Report: Master Production Scheduling and Delivery Allocation Project for Anker Innovation

Zhang Haoran
School of Data Science
The Chinese University of HongKong, Shenzhen
122090735@link.cuhk.edu.cn

Zeng Yang
Financial Engineering Program
The Chinese University of HongKong, Shenzhen
122090714@link.cuhk.edu.cn

1 Introduction

The optimization of supply chain operations is a critical endeavor for global enterprises such as Anker Innovation, who manages complex, large-scale production and distribution networks. This report presents a comprehensive study on the Master Production Scheduling (MPS) and Delivery Allocation (DA) models developed to enhance the efficiency of Anker's supply chain. To address the multifaceted challenges of Anker's operations, we decompose the supply chain into two interconnected optimization problems. The MPS model focuses on upstream production planning, determining optimal order quantities, factory assignments, etc.. Conversely, the DA model addresses downstream logistics, optimizing the allocation of on-hand inventory and incoming purchase orders (POs) to meet regional demand orders. By integrating these models, our approach ensures seamless coordination between long-term production strategies and real-time distribution tactics, tailored to Anker's large-scale operations. Utilizing rigorous Mixed Integer Programming (MIP) and Linear Programming (LP) formulations, carefully defined model size reduction techniques, and computational optimization, this framework delivers actionable solutions that enhance cost efficiency, service reliability, and adaptability to dynamic market demands. Moreover, for each model, we have established the entire workflow, from data pre-processing and model building to model output. Our code is accessible via link `https://github.com/ZH-Haoran/Anker-MPS-Project`.

2 Master Production Scheduling

2.1 Problem Description

The Master Production Scheduling (MPS) problem constitutes a pivotal upstream decision-making process within Anker's supply chain. Upon receiving demand forecasts, the MPS model is tasked with determining optimal ordering decisions to effectively balance inventory holding costs with the imperative of meeting customer demand. A significant challenge for Anker emerges, particularly under conditions of constrained production capacity, where the company struggles to quantitatively assess whether to place orders in advance—thereby incurring additional inventory costs—or to forego fulfilling certain demands, risking potential revenue losses.

The model must exhibit sufficient flexibility to accommodate Anker's unique business scenarios, such as new product launches, product discontinuation, and proactive inventory stocking in preparation for major promotional events. The complexity of this problem is further amplified by the competition for production capacity across multiple factories. This arises from the fact that a single stock-keeping unit

(SKU) may be manufactured in multiple factories, while each factory has the capability to produce multiple SKUs, creating a multifaceted decision-making landscape.

Operationally, the MPS model is required to generate weekly decisions specifying which factory should produce each SKU and the corresponding order quantities. The model will be implemented in a rolling horizon framework, executed either on a weekly basis or in response to significant shifts in demand forecasts. Decisions for the current week will directly inform actual ordering quantities, while projections for subsequent weeks will serve as a critical foundation for inventory planning and early warning systems to mitigate potential supply chain disruptions.

2.2 Methodology

To deal with the problems described in Sec. 2.1, we develop a Mixed Integer Programming (MIP) operation model. We list all the notations for our MPS model in Table 1. In this section, we well first introduce the model with the meaning of objective function and each constraint (Sec. 2.2.1) and then demonstrate how it is compatible with the aforementioned special scenarios (Sec. 2.2.2).

2.2.1 MPS Model

This operation model aims to balance the inventory cost and the possible profit loss due to unmet demand. To optimize these objective, we propose the objective function of MPS model to be

$$\textit{Minimize} \quad \sum_{p\in\mathcal{P}} \left(\sum_{t=1}^{T} h_p \cdot I_{p,t} + \sum_{t=SLA_p^S}^{T} y_p \cdot u_{p,t} + \sum_{t=SLA_p^S}^{T} w_p s_{p,t} \right)$$

where the first and second term refers to the total inventory cost and the total profit loss of unmet demand, respectively, and the last term is the penalty for unmet target inventory, motivating the model to fulfill some personalized requirements such as safety stock level.

Factory Capacity Constraint

$$\sum_{\substack{p\in\mathcal{P}_f \\ \tau+SLA_p^S-SLA_p^T-1=t}} x_{f,p,\tau} \cdot o_{f,p,t} \le C_{f,t} - \sum_{\substack{p\in\mathcal{P}_f \\ \tau-SLA_p^T-1=t}} \widetilde{I}_{f,p,\tau} \cdot o_{f,p,\tau}, \quad \forall f \in \mathcal{F}, t \in \mathcal{T}$$

where the right-hand side shows the remaining capacity of factory f in week t and the left-hand side represents the corresponding capacity that will be occupied by the orders placed in this round of the plan. The remaining capacity is calculated by subtracting the capacity occupied by the purchased orders (POs, which have been determined before this model round) from the total normalized capacity. Additionally, the capacity occupation of each order is equal to the quantity of SKUs included in the order multiplied by its unit capacity occupation. Notably, since we are not able to know exactly when the factory will schedule our production, model are estimating the time the factory's capacity is occupied based on a principle that "the week the capacity is occupied is the week prior to the week the order starts to be transported".

Minimum Order Quantity (MOQ) Constraints

$$x_{f,p,t} \ge Q_p \cdot z_{f,p,t}, \quad \forall f \in \mathcal{F}, p \in \mathcal{P}_f \cap \mathcal{P}_Q, t \in \mathcal{T}$$
$$x_{f,p,t} \le M \cdot z_{f,p,t}, \quad \forall f \in \mathcal{F}, p \in \mathcal{P}_f \cap \mathcal{P}_Q, t \in \mathcal{T}$$

In these constraints, we require the order quantity $x_{f,p,t}$ must be larger or equal to MOQ when the model has decided to place order (i.e., $z_{f,p,t} = 1$); otherwise, they will become $0 \le x_{f,p,t} \le M$, implying that the MOQ constraints are totally relaxed.

Target Inventory Level Constraint

$$I_{p,t-1} + \widetilde{I}_{p,t} + \sum_{f\in\mathcal{F}_p} \sum_{\substack{\tau\in\mathcal{T}: \\ \tau+SLA_p^S=t}} x_{f,p,\tau} + s_{p,t} \ge R_{p,t}, \quad \forall p \in \mathcal{P}, t \in \mathcal{T}$$

where the left-hand side refers to the available inventory, including three components: (i) remaining inventory of the previous week; (ii) the POs that will arrive in this week and; (iii) the orders placed

by this round of plan and will arrive in this week. Additionally, $s_{p,t}$ is a slack variable with the meaning of unmet target inventory quantity, preventing from infeasibility when the capacities of all the factories are insufficient. The right-hand side is the target inventory level that can be determined by needs. Generally, it stands for safety stock level, calculated by

$$R_{p,t} = \sum_{t'=t}^{\min\{t+S_p,|\mathcal{T}|\}} D_{p,t'}$$

Moreover, it can also incorporate the need of advance stocking. We will provide a further explanation on Sec. 2.2.2.

Inventory Balance Constraints

$$I_{p,t} = I_{p,t-1} + \widetilde{I}_{p,t} + \sum_{f\in\mathcal{F}_p} \sum_{\substack{\tau\in\mathcal{T}:\\ \tau+SLA_p^S=t}} x_{f,p,\tau} - D_{p,t} + u_{p,t}, \qquad \forall p\in\mathcal{P}, t\in\mathcal{T}$$

$$u_{p,t} \le M\cdot(1-e_{p,t}), \qquad \forall p\in\mathcal{P}, t\in\mathcal{T}$$

$$I_{p,t} \le M\cdot e_{p,t}, \qquad \forall p\in\mathcal{P}, t\in\mathcal{T}$$

The first equality is the main body of inventory balance constraint. That is, if the available inventory is sufficient, part of it will be used to meet the demand and the other will become the remaining inventory. Otherwise, the slack variable unmet demand quantity $u_{p,t}$ will be activated and the available inventory will equal to the sum of demand and unmet demand quantity.
The another two inequalities aim to enforce available inventory to be completely used for demand rather than be kept to meet the target inventory level in the next week. Specifically, when the demand of SKU p in week t is fully satisfied ($e_{p,t}=1$) then we require that there is no unmet demand quantity. In contrast, when the model decides to give up the demand ($e_{p,t}=0$), the remaining inventory must be zero as well. To conclude, they together ensure that the model cannot control how much inventory is used to meet the current week's demand by adjusting the "unmet demand quantity $u_{p,t}$" variable. This is because we only want the model to optimize ordering decisions, rather than balance demand fulfillment rate and target inventory level.

Discontinuation Constraint

$$x_{f,p,t} = 0, \quad \forall p\in\mathcal{P}_{off}, \forall f\in\mathcal{F}_p, \forall t > \bar{t}_p$$

where we require the order quantity of the to-be-discontinued SKUs to be zero after the production stop week. Since this constraint is for the discontinuation scenario, we defer a more in-depth explanation of it to Sec. 2.2.2.

2.2.2 Special Scenarios

In this section, we will elaborate on how our model can be compatible with three specific business scenarios listed in Sec. 2.1.

Product Launch. When a product is about to launch, we may already have had the demand prediction of it after the launch week. Thus, the MPS model is responsible to prepare products by ordering in advance. This scenario is equivalent to a situation where demand is zero at the beginning and then becomes non-zero after launch, so the model can automatically optimize the stocking process without any modification.

Product Discontinuation. If a product is going to be discontinued, the company will terminate its production in a specific week beforehand. At this point, the remaining inventory and POs will be used to meet market demand before the product is officially discontinued. Therefore, in the discontinuation constraint in Sec. 2.2.1, we enforce all order quantities after the production stop week to be 0. At this time, because the model does not want to bear the cost of lost demand, it will make production scheduling decisions in advance, which perfectly aligns with the effect we want to achieve.

Advance Stocking for Major Promotions. During major promotions, such as Double Eleven, factory capacity may experience significant bottlenecks, and at the same time, there will be a business need for early replenishment. At this point, the model needs to consider placing orders earlier for advance stocking. We noticed that the requirements for pre-stocking ahead of major promotions and safety stock requirements share the same nature, i.e., we aim for the available inventory in a given week to reach a certain level. However, when actually running the inventory balance equation, we only subtract the actual predicted demand. Therefore, we cannot directly add safety stock and pre-stocking quantities to the predicted demand. Instead, we need to introduce a variable $R_{p,t}$, representing the "target inventory level," to combine safety stock with potential pre-stocking quantities. In this scenario,

$$R_{p,t} = \sum_{t'=t}^{\min\{t+S_p,|\mathcal{T}|\}} D_{p,t'} + \widetilde{D}_{p,t}$$

where $\widetilde{D}_{p,t}$ refers to the pre-stocking quantities in week t. Additionally, by adjusting the calculation logic of $R_{p,t}$, we can effectively extend the model to more similar business scenarios.

2.2.3 Model Scale Reduction

Even though the model has been established, due to the large scale of Anker's actual business data, we have to employ some techniques to reduce the model's size and accelerate solving.

Scale Reducing Technique. We mainly adopt three tricks to significantly reduce the size of MPS model:

1. Separate the data from supply centers Y and Z and solve them independently, as in Anker's business, the data from the two supply centers can be completely decoupled and do not affect each other.
2. Declare only the necessary variables and constraints. As shown in the model in Sec. 2.2.1, we carefully selected the sets to be constrained by the corresponding constraints when establishing them, and only declared the decision variables that the model actually uses.
3. Pre-aggregate the capacities of all production lines in the same factory as the factory capacity to reduce the factory dimension from $|\mathcal{F}| \times |\mathcal{L}|$ to $|\mathcal{F}|$, where $|\mathcal{L}|$ stands for the total number of production lines in all factories.

Notably, these techniques do not compromise the model's solution accuracy, as no approximation method is being used.

Model Scale Comparison. Initially, we consider the triplet `(supply center, factory, production line)` as one factory object and the pair `(supply center, SKU id)` as one SKU object. After data cleaning, we obtain 3408 SKU objects and 157 factory objects. Without the above techniques, the number of non-zero elements in our model reaches over 10 million, with 170,000 integer variables, which is an unacceptable scale. Local memory cannot even support completing the solution. However, after the reduction of model size, the model parameter counts for supply centers Y and Z have both been reduced to the order of 100,000. When solved using the Cardinal Optimizer (COPT) on the Center for Optimization Algorithm Platform (COAP), the two models achieved relative gap of 0.329% and 1.737% within 300 seconds, with objective function values of 7.17×10^7 and 1.55×10^8, respectively. In large-scale real-world scenarios, such errors are already acceptable.

Optimization Acceleration. Since integer variables lead to an exponential increase in computational complexity during solving, the number of integer variables is the main bottleneck in optimization. Considering that the impact of slight errors in order quantities is very limited on objective function, we relax the order quantity variables from integer to continuous types and round them to integers after solving as the final order decision. In this way, we can completely solve these two models (for supply center Y and Z, respectively) within 200 seconds, meaning that the relative gap reaches 0.01%. We will further justify this strategy in Sec. 2.3.

2.3 Numerical Results

To efficiently check the model's accuracy and performance, we have visualized and selected some representative SKUs, and their inventory curves are shown in the following figures, where blue lines are the available inventory in each week, orange lines are the required/target inventory level, green lines are the demands and the yellow line is the arrived order quantities which are purchased in these round. Notably, the red dashed line represents the supply weeks, after which the results of this round of model optimization take effect, because the products ordered on the first day will arrive in this week.

The first three figures show three representative SKUs and all of them is solved using continuous-type decision variables. Fig. 1 shows a perfect-operated SKU, which satisfies the target inventory level during the whole plan weeks. Fig. 2 illustrates a typical case of "oversupply", and indeed, the model does not place any orders. However, the order for Fig. 3 is one with relatively limited production capacity. Therefore, after weighing the inventory cost against the penalty for not meeting the target inventory, no ordering decision was made. It only met the expected demand but did not fully satisfy the target inventory. Meanwhile, Fig. 4 displays the inventory curve for the same SKU3356 when using integer variable settings (recall that the model is not fully solved at this point). It shows that the model opts to place an order for 300 units of the SKU (the minimum order quantity of it) to arrive in Week 16 when there is available capacity. However, it can be easily verified that it is suboptimal. After Week 16, the demand for this product is very low, meaning most of these 300 units will incur additional inventory costs, making it better to forgo this portion of demand entirely. The comparison between Fig. 3 and Fig. 4 also demonstrates the rationality of our strategy to relax the decision variables into a continuous form to obtain the optimal solution.

To conclude, through mathematical modeling, scale reduction, solution acceleration, and visualization, we have obtained the optimal solution for the MPS problem and conducted relevant validation, while accommodating Anker's various practical business needs.

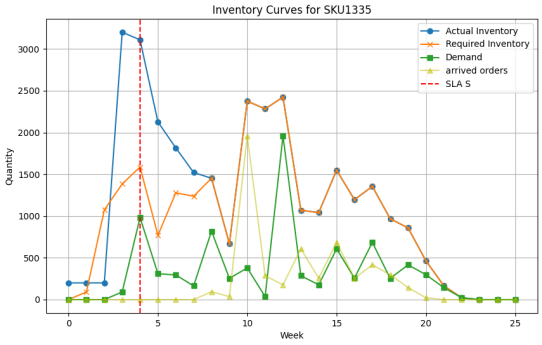

Figure 1: SKU1335 Inventory Curve

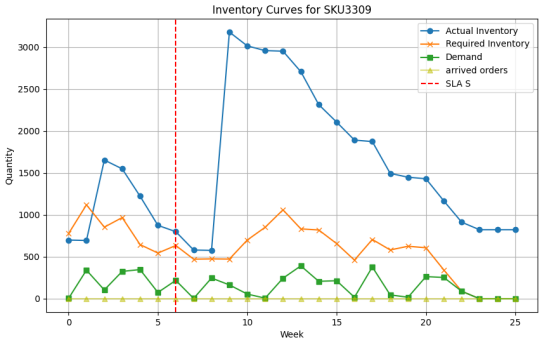

Figure 2: SKU3309 Inventory Curve

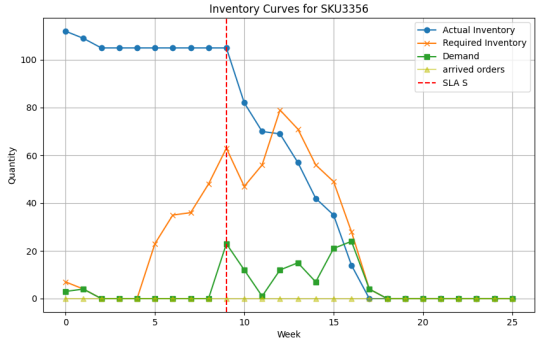

Figure 3: SKU3356 Inventory Curve (Continuous Variable)

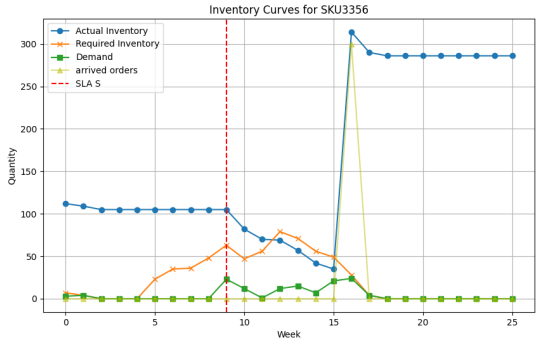

Figure 4: SKU3356 Inventory Curve (Integer Variable)

3 Delivery Allocation

3.1 Problem Description

Delivery Allocation (DA) Model is a downstream task of Anker's supply chain. Effective logistics downstream operations are crucial for Anker to fulfilling subdistributor demand while maintaining cost efficiency and service quality. Following the execution of Master Production Scheduling (MPS), Anker face a second-level operational challenge: how to allocate incoming (POs) and existing product resources (inventory) to a network of geographically distributed subdistributors, each issuing demand orders with specific timing and quantity requirements. This leads to what we define as the Delivery Allocation (DA) Problem. The DA model is designed to determine, for each demand order, the optimal sourcing decisions from either on-hand inventory or scheduled Purchase Orders (POs), the quantity to allocate, and the precise delivery timing—while minimizing penalties associated with late fulfillment. Unlike the MPS model, which is concerned primarily with long-term planning and upstream production decisions, the DA problem operates at a finer temporal and spatial resolution, involving real-time delivery planning under tight logistical constraints.

The Delivery Allocation (DA) problems is characterized by a high degree of operational complexity. A primary consideration is the region capacity, which imposes an upper bound to the total volume of SKUs that can be delivered to each region during any given time period. This constraint models practical limitations such as labor availability, docking station throughput in regional warehouses, and regional subdistributor level. When demand is temporally concentrated or unevenly distributed across regions, satisfying all orders without breaching these limits becomes a non-trivial allocation challenge. Another fundamental difficulty arises from the inventory capacity constraint and PO capacity constraint, which ensures that the allocation from on-hand inventory or up-coming POs do not exceed the available quantity. Given that inventory and POs are finite and are typically shared

across multiple orders and regions, the model must decide judiciously how to ration the distribution among competing demands, potentially prioritizing orders with tighter deadlines or higher penalties for delay. Furthermore, the PO Arrival Time Constraint prohibits any usage of a PO prior to its scheduled arrival. This temporal dependency requires the model to reason not only about demand urgency but also about the synchronization of supply streams. Arguably the most complex aspect is captured in the Demand Balance Constraint, which requires that the total allocated quantity for each order — whether from inventory or future POs — must meet the full demand, or else be recorded as a shortfall through a slack variable. The presence of this slack is associated with delay penalties through the EPD and RPD Penalty Functions, which track how late the unmet quantity is relative to the customer's requested and the planned delivery dates (EPD & RPD). These penalties are time-sensitive and weighted by SKU-specific coefficients, allowing the model to reflect differentiated service levels and business priorities. However, this feature also introduces significant heterogeneity into the objective function, making the trade-offs between competing orders highly non-linear in practice. In particular, delays in high-priority SKUs may carry disproportionately large penalties, compelling the model to make trade-off between global efficiency and local service-level commitments.

3.2 Methodology

In the following section, we will demonstrate our formulation of the problem described in Sec. 3.1. And all the parameter's notation for DA model is listed in Table 2. We developed a Linear Programming (LP) model, and detailed objective function and constraints are illustrating below:

3.2.1 DA Model

Objective Function

This model seek to allocate on-hand inventory and scheduled POs to reasonably satisfy regional demand orders. The following objective functions is proposed

$$\textit{Minimize} \quad \sum_{r \in \mathcal{R}} \sum_{o \in \mathcal{O}_r} \left(Delay_o^{RPD} \cdot u_o + Delay_o^{EPD} \cdot v_o \right)$$

The objective is to minimize total delivery penalties across all demand orders. Penalty is composed of delivery delays relative to the RPD (Requested Packing Date) and EPD (Estimated Packing Date), respectively, with delay costs weighted by order-specific priority coefficients.

Region Capacity Constraint

$$\sum_{o \in \mathcal{O}_r} \left(x_{o,t} + \sum_{\substack{j \in \mathcal{J}: \\ \tau_j \leq t}} y_{j,o,t} \right) \leq C_{r,p,t}, \quad \forall r \in \mathcal{R}, t \in \mathcal{T}, p \in \mathcal{P}$$

where the left-hand side refers to the total allocation quantity from inventory and POs to region r at time t. This constraint ensure that the total quantity of product p delivered to all orders in region r at time t does not exceed the regional capacity $C_{r,p,t}$. The delivery can originate from inventory ($x_{o,t}$) or from eligible POs ($y_{j,o,t}$) that already arrived at time t. This reflect the last-mile delivery or regional warehousing logistics.

Inventory Capacity Constraint

$$\sum_{\substack{o \in \mathcal{O}_r \\ p = \mathcal{P}_o}} x_{o,t} \leq I_p, \quad \forall\, p \in \mathcal{P}, r \in \mathcal{R}$$

where the left-hand side shows the total amount of product p that comes from inventory. This constraint limits the amount of product p that can be assigned from inventory to orders which demand product p. It ensures that the allocation from inventory do not exceed available stock I_p. This reflects the physical inventory constraints at the warehouse level.

PO's Capacity Constraint

$$\sum_{\substack{o\in\mathcal{O}_r\\ \mathcal{P}_o=\mathcal{P}_j}} y_{j,o,t} \le Q_j, \quad \forall\, j\in\mathcal{J}, r\in\mathcal{R}$$

where the left-hand side is the total amount of allocation that comes from PO j. This constraint restricts the total quantity assigned from PO j to all relevant demand orders at time t does not exceed the PO's quantity Q_j. It enforces correct capacity tracking for incoming shipments.

PO's Arrival Time Constraint

$$y_{j,o,t} = 0, \quad \forall j\in\mathcal{J}, o\in\mathcal{O}_r, r\in\mathcal{R}, t\in\mathcal{T}, t<\tau_j$$

where the left-hand side refers to the amount assigned from PO j before its arrival time τ_j. This constraint ensures that a PO j cannot be used to fulfill any demand order o before its arrival time. This models the fundamental timing logic that goods cannot be allocated from a shipment that has not yet been received.

Demand Balance Constraint

$$\sum_{t'=\mathrm{RPD}_o}^{t}\left(x_{o,t'} + \sum_{\substack{j\in\mathcal{J}:\\ \tau_j\le t'}} y_{j,o,t'}\right) + z_{o,t} = d_o, \quad \forall\, o\in\mathcal{O}_r, r\in\mathcal{R}, t\in\mathcal{T}, t\ge \mathrm{RPD}_o$$

where the left-hand side refers to three terms, (i) cumulative allocation from inventory, (ii) cumulative allocation from POs, (iii) unmet demand. This constraint ensures that for each demand order o, the cumulative allocation from both inventory and POs up to time t, plus the unmet demand $z_{o,t}$ must equal to the total demand quantity d_o. The slack variable $z_{o,t}$ captures the shortfall in fulfillment, which subsequently lead to penalty in our objective function.

RPD & EPD Penalty Function

$$Delay_o^{EPD} = \sum_{\substack{t\in\mathcal{T}:\\ t>\mathrm{EPD}_o}} (t-\mathrm{EPD}_o)\cdot z_{o,t}, \quad \forall r\in\mathcal{R}, o\in\mathcal{O}_r$$

$$Delay_o^{RPD} = \sum_{\substack{t\in\mathcal{T}:\\ t>\mathrm{RPD}_o}} (t-\mathrm{RPD}_o)\cdot z_{o,t}, \quad \forall r\in\mathcal{R}, o\in\mathcal{O}_r$$

where the right-hand side refers to the calculation of penalty function. These two equations calculate the EPD and RPD delay fulfillment by summing the unmet demand $\times$ delay days, respectively. The longer the time exceed the deadlines and the larger the unmet demand quantity the greater the penalty. This penalty function enforces the model for earlier and smaller delivery regarding the penalty. In addition, the penalty function could change based on different business scenarios, for example, we can use a quadratic function for differentiated prioritization of demand orders and its EPD, RPD.

3.3 Model Scale Reduction

Original Scale

After cleaning the data given by Anker, the problem include **2017 demand orders**, **502 POs**, and a time horizon of **44 weeks**. Thus, the full Delivery Allocation (DA) model would contain millions variables. Specifically, the three-dimension decision variable $y_{j,o,t}$, capturing allocation from PO j to order o at time t, will have size that exceed 44 millions ($2017\times 502\times 44$), which result in a prohibitively problem size. Hence, the implementation of full variable space will lead to computation intractability and memory outflows during initial construction of the model (Figure 5).

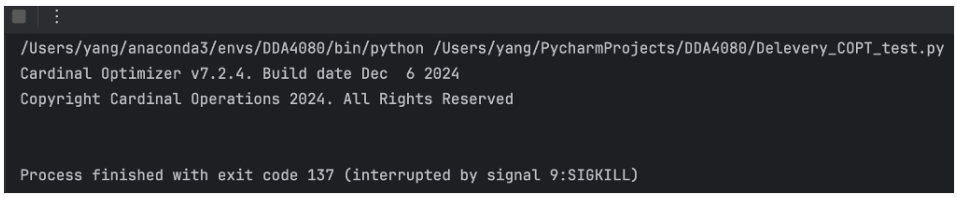

Figure 5: Failure of Construction the Original Problem

Scale Reducing Technique

To address the size problem, we employed a multi-step model reduction strategy, by reconstruct the time set, decomposed demand orders, partitioned into sub-problems.

1. **Reducing Time Set:**
   We proposed the following reconstructed time set:

   $$\mathcal{T} = \{RPD\} \cup \{EPD\} \cup \{PO's\ Arrival\ time\}$$

   The time set could be safely reduced by only retaining the time points corresponding to RPDs, EPDs, and PO's arrival dates. The operation is harmless to the optimality of the problem since once a demand order is raised, the transaction will be fulfilled at RPD or EPD if current quantity is sufficient, otherwise, the transaction will occur at the time when new a PO comes in.

2. **Decomposing Multi-SKU Demand Orders:**
   We decomposed the multi-SKU demand orders into individual sub-orders, that each containing one SKU, by adding a fulfillment order row.

   $$PPJP202412240347 \rightarrow (PPJP202412240347,\ 1),\ (PPJP202412240347,\ 2)$$

   Such that each order code plus fulfillment order row forming a unique code that determine the demand orders with one SKU only. This transformation avoid unnecessary cross-product expansion in the model formulation, and allow the model to focus on SKU-level allocation decision independently.

3. **Partitioning Final Problem:**
   Finally, we partitioned the entire problem into independent sub-problems by SKU, enabling parallel computation and isolating interdependencies. This operation significantly decrease the variables count and memory usage per sub-problem.

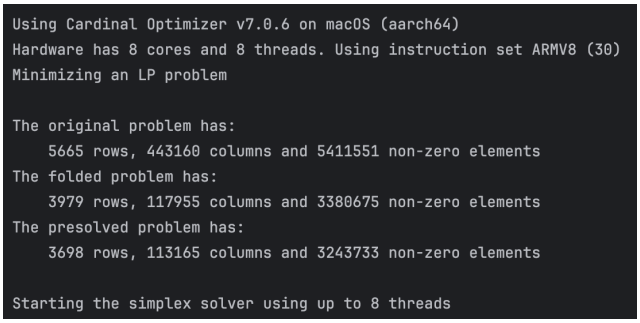

Figure 6: Reduced Problem Size

As a result of the scale reducing operations, the total model size was reduced from approximately 40 millions variables to about 5 millions (Figure 6), allowing the problem to be solved efficiently under 30 seconds.

3.4 Numerical Results

The Delivery Allocation (DA) model demonstrates robust performance in handling large-scale, time-varing task. By utilizing COPT solver, the model achieves a solution time of under **30 seconds**, and a on-time-delivery-rate of **94.45%**, with only 112 out of 2017 demand orders experiencing delay.

We will illustrate our result through **SKU220** for clarity:
As seeing in Figure 7, our model completes the allocation with zero EPD penalty. In contract, Figure 8 simulates a naive way of assigned the products by the coming order of the demand order. EPD penalty is incurred at the very last two demand orders, which implies that the effectiveness of our model on refining Anker's allocation strategy, ensuring better cost efficiency in real world supply chain operation.

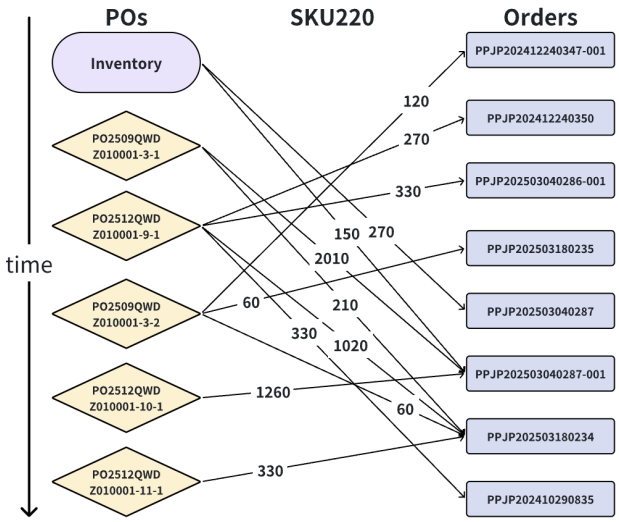

Figure 7: Result of DA Model (SKU220)

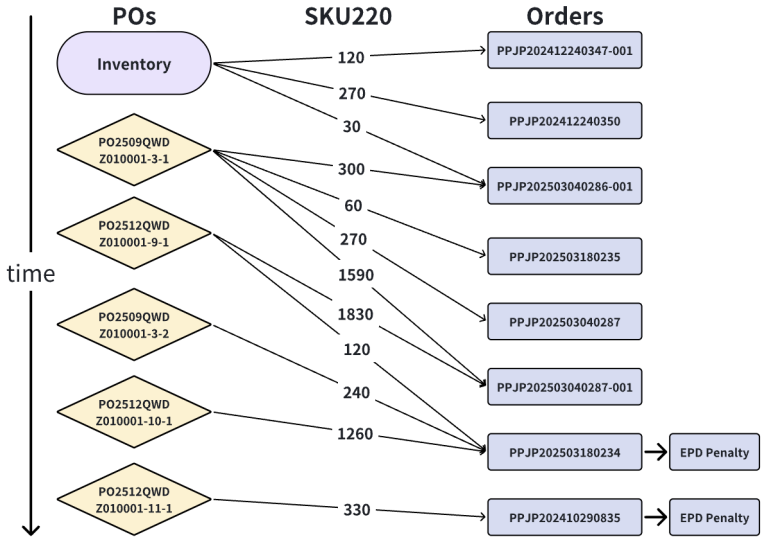

Figure 8: Result of Naive Method (SKU220)

4 Conclusion

This project presents a comprehensive two-stage optimization framework aimed at improving supply chain efficiency and delivery responsiveness for Anker Innovation. By integrating both upstream and downstream decision-making through the Master Production Scheduling (MPS) model and the Delivery Allocation (DA) model, we provide a unified approach that addresses the full spectrum of challenges inherent in production and logistics planning.

The MPS model determines 26 weeks production quantities, aiming to minimize total cost, including inventory holding, lost sales, and the penalty of unmet target inventory. It accounts for realistic business scenarios such as product launches, discontinuations, and advance stocking. Our model could effectively reach an optimal solution within 200 seconds and the solution has been verified via visualization.

The DA model assigns allocation for over 2000 demand orders, more than 500 POs, with a time period of about 1 yaer, trying to minimize delay penalties weighted by SKU-level priority coefficients. The model consider complex constraints including regional capacity, PO availability, Inventory level, and delivery timing (RPD and EPD). The model is able to find an optimal solution within 30 seconds, with a on-time-delivery-rate of 94.45%.

Together, the two models form a complete, data-driven decision framework that effectively bridges production and delivery planning. Our results demonstrate that the MPS model ensures efficient production schedules under uncertainty with varying scenario, while the DA model are able to provide a better allocation strategy compared to Anker's current usage. Overall, our work provides a scalable, flexible, and implementable optimization solution for large-scale supply chain coordination.

5 Contributions and Roles

The initial modeling of both two models was jointly completed by Zhang Haoran and Zeng Yang. Additionally, Zhang Haoran and Zeng Yang were primarily responsible for the later development of the MPS model and the DA model, respectively, including code implementation, model refinement, and strategies formulation for model scale reduction.

Table 1: Parameters and Variables of the MPS Model

Sets	
$\mathcal{T}$	Total planning weeks, indexed by t.
$\mathcal{P}$	SKU set, indexed by p.
$\mathcal{F}$	factory set, indexed by f.
$\mathcal{P}_f$	set of SKUs that can be produced at factory f, indexed by p.
$\mathcal{P}_Q$	set of SKUs with nonzero MOQ, indexed by p.
$\mathcal{P}_{off}$	set of SKUs that are about to be discontinued, indexed by p.
$\mathcal{F}_p$	set of factories that are able to produce SKU p, indexed by f.
$\mathcal{L}_f$	set of production lines in factory f, indexed by l.
Parameters	
$D_{p,t}$	demand of SKU p in week t.
$C_{f,l,p,t}$	production capacity of line l at factory f in week t for SKU p.
$C_{f,p,t}$	production capacity of factory f in week t for SKU p, calculated by $C_{f,p,t} = \sum_{l \in \mathcal{L}_f} C_{f,l,p,t}$.
$C_{f,t}$	normalized capacity of factory f in week t, calculated by $C_{f,t} = \max_{p \in \mathcal{P}_f} C_{f,p,t}$.
$o_{f,p,t}$	unit capacity occupation of SKU p at factory f in week t.
h_p	unit inventory cost of SKU p.
y_p	unit profit loss due to unmet demand for SKU p.
w_p	unit penalty coefficient for unmet target inventory of SKU p.
$\widetilde{I}_{p,t}$	estimated receipt quantity of in-transit POs for SKU p in week t.
SLA_p^S	supply weeks for SKU p.
SLA_p^T	transportation weeks for SKU p.
Q_p	minimum order quantity for SKU p.
$\bar{t}_p$	production stop time of SKU p that is going to be discontinued.
S_p	number of weeks to be considered for safety stock of SKU p.
$R_{p,t}$	target inventory level for SKU p in week t.
M	big M where $M = 10^8$.
Decision Variables	
$x_{f,p,t}$	order quantity of SKU p at factory f in week t.
$I_{p,t}$	remaining inventory of SKU p in week t, where $I_{p,0}$ refers to the initial inventory.
$u_{p,t}$	unmet demand quantity for SKU p in week t.
$s_{p,t}$	unmet target inventory quantity for SKU p in week t.
$z_{f,p,t}$	binary variable indicating whether SKU p is ordered at factory f in week t.
$e_{p,t}$	binary variable indicating whether demand of SKU p in week t is fully satisfied.

Table 2: Parameters and Variables of the DA Model

Sets	
$\mathcal{T}$	Time period, indexed by t.
$\mathcal{O}$	Demand order set, indexed by o.
$\mathcal{P}$	Product set, indexed by p.
$\mathcal{J}$	PO (purchased order) set, indexed by j.
$\mathcal{R}$	Region set, indexed by r.
$\mathcal{O}_r$	Demand Orders under region r.
Parameters	
$\mathcal{P}_o$	Products under demand order o
$\mathcal{P}_j$	Products under PO j.
τ_j	Arrival time of PO j.
$C_{r,p,t}$	Capacity of product p under region r at time t.
Q_j	Quantity of PO j.
d_o	Demand quantity for demand order o.
RPD_o	RPD (requested packing list date) of order o.
EPD_o	EPD (estimated packing list date) of order o.
u_o	Penalized coefficient of RPD delay of order o.
v_o	Penalized coefficient of EPD delay of order o.
Decision Variables	
$x_{o,t}$	Quantity distributed from inventory to demand order o at time t.
$y_{j,o,t}$	Quantity distributed from PO j to demand order o at time t.
$z_{o,t}$	Quantity of unmet demand of demand order o at time t.
$Delay_o^{RPD}$	RPD delay penalty of order o.
$Delay_o^{EPD}$	EPD delay penalty of order o.