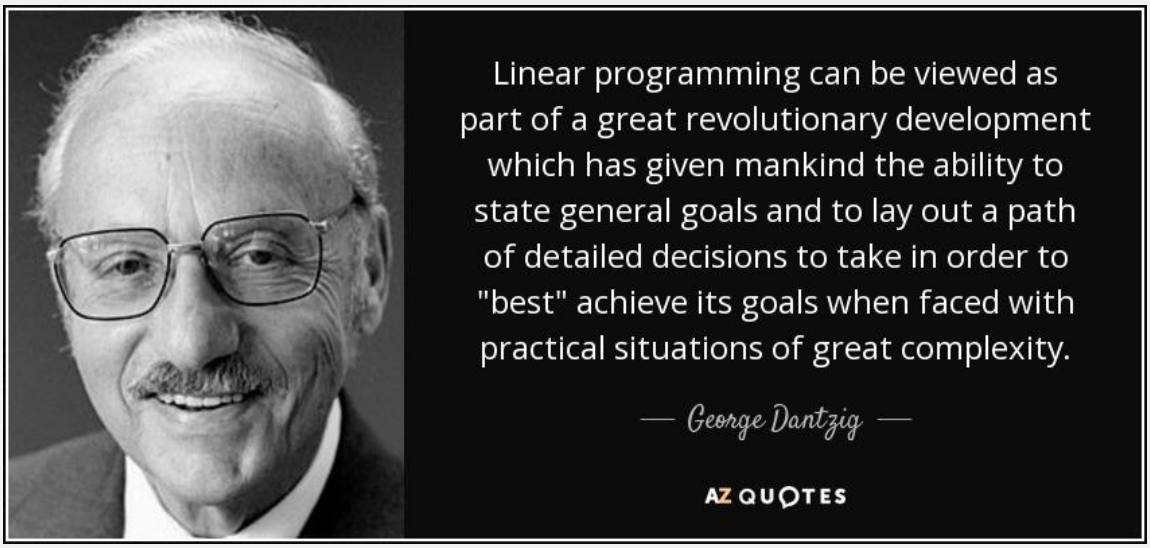

Deterministic Models

SUPERCHIP OPTIMIZATION

Oluwasegun Adegoke(G01368958) and Saiphani Chandra Vuppala (G01440607)
OR-541 | 30TH April, 2023

INTRODUCTION

Superchip is Virginia-based computer chip manufacturer, superchip has five manufacturing facilities located in Alexandria, Richmond, Norfolk, Roanoke and Charlotteville. Superchip computer chips portfolio consist of 30 chip types, each facility can produce all 30 different chip types. However, each facility has different production capacity levels. Further, because of each facility having different equipment and costs for set-up processes varying between facilities, the cost of manufacturing each chip is facility dependent. As part of its sales and distribution system, Super Chip distributes to 23 sales regions across the U.S. Due to variations in shipping distances and shipping material requirements, different shipping costs are associated with shipping different computer chips from each production facility to each sales region.

Superchip provided an excel file containing data on, production capacity at each facility, demand for each computer chip in each sales region in the fiscal year, costs for shipping each computer chip from each facility to each sales region and production costs for each computer chip at each facility.

Super Chip is seeking advice on how to perform their manufacturing and distribution operations for the upcoming financial year. As part of this advice, Super Chip would like to examine specific strategic questions.

- Suggest a different production policy, how does it compare to the current policy in terms of costs?
- Super Chip is considering expanding the production capacity at a single facility by purchasing additional equipment. They are looking for a recommendation on which facility to invest the capital?
- It is estimated that next year's demand is going to increase by 10% across all the sales regions. Does Super Chip have sufficient capacity to handle the estimated increase in demand?
- Super Chip is evaluating new manufacturing technologies. It is estimated that one of these new technologies could reduce production costs for all the chips by 15%. If Super Chip was to evaluate this new manufacturing technology in one of its facilities, which facility should receive the new technology?

This project objective was to provide a memo for the superchip management team based on the recommendation on how Superchip can optimize their manufacturing and distribution operations to improve their financial performance.

REPORT STRUCTURE

This report began with an introduction that provides an overview of Superchip's current manufacturing and distribution operations, as well as the purpose of the report. The next section provides background information on the Excel file provided by Superchip. The optimization model developed for Superchip is then discussed in detail, including an analysis of cost savings achieved through production and distribution cost reduction. The results section explains how chip types were allocated to facilities based on comparative advantage and offers recommendations for specific strategic questions posed by Superchip. A discussion of potential drawbacks or limitations to Superchip's current optimization strategies follows. Finally, the report concludes with a summary of key findings and recommendations for Superchip's manufacturing and delivery activities, as well as final thoughts on the effectiveness of the optimization model developed in this report.

METHODOLOGY

Current policy

In order to provide a recommendation for superchip management, it is important to know their current policy. We know each facility produces each of the 30 types of chips at levels that are proportional to the facility's total portion of production capacity. That is if facility x has y% of the total production capacity across all facilities, then facility x currently produces y% of every chip's total demand. In other to find the current production quantity and cost we used python to calculate.

First, we used the panda's library in python to create a data frame from the excel sheet provide. A Data Frame is a two-dimensional, size-mutable, tabular data structure in Python. It's like a spreadsheet or a SQL table, where data is organized in rows and columns. From this data frame we created three lists and four dictionaries. A dictionary is a data structure in Python that stores a collection of key-value pairs. Each key is associated with a value, and you can use the key to retrieve the corresponding value. Where these lists represent the facilities, chips, and regions while the dictionaries represented production capacity for each facility(prod_cap), the demand for each chip type in different regions(demand), the cost of producing chip types in the five different facilities(prod_costs) and the cost of shipping the chip types from the facilities to the different regions(shipping_costs).

	Facility	Computer Chip	Production Cost ($ per chip)
0	Alexandria	1	59.79
1	Alexandria	2	52.80
2	Alexandria	3	58.20
3	Alexandria	4	44.46
4	Alexandria	5	52.71
..	...	...	...
145	Charolottesville	26	46.06
146	Charolottesville	27	54.05
147	Charolottesville	28	58.42
148	Charolottesville	29	53.98
149	Charolottesville	30	50.48

Figure 1 Sample of the data-frame

Secondly, we created three new dictionaries representing the total demand for each chip type across all regions(total_demand), production quantity for each chip type in each facility(prod_qty) and quantity of each chip type shipped from each facility to each region(shipped_qty). These dictionaries were added to calculate the total cost of production. The total_demand dictionary was used to create the prod_qty dictionary by multiplying the proportional facility capacity to the demand for each chip.

Finally in order to find the total production cost, a summation of the prod_costs multiply by the prod_qty for all chip types in each facility. The result for the current policy production cost is $54,246,470.75 and that this information will serve as a baseline for comparing alternative production policies.

Alternative Policy

From the baseline we can start optimizing the current production and distribution policy, to optimize superchip operations we used a technique created by George Dantzig while in graduate school called linear programming. Linear programming (LP) is a mathematical optimization technique that is used to maximize or minimize a linear objective function subject to linear constraints. In LP, the objective function and the constraints are linear functions of the decision variables, which are the variables that we want to optimize.

We need to first formulate a mathematical model that captures the key elements of the problem, including production capacity, demand, shipping costs, and production costs. Decision variable is a variable that we want to optimize or decide about , this LP will use two decision variables representing superchip two operations; production and distribution of chips. Decision variable 1: No of chip i produced in facility j, Decision variable 2: No of chip i shipped from facility j to region k.

Decision variables are used to represent the quantities that we want to determine to achieve an optimal solution to a problem. These variables will be denoted by X_{ij} and Y_{ijk} where i = number of chips i.e., 1-30, k = number of sales regions i.e., 1-23, j = number of facilities i.e., 1-5.

Secondly, we need to formulate an objective function. The objective function is a mathematical expression that represents the goal or objective of the problem. It is a function of the decision variables that we want to optimize. Therefore, the objective function represents the quantity that we want to optimize or minimize. The objective function for this problem will be.

$$\text{Min}\, Z = \Sigma\Sigma(Pij * Xij\ ) + \ \Sigma\Sigma\Sigma(Sijk * Yijk)$$

Where Pij is the production cost of chip i in facility j (prod_costs) and Sijk is the cost of shipping chip type i from facility j to region k. By using this this objective function, we will be able to get the minimal cost of production and shipping to the sales region. The first part of objective function represents production cost, and the latter is shipping cost. The basic Feasible Solution will be is a solution that satisfies all the constraints of the linear programming problem, which will be the number of chips i to produce at each facility and the number of chips i to ship from each facility to each region.

Finally, we need to formulate the constraints for this LP. Constraints are restrictions placed on the decision variables to limit the feasible set of solutions. The feasible set of solutions is the set of all values of the decision variables that satisfy all the constraints. The first set of constraints we added was to limit the production of each facility to the facility production capacity. This constraint is the summation of each chip type produced in one facility is less than or equal to the facility production capacity. i.e., ΣX_{ij} i=1-23 for chip types <= Prod_cap for j=1-5 for facilities, this added up to five constraints. The second sets of constraints added was to ensure that the chips shipped meet the demand for all chip types. This constraint is the summation of chip type ship from all facilities is greater or equal the demand for the chip type. i.e., ΣY_{ijk} >= demand. The third set constraints ensure that each chip types from each facility to each region is less than or equal to the chip type produced in the facility. i.e., $\sum Y_{ijk} <= X_{ij}$ for all i, j. The final constraint is a non-negativity constraint that require the decision variables to be non-negative, such as $Y_{ijk} >= 0$ and $X_{ij} >= 0$.

Implementation: Gurobi and Python Code

Python is a high-level, interpreted programming language that was first released in 1991 by Guido van Rossum. It is widely used for a variety of purposes, including web development, scientific computing, data analysis, artificial intelligence, and more. Gurobi is a commercial optimization solver that is widely used to solve optimization problems in various fields, including operations research, supply chain management, finance, engineering, and more. It is developed and marketed by Gurobi Optimization, Inc [4]. Gurobi uses advanced algorithms and techniques to find solutions that optimize a given objective function subject to constraints.

We used the same dictionaries and lists from the current policy, then we imported gurobi.

"Import gurobipy as gp.

m = gp. Model ('Super chip Optimization')

m.modelsense = gp.GRB.MINIMIZE

m.update ()"

To define the decision variables in python, gurobi library provides a statement "addVar" and different type of variable type (continuous, integer, binary, Semi-continuous and semi-integer). We initial chose integer but couldn't run the shadow price code with integer variable, so we switched it to continuous variable to obtain an estimate for the shadow prices. The lower bound is "0" for the non-negativity constraint. To create the first decision variable X_{ij} we used two "for" loops, to iterate over the chips and facilities lists, we created the second decision variable Y_{ijk} using three "for" loops, to iterate over the chips, facilities, and regions list. We stored the decision variable in dictionaries.

"

```
# Creating decision variable x_ij
x = {}
for i in chips:
    for j in facilities:
        x[i, j] = m.addVar(lb = 0, vtype=gp.GRB.INTEGER, name=f'x{i}_{j}')
# Creating decision variable y_ijk
y = {}
for i in chips:
    for j in facilities:
        for k in regions:
            y[i, j, k] = m.addVar(lb = 0, vtype=gp.GRB.INTEGER, name=f'y{i}_{j}_{k}')
```

m.update ()"

Creating the objective function in python, gurobi provides a function called "setObjective", we separated the production and distribution decision variables in order to have the ability to separately print production and shipping cost. Quicksum is a function provided by the gurobipy that simplifies the process of summing up a collection of decision variables or expressions. We used two "for" loops to iterate over chips and facilities in both prod_costs and x dictionaries, multiply them then quicksum for production_cost. For shipping_cost we used three "for" loops to iterate over chips, facilities, and regions in both shipping_costs and y dictionaries, multiply them then quicksum.

" # Set objective function.

production_cost = gp.quicksum(prod_costs[i, j] * x[i, j] for i in chips for j in facilities)

shipping_cost = gp.quicksum(shipping_costs[i, j, k] * y[i, j, k] for i in chips for j in facilities for k in regions)

m.setObjective(production_cost + shipping_cost, gp.GRB.MINIMIZE)

m.update()"

Gurobipy provides a function in python to add constraints "m.addConstr". to adda the production capacity constraint we used a "for" loop to iterate over the x decision variable and prod_cap dictionaries, for every summation of j in the decision variable dictionary is less than or equal to value for j in prod_cap dictionary. The second constraint specifies that the sum of the decision variables y[i, j, k] for all facilities j should be greater than or equal to the demand[i, k] for that chip in that region. In other words, the demandConstraint sets a lower bound on the amount of chips that need to be shipped to each region, ensuring that the total demand for every chip in all regions is met. The third set of constraint a combination of chip i and facility j. We used quicksum to specifies that the sum of the decision variables y[i, j, k] for all regions k should be less than or equal to the decision variable x[i, j], which represents the production of chip i in facility j. In other words, the shippingConstraint ensures that the amount of chip i shipped from facility j to region k does not exceed the amount of chip i produced in facility j.

"# production capacity constraint.

c = {}

for j in facilities:

c[j] = m.addConstr(gp.quicksum(x[i, j] for i in chips) <= prod_cap[j])

 # demand constraint.

demandConstraint={}

```
for i in chips:
    for k in regions:
        demandConstraint=m.addConstr(gp.quicksum(y[i, j, k] for j in facilities) >= demand[i, k])


#ensuring the chip is produced
shippingConstraint={}
for i in chips:
    for j in facilities:
        shippingConstraint=m.addConstr(gp.quicksum(y[i, j, k] for k in regions) - x[i, j] <= 0)
m.update()”
```

Shadow price represents the change in the optimal objective function value for a small change in the right-hand side of a constraint. It can be interpreted as the marginal value of a unit increase in the constraint. Superchip requires a recommendation for expanding the production capacity at a single facility by purchasing additional equipment. In our Model we already have a constraint on production capacity so if we calculate the shadow price for a change in the right-hand side of the production capacity constraints. Gurobi provides a function to solve for shadow prices ".pi". To solve for this, we iterated of the production capacity constraints and printed the shadow prices for each facility.

```
“if m.status == gp.GRB.Status.OPTIMAL:
    for j in facilities:
        print(f'Shadow Price for Production Capacity Constraint {j} = {c[j].Pi}')
else:
    print('No feasible solution found.')”
```

Superchip estimates and increase in demand across all sales regions and want to know if they have sufficient capacity to handle the estimated increase in demand? If so, what are the associated costs for filling the new demand in comparison to this year's demand. To find this out we just needed to switch the demand dictionary used in the current policy and optimization model to represent the new demand. To do this we just multiply the current value of the dictionary buy "1.10" to represent a 10% increase in demand.

```
“ # creating a dictionary for a 10% increase in demand.
new_demand = {}
```

```
for key, value in total_demand.items():

   new_demand[key] = value * 1.1”
```

Super Chip is evaluating new manufacturing technologies. It is estimated that one of these new technologies could reduce production costs for all the chips by 15%. Super Chip wants a recommendation on facility should receive the new technology. To make this recommendation we wanted to know which facility has the highest production cost and then make the recommendation for this technology to be assigned to this facility to reduce the overall production cost. To do this we created a data fame then a dictionary from our recommended production policy. After doing this we calculated the production cost each facility from our recommended policy.

```
“df_production = pd.read_excel('chip.xlsx')

# creating a dictionary for our recommended chip_production.

quantity = {(row['Chip'], row['Facility']): row['Value'] for index, row in df_production.iterrows()}

# Creating a dictionary to store the total production cost for each facility based on our recommendation

new_new_cost = {}

for j in facilities:

   for i in chips:

      cost = 0

      if (i,j) in prod_costs.keys():

         costs = prod_costs[(i,j)]

         cost_prod = quantity[(i,j)] * costs

         cost += cost_prod

         new_new_cost[j] = cost

for j in facilities:

   print(f"The cost of production in {j} facility based on our recommendation is {new_cost[j]}")

# Creating a dictionary to store the total production cost for each facility if technology is implemented

new_reduced_cost = {}

for j in facilities:
```

```
    new_reduced_cost[j] = 0.85 * new_new_cost[j]

new_max_key = max(new_reduced_cost, key=new_reduced_cost.get)

print(f"We Recommended new technology go to {max_key} facility")"
```

RESULTS

Superchip's current production and distribution strategy to meet the current fiscal demand costs $63,941,054.15. Through Gurobi optimization, we were able to achieve a minimal production and distribution cost of $49,083,430.39 based on the current facilities' capacity and yearly demand. Most of the cost decrease came from the distribution cost, as Superchip's current policy doesn't consider shipping costs to different regions when allocating production. Additionally, the decrease in production cost came from allocating chip types to the facilities which can produce them most cost-effectively. Under our recommended policy, each facility won't produce all chip types. For example, only 11 chip types are produced in Alexandria, 11 chip types in Richmond, 5 chip types in Roanoke, 7 chip types in Norfolk, and only 2 chip types in Charlottesville. This policy means that each facility, based on its capacity, can focus on chips it can produce cost-efficiently, the optimal production cost is $49083430.39 and shipping cost is $1773750.

To recommend the facility that would provide the best value in increasing capacity, we sug gest investing in Richmond. Although Gurobi doesn't provide shadow prices for integer problems, w e noticed that when we switched the variable type to continuous, we obtained the same objective fun ction value and decision variable values. Based on this, we used the shadow prices from the continuo us variable to provide the recommendation. All but one constraint in the production capacity constrai nt set yielded shadow prices of zero. The constraint that represents the production capacity for Rich mond had a shadow price of -0.699, indicating that for every unit increase in production capacity in R ichmond, the total cost would decrease by $0.7.

Superchip does have the ability to fulfill a 10% increase demand, based on their current production policy which will cost $59,671,117.8 from this year $54,246,470.75 which is a 9% increase. But if our strategy is implemented the associated cost for a 10% increase will be $54,024,420.27 which will be an 8.47% decrease in cost price compared to the current. Superchip also wanted to know which facility needed new manufacturing equipment to reduce the cost of production by 15% at the facility. We chose the facility with the highest production cost, which is Alexandria.

DISCUSSION

The optimization model developed for Superchip was able to provide significant cost savings by reducing both production and distribution costs. The model was able to achieve this by allocating the production of chip types to the facilities that can produce them most cost-effectively. The approach of focusing on producing chips where facilities have a comparative advantage led to each facility specializing in a subset of the 30 chip types. This optimization approach resulted in a more efficient use of the production capacity and reduced the production costs.

Our recommended production policy would result in a different approach to production planning than what Superchip is currently using. Currently, each facility produces all 30 chip types

based on the demand at each sales region. In contrast, our recommended policy would allocate the production of each chip type to the facilities where they can be produced most cost-effectively.

In terms of capacity expansion, we recommend investing in the Richmond facility. This recommendation is based on the shadow prices obtained from the continuous optimization model, indicating that increasing production capacity at the Richmond facility would result in the greatest cost reduction. However, it is important to note that the shadow prices obtained from the continuous model may not necessarily be valid for the integer model. Therefore, Superchip may need to consider conducting a sensitivity analysis to assess the robustness of the recommendation.

Regarding meeting the estimated 10% increase in demand, Superchip has the capacity to fulfill the increased demand under both the current production policy and our recommended policy. However, our recommended policy would result in a lower cost compared to the current policy. This cost reduction could be significant for Superchip, as it would result in an 8.47% decrease in cost price.

Finally, we recommend that Superchip invests in the Alexandria facility to evaluate new manufacturing technologies. This recommendation is based on the fact that Alexandria has the highest production cost among the five facilities. Introducing a new manufacturing technology at Alexandria has the potential to reduce production costs by 15%, resulting in significant cost savings for Superchip. However, the company would need to consider the investment cost of introducing new manufacturing technology and the potential impact on the overall production capacity and cost structure.

In conclusion, Superchip can significantly improve their financial performance by implementing our recommended production policy, investing in the Richmond facility to expand production capacity, and investing in the Alexandria facility to evaluate new manufacturing technologies.

Memo

To: Super Chip management

From: Oluwasegun Adegoke *OA* and SaiPhani Chandra Vuppala *SPCV*

SUBJECT: Analysis of production and shipping costs

CC: Dr Hadi El Amine

We are writing to provide you with an analysis of the production and shipping costs of our computer chips. The data used in this analysis was obtained from Dr Hadi El Amine and the calculations were made using the Gurobi and Python programming language.

Your current production costs are broken down as follows:

- Production costs: The total production costs for all chip types and facilities are $54,246,470.75.

To optimize the production and shipping policies, we created a mathematical minimization model that calculates the optimal production quantity for each chip type in each facility, as well as the quantity shipped from each facility to each sales region. This model considers the production capacity of each facility, the demand for each chip type in each sales region, and the shipping costs for each chip type and facility.

Production Costs: Our model has revealed the cost for this new production strategy is $47,309,680.4. This represents a cost savings of 12.7% compared to your current production strategy.

Shipping Costs: We have also optimized the shipping strategy, resulting in a total shipping cost of $1,773,750. This represents a cost savings of 81.67% compared to your current distribution strategy.

Overall Cost Savings: By optimizing both our production and shipping strategies, we have achieved a total cost of $49,083,430.39. This is a cost savings of 23.24% compared to your current production and shipping strategies.

Next Steps: We recommend implementing our optimized production and shipping strategy that utilizes data from attached csv file, which details the production of each chip type in each facility. By integrating the information from these sheets, you can maximize cost savings and improve overall efficiency. We suggest implementing this strategy as soon as possible to start reaping the benefits.

Investment: We are aware that management has additional cash flow and want to expand the production capacity at a single facility. We recommend investing in expanding the production capacity at our Richmond facility. For every increase in chip production at this facility, there is a decrease of $0.7 in total cost. This cost savings can help improve our overall profitability.

Increase Demand: Superchip currently has the capacity to field chips for a 10 percent increase in demand, this will have an associated cost of $59,671,117.8. But if our strategy is implemented the associated cost for a 10 percent increase will be $54,024,420.27

New Technology: We are aware of a potential reduction in production cost from a new technology, we recommend this technology be used in the Alexandria facility because Alexandria has the highest production cost.