

## 第5章 多阶段问题与线性决策规则

### (Multi-stage problem and linear decision rule)

陈植, 汤勤深

第三章和第四章分别介绍了经典鲁棒优化和分布鲁棒优化的方法和模型。然而这些方法和模型主要针对单阶段 (single stage) 的问题。而在企业或者决策者面对的问题中, 很大一部分都是多阶段的 (Multi-stage)。也即, 决策者需要在一定的时间内, 按时间顺序进行多个决策, 且决策时只知过去已发生的随机变量 (random variable) 而无法预知未来的。多阶段问题的这些特性, 使得对相应问题的建模和求解特别复杂。在优化领域, 一般是用随机规划或者动态规划进行建模和求解。然而, 随机规划和动态规划都遭受“维度诅咒” (curse of dimensionality)。

针对随机规划和动态规划的这个致命缺点, 鲁棒优化用一些决策规则 (decision rule) 进行规避。现有的决策规则已经可以达到很好的近似效果, 甚至在某些条件下有一些决策规则可以达到最优。

在这一章, 我们将聚焦于如何用这些决策规则对多阶段问题 (主要是两阶段问题) 进行求解。在此之前, 我们先着重介绍两阶段的随机规划问题。

**备注:** 正如后面会讨论, 决策规则近来越来越多地被改称为近似规则 (recourse approximation)。如若称为决策规则, 则有后面阶段的决策将根据规则直接得到之嫌。而实际上, 如果直接使用决策规则对后面阶段决策进行决策, 其效果非常不可控。大多时候, 模型会表现非常差。在模型的实施过程中, 往往都是用滚动法 (rolling horizon), 也即, 在知道这一期的不确定性之后, 将现在系统的状态当成初始状态, 重新对模型进行求解, 以获得下一期的最优决策。从这个角度来说, 我们使用一定的规则去近似未来的决策和不确定性之间的关系, 从而达到简化模型的效果。

### 5.1 随机规划 (Stochastic programming)

为了更好地理解随机规划, 我们举例如下。

**示例 5.1 (单阶段库存管理模型):** 小王前不久在小区开了一个小卖部。他发现, 小区对苹果的需求  $\tilde{d}$  满足分布  $F(\cdot)$ 。他每天需要决定向 20 里外小张定  $x$  斤苹果。每斤苹果的进货单价为  $c$ , 销售价格为  $p$ 。若库存不足, 也即真实需求  $d$  大于订货量  $x$ , 居民可以先下单, 等有货了再送过去。针对这种情况, 小王一般会给一定的折扣。折算下

来，每一斤苹果将增加  $b$  的成本。而如果订货量太多，也即  $x - d \geq 0$ ，苹果可能会变质，折算下来一斤苹果将增加  $h$  的成本。小王的利润为：

$$\pi(x, d) = p \min\{x, d\} - cx - b(d - x)^+ - h(x - d)^+.$$

小王想最大化他的期望利润，也即他将要解以下问题：

$$\begin{aligned} \max \quad & \mathbb{E}_{\mathbb{P}} [\pi(x, \tilde{d})] \\ \text{s.t.} \quad & x \geq 0 \end{aligned} \tag{5.1}$$

示例5.1就是一个在商业环境中随处可见的随机规划问题：在已知随机分布和满足一定的约束的情况下，进行决策使得期望利润最大化。一般地，随机规划关注以下问题：

$$\begin{aligned} \min \quad & \mathbb{E}_{\mathbb{P}} [g(\mathbf{x}, \tilde{\xi})] \\ \text{s.t.} \quad & \mathbb{E}_{\mathbb{P}} [f_i(\mathbf{x}, \tilde{\xi})] \leq 0 \quad \forall i \in [I]. \end{aligned}$$

其中， $\tilde{\xi}$  为随机变量。

示例5.1是一个单阶段的问题。但是，如果我们把卖多少苹果， $\min\{x, d\}$ ，当成决策  $y$ ，那示例5.1就变成了一个两阶段的问题：第一阶段，决定定多少苹果，之后观测到需求；第二阶段，决定卖多少苹果。也即，模型(5.1)可以写成

$$\begin{aligned} \max \quad & -cx - b\mathbb{E}_{\mathbb{P}} [\tilde{d}] - hx + \mathbb{E}_{\mathbb{P}} [g(x, \tilde{d})] \\ \text{s.t.} \quad & x \geq 0 \end{aligned}$$

其中，

$$\begin{aligned} g(x, d) = \max \quad & (p + b + h)y \\ \text{s.t.} \quad & y \leq x, \\ & y \leq d. \end{aligned}$$

此模型可归类于由Dantzig (1955) 首先引入的经典的（线性）两阶段随机规划问题（two stage stochastic programming）：第一阶段的决策变量为  $\mathbf{x} \in \mathbb{R}^{N_1}$ ，也称为“现时决策（here-and-now decision）”。不失一般性，假设  $\mathbf{x}$  的可行集为  $X = \{\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ 。与  $\mathbf{x}$  相关的成本参数  $\mathbf{c} \in \mathbb{R}^{N_1}$ 。之后，随机变量  $\tilde{\xi} \in \mathcal{W} \in \mathbb{R}^{I_1}$  实现为  $\xi$ 。其中  $\mathcal{W}$  为  $\tilde{\xi}$  的支撑集。第二阶段决策变量为  $\mathbf{y}$ ，也称“等待决策（wait-and-see decision）”。其相应的成本参数为  $\mathbf{q} \in \mathbb{R}^{N_2}$ 。此两阶段随机规划问题模型如下：

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{x} + \mathbb{E}_{\mathbb{P}} [g(\mathbf{x}, \tilde{\xi})], \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{5.2}$$



其中

$$\begin{aligned} g(\mathbf{x}, \boldsymbol{\xi}) &= \min \mathbf{q}^\top \mathbf{y}, \\ \text{s.t. } \mathbf{T}(\boldsymbol{\xi})\mathbf{x} + \mathbf{W}\mathbf{y} &= \mathbf{h}(\boldsymbol{\xi}), \\ \mathbf{y} &\geq \mathbf{0}. \end{aligned} \quad (5.3)$$

模型(5.3)中,  $\mathbf{T} \in \mathcal{R}^{I_1, M \times N_1}$ ,  $\mathbf{h} \in \mathcal{R}^{I_1, M}$  是  $\boldsymbol{\xi} \in \mathcal{W}$  的函数。在接下来的讨论中, 我们假设他们仿射依赖于  $\boldsymbol{\xi} \in \mathbb{R}^{I_1}$ :

$$\mathbf{T}(\boldsymbol{\xi}) = \mathbf{T}^0 + \sum_{i \in [I_1]} \mathbf{T}^i \xi_i, \quad \mathbf{h}(\boldsymbol{\xi}) = \mathbf{h}^0 + \sum_{i \in [I_1]} \mathbf{h}^i \xi_i,$$

其中,  $\mathbf{T}^0, \dots, \mathbf{T}^{I_1} \in \mathbb{R}^{M \times N_1}$ ,  $\mathbf{h}^0, \dots, \mathbf{h}^{I_1} \in \mathbb{R}^M$ 。

另外, 矩阵  $\mathbf{W}$  称为补偿矩阵 (recourse matrix)。第二阶段的模型(5.3)不一定总是有可行解。但是如果  $\mathbf{W}$  是完全补偿的 (complete recourse) --- 对任意的  $\mathbf{z} \in \mathbb{R}^M$ , 存在  $\mathbf{y} \in \mathbb{R}^{N_2}$  使得  $\mathbf{W}\mathbf{y} \geq \mathbf{z}$  --- 那么可以保证对于所有  $\mathbf{x} \in \mathbb{R}^{N_1}$  和  $\boldsymbol{\xi} \in \mathbb{R}^{I_1}$ , 模型(5.3)都有可行解。然而完全补偿的假设过于苛刻, 有些问题不一定具有这个性质。通常情况下, 我们会假设模型(5.3)对于所有的  $\mathbf{x} \in \mathcal{X}$  和  $\boldsymbol{\xi} \in \mathcal{W}$  都有可行解, 也即模型(5.3)具有相对完全补偿 (relatively complete recourse)。

然而, 两阶段随机模型具有以下难点:

1. 大量变量和约束。
2. 难以获得  $\tilde{\boldsymbol{\xi}}$  的集中分布。
3. 难以评估 (evaluate) 目标函数。尤其当  $\tilde{\boldsymbol{\xi}}$  的维度较大时。
4. 难以获得一个第一阶段的可行解  $\mathbf{x}$  能够保证第二阶段的解  $\mathbf{y}$  也是可行的。

以上难点, 使得最简单的两阶段随机规划问题是一个  $NP$ -难的问题; 而如果阶段大于 2, 这个问题则是一个  $PSPACE$ -难的问题 (Dyer and Stougie, 2006)。为了部分解决以上的这些问题, 我们接下来介绍动态鲁棒优化和近似决策规则。

## 5.2 动态鲁棒优化 (Dynamic robust optimization)

在两阶段随机规划问题(5.2)中, 假如  $\tilde{\boldsymbol{\xi}}$  的分布  $\mathbb{P}$  的具体分布未知, 但是可以构建某个模糊集  $\mathcal{P}$  使得真实的分布在这个集合中, 那么我们可以得到以下动态鲁棒优化模型:

$$\begin{aligned} \min \mathbf{c}^\top \mathbf{x} + \sup_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}} [g(\mathbf{x}, \tilde{\boldsymbol{\xi}})], \\ \text{s.t. } \mathbf{A}\mathbf{x} &= \mathbf{b}, \\ \mathbf{x} &\geq \mathbf{0}, \end{aligned} \quad (5.4)$$

其中

$$\begin{aligned} g(\mathbf{x}, \boldsymbol{\xi}) &= \min \mathbf{q}^\top \mathbf{y} \\ \text{s.t. } \mathbf{T}(\boldsymbol{\xi})\mathbf{x} + \mathbf{W}\mathbf{y} &= \mathbf{h}(\boldsymbol{\xi}), \\ \mathbf{y} &\geq \mathbf{0}. \end{aligned}$$



我们可以等价地把模型(5.4)写成：

$$\begin{aligned} Z^* &= \min \mathbf{c}^\top \mathbf{x} + \sup_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}} [\mathbf{q}^\top \mathbf{y}(\tilde{\xi})], \\ \text{s.t. } &\mathbf{Ax} = \mathbf{b}, \\ &\mathbf{T}(\xi)\mathbf{x} + \mathbf{Wy}(\xi) = \mathbf{h}(\xi) \quad \forall \xi \in \mathcal{W}, \\ &\mathbf{y} \in \mathcal{R}^{I_1, N_2}, \\ &\mathbf{x} \geq \mathbf{0}. \end{aligned} \quad (5.5)$$

然而，模型(5.5)一般来说是不可解的，因为  $\mathbf{y}$  是  $\xi$  的任意一个函数。如果我们假设  $\mathbf{y}$  和  $\xi$  之间的映射是可知的，比如是仿射或者二次的，那么模型(5.5)是否可能有解呢？答案是肯定的。

**备注：**模型(5.2)是一个分布鲁棒优化的两阶段模型。由第三章和第四章可知，当模糊集只包含随机变量的支撑集时，分布鲁棒优化模型退化传统的鲁棒优化模型。因此，这一章节只讨论分布鲁棒优化下的多阶段问题和线性决策规则。

### 5.3 线性决策规则 (Linear decision rule)

线性决策规则 (LDR) 是一种在动态优化模型中，假设当前阶段的决策线性依赖于 (之前阶段) 随机变量的决策机制。也即，这是对决策变量和随机变量之间复杂关系的一种近似。相对应的，也有非线性的决策规则，比如二次决策规则 (Quadratic Decision Rule, Ben-Tal et al. 2009) 和多项式决策规则 (Polynomial Decision Rules, Bertsimas et al. 2011)。

决策规则的提出旨在降低随机规划问题中的维度。有关于早期决策规则和随机规划结合的文献可参考 Garstka and Wets 等人 1974 年写的综述。然而，由于此近似所得模型过于保守而被弃用。之后，Ben-Tal et al. (2004) 创造性地将 LDR 和鲁棒优化相结合，使得线性决策规则焕发出勃勃生机。具体地，对于模型(5.5)我们可以假设  $\mathbf{y}$  是  $\xi$  的仿射函数，

$$\mathbf{y}(\xi) = \mathbf{y}^0 + \sum_{i \in [I_1]} \mathbf{y}_i^1 \xi_i.$$

不失一般性，我们定义以下集合

$$\mathcal{L}^{I,N} = \left\{ \mathbf{y} \in \mathcal{R}^{I,N} \mid \begin{array}{l} \exists \mathbf{y}^0, \mathbf{y}_i^1, i \in [I_1] : \\ \mathbf{y}(\xi) = \mathbf{y}^0 + \sum_{i \in [I_1]} \mathbf{y}_i^1 \xi_i \end{array} \right\}.$$



在线性规则下, 模型(5.5)可以写成

$$\begin{aligned}
 Z^L &= \min \mathbf{c}^\top \mathbf{x} + \sup_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}} [\mathbf{q}^\top \mathbf{y}(\boldsymbol{\xi})], \\
 \text{s.t. } &\mathbf{A}\mathbf{x} = \mathbf{b}, \\
 &\mathbf{T}(\boldsymbol{\xi})\mathbf{x} + \mathbf{W}\mathbf{y}(\boldsymbol{\xi}) = \mathbf{h}(\boldsymbol{\xi}) \quad \forall \boldsymbol{\xi} \in \mathcal{W}, \\
 &\mathbf{y} \in \mathcal{L}^{I_1, N_2}, \mathbf{x} \geq \mathbf{0}.
 \end{aligned} \tag{5.6}$$

由此, 我们可以得到模型(5.5)的一个上界。

#### 定理 5.1

$$Z^* \leq Z^L.$$

既然是上界, 那么很自然的问题是, 这个近似的效果如何? 近似之后的问题和原问题的差距多大? 什么条件下这两个问题是一样的, 也即, 什么条件可以保证线性决策规则是最优的? 对于前两个问题, 据笔者所知, 暂时没有一般性的结论。而对于第三个问题, Iancu et al. (2013) 给出了模糊集中只包含随机变量的支撑集时, 线性决策规则最优的条件和技术性假设。Bertsimas et al. (2010) 和 Bertsimas and Goyal (2012) 探讨了某几种特殊的多阶段问题中 LDR 最优的条件。而对于下一小节要介绍的拓展式线性决策规则 (ELDR), Bertsimas et al. (2019) 证明了当第二阶段的决策变量为一维时, ELDR 为最优。He et al. (2020) 证明了 ELDR 对于车辆调度问题 (vehicle repositioning problem) 在满足一定的技术性假设条件下, 对于任意维度的补偿决策都是最优的。对于第五小节要介绍的情景仿射补偿近似规则, Perakis et al. (2020) 证明了在三阶段的定价和库存模型中, 当只考虑一个产品时, 事件式近似法则是最优的。而对于多个产品的情况, 从数值例子来看, 也接近于最优。

LDR 的运用使得多阶段的鲁棒优化问题受到了越来越多的学者的关注。然而, 线性决策规则有一个很明显的缺点, 近似模型太保守或者容易使得模型不可解。比如, Chen et al. (2008) 指出, 当  $\boldsymbol{\xi}$  的支撑集  $\mathcal{W} = (-\infty, +\infty)$  时,

$$\mathbf{y}(\boldsymbol{\xi}) = \mathbf{y}^0 + \sum_{i \in [I_1]} \mathbf{y}_i^1 \xi_i \geq \mathbf{0},$$

可以得到  $\mathbf{y}_i^1 = \mathbf{0}, \forall i \in [I_1]$ 。此时,  $\mathbf{y}(\boldsymbol{\xi}) = \mathbf{y}^0$  是静态的 (Static policy), 而不是动态地依赖于  $\boldsymbol{\xi}$ 。这就很容易导致所得的解过于保守或者模型不可解。比如考虑如下的随机优化问题:

$$\begin{aligned}
 \min \mathbb{E}_{\mathbb{P}} [y_1(\boldsymbol{\xi}) + y_2(\boldsymbol{\xi})] \\
 \text{s.t. } &y_1(\boldsymbol{\xi}) - y_2(\boldsymbol{\xi}) = h(\boldsymbol{\xi}), \\
 &y_1(\boldsymbol{\xi}) \geq 0, y_2(\boldsymbol{\xi}) \geq 0.
 \end{aligned}$$

如果  $\boldsymbol{\xi}$  的支撑集  $\mathcal{W} = (-\infty, +\infty)$ , 那么  $y_1(\boldsymbol{\xi}) = y_1^0, y_2(\boldsymbol{\xi}) = y_2^0$ 。而此时, 等式  $y_1(\boldsymbol{\xi}) - y_2(\boldsymbol{\xi}) = h(\boldsymbol{\xi})$  将无法被满足。有鉴于此, Chen et al. (2008) 提出了偏转线性



决策规则 (Deflected Linear Decision Rule, DLDR) 和分离线性决策规则 (Segregated Linear Decision Rule, SLDR)。进一步地, See and Sim (2010) 提出了截断线性决策规则 (Truncated linear decision rule), Goh and Sim (2010) 则将 DLDR 和 SLDR 扩展到双偏转线性决策规则 (Bideflected Linear Decision Rule) 和广义分离线性决策规则 (Generalized Segregated Linear Decision Rule)。

对于 LDR 在多阶段问题中的更多的运用, 读者可以阅读 Delage and Iancu (2015) 和 Georghiou et al. (2019)。

下面, 我们举个 LDR 在多阶段鲁棒库存管理中的例子 (See and Sim, 2010; Bertsimas et al., 2019)。

**示例 5.2 (多阶段库存管理模型) :** 假如示例 5.1 中的小王每天都要定苹果。总共要定  $T$  天。所有的成本都是波动的, 也即都跟  $t \in [T]$  相关。假如第  $t$  天早上的库存水平 (inventory level) 为  $y_t$ , 那么第  $t+1$  天的库存水平  $y_{t+1}$  可以通过  $y_t, x_t$  和  $d_t$  得到:

$$y_{t+1} = y_t + x_t - d_t.$$

假设需求是随机变量  $\tilde{z}$  的函数

$$d_t(\tilde{z}_t) = \tilde{z}_t + \alpha \tilde{z}_{t-1} + \cdots + \alpha \tilde{z}_1 + \mu,$$

其中,  $\alpha \in [0, 1]$ ,  $\tilde{z}_t \triangleq (\tilde{z}_1, \dots, \tilde{z}_t)$ 。  $\tilde{z}_t$  的期望为零且  $\tilde{z}_t$  和  $\tilde{z}_s (s \neq t, s, t \in [T])$  两两互不相关 (uncorrelated)。那么, 我们可以把小王这  $T$  天的问题写成如下鲁棒优化模型:

$$\begin{aligned} \min \quad & \sup_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}} \left[ \sum_{t \in [T]} c_t x_t(\tilde{z}_{t-1}) + v_t(\tilde{z}_t) \right] \\ \text{s.t.} \quad & y_{t+1}(\mathbf{z}_t) = y_t(\mathbf{z}_{t-1}) + x_t(\mathbf{z}_{t-1}) - d_t(\mathbf{z}_t) \quad \forall \mathbf{z} \in \mathcal{W}, t \in [T], \\ & v_t(\mathbf{z}_t) \geq h_t y_{t+1}(\mathbf{z}_t) \quad \forall \mathbf{z} \in \mathcal{W}, t \in [T], \\ & v_t(\mathbf{z}_t) \geq -b_t y_{t+1}(\mathbf{z}_t) \quad \forall \mathbf{z} \in \mathcal{W}, t \in [T], \\ & 0 \leq x(\mathbf{z}_{t-1}) \leq \bar{x}_t \quad \forall \mathbf{z} \in \mathcal{W}, t \in [T], \\ & x_t \in \mathcal{L}^{t-1,1}, y_{t+1} \in \mathcal{L}^{t,1}, v_t \in \mathcal{L}^{t,1} \quad \forall t \in [T]. \end{aligned}$$

## 5.4 拓展式线性决策规则 (Extended linear decision rule)

在章节 4.1.1 中, 我们介绍了基于广义矩信息的模糊集。其中, Wiesemann et al. (2014) 巧妙地引入了辅助随机变量  $\tilde{\mathbf{u}}$  使得升维之后的模糊集 (lifted ambiguity set) 中的约束皆化为线性约束, 而将非线性部分转移到了支撑集中。相应地, 在模型 (5.5) 中, 如果假设  $\mathbf{y}$  是  $\boldsymbol{\xi}$  和  $\mathbf{u}$  的仿射函数, 也即

$$\mathcal{L}^{I+J,N} = \left\{ \mathbf{y} \in \mathcal{R}^{I+J,N} \mid \begin{aligned} & \exists \mathbf{y}^0, \mathbf{y}_i^1, \mathbf{y}_j^2 \in \mathbb{R}^N, \forall i \in [I], j \in [J] : \\ & \mathbf{y}(\boldsymbol{\xi}, \mathbf{u}) = \mathbf{y}^0 + \sum_{i \in [I]} \mathbf{y}_i^1 \xi_i + \sum_{j \in [J]} \mathbf{y}_j^2 u_j \end{aligned} \right\}.$$



我们称之为拓展式线性决策规则 (Extended Linear Decision Rule, ELDR)。

在 ELDR 下, 模型(5.5)可以写成

$$\begin{aligned} Z^E &= \min \mathbf{c}^\top \mathbf{x} + \sup_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}} [\mathbf{q}^\top \mathbf{y}(\tilde{\xi}, \tilde{\mathbf{u}})], \\ \text{s.t. } \mathbf{Ax} &= \mathbf{b}, \\ T(\xi)\mathbf{x} + \mathbf{Wy}(\xi, \tilde{\mathbf{u}}) &= \mathbf{h}(\xi) \quad \forall \xi \in \mathcal{W}, \\ \mathbf{y} &\in \mathcal{L}^{I_1+I_2, N_2}, \mathbf{x} \geq \mathbf{0}. \end{aligned} \quad (5.7)$$

由此, 我们可以得到一个比 LDR 更好的近似模型。

### 定理 5.2

$$Z^* \leq Z^E \leq Z^L.$$

详细的证明请参考 [Bertsimas et al. \(2019\)](#).

## 5.5 事件式近似法则 (Event-wise affine recourse approximation)

[Chen et al. \(2019\)](#) 提出了鲁棒随机优化 (Robust Stochastic Optimization, RSO) 模型的统一框架。在这个框架中, 定义静态决策  $\mathbf{w} \in \mathbb{R}^{J_w}$ , 连续随机变量  $\tilde{z}$ , 和离散随机变量  $\tilde{s}$ 。定义只取决于离散随机变量  $\tilde{s}$  的动态决策  $\mathbf{x}(s) : [S] \mapsto \mathbb{R}^{J_x}$ , 以及同时取决于连续随机变量  $\tilde{z}$  和离散随机变量  $\tilde{s}$  的动态决策  $\mathbf{y}(s, \mathbf{z}) : [S] \times \mathbb{R}^{I_z} \mapsto \mathbb{R}^{J_y}$ 。与线性近似法则类似, 对应离散随机变量的不同取值, 动态决策  $\mathbf{y}(s, \mathbf{z})$  为连续随机变量  $\tilde{z}$  的不同的线性函数:

$$\mathbf{y}(s, \mathbf{z}) \triangleq \mathbf{y}^0(s) + \sum_{i \in [I_z]} \mathbf{y}^i(s) z_i.$$

其中, 系数  $\mathbf{y}^0(s), \dots, \mathbf{y}^{I_z}(s)$  是最终模型的实际决策变量。

定义线性映射

$$\begin{cases} \mathbf{a}_m(s, \mathbf{z}) \triangleq \mathbf{a}_{ms}^0 + \sum_{i \in [I_z]} \mathbf{a}_{ms}^i z_i, \\ \mathbf{b}_m(s, \mathbf{z}) \triangleq \mathbf{b}_{ms}^0 + \sum_{i \in [I_z]} \mathbf{b}_{ms}^i z_i, \\ \mathbf{c}_m(s) \triangleq \mathbf{c}_{ms}, \\ d_m(s, \mathbf{z}) \triangleq d_{ms}^0 + \sum_{i \in [I_z]} d_{ms}^i z_i, \end{cases} \quad \forall m \in [M] \cup \{0\}.$$

其中, 参数维度如下

$$\mathbf{a}_{ms}^i \in \mathbb{R}^{J_w}, \mathbf{b}_{ms}^i \in \mathbb{R}^{J_x}, \mathbf{c}_{ms} \in \mathbb{R}^{J_y}, d_{ms}^i \in \mathbb{R} \quad \forall i \in [I_z] \cup \{0\}, s \in [S].$$

RSO 模型的目标函数取分布集合  $\mathcal{F}$  (稍后介绍) 下的最坏期望

$$\sup_{\mathbb{P} \in \mathcal{F}} \mathbb{E}_{\mathbb{P}} [\mathbf{a}'_0(\tilde{s}, \tilde{\mathbf{z}}) \mathbf{w} + \mathbf{b}'_0(\tilde{s}, \tilde{\mathbf{z}}) \mathbf{x}(\tilde{s}) + \mathbf{c}'_0(\tilde{s}) \mathbf{y}(\tilde{s}, \tilde{\mathbf{z}}) + d_0(\tilde{s}, \tilde{\mathbf{z}})].$$



RSO 模型主要包含两类约束。第一类“硬”线性约束 ( $m \in \mathcal{M}_1$ ) 为一般鲁棒约束, 需要在随机变量任意可能的取值下均满足:

$$\mathbf{a}'_m(s, \mathbf{z})\mathbf{w} + \mathbf{b}'_m(s, \mathbf{z})\mathbf{x}(s) + \mathbf{c}'_m(s)\mathbf{y}(s, \mathbf{z}) + d_m(s, \mathbf{z}) \leq 0 \quad \forall \mathbf{z} \in \mathcal{Z}_s, s \in [S].$$

第二类“软”线性约束 ( $m \in \mathcal{M}_2$ ) 与目标函数类似, 考虑分布集合  $\mathcal{F}$  下的最坏期望, 并要求该最坏期望不为正:

$$\sup_{\mathbb{P} \in \mathcal{F}} \mathbb{E}_{\mathbb{P}} [\mathbf{a}'_m(\tilde{s}, \tilde{\mathbf{z}})\mathbf{w} + \mathbf{b}'_m(\tilde{s}, \tilde{\mathbf{z}})\mathbf{x}(\tilde{s}) + \mathbf{c}'_m(\tilde{s})\mathbf{y}(\tilde{s}, \tilde{\mathbf{z}}) + d_m(\tilde{s}, \tilde{\mathbf{z}})] \leq 0 \quad \forall m \in \mathcal{M}_2.$$

除以上两类约束之外, 在离散随机变量的不同取值下, RSO 还包含非线性约束 (如凸约束, 整数约束等)

$$\mathbf{r}(s) \triangleq (\mathbf{w}, \mathbf{x}(s), \mathbf{y}^0(s), \dots, \mathbf{y}^{I_z}(s)) \in \mathcal{X}_s \quad \forall s \in [S],$$

### 5.5.1 事件式近似法则

记离散随机变量  $\tilde{s}$  的取值范围为  $[S]$ 。特别地, 离散随机变量  $\tilde{s}$  每一个取值  $s$  对应一个情景  $s$ 。定义由情景组成的一个非空集合为一个事件  $\mathcal{E} \subseteq [S]$ 。如此, 全部情景的一个划分 (partition) 定义了一个相互独立 (mutually exclusive) 又完全穷尽 (collectively exhaustive) 的 MECE 事件集合, 记为  $\mathcal{C}$ 。相应地, 满足  $\mathcal{H}_{\mathcal{C}}(s) = \mathcal{E}$  函数  $\mathcal{H}_{\mathcal{C}} : [S] \mapsto \mathcal{C}$  确定了情景  $s$  在一个 MECE 事件集合中唯一所属的事件  $\mathcal{E}$ 。

给定一个 MECE 事件集合, 事件式静态近似法则定义如下

$$\mathcal{A}(\mathcal{C}) \triangleq \left\{ x : [S] \mapsto \mathbb{R} \mid \begin{array}{l} x(s) = x^{\mathcal{E}}, \mathcal{E} = \mathcal{H}_{\mathcal{C}}(s) \\ \text{for some } x^{\mathcal{E}} \in \mathbb{R} \end{array} \right\},$$

亦即, 不同事件下, 静态决策不同。

类似地, 事件式线性近似法则定义如下

$$\bar{\mathcal{A}}(\mathcal{C}, \mathcal{I}) \triangleq \left\{ y : [S] \times \mathbb{R}^{I_z} \mapsto \mathbb{R} \mid \begin{array}{l} y(s, \mathbf{z}) = y^0(s) + \sum_{i \in \mathcal{I}} y^i(s) z_i \\ \text{for some } y^0, y^i \in \mathcal{A}(\mathcal{C}), i \in \mathcal{I} \end{array} \right\}$$

其中, 信息集合  $\mathcal{I} \subseteq [I_z]$  为连续随机变量  $\tilde{\mathbf{z}}$  的部分索引 (indices), 声明了连续随机变量  $\tilde{\mathbf{z}}$  中, 事件式线性近似法则所能线性依赖的成分。事件式线性近似法则声明了在不同事件下, 动态决策不同, 并且动态决策为连续随机变量  $\tilde{\mathbf{z}}$  的线性函数。

基于事件式近似法则, 完整的 RSO 模型如下

$$\begin{aligned} \min \quad & \sup_{\mathbb{P} \in \mathcal{F}} \mathbb{E}_{\mathbb{P}} [\mathbf{a}'_0(\tilde{s}, \tilde{\mathbf{z}})\mathbf{w} + \mathbf{b}'_0(\tilde{s}, \tilde{\mathbf{z}})\mathbf{x}(\tilde{s}) + \mathbf{c}'_0(\tilde{s})\mathbf{y}(\tilde{s}, \tilde{\mathbf{z}}) + d_0(\tilde{s}, \tilde{\mathbf{z}})] \\ \text{s.t.} \quad & \mathbf{a}'_m(s, \mathbf{z})\mathbf{w} + \mathbf{b}'_m(s, \mathbf{z})\mathbf{x}(s) + \mathbf{c}'_m(s)\mathbf{y}(s, \mathbf{z}) + d_m(s, \mathbf{z}) \leq 0 & \forall \mathbf{z} \in \mathcal{Z}_s, s \in [S], m \in \mathcal{M}_1, \\ & \sup_{\mathbb{P} \in \mathcal{F}} \mathbb{E}_{\mathbb{P}} [\mathbf{a}'_m(\tilde{s}, \tilde{\mathbf{z}})\mathbf{w} + \mathbf{b}'_m(\tilde{s}, \tilde{\mathbf{z}})\mathbf{x}(\tilde{s}) + \mathbf{c}'_m(\tilde{s})\mathbf{y}(\tilde{s}, \tilde{\mathbf{z}}) + d_m(\tilde{s}, \tilde{\mathbf{z}})] \leq 0 & \forall m \in \mathcal{M}_2, \\ & (\mathbf{w}, \mathbf{x}(s), \mathbf{y}^0(s), \dots, \mathbf{y}^{I_z}(s)) \in \mathcal{X}_s & \forall s \in [S] \\ & x_j \in \mathcal{A}(\mathcal{C}_x^j) & \forall j \in [J_x], \\ & y_j \in \bar{\mathcal{A}}(\mathcal{C}_y^j, \mathcal{I}_y^j) & \forall j \in [J_y]. \end{aligned}$$





其中,  $C_x^j, j \in [J_x]$ ,  $C_y^j, j \in [J_y]$  为 MECE 事件集合,  $\mathcal{I}_y^j, j \in [J_y]$  为信息集合。

### 5.5.2 事件式分布模糊集

事件式分布模糊集刻画了连续随机变量  $\tilde{z}$  和离散随机变量  $\tilde{s}$  的联合分布的分布性质, 包含了联合分布的分布信息。事件式分布模糊集取如下一般形式

$$\mathcal{F} = \left\{ \mathbb{P} \in \mathcal{P}_0(\mathbb{R}^{I_z} \times [S]) \mid \begin{array}{l} (\tilde{z}, \tilde{s}) \sim \mathbb{P} \\ \mathbb{E}_{\mathbb{P}}[\tilde{z} \mid \tilde{s} \in \mathcal{E}_k] \in Q_k \quad \forall k \in [K] \\ \mathbb{P}[\tilde{z} \in \mathcal{Z}_s \mid \tilde{s} = s] = 1 \quad \forall s \in [S] \\ \mathbb{P}[\tilde{s} = s] = p_s \quad \forall s \in [S] \\ \text{for some } \mathbf{p} \in \mathcal{P} \end{array} \right\}$$

其中,  $\mathcal{E}_k, k \in [K]$  为不同事件(注意, 这些事件不需要组成 MECE 事件集合),  $\mathcal{Z}_s, s \in [S]$ ,  $Q_k, k \in [K]$ , 和  $\mathcal{P} \subseteq \{\mathbf{p} \in \mathbb{R}_{++}^S \mid \sum_{s \in [S]} p_s = 1\}$  为封闭的凸集合。事件式分布模糊集声明了

- (1) 不同事件 ( $\mathcal{E}_k$ ) 下连续随机变量  $\tilde{z}$  的事件期望 (即条件期望)。
- (2) 不同情景 ( $s$ ) 下连续随机变量  $\tilde{z}$  的支撑集合 (即条件支撑集合)。
- (3) 不同情景 ( $s$ ) 发生的概率。

不确定集合  $Q_k, k \in [K]$  和  $\mathcal{P} \subseteq \{\mathbf{p} \in \mathbb{R}_{++}^S \mid \sum_{s \in [S]} p_s = 1\}$  分别允许条件信息 (1) 和 (3) 亦可以是不确定的。

Chen et al. (2019) 证明了事件式分布模糊集有非常好的普适性。它可以描述随机优化中常用的确定的离散分布 (deterministic discrete distribution), 以及分布鲁棒优化中用到的不确定的离散分布 (uncertain discrete distribution), 确定的 (或不确定的) 混合分布 (mixture distribution), 基于矩信息的分布模糊集 (moments ambiguity set), 以及数据驱动下 (1) 基于机器学习聚类或分类算法的分布模糊集 (K-means ambiguity set) 与 (2) 基于 Wasserstein 距离的分布模糊集 (Wasserstein ambiguity set)。

### 5.5.3 经典鲁棒优化转化

给定情景  $s$ , RSO 模型中目标函数和“软 (硬)”约束实际上是决策变量和连续随机变量  $\tilde{z}$  的取值  $z$  的双线性函数。因为, 我们可以将它们方便地记为

$$\mathbf{a}'_m(s, \mathbf{z})\mathbf{w} + \mathbf{b}'_m(s, \mathbf{z})\mathbf{x}(s) + \mathbf{c}'_m(s)\mathbf{y}(s, \mathbf{z}) + d_m(s, \mathbf{z}) \triangleq \mathbf{r}'(s)\mathbf{G}_m(s)\mathbf{z} + h_m(s) \quad \forall m \in [M] \cup \{0\}.$$

其中,  $\mathbf{G}_m(s) \in \mathbb{R}^{J_r \times I_z}$  和  $h_m(s) \in \mathbb{R}$  为参数。这样的双线性函数在事件式分布模糊集下的最坏期望可以通过求解一个经典鲁棒优化模型得到。换句话说, RSO 模型可以很方便地通过的配套建模工具包进行建模。目前, Chen et al. (2019) 论文中所提到的建模工具包 RSOME 的 MATLAB 版本 (<https://sites.google.com/view/rsome/home>) 和 Python 版本 (<https://xiongpengnus.github.io/rsome/>) 都已发布。读者可以下载进行测试, 并通过用户手册中的实例学习 RSO 的应用场景。在后续章节也将进行相关的介绍。



**定理 5.3: 模型等价转换**

最坏期望

$$\sup_{\mathbb{P} \in \mathcal{F}} \mathbb{E}_{\mathbb{P}} [\mathbf{r}'(\tilde{s}) \mathbf{G}_m(\tilde{s}) \tilde{\mathbf{z}} + h_m(\tilde{s})]$$

等于如下经典鲁棒优化模型的最优目标函数值

$$\begin{aligned} \inf \quad & \gamma \\ \text{s.t.} \quad & \gamma \geq \boldsymbol{\alpha}' \mathbf{p} + \sum_{k \in [K]} \boldsymbol{\beta}'_k \boldsymbol{\mu}_k & \forall \mathbf{p} \in \mathcal{P}, \frac{\boldsymbol{\mu}_k}{\sum_{s \in \mathcal{E}_k} p_s} \in \mathcal{Q}_k, k \in [K], \\ & \alpha_s + \sum_{k \in \mathcal{K}_s} \boldsymbol{\beta}'_k \mathbf{z} \geq \mathbf{r}'(s) \mathbf{G}_m(s) \mathbf{z} + h_m(s) & \forall \mathbf{z} \in \mathcal{Z}_s, s \in [S], \\ & \gamma \in \mathbb{R}, \boldsymbol{\alpha} \in \mathbb{R}^S, \boldsymbol{\beta}_k \in \mathbb{R}^{I_z} & \forall k \in [K], \end{aligned}$$

其中对每一个  $s \in [S]$ ,  $\mathcal{K}_s = \{k \in [K] \mid s \in \mathcal{E}_k\}$ .



## 本章参考文献



- Ben-Tal, Aharon, Alexander Goryashko, Elana Guslitzer, and Arkadi Nemirovski**, “Adjustable robust solutions of uncertain linear programs,” *Mathematical Programming*, 2004, 99 (2), 351–376.
- , **Laurent El Ghaoui, and Arkadi Nemirovski**, *Robust optimization*, Vol. 28, Princeton University Press, 2009.
- Bertsimas, Dimitris and Vineet Goyal**, “On the power and limitations of affine policies in two-stage adaptive optimization,” *Mathematical Programming*, 2012, 134 (2), 491–531.
- , **Dan A Iancu, and Pablo A Parrilo**, “Optimality of affine policies in multistage robust optimization,” *Mathematics of Operations Research*, 2010, 35 (2), 363–394.
- , **Dan Andrei Iancu, and Pablo A Parrilo**, “A hierarchy of near-optimal policies for multistage adaptive optimization,” *IEEE Transactions on Automatic Control*, 2011, 56 (12), 2809–2824.
- , **Melvyn Sim, and Meilin Zhang**, “Adaptive distributionally robust optimization,” *Management Science*, 2019, 65 (2), 604–618.
- Chen, Xin, Melvyn Sim, Peng Sun, and Jiawei Zhang**, “A linear decision-based approximation approach to stochastic programming,” *Operations Research*, 2008, 56 (2), 344–357.
- Chen, Zhi, Melvyn Sim, and Peng Xiong**, “Robust Stochastic Optimization: The Synergy of Robust Optimization and Stochastic Programming,” 2019, p. Available at *Optimization Online*.
- Dantzig, George B**, “Linear programming under uncertainty,” *Management Science*, 1955, 1 (3-4), 197–206.
- Delage, Erick and Dan A Iancu**, “Robust multistage decision making,” in “The operations research revolution,” *INFORMS*, 2015, pp. 20–46.
- Dyer, Martin and Leen Stougie**, “Computational complexity of stochastic programming problems,” *Mathematical Programming*, 2006, 106 (3), 423–432.
- Garstka, Stanley J and Roger J-B Wets**, “On decision rules in stochastic programming,” *Mathematical Programming*, 1974, 7 (1), 117–143.
- Georghiou, Angelos, Daniel Kuhn, and Wolfram Wiesemann**, “The decision rule approach to optimization under uncertainty: methodology and applications,” *Computational Management Science*, 2019, 16 (4), 545–576.
- Goh, Joel and Melvyn Sim**, “Distributionally robust optimization and its tractable approximations,” *Operations Research*, 2010, 58 (4-part-1), 902–917.
- He, Long, Zhenyu Hu, and Meilin Zhang**, “Robust repositioning for vehicle sharing,” *Manufacturing & Service Operations Management*, 2020, 22 (2), 241–256.
- Iancu, Dan A, Mayank Sharma, and Maxim Sviridenko**, “Supermodularity and affine policies in dynamic robust optimization,” *Operations Research*, 2013, 61 (4), 941–956.
- Perakis, Georgia, Melvyn Sim, Qinshen Tang, and Peng Xiong**, “Robust pricing and production with information partitioning and adaptation,” 2020.
- See, Chuen-Teck and Melvyn Sim**, “Robust approximation to multiperiod inventory management,” *Operations Research*, 2010, 58 (3), 583–594.
- Wiesemann, Wolfram, Daniel Kuhn, and Melvyn Sim**, “Distributionally robust convex optimization,” *Operations Research*, 2014, 62 (6), 1358–1376.