Gurobi 供应链物流案例（波次优化、订单配送优化）

刃之砺信息科技（上海）有限公司

说明

智能算法是实现科学决策的核心和基础。只有在应用中落地，智能算法才能发挥应有的价值和威力。但落地的过程并不平坦。为了更好地让智能算法在实践中成功应用，我们特举办案例讲座，介绍在一些重要应用中采用智能算法的方法、步骤和经验。本次讲座以供应链物流领域中常见的波次优化和订单配送优化为案例。敬请关注我们后续其他案例。

课程大纲

(1) 波次优化(BatchOpt)

介绍波次优化问题，建立数学模型以及给出一些模型改进方式，并调用Gurobi求解模型。

(2) 订单配送案例

描述问题，介绍预处理思路以及数学模型，并通过Gurobi获得结果。

波次优化(BatchOpt)

- 问题描述

电商等仓储物流中心每天需要配送几万甚至十几万包裹,每个包裹需要包含来自于不同仓库,不同库区,不同数量的物品。如果规划不合理,可能会导致跨库、跨区域数量过多,订单配货时间过长,物流核心设备无效冗余运作距离过长,库存周转效率降低等问题。

例如某电商有P个包裹(每个包裹含多件商品,可能分布在多个仓库(暂时不考虑不同库区))需将其分到B个波次中,要求:

（1）每个波次的商品件数在[G1,G2]件之间;

（2）每个波次的包裹数量在[P1,P2]之间;

（3）波次包含的不同的仓库数量最少。

波次优化(BatchOpt)

- 输入数据格式

pakage_no：包裹号

warehouse：仓库

goods_qty：在仓库中的商品件数

pakage_no	warehouse	goods_qty
302826075	2号仓-13	2
302826075	2号仓-23	1
302826075	5号仓-3	1
311001535	2号仓-24	1
311001535	佳明仓-3	1
311308096	2号仓-11	3
311308096	2号仓-13	1
311308096	5号仓-4	2
314553595	2号仓-12	1
314553595	2号仓-14	1
314553595	2号仓-22	3
314553595	4号仓-2	1
314553595	5号仓-2	2
302826079	2号仓-13	1
302826079	2号仓-23	2
302826079	5号仓-3	2

波次优化(BatchOpt)

- 问题模型(version 1)

参数：

set I	包裹集合
set J	波次集合
set K	仓库集合
B	波次总数
M	足够大的数
Q_i	包裹 i 的商品件数
c_{ik}	包裹 i 是否有仓库 k 的商品
G_1, G_2	单一波次商品件数限制
P_1, P_2	单一波次包裹数量限制

变量：

x_{ij}	*binary*，包裹 i 是否属于波次 j
y_{jk}	*binary*，波次 j 是否用到仓库 k
z_j	*binary*，波次 j 是否用到

$$\text{minimize} \sum_{j\in J}\sum_{k\in K} y_{jk}$$

$$\sum_{j\in J} z_j = B \quad (1)$$

$$\sum_{j\in J} x_{ij} = 1 \quad \forall i \in I \quad (2)$$

$$G_1 z_j \le \sum_{i\in I} Q_i x_{ij} \le G_2 z_j \quad \forall j \in J \quad (3)$$

$$P_1 z_j \le \sum_{i\in I} x_{ij} \le P_2 z_j \quad \forall j \in J \quad (4)$$

$$M y_{jk} \ge \sum_{i\in I} c_{ik} x_{ij} \quad \forall j \in J, k \in K \quad (5)$$

目标函数使得波次包含的不同的仓库数量总和最少；约束(1)保证分配的波次数量；约束(2)保证每个包裹都安排到波次中去；约束(3)保证单一波次商品件数在区间$[G_1, G_2]$中；约束(4)保证单一波次包裹数量在区间$[P_1, P_2]$中；约束(5)确认波次 j 是否用到仓库 k。

波次优化(BatchOpt)

- 问题模型(version 1)

若电商共有P个包裹需将其分到B个波次中,且假设涉及到的仓库数量为K,计算模型version1规模:

model	变量数	约束数
version 1	P*B+K*B+B	P+4B+B*K+1

本次测试案例包裹数量57780,波次数量为107,仓库数量为4。

model	变量数	约束数
version 1	6182995	58637

version 1模型建模思路容易想到,但模型规模会随着包裹和波次数量增加而快速增加,因此求解该模型耗时较多。而在电商的场景下,**波次优化往往需要快速响应**,因此该模型在实际场景下很难直接应用。

波次优化(BatchOpt)

- 问题模型(version 2)思路

目标是使得所有波次包含的不同的仓库数量总和最少,因此建模的时候是否可以**将重点只放在仓库上**?

表格给出的数据不考虑区域,共涉及到四个仓库{2号仓,4号仓,5号仓,佳明仓},现在做一个模式(pattern)统计,包裹涉及到的仓库记为1,否则记为0。

例如包裹314553595涉及2号仓,4号仓,5号仓,对应的模式为{1,1,1,0}。

pakage_no	warehouse	goods_qty
302826075	2号仓-13	2
302826075	2号仓-23	1
302826075	5号仓-3	1
311001535	2号仓-24	1
311001535	佳明仓-3	1
311308096	2号仓-11	3
311308096	2号仓-13	1
311308096	5号仓-4	2
314553595	2号仓-12	1
314553595	2号仓-14	1
314553595	2号仓-22	3
314553595	4号仓-2	1
314553595	5号仓-2	2
302826079	2号仓-13	1
302826079	2号仓-23	2
302826079	5号仓-3	2

波次优化(BatchOpt)

- 问题模型(version 2)思路

统计结果 {2号仓,4号仓,5号仓,佳明仓}:

pakage_no	Pattern	goods_qty
302826075	{1, 0, 1, 0}	4
311001535	{1, 0, 0, 1}	2
311308096	{1, 0, 1, 0}	6
314553595	{1, 1, 1, 0}	8
302826079	{1, 0, 1, 0}	5

发现包裹302826075, 311308096和302826079对应的模模式是一致的, 因此用涉及仓库模式取代具体包裹可能会极大降低问题规模(只要考虑波次用了该模式多少次而不需要区分具体包裹)。

pakage_no	warehouse	goods_qty
302826075	2号仓-13	2
302826075	2号仓-23	1
302826075	5号仓-3	1
311001535	2号仓-24	1
311001535	佳明仓-3	1
311308096	2号仓-11	3
311308096	2号仓-13	1
311308096	5号仓-4	2
314553595	2号仓-12	1
314553595	2号仓-14	1
314553595	2号仓-22	3
314553595	4号仓-2	1
314553595	5号仓-2	2
302826079	2号仓-13	1
302826079	2号仓-23	2
302826079	5号仓-3	2

波次优化(BatchOpt)

- 问题模型(version 2)思路

问题要求单一波次的商品件数在[G1,G2]件之间,直接用{1,0,1,0}取代包裹302826075,311308096和302826079时,发现不好区分出包裹商品的件数,因此考虑在模式里增加商品件数这一维度:

Pattern：{包裹商品的件数,2号仓,4号仓,5号仓,佳明仓}

新模式的统计结果为:

pakage_no	Pattern	goods_qty
302826075	{4, 1, 0, 1, 0}	4
311001535	{2, 1, 0, 0, 1}	2
311308096	{6, 1, 0, 1, 0}	6
314553595	{8, 1, 1, 1, 0}	8
302826079	{5, 1, 0, 1, 0}	5

虽然模式的数量相对于原模式有所增加,但可以区分出包裹商品的件数,方便建立按模式的新模型。

波次优化(BatchOpt)

- 问题模型(version 2)

参数：$set\ P$ 模式集合
$set\ J$ 波次集合
$set\ K$ 仓库集合
B 波次总数
M 足够大的数
q_p 模式p的包裹的数量
Q_p 模式p的商品件数
c_{pk} 模式p是否涉及仓库 k
G_1, G_2 单一波次商品件数限制
P_1, P_2 单一波次包裹数量限制

变量：x_{pj} $integer$，波次j 使用模式p的次数
y_{jk} $binary$，波次j 是否用到仓库 k
z_j $binary$，波次j 是否用到

$$minimize \sum_{j\in J}\sum_{k\in K} y_{jk}$$

$$\sum_{j\in J} z_j = B \qquad (1)$$

$$\sum_{j\in J} x_{pj} = q_p \quad \forall\, p\in P \qquad (2)$$

$$G_1 z_j \le \sum_{p\in P} Q_p x_{pj} \le G_2 z_j \quad \forall j\in J \qquad (3)$$

$$P_1 z_j \le \sum_{p\in P} x_{pj} \le P_2 z_j \quad \forall\, j\in J \qquad (4)$$

$$M y_{jk} \ge \sum_{p\in P} c_{pk} x_{pj} \quad \forall\, j\in J, k\in K \qquad (5)$$

目标函数使得波次包含的不同的仓库数量总和最少；约束(1)保证分配的波次数量；约束(2)保证模式使用次数和其包裹数量匹配；约束(3)保证单一波次商品件数在区间$[G_1, G_2]$中；约束(4)保证单一波次包裹数量在区间$[P_1, P_2]$中；约束(5)确认波次 j 是否用到仓库 k。

波次优化(BatchOpt)

- 问题模型(version 2)

本次测试案例包裹数量57780,波次数量为107,仓库数量为4。按模式定义计算出模式数量为106,模型version2规模：

model	**变量数**	**约束数**
version 1	6182995	58637
version 2	11877	963

与version 1模型相比,version 2模型不区分具体的包裹而通过模式抓住包裹的特性,从而极大的降低了问题的规模,减少了模型求解的时间。

波次优化(BatchOpt)

- 测试案例数值计算

测试案例包裹数量为57780,波次数量为107,仓库数量为4,[G1,G2]=[1800,3000],[P1,P2]=[500,550],详细测试结果:

波次数量	每个波次包裹数量	每个波次包裹件数	仓库数量	运行时间(s)	GAP
107	[500, 550]	[1800, 3000]	251	114.196	8.7649%
[100, 120]	[500, 550]	[1800, 3000]	250	114.516	8.4000%

波次优化(BatchOpt)

- 问题进一步讨论

电商等仓储物流中心每天需要配送几万甚至十几万包裹,每个包裹需要包含来自于不同仓库,不同库区,不同数量的物品。如果规划不合理,可能会导致跨库、跨区域数量过多,订单配货时间过长,物流核心设备无效冗余运作距离过长,库存周转效率降低等问题。

例如某电商有P个包裹(每个包裹含多件商品,可能分布在多个仓库的不同库区需将其分到B个波次中,要求:

（1）每个波次的商品件数在[G1,G2]件之间;

（2）每个波次的包裹数量在[P1,P2]之间;

（3）波次包含的不同的库区数量最少。

pakage_no	warehouse	goods_qty
302826075	2号仓-13	2
302826075	2号仓-23	1
302826075	5号仓-3	1
311001535	2号仓-24	1
311001535	佳明仓-3	1

波次优化(BatchOpt)

- 问题进一步讨论

如果按照之前思路,用库区构建模式(Pattern),只需在模型(version 2)基础上修改部分参数及变量定义,得到按库区构建模式的问题模型(version 3):

参数： $set\ P$ 模式集合(库区)

$set\ K$ 库区集合

c_{pk} 模式p是否有库区 k 的商品

变量： y_{jk} $binary$,波次j 是否用到库区 k

$$minimize \sum_{j\in J}\sum_{k\in K} y_{jk}$$

$$\sum_{j\in J} z_j = B \quad (1)$$

$$\sum_{j\in J} x_{pj} = q_p \quad \forall\ p \in P \quad (2)$$

$$G_1 z_j \le \sum_{p\in P} Q_p x_{pj} \le G_2 z_j \quad \forall j \in J \quad (3)$$

$$P_1 z_j \le \sum_{p\in P} x_{pj} \le P_2 z_j \quad \forall\ j \in J \quad (4)$$

$$M y_{jk} \ge \sum_{p\in P} c_{pk} x_{pj} \quad \forall\ j \in J, k \in K \quad (5)$$

目标函数使得波次包含的不同的**库区**数量总和最少；约束(1)保证分配的波次数量；约束(2)保证每个包裹都安排到波次中去；约束(3)保证单一波次商品件数在区间$[G_1, G_2]$中；约束(4)保证单一波次包裹数量在区间$[P_1, P_2]$中；约束(5)确认波次 j 是否用到**库区** k。

波次优化(BatchOpt)

- 问题进一步讨论

测试案例包含库区数量为22,如果按照version 2的思路,用库区构建模式(Pattern),模式的数量为24927,统计模型规模如下表:

model	变量数	约束数
version 1	6182995	58637
version 2	11877	963
version 3	2669650	27710

version 3与 version 1面临类似的问题,模型规模会随着包裹和波次数量增加而快速增加,因此需要寻找一种相对更快地处理方式。

波次优化(BatchOpt)

- 问题进一步讨论

观察全部输入数据,对仓库和库区统计如右表,发现判断两个包裹是否属于同一库区,可以先判断它们是否属于同一仓库,如果属于不同仓库,那么这两包裹一定属于不同库区。因此将问题分解为两阶段处理:

第一阶段: 按仓库确认每个波次使用模式(按仓库的pattern)的数量,对应模型(version 2);

第二阶段: 在第一阶段的基础上,大幅度缩减模型(version 1/version 3)规模,进而求解获得最终结果。

pakage_no	warehouse	goods_qty
302826075	2号仓-13	2
302826075	2号仓-23	1
302826075	5号仓-3	1
311001535	2号仓-24	1
311001535	佳明仓-3	1

仓库	区域
2号仓	11,12,13,14,15,22,23,24,25
4号仓	1,2
5号仓	1,2,3,4
佳明仓	2,3,4,5,6

波次优化(BatchOpt)

- 问题进一步讨论

第一阶段： 直接求解模型(version 2)；

第二阶段： 通过第一阶段的计算,可以获得模型(version 4)：

参数：
set $\bar{J}$ 用到的波次集合
set $\bar{J}_p$ 用到的波次使用模式(仓库)集合
$set\ I_p$ 模式p包含的包裹集合
$set\ K$ 库区集合
c_{ik} 包裹 i 是否有区域 k的商品
$\bar{x}_{pj}$ 波次$j(j \in \bar{J})$使用**模式p的次数**
M 足够大的数

变量：
x_{ij} $binary$,包裹 i 是否属于波次 j
y_{jk} $binary$,波次 j 是否用到区域 k

$$minimize \sum_{j\in\bar{J}}\sum_{k\in K} y_{jk}$$

$$\sum_{i\in I_p} x_{ij} = \bar{x}_{pj} \quad \forall j \in \bar{J}, p \in \bar{J}_p \quad (1)$$

$$\sum_{j\in\bar{J}} x_{ij} = 1 \quad \forall i \in I_p, p \in \bar{J}_p \quad (2)$$

$$My_{jk} \geq \sum_{i\in I_p} c_{ik}x_{ij} \quad \forall j \in \bar{J}, k \in K, p \in \bar{J}_p \quad (3)$$

目标函数使得波次包含的不同的库区数量总和最少；约束(1)保证包裹数量要和模式使用次数匹配；约束(2)保证每个包裹都安排到波次中去；约束(3)确认波 j是否用到库区 k。

波次优化(BatchOpt)

- 问题进一步讨论

测试案例包裹数量为57780,波次数量为107,仓库数量为4,库区数量为22,[G1,G2]=[1800,3000],[P1,P2]=[500,550],详细测试结果,对应测试代码:

波次数量	包裹数量	包裹件数	仓库数量	库区数量 (version 4)	运行时间(s)	第一阶段GAP	第二阶段GAP
107	[500, 550]	[1800, 3000]	251	1591	225.922	8.7649%	58.0767%
[100, 120]	[500, 550]	[1800, 3000]	250	1584	225.552	8.4000%	59.0278%

波次优化(BatchOpt)

- 问题进一步讨论

第二阶段： 模型(version 4)考虑了具体包裹,在本阶段中也可以引入模式(库区),构建模型(version 5)。

参数： $set\ \bar{J}$ 用到的波次集合

$set\ \bar{J}_p$ 用到的波次使用模式(仓库)集合

$set\ I_p$ 模式(仓库)p包含的库区模式集合

$set\ K$ 仓库集合

q_i 库区模式i的包裹的数量

c_{ik} 库区模式i 是否涉及区域 k

$\bar{x}_{pj}$ 波次$j(j \in \bar{J})$使用模式p的次数

M 足够大的数

变量： x_{ij} $integer$,波次 j 使用模式i的次数

y_{jk} $binary$,波次 j 是否用到区域 k

$$minimize \sum_{j\in\bar{J}}\sum_{k\in K} y_{jk}$$

$$\sum_{i\in I_p} x_{ij} = \bar{x}_{pj} \quad \forall\, j \in \bar{J}, p \in \bar{J}_p \qquad (1)$$

$$\sum_{j\in\bar{J}} x_{ij} = q_i \quad \forall\, i \in I_p, p \in \bar{J}_p \qquad (2)$$

$$My_{jk} \geq \sum_{i\in I_p} c_{ik}x_{ij} \quad \forall\, j \in \bar{J}, k \in K, p \in \bar{J}_p \qquad (3)$$

目标函数使得波次包含的不同的库区数量总和最少；约束(1)保证模式(库区)数量要和模式(仓库)使用次数匹配；约束(2)保证模式(库区)使用量和包裹数量一致；约束(3)确认波次 j是否用到库区 k。

波次优化(BatchOpt)

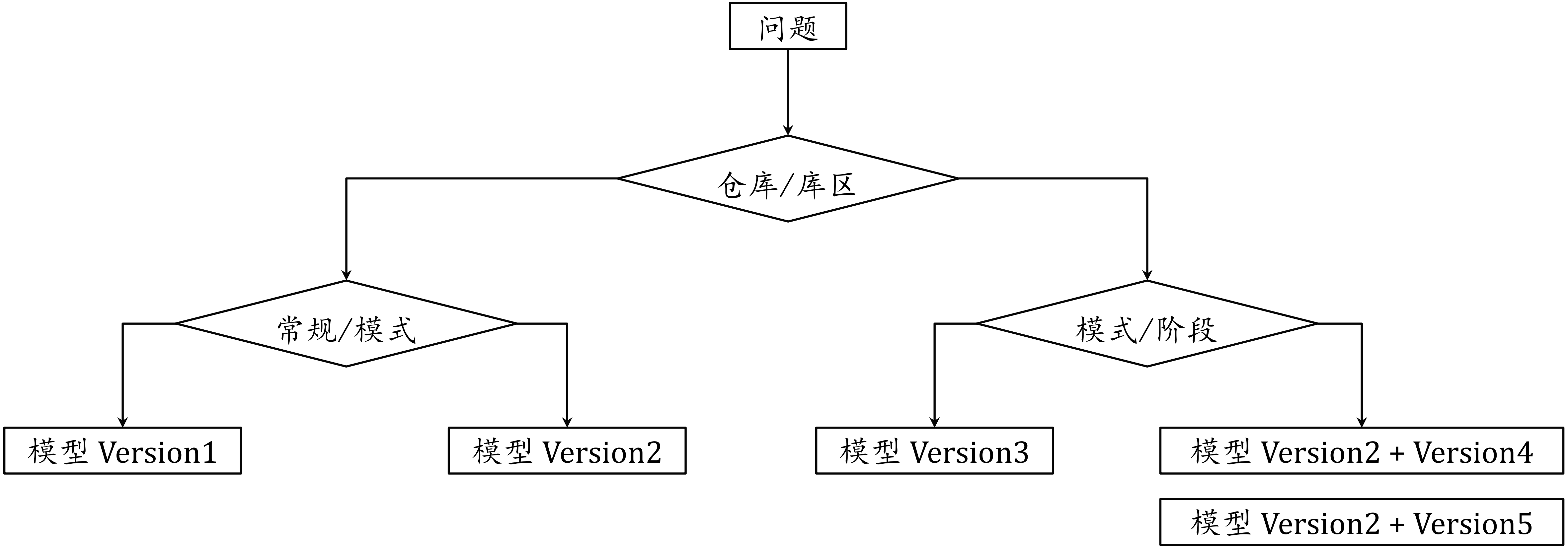
- 问题进一步讨论

测试案例包裹数量为57780,波次数量为107,仓库数量为4,库区数量为22,[G1,G2]=[1800,3000],[P1,P2]=[500,550],详细测试结果,对应测试代码:

波次数量	包裹数量	包裹件数	仓库数量	库区数量 (version 5)	运行时间(s)	第一阶段GAP	第二阶段GAP
107	[500, 550]	[1800, 3000]	251	1591	217.829	8.7649%	57.3853%
[100, 120]	[500, 550]	[1800, 3000]	250	1511	217.679	8.4000%	56.3865%

波次优化(BatchOpt)

- 案例小结



订单配送案例

- 问题描述

在计划期内，企业需要将该周期内所有的订单按照要求配送到客户手中。通常情况下，市场中存在众多的物流供应商，且不同物流供应商零担和整车的规格和费用可能会存在差异，因此如何选择合适的物流供应商使得总配送成本最低是企业需要面对的问题。

从企业的角度，需要考虑如下两个问题，且这两个问题相互影响：

(1) 不同订单之间如何组合；

(2) 挑选物流供应商以及配送方式(零担或整车)。

EDGESTONE 刃之砺 www.gurobi.cn

订单配送案例

- 输入数据

订单信息：起终点，重量，配送时间等数据。

DN#发货单号	N/W (T)净重	G/W (T)毛重	City	ETD Plant 预计提货日期	ETA期望达到日期
1	5.25	5.55	上海	2018/1/25	2018/1/26
2	3.00	3.07	广州	2018/1/23	2018/1/25
3	2.00	2.05	广州	2018/1/23	2018/1/25
4	15.00	15.35	广州	2018/1/23	2018/1/25
5	22.98	22.98	东莞	2018/1/24	2018/1/28

物流供应商信息：所有供应商不同车型的报价，载重等数据。例如某个物流供应商给出的报价方案：

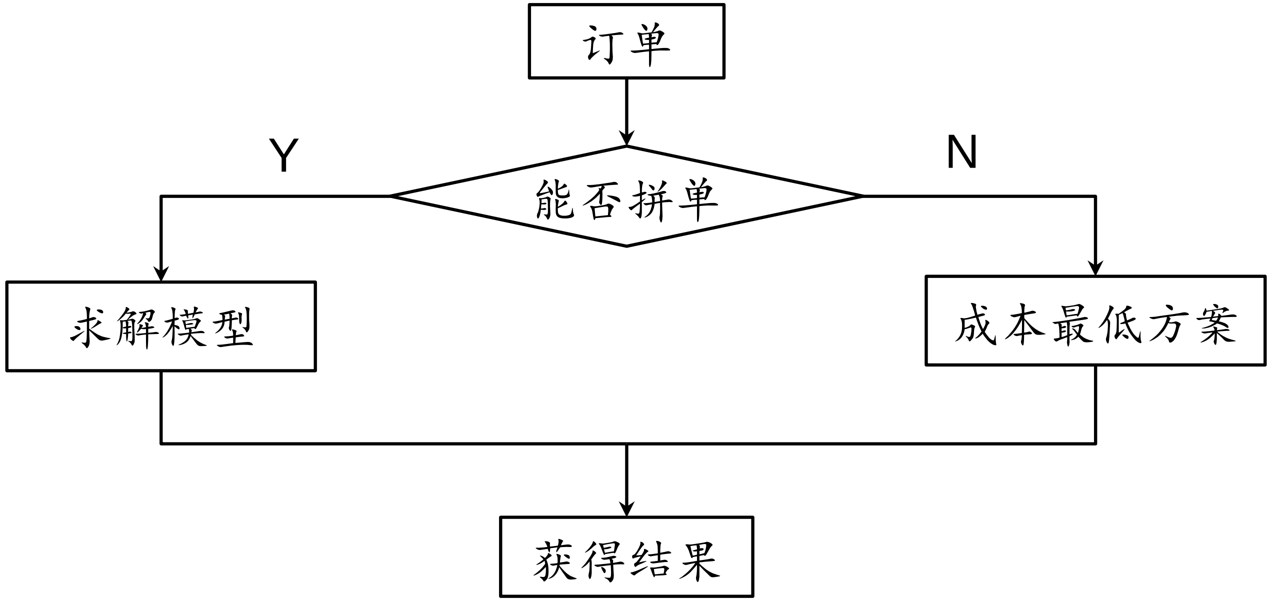
南京到	距离	Below 500KG (Sample)	1-3MT	3-5MT	5-10MT	10-20MT	>20MT	零担时间	2MT栏板	2MT箱式	2MT其他	5MT栏板	5MT箱式	5MT其他	10MT栏板	10MT箱式	10MT其他	15MT栏板	15MT箱式	15MT其他	20MT栏板	20MT箱式	20MT其他	25MT栏板	25MT箱式	25MT其他	30MT栏板	30MT箱式	30MT其他	32MT栏板	32MT箱式	32MT其他	整车时间
嘉兴	115	450	450	390	330			2	945	973		1180	1134		1350	1377		1750	1802		1800	1836	2340	2250	2295		2610	2662					2
无锡	138	410	410	350	290			2	1215	1251		1350	1458		1600	1632		1900	1957		2100	2142	2730	2625	2677		3045	3105					2
杭州	190	500	500	440	380			2	1440	1483		1600	1728		1850	1887		2250	2317		2700	2754	3510	3375	3442		3915	3993					2
江阴	182	450	450	390	330			2	1440	1483		1600	1728		1850	1887		2250	2317		2700	2754	3510	3375	3442		3915	3993					2
武汉	856	566.5	595	555	530			3				4700	4935		5250	5355		6200	6386		8040	8200		8442	8610		9792	9987					3.5
青岛	798	566.5	595	555	530			3				4700	4935		5250	5355		6500	6695		8040	8200		8450	8619		9802	9998					3.5
济南	898	640	700	640	590			4				4800	5040		5400	5508		7150	7364		8580	8751		9009	9189		10450	10659					4
郑州	985	710	800	740	690			5				5215	5475.7		6015	6135.3		7765	7997		9318	9504		9783.9	9979		11349	11575					5
福州	905	710	800	740	690			4				5800	6090		7000	7140		8750	9012		10500	10710		11025	11245		12789	13044					4

订单配送案例

- 问题思路

对计划周期内所有订单来说，如果给出一种可行的订单组合方案，根据物流供应商的报价，企业可以很容易制定出总成本最低的配送方案。

对单个特定的订单而言，可以很容易判断与其它订单的关系：



订单配送案例

- 问题思路

对计划期内的订单来说,最难处理的情况是所有的订单都可以拼单(相同起终点和相同配送时间),这时所有的订单都会出现在模型中,极大的增加了问题规模和求解难度。但现实情况,订单往往会有不同的起终点和配送时间,因此能够拼单的订单量不大,例如有{A,B,C,D,E,F,G,H}共8个订单,根据订单信息可以将订单分为{A},{B, H},{C},{D},{E,F},{G}共6个订单集合。只有元素为多个的集合可以拼单,这时由模型决定是否拼单以及配送方式。对不能拼单的订单可以直接找到最合适的配送方案(价格最低)。

订单配送案例

- 预处理(不可以拼单的订单)

目的：直接找到成本最低的配送方案，即确定由哪个物流供应商采取何种方式(整车或零担)配送。

方法：(1) 零担运输，找出可以用的零担车型，主要考虑时间，地点和载量的匹配。挑选出成本最低的物流供应商。

(2) 整车运输，同理找出可用的整车车型，注意考虑载量约束时需要满足**整车率**。从可行方案中挑选出成本最低的配送方案。

订单配送案例

- 预处理(可以拼单的订单)

  目的：以某一订单为标准，识别所有可以和该订单拼单的订单，把这些订单组成一个集合，在该集合上建立数学模型。

  方法：对某一特定订单，遍历所有未完成订单，搜索可以拼单的订单。能否拼单取决于起终点和配送时间是否匹配。为了进一步减少模型中的变量数量，对每一个物流供应商，找到所有能够使用的零担和整车车型，不能使用的车型不会出现在变量中。

订单配送案例

- 拼单订单模型

Model Parameters :

O the set of Orders.

P the set of logistics providers.

V the set of types of trucks of a logistics provider.

M the set of the number of available trucks of a type of a logistics provider.

$\bar{c}_p^v$ the cost of the v-type truck of logistics provider p.

w_i the weight of the i-th order.

W_p^v the capacity of v-type truck of logistics provider p.

订单配送案例

- 拼单订单模型

Model Variables :

x_{ip}^{vm} binary variables, equals one if the order i is assigned to the m-th truck of v-type of logistics provider p, otherwise zero.

y_{p}^{vm} binary variables, equals one if the m-th truck of v-type of logistics provider p is used, otherwise zero.

订单配送案例

- 拼单订单模型

$$minimize \quad \sum_{p,v,m} \bar{c}_p^v x_{ip}^{vm}$$

Constraint 1: all orders have to be delivered.

$$\sum_{p,v,m} x_{ip}^{vm} = 1, \ \forall i \in O$$

Constraint 2: the capacity limitation of the m-th truck which is used.

$$\sum_i w_i x_{ip}^{vm} \leq W_p^v y_p^{vm}, \ \forall p \in P, v \in V, m \in M$$

Constraint 3: the m-th truck of v-type of logistics provider p is used

$$\sum_i x_{ip}^{vm} \leq M y_p^{vm}, \ \forall p \in P, v \in V, m \in M$$

Constraint 4: variables

$$x_{ip}^{vm}, y_p^{vm} \in \{0,1\}, \forall i \in O, p \in P, v \in V, m \in M$$

订单配送案例

- 测试案例数值计算

具体的配送方案：主要包括选择的物流供应商，车型，运输方式，拼单等信息

订单编号	N/W (T) 净重	G/W (T) 毛重	预计提货日期	期望到达日期	目的地	物流商	使用车型	运输方式	组合订单编号	费用
1	5.25	5.55	2018/01/25	2018/01/26	上海	物流商1	5-10MT	零担		826.95
2	3	3.069	2018/01/23	2018/01/25	广州	物流商2	5-10MT	零担	[224]	1749.33
3	2	2.046	2018/01/23	2018/01/25	广州	物流商2	5-10MT	零担	[192]	1166.22

订单配送案例

- 方案优势

(1) 将所有订单按照是否能够拼单分为两大类。对不能拼单的订单通过简单的处理直接找到最优的配送方案;

(2) 对能够拼单的订单,建立了数学规划模型,并借助预处理进一步缩减变量数量。并且该模型可以直接被Gurobi求解;

(3) 考虑到实际能够拼单的量不是很大,该方案可以很快得到问题的最优解。

谢谢各位对 Gurobi 中文网络课程的支持。

我们会不断推出专题培训,敬请关注

www.gurobi.cn