Department of Mechanical
& Industrial Engineering
Faculty of Engineering
& Architectural Science

Department of Mechanical and Industrial Engineering

Please select your current program below:

☐ **Industrial Engineering**

Course Number	IND604
Course Title	Operations Research II
Semester/Year	Winter 2023
Instructor	Aliaa Alnaggar
Section Number	1

FINAL REPORT

Report Title	5135 Creekbank Chips Factory Production & Shipping Problem
Group Number	
Submission Date	9th April, 2023
Due Date	9th April, 2023

Student Name	Student ID (xxxx1234)	Signature*
Rufat Akhmetov	xxxx42983	R.A
Raphael Matti	xxxx63798	R.M
Sheikh Abid Rahman	xxxx42494	S.A.R

(Note: Remove the first 4 digits from your student ID)

**By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work <u>will result in an investigation of Academic Misconduct and may result in a "0" on the work,</u> an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: http://www.ryerson.ca/senate/policies/pol60.pdf.*

Abstract

The main objective of this project is to utilize operations research optimization concepts and methods to determine the optimal solution to a problem defined that is related to processes of production in a potato chip manufacturing facility. The goal is to minimize the cost of production which includes the unit costs, fixed cost, and inventory cost. The problem in hand is meant to be solved with the help of Gurobi Python based on the constraints such as maximum inventory and production restrictions.

The particular problem of interest in this facility is determining the optimal production schedule for the next 35 days and creating a queueing theory that allows the warehouse to balance demand and capacity of outgoing goods with given constraints such as number of forklift drivers available.

Table of Contents

Introduction

The issue at hand is that the current factory (SuperPufft Snacks Corporation in Mississauga) production levels are at capacity. To fulfill all the orders that are coming in, the owners have suggested repurposing 5135 Creekbank Road, another facility that they own that is currently a furniture store. The facility is 100,000 square feet and has 7 delivery doors located at the northeast corner of the layout, there is another delivery door west of the layout that can accommodate one more truck that is reserved for receiving potatoes that load right onto a conveyor belt.

The project utilizes the following optimization techniques to achieve efficiency:

(i) Due to the large number of variables along with restrictions, and demand we are dealing with a complex problem that can be solved by breaking down the problem into subproblems using the optimization technique of dynamic programming. Through dynamic programming, the optimal production schedule can be determined such that costs are low and customers' orders (demands) are satisfied.

(ii) A major part of the operations process at the facility is shipping. With the limitations of 7 delivery doors and a large number of truck arrivals at the dock for pick-up, queuing theory must be applied to analyze and optimize the truck queues at the facility. Through Queuing theory, we plan to minimize the waiting times for each truck and ship the products efficiently and in a timely manner.

Methodologies

Assumptions:

Refer to the appendix (Figure 1 and Figure 2) for examples of a production report at **perfect production conditions.** What makes the production conditions “perfect” are written below.

Yield assumptions:

We can assume from the Heat And Control website that when the line is running, the production yield per hour is fixed, for example. If the PC-32 machine is selected, the production yield when the line is running is approximately 3200 LBS per hour, in the actual production yield in real life, there may be waste and recalls, and Heat and Control have taken this into account the production yield. [1]

Worker assumptions:

We can assume that once the factory machinery and layout have been selected, the number of workers is fixed at each part of the production line, therefore there is no variability in the time needed to complete a step and no sensitivity analysis is to be conducted. In the real world, there is no set staffing system, workers may call in sick or not work and there may not be a set system of 100 workers per shift, to better calculate the cost of running the production line. We made the assumption that there are 100 workers per shift working at an average of $18 per hour.

Quality Control Assumptions:

We can assume that as production is being produced, quality control is being done and that there are no recalls taking place. In the actual workplace, there may be recalls and product issues that may take place that may affect orders and reduce the amount of final product present.

Reliability Analysis:

1 problem that the dynamic programming system doesn't account for can be considered if the line has to stop in an emergency or the machine breaks down. The way to minimize this issue is to make sure that critical parts of the line are very reliable such as the fryer, and that maintenance is done regularly. On the other splitting parts that occur further along the line, it is less critical due to having multiple machines on standby that can take over the work. The assumption is that the line is running and no parts of it must stop meaning that production yield is stable.

Problem Definition

The potato chip manufacturing facility is planning to expand its operations to 5135 Creekbank Road and wants to generate capital for future expansions and development. They want to fulfill an order placed by Loblaws of 2 Million pounds of the finished product. ***(Demand).*** The order must be completed within 35 days ***(Time constraint).*** The factory is limited by the PC-32 Machine which produces 3200 Lbs/h for a total of 76800 Lbs/day. In 35 days there is the capability to make 2688000 Lbs ***(Production Constraint at time t).*** The warehouse can hold up to 300000 Lbs at time t ***(Inventory Constraint in t).*** And there is a cost associated with having the warehouse of .20 cents per lb. A similar modeling process can be seen in the Journal of Applied Mathematics (Gang et al, 2012), in their inventory problem they model it around multistage and variable demand.

The cost of potatoes is $1 per lb, 100 workers in 3 shifts total working at $18 average, line workers are minimum wage while mechanics and forklift operators are paid more, we can then assume that the 100x18x24= the cost incurred by employees only = $43200 per day.
The cost of Potatoes can come out to roughly $1 per Lb with transportation costs incurred.

Decision Variables

$$x_t\text{: } production\ units\ in\ week\ t$$

$$I_t\text{: } starting\ inventory\ in\ week\ t$$

$$y_t = \begin{cases} 1 & if\ x_t > 0 \\ 0 & otherwise \end{cases}$$

Parameters

$$C_t\text{: } Production\ cost\ per\ unit$$

$$f_t\text{: } Fixed\ setup\ cost$$

$$h_t\text{: } inventory\ holding\ cost$$

$$W_t = \ maximum\ number\ of\ units\ production\ in\ t$$

$$L_t = \ maximum\ inventory\ amount\ in\ t$$

$$d_t = \ demand\ in\ t$$

Objective

$$min \; = \; \sum_{t=1}^{T} C_t x_t + f_t y_t + h_t I_t$$

$$I_t + x_t \; = \; d_t + I_{t+1} \qquad for\; t \; = \; 1, 2 \,\ldots\ldots\, T$$

$$x_t \leq W_t$$

$$I_t \leq L_t$$

$$x_t \leq M y_t$$

$$x_t \,,\, I_t \; \geq \; 0$$

$$y_t \in \{0, 1\}$$

Problem Set

Dynamic Programming:

In order to determine the optimal production schedule for the next 35 days, we use dynamic programming to model and solve an inventory problem for this potato chip manufacturer. The total demand at the end of the 35-day period is 2,000,000 pounds of packaged and flavored products. The facility incurs a set-up cost of $135,000 every time they start production, the machines are capped at producing 76,800 pounds per day. The production cost for each pound is $1.3. The warehouse space allows for a total of 300,000 pounds of the packaged product, and the factory must pay a $0.2 charge for every pound of the finished product they hold for time t.

Queuing theory I:

The factory has hired **7 Counterbalance Forklift drivers with equal skill level** for shipping of finished goods. The loading docks can be seen as a parallel system in which each shipping door is available between the hours of 8AM(time 0) to 4PM (8 hours). The arrival can be determined to be a Poisson distribution [3] We want to aim for there to be no customers balking as the cost to have a truck go back ranges from a $300-2500 return fee depending on how far a customer has to come from and if sending it back means not fulfilling an order in. An average of 12 trucks arrive per hour and they can choose an empty door randomly as they become empty. The process takes 30 minutes on average. We can assume a first come first serve basis, FCFS. A similar process can be seen at the Alexandria Port in

Egypt, whereby investigating the pattern of ship traffic, a steady state poisson distribution was created to find the optimum number of ship berths (El- Naggar,2010). As well, El-Naggar states that it is key to understand that the queue is a straight line of ships waiting for a berth and then as a berth, or server in our case is emptied, the next ship can enter the port.

The base system can be seen as an (M/M/7/GD/∞/∞)
Input = Trucks enter the delivery door as they become empty.
Output = The truck has been loaded with the finished product, and has left the system.

Question 1:With 7 drivers, what is the percent utilization for a fixed number of trucks coming in per hour?

λ= 12 trucks/ hour
μ= 2 Customers/ hour served per driver.
ρ=λ / S(number of servers)(μ)
ρ= 12/ 7(2) = 0.857 or 85.7% utilization.

Question 2: We do not want the yard to be too crowded, so what is the expected number of truckers total at the yard?

Lq =ρ (J>= S)ρ / 1-ρ
= (0.65) (0.857) / 1-(0.857) = 3.895 trucks + λ /μ = 3.895 trucks + 12/2 = 9.89 trucks in the system.

9.89 Trucks between shipping doors and yards means there are 2-3 trucks waiting for a door to become free. From the picture, we can see that there is ample space at the NorthEast parking lot to accommodate 2-3 more 53" trailers waiting.

Question 3: What is the average waiting time in the queue?

$Wq = Lq/\lambda = 3.895 / 12 = 0.3246$ hours = 19.4777 mins

The Person has to wait 19.477 minutes on average in the queue before entering, during this time it is critical that the drivers sign in and tell the forklift operator that they have arrived.

Question 4: What is the average time a truck driver spends in the system?(queue + loading)

$Ws = Lq/\lambda = 9.89 / 12 = 0.82416$ hours = 49.45 mins total.

If the truck drivers complain about this amount of waiting time and begin to balk, the recommendation from us is to increase the amount of delivery doors, but this will incur a large setup cost. Or to hire more skilled forklift drivers that can do non straight loading patterns, meaning that more product can be delivered in less deliveries in the same amount of time*(Garner, 2023)*.

Straight: Normal loading procedure, a 53'' trailer holds 24 pallets standard size 48' x 40'.

Turned T: Loading 1 by length and one by width, holding 28 pallets.

Pinwheeled: Holding all pallets horizontally allows up to 30 pallets to be stored on the truck.

Birth and Death scenarios: We can say that the shipping process can be assumed as a birth and death process and that the factory is not concerned with anything past the shipping process. If there are 14 Trucks per hour served = , how many trucks will SuperPufft serve in a year? Assumption is that The company will operate its facility and lines at least 340 days out of the year. 8 hours a day = 2720 hours

λ= 112 trucks per day average. Or 14 trucks per hour

112 x 340 =38080 Trucks served a year.

If the truck carries 24 pallets, Each pallet containing 281.6 Lbs (Figure 1: appendix) 6758.4 x 38080 trucks = due to shipping constraints the average weight output of finished goods in the current state =257,359,872 LBS of finished product can be processed per year in current state (BIG M).

C) What is the probability that there are no deliveries in a whole day? That may mean a snowstorm occurs, or that the warehouse is completely empty.

$\rho0\ (1)= [(112 \times 1)^0\ e^{(-112x1)}] / 0! = e^{-112} = 2.28569368E-49$

The chance of no trucks coming the whole day is extremely low, so there should always be at least 1 Forklift operator at the factory to serve them.

D) What is the probability that all the forklift drivers are busy, this is best avoided as it increases the chance for accidents on the warehouse floor. Consider steady state probability, at 12 trucks served per hour.

$$\mathbf{P(n >= s) = [\ (1 / (s)!\ )*(\lambda/\mu)^s * (s*\mu)/(s\mu - \lambda)^2] * Po}$$

$$\mathbf{[\ (1 / (7)!\ )*(12/2)^7 * (7*2)/(14 - 12)^2 * Po}$$

$$\mathbf{P(n >= s) = 0.3069}$$

Python Code

```
!pip install gurobipy
import gurobipy as gp
from gurobipy import GRB

T = 35 ## number of days

inventorymodel = gp.Model('Inventory')

x = inventorymodel.addVars(T, name='production quantity at time t')
I = inventorymodel.addVars(T, name='inventory quantity at time t')
y = inventorymodel.addVars(T, vtype=GRB.BINARY, name='produce: (Yes = 1), (No = 0')

d = [57000]*T  ## demand at time t
w = [76800]*T  ## maximum production amount in t
L = [300000]*T  # maximum inventory amount in t
c = [1.3]*T  # production cost per pound
f = [135000]*T  ## fixex start up cost for the productinon line
h = [0.2]*T  # example inventory holding cost per pound

inventorymodel.addConstrs((I[t] + x[t] == d[t] + I[t+1] for t in range(T-1)), name='starting inventory constraint')
inventorymodel.addConstrs((I[t] + x[t] == d[t] for t in [T-1]), name='final inventory constraint')
inventorymodel.addConstrs((x[t] <= w[t]*y[t] for t in range(T)), name='production limit')
inventorymodel.addConstrs((I[t] <= L[t] for t in range(T)), name='maximum inventory amount in t')
inventorymodel.addConstrs((x[t] <= w[t] for t in range(T)), name='maximum production amount in t')

inventorymodel.setObjective(gp.quicksum(c[t]*x[t] + f[t]*y[t] + h[t]*I[t] for t in range(T)), GRB.MINIMIZE)

inventorymodel.optimize()

for t in range(T):
    print(f"Day {t+1}: Production Quantity = {x[t].x}, Inventory Quantity = {I[t].x}, Produce = {y[t].x}")
```

Analysis and Results

Output:

```
Optimize a model with 140 rows, 105 columns and 244 nonzeros
Model fingerprint: 0xf5a79b42
Variable types: 70 continuous, 35 integer (35 binary)
Coefficient statistics:
  Matrix range     [1e+00, 8e+04]
  Objective range  [2e-01, 1e+05]
  Bounds range     [1e+00, 1e+00]
  RHS range        [6e+04, 3e+05]
Found heuristic solution: objective 7120800.0000
Presolve removed 71 rows and 1 columns
Presolve time: 0.00s
Presolved: 69 rows, 104 columns, 172 nonzeros
Variable types: 69 continuous, 35 integer (35 binary)

Root relaxation: objective 5.394312e+06, 107 iterations, 0.00 seconds (0.00 work uni

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 5394312.19    0   27 7120800.00 5394312.19  24.2%     -    0s
H    0     0                    6329820.0000 5394312.19  14.8%     -    0s
     0     0 5476890.79    0   27 6329820.00 5476890.79  13.5%     -    0s
     0     0 5488303.22    0   28 6329820.00 5488303.22  13.3%     -    0s
     0     0 5493223.79    0   24 6329820.00 5493223.79  13.2%     -    0s
H    0     0                    5705580.0000 5493223.79  3.72%     -    0s
     0     0 5493852.10    0   28 5705580.00 5493852.10  3.71%     -    0s
     0     0 5493894.36    0   28 5705580.00 5493894.36  3.71%     -    0s
     0     0 5493894.36    0   28 5705580.00 5493894.36  3.71%     -    0s
H    0     0                    5664180.0000 5493894.36  3.01%     -    0s
     0     2 5495252.25    0   28 5664180.00 5495252.25  2.98%     -    0s
H  785   404                    5653740.0000 5523013.00  2.31%   3.9    0s

Cutting planes:
  Gomory: 1
  Implied bound: 29
  MIR: 23
  Flow cover: 17
  Inf proof: 4
  Relax-and-lift: 1

Explored 15688 nodes (76068 simplex iterations) in 3.25 seconds (0.67 work units)
Thread count was 2 (of 2 available processors)
```

```
Optimal solution found (tolerance 1.00e-04)
Best objective 5.653740000000e+06, best bound 5.653301819046e+06, gap 0.0078%
Day 1: Production Quantity = 0.0, Inventory Quantity = 300000.00000000105, Produce = 0.0
Day 2: Production Quantity = 0.0, Inventory Quantity = 243000.00000000102, Produce = -0.0
Day 3: Production Quantity = 0.0, Inventory Quantity = 186000.00000000102, Produce = 0.0
Day 4: Production Quantity = 0.0, Inventory Quantity = 129000.00000000102, Produce = -0.0
Day 5: Production Quantity = 0.0, Inventory Quantity = 72000.00000000102, Produce = 0.0
Day 6: Production Quantity = 41999.999999998974, Inventory Quantity = 15000.000000001026, Produce = 1.0
Day 7: Production Quantity = 74400.0, Inventory Quantity = 0.0, Produce = 1.0
Day 8: Production Quantity = 76800.0, Inventory Quantity = 17400.0, Produce = 1.0
Day 9: Production Quantity = 76800.0, Inventory Quantity = 37200.0, Produce = 1.0
Day 10: Production Quantity = 0.0, Inventory Quantity = 57000.0, Produce = 0.0
Day 11: Production Quantity = 74399.99999999953, Inventory Quantity = 0.0, Produce = 1.0
Day 12: Production Quantity = 76800.0, Inventory Quantity = 17399.999999999534, Produce = 1.0
Day 13: Production Quantity = 76800.0, Inventory Quantity = 37199.999999999534, Produce = 1.0
Day 14: Production Quantity = 0.0, Inventory Quantity = 56999.999999999534, Produce = -0.0
Day 15: Production Quantity = 74400.0, Inventory Quantity = 0.0, Produce = 1.0
Day 16: Production Quantity = 76800.0, Inventory Quantity = 17399.999999999534, Produce = 1.0
Day 17: Production Quantity = 76800.0, Inventory Quantity = 37199.999999999534, Produce = 1.0
Day 18: Production Quantity = 0.0, Inventory Quantity = 56999.999999999534, Produce = 0.0
Day 19: Production Quantity = 74400.00000000047, Inventory Quantity = 0.0, Produce = 1.0
Day 20: Production Quantity = 76800.0, Inventory Quantity = 17400.0, Produce = 1.0
Day 21: Production Quantity = 76800.0, Inventory Quantity = 37200.0, Produce = 1.0
Day 22: Production Quantity = 0.0, Inventory Quantity = 57000.0, Produce = 0.0
Day 23: Production Quantity = 74400.0, Inventory Quantity = 0.0, Produce = 1.0
Day 24: Production Quantity = 76800.0, Inventory Quantity = 17400.0, Produce = 1.0
Day 25: Production Quantity = 76800.0, Inventory Quantity = 37200.0, Produce = 1.0
Day 26: Production Quantity = 0.0, Inventory Quantity = 57000.0, Produce = 0.0
Day 27: Production Quantity = 57000.0, Inventory Quantity = 0.0, Produce = 1.0
Day 28: Production Quantity = 74400.00000000001, Inventory Quantity = 0.0, Produce = 1.0
Day 29: Production Quantity = 76800.0, Inventory Quantity = 17400.000000000015, Produce = 1.0
Day 30: Production Quantity = 76800.0, Inventory Quantity = 37200.000000000015, Produce = 1.0
Day 31: Production Quantity = 0.0, Inventory Quantity = 57000.000000000015, Produce = 0.0
Day 32: Production Quantity = 74399.99999999999, Inventory Quantity = 1.4551915228366852e-11, Produce = 1.0
Day 33: Production Quantity = 76800.0, Inventory Quantity = 17400.0, Produce = 1.0
Day 34: Production Quantity = 76800.0, Inventory Quantity = 37200.0, Produce = 1.0
Day 35: Production Quantity = 0.0, Inventory Quantity = 57000.0, Produce = 0.0
```

The optimal schedule demonstrates that the cheapest feasible solution is to produce and store 300000 pounds on day 1, then for the next 4 days there will be no production as stock is sold from storage. Due to insufficient inventory, the facility must produce 42000 lbs on day 6, 74400 lbs on day 7, 76800 lbs on day 8 and 9. On day 10 with a satisfactory amount of inventory, no production is required. 74399 lbs, 76800 lbs, and 76800 lbs are to be produced on day 11, 12, and 13 respectively in order to meet the demand.

Table 1: Costs per day for the optimal production schedule

Day	**Cost**	**Day**	**Cost**	**Day**	**Cost**	**Day**	**Cost**
1	138000	11	209099.96	21	238320	31	11400
2	18600	12	231720.03	22	242280	32	231720
3	217859.81	13	238320.01	23	11400	33	238320
4	242280	14	242279.85	24	231720	34	242280
5	11400.23	15	11400.30	25	238320	35	11400
6	209099.8	16	231719.78	26	242280		
7	231720	17	238320	27	11400		
8	238320	18	242280	28	231720		

9	242280	19	11400	29	238320		
10	11400.06	20	231720	30	242280		

Sensitivity Analysis

An unforeseen weather condition has caused damage to the warehouse facility and has resulted in a leak within the warehouse. The storage limit was affected by the leak, their warehouse capacity was decreased by half. A number of machines were also affected throughout the storm, management now has to pay an additional cost for maintenance and service during operations. Therefore the set-up cost and production is 15% greater than before, using the same dynamic programming model but with new constraints we are able to determine their production schedule in this scenario.

```
!pip install gurobipy
import gurobipy as gp
from gurobipy import GRB

T = 35 ## number of days

inventorymodel = gp.Model('Inventory')

x = inventorymodel.addVars(T, name='production quantity at time t')
I = inventorymodel.addVars(T, name='inventory quantity at time t')
y = inventorymodel.addVars(T, vtype=GRB.BINARY, name='produce: (Yes = 1), (No = 0')

d = [57000]*T  ## demand at time t
w = [76800]*T  ## maximum production amount in t
L = [150000]*T  # maximum inventory amount in t
c = [1.495]*T  # production cost per pound
f = [155250]*T  ## fixex start up cost for the productinon line
h = [0.2]*T  # example inventory holding cost per pound

inventorymodel.addConstrs((I[t] + x[t] == d[t] + I[t+1] for t in range(T-1)), name='starting inventory constraint')
inventorymodel.addConstrs((I[t] + x[t] == d[t] for t in [T-1]), name='final inventory constraint')
inventorymodel.addConstrs((x[t] <= w[t]*y[t] for t in range(T)), name='production limit')
inventorymodel.addConstrs((I[t] <= L[t] for t in range(T)), name='maximum inventory amount in t')
inventorymodel.addConstrs((x[t] <= w[t] for t in range(T)), name='maximum production amount in t')

inventorymodel.setObjective(gp.quicksum(c[t]*x[t] + f[t]*y[t] + h[t]*I[t] for t in range(T)), GRB.MINIMIZE)

inventorymodel.optimize()

for t in range(T):
    print(f"Day {t+1}: Production Quantity = {x[t].x}, Inventory Quantity = {I[t].x}, Produce = {y[t].x}")
```

```
Optimize a model with 140 rows, 105 columns and 244 nonzeros
Model fingerprint: 0x7d3daa50
Variable types: 70 continuous, 35 integer (35 binary)
Coefficient statistics:
  Matrix range     [1e+00, 8e+04]
  Objective range  [2e-01, 2e+05]
  Bounds range     [1e+00, 1e+00]
  RHS range        [6e+04, 2e+05]
Found heuristic solution: objective 8187210.0000
Presolve removed 71 rows and 1 columns
Presolve time: 0.00s
Presolved: 69 rows, 104 columns, 172 nonzeros
Variable types: 69 continuous, 35 integer (35 binary)

Root relaxation: objective 6.566034e+06, 136 iterations, 0.00 seconds (0.00 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 6566033.67    0   30 8187210.00 6566033.67  19.8%     -    0s
H    0     0                    7804395.0000 6566033.67  15.9%     -    0s
     0     0 6661261.25    0   29 7804395.00 6661261.25  14.6%     -    0s
H    0     0                    7683105.0000 6661261.25  13.3%     -    0s
     0     0 6672398.53    0   31 7683105.00 6672398.53  13.2%     -    0s
     0     0 6673373.36    0   31 7683105.00 6673373.36  13.1%     -    0s
     0     0 6675928.55    0   28 7683105.00 6675928.55  13.1%     -    0s
H    0     0                    6925725.0000 6675928.55  3.61%     -    0s
     0     0 6677334.69    0   31 6925725.00 6677334.69  3.59%     -    0s
     0     0 6677459.66    0   31 6925725.00 6677459.66  3.58%     -    0s
     0     0 6688527.68    0   25 6925725.00 6688527.68  3.42%     -    0s
     0     0 6690625.00    0   27 6925725.00 6690625.00  3.39%     -    0s
     0     0 6692090.04    0   27 6925725.00 6692090.04  3.37%     -    0s
     0     0 6693353.88    0   26 6925725.00 6693353.88  3.36%     -    0s
     0     0 6693497.18    0   27 6925725.00 6693497.18  3.35%     -    0s
     0     0 6693729.20    0   27 6925725.00 6693729.20  3.35%     -    0s
     0     0 6694464.27    0   26 6925725.00 6694464.27  3.34%     -    0s
     0     0 6695097.21    0   27 6925725.00 6695097.21  3.33%     -    0s
     0     0 6695172.60    0   27 6925725.00 6695172.60  3.33%     -    0s
     0     0 6695189.92    0   27 6925725.00 6695189.92  3.33%     -    0s
     0     0 6695628.26    0   26 6925725.00 6695628.26  3.32%     -    0s
     0     0 6696573.42    0   27 6925725.00 6696573.42  3.31%     -    0s
     0     0 6696641.15    0   27 6925725.00 6696641.15  3.31%     -    0s
     0     0 6696669.99    0   27 6925725.00 6696669.99  3.31%     -    0s
     0     0 6697614.96    0   27 6925725.00 6697614.96  3.29%     -    0s
     0     0 6698439.17    0   28 6925725.00 6698439.17  3.28%     -    0s
     0     0 6698439.17    0   28 6925725.00 6698439.17  3.28%     -    0s
H    0     0                    6918405.0000 6698439.17  3.18%     -    0s
H    0     0                    6877365.0000 6698439.17  2.60%     -    0s
     0     2 6698758.81    0   28 6877365.00 6698758.81  2.60%     -    0s
H 1171   421                    6870405.0000 6760886.86  1.59%   4.3    0s
```

```
Cutting planes:
  Cover: 2
  Implied bound: 27
  MIR: 42
  Flow cover: 49
  Inf proof: 17
  Relax-and-lift: 1

Explored 16361 nodes (111290 simplex iterations) in 3.88 seconds (1.36 work units)
Thread count was 2 (of 2 available processors)

Solution count 8: 6.8704e+06 6.87041e+06 6.87736e+06 ... 8.18721e+06

Optimal solution found (tolerance 1.00e-04)
Best objective 6.870405000000e+06, best bound 6.869984848790e+06, gap 0.0061%
Day 1: Production Quantity = 0.0, Inventory Quantity = 150000.0, Produce = 0.0
Day 2: Production Quantity = 0.0, Inventory Quantity = 93000.0, Produce = -0.0
Day 3: Production Quantity = 58200.0, Inventory Quantity = 36000.0, Produce = 1.0
Day 4: Production Quantity = 76800.0, Inventory Quantity = 37200.0, Produce = 1.0
Day 5: Production Quantity = 0.0, Inventory Quantity = 57000.0, Produce = 0.0
Day 6: Production Quantity = 57000.0, Inventory Quantity = 0.0, Produce = 1.0
Day 7: Production Quantity = 74399.99999999958, Inventory Quantity = 0.0, Produce = 1.0
Day 8: Production Quantity = 76800.0, Inventory Quantity = 17399.999999999578, Produce = 1.0
Day 9: Production Quantity = 76800.0, Inventory Quantity = 37199.99999999958, Produce = 1.0
Day 10: Production Quantity = 0.0, Inventory Quantity = 56999.99999999958, Produce = 0.0
Day 11: Production Quantity = 74400.00000000041, Inventory Quantity = 0.0, Produce = 1.0
Day 12: Production Quantity = 76800.00000000001, Inventory Quantity = 17399.99999999998, Produce = 1.0
Day 13: Production Quantity = 76800.0, Inventory Quantity = 37200.0, Produce = 1.0
Day 14: Production Quantity = 0.0, Inventory Quantity = 57000.0, Produce = 0.0
Day 15: Production Quantity = 74400.0, Inventory Quantity = 0.0, Produce = 1.0
Day 16: Production Quantity = 76800.0, Inventory Quantity = 17400.0, Produce = 1.0
Day 17: Production Quantity = 76800.0, Inventory Quantity = 37200.0, Produce = 1.0
Day 18: Production Quantity = 0.0, Inventory Quantity = 57000.0, Produce = 0.0
Day 19: Production Quantity = 74400.0, Inventory Quantity = 0.0, Produce = 1.0
Day 20: Production Quantity = 76800.0, Inventory Quantity = 17400.0, Produce = 1.0
Day 21: Production Quantity = 76800.0, Inventory Quantity = 37200.0, Produce = 1.0
Day 22: Production Quantity = 0.0, Inventory Quantity = 57000.0, Produce = 0.0
Day 23: Production Quantity = 57000.0, Inventory Quantity = 0.0, Produce = 1.0
Day 24: Production Quantity = 74399.99999999974, Inventory Quantity = 0.0, Produce = 1.0
Day 25: Production Quantity = 76800.0, Inventory Quantity = 17399.99999999973, Produce = 1.0
Day 26: Production Quantity = 76800.0, Inventory Quantity = 37199.99999999973, Produce = 1.0
Day 27: Production Quantity = 0.0, Inventory Quantity = 56999.99999999973, Produce = 0.0
Day 28: Production Quantity = 74400.00000000009, Inventory Quantity = 0.0, Produce = 1.0
Day 29: Production Quantity = 76800.00000000019, Inventory Quantity = 17399.999999999825, Produce = 1.0
Day 30: Production Quantity = 76800.0, Inventory Quantity = 37200.000000000015, Produce = 1.0
Day 31: Production Quantity = 0.0, Inventory Quantity = 57000.000000000015, Produce = 0.0
Day 32: Production Quantity = 74400.00000000028, Inventory Quantity = 0.0, Produce = 1.0
Day 33: Production Quantity = 76800.0, Inventory Quantity = 17400.000000000007, Produce = 1.0
Day 34: Production Quantity = 76800.0, Inventory Quantity = 37200.00000000001, Produce = 1.0
Day 35: Production Quantity = 0.0, Inventory Quantity = 57000.0, Produce = -0.0
```

Table 2: Costs per day for the optimal production schedule after changes

Day	Cost	Day	Cost	Day	Cost	Day	Cost
1	158350	11	266478	21	277506	31	11400
2	18600	12	273546	22	11400	32	266478
3	24959	13	277506	23	240465	33	273546
4	277506	14	11400	24	266478	34	277506
5	11400	15	266478	25	273546	35	11400
6	240465	16	27346	26	277506		
7	266478	17	277506	27	11400		
8	277506	18	11400	28	266478		
9	11400	19	266478	29	273546		
10	266478	20	273546	30	277506		

Future dynamic programming approach: Vehicle Routing Problem

For their logistics department, they have hired a 3-rd party delivery company to satisfy their shipping needs. They currently have 3 6-axle trucks which charge 3.55$ per kilometer traveled [1]. Given that the drivers work a 12 hour shift, they must return the truck to the facility before the end of their shift for the next working day. They currently have 3 6-axle container trucks that deliver orders to customer supermarket locations within a 150 km radius from the facility located at 5135 Creekbank Road Mississauga.

1 : Production Facility 5135 Creekbank Rd Mississauga
2 : Vince's Market 55 Queen St S Unit 1, Tottenham
3 : Walmart Supercentre 545 Holland St W, Bradford
4 : Walmart Supercentre 150 McEwan Dr E, Bolton
5 : Food Basics 9600 Islington Ave, Woodbridge
6 : Food Basics 555 Essa Rd, Barrie
7 : Nick's NO FRILLS 165 Wellington St W, Barrie
8 : Food Basics 380 Eramosa Rd, Guelph
9 : Food Basics, 84 Lynden Rd, Brantford
10: Metro 395 Wellington Rd, London
11: Loblaws Richmond Street, 1740 Richmond St, London

Table 3: Distances in kilometers between destinations and demand for integer programming problems.

To (j) / From	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*

(i)											
1		53.5	64.5	38.6	27.8	91.1	95	67.6	85	175	171
2	53.5		23.1	21.8	37.8	44.9	54	73.3	138	192	196
3	64.5	23.1		40.8	40.2	32.8	36.7	98.1	149	239	249
4	38.6	21.8	40.8		12.8	67.6	71.4	70.9	126	207	217
5	27.8	37.8	40.2	12.8		66.3	70.2	85	113	183	192
6	91.1	44.9	32.8	67.6	66.3		7.1	147	175	265	255
7	95	54	36.7	71.4	70.2	7.1		128	181	269	261
8	67.6	73.3	98.1	70.9	85	147	128		118	132	130
9	85	138	149	126	113	175	181	118		94.7	102
10	175	192	239	207	183	265	269	132	94.7		9.6
11	171	196	249	217	192	255	261	130	102	9.6	

Assumptions

The distances between each location was determined using Google Maps, assuming the driver drives the truck to the destination accident-free. For budgeting purposes, the production company prefers to satisfy orders that are less than 100km away from their facility. Let i and j be the locations mentioned above. They must make 6 deliveries in one working day, the facility operates Monday to Friday.

x_{ij} is a binary variable where, $x_{ij} = 1$ if the delivery driver will drive from i to j, $x_{ij} = 0$ otherwise.

d_{ij} is the distance traveled between location i and j.

Objective function: $min\ z = 3.55\ \boldsymbol{\Sigma i \Sigma j}\ d_{ij} x_{ij}$

Constraints:

$\boldsymbol{\Sigma i \Sigma j}\ d_{ij} x_{ij} \leq 1,000\ (km\ /\ day)$
$\boldsymbol{\Sigma i \Sigma j}\ d_{1j} x_{1j} \leq 150\ \text{(km)}$
$\boldsymbol{\Sigma j}\ x_{1j} + \boldsymbol{\Sigma j}\ x_{2j} + \boldsymbol{\Sigma j}\ x_{3j} + \boldsymbol{\Sigma j}\ x_{4j} + \boldsymbol{\Sigma j}\ x_{5j} + \boldsymbol{\Sigma j}\ x_{6j} + \boldsymbol{\Sigma j}\ x_{7j} + \boldsymbol{\Sigma j}\ x_{8j} + \boldsymbol{\Sigma j}\ x_{9j} + \boldsymbol{\Sigma j}\ x_{10j} + \boldsymbol{\Sigma j}\ x_{11j} \leq 6$

$x_{1j} \in \{ 0, 1 \}$ for all i and j

Due to the complexity of the dynamic programming vehicle routing problem, the team was unable to generate and solve a concrete model to determine the best routes to take.
However, by analyzing the stages, states and constraints. If the demand in London, Ontario, and Southern Ontario increases, it is best that an expansion of factories is done to service those regions to satisfy constraints and minimize the cost. As well, the customer base may not be consistent from order to order, meaning that creating the best possible vehicle routing path may vary from week to week.

Conclusion

With the help of dynamic programming and queuing theory, the optimal solution was figured out with production and warehouse constraints in mind. The analysis was performed based on the restrictions, demands, and other constraints that determine the efficiency and profitability of the production facility. These techniques allowed us to make a final recommendation to SuperPufft Snacks Corporation and create a shipping queuing theory that is feasible and safe. In future iterations of programming, we wish to look at the vehicle routing problem as well as create a receiving schedule for the receiving of boxes and film that are to be used along the line.

References

[1] Berry, R. Y. A. N. (2006). Queuing theory. *Senior Project Archive*, 1-14.

[2] El-Naggar, M. E. (2010). Application of queuing theory to the container terminal at Alexandria seaport. *Journal of Soil Science and Environmental Management*, *1*(4), 77-85.

[3] Garner, A. (2023, February 1). How many pallets fit in a 53-foot dry van trailer? Worldwide Transportation Solutions. Retrieved April 10, 2023, from https://www.atsinc.com/blog/how-many-pallets-fit-in-trailer-explained

[4] Gang, D., Li, C., Yin-Zhen, L., Jie-Yan, S., & Tanweer, A. (2012, November 13). Optimization on production-inventory problems with multistage and varying demand. Retrieved April 11, 2023, from https://www.hindawi.com/journals/jam/2012/648262/

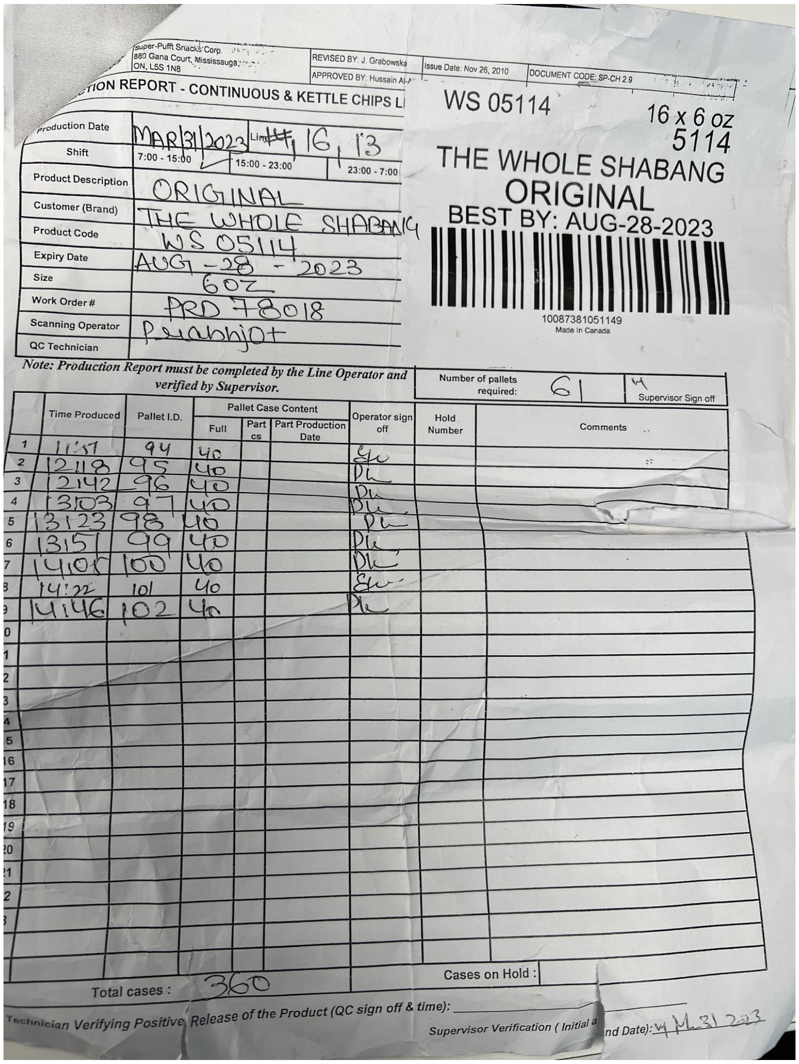
[5] Logistics, S. (n.d.). Warehouse Space Calculations. from https://www.spartanlogistics.com/warehouse-space-calculator

[6] Potato Chip Fryer. (n.d.), from https://www.heatandcontrol.com/model/potato-chip-fryer

[7] What is the cost of Step Deck Freight Shipping in Ontario? OCT Freight Shipping. (2021, April 8). Retrieved April 10, 2023, from https://octi.ca/what-is-the-cost-of-step-deck-freight-shipping-in-ontario/

Appendix

Figure 1: Continuous Production Report of "THE WHOLE SHABANG" chips provided by Hussain Al-Ali for a 360 case order.



Super-Pufft Snacks Corp.
880 Gana Court, Mississauga,
ON, L5S 1N8

REVISED BY: J. Grabowska | APPROVED BY: Hussain Al-A | Issue Date: Nov 26, 2010 | DOCUMENT CODE: SP-CH 2.9

…TION REPORT - CONTINUOUS & KETTLE CHIPS L…

Production Date	MAR/31/2023 Line #4, 16, 13
Shift	7:00 - 15:00 ✓ / 15:00 - 23:00 / 23:00 - 7:00
Product Description	ORIGINAL
Customer (Brand)	THE WHOLE SHABANG
Product Code	WS 05114
Expiry Date	AUG - 28 - 2023
Size	6OZ
Work Order #	PRD 78018
Scanning Operator	Prabhjot
QC Technician	

WS 05114
16 x 6 oz
5114
THE WHOLE SHABANG
ORIGINAL
BEST BY: AUG-28-2023
10087381051149
Made In Canada

Note: Production Report must be completed by the Line Operator and verified by Supervisor.

Number of pallets required: 61 | Supervisor Sign off

	Time Produced	Pallet I.D.	Pallet Case Content: Full	Part cs	Part Production Date	Operator sign off	Hold Number	Comments
1	11:57	94	40			[illegible]		
2	12:18	95	40			[illegible]		
3	12:42	96	40			[illegible]		
4	13:03	97	40			[illegible]		
5	13:23	98	40			[illegible]		
6	13:51	99	40			[illegible]		
7	14:05	100	40			[illegible]		
8	14:22	101	40			[illegible]		
9	14:46	102	40			[illegible]		
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								

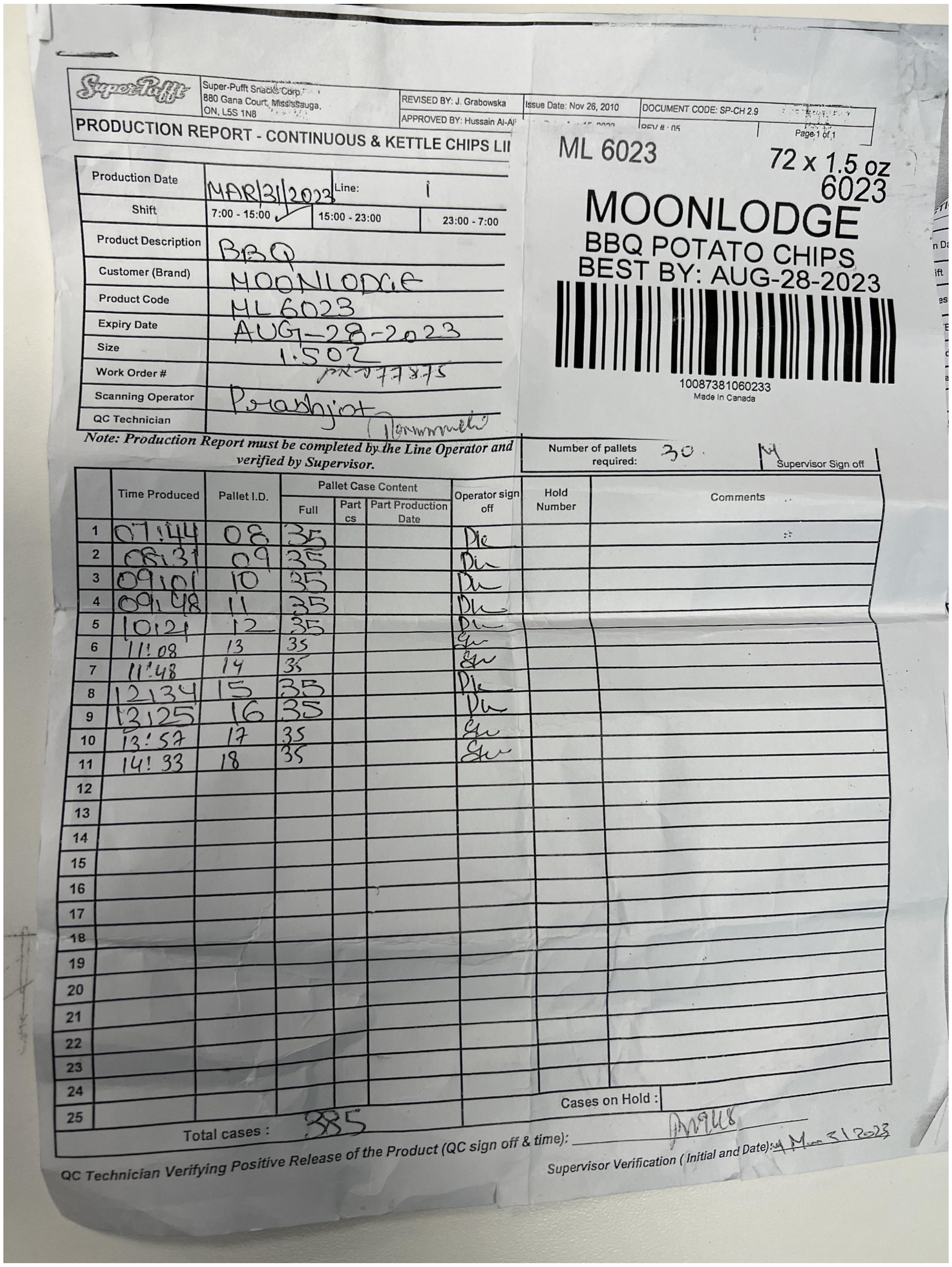
Total cases: 360

Cases on Hold:

Technician Verifying Positive Release of the Product (QC sign off & time): ______

Supervisor Verification (Initial and Date): M.31.2023

Figure 2: Continuous Production Report of "MOONLODGE BBQ" chips provided by Hussain Al-Ali For order of 385 cases.



Super-Pufft Snacks Corp.
880 Gana Court, Mississauga, ON, L5S 1N8

REVISED BY: J. Grabowska	Issue Date: Nov 26, 2010	DOCUMENT CODE: SP-CH 2.9
APPROVED BY: Hussain Al-Al		REV # : 05 — Page 1 of 1

PRODUCTION REPORT - CONTINUOUS & KETTLE CHIPS LI

Production Date	MAR/31/2023	Line:	1
Shift	7:00 - 15:00 ✓	15:00 - 23:00	23:00 - 7:00
Product Description	BBQ		
Customer (Brand)	MOONLODGE		
Product Code	ML 6023		
Expiry Date	AUG-28-2023		
Size	1.5OZ		
Work Order #	PR 077875		
Scanning Operator	Prabhjot		
QC Technician	Tammmeth		

ML 6023
72 x 1.5 oz
6023
MOONLODGE
BBQ POTATO CHIPS
BEST BY: AUG-28-2023
10087381060233
Made in Canada

Note: Production Report must be completed by the Line Operator and verified by Supervisor.

Number of pallets required: 30 — Supervisor Sign off

	Time Produced	Pallet I.D.	Pallet Case Content: Full	Part cs	Part Production Date	Operator sign off	Hold Number	Comments
1	07:44	08	35			Dk		
2	08:31	09	35			Dk		
3	09:01	10	35			Dk		
4	09:48	11	35			Dk		
5	10:21	12	35			Dk		
6	11:08	13	35			Gu		
7	11:48	14	35			Gu		
8	12:34	15	35			Dk		
9	13:25	16	35			Dk		
10	13:57	17	35			Gu		
11	14:33	18	35			Gu		
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								

Total cases : 385

Cases on Hold :

QC Technician Verifying Positive Release of the Product (QC sign off & time): ____

Supervisor Verification (Initial and Date): Mar 31 2023