Bayer Business Service - 2020 INFORMS O.R. Case

Executive Summary

Headquartered in Chicago, IL, Jelly Bean (JB) Manufacturing requires the establishment of an effective distribution network for raw materials movement across the five different manufacturing facilities in the Midwestern region in the U.S. The aforementioned objective entails the consideration of the resource constraints in each of these manufacturing facilities, specifically in several stages of the process flow. Further restrictions include transportation capacity and facility-based storage capacity at the raw material level. Considering there is a seasonal demand to be met, JB Manufacturing focuses on satisfying maximum demand and minimizing the manufacturing and transportation costs.

As we attempted to frame the problem, we identified several challenges associated with the problem statement:

- No deterministic internal work orders
- A large number of product variation
- Stochastic processing rate of packaging and pre finish
- A wide array of constraints
- Multiobjective optimization problem

As our team approached the problem, we identified the state variables capturing all the possible events for the entire system and developed a complex algorithm. Afterward, we used discrete events simulation to estimate the productions, lost sales and production capacities at different sites. Finally, we incorporated simulation based optimization for internal work order generation and raw material transportation. For tools, we largely worked on Python for simulation and optimization.

Team Makeup & Process

The formulation of the team came at an advantage from several perspectives. All of the team members were Masters Students in Industrial Engineering major. Two of the team members were specializing in operations research track and the rest of them in supply chain management. Team members with such specialized background helped not only in analyzing the problem from multiple perspectives, but also in formulating specific problems in a structured manner and devising industrially optimized solutions. Moreover, three of the team members had extended work experience in supply chain operations in manufacturing industries. The other three team members had work experiences in technology industry. Such work experience helped tremendously to visualize and break down the problem into specific considerations within a short period of time. All of the team members had a certain level of python and data analytics expertise, which along with other aforementioned skills, provided a platform for effective communication, high-level brainstorming, and seamless workflow.

The entire workload was divided into two core segments: optimization research and organizational planning. Optimization research works included data structuring, development of simulation and optimization models in python. These responsibilities were given to three members with better expertise in coding and simulation. Organizational planning included flowchart development of the discrete events in the entire manufacturing process, model implementation, and formal report write-ups. For the entire time frame of the project, the whole team worked together to ensure effective understanding of every approaches attempted. Moreover, to establish synchronized workflow, information and material flow were maintained through cloud storages. The only challenge that surfaced during the entire project was the limited time allocation for the work due to the conflicting class schedules of the team members and winter break.

Framing the Problem

JB Manufacturing currently has five manufacturing facilities in Detroit, MI, Columbus,OH, Greenbay, WI, Omaha, NE, and Springfield, MO. The orders to be placed in these manufacturing facilities are being generated and regulated through the headquarter in Chicago, IL. The primary goal of the company is to meet the Halloween demand for jelly beans, which needs to be produced within the timeframe of April 1st (8am) to September 30th (5pm) with 24/7 operating time. Moreover, JB Manufacturing identifies any production after 5pm on September 30th as “lost sales”; therefore, the company plans to maximize the production during the given time period. Before every season, the headquarter generates an order bank, where a particular order consists of customer demand for a certain color, size, flavor, and packaging combination. Based on the order bank, we get an idea about the different demand quantities of 4800 combinations (number of combinations derived from 40 different colors, 5 sizes, 12 flavors, and 2 packaging), which helped us developing the simulation model. A detailed algorithm in the form of flow chart was meticulously designed to capture all the possible discrete events in the manufacturing process along with the constraints and assumptions. The simulation model was executed based on the flowchart. Further discussion on the flowchart is delineated in the methodology section.

We used discrete event simulation approach to simulate the system. To specify the simulation modeling, we have defined the following:

Assumptions:

1. All the filling rates are instantaneous.
2. Flavor was assigned to the PFI drums right before the pre finish operation
3. PFI drum feed to the flavoring tank would be based on the homogeneity of flavor instead of FIFO
4. In a single que, one PFI drum would be unloaded to the flavoring tank at a time
5. In a single que, one PI drum would be unloaded for packaging

Events:

The following events would occur in the simulation:

1. Realizing a RMI drum in the classifier
2. Size segmentation (classifying) finish at classifier
3. Flavoring finish (for each tank)
4. Flavoring change finish (finish waiting for flavor change in each tank)
5. Boxing finish in the boxing machine
6. Bagging finish in the bagging machine

Indices and State Variables:

Here we define all of the state variables that we are keeping track of them during the time. Before defining them we need to define the required indices:

1. i: represents the color of the material
2. j: represents the size of the material
3. k: represents the flavor of the material
4. l: represents the type of the packaging (boxing or bagging)
5. m: represents the index of flavoring tank

Now we can define following state variables:

1. Y_{ij}: Amount of the material of color i and size j that is left in the classifier machine
2. L: 0 if the classifier machine is idle, 1 if the machine is busy
3. I_m: 0 if the tank m is idle, 1 if the tank is busy
4. $Tank_m\_mtl$: amount of the material that is left into the tank m
5. J_{xm} ($x = b$ for boxing machine, g for bagging machine, m = number of machines): *0* if the machine is idle, *1* if the machine is busy
6. F_{ijkl}: Amount of the material produced by packing type l of color i and size j
7. *Size of the RMI drum que*
8. *Size of the Sized_que_j*: This que represents empty PFI drums assigned to size
9. *Size of the tank_que*: This que represents filled PFI drums waiting to be assigned to the flavoring tank

10. *Size of the box_que*: This que represents filled PI drums waiting to be assigned to the boxing machine

11. *Size of the bag_que*: This que represents filled PFI drums waiting to be assigned to the bagging machine

Entities: This is the list of all entities that exists in the

1. RMI drum
2. PFI drum
3. PI drum

Data

We observed that the demand quantities in the order bank and RMI tables varied largely, therefore, size break down quantities based on the color, size, flavor, and packaging were required for simulation. To approach this issue, we identified 4800 combinations of jelly beans providing an equal number of normal distributions (4800) with different means and standard deviations. A random demand was created for each combination using the mean and standard deviation. The color wise random demands were compared to the color wise quantity in RMI table to find a 'weight'. Later on, that weight was multiplied with the random demands from each combination to find the final demand for a particular size breakdown that closely matched the RMI quantity distribution. The final demand table looked like below for a specific size:

Index	Color	Size	Flavor	Pack	Demand
0	Coloring Agent21	S5	F7	Bag	3780
1	Coloring Agent21	S5	F7	Box	3483
2	Coloring Agent21	S5	F8	Bag	0
3	Coloring Agent21	S5	F8	Box	0
4	Coloring Agent21	S5	F9	Bag	2221
5	Coloring Agent21	S5	F9	Box	3109
6	Coloring Agent21	S5	F2	Bag	6347
7	Coloring Agent21	S5	F2	Box	1834
8	Coloring Agent21	S5	F12	Bag	3958
9	Coloring Agent21	S5	F12	Box	2331
10	Coloring Agent21	S5	F11	Bag	2725
11	Coloring Agent21	S5	F11	Box	3362
12	Coloring Agent21	S5	F6	Bag	4279
13	Coloring Agent21	S5	F6	Box	0
14	Coloring Agent21	S5	F4	Bag	2377
15	Coloring Agent21	S5	F4	Box	0
16	Coloring Agent21	S5	F10	Bag	3193

For the convenience of simulation, we defined some data structures to understand state of the system. Following are the main data structures that we have used for coding and simulation part. Here we just read the data provided by the source and tried to capture all entities with required attributes in a meaningful way for the sake of coding.

PI: For each site, PI data structure comprises of 7 variables: ‘Drum’, ‘Capacity’, ‘Color’, ‘Size’, ‘Flavor’, ‘Pack’ and ‘Level’. This data structure has been derived from ‘Pack inventory Drum.csv’ file. Here Variable ‘Pack” refers to Packaging type.

Here is an example of this data structure ran in python:

Index	Site	Drum	Capacity	Color	Size	Flavor	Level
0	Detroit, MI	FI Drum 1	20000	nan	0	nan	0
1	Detroit, MI	FI Drum 2	20000	nan	0	nan	0
2	Detroit, MI	FI Drum 3	20000	nan	0	nan	0
3	Detroit, MI	FI Drum 4	20000	nan	0	nan	0
4	Detroit, MI	FI Drum 5	20000	nan	0	nan	0
5	Detroit, MI	FI Drum 6	20000	nan	0	nan	0
6	Detroit, MI	FI Drum 7	20000	nan	0	nan	0
7	Detroit, MI	FI Drum 8	20000	nan	0	nan	0

PFI: For each site, PFI data structure comprises of 6 variables: ‘Drum’, ‘Capacity’, ‘Color’, ‘Size’, ‘Flavor’, and ‘Level’. This data structure has been derived from 'Pre-finish Inventory Drum.csv' file.
Here is an example of this data structure ran in python:

Index	Site	Drum	Capacity	Color	Size	Flavor	Level
0	Detroit, MI	PFI Drum1	10000	nan	nan	nan	0
1	Detroit, MI	PFI Drum2	10000	nan	nan	nan	0
2	Detroit, MI	PFI Drum3	10000	nan	nan	nan	0
3	Detroit, MI	PFI Drum4	10000	nan	nan	nan	0
4	Detroit, MI	PFI Drum5	10000	nan	nan	nan	0
5	Detroit, MI	PFI Drum6	10000	nan	nan	nan	0
6	Detroit, MI	PFI Drum7	10000	nan	nan	nan	0
7	Detroit, MI	PFI Drum8	10000	nan	nan	nan	0
8	Detroit, MI	PFI Drum9	10000	nan	nan	nan	0
9	Detroit, MI	PFI Drum10	10000	nan	nan	nan	0

We have used one function named ‘*bin_segmentor*’ which categorize the PFI bins into sizes that it will used for carrying. After using the ‘*bin_segmentor*’ we get this one:

Index	Site	Drum	Capacity	Color	Size	Flavor	Level
0	Detroit, MI	PFI Drum01	10000	nan	S1	nan	0
1	Detroit, MI	PFI Drum02	10000	nan	S1	nan	0
2	Detroit, MI	PFI Drum03	10000	nan	S1	nan	0
3	Detroit, MI	PFI Drum04	10000	nan	S2	nan	0
4	Detroit, MI	PFI Drum05	10000	nan	S2	nan	0
5	Detroit, MI	PFI Drum06	10000	nan	S2	nan	0

We have used another useful function called '*sitewise_demand_preparation'* which prepares the demand distribution or combination based on work order for a specific site:

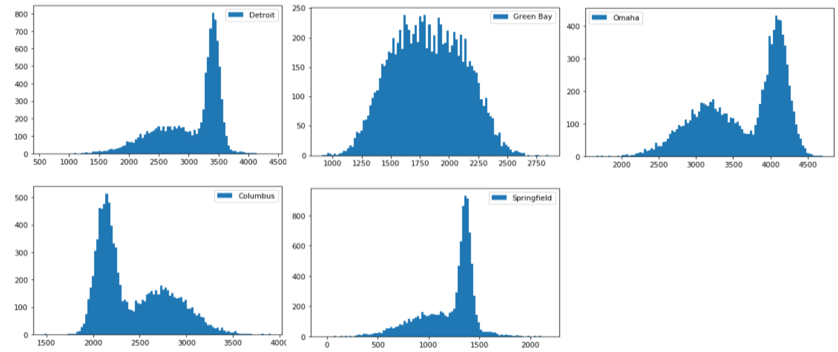
Index	Color	Size	Flavor	Pack	Demand
0	Coloring Agent02	S5	F7	Bag	6710
1	Coloring Agent02	S5	F7	Box	5145
2	Coloring Agent02	S5	F8	Bag	3381
3	Coloring Agent02	S5	F8	Box	5179
4	Coloring Agent02	S5	F9	Bag	8578
5	Coloring Agent02	S5	F9	Box	4704
6	Coloring Agent02	S5	F2	Bag	14785

We defined FEL (first event list) as *'FEL_builder'* in the simulation to keep track of the created events with corresponding times and save them in the following structure:

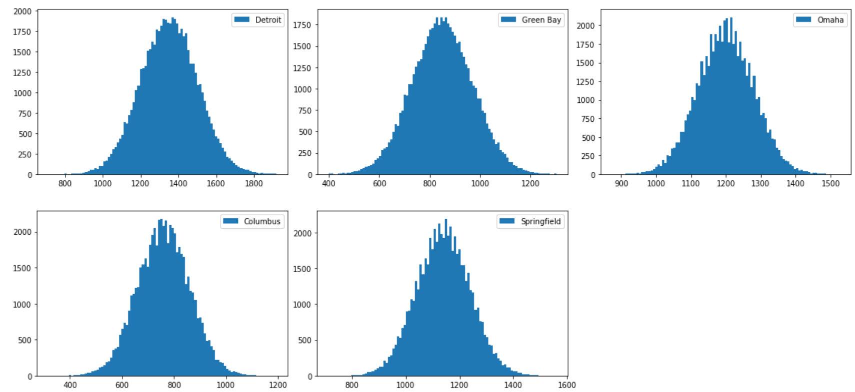
```
In [136]: FEL
Out[136]:
                       Event         Time
0                Classifying    87.719298
1       Changed Flavor in Tank1     5.000000
2  Finished Flavoring in tank 1  1435.200000
3  Finished Flavoring in tank 1  1191.800000
4             finished boxing   641.800000
5            finished bagging   793.400000
```

This list would always captures the name of the event and its occurrence time.

Dataset distribution:



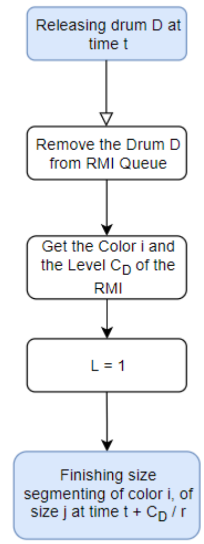
Packaging rate



Pre-finish operation rate

Methodology Approach & Model Building

<u>*Development of algorithm:*</u>



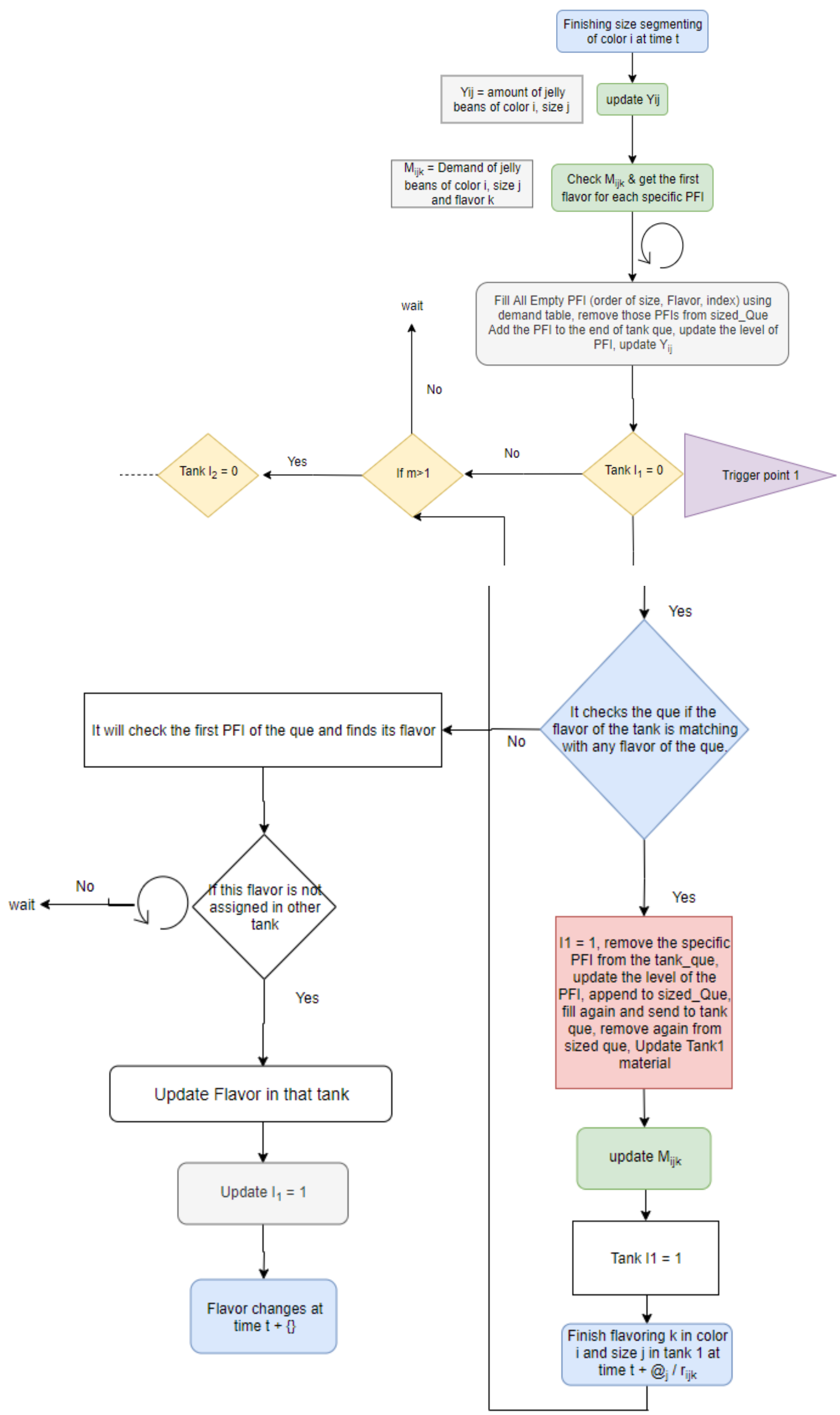
Since the color coating has already been done, the colored jelly beans are considered as "raw material" stored in RMI drums for further processing. Hence, the first event for the flowchart was to **release a RMI drum**. The release of drum depends on the color sequence specified in the internal workorder, followed by the RMI drum index number, given that one color can have multiple drums. The quantity of the jelly beans in each RMI drum was specified by C_D. At a particular time t, drum D would be released and the next event, **size segmentation**, would be triggered. From the *Classifier Split* table, the varying size break down for each color was retrieved. The classifier would take C_D / r (r being the processing rate of the classifier) amount of time to finish size segmentation based on that split percentage, and we declared a variable Y_{ij} which accounted for the quantity of sized jelly beans for color i and size j. The classifier's occupancy would be denoted by a binary variable L (L = 1 when classifier is busy, 0 o.w.). The release of the next RMI would depend on some scenarios and therefore, be decided later. The time t would be a global variable which would be updated at every stage of the simulation.

We generated color, size, and flavor wise random demands to segregate the RMI drum quantities of jelly beans (detailed procedure delineated in the data section). This procedure helped to track the filling process in the PFI drums and several further operations. Since there were no loading/unloading rate provided, we assumed that all types of drums would load/unload instantaneously. The minimum requirement of empty PFI drums to release a new RMI drum would be ensured in the later phase of the algorithm. For the convenience of simulation, we assigned the flavor to the PFI drums before the flavoring process. We would declare another variable M_{ijk} to check the demand of jelly beans of color i, size j, and flavor k ready to be fed into the flavoring machine. Once the flavoring process is done for each PFI, we would update M_{ijk}. This update would relate to the randomly generated demand that we introduced in the previous event.

All the filled PFI drums would be fed in a single queue, waiting to be unloaded in the available flavoring tanks. The tank's occupancy would be denoted by a binary variable I_m (I_m = 1 when tank is busy, 0 o.w.; m = number of tanks in one particular site). Instead of considering FIFO policy, we assumed that the unloading of PFI drum would be based on the homogeneity of the flavor currently being processed in the tank. The choice could be made in two ways: either unloading of a PFI drum with same flavor, or introducing a new flavor to the tank. Operating in such a way would dramatically reduce the "change-over time" required to change the flavor. Based on the dataset given for pre-finish operation, we observed a normal distribution for the processing rates for a specific size and flavor. The tank would finish flavoring at $t + @_{jk} / r_{ijk}$ time, where

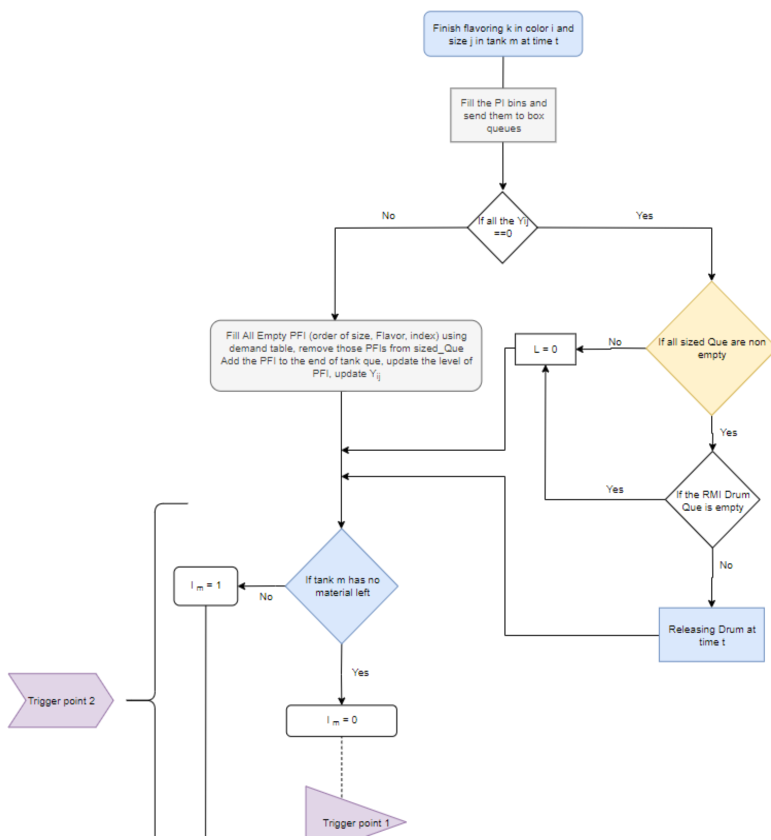
α_{jk} = quantity of jelly beans of size j and flavor k in a PFI drum

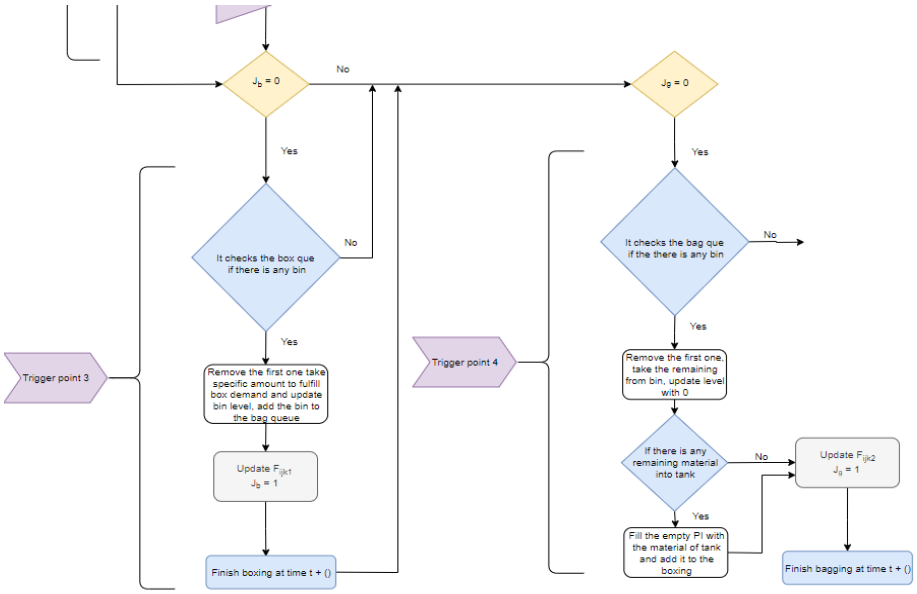
r_{ijk} = processing rate of tank for color i, size j, and flavor k



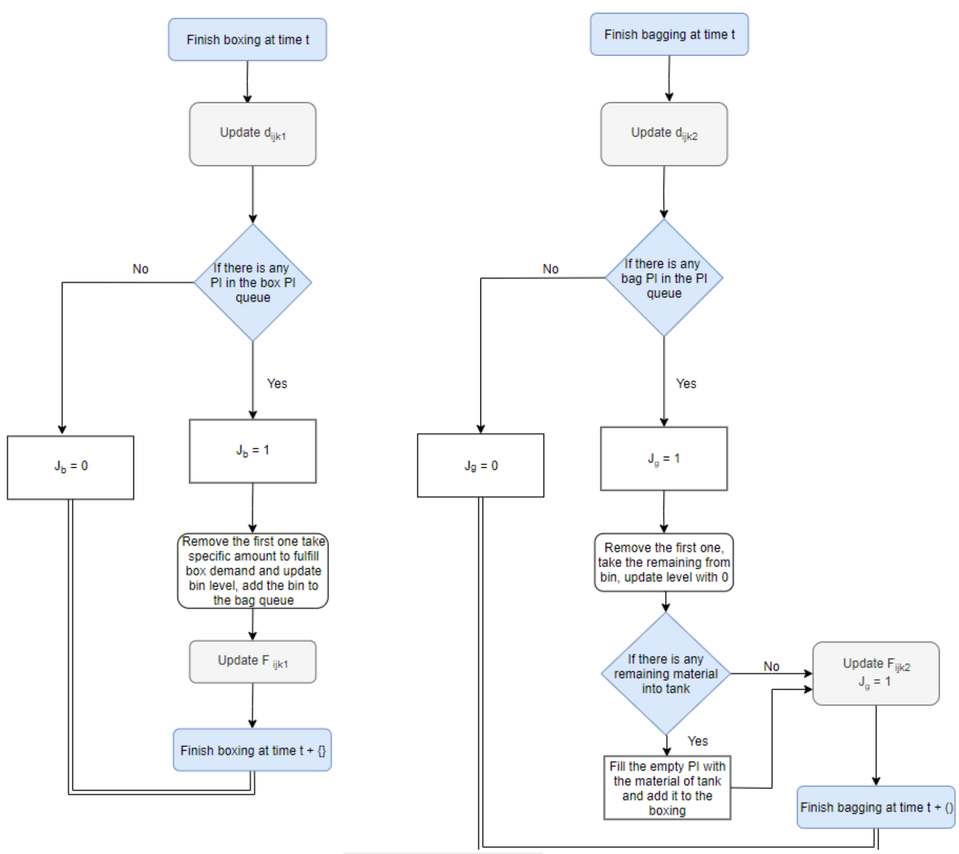
The PI drums would be filled up after the flavoring process, and simultaneously, whether the classifier would have any jelly beans remaining to be loaded to PFI drums would be checked. This decision would lead to two events: if the classifier is not empty, the PFI drums would be continuing to be filled with jelly beans, being transferred to the flavoring tank and empty PFI drums would be returned to the queue in front of the classifier. If the classifier is empty, then we would check two decisions: the minimum requirement to release a RMI drum (at least five available PFI drums), and available RMI drums. Whether an RMI is released or not, along with the first scenario, it would lead to the checking of flavoring tank being occupied or not. Referring to the previous section, either homogeneous flavor of PFI drum would be unloaded, or new flavor would be introduced.

The filled PI drums would be staged in the packaging section. The packaging machines would be declared with a variable J_{xm} ($x = b$ for boxing machine, g for bagging machine, m = number of machines). Since we had 4800 combinations of demands, and the boxing machine would get priority over bagging, the PI drums would be fed on to the boxing machine based on box demand. Once all the boxing demands are satisfied, the unmet demands of bagging would be processed afterward with the remaining level of beans in the PI drums. The quantity of packaged items would be kept track using a variable F_{ijkl} . At the end of processing, time t would be updated. Based on the dataset given for packaging operation, we observed a bimodal distribution for the processing rates of packaging. For the convenience of simulation, we considered normal distribution for both box and bag packaging. We considered that for each combination, there would be always precedence for machines, but the machines would run at the same time, otherwise there would be significant idle time in the packaging station and a bottleneck situation in front of the packaging station.





After the packaging stage, the empty PI drums would be taken back and staged right next to the flavoring machine. The machines would continue checking for PI drums in the box/bag queue and operate accordingly.



Simulation modelling in Python:

We conducted the simulation for each of the manufacturing sites separately to estimate their production capabilities within the required time frame. At the beginning of the simulation, the RMI drums for a site were fed into the classifier one at a time. Afterward, the quantity of the sized jelly beans for a particular color were stored in memory (shown in the picture).

Index	Color	Size	Level
0	Coloring Agent13	S1	14484
1	Coloring Agent13	S2	4494
2	Coloring Agent13	S3	5558
3	Coloring Agent13	S4	23977
4	Coloring Agent13	S5	570

The number of PFI drums were segmented according to the same classifier split percentage and at a particular time, the available PFI drums for a specific size are shown below. The level would indicate the current level of jelly beans inside the PFI drum (which in this case is 0). One PFI drum would leave for the pre-finish operation as soon as it would be loaded to either 95% of the capacity or whatever amount left for that size in the classifier. Based on every PFI drum load, the levels of the sized jelly beans in the previous table would be updated. The PFI drum number would ensure the precedence of drums to be loaded with jelly beans from the classifier.

Index	Site	Drum	Capacity	Color	Size	Flavor	Level
0	Detroit, MI	PFI Drum1	10000	Coloring Agent1	S1	F7	0
1	Detroit, MI	PFI Drum2	10000	Coloring Agent1	S1	F7	0
2	Detroit, MI	PFI Drum3	10000	Coloring Agent1	S1	F7	0

In the pre-finish operation, the queue of the loaded PFI drums would like below.

Index	Site	Drum	Capacity	Color	Size	Flavor	Level
0	Detroit, MI	PFI Drum6	10000	Coloring Agent1	S2	F9	1431
1	Detroit, MI	PFI Drum12	10000	Coloring Agent1	S4	F9	770
2	Detroit, MI	PFI Drum15	10000	Coloring Agent1	S5	F9	1351
3	Detroit, MI	PFI Drum11	10000	Coloring Agent1	S4	F7	609
4	Detroit, MI	PFI Drum14	10000	Coloring Agent1	S5	F7	8578

Since we assumed that one PFI drum could be unloaded to the flavoring tank at a time, the picture shows the details of the PFI drum being unloaded to the tank from the queue.

Index	6
Site	Detroit, MI
Drum	PFI Drum5
Capacity	10000
Color	Coloring Age…
Size	S2
Flavor	F7
Level	7409

Index	Site	Drum	Capacity	Color	Size	Flavor	Level
0	Detroit, MI	FI Drum 1	20000	Coloring Agent40	S4	F5	0
1	Detroit, MI	FI Drum 2	20000	Coloring Agent40	S2	F5	0
2	Detroit, MI	FI Drum 4	20000	Coloring Agent40	S4	F9	0
3	Detroit, MI	FI Drum 5	20000	Coloring Agent40	S2	F5	0
4	Detroit, MI	FI Drum 6	20000	Coloring Agent40	S3	F8	0
5	Detroit, MI	FI Drum 7	20000	Coloring Agent36	S5	F9	0
6	Detroit, MI	FI Drum 8	20000	Coloring Agent36	S5	F5	0
7	Detroit, MI	FI Drum 3	20000	Coloring Agent40	S2	F5	0

The empty PI drums shown in this table would be loaded according to the lowest PI drum number.

For a specific loaded PI drum, it would be staged in front of the boxing machine and the level of jelly beans would satisfy the box packaging demand first. Had there be any remaining quantity in the PI drum after box packaging, it would be appended in the queue in front of the bagging machine and the level of the same drum would be updated in the next table. Otherwise, the empty PI drum would be returned to the pre finish operation.

Index	Site	Drum	Capacity	Color	Size	Flavor	Level
0	Detroit, MI	FI Drum 7	20000	Coloring Age…	S3	F2	1238

Index	Site	Drum	Capacity	Color	Size	Flavor	Level
0	Detroit, MI	FI Drum 6	20000	Coloring Agent1	S2	F6	0
1	Detroit, MI	FI Drum 7	20000	Coloring Agent1	S5	F5	898

The *FEL_builder* (a FEL function developed for simulation purpose) showed the real time update of the events simultaneously working in the simulation. The picture below provides a glimpse of several consecutive events. An RMI drum, would release jelly beans at time 2697.56 hrs and the classifier would finish size segmentation at time 2720.08 hrs. The first pre-finish of these sized jelly beans would be finished at 2725.66 hrs (contingent on the availability of flavoring tank from last operation) and the first packaging operation (Boxing) would be finished at 2748.29 hrs. The *FEL_builder* would also capture the timestamp when a new flavor would be completed introducing (2724 hrs and flavoring finishes at 2725.87 hrs for tank 2). These procedures would kept on running and contingent on the minimum requirement of PFI drums and classifier status, the next RMI would be released at time 2776.32 hrs.

```
              Event     Time
0    Releasing Drum  2697.56
1            Boxing  2697.73
2  Flavoring in tank1  2701.62
3       Classifying  2720.08
------------------------------
              Event     Time
0            Boxing  2697.73
1  Flavoring in tank1  2701.62
2       Classifying  2720.08
3    Bagging in mc1  2699.37
------------------------------
```

```
                  Event     Time
0    Flavoring in tank2  2724.72
1    Flavoring in tank3  2724.99
2    Flavoring in tank1  2725.66
3  Flavor Changing tank2     2724
4        Bagging in mc1  2724.96
------------------------------
                  Event     Time
0  Flavor Changing tank2     2724
1        Bagging in mc1  2724.96
2    Flavoring in tank3  2724.99
3    Flavoring in tank1  2725.66
4    Flavoring in tank2  2725.87
```

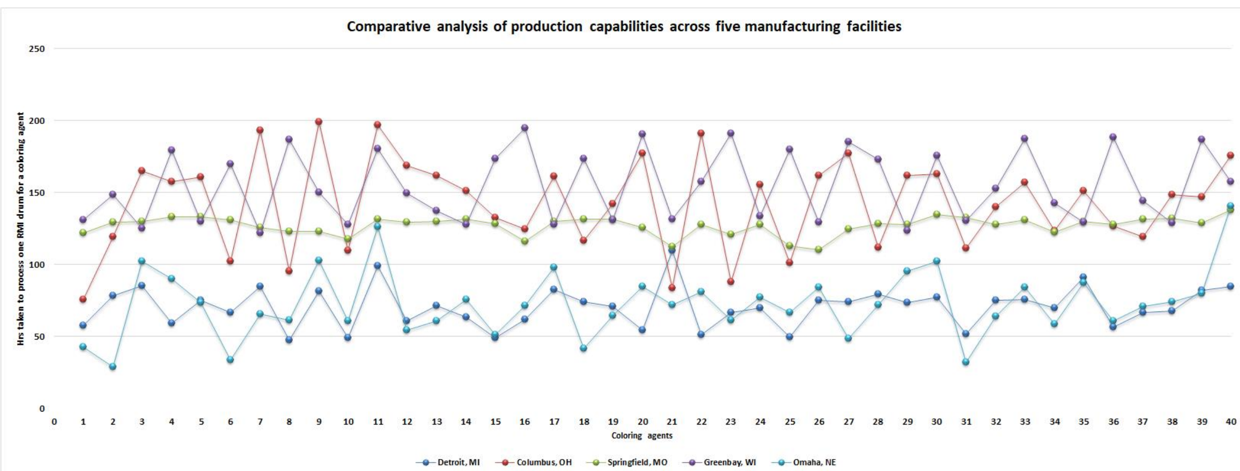
```
------------------------------
              Event     Time
0            Boxing  2748.29
1    Bagging in mc1  2748.61
2  Flavoring in tank1  2753.95
------------------------------
              Event     Time
0    Bagging in mc1  2748.61
1  Flavoring in tank1  2753.95
2    Bagging in mc1  2748.79
------------------------------
```

```
------------------------------
              Event     Time
0            Boxing  2776.25
1  Flavoring in tank1  2776.32
2    Bagging in mc1  2778.09
------------------------------
              Event     Time
0  Flavoring in tank1  2776.32
1    Bagging in mc1  2778.09
2    Releasing Drum  2776.32
3  Flavoring in tank1  2783.54
4            Boxing  2776.41
------------------------------
```

The FEL_builder would finish running once all the coloring agents would be processed. By that time, the specific hours needed for each coloring agent to finish processing would be stored in the memory. Based on this information, total production, possible lost sales, and cost could be derived for one particular site.

```
Coloring Agent36 takes 60.725289603955844 hours
Coloring Agent37 takes 70.96300183120593 hours
Coloring Agent38 takes 74.14594789682178 hours
Coloring Agent39 takes 79.68744569462251 hours
Coloring Agent40 takes 140.49195335991317 hours
--------------FEL is empty----------
Round0
Total cost : $1288987.364
Total production : 3922023.01b
Total Loss sales : 0.01b
```

To proceed with the optimization modelling, we decided to simulate each site with similar number of RMI drums and capacity, and then compare the production capabilities. The outcome showed that facilities in Detroit, MI and Omaha, NE could finish processing in significantly less time than the other facilities. Moreover, facilities in Columbus, OH, Greenbay, WI posed higher variability in total processing time for different coloring agents than the others. The facility in Springfield, MO appeared to be steadier in terms of variability in total processing time for different coloring agents.



Optimization modelling:

Sets:

Dr: set of all drums in all sites

M: set of sites

C: set of colors

Data:

$C_{b,n}$: cost of production of each box at site *n*; $\forall\, n \in M$

$C_{g,n}$: cost of production of each bag at site *n*; $\forall\, n \in M$

$\overline{a_{c,b}}$: Average amount of box produced with one pound of color *c* (given in demand distribution); $\forall\, c \in C$

$\overline{a_{c,g}}$: Average amount of bag produced with one pound of color *c* (given in demand distribution); $\forall\, c \in C$

W_b: Weight of each box

W_g: Weight of each bag

$$S_{d,m} = \begin{cases} 1, if\ drum\ d\ is\ in\ site\ m\ (before\ transportation) \\ 0, if\ otherwise \end{cases} ; \quad \forall\, d \in Dr,\ \forall\, m \in M$$

$d_{m,n}$: Distance between site *m* and *n*; $\forall\, m, n \in M$

C_T: Transportation cost = \$3.50/mile per 50,000 lbs.

L_T: Truck load capacity = 50,000 lbs.

Per^c : Shows what percentage of raw material should be assigned to color *c* which is equal to the percentage of total production of color c derived from order bank; $\forall\, c \in C$

$\overline{P_n}$: Average amount of production produced in site *n* (retrieved from simulation) $\forall\, n \in M$

Cap_d: Capacity of drum *d;* $\forall\, d \in Dr$

TT: Transportation threshold (maximum amount that can be transported from one site to another)

ε: Appropriate number to control the total amount of production of each color

α_n: An upper bound for total lost sales production (production after September 30th 5pm) in site *n*. In implementation we assumed that, α_n is equal to $0.1 \times \overline{P_n}$

Decision variables:

X_d: Amount of raw material assigned to drum d; $\forall\ d \in Dr$

$y_{c,n}$: Amount of raw material of color *c* at site *n* after transportation; $\forall\ c \epsilon C$, $\forall\ n \in M$

$$T_{d,n} = \begin{cases} 1, if\ drum\ d\ is\ transported\ to\ site\ n \\ 0, if\ otherwise \end{cases} \quad ; \quad \forall\ d \in Dr, \forall\ n \in M$$

$$J_{d,c} = \begin{cases} 1, if\ raw\ material\ of\ color\ c\ is\ assigned\ to\ drum\ d \\ 0, if\ otherwise \end{cases} \quad ; \quad \forall\ c \epsilon C, \forall\ d \in Dr$$

Optimization model (M1):

$$max \sum_{d \in Dr} X_d$$

$$\min \ \Big(\sum_{n \in M} [\frac{C_{b,n}}{W_b} (\sum_{c \in C} \overline{a_{c,b}}\ y_{c,n}) + \frac{C_{g,n}}{W_g} (\sum_{c \in C} \overline{a_{c,g}}\, y_{c,n})] \ +$$

$$\left[\sum_{n \in M} \sum_{d \in Dr} T_{d,n} \left(\sum_{n \in M} S_{d,m}\, d_{m,n}\right) X_d\right] * \frac{C_T}{L_T}\Big)$$

Subject to

1) $T_{d,m} \leq 1 - S_{d,m} \quad ; \quad \forall\ d \in Dr,\ \forall\ m \in M$

2) $\sum_{\forall\ n \in M} T_{d,n} \ \leq 1 \quad ; \quad \forall\ d \in Dr$

3) $\sum_{d \in Dr} J_{d,c}\, X_d \leq Per^c \sum_{d \in Dr} X_d + \varepsilon \quad ; \quad \forall\ c \in C$

4) $\sum_{d \in Dr} J_{d,c}\, X_d \ \geq Per^c \sum_{d \in Dr} X_d - \varepsilon \quad ; \forall\ c \in C$

5) $\sum_{c \in C} (y_{c,n}) - \overline{P_n} \leq \alpha_n \quad ; \forall\ n \in M$

6) $\sum_{d \in Dr} S_{d,n}\, J_{d,c}\, X_d + \sum_{d \in Dr} T_{d,n}\, J_{d,c}\, X_d - \sum_{d \in Dr} \sum_{m \in M} S_{d,n}\, T_{d,m}\, J_{d,c}\, X_d = y_{c,n} \quad ; \forall\ c \in C, \quad \forall\ n \in M$

7) $X_d \leq Cap_d \qquad \forall\, d \in Dr$

8) $\sum_{d \in Dr} S_{d,m}\, T_{d,n}\, X_d \ \leq TT \quad ; \forall\, m, n \in M$

9) $\sum_{\forall\, c \in C} J_{d,c} \ = 1\ ; \forall\, d \in Dr$

10) $X_d \ \geq 0 \quad \forall\, d \in Dr; \quad y_{c,n} \ \geq 0 \ \ \forall\, c \in C,\ \forall\, n \in M$

11) $T_{d,n} = \{0, 1\}\ \ d \in Dr\ \forall\, n \in M\ ; J_{d,c} = \{0, 1\}\ \ \forall\, c \in C, \forall\, d \in Dr$

$\blacksquare$

The first objective represents the total production and second represents the total cost including production costs and transportation costs.

where

- constraint 1 implies that a drum must not be transported to a site where it actually belong to
- constraint 2 shows that a drum can be transported at most one time, from the site it belonged to another
- constraint 3 and constraint 4 show that the total amount of raw material of color c that is assigned to all drums must be in a range that follows the percentage of the total amount of that specific color in the order bank
- constraint 5 shows that total amount of lost sales in terms of production in each site must not exceed specific upper bound (α)
- constraint 6 shows that the total amount of color *c* in site *n* after transportation equals to the total amount of that color in site *n* before transportation in addition to the amount that is received after transportation, deducting the amount it sends to other sites after transportation
- constraint 7 shows that the amount assigned to each drum must not exceed its capacity
- constraint 8 shows that the amount that is transported from one site to another must not exceed transportation threshold
- constraint 9 shows that each drum must contain only one color
- constraint 10 and 11 are non-negativity and binary constraints

Since constraint no. 6 consists cubic terms, hence, we modified the model to be in a quadratic format to approach the optimization model in Gurobi solver. Therefore, we defined new variables and added more constraints as followed:

New variable:

$$Z_{d,n,c} = \begin{cases} 1, if drum\ d\ carrying\ raw\ material\ of\ color\ c\ is\ transported\ to\ site\ n \\ 0, if\ otherwise \end{cases}$$

$\forall\ d \in Dr, \forall\ c \in C, \forall\ n \in M$

$$L_{d,n,c} = \begin{cases} 1, if drum\ d\ carrying\ raw\ material\ of\ color\ c\ is\ departed\ from\ site\ n \\ 0, if\ otherwise \end{cases}$$

$\forall\ d \in Dr, \forall\ c \in C, \forall\ n \in M$

New constraints:

$$T_{d,n} + J_{d,c} \leq Z_{d,n,c} + 1 \quad ; \quad \forall\ d \in Dr, \forall\ c \in C, \forall\ n \in M$$

$$Z_{d,n,c} \leq T_{d,n} \quad ; \forall\ d \in Dr, \forall\ c \in C, \forall\ n \in M$$

$$Z_{d,n,c} \leq J_{d,c} \quad ; \forall\ d \in Dr, \forall\ c \in C, \forall\ n \in M$$

These constraints implies that if drum *d* carrying raw material of color *c* **and** is transported to site *n*, the value of $Z_{d,n,c}$ would become one.

$$J_{d,c} + \sum_{n \in M} (S_{d,n}\, T_{d,m}) \leq L_{d,n,c} + 1 \quad ; \quad \forall\ d \in Dr, \forall\ c \in C, \forall\ n \in M$$

$$L_{d,n,c} \leq \sum_{n \in M} (S_{d,n}\, T_{d,m}) \quad ; \forall\ d \in Dr, \forall\ c \in C, \forall\ n \in M$$

$$L_{d,n,c} \leq J_{d,c} \quad ; \forall\ d \in Dr, \forall\ c \in C, \forall\ n \in M$$

These constraints implies that if drum *d* carrying raw material of color *c* **and** is departed from site *n*, the value of $L_{d,n,c}$ would become one.

The new quadratic model is as followed *(M2):*

$$max \sum_{d \in Dr} X_d$$

$$\min \ (\sum_{n \in M}[\frac{C_{b,n}}{W_b} \ (\sum_{c \in C} \overline{a_{c,b}} \ y_{c,n}) + \frac{C_{g,n}}{W_g} (\sum_{c \in C} \overline{a_{c,g}} \, y_{c,n})] \ +$$

$$\left[\sum_{n \in M} \sum_{d \in Dr} T_{d,n} \left(\sum_{n \in M} S_{d,m} \, d_{m,n}\right) X_d\right] * \frac{C_T}{L_T})$$

Subject to

1) $T_{d,m} \leq 1 - S_{d,m} \qquad ; \ \forall \ d \in Dr, \ \forall \ m \in M$

2) $\sum_{\forall \ n \in M} T_{d,n} \ \leq 1 \qquad ; \ \forall \ d \in Dr$

3) $\sum_{d \in Dr} J_{d,c} \, X_d \leq Per^c \sum_{d \in Dr} X_d + \varepsilon \qquad ; \quad \forall \ c \in C$

4) $\sum_{d \in Dr} J_{d,c} \, X_d \ \geq Per^c \sum_{d \in Dr} X_d - \varepsilon \qquad ; \forall \ c \in C$

5) $\sum_{c \in C} (y_{c,n}) - \overline{P_n} \leq \alpha_n \qquad ; \forall \ n \in M$

6) $\sum_{d \in Dr} S_{d,n} \, J_{d,c} \, X_d + \sum_{d \in Dr} Z_{d,n,c} \ \ X_d - \sum_{d \in Dr} L_{d,n,c} \ X_d = y_{c,n} \qquad ; \forall \ c \in C, \qquad \forall \ n \in M$

7) $X_d \leq Cap_d \qquad \forall \ d \in Dr$

8) $\sum_{d \in Dr} S_{d,m} \, T_{d,n} \, X_d \ \leq TT \qquad ; \forall \ m, n \in M$

9) $T_{d,n} + \ J_{d,c} \ \leq Z_{d,n,c} + 1 \quad ; \ \forall \ d \in Dr, \forall \ c \in C, \forall \ n \in M$

10) $Z_{d,n,c} \ \leq T_{d,n} \quad ; \forall \ d \in Dr, \forall \ c \in C, \forall \ n \in M$

11) $Z_{d,n,c} \ \leq J_{d,c} \quad ; \ \forall \ d \in Dr, \forall \ c \in C, \forall \ n \in M$

12) $J_{d,c} \ + \sum_{n \in M} (S_{d,n} \, T_{d,m}) \leq L_{d,n,c} + 1 \qquad ; \quad \forall \ d \in Dr, \forall \ c \in C, \forall \ n \in M$

13) $L_{d,n,c} \ \leq \sum_{n \in M} (S_{d,n} \, T_{d,m}) \quad ; \ \forall \ d \in Dr, \forall \ c \in C, \forall \ n \in M$

14) $L_{d,n,c} \leq J_{d,c}$; $\forall\, d \in Dr, \forall\, c \in C, \forall\, n \in M$

15) $\sum_{c \in C} J_{d,c} = 1$; $\forall\, d \in Dr$

16) $X_d \geq 0 \quad \forall\, d \in Dr; \quad y_{c,n} \geq 0 \;\; \forall\, c \in C, \;\forall\, n \in M$

17) $T_{d,n} = \{0,1\}\; d \in Dr \;\forall\, n \in M$; $J_{d,c} = \{0,1\} \;\forall\, c \in C, \forall\, d \in Dr; \quad Z_{d,n,c}, L_{d,n,c} = \{0,1\}$

$\forall\, d \in Dr, \forall\, c \in C, \forall\, n \in M$ ■

In order to approach multi-objective optimization problem, firstly we optimized first and second objectives based on constraints in M2 model (excluding constraint 5) to get some estimations about both objectives. The results are as followed (χ representing set of total decision variables):

Objective (f_i)	Minimum (f_i^{min})	Maximum (f_i^{max})
$f_1(\chi) = \sum_{d \in Dr} X_d$	0	45800000
$f_2(\chi) = \sum_{n \in M} [\frac{C_{b,n}}{W_b} (\sum_{c \in C} \overline{a_{c,b}}\, y_{c,n}) + \frac{C_{g,n}}{W_g} (\sum_{c \in C} \overline{a_{c,g}}\, y_{c,n})] + \left[\sum_{n \in M} \sum_{d \in Dr} T_{d,n} \left(\sum_{n \in M} S_{d,m}\, d_{m,n}\right) X_d\right] * \frac{C_T}{L_T}$	0	52807400 + 131250 = 52938650

Now, we can scale our objectives to get a single objective as followed *(M3):*

$$max \left(\frac{f_1(\chi) - f_1^{min}}{f_1^{max} - f_1^{min}}\right) - \left(\frac{f_2(\chi) - f_2^{min}}{f_2^{max} - f_2^{min}}\right)$$

Subject to

1) $T_{d,m} \leq 1 - S_{d,m}$; $\forall\, d \in Dr, \;\forall\, m \in M$

2) $\sum_{\forall\, n \in M} T_{d,n} \leq 1$; $\forall\, d \in Dr$

3) $\sum_{d \in Dr} J_{d,c} X_d \leq Per^c \sum_{d \in Dr} X_d + \varepsilon$; $\forall\, c \in C$

4) $\sum_{d \in Dr} J_{d,c} X_d \geq Per^c \sum_{d \in Dr} X_d - \varepsilon$; $\forall\, c \in C$

5) $\sum_{c \in C} (y_{c,n}) - \overline{P_n} \leq \alpha_n \qquad ; \forall\, n \in M$

6) $\sum_{d \in Dr} S_{d,n}\, J_{d,c}\, X_d + \sum_{d \in Dr} Z_{d,n,c}\;\; X_d - \sum_{d \in Dr} L_{d,n,c}\;\, X_d = y_{c,n} \qquad ; \forall\, c \in C, \qquad \forall\, n \in M$

7) $X_d \leq Cap_d \qquad \forall\, d \in Dr$

8) $\sum_{d \in Dr} S_{d,m}\, T_{d,n}\, X_d \;\leq TT \qquad ; \forall\, m, n \in M$

9) $T_{d,n} + J_{d,c} \;\leq Z_{d,n,c} + 1 \quad ; \forall\, d \in Dr, \forall\, c \in C, \forall\, n \in M$

10) $Z_{d,n,c} \;\leq T_{d,n} \quad ; \forall\, d \in Dr, \forall\, c \in C, \forall\, n \in M$

11) $Z_{d,n,c} \;\leq J_{d,c} \quad ; \forall\, d \in Dr, \forall\, c \in C, \forall\, n \in M$

12) $J_{d,c} \;+ \sum_{n \in M} (S_{d,n}\, T_{d,m}) \leq L_{d,n,c} + 1 \qquad ; \quad \forall\, d \in Dr, \forall\, c \in C, \forall\, n \in M$

13) $L_{d,n,c} \;\leq \sum_{n \in M} (S_{d,n}\, T_{d,m}) \quad ; \forall\, d \in Dr, \forall\, c \in C, \forall\, n \in M$

14) $L_{d,n,c} \;\leq J_{d,c} \qquad ; \forall\, d \in Dr, \forall\, c \in C, \forall\, n \in M$

15) $\sum_{c \in C} J_{d,c} \;= 1 \,; \;\forall\, d \in Dr$

16) $X_d \;\geq 0 \quad \forall\, d \in Dr; \quad y_{c,n} \;\geq 0 \;\; \forall\, c \in C, \;\; \forall\, n \in M$

17) $T_{d,n} = \{0, 1\}\; d \in Dr \;\forall\, n \in M\, ; J_{d,c} = \{0, 1\} \;\forall\, c \,\epsilon\, C, \forall\, d \in Dr; \quad Z_{d,n,c}, L_{d,n,c} = \;\{0, 1\}$

$\forall\, d \in Dr, \forall\, c \in C, \forall\, n \in M$ ■

We used the following procedure to solve the problem:

- First, we solved model M3 excluding the constraint number 5 (this constraint requires estimation that comes from the simulation results) with Gurobi.
- Then, we used the output of the optimization model (like amount of raw material at each RMI drum in each site, color assigned to each RMI drum and etc.) as inputs to the simulation model.
- Afterwards, we simulated the system using discrete event simulation in python to estimate $\overline{P_n}$.
- At the next step, we solved M3 including the constraint number 5 using Gurobi solver with time limitation of 8 hours. Then, we reported the final solution.
- At the final step, we used the previous solution of the optimization model to run in the simulation again to get some estimations to complete the results.

Analytics Solution and Results

Based on the simulation based optimization, we found out that raw material flow would be maintained only in the following directions:

Table for transportation costs:

	Green Bay	Omaha	Springfield	Columbus	Detroit
Green Bay					
Omaha					
Springfield					
Columbus	320000 lbs/ $11872.00				
Detroit		300000 lbs / $15,330.00	267844.62 lbs / $14,061.75		

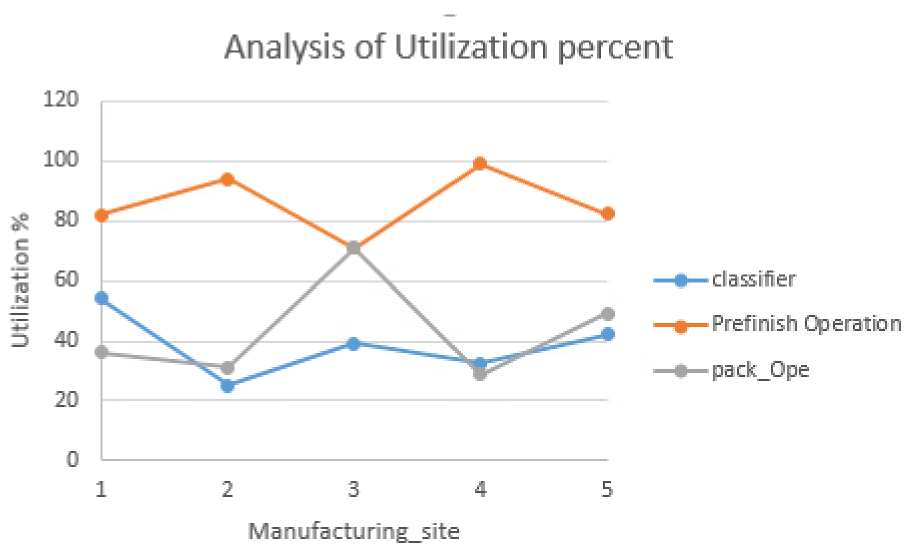
Table for production amounts, days, and costs:

Manufacturing Site	Total Production (lbs)	Total number of days to complete production	Total Production Cost ($)	Lost sale (lbs)
Green Bay	3233999.99	203.42	4273179.49	330000
Omaha	5742453.99	214.66	5928666.81	874158.62
Springfield	2569824.33	135.33	4800225.14	0
Columbus	3535284.9	199.65	5926546.31	325713.22
Detroit	8550168.11	252.36	9402549.84	2419999.99

We can see that as the number of days increases, the lost sales increase as well and so does the cost.

Table for determining the bottlenecks at each site:

Manufacturing Site	Classifier (Utilization)%	Pre-finish Operation (Utilization) %	Pack Operation (Utilization) %	Bottleneck (Operation name)
1. Green Bay	54.31	81.91	36.15	Pre-finish Operation
2. Omaha	25.27	93.91	31.25	Pre-finish Operation
3. Springfield	39.31	71.03	70.91	Pre-finish Operation/ Pack Operation
4. Columbus	32.41	99.18	28.90	Pre-finish Operation
5. Detroit	42.35	82.36	49.12	Pre-finish Operation



Insights for JB Manufacturing Executive Team:

- Since pre-finish operation is the bottleneck in all the facilities, to improve the situation, JB Manufacturing could
    - Increase the number of pre-finish equipments
    - Increase the processing rates of pre-finish operations by improving the existing machines
- Increase the number of equipments at each operation would significantly reduce the lost sales, which consequently would increase the profit
- The machine setup can be devised in such a way that travel time of drums in between the stations is minimum
- Machine downtime and maintenance time should be considered in the optimization in order to make the solution more realistic