Doesn't supply from this company.
Shanghai, China :
   BHP :
   Doesn't supply from this company.
   Ganfeng Lithium Co. :
         Metal : 10.00
         Plastics : 6.00
   Ganfeng Lithium Co.  Bill:  1020.00
   Glencore :
   Doesn't supply from this company.
   Modine Manufacturing Co. :
         Fur : 2.00
         Batteries Materials : 10.00
   Modine Manufacturing Co.  Bill:  1360.00
   Rohm And Haas Company :
   Doesn't supply from this company.
Virginia, Nevada :
   BHP :
   Doesn't supply from this company.
   Ganfeng Lithium Co. :
         Metal : 5.00
         Plastics : 11.00
   Ganfeng Lithium Co.  Bill:  3990.00
   Glencore :
         Plastics : 10.00
         Fur : 7.00
   Glencore  Bill:  3430.00
   Modine Manufacturing Co. :
   Doesn't supply from this company.
   Rohm And Haas Company :
         Metal : 30.00
         Batteries Materials : 35.00
   Rohm And Haas Company  Bill:  7100.00
```

Figure 4. *The algorithm that compares prices for materials from different suppliers, checks the availability of materials and outputs the optimal solution.*

Conclusion

By ensuring that the right amount of raw materials and components are available, Tesla can avoid delays in production due to shortages. This can help the company meet its Model 3 production goals and avoid lost revenue due to missed sales opportunities. Additionally, by ensuring that manufacturing plants do not exceed their production capacity, Tesla can avoid the costs associated with overproduction, such as excess inventory and storage costs. To sum up, we solved the chosen optimization problem and analyzed the solution in the context of Tesla operations and their optimization. We ran the program several times with different input parameters to understand their effect on the solution and provided recommendations based on our conclusions.

Appendix

Appendix A

An optimization problem for Tesla's supply chain management might involve finding the most efficient and cost-effective way to source raw materials, manufacture and assemble components, and distribute finished products to customers. The objective of the problem would be to minimize costs and maximize profitability while meeting customer demand and maintaining high levels of quality and reliability.

One possible formulation of this optimization problem could be as follows:

- Minimize: The total cost of sourcing raw materials and components, including transportation, storage, and handling fees.
- Maximize: The overall efficiency of the Model 3 supply chain.

Subject to the following constraints:

- The number of raw materials and components needed for production must be available in sufficient quantities to meet demand.
- The production capacity of each manufacturing plant must not be exceeded.
- The logistics network must be able to deliver finished products in a timely manner.
- Raw materials, production, and transportation costs must be within the budget.
- The total cost of distributing finished products to customers, including transportation, warehousing, and fulfillment costs.
- Customer demand constraints, ensuring that sufficient quantities of finished products are available to meet demand.
- Quality and reliability constraints, ensuring that all products meet Tesla's high

standards for quality and reliability.

- Capacity constraints, ensuring that the supply chain has sufficient capacity to meet demand without exceeding available resources.

This optimization problem could be either convex or non-convex, depending on the specific details of the supply chain and the assumptions made about the cost and demand functions. In general, however, it is likely to be a complex and challenging optimization problem, requiring advanced mathematical techniques and computational tools to solve.

Appendix B

	Metal	Plastics	Fur	Batteries Materials	
BHP	20	20			
Ganfeng Lithiur	25	50			
Glencore		10	70	40	
Modine Manufacturing Co.			20	50	
Rohm And Haas	30			40	

	Metal	Plastics	Fur	Batteries Materials	
BHP	20	80			
Ganfeng Lithiu	25	75			
Glencore		60	210	70	
Modine Manufacturing Co.			190	50	
Rohm And Haa	15			60	

Supplier stock & Raw material costs

	Fremont, U	Shanghai, C	Virginia, Nevada	
BHP	20	40	200	
Ganfeng Lithium	60	20	190	
Glencore	300	40	80	
Modine Manufact	20	40	150	
Rohm And Haas C	200	50	70	

	Metal	Plastics	Fur	Batteries Materials	
Model 3	5	3	1	5	
Model Y					
Model X					
Model S					

Raw material shipping & Product requirements

	Fremont, U	Shanghai, C	Virginia, Nevada	
Model 3	6	2	7	
Model Y				
Model X				
Model S				

	Fremont, US	Shanghai, Chi	Virginia, Nevada	
Model 3	140	130	150	
Model Y				
Model X				
Model S				

Production capacity & Production cost

	USA	China	Europe	Other
Model 3	7	5	3	
Model Y				
Model X				
Model S				

	USA	China	Europe	Other
Fremont, US	50	70	30	
Shanghai, Chi	90	20	40	
Virginia, Neva	10	80	100	

Customer demand & Shipping costs

The full document can be accessed here.

Appendix C

The following code was inspired by Leyer (2022).

```
# Function to collect data from Excel
def collect(file, sheet):
  a = pd.read_excel(file, sheet_name = sheet, index_col=0)
  return a

# Documents' and sheets' names
file = "CS164 - Final Project.xlsx"
sheet1 = "Supplier stock"
sheet2 = "Raw material costs"
sheet3 = "Raw material shipping"
sheet4 = "Product requirements"
sheet5 = "Production capacity"
sheet6 = "Customer demand"
sheet7 = "Production cost"
sheet8 = "Shipping costs"

# We created an Excel doc with 8 matricies that shows relations between main variables
SS = collect(file, sheet1)
RMC = collect(file, sheet2)
RMS = collect(file, sheet3)
PR = collect(file, sheet4)
PCAP = collect(file, sheet5)
CD = collect(file, sheet6)
PC = collect(file, sheet7)
SC = collect(file, sheet8)

# Since we focus only on model 3, we don't put values for other models, so they have NaN values that we transfer to 0
SS = SS.fillna(0)
RMC = RMC.fillna(0)
RMS = RMS.fillna(0)
PR = PR.fillna(0)
PCAP = PCAP.fillna(0)
CD = CD.fillna(0)
PC = PC.fillna(0)
SC = SC.fillna(0)
```

Appendix D

```
# Listing Decision Variables (code is taken from OR Tools open database).
solver = pywraplp.Solver('LPWrapper', pywraplp.Solver.GLOP_LINEAR_PROGRAMMING)
orders = {}
production_volume = {}
delivery = {}
```

Appendix E

```
# Order for Supplier from Every Manufacture
print("\nOrder for Supplier from Every Manufacture")
for factory in Manufactures:
    print(factory,":")
    for supplier in suppliers:
        factory_cost = 0
        print("  ",supplier,":")
        for material in materials:
          costs = orders[(factory, material, supplier)].solution_value()
          if costs > 0:
            print("\t",material,":", "%.2f" % costs)
            factory_cost += orders[(factory, material, supplier)].solution_value() * RMC.loc[supplier][material]
            factory_cost += orders[(factory, material, supplier)].solution_value() * float(RMS.loc[supplier][factory])
        if factory_cost > 0:
          print("  ",supplier," Bill: ", "%.2f" % factory_cost)
        if factory_cost == 0:
          print("  ","Doesn't supply for this location.")
```

References

Supply Science. (2022, November 16). Supply Chain Optimization with Python. YouTube. Retrieved December 16, 2022, from https://www.youtube.com/watch?v=gF9ds3CH3N4

PK: An Excel Expert. (2021, February 14). Transport Cost Optimization Using Solver in Excel. YouTube. Retrieved December 16, 2022, from https://www.youtube.com/watch?v=RUKUq9670Wc

Zulfan. (2022, November 11). 07 Supply Chain Optimization. YouTube. Retrieved December 16, 2022, from https://www.youtube.com/watch?v=xHQ2ODZWTRk

Wike. (1999, February). Supply Chain Management Optimization. North Carolina State University. http://dspace.mit.edu/bitstream/handle/1721.1/9763/42800135-MIT.pdf?sequence=2&isAllowed=y

S. (2022, November 6). Tesla Supply Chain Issues and Lessons! - Supply Chain Game ChangerTM. Supply Chain Game ChangerTM. Retrieved December 16, 2022, from https://supplychaingamechanger.com/drawing-lessons-from-teslas-supply-chain-issues/

What's behind Tesla's supply chain woes? (n.d.). Supply Management. Retrieved December 16, 2022, from https://www.cips.org/supply-management/news/2017/december/whats-behind-teslas-supply-chain-woes/

Tesla Production, Deliveries Graphed Through Q3 2022: New Record. (2022, October 3). InsideEVs. Retrieved December 16, 2022, from https://insideevs.com/news/613845/tesla-production-deliveries-2022q3/

El Yasmine Ait Si Larbi, Abdelghani Bekrar, Damien Trentesaux, & Bouziane Beldjilali. (2012, June). MULTI-STAGE OPTIMIZATION IN SUPPLY CHAIN: AN INDUSTRIAL CASE STUDY. Retrieved December 16, 2022, from https://hal.archives-ouvertes.fr/hal-00728633/document

K. Leyer (2022). *GitHub - KarthikMurugadoss1804/Supply-Chain-optimization--Linear-Programming: The aim of the project is leverage linear programming in supply chain optimization problem*. GitHub. Retrieved December 16, 2022, from https://github.com/KarthikMurugadoss1804/Supply-Chain-optimization---Linear-Programming