Uncertainty Modeling and Optimization-不确定性建模与优化-应用篇

0. 前言

学习知识是需要巩固和应用的。尤其是，当目标的研究方向本身就是应用而非理论。第一部分的理论知识已经介绍了CCP, SP, RO, ARO, DRO, ADRO的很多知识，但除了简单的例子，还并没有介绍应用。在应用篇，我会结合一些管理科学方向的，发表于顶级期刊的论文，从建模、算法两个角度介绍这些方法在管理类问题中的应用。暂时想要介绍的应用和文章包括但不限于：

- Facility Location (Chun Cheng的IJOO和Omega，Delage的TS)
- Vehicle Routing (Yu Zhang和Melvyn Sim的两篇文章，MP和OR)
- Branch and Price 求解 (TS)
- RO和SP建模对比 (Yossiri, TS)
- 前沿的建模 (Chan AED，Chun Cheng 无人机)

这些文章使用的方法大多数是ARO、DRO、ADRO的进阶内容。也是我不太熟悉的部分。我将在大概一个星期时间里分几章介绍完，并在日后视情况补充。希望我个人在研究这些文章的过程中能够对这些建模方法有更深的了解，有能力做接下来的工作。

大家一起学习，一起进步！

1. Facility Location

在设施选址问题中存在很多不确定性。例如客户点需求不确定，设施的毁坏风险等。这类问题往往是multi-stage的，所以基本使用ARO。

Robust Facility Location Under Disruptions

首先介绍[1]。这篇文章主要解决一种disruption的不确定性，意思是开放的工厂可能因为一些原因毁坏，毁坏后就无法继续生产。[1]对这种情况设计了ARO模型和基于Linear Decision Rule（LDR）的近似模型，通过基于对偶和枚举的C&CG(行列生成)和benders求解。

引入以下变量和参数，并构建确定性问题的模型：

y_i：是否选择在i开工厂。 x_{ij}：j的货物从i处满足的比例。

f_i：在i开工厂的固定成本。 h_i：客户i的需求。

d_{ij}：i到j的单位运输成本。 C_j：工厂j的容量约束。

$$
\begin{aligned}
\min_{x,y}\quad & \sum_j f_j \cdot y_j + \sum_i \sum_j h_i \cdot d_{ij} \cdot x_{ij} \\
\text{s.t.}\quad & \sum_j x_{ij} \geq 1, \quad \forall i \\
& x_{ij} \leq y_j, \quad \forall i,j \\
& \sum_i h_i \cdot x_{ij} \leq C_j \cdot y_j, \quad \forall j. \\
& y_j \in \{0,1\}, \quad \forall j \\
& x_{ij} \geq 0, \quad \forall i,j
\end{aligned}
$$

目标函数是最小化开厂固定成本和货物运输成本。每个客户必须满足所有需求。只有开厂才能生产货物。每个厂有生产容量上限。当我们去掉和加上约束3时，问题分别为uncapacitated/capacitated fixed-charge location problem (UFLP/CFLP)。

考虑budgeted uncertainty set，构造如下ARO模型：

$$Z(k) = \left\{ z \in \{0,1\}^J : \sum_j z_j \le k \right\}$$

$$\begin{aligned} \min_{y} \quad & \sum_j f_j \cdot y_j + \sup_{z \in Z(k)} g(y,z) \\ \text{s.t.} \quad & y_j \in \{0,1\}, \quad \forall j \end{aligned}$$

$$\begin{aligned} g(y,z) = \min_{x,u} \quad & \sum_i \sum_j h_i \cdot d_{ij} \cdot x_{ij} + \sum_i p_i h_i u_i \\ \text{s.t.} \quad & \sum_j x_{ij} + u_i \ge 1, \quad \forall i \\ & x_{ij} \le y_j (1 - z_j), \quad \forall i,j \\ & \sum_i h_i \cdot x_{ij} \le C_j \cdot y_j (1 - z_j), \quad \forall j. \\ & x_{ij} \ge 0, \quad \forall i,j \\ & u_i \ge 0, \quad \forall i. \end{aligned}$$

问题是熟悉的两阶段形式。这里的不确定集虽然是budgeted，但和之前的形式不完全一样，可以对比看看。同样，子问题的约束3决定问题为UFLP还是CFLP。在新的问题中，$z_j = 1$表示一个工厂毁坏。工厂毁坏后，一些客户可能无法被完全服务。u_i表示无法服务的部分。这部分货物被加上p_i项的惩罚值。

为了说明ARO的效果，我们把static RO的模型写出来对比：

$$
\begin{aligned}
\min_{y,x,u}\quad & \sum_j f_j \cdot y_j + \sum_i \sum_j h_i \cdot d_{ij} \cdot x_{ij} + \sum_i p_i h_i u_i \\
\text{s.t.}\quad & \sum_j x_{ij} + u_i \ge 1, && \forall i \\
& x_{ij} \le y_j (1-z_j), && \forall i,j, \forall z \in Z. \\
& \sum_i h_i \cdot x_{ij} \le C_j \cdot y_j (1-z_j), && \forall j. \ \forall z \in Z. \\
& x_{ij} \ge 0, && \forall i,j \\
& u_i \ge 0, && \forall i. \\
& y_j \in \{0,1\}, && \forall j
\end{aligned}
$$

简单的说，就是将二阶段问题直接写成一阶段。这显然与原问题不等价，扩大了约束的数量，使得解更保守了。与原问题对应的解应该将x_{ij}写成$x_{ij}(z)$（u_i同理）。这是对ARO的复习。

有以下两个性质：

1. 对于给定的一阶段解$\hat{y}$，对于两种场景$z_1, z_2 \in Z(k)$，如果z_1可用的工厂的集合（即，$\hat{y}_j = 1, z_j = 0$）是z_2的子集，则z_1的第二阶段成本（成为recourse cost）要大于z_2。
2. 记一阶段解中可用工厂数量为m，即$\sum_j \hat{y}_j = m$。若$m > k$，则在二阶段解中，有一些工厂能服务客户；若$m \le k$，则在二阶段解中所有工厂关闭，所有客户无法满足。

这很好理解，因为二阶段的鲁棒问题是worst case，一定会尽量关闭更多工厂。关了更多工厂，成本一定更大。第二个性质可以用于后续优化算法。

ARO的C&CG和benders算法在理论篇第五章都有介绍。C&CG在原问题中逐步加入不同场景下子问题的原始约束：

$$
\begin{aligned}
\phi = \min_{\mathbf{y}, s, \{\mathbf{x}^l\}_{l=1}^n, \{\mathbf{u}^l\}_{l=1}^n} \quad & s, \\
\text{s.t.}\quad & s \ge \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} h_i d_{ij} x_{ij}^l + \sum_{i\in I} p_i h_i u_i^l && \forall l \in \{1,\dots,n\}, \\
& \sum_{j\in J} x_{ij}^l + u_i^l \ge 1 && \forall l \in \{1,\dots,n\}, i \in I, \\
& x_{ij}^l \le y_j(1-z_j^l) && \forall l \in \{1,\dots,n\}, i \in I, j \in J, \\
& y_j \in \{0,1\} && \forall j \in J, \\
& x_{ij}^l \ge 0 && \forall l \in \{1,\dots,n\},\ i \in I, j \in J, \\
& u_i^l \ge 0 && \forall l \in \{1,\dots,n\}, i \in I.
\end{aligned}
$$

当一阶段解满足$\sum_j \hat{y}_j > k$时，子问题在UFLP下为如下的最小流问题形式：

$$
\begin{aligned}
\psi = \min_{\mathbf{x},\mathbf{u}} & \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\sum_{j\in \bar{J}} h_i d_{ij} x_{ij} + \sum_{i\in I} p_i h_i u_i, \\
\text{s.t.} \quad & \sum_{j\in\bar{J}} x_{ij} + u_i \geq 1 && \forall i\in I, \\
& x_{ij}, u_i \geq 0 && \forall i\in I, j\in\bar{J}.
\end{aligned}
$$

由于这里的不确定集非常简单，只有有限个场景，因此可以选择对所有场景的z做枚举，而不是作为决策变量。（这里有一点疑问，其实这种不确定集下的z的数量应该是指数增长的）

对子问题的求解，文中给出了三种方法。第一种是直接求解。第二种是利用python包NetworkX 2.0求解。第三种是对不同的i划分成多个子问题，因为每个i在UFLP中是相互独立的。但是，在CFLP中，由于工厂的容量限制，每个i不再独立。综合来看，第二种方法不比第三种差太多，但都优于第一种。

由性质2，如果不满足，直接按惩罚值计算即可。对CFLP，只需要加入工厂容量的约束。

算法的一种加速策略是Multiple Scenario Generation，也就是每次往主问题中加入多个场景。这与branch and price中往主问题里加入多个reduced cost为负的列类似。如果将z作为决策变量，每次解子问题后只能得到一个场景，需要选择近似场景加入主问题；但我们在第一种情况中选择对所有z做枚举，因此不需要这种近似。

C&CG的初始解可以从确定性问题的初始解开始。

此外，[1]还对比了基于对偶的C&CG算法和benders分解两种方法和上述基于枚举的C&CG算法的结果。

基于对偶的C&CG算法将z作为决策变量的方法是对子问题求对偶，将两个max合并。这个我们之前也讲过。因为不涉及DRO中的半无限规划，求对偶的过程只是LP，比较简单，直接给出：

$$
\begin{aligned}
\psi = \max_{\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\beta}} & \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\lambda_i - \sum_{i\in I}\sum_{j\in J}\hat{y}_j(1-z_j)\beta_{ij}, \\
\text{s.t.} \quad & \lambda_i - \beta_{ij} \leq h_i d_{ij} && \forall i\in I, j\in J, \\
& \lambda_i \leq p_i h_i && \forall i\in I, \\
& \sum_{j\in J} z_j \leq k, \\
& z_j \in \{0,1\} && \forall i\in I, \\
& \lambda_i \geq 0 && \forall i\in I, \\
& \beta_{ij} \geq 0 && \forall i\in I, j\in J.
\end{aligned}
$$

其中目标函数的决策变量相乘的形式可以用大M线性化。而第一种基于枚举的方法不需要M，这也是他的一大优势。

基于benders分解的方法求解和上面一样的子问题，只是加入主问题的约束是对偶之后的：

$$
\begin{aligned}
\phi = \min_{\mathbf{y},s} \; & s, \\
\text{s.t.} \quad & s \geq \sum_{j\in J} f_j y_j + \sum_{i\in I} \hat{\lambda}_i^l - \sum_{i\in I}\sum_{j\in J} \hat{\beta}_{ij}^l y_j (1-\hat{z}_j^l) \qquad \forall l \in \{1,\dots,n\}, \\
& y_j \in \{0,1\} \qquad \forall j \in J,
\end{aligned}
$$

这也在理论篇第五章讲过。总体来说，benders应该不如C&CG，这是那篇OR letter的结论。

在前面的问题中，第二阶段决策变量x_{ij}不受随机性因素影响。我们之前提到过一种处理方法，叫Affine Decision Rule（LDR）。对LDR，首先需要将不确定集表示成convex hull的形式 $\mathbb{Z}'(k) = \left\{\mathbf{z} \in \mathbb{R}^{|J|} : 0 \leq \mathbf{z} \leq 1, \sum_{j\in J} z_j \leq k\right\}$，然后令二阶段决策变量表示为仿射函数形式 $x_{ij} = \mathbf{W}_{ij}^T\mathbf{z} + w_{ij}$，$u_i = \mathbf{A}_i^T\mathbf{z} + a_i$。注意在LDR中，$W, w, A, a$都是决策变量。最后得到经典RO形式的问题：

$$
\begin{aligned}
\min_{\mathbf{y},\mathbf{W},\mathbf{w},\mathbf{A},\mathbf{a},s} \; & s, \\
\text{s.t.} \quad & s \geq \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} h_i d_{ij}(\mathbf{W}_{ij}^T\mathbf{z} + w_{ij}) + \sum_{i\in I} p_i h_i (\mathbf{A}_i^T\mathbf{z} + a_i) && \forall \mathbf{z} \in \mathbb{Z}'(k), \\
& \sum_{j\in J} (\mathbf{W}_{ij}^T\mathbf{z} + w_{ij}) + (\mathbf{A}_i^T\mathbf{z} + a_i) \geq 1 && \forall \mathbf{z} \in \mathbb{Z}'(k), i \in I, \\
& \mathbf{W}_{ij}^T\mathbf{z} + w_{ij} \leq y_j(1-z_j) && \forall \mathbf{z} \in \mathbb{Z}'(k), i \in I, j \in J, \\
& y_j \in \{0,1\} && \forall j \in J, \\
& \mathbf{W}_{ij}^T\mathbf{z} + w_{ij} \geq 0 && \forall \mathbf{z} \in \mathbb{Z}'(k), i \in I, j \in J, \\
& \mathbf{A}_i^T\mathbf{z} + a_i \geq 0 && \forall \mathbf{z} \in \mathbb{Z}'(k), i \in I.
\end{aligned}
$$

这个问题可以很快转化成LP。

现在只需要证明convex hull的$Z'(k)$等价于$Z(k)$。这也很好解释，简单的说，在worst case下z_j肯定取0或1，不可能是中间值。因此对于原问题，两个不确定集是等价的。

[1]引用了其他文章中使用多个不确定集的拓展。这可能是应用类文章的一个可考虑方向。

上面的基于枚举的方法可以避免一些对偶理论无法使用的情况，比如，第二阶段决策存在整数变量。此时对偶无法进行，因此基于对偶的C&CG和benders无法使用。但是枚举依旧可行。

Robust Facility Location Under Demand Uncertainty and Disruptions

从[2]的文章题目中就能看出，相比于[1]最大的区别就是增加了客户需求的uncertainty。其实我觉得这个更重要，随机性更大。在上一节里，客户需求由参数h_i给定。在这里，h_i将转变为随机变量。我们假设它依旧符合budgeted uncertainty set。模型写作：

$$U_h(\Gamma)=\left\{h\in\mathbb{R}_+^I:\ h_i=\bar{h}_i+\theta_i\cdot\Delta h_i,\ 0\le\theta_i\le 1,\ \sum_i\theta_i\le\Gamma\right\}$$

$$Z(k)=\left\{z\in\{0,1\}^J:\ \sum_j z_j\le k\right\}.$$

$$W=\left\{(h,z)\in\mathbb{R}_+^I\times\{0,1\}^J:\ h\in U_h(\Gamma),\ z\in Z(k)\right\}.$$

一阶段:
$$\begin{aligned}\min_{y}\quad & \sum_j f_j\cdot y_j+\max_{(h,z)\in W} g(y,h,z)\\ \text{s.t.}\quad & y_j\in\{0,1\},\ \forall j.\end{aligned}$$

$$\begin{aligned}g(y,h,z)=\min_{x,u}\quad & \sum_i\sum_j d_{ij}x_{ij}+\sum_i p_i u_i\\ \text{s.t.}\quad & \sum_j x_{ij}+u_i\ge h_i,\ \forall i\\ & \sum_i x_{ij}\le c_j y_j(1-z_j),\ \forall j\\ & x_{ij}\ge 0,\ \forall i,j\\ & u_i\ge 0,\ \forall i\end{aligned}$$

模型和上一节基本一致。这应该是整个系列中第一次遇到多组随机变量分别属于不同不确定集的情况了，虽然二者都属于budgeted uncertainty set。

继续使用C&CG求解。由于这次的不确定集不适合枚举，所以采用传统的方法，将不确定参数作为决策变量。使用KKT定理得到子问题：

$$
\begin{aligned}
\psi = \max_{\substack{\boldsymbol{x},\boldsymbol{u},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{w}^{\alpha},\\ \boldsymbol{w}^{\beta},\boldsymbol{w}^{x},\boldsymbol{w}^{u},\boldsymbol{z},\boldsymbol{\theta}}} \quad & \sum_{j\in J} f_j\hat{y}_j + \sum_{i\in I}\sum_{j\in J} d_{ij}x_{ij} + \sum_{i\in I} p_i u_i \\
\text{s.t.} \quad & \sum_{j\in J} x_{ij} + u_i \ge \bar{h}_i + \theta_i h_i^{\Delta} && \forall i\in I, \\
& \sum_{i\in I} x_{ij} \le C_j\hat{y}_j(1-z_j) && \forall j\in J, \\
& \alpha_i - \beta_j \le d_{ij} && \forall i\in I,\ j\in J, \\
& \alpha_i \le p_i && \forall i\in I, \\
& \sum_{j\in J} x_{ij} + u_i \le \bar{h}_i + \theta_i h_i^{\Delta} + M_i^{\alpha}(1-w_i^{\alpha}) && \forall i\in I, \\
& \alpha_i \le M_i^{\alpha} w_i^{\alpha} && \forall i\in I, \\
& \sum_{i\in I} x_{ij} \ge C_j\hat{y}_j(1-z_j) + M_j^{\beta}(w_j^{\beta}-1) && \forall j\in J, \\
& \beta_j \le M_j^{\beta} w_j^{\beta} && \forall j\in J, \\
& \alpha_i - \beta_j \ge d_{ij} + M_{ij}^{x}(w_{ij}^{x}-1) && \forall i\in I,\ j\in J \\
& x_{ij} \le M_{ij}^{x} w_{ij}^{x} && \forall i\in I,\ j\in J \\
& \alpha_i \ge p_i + M_i^{u}(w_i^{u}-1) && \forall i\in I, \\
& u_i \le M_i^{u} w_i^{u} && \forall i\in I, \\
& \theta_i \le 1 && \forall i\in I, \\
& \sum_{i\in I} \theta_i \le \Gamma_h, \\
& \sum_{j\in J} z_j \le k, \\
& x_{ij}, u_i, \alpha_i, \beta_j, \theta_i \ge 0 && \forall i\in I,\ j\in J, \\
& w_i^{\alpha}, w_j^{\beta}, w_{ij}^{x}, w_i^{u}, z_j \in \{0,1\} && \forall i\in I,\ j\in J.
\end{aligned}
$$

We set $M_i^{\alpha} = p_i, M_j^{\beta} = \max\{C_j, \max_i\{d_{ij}(\bar{h}_i + h_i^{\Delta}), p_i(\bar{h}_i + h_i^{\Delta})\}\}, M_{ij}^{x} = \max\{C_j, d_{ij}(\bar{h}_i + h_i^{\Delta}), p_i(\bar{h}_i + h_i^{\Delta})\} + d_{ij}, M_i^{u} = \max\{p_i, \bar{h}_i + h_i^{\Delta}\}$.

突然想起来前一节没用过KKT定理，而是使用对偶定理。什么时候应该用KKT，什么时候应该用对偶呢？这个问题我还没有答案。从理论上将，KKT应该比对偶适用范围更广，但对偶求解一般更快。

[2]还将问题拓展到另一类问题上，这里略过。

Robust Location–Transportation Problem

Location–Transportation Problem (LTP) 是指在一系列facility location里选择一些open，以服务一系列customer location的需求。问题分为多个period。其中，open工厂需要成本，工厂有生产容量，顾客有需求量。满足需求可以获利，不满足不损失；生产产品需要成本，不能存储产品到下一个时期。工厂的产品可以服务不同客户，有运输成本。引入以下参数：

L: facility 数量.　N: customer 数量.　T: period 数量.

Z_i: 工厂i的生产容量.　p_i^t: t时期工厂i生产的货物量.　I_i: 是否在i开厂.

γ_{ij}^t: t时期工厂i给客户j的产品数.　ζ_i^t: t时期客户i的需求.

η: 产品单位售价.　c_i: 产品在工厂i的单位成本.

$C_{0i} Z_i + f_i I_i$: 开厂的变动成本＋固定成本.

在deterministic环境下，问题的MILP模型如下：

$$
\begin{aligned}
\max_{I, Z, \gamma, p} \quad & \sum_{t=1}^{T}\left[\sum_{i=1}^{L}\sum_{j=1}^{N}(\eta - d_{ij})\gamma_{ij}^t - c^T p^t\right] - (C_0^T Z + f^T I) \\
\text{s.t.} \quad & \sum_i \gamma_{ij}^t \le \zeta_j, \quad \forall j, \forall t, \ \forall \zeta \in D. \quad \text{-- ② Static Robust} \\
& \sum_j \gamma_{ij}^t \le p_i^t, \quad \forall i, \forall t. \quad \text{— ① Deterministic} \\
& p^t \le Z, \quad \forall t \\
& \gamma^t \ge 0, \quad \forall t \\
& Z \le M \cdot I. \\
& I \in \{0, 1\}^L.
\end{aligned}
$$

其中第一个约束表示可以卖出的货物不能超过客户需求量上限，第二、三个约束表示单个工厂总生产量不超过上限，第五个约束表示只有建厂才能生产。这个问题与前面的CFLP的区别在于，目标函数是最大化利润；客户需求不一定要满足；客户需求为多阶段。大致来看和前面还是类似的。

添加红字部分之后，问题转化为经典RO模型。这显然是不合理的，因此有如下ARO模型：

$$
\max_{I,Z} \min_{\zeta\in\mathcal{D}} \sum_t h_t(I,Z,\zeta^t)-(c_0^T Z+f^T I)
$$

$$
\text{s.t.} \quad Z\le MI,\quad I\in\{0,1\}^L
$$

— ③ MRLTP.

$$
\begin{aligned}
h_t(I,Z,\zeta^t)=\max_{\gamma^t,p^t}\ & \sum_i\sum_j (r_j-d_{ij})\gamma_{ij}^t - c^T\cdot p^t\\
\text{s.t.}\ & \sum_i \gamma_{ij}^t \le \zeta_j,\quad \forall j\\
& \sum_j \gamma_{ij}^t \le p_i^t,\quad \forall i\\
& p^t\le Z,\\
& \gamma^t\ge 0,
\end{aligned}
$$

在这个问题的基础上，已经有前者提出了近似估计的方法：

$\gamma_{ij}^t := x_{ij}^t\zeta_j$　　x_{ij}^t: i到j的占ζ_j的比例.

$$
\begin{aligned}
\max_{I,Z,\gamma,p}\ \min_{\zeta\in\mathcal{D}}\ & \sum_{t=1}^T\left[\sum_{i=1}^L\sum_{j=1}^N (r_j-d_{ij})x_{ij}^t\zeta_j^t - C^T\cdot p^t\right]-(c_0^T Z+f^T\cdot I)\\
\text{s.t.}\ & \sum_j x_{ij}^t\zeta_j^t\le p_i^t,\quad \forall i,\ \forall t,\ \forall\zeta\in\mathcal{D}.\\
& p^t\le Z,\quad \forall t\\
& \sum_i x_{ij}^t\le 1,\quad \forall j,\ \forall t\\
& Z\le M\cdot I.\\
& x^t\ge 0.\quad \forall t.\\
& I\in\{0,1\}^L.
\end{aligned}
$$

— ④. FVB.

这个估计没有使用传统的LDR，而是将生产量转化成百分比。这显然不够好，感觉和直接转化没有差很多。[3]就在这个基础上使用LDR，提出了6个ARO模型进行优化。在后续模型中考虑budgeted不确定集。

第一组模型成为客户驱动的模型，将根据客户的需求信息来调整交付给该客户的货物的大小，也就是说$\gamma=f(\zeta)$。这个模型将问题5的线性函数$\gamma_{ij}^t := X_{ij}^t\zeta_j^t$转换成仿射函数$\gamma_{ij}^t := X_{ij}^t\zeta_j^t+W_{ij}^t$:

(RFVB1)

$$\underset{\mathbf{I},\mathbf{Z},\mathbf{X},\mathbf{W}}{\text{maximize}} \quad \min_{\zeta\in\mathcal{D}} \left\{ \sum_t \sum_i \sum_j (\eta - d_{ij} - c_i)\cdot(X_{ij}^t \zeta_j^t + W_{ij}^t) - (\mathbf{c}_0^T \mathbf{Z} + \mathbf{f}^T \mathbf{I}) \right\} \tag{6a}$$

subject to

$$\sum_i (X_{ij}^t \zeta_j^t + W_{ij}^t) \le \zeta_j^t, \quad \forall \boldsymbol{\zeta} \in \mathcal{D}, \forall j, \forall t, \tag{6b}$$

$$\sum_j (X_{ij}^t \zeta_j^t + W_{ij}^t) \le Z_i, \quad \forall \boldsymbol{\zeta} \in \mathcal{D}, \forall i, \forall t, \tag{6c}$$

$$X_{ij}^t \zeta_j^t + W_{ij}^t \ge 0, \quad \forall \boldsymbol{\zeta} \in \mathcal{D}, \forall i, \forall j, \forall t, \tag{6d}$$

$$\mathbf{Z} \le M\mathbf{I}, \quad \mathbf{I} \in \{0,1\}^L. \tag{6e}$$

然后做了一点简化，去掉了参数P^t。

为了适应分段放射函数的情况，也就是理论篇提到的分段放射函数决策准则，引入提升的budgeted不确定集，得到模型2。这我们应该是讲过的。

$$\mathcal{D} = \{\boldsymbol{\zeta} \in \mathbb{R}^{N\times T} \mid \exists (\boldsymbol{\zeta}^+, \boldsymbol{\zeta}^-) \in \mathcal{D}_2, \boldsymbol{\zeta} = \bar{\boldsymbol{\zeta}} + \boldsymbol{\zeta}^+ - \boldsymbol{\zeta}^-\},$$

where

$$\mathcal{D}_2 = \left\{ \begin{array}{l} (\boldsymbol{\zeta}^+, \boldsymbol{\zeta}^-) \\ \quad \in \mathbb{R}^{N\times T} \times \mathbb{R}^{N\times T} \end{array} \middle| \begin{array}{c} \exists (\boldsymbol{\delta}^+, \boldsymbol{\delta}^-) \in \mathbb{R}^{N\times T} \times \mathbb{R}^{N\times T}, \\ \boldsymbol{\delta}^+ \ge 0, \boldsymbol{\delta}^- \ge 0, \\ \|\boldsymbol{\delta}^+ + \boldsymbol{\delta}^-\|_\infty \le 1, \\ \|\boldsymbol{\delta}^+ + \boldsymbol{\delta}^-\|_1 \le \Gamma, \\ \zeta_j^{t+} = \hat{\zeta}_j^{t+} \delta_j^{t+}, \\ \zeta_j^{t-} = \hat{\zeta}_j^{t-} \delta_j^{t-} \quad \forall j \forall t \end{array} \right\}.$$

$$\underset{\mathbf{I},\mathbf{Z},\mathbf{X}^+,\mathbf{X}^-,\mathbf{W}}{\text{maximize}} \min_{(\boldsymbol{\zeta}^+,\boldsymbol{\zeta}^-)\in\mathscr{D}_2} \left\{ \sum_t \sum_i \sum_j (\eta - d_{ij} - c_i) \cdot (X_{ij}^{t+}\zeta_j^{t+} + X_{ij}^{t-}\zeta_j^{t-} + W_{ij}^t) - (\mathbf{c}_0^T\mathbf{Z} + \mathbf{f}^T\mathbf{I}) \right\} \quad (7a)$$

subject to

$$\sum_i (X_{ij}^{t+}\zeta_j^{t+} + X_{ij}^{t-}\zeta_j^{t-} + W_{ij}^t) \le \zeta_j^t, \quad \forall(\boldsymbol{\zeta}^+,\boldsymbol{\zeta}^-)\in\mathscr{D}_2, \forall j, \forall t, \quad (7b)$$

$$\sum_j (X_{ij}^{t+}\zeta_j^{t+} + X_{ij}^{t-}\zeta_j^{t-} + W_{ij}^t) \le Z_i, \quad \forall(\boldsymbol{\zeta}^+,\boldsymbol{\zeta}^-)\in\mathscr{D}_2, \forall i, \forall t, \quad (7c)$$

$$X_{ij}^{t+}\zeta_j^{t+} + X_{ij}^{t-}\zeta_j^{t-} + W_{ij}^t \ge 0, \quad \forall(\boldsymbol{\zeta}^+,\boldsymbol{\zeta}^-)\in\mathscr{D}_2, \forall i, \forall j, \forall t, \quad (7d)$$

$$\mathbf{Z} \le M\mathbf{I}, \quad \mathbf{I}\in\{0,1\}^L. \quad (7e)$$

budgeted的lifted不确定集在之前专门在例子里说明，这里不细讲了。

第二组决策准则成为市场驱动。其含义在于，之前送给客户j的货物γ_{ij}^t只由客户j的产量ζ_j^t决定，而市场驱动则意味着由所有客户，也就是向量ζ^t决定（其实下一组模型也很好猜了，就是剩下那个index的含义）。替换$\gamma_{ij}^t := \left(\mathbf{X}_{ij}^t\right)^T \zeta^t + W_{ij}^t$。注意这里的$X$是向量，表示都是同一个值的复制。这是源于这样的想法：当距离工厂较近的客户需求量增加时，应该将更多货物运到这些客户处，而不是距离工厂较远的客户处。注意距离由运输成本参数d_{ij}度量。因此得到模型3：

(AARC)

$$\underset{\mathbf{I},\mathbf{Z},\mathbf{X},\mathbf{W}}{\text{maximize}} \min_{\boldsymbol{\zeta}\in\mathscr{D}} \left\{ \sum_t \sum_i \sum_j (\eta - d_{ij} - c_i) \cdot ((\mathbf{X}_{ij}^t)^T\boldsymbol{\zeta}^t + W_{ij}^t) - (\mathbf{c}_0^T\mathbf{Z} + \mathbf{f}^T\mathbf{I}) \right\} \quad (8a)$$

$$\text{subject to} \quad \sum_i ((\mathbf{X}_{ij}^t)^T\boldsymbol{\zeta}^t + W_{ij}^t) \le \zeta_j^t, \quad \forall\boldsymbol{\zeta}\in\mathscr{D}, \forall j, \forall t, \quad (8b)$$

$$\sum_j ((\mathbf{X}_{ij}^t)^T\boldsymbol{\zeta}^t + W_{ij}^t) \le Z_i, \quad \forall\boldsymbol{\zeta}\in\mathscr{D}, \forall i, \forall t, \quad (8c)$$

$$(\mathbf{X}_{ij}^t)^T\boldsymbol{\zeta}^t + W_{ij}^t \ge 0, \quad \forall\boldsymbol{\zeta}\in\mathscr{D}, \forall i, \forall j, \forall t, \quad (8d)$$

$$\mathbf{Z} \le M\mathbf{I}, \quad \mathbf{I}\in\{0,1\}^L. \quad (8e)$$

考虑分段仿射准则得到模型4：

(LAARC)

$$\begin{aligned}
\underset{\mathbf{I},\mathbf{Z},\mathbf{X}^{+},\mathbf{X}^{-},\mathbf{W}}{\text{maximize}} \quad \min_{(\zeta^{+},\zeta^{-})\in\mathscr{D}_2} & \Bigg\{ \sum_t \sum_i \sum_j (\eta - d_{ij} - c_i) \\
& \cdot((\mathbf{X}_{ij}^{t+})^T \zeta^{t+} + (\mathbf{X}_{ij}^{t-})^T \zeta^{t-} + W_{ij}^t) \\
& - (\mathbf{c}_0^T \mathbf{Z} + \mathbf{f}^T \mathbf{I}) \Bigg\} \qquad \text{(9a)} \\
\text{subject to} \quad & \sum_i ((\mathbf{X}_{ij}^{t+})^T \zeta^{t+} + (\mathbf{X}_{ij}^{t-})^T \zeta^{t-} + W_{ij}^t) \le \zeta_j^t, \\
& \forall (\zeta^{+},\zeta^{-}) \in \mathscr{D}_2, \forall j, \forall t, \qquad \text{(9b)} \\
& \sum_j ((\mathbf{X}_{ij}^{t+})^T \zeta^{t+} + (\mathbf{X}_{ij}^{t-})^T \zeta^{t-} + W_{ij}^t) \le Z_i, \\
& \forall (\zeta^{+},\zeta^{-}) \in \mathscr{D}_2, \forall i, \forall t, \qquad \text{(9c)} \\
& (\mathbf{X}_{ij}^{t+})^T \zeta^{t+} + (\mathbf{X}_{ij}^{t-})^T \zeta^{t-} + W_{ij}^t \ge 0, \\
& \forall (\zeta^{+},\zeta^{-}) \in \mathscr{D}_2, \forall i, \forall j, \forall t, \qquad \text{(9d)} \\
& \mathbf{Z} \le M\mathbf{I}, \quad \mathbf{I} \in \{0,1\}^L. \qquad \text{(9e)}
\end{aligned}$$

[3]接下来第一次提出了一个方法，在二阶段问题里放松不确定性。准确的说，通过在不确定性存在的约束中添加额外的变量来减小其对问题的影响。

$$\begin{aligned}
h_t(\mathbf{I},\mathbf{Z},\zeta^t) = \max_{\mathbf{Y}^t,\mathbf{P}^t,\boldsymbol{\theta}^t} & \left\{ \sum_i \sum_j (\eta - d_{ij} - c_i) Y_{ij}^t - \underline{\sum_j u_j \theta_j^t} \right\} \qquad \text{(10a)} \\
\text{subject to} \quad & \sum_i Y_{ij}^t \le \zeta_j^t + \underline{\theta_j^t}, \quad \forall j, \qquad \text{(10b)} \\
& \sum_j Y_{ij}^t \le Z_i, \quad \forall i, \qquad \text{(10c)} \\
& \mathbf{Y}^t \ge 0, \quad \boldsymbol{\theta}^t \ge 0, \qquad \text{(10d)}
\end{aligned}$$

如图，二阶段问题相比于原问题3添加了红线部分的额外变量θ_j^t，表示如果违背该约束，目标函数会受到惩罚，也就是一个拉格朗日松弛。参数u取一个足够大的常数。顺便省略P^t。

类似γ，令$\theta_j^t := S_j^{t+}\zeta_j^{t+} + S_j^{t-}\zeta_j^{t-}$的分段仿射，得到模型5：

(ELAARC)

$$\underset{\substack{\mathbf{I},\mathbf{Z},\mathbf{X}^+,\mathbf{X}^-,\\ \mathbf{W},\mathbf{S}^+,\mathbf{S}^-}}{\text{maximize}} \quad \min_{(\zeta^+,\zeta^-)\in\mathscr{D}_2} \Bigg\{ \sum_t \sum_i \sum_j ((\mathbf{X}_{ij}^{t+})^T \boldsymbol{\zeta}^{t+} + (\mathbf{X}_{ij}^{t-})^T \boldsymbol{\zeta}^{t-} + W_{ij}^t) - (\mathbf{c}_0^T \mathbf{Z} + \mathbf{f}^T \mathbf{I}) - \sum_t \sum_j u_j (S_j^+ \zeta_j^{t+} + S_j^- \zeta_j^{t-}) \Bigg\} \tag{11a}$$

subject to

$$\sum_i ((\mathbf{X}_{ij}^{t+})^T \boldsymbol{\zeta}^{t+} + (\mathbf{X}_{ij}^{t-})^T \boldsymbol{\zeta}^{t-} + W_{ij}^t) \le \zeta_j^t + S_j^{t+}\zeta_j^{t+} + S_j^{t-}\zeta_j^{t-}, \forall (\boldsymbol{\zeta}^+,\boldsymbol{\zeta}^-) \in \mathscr{D}_2, \forall j, \forall t, \tag{11b}$$

$$S_j^{t+}\zeta_j^{t+} + S_j^{t-}\zeta_j^{t-} \ge 0, \forall (\boldsymbol{\zeta}^+,\boldsymbol{\zeta}^-) \in \mathscr{D}_2, \forall j, \forall t, \tag{11c}$$

$$(9c)\text{–}(9e), \tag{11d}$$

上述两个模型还可以稍作简化，这里暂时不写了。

最后一类模型称为历史数据驱动型。也就是令

$$\gamma_{ij}^t = \sum_{t'=1}^{t} \left(\mathbf{X}_{ij}^{tt'}\right)^T \zeta^{t'} + W_{ij}^t$$

变成受历史数据影响的仿射函数。带入简化后的模型5，得到模型6：

(HD-ELAARC)

$$\underset{\mathbf{I},\mathbf{Z},\mathbf{X}^-,\mathbf{W},\mathbf{S}^-}{\text{maximize}} \quad \min_{\zeta^-\in\mathscr{D}_3} \Bigg\{ \sum_t \sum_i \sum_j (\eta - d_{ij} - c_i) \cdot \left(\sum_{t'=1}^{t} (\mathbf{X}_{ij}^{tt'-})^T \boldsymbol{\zeta}^{t'-} + W_{ij}^t \right) - (\mathbf{c}_0^T \mathbf{Z} + \mathbf{f}^T \mathbf{I}) - \sum_t \sum_j u_j \left(\sum_{t'=1}^{t} S_j^{tt'-} \zeta_j^{t'-} \right) \Bigg\} \tag{13a}$$

$$\text{subject to} \quad \sum_i \left(\sum_{t'=1}^{t} (\mathbf{X}_{ij}^{tt'-})^T \boldsymbol{\zeta}^{t'-} + W_{ij}^t \right) \le \bar{\zeta}_j^t - \zeta_j^{t-} + \sum_{t'=1}^{t} S_j^{tt'-} \zeta_j^{t'-}, \quad \forall \boldsymbol{\zeta}^- \in \mathscr{D}_3, \forall j, \forall t, \tag{13b}$$

$$\sum_j \left(\sum_{t'=1}^{t} (\mathbf{X}_{ij}^{tt'-})^T \boldsymbol{\zeta}^{t'-} + W_{ij}^t \right) \le Z_i, \quad \forall \boldsymbol{\zeta}^- \in \mathscr{D}_3, \forall i, \forall t, \tag{13c}$$

$$\sum_{t'=1}^{t} (\mathbf{X}_{ij}^{tt'-})^T \boldsymbol{\zeta}^{t'-} + W_{ij}^t \ge 0, \quad \forall \boldsymbol{\zeta}^- \in \mathscr{D}_3, \forall i, \forall j, \forall t, \tag{13d}$$

$$\sum_{t'=1}^{t} S_j^{tt'-} \zeta_j^{t'-} \ge 0, \quad \forall \boldsymbol{\zeta}^- \in \mathscr{D}_3, \forall j, \forall t, \tag{13e}$$

$$\mathbf{Z} \le M\mathbf{I}, \quad \mathbf{I} \in \{0,1\}^L, \tag{13f}$$

提出这么多模型的目的都是为了减小FVB模型的保守性，也就是让目标函数更大。[3]从目标函数和最优解目标值证明了这些模型之间的关系。总的来说，后面提出的模型效果更好。

求解算法使用行生成，也就是benders。不是很感兴趣，暂时也不写啦。

Reference

[1] Chun Cheng, Yossiri Adulyasak, Louis-Martin Rousseau.Robust Facility Location Under Disruptions. INFORMS Journal on Optimization.

[2] Chun Cheng, Yossiri Adulyasak, Louis-Martin Rousseau. Robust facility location under demand uncertainty and facility disruptions. Omega.

[3] Amir Ardestani-Jaafari, Erick Delage. The Value of Flexibility in Robust Location–Transportation Problems. Transportation Science.

2. Vehicle Routing-1

VRP部分有很多很有意思的RO相关研究，要写完估计很难。我打算暂时写一小块内容，等以后有机会补充一些。不熟悉VRP的同学请自行查阅资料。

一般来说，VRP会有两方面不确定性，客户需求不确定与旅行时间不确定。对于VRPTW，或是带deadline的VRPD来说，讨论旅行时间不确定会很麻烦，一般的论文会讨论软时间窗的情况。但是，[1]和其前作，以及章宇的两篇文章，用了一些有趣的指标来反应未到达的情况。这些指标理解起来有点复杂，不是简单的LP，但却很有意思。

Robust在VRP中有三个层次，robust on customer，route和network。[2]中给了一些例子，建议大家看一看。

在众多关于RVRP的文章里，首先想介绍的是一篇对比了SVRP和DVRP的文章。这篇文章首先给出了基于多商品流的VRP模型，这是一个在确定性问题中很少见、但经常出现在robust问题中的模型。其次使用了一种基于经济学中CE（确定等值，Certainty equivalent）的度量违背deadline的情况。最后，在stochastic问题和分布式鲁棒问题中，用到了很多非线性的约束，最终转化成branch-and-cut可求解的问题，非常有趣且有挑战性。虽然这篇文章难度很大，但还是很有必要攻克下来的（以后和Yossiri聊就能接上话了）。

基于多商品流的VRP模型

文章的一个缺点是用了很多花体字母，看的很烦。手写的时候我就不用花体了。首先需要引入一些变量：

$G=(N,A)$ 的有向网络，$N=\{1,\cdots,n\}$ 为客户集，1和 n 为depot.

$N_R \subseteq N$ 为必须服务的客户，$N_D \subseteq N$ 为有 deadline 的客户.

对点集 H，$\delta^-(H)$ 为进入 H 的边集，$\delta^+(H)$ 为离开．$(\delta^-(H)=\{(i,j)\in A \mid i\in N\setminus H, j\in H\})$

$K=\{1,\cdots,m\}$ 为车辆集.

决策变量：$s_a^i=1$，若边 a 是某条从1到 i 的路径中的边.

$x_a=1$，若某辆车经过边 a.

z_i，点 i 被访问的次数.

基于多商品流的模型需要以下两组约束：

路径集约束：

$$
\varphi^{VRP}=\left\{ s \;\middle|\; \begin{array}{l}
\sum_{a\in\delta^+(1)} s_a^i = z_i, \quad \forall i\in N_D \\
\sum_{a\in\delta^-(u)} s_a^i - \sum_{a\in\delta^+(u)} s_a^i = 0, \quad \forall i\in N_D,\ u\in N\setminus\{1,i,n\} \\
\sum_{a\in\delta^-(i)} s_a^i - \sum_{a\in\delta^+(i)} s_a^i = z_i, \quad \forall i\in N_D\cup\{n\}, \\
0\le s_a^i \le x_a, \quad \forall i\in N_D,\ \forall a\in A \\
(x,z)\in H^{VRP}
\end{array}\right\}
$$

控制多车辆的约束：

$$
H^{VRP}=\left\{ \begin{array}{l} x\in\{0,1\}^{|A|}, \\ z\in Z_+^{|N|} \end{array} \;\middle|\; \begin{array}{l}
1\le z_i\le m, \quad \forall i\in\{1,n\}. \\
z_i=1, \quad \forall i\in N_R\setminus\{1,n\}. \\
z_i\le 1, \quad \forall i\in N\setminus N_R. \\
\sum_{a\in\delta^+(i)} x_a = z_i, \quad i\in N\setminus\{n\}. \\
\sum_{a\in\delta^-(i)} x_a = z_i, \quad i\in N\setminus\{1\}. \\
\sum_{a\in\delta^+(H)} x_a \ge \sigma(H)\, z_j, \\
\quad \forall H\subseteq N\setminus\{1,n\}: |H|\ge 2,\ \forall j\in H
\end{array}\right\}
$$

习惯MTZ模型的人可能不太习惯看多商品流模型。先看第一个集合，第一个约束表示从起点出发的点i的约束，第二、三个约束表示流平衡，可以分别按u在i的前、中、后三种情况理解。最后一个约束也好理解。再看第二个集合，第一个约束表示起点的z，第二个表示必须经过的点，第三个约束表示不必须经过的点，后两个约束表示x_a与z_i的关系。最后一个约束为消（子）环约束。其中$\sigma(H)$表示经过集合H的最小车辆数量。

对单路径、无容量约束的模型，直接令$\sigma = 1$即可。对多路径、容量约束的模型，令$\sigma = \lceil \sum_{i\in H} d_i / Q \rceil$。如果想简单地将模型修改为多路情况，需要对每个变量增加下标k，表示k辆车。但这会大大增加模型的复杂度，变成3-index。为了维持2-index的情况，可以使用一种复制节点的手段，将终点复制成K个，再添加一个超级节点，所有终点连接到超级节点，长度为0，用这个点替换原先的终点。在两条路的情况下，变换如下：

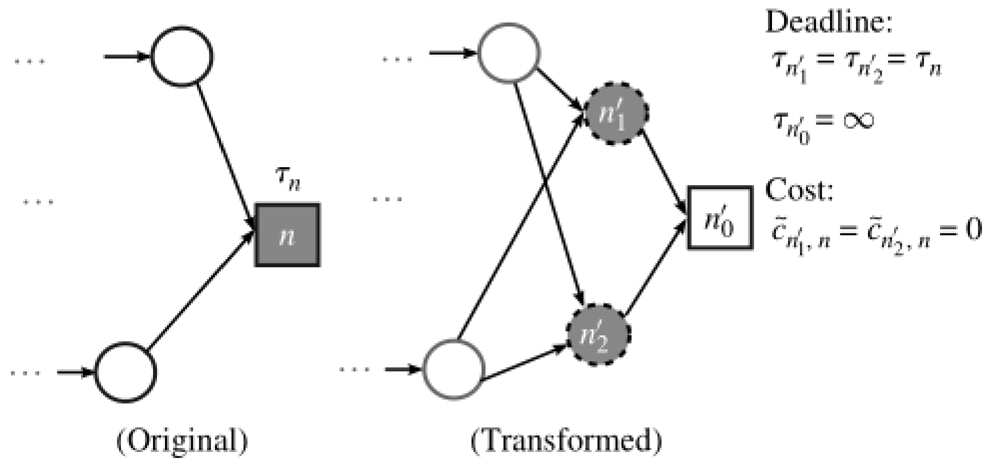

Figure 2 Network Transformation for the Problem with Multiple Capacitated Vehicles

灰色代表depot，方块代表问题的终点。这样转换之后，不同的车辆可以用不同的depot区分开。

由于每个depot的复制都是相同的，对任何一个解，可以将俩条路的前面部分和depot的两个复制节点交换，然后得到一个对称的新解。这也会增大搜索空间。因此可以加入破坏对称性的约束来加速。约束如下：

$$\sum_{a\in\mathcal{A}} \mu_a s_a^{n'_k} \geq \sum_{a\in\mathcal{A}} \mu_a s_a^{n'_{k+1}} \quad \forall 1 \leq k \leq |\mathcal{K}| - 1.$$

其中参数μ_a可以是边a的旅行时间的上下限或均值（因为是随机变量）。这样时间长的路径就会自动连接到k较小的depot的复制上。

随机规划模型

首先引入一些参数：

$\tilde{c}_a$：边 a 的旅行时间，为随机变量.

$\tilde{t}_i$：节点 i 的到达时间. $\tilde{t}_i = \sum_{a \in A} \tilde{c}_a \cdot s_a^i$

τ_i：节点 i 的deadline.

我们的目标是最小化违背每个deadline的概率之和：

$$
\begin{aligned}
\min \quad & \sum_{i \in N_D} P(\tilde{t}_i > \tau_i) \\
\text{s.t.} \quad & \tilde{t}_i = \sum_{a \in A} \tilde{c}_a \cdot s_a^i, \quad \forall i \in N_D. \\
& (s) \in \varphi^{VRP}.
\end{aligned}
$$

对给定的 $\bar{s}$，记 $A_{\bar{s}}^i = \{a \in A \mid \bar{s}_a^i = 1\}$，

$$\beta_{\bar{s}}^t = P(t_{\bar{s}}^i > \tau_i)$$

$$
\begin{aligned}
\min \quad & \sum_{i \in N_D} \rho_i \\
\text{s.t.} \quad & \sum_{a \in A_p^i} \beta_p^i (s_a^i - 1) + \beta_p^i \leq \rho_i, \quad \forall i \in N_D, \ \forall (p) \in \varphi^{VRP}. \\
& \rho_i \geq 0, \quad \forall i \in N_D. \\
& (s) \in \varphi^{VRP}.
\end{aligned}
$$

上面给出了两个模型，其实是等价的。第一个模型很简单，基于前文的多商品流模型得到解s，然后计算出t，进而得到目标函数。但是目标函数不是线性形式，因此需要转化成下面的模型。下面的模型可以用branch-and-cut的形式解决。

约束1实际上表达了这样的含义：如果解p被s覆盖，则左侧部分为β_p^i，否则左侧为一个负值，约束失效。因此，对每一个ϕ的解p都满足约束1时，这个约束与上面模型的约束1等价。可以通过不断添加p的分解算法得到下面模型的最优解。

在随机规划中，我们将每个不确定变量视作已知分布的随机变量。因此在一直解p的时候就能计算出β_p^t里面的的分布函数，进而计算出概率值。问题变成一个线性问题。

此外，SP还有一个常用解法，Sample Average Approximation。简单的说，就是通过蒙特卡洛模拟采样随机变量的不同实现值，来代替随机变量的密度、分布函数。使用SAA时模型转变为：

$\omega \in \Omega$：$\tilde{c}$ 的可能取值的场景集合.

$\phi_{i\omega} = 1$，表示 ω 场景下节点 i 的到达时间超过deadline.

$M_{i\omega}$：大M.

$$\min \frac{1}{|\Omega|} \sum_{i \in N_D} \sum_{\omega \in \Omega} \phi_{i\omega}$$

$$\text{s.t.} \quad \sum_{a \in A} c_{a\omega} s_a^i \le \tau_i + M_{i\omega} \phi_{i\omega}, \quad \forall i \in N_D, \forall \omega \in \Omega.$$

$$\phi_{i\omega} \in \{0, 1\}, \quad \forall i \in N_D, \omega \in \Omega.$$

$$(s) \in \mathcal{Y}^{VRP}.$$

$$\beta_j^i = \frac{1}{|\Omega|} \cdot \sum_{\omega \in \Omega} \phi_{i\omega}$$

SAA需要较大数量的样本，因此会导致模型变大，很难求解。

鲁棒优化模型

考虑分布式鲁棒优化，模糊集为一阶矩约束：

$$F=\left\{\mathbb{P} \mid E_{\mathbb{P}}(\tilde{c})=\mu,\ \mathbb{P}(\tilde{c}\in[\underline{c},\bar{c}])=1\right\}. \quad \text{—(1)}$$

$$C_{\alpha_i}(\tilde{t}_i)=\alpha_i \ln E_{\mathbb{P}}\left(\exp\left(\frac{\tilde{t}_i}{\alpha_i}\right)\right),\ \alpha_i\geq 0. \quad \text{—(2)}$$

$$C_{\alpha_i,F}(\tilde{t}_i)=\sup_{\mathbb{P}\in F}\alpha_i \ln E_{\mathbb{P}}\left(\exp\left(\frac{\tilde{t}_i}{\alpha_i}\right)\right) \quad \text{—(3)}$$

$$\inf\left\{\alpha_i \mid C_{\alpha_i,F}(\tilde{t}_i)\leq \tau_i,\ \alpha_i\geq 0\right\}. \quad \text{—(4)}$$

$$\rho_\tau(\tilde{t}_i)=\inf\left\{\sum_{i\in N_D}\alpha_i \mid C_{\alpha_i,F}(\tilde{t}_i)\leq \tau_i,\ \alpha_i\geq 0,\ \forall i\in N_D\right\} \quad \text{—(5)}$$

（1）为模糊集。模糊集给定了期望和上下界。

（2）描述了一个经济学概念，certainty equivalent。表示随机变量$\tilde{t}_i$的certainty equivalent。其中，α_i可以理解为一个违背deadline的风险容忍系数（risk tolerance parameter）。针对这个certainty equivalent，worst-case被描述为（3）。目标函数是最小化α，因此写作（4）。对于整个系统而言，定义目标为最小化α_i之和，写作（5）。采用这个概念为目标函数还有其他优势和性质，在这里暂时不介绍，在另一篇文章里再写。

得到模型：

$$\begin{aligned}
\inf \quad & \sum_{i\in N_D}\alpha_i \\
\text{s.t.} \quad & h(\alpha_i, s^i)\leq \tau_i,\ \forall i\in N_D. \qquad (\lambda_i) \\
& \alpha_i\geq 0,\ \forall i\in N_D. \\
& (s)\in \varphi^{VRP}
\end{aligned}$$

$$h(\alpha_i, s^i)=\alpha_i \ln E_{\mathbb{P}}\left(\exp\left(\frac{\tilde{c}^T\cdot s^i}{\alpha_i}\right)\right)$$

第一个约束显然不是线性的。利用一种拉格朗日函数的次梯度方法求解。首先写出对偶问题（1）：

$$\mathcal{L}(s,\alpha,\lambda) = \sum_{i\in N_D} \alpha_i - \sum_{i\in N_D} \lambda_i \cdot \left[\alpha_i \ln E_{\mathbb{P}}\left(\exp\left(\frac{\tilde{c}^T s^i}{\alpha_i}\right)\right) - \tau_i\right] \quad \text{—(1)}$$

$$\sum_{a\in A} \sup_{\mathbb{P}\in F} E_{\mathbb{P}}(\tilde{c}_a) s_a^i \le \tau_i \quad \text{—(2)}$$

$$f^r(s) = \inf_{\alpha\ge 0} \sup_{\lambda\ge 0} \mathcal{L}(s,\alpha,\lambda) = \sup_{\lambda\ge 0} \inf_{\alpha\ge 0} \mathcal{L}(s,\alpha,\lambda). \quad \text{—(3)}$$

为了满足强对偶定理，需要满足条件（2）Slater条件。由于这个约束符合模型含义，因此直接加入模型。在强对偶下，满足（3）。

另一篇论文的结论有如下关系：

$$\begin{aligned} f^r(s) - f^r(p) &= \sup_{\lambda\ge 0} \inf_{\alpha\ge 0} \mathcal{L}(s,\alpha,\lambda) - \sup_{\lambda\ge 0} \inf_{\alpha\ge 0} \mathcal{L}(p,\alpha,\lambda) \\ &\ge \inf_{\alpha\ge 0} \left[\mathcal{L}(s,\alpha,\lambda) - \mathcal{L}(p,\alpha,\lambda)\right] \\ &\ge d_p^L(p,\alpha^*,\lambda^*)(s-p), \end{aligned}$$

$$\text{where } (\alpha^*,\lambda^*) = \left\{(\bar{\alpha},\bar{\lambda}) \,\middle|\, \mathcal{L}(s,\bar{\alpha},\bar{\lambda}) = \sup_{\lambda\ge 0} \inf_{\alpha\ge 0} \mathcal{L}(s,\alpha,\lambda)\right\}$$

$$d_p^L(p,\alpha^*,\lambda^*) = d_p^f(p) \text{ 是 } \mathcal{L}(p,\alpha^*,\lambda^*) \text{ 的次梯度.}$$

$$f^r(s) \ge f^r(p) + d_p^L(p,\alpha^*,\lambda^*)(s-p), \quad \forall (p) \in \varphi^{VRP}$$

因此可转化为最后的约束。问题因此可以写成：

$$
\begin{aligned}
\inf \quad & w \\
\text{s.t.} \quad & f^r(p) + d_p^f(p)(s-p) \le w, \quad \forall (p) \in \varphi^{VRP} \\
& \sum_{a \in A} \sup_{P \in F} E_P(\tilde{c}_a)\, s_a^i \le \tau_i, \quad \forall i \in N_0 \\
& \alpha_i \ge 0, \quad \forall i \in N_0 \\
& (s) \in \varphi^{VRP}.
\end{aligned}
$$

这个问题又可以用分解算法逐步添加约束解决。

基础的建模部分大概到这里。[1]还补充了service time、软时间窗等情况下的建模方法。算法采用Branch-and-cut，总体来说和前一章的方法类似，都是基于decomposition的算法。

这篇文章在VRP体系下还是挺有了解的必要的，尤其是结合SP的背景一起。后面章宇的文章的目标函数也是改进自这里。其实圈子都挺小的，每篇重要的文章作者里都有一些大佬。

由于个人时间安排，后续更新可能会搁置一会儿。请大家见谅。

Reference

[1] Yossiri Adulyasak, Patrick Jaillet. Models and Algorithms for Stochastic and Robust Vehicle Routing with Deadlines. Transportation Science.

[2] 章宇。鲁棒优化及其在车辆路径问题中的应用简介。https://www.bilibili.com/video/BV1tK4y1e7fQ?spm_id_from=333.337.search-card.all.click

2. Vehicle Routing-2

Branch and price for RVRP

VRP with DRO

Reference

[1] Pedro Munari, Alfredo Moreno, Jonathan De La Vega, Douglas Alem, Jacek Gondzio, Reinaldo Morabito. The Robust Vehicle Routing Problem with Time Windows: Compact Formulation and Branch-Price-and-Cut Method. Transportation Science.

[2] Yossiri Adulyasak, Patrick Jaillet. Models and Algorithms for Stochastic and Robust Vehicle Routing with Deadlines. Transportation Science.

[3] Yu Zhang, Roberto Baldacci, Melvyn Sim, Jiafu Tang. Routing optimization with time windows under uncertainty. Mathmetical Programming.

[4] Yu Zhang, Zhenzhen Zhang, Andrew Lim, Melvyn Sim. Robust Data-Driven Vehicle Routing with Time Windows. Operations Research.

3. ADRO in some advanced problems

Drone delivery

Reference

[1] Chun Cheng, Yossiri Adulyasak, Louis-Martin Rousseau, Melvyn Sim. Robust Drone Delivery with Weather Information. Optimization Online.