

# SKD 出货建议算法文档

## 1. 概述

本算法旨在解决 SKD（半散件）需求的库存分配问题。

在 CKD（全散件）基础上，进一步考虑库存仓库的优先级，严格遵循 BOM 结构的物料分配规则，并确保需求齐套，不允许超量出货。

每个 site 作为一个独立的出货单元，不同 site 之间的需求和库存不共享。通过合理分配库存，算法主要优化目标包括：

1. 最大化需求满足率，确保尽可能多的需求被满足
2. 优先满足较早周次的需求，确保早期需求不会被后期需求挤占库存
3. 优先使用入库较早的库存，避免库存老化

通过建立混合整数规划模型，在满足约束条件的前提下，优化出库计划。

## 2. 问题描述

### 库存类型

- 库存分为小箱、大箱和Open PO
  - 小箱不可拆分，需整箱出库
  - 大箱由多个小箱组成，同一site内可共享
  - Open PO为未装箱库存，按量出库，仅包含散料

### 库存分配规则

- 优先使用与需求 `kitting_location` 匹配的stock库存和open po，次之使用与需求 `ship_location` 匹配的 stock 库存
- 其中与需求 `kitting_location` 匹配的stock库存下严格按照 bom 结构优先级出库，即先出库半成品，再出库散料

3. 数学模型

3.1 符号说明

符号	描述
【需求属性】	
$i$	需求编号, $i = 1, 2, \dots, I$
$Q_i$	需求 $i$ 的需求量
$W_i$	需求 $i$ 的周次
$B_i$	需求 $i$ 的 BOM 结构, $B_i = \{b_{i,1}, b_{i,2}, \dots\}$ , $b_{i,j}$ 是一个物料节点
$sl_i$	需求 $i$ 的ship_location
$kl_i$	需求 $i$ 的kitting_location
【BOM属性】	
$b_{i,j}$	需求 $i$ 的第 $j$ 个物料节点, 引入一个虚拟成品节点 $b_{i,0}$
$pn_{i,j}$	物料编号
$parent\_pn_{i,j}$	父物料编号 (为空表示最上层物料)
$unit\_qty_{i,j}$	单位用量, 表示成品需要的当前物料数量
$alt\_pn_{i,j}$	主料 $\cup$ 替代物料列表
【库存属性】	
$s$	库存(小箱)编号, $s = 1, 2, \dots, S$
$p_s$	库存 $s$ 的物料号
$q_s$	库存 $s$ 的物料数量
$d_s$	库存 $s$ 的入库日期
$l_s$	库存 $s$ 的库位
$m_s$	库存 $s$ 的合箱号, $m_s = c$ 表示 $s$ 属于大箱 $c$ (为空表示未合箱)
【Open PO 属性】	

符号	描述
$k$	Open PO 编号, $k = 1, 2, \dots, K$
$pn_k$	Open PO $k$ 的物料编号
$q_k$	Open PO $k$ 的物料数量
$l_k$	Open PO $k$ 的kitting_location
<b>【决策变量】</b>	
$x_{i,s}$	$x_{i,s} \in \{0, 1\}$ , 是否使用小箱子 $s$ 满足需求 $i$ , $x_{i,s} = 1$ 表示使用, 否则为 0
$y_i$	$y_i \in \{0, 1\}$ , 需求 $i$ 是否被满足, $y_i = 1$ 表示完全满足, 否则为 0
$z_c$	$z_c \in \{0, 1\}$ , 是否在出货单元中使用大箱 $c$ , $z_c = 1$ 表示使用, 否则为 0
$m_{i,k}$	$m_{i,k} \in \mathbb{Z}$ , Open PO $k$ 用于满足需求 $i$ 分配的物料数量, $m_{i,k} = 0$ 表示未使用
<b>【辅助变量】</b>	
$Supply_{i,j}$	$Supply_{i,j} \in \mathbb{Z}$ , bom 节点 $b_{i,j}$ 的实际出货量
$o_{i,j}$	$o_{i,j} \in \mathbb{Z}$ , bom 节点 $b_{i,j}$ 的下层物料供应量
$l_{i,j}$	$l_{i,j} \in \{0, 1\}$ , bom 节点 $b_{i,j}$ 是否考虑下层散料供应

### 3.2 约束条件

#### 1. 需求满足约束:

- 需求必须齐套才允许出库, 即要么完全满足, 要么不满足

$$Supply_{i,0} = Q_i \cdot y_i \quad \forall i \in I$$

#### 2. Bom 结构约束:

所有 BOM 物料都齐套时, 需求才会被满足

- 对于有下层展开的物料节点
  - 对于半成品物料节点 (机头), 其供应量= 当前物料的 kitting\_location 供应量+该物料的下层供应量

$$Supply_{i,j} = \sum_{p_s \in alt\_pn_{i,j}} x_{i,s} \cdot q_s + o_{i,j} \quad \forall i, j$$

其中，下层供应量= 下层子物料的供应量/单位用量

$$o_{i,j} = \frac{Supply_{i,k}}{unit\_qty_{i,k}} \quad \forall i, j, parent\_pn_{i,k} = b_{i,j}$$

- 对于无下层展开的物料节点

- 对于没有下层展开的物料节点（包材/机头散料），其供应量= 当前物料 stock 供应量和open po供应量

$$Supply_{i,j} = \sum_{p_s \in alt\_pn_{i,j}} x_{i,s} \cdot q_s + \sum_{pn_k \in alt\_pn_{i,j}} m_{i,k} \quad \forall i, j$$

### 3. Bom 库存使用顺序约束：

半成品库存优先，库存不足时才向下层展开

- 对于有下层展开的物料节点，只有 kitting\_location 对应的 stock 仓库中存在半成品库存，所以对于有下层节点的物料, stock 必须优先分配再考虑下层供应。

$$o_{i,j} \leq M \cdot l_{i,j} \quad \forall i, j$$

$$Supply_{i,j} \cdot (1 - l_{i,j}) \leq \sum_{p_s \in alt\_pn_{i,j}} x_{i,s} \cdot q_s \quad \forall i, j$$

其中：

- $l_{i,j} = 1$ ，表示只有  $\sum_{p_s \in alt\_pn_{i,j}} x_{i,s} \cdot q_s$  库存不足时允许下层供应
- $l_{i,j} = 0$ ，表示仅使用半成品库存

### 4. 小箱出库约束：

- 每个小箱  $s$  只能分配给一个需求

$$\sum_{i=1}^I x_{i,s} \leq 1 \quad \forall s \in S$$

### 5. 大箱出库约束：

- 大箱出库时，所有小箱必须一起分配

$$x_{i,s} = z_{u,c} \quad \forall c, m_s = c, i \in I$$

### 6. OPEN PO 使用约束：

- 每个 Open PO  $k$  的分配量不能超过其库存量

$$\sum_{i=1}^I m_{i,k} \leq q_k \quad \forall k \in K$$

### 3.3 目标函数

采用加权多目标优化，将各目标函数通过加权求和转化为单个优化目标求解

$$\min \alpha_1 f_1 + \alpha_2 f_2 + \alpha_3 f_3 + f_{priority}$$

#### 1. 最大化需求的满足数量：

优先满足更多的需求,最小化未满足的需求量

$$f_1 = I - \sum_{i \in I} y_i$$

#### 2. 最小化需求被满足的周次：

优先满足较早周次的需求

$$f_2 = \sum_{i \in I} y_i \cdot W_i$$

#### 3. 最小化出库物料的库存日期：

优先使用较早的库存

$$f_3 = \sum_{s \in S} \sum_{i \in I} x_{i,s} \cdot d_s$$

#### 4. 最小化仓库出库优先级惩罚：

对于每个物料，如果 `kitting_location` 不存在可用库存，才考虑 `ship_location` 的库存。添加惩罚项，如果违反顺序，目标函数值会大幅增大。

计算出库优先级惩罚项：

$$f_{priority} = \sum_{i \in I} \sum_{j \in B_i} \left( \sum_{s \in S} x_{i,s} \cdot loc_{priority}(l_s, sl_i, kl_i) \right) + \sum_{k \in K} \frac{m_{i,k}}{q_k} \cdot loc_{priority}(l_k, sl_i, kl_i)$$

其中，

$loc_{priority}(l_s, sl_i, kl_i)$  是库位匹配优先级，强制 `kitting_location` 先于 `ship_location`

$$loc_{priority}(l_s, sl_i, kl_i) = \begin{cases} 10000 & \text{如果 } l_s = sl_i \\ 100 & \text{如果 } l_s = kl_i \text{ 且库存来源为 Open PO} \\ 1 & \text{如果 } l_s = kl_i \text{ 且库存来源为 stock} \end{cases}$$

权重设置

权重系数的设置是多目标优化中的关键环节，需要根据数据规模动态调整

1. 量纲对齐

实时计算当前问题的目标范围,动态缩放, 保证目标函数的量纲一致性,获取对齐后的目标函数

目标函数	原始范围	归一化范围
$f_1$ （未满足需求）	$0\sim I$	$0\sim 1$
$f_2$ （需求周次）	$0\sim max(W_i)$	$0\sim 1$
$f_3$ （库存日期）	$0\sim max(d_s)$	$0\sim 1$

2. 业务优先级权重

目标函数	业务优先级	归一化后权重
$f_1$	高	100
$f_2$	中	10
$f_3$	低	1

3. 验证和调整

通过测试数据模拟不同的权重组合，观察优化结果，确定合理的权重值

4. 总结

通过为库存库位和 BOM 结构中的物料节点分配合理的优先级，确保严格的库存使用顺序控制。  
通过建立混合整数规划模型，算法能够在满足约束条件的前提下，最大化需求满足率，并优先使用早期库存和较早周次的需求。

注：

- 1. 需求齐套的一个数量 等于 一个机头（或机头展开一层的物料）+ 包材物料,只有kitting location中存在半成品。
- 2. 库存分配具体到bom层。