
安克创新：主生产计划与配送分配项目

张浩然

数据科学学院

香港中文大学（深圳）

122090735@link.cuhk.edu.cn

曾洋

金融工程项目

香港中文大学（深圳）

122090714@link.cuhk.edu.cn

1 引言

供应链运营的优化对于像安克创新这样的全球企业至关重要，这些企业管理着复杂的大规模生产和分销网络。本报告提出了一个关于主生产计划（MPS）和配送分配（DA）模型的全面研究，旨在提高安克供应链的效率。为了应对安克运营的多方面挑战，我们将供应链分解为两个相互关联的优化问题。MPS 模型专注于上游生产计划，确定最佳订单数量、工厂分配等。相反，DA 模型解决下游物流问题，优化现有库存和即将到货的采购订单（POs）以满足区域需求订单。通过整合这些模型，我们的方法确保了长期生产战略与实时分销策略之间的无缝协调，专为安克的大规模运营量身定制。利用严格的混合整数规划（MIP）和线性规划（LP）公式、精心定义的模型规模缩减技术以及计算优化，该框架提供可操作的解决方案，以提高成本效率、服务可靠性和对动态市场需求的适应性。此外，对于每个模型，我们建立了从数据预处理、模型构建到模型输出的完整工作流程。我们的代码可通过链接访问 <https://github.com/ZH-Haoran/Anker-MPS-Project>。

2 主生产计划

2.1 问题描述

主生产计划（MPS）问题是安克供应链中一个关键的上游决策过程。在接收到需求预测后，MPS 模型的任务是确定最佳订购决策，以有效平衡库存持有成本与满足客户需求的迫切需要。安克面临的一个重大挑战，特别是在生产能力受限的情况下，是公司难以定量评估是提前下订单（从而产生额外的库存成本）还是放弃满足某些需求（可能导致收入损失）。

该模型必须具有足够的灵活性，以适应安克独特的业务场景，例如新产品发布、产品停产以及为重大促销活动提前备货。由于多个工厂争夺生产能力，该问题的复杂性进一步放大。这是因为单一库存单位（SKU）可能在多个工厂生产，而每个工厂能够生产多个 SKU，形成了多方面的决策环境。

在操作上，MPS 模型需要生成每周决策，指定哪个工厂应生产每个 SKU 以及相应的订单数量。该模型将在滚动时间框架内实施，每周执行一次，或在需求预测发生重大变化时执行。当前周的决策将直接指导实际订单数量，而未来几周的预测将为库存计划和早期预警系统提供关键基础，以缓解潜在的供应链中断。

2.2 方法

为了解决 Sec. 2.1 中描述的问题，我们开发了一个混合整数规划 (MIP) 操作模型。我们在表 1 中列出了 MPS 模型的所有符号。在本节中，我们将首先介绍模型，包括目标函数和每个约束的含义 (Sec. 2.2.1)，然后展示它如何兼容前面提到的特殊场景 (Sec. 2.2.2)。

2.2.1 MPS 模型

该操作模型旨在平衡库存成本和未满足需求可能导致的利润损失。为了优化这些目标，我们提出 MPS 模型的目标函数为：

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

其中，第一项和第二项分别表示总库存成本和未满足需求的总利润损失，最后一项是未满足目标库存的惩罚，激励模型满足一些个性化需求，如安全库存水平。

工厂容量约束

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

其中，右侧表示工厂 f 在周 t 的剩余产能，左侧表示本次计划中下单将占用的相应产能。剩余产能通过从总标准化容量中减去已确定采购订单 (POs，在本轮模型运行之前已确定) 占用的产能计算。此外，每个订单的产能占用等于订单中包含的 SKU 数量乘以其单位产能占用。值得注意的是，由于我们无法准确知道工厂何时安排生产，模型根据“订单开始运输前一周为产能占用周”的原则估算工厂产能占用的时间。

最小订单量 (MOQ) 约束

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

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

在这些约束中，我们要求订单数量 $x_{f,p,t}$ 在模型决定下单时 (即 $z_{f,p,t} = 1$) 必须大于或等于 MOQ；否则，它们变为 $0 \leq x_{f,p,t} \leq M$ ，意味着 MOQ 约束完全放宽。

目标库存水平约束

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

其中，左侧表示可用库存，包括三个部分：(i) 前一周的剩余库存；(ii) 本周将到达的 PO 单；(iii) 本轮计划下单并在本周到达的订单。此外， $s_{p,t}$ 是一个松弛变量，表示未满足的目标库存量，以防止所有工厂容量不足时出现不可行解。右侧是根据需求确定的目标库存水平，通常表示安全库存水平，计算公式为：

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

此外，它还可以包含提前备货的需求。我们将在 Sec. 2.2.2 中进一步解释。

库存平衡约束

$$\begin{aligned} I_{p,t} &= I_{p,t-1} + \tilde{I}_{p,t} + \sum_{f \in \mathcal{F}_p} \sum_{\substack{\tau \in \mathcal{T}; \\ \tau + SLA_p^S = t}} x_{f,p,\tau} - D_{p,t} + u_{p,t}, & \forall p \in \mathcal{P}, t \in \mathcal{T} \\ u_{p,t} &\leq M \cdot (1 - e_{p,t}), & \forall p \in \mathcal{P}, t \in \mathcal{T} \\ I_{p,t} &\leq M \cdot e_{p,t}, & \forall p \in \mathcal{P}, t \in \mathcal{T} \end{aligned}$$

第一个等式是库存平衡约束的主体部分。即，如果可用库存充足，一部分将用于满足需求，另一部分将成为剩余库存。否则，松弛变量未满足需求量 $u_{p,t}$ 将被激活，可用库存将等于需求和未满足需求量的总和。另两个不等式旨在强制可用库存完全用于满足需求，而不是保留以满足下一周的目标库存水平。具体来说，当 SKU p 在周 t 的需求完全满足 ($e_{p,t} = 1$) 时，我们要求没有未满足需求量。相反，当模型决定放弃需求 ($e_{p,t} = 0$) 时，剩余库存也必须为零。总之，它们共同确保模型无法通过调整“未满足需求量 $u_{p,t}$ ”变量来控制用于满足当前周需求的库存量。这是因为我们只希望模型优化订购决策，而不是平衡需求满足率和目标库存水平。

停产约束

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

其中，我们要求即将停产的 SKU 在生产停止周后的订单数量为零。由于此约束针对停产场景，我们将在 Sec. 2.2.2 中提供更深入的解释。

2.2.2 特殊场景

在本节中，我们将详细说明我们的模型如何兼容 Sec. 2.1 中列出的三种特定业务场景。

产品上市 当一个产品即将上市时，我们可能已经有了上市后数周的需求预测。因此，MPS 模型负责通过提前订购来准备产品。这种场景等同于需求在一开始为零，然后在发布后变为非零，因此模型可以自动优化备货过程，无需任何修改。

产品下市 如果一个产品将要下市，公司将在特定周之前终止其生产。此时，剩余库存和采购订单将用于满足市场需求，直到产品正式下市。因此，在 Sec. 2.2.1 的停产约束中，我们强制要求生产停止周后的所有订单数量为零。此时，由于模型不想承担需求损失的成本，它会提前做出生产调度决策，这与我们希望实现的效果完美契合。

大促的提前备货 在重大促销活动（如双十一）期间，工厂产能可能会出现显著瓶颈，同时会有提前补货的业务需求。此时，模型需要考虑提前下单以进行备货。我们注意到，重大促销前的预备货需求和安全库存需求具有相同的性质，即我们希望在特定周的可用库存达到一定水平。然而，在实际运行库存平衡方程时，我们仅减去实际预测需求。因此，我们不能直接将安全库存和预备货量添加到预测需求中。相反，我们需要引入一个变量 $R_{p,t}$ ，表示“目标库存水平”，以结合安全库存和潜在的预备货量。在这种场景中，

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

其中 $\tilde{D}_{p,t}$ 表示周 t 的预备货量。此外，通过调整 $R_{p,t}$ 的计算逻辑，我们可以有效地将模型扩展到更多类似的业务场景。

2.2.3 模型规模缩减

尽管模型已经建立，但由于安克实际业务数据的庞大规模，我们必须采用一些技术来缩减模型规模并加速求解。

规模缩减技巧 我们主要采用了三种技巧来显著减少 MPS 模型的规模：

1. 将供应中心 Y 和 Z 的数据分开并独立求解，因为在安克的业务中，两个供应中心的数据可以完全解耦，互不影响。
2. 仅声明必要的变量和约束。如 Sec. 2.2.1 中所示，我们在建立模型时谨慎选择受相应约束限制的集合，并仅声明模型实际使用的决策变量。
3. 将同一工厂内所有生产线的容量预聚合为工厂容量，将工厂维度从 $|\mathcal{F}| \times |\mathcal{L}|$ 减少到 $|\mathcal{F}|$ ，其中 $|\mathcal{L}|$ 表示所有工厂的生产线总数。

值得注意的是，这些技术不会损害模型的求解精度，因为这里没有使用任何近似方法。

模型规模比较 最初，我们将三元组（供应中心，工厂，生产线）视为一个工厂对象，将二元组（供应中心，SKU ID）视为一个 SKU 对象。在数据清理后，我们获得了 3408 个 SKU 对象和 157 个工厂对象。如果不使用上述技术，模型的非零元素数量将达到千万量级，包含 17 万个整数变量，这是一个不可接受的规模。本地内存甚至无法支持完成求解。然而，在模型规模缩减后，供应中心 Y 和 Z 的模型参数数量均减少到十万的量级。在优化算法平台（COAP）上使用 Cardinal Optimizer (COPT) 求解时，两个模型在 300 秒内分别实现了 0.329% 和 1.737% 的相对容差，目标函数值分别为 7.17×10^7 和 1.55×10^8 。在现实世界的大规模场景中，这样的误差已经可以接受。

求解加速 由于整数变量在求解过程中会导致计算复杂度的指数增长，整数变量的数量是优化的主要瓶颈。考虑到订单数量的轻微误差对目标函数的影响非常有限，我们将订单数量变量从整数类型放宽为连续类型，并在求解后四舍五入为整数作为最终订单

决策。通过这种方式，我们可以在 200 秒内完全求解这两个模型（分别针对供应中心 Y 和 Z），即相对容差达到 0.01%。我们将在 Sec. 2.3 中进一步证明这一策略的合理性。

2.3 数值结果

为了有效检查模型的准确性和性能，我们对一些具有代表性的 SKU 进行了可视化选择，其库存曲线如下图所示，其中蓝色线条表示每周的可用库存，橙色线条表示目标库存水平，绿色线条表示需求，黄色线条表示本轮采购的到达订单数量。值得注意的是，红色虚线表示供应周，在此之后本轮模型优化的结果生效，因为第一天下单的产品将在本周到达。

前三个图显示了三个具有代表性的 SKU，所有这些 SKU 均使用连续型决策变量求解。图 1 显示了一个完美运行的 SKU，在整个计划周内满足目标库存水平。图 2 展示了一个典型的“供应过剩”案例，实际上，模型没有下任何订单。然而，图 3 的订单是一个生产能力相对有限的 SKU。因此，在权衡库存成本与未满足目标库存的惩罚后，没有做出订购决策。它仅满足了预期需求，但未完全满足目标库存。同时，图 4 显示了在使用整数变量设置时（此时模型未完全求解）相同 SKU3356 的库存曲线。它显示模型选择在第 16 周下单 300 单位的 SKU（其最小订单量），当时有可用产能。然而，可以轻易验证这是次优的。第 16 周后，该产品的需求非常低，意味着这 300 单位中的大部分将产生额外的库存成本，因此完全放弃这部分需求会更好。图 3 和图 4 的比较也证明了我们决策变量放宽为连续形式以获得最优解的策略的合理性。

总结来说，我们通过数学建模、规模缩减、求解加速和可视化，获得了 MPS 问题的最优解，并进行了相关验证，同时使得模型能够兼容安克的各种实际业务需求。

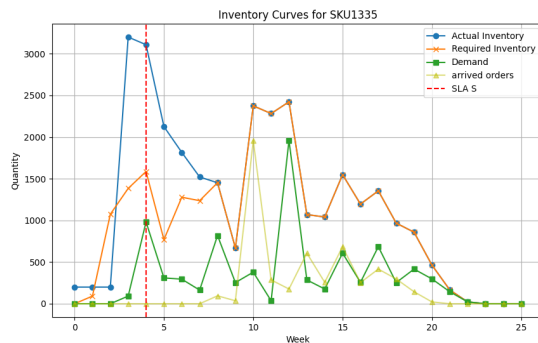


图 1: SKU1335 库存曲线

3 配送分配

3.1 问题描述

配送分配（DA）模型是安克供应链的下游任务。有效的下游物流运营对于安克在保持成本效率和服务质量的同时满足分销商需求至关重要。在执行主生产计划（MPS）后，安克面临第二个层面的运营挑战：如何将即将到货的采购订单（POs）和现有产品资源

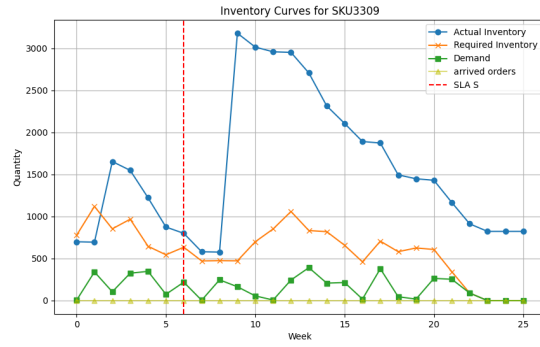


图 2: SKU3309 库存曲线

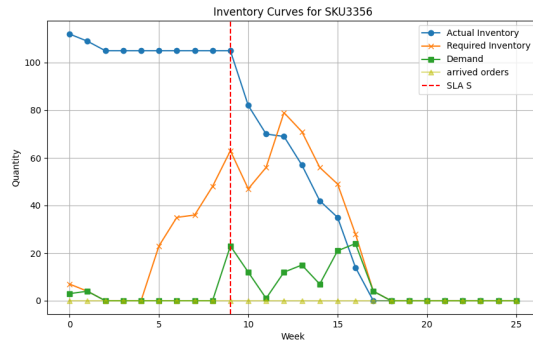


图 3: SKU3356 库存曲线（连续变量）

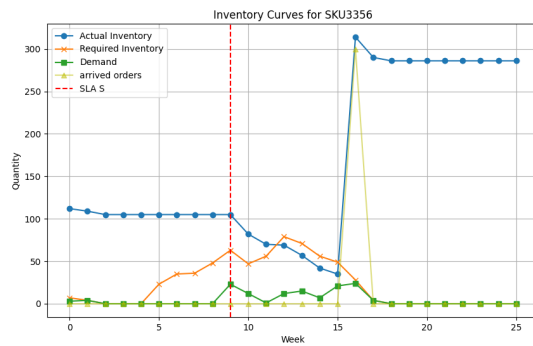


图 4: SKU3356 库存曲线（整数变量）

(库存) 分配给地理分布的分销商网络，每个分销商发出具有特定时间和数量要求的需求订单。这导致了我们的配送分配 (DA) 问题。DA 模型旨在为每个需求订单确定从现有库存或计划采购订单 (POs) 的最佳采购决策、分配数量和精确的交付时间，同时最小化与延迟履行相关的惩罚。与主要关注长期规划和上游生产决策的 MPS 模型不同，DA 问题在时间和空间分辨率上更精细，涉及在严格物流约束下的实时交付规划。

配送分配 (DA) 问题具有高度的运营复杂性。主要考虑因素之一是区域容量，这为每个时间段内向每个区域交付的 SKU 总量设定了上限。此约束模拟了实际限制，例如劳动力可用性、区域仓库的码头吞吐量和区域分销商级别。当需求在时间上集中或跨区域分布不均时，在不违反这些限制的情况下满足所有订单成为一个非平凡的分配挑战。另一个基本困难来自库存容量约束和采购订单容量约束，这确保了从现有库存或即将到货的采购订单的分配不超过可用数量。鉴于库存和采购订单是有限的，并且通常在多个订单和区域之间共享，模型必须谨慎决定如何在竞争需求之间分配资源，可能会优先考虑期限更严格或延迟惩罚更高的订单。此外，采购订单到达时间约束禁止在计划到达之前使用任何采购订单。这种时间依赖性要求模型不仅要考虑需求的紧急性，还要考虑供应流的同步。可以说，最复杂的方面体现在需求平衡约束中，该约束要求每个订单的分配总量（无论是来自库存还是未来采购订单）必须满足全部需求，否则通过松弛变量记录为短缺。松弛变量的存在通过 EPD 和 RPD 惩罚函数与延迟惩罚相关联，这些函数跟踪未满足数量相对于客户请求和计划交付日期 (EPD 和 RPD) 的延迟时间。这些惩罚是时间敏感的，并按 SKU 特定的系数加权，使模型能够反映差异化的服务水平和业务优先级。然而，这一特性也为目标函数引入了显著的异质性，使得竞争订单之间的权衡在实践中高度非线性。特别是，高优先级 SKU 的延迟可能带来不成比例的巨大惩罚，迫使模型在全局效率和本地服务水平承诺之间进行权衡。

3.2 方法

在接下来的部分中，我们将展示 Sec. 3.1 中描述的问题的公式化。DA 模型的所有参数符号列于表 2 中。我们开发了一个线性规划 (LP) 模型，详细的目标函数和约束如下所述：

3.2.1 DA 模型

目标函数

该模型旨在合理分配现有库存和计划采购订单以满足区域需求订单。提出以下目标函数：

$$\text{Minimize} \quad \sum_{r \in \mathcal{R}} \sum_{o \in \mathcal{O}_r} (\text{Delay}_o^{\text{RPD}} \cdot u_o + \text{Delay}_o^{\text{EPD}} \cdot v_o)$$

目标是最小化所有需求订单的交付惩罚总和。惩罚由相对于 RPD（请求装箱日期）和 EPD（预计装箱日期）的交付延迟组成，延迟成本由订单特定的优先级系数加权。

区域容量约束

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

其中，左侧表示从库存和采购订单分配到区域 r 在时间 t 的总分配量。此约束确保产品 p 在时间 t 交付到区域 r 的所有订单的总量不超过区域容量 $C_{r,p,t}$ 。交付可以来自库

存 $(x_{o,t})$ 或已到达时间 t 的合格采购订单 $(y_{j,o,t})$ 。这反映了最后一英里交付或区域仓储物流。

库存容量约束

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

其中，左侧表示来自库存的产品 p 的总量。此约束限制了从库存分配给需求产品 p 的订单的产品 p 数量。它确保从库存的分配不超过可用库存 I_p 。这反映了仓库级别的物理库存约束。

采购订单容量约束

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

其中，左侧表示来自采购订单 j 的分配总量。此约束限制了从采购订单 j 分配给所有相关需求订单在时间 t 的总量不超过采购订单的数量 Q_j 。它强制执行对即将到货的正确容量跟踪。

采购订单到达时间约束

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

其中，左侧表示在采购订单 j 到达时间 τ_j 之前分配的总量。此约束确保采购订单 j 在到达时间之前不能用于满足任何需求订单 o 。这模拟了货物在未收到之前无法分配的基本时间逻辑。

需求平衡约束

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

其中，左侧包括三个部分：(i) 累计库存分配，(ii) 累计采购订单分配，(iii) 未满足需求。此约束确保对于每个需求订单 o ，截至时间 t 的库存和采购订单的累计分配加上未满足需求 $z_{o,t}$ 必须等于总需求量 d_o 。松弛变量 $z_{o,t}$ 捕获履行中的短缺，随后在我们的目标函数中导致惩罚。

RPD 和 EPD 惩罚函数

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

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

其中，右侧表示惩罚函数的计算。这两个方程通过未满足需求 \times 延迟天数分别计算 EPD 和 RPD 延迟履行。超过截止日期的时间越长，未满足需求量越大，惩罚越大。此惩罚函数强制模型在惩罚方面实现更早和更小的交付。此外，惩罚函数可以根据不同的业务场景进行更改，例如，我们可以使用二次函数来区分需求订单及其 EPD、RPD 的优先级。

3.3 模型规模缩减

原始规模

在清理安克提供的数据库后，问题包括 **2017 个需求订单**，**502 个采购订单**，以及 **44 周** 的时间范围。因此，完整的配送分配（DA）模型将包含数百万个变量。具体来说，捕获从采购订单 j 到订单 o 在时间 t 的分配的三维决策变量 $y_{j,o,t}$ 将具有超过 4400 万 ($2017 \times 502 \times 44$) 的规模，这导致了几乎不可解的问题规模。因此，完整变量空间的实现将导致计算不可处理和模型初始构建期间的内存溢出（图 5）。

```

/Users/yang/anaconda3/envs/DDA4080/bin/python /Users/yang/PycharmProjects/DDA4080/Delevery_COPT_test.py
Cardinal Optimizer v7.2.4. Build date Dec 6 2024
Copyright Cardinal Operations 2024. All Rights Reserved

Process finished with exit code 137 (interrupted by signal 9:SIGKILL)

```

图 5: 原始问题构建失败

规模缩减技术

为了解决规模问题，我们采用了多步骤模型缩减策略，通过重构时间集合、分解需求订单、划分为子问题。

1. 减少时间集合:

我们提出了以下重构时间集合:

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

通过仅保留与 RPD、EPD 和采购订单到达日期相对应的时间点，可以安全地减少时间集合。此操作对问题的优化无害，因为一旦提出需求订单，如果当前数量充足，交易将在 RPD 或 EPD 完成，否则，交易将在新的采购订单到达时发生。

2. 分解多 SKU 需求订单：

我们将多 SKU 需求订单分解为包含单一 SKU 的子订单，通过添加履行订单行：

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

这样，每个订单代码加上履行订单行形成一个唯一代码，确定仅包含一个 SKU 的需求订单。这种转换避免了模型公式中不必要的跨产品扩展，并允许模型独立关注 SKU 级别的分配决策。

3. 划分最终问题：

最后，我们将整个问题按 SKU 划分为独立的子问题，支持并行计算并隔离相互依赖性。此操作显著减少了每个子问题的变量计数和内存使用量。

```
Using Cardinal Optimizer v7.0.6 on macOS (aarch64)
Hardware has 8 cores and 8 threads. Using instruction set ARMV8 (30)
Minimizing an LP problem

The original problem has:
    5665 rows, 443160 columns and 5411551 non-zero elements
The folded problem has:
    3979 rows, 117955 columns and 3380675 non-zero elements
The presolved problem has:
    3698 rows, 113165 columns and 3243733 non-zero elements

Starting the simplex solver using up to 8 threads
```

图 6: 缩减后的问题规模

通过规模缩减操作，模型总规模从大约 4000 万个变量减少到大约 500 万个（图 6），使问题能够在 30 秒内高效求解。

3.4 数值结果

配送分配（DA）模型在处理大规模、时间变化的任务方面表现出色。使用 COPT 求解器，模型实现了解时间不到 **30 秒**，按时交付率为 **94.45%**，2017 个需求订单中只有 112 个出现延迟。

我们将通过 **SKU220** 来说明我们的结果：

如图 7 所示，我们的模型以零 EPD 惩罚完成分配。相比之下，图 8 模拟了一种按需求订单到达顺序分配产品的简单方式。在最后两个需求订单中产生了 EPD 惩罚，这表明我们的模型在优化安克的分配策略、确保现实世界供应链操作中更高的成本效率方面是有效的。

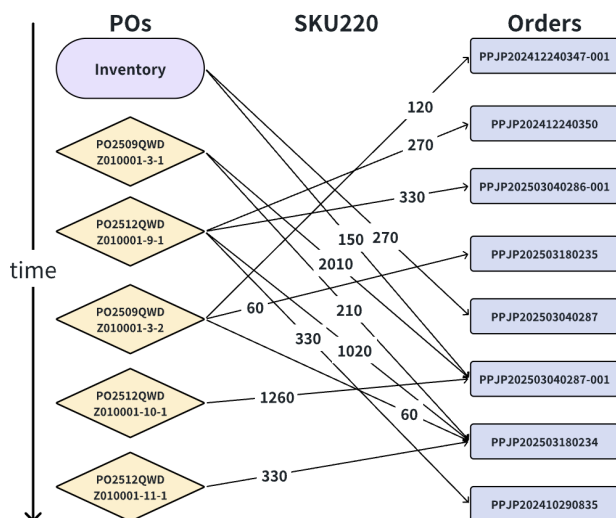


图 7: DA 模型结果 (SKU220)

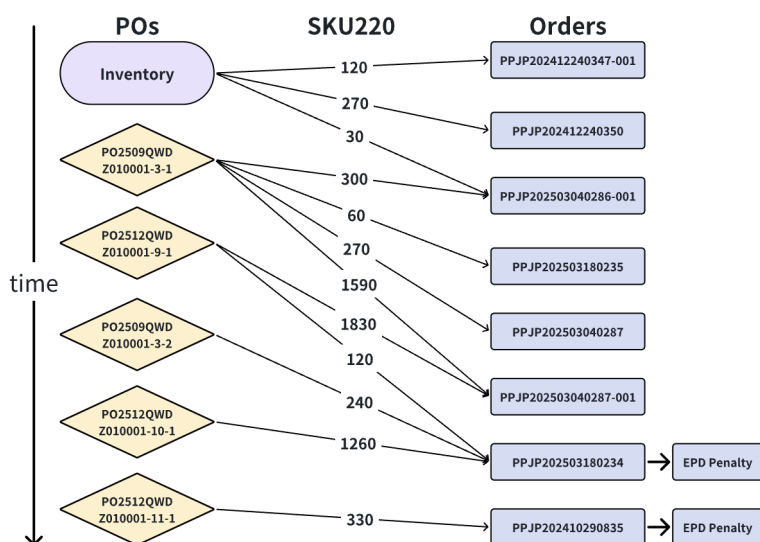


图 8: 简单方法结果 (SKU220)

4 结论

本项目提出了一个全面的二阶段优化框架，旨在提高安克创新的供应链效率和交付响应能力。通过整合主生产计划（MPS）模型和配送分配（DA）模型，解决生产和物流规划中固有的全谱挑战，提供了一个统一的解决方案。

MPS 模型确定 26 周的生产数量，旨在最小化总成本，包括库存持有成本、失销成本和未满足目标库存的惩罚。它考虑了现实的业务场景，如产品发布、停产和提前备货。我们的模型能够在 200 秒内有效达到最优解，并通过可视化验证了解决方案。

DA 模型为超过 2000 个需求订单、500 多个采购订单分配资源，时间跨度约为 1 年，试图最小化按 SKU 级优先级系数加权的延迟惩罚。该模型考虑了复杂的约束，包括区域容量、采购订单可用性、库存水平和交付时间（RPD 和 EPD）。模型能够在 30 秒内找到最优解，按时交付率为 94.45%。

这两个模型共同形成了一个完整的、数据驱动的决策框架，有效连接了生产和交付计划。我们的结果表明，MPS 模型在不确定性下确保高效的生产计划，而 DA 模型能够提供比安克当前使用更好的分配策略。总体而言，我们的工作为大规模供应链协调提供了可扩展、灵活且可实施的优化解决方案。

5 贡献和角色

两个模型的初始建模由张浩然和曾洋共同完成。此外，张浩然和曾洋分别主要负责 MPS 模型和 DA 模型的后期开发，包括代码实现、模型优化和模型规模缩减策略的制定。

表 1: MPS 模型的参数和变量

集合	
\mathcal{T}	总计划周数, 按 t 索引。
\mathcal{P}	SKU 集合, 按 p 索引。
\mathcal{F}	工厂集合, 按 f 索引。
\mathcal{P}_f	工厂 f 可生产的 SKU 集合, 按 p 索引。
\mathcal{P}_Q	具有非零最小订单量 (MOQ) 的 SKU 集合, 按 p 索引。
\mathcal{P}_{off}	即将下市的 SKU 集合, 按 p 索引。
\mathcal{F}_p	能够生产 SKU p 的工厂集合, 按 f 索引。
\mathcal{L}_f	工厂 f 中的生产线集合, 按 l 索引。
参数	
$D_{p,t}$	SKU p 在周 t 的需求。
$C_{f,l,p,t}$	工厂 f 中生产线 l 在周 t 对 SKU p 的产能。
$C_{f,p,t}$	工厂 f 在周 t 对 SKU p 的产能, 通过以下公式计算 $C_{f,p,t} = \sum_{l \in \mathcal{L}_f} C_{f,l,p,t}。$
$C_{f,t}$	工厂 f 在周 t 的标准产能, 通过以下公式计算 $C_{f,t} = \max_{p \in \mathcal{P}_f} C_{f,p,t}。$
$o_{f,p,t}$	SKU p 在工厂 f 在周 t 的单位产能占用。
h_p	SKU p 的单位库存成本。
y_p	因未满足需求导致的 SKU p 的单位利润损失。
w_p	SKU p 未满足目标库存的单位惩罚系数。
$\tilde{I}_{p,t}$	SKU p 在周 t 的在途采购订单的预计接收量。
SLA_p^S	SKU p 的供应周数。
SLA_p^T	SKU p 的运输周数。
Q_p	SKU p 的最小订单量。
\bar{t}_p	即将下市的 SKU p 的停产时间。
S_p	SKU p 安全库存考虑的周数。
$R_{p,t}$	SKU p 在周 t 的目标库存水平。
M	大 M , 其中 $M = 10^8$ 。
决策变量	
$x_{f,p,t}$	SKU p 在工厂 f 在周 t 的订单数量。
$I_{p,t}$	SKU p 在周 t 的剩余库存, 其中 $I_{p,0}$ 指初始库存。
$u_{p,t}$	SKU p 在周 t 的未满足需求量。
$s_{p,t}$	SKU p 在周 t 的未满足目标库存量。
$z_{f,p,t}$	二元变量, 指示 SKU p 在工厂 f 在周 t 是否下单。
$e_{p,t}$	二元变量, 指示 SKU p 在周 t 的需求是否完全满足。

表 2: DA 模型的参数和变量

集合	
\mathcal{T}	时间段, 按 t 索引。
\mathcal{O}	需求订单集合, 按 o 索引。
\mathcal{P}	产品集合, 按 p 索引。
\mathcal{J}	采购订单 (PO) 集合, 按 j 索引。
\mathcal{R}	区域集合, 按 r 索引。
\mathcal{O}_r	区域 r 下的需求订单。
参数	
\mathcal{P}_o	需求订单 o 下的产品。
\mathcal{P}_j	采购订单 j 下的产品。
τ_j	采购订单 j 的到达时间。
$C_{r,p,t}$	产品 p 在区域 r 时间 t 的容量。
Q_j	采购订单 j 的数量。
d_o	需求订单 o 的需求数量。
RPD_o	订单 o 的请求装箱日期 (RPD)。
EPD_o	订单 o 的预计装箱日期 (EPD)。
u_o	订单 o 的 RPD 延迟惩罚系数。
v_o	订单 o 的 EPD 延迟惩罚系数。
决策变量	
$x_{o,t}$	从库存分配给需求订单 o 在时间 t 的数量。
$y_{j,o,t}$	从采购订单 j 分配给需求订单 o 在时间 t 的数量。
$z_{o,t}$	需求订单 o 在时间 t 的未满足需求量。
$Delay_o^{RPD}$	订单 o 的 RPD 延迟惩罚。
$Delay_o^{EPD}$	订单 o 的 EPD 延迟惩罚。